Interactive Ubiquitous Displays Based on Steerable Projection

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

vorgelegt von
Lübomira Spassova

Saarbrücken
17. Mai 2011

Dekan:
Prof. Dr. Holger Hermanns
Vorsitzender des Prüfungsausschusses:
Prof. Dr.-Ing. Philipp Slusallek
Berichterstatter:
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
Prof. Dr. Antonio Krüger
Promovierter akademischer Mitarbeiter der Fakultät:
Dr.-Ing. Ralf Jung
Tag des Kolloquiums:
14.06.2011

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Lübomira Spassova

Saarbrücken, 17. Mai 2011

Research serves to make building stones

out of stumbling blocks.

Arthur D. Little

Danksagung / Acknowledgment

Zunächst möchte ich mich an dieser Stelle bei meinem Doktorvater Herrn Professor Wolfgang Wahlster bedanken, der es mir ermöglicht hat, meine Promotion unter seiner Leitung durchzuführen und der mich vor allem in der anstrengenden Schlussphase sehr unterstützt hat. Einen großen Dank möchte ich außerdem meinem Zweitbetreuer Herrn Professor Antonio Krüger dafür aussprechen, dass er mir die Möglichkeit gegeben hat, meine Arbeit an seinem Lehrstuhl abzuschließen.

Ganz herzlich danke ich auch meinen Kollegen Boris Brandherm, Patrick Gebhard, Ralf Jung, Gerrit Kahl, Markus Löchtefeld und Michael Schmitz (und allen anderen, die ich an dieser Stelle vielleicht vergessen habe) für die vielen Anregungen und inspirierenden Diskussionen, die diese Arbeit wesentlich geprägt haben.

Nicht vergessen sollte man außerdem die Studenten, die im Rahmen ihrer Bachelor-, Master-, Diplom- oder Hiwi-Arbeit einen wichtigen Beitrag zur Realisierung dieser Arbeit geleistet haben: Carsten Greiveldinger, Fei Guo, Gerrit Kahl, Jens Müller, Johannes Schäfer, Torsten Spieldenner und Stilian Stanev.

Einen ganz lieben Dank an Suzanne und Jim Dowaliby, die diese Arbeit in kürzester Zeit sehr gründlich Korrektur gelesen haben.

Schließlich danke ich von ganzem Herzen meiner Familie und allen meinen Freunden, die mich während meiner Promotionszeit in jeder Hinsicht unterstützt und mir wenn nötig immer den Rücken gestärkt und freigehalten haben.

This work has been supported by the International Research Training Group *Language Technology and Cognitive Systems* at Saarland University.

Kurzzusammenfassung

Mit zunehmender Miniaturisierung der Computer und ihrer Einbettung in der physikalischen Umgebung werden neue Arten der visuellen Ausgabe notwendig. Im Bereich des Ubiquitous Computing (Rechnerallgegenwart) werden flexible und anpassungsfähige Displays benötigt, um eine Anzeige von visuellen Inhalten unmittelbar in der physikalischen Umgebung zu ermöglichen. In dieser Dissertation führen wir das Konzept des *Display-Kontinuums* und der *Virtuellen Displays* als Instrument der Mensch-Maschine-Interaktion ein. In diesem Zusammenhang präsentieren wir eine mögliche Display-Kontinuum-Realisierung, die auf der Verwendung steuerbarer Projektion basiert, und wir beschreiben mehrere verschiedene Interaktionsmethoden, mit denen man das Display-Kontinuum und die darauf platzierten Virtuellen Displays steuern kann.

Short Abstract

The ongoing miniaturization of computers and their embedding into the physical environment require new means of visual output. In the area of Ubiquitous Computing, flexible and adaptable display options are needed in order to enable the presentation of visual content in the physical environment. In this dissertation, we introduce the concepts of *Display Continuum* and *Virtual Displays* as new means of human-computer interaction. In this context, we present a realization of a Display Continuum based on steerable projection, and we describe a number of different interaction methods for manipulating this Display Continuum and the Virtual Displays placed on it.

Zusammenfassung

In dieser Arbeit wird die Konzeption und eine prototypische Realisierung eines Frameworks für Dynamische Ubiquitäre Virtuelle Displays (DUVDs) vorgestellt, die es erlauben, visuelle Inhalte auf geeignete Flächen einer entsprechend instrumentierten Umgebung zu platzieren und zu manipulieren. Obwohl die theoretischen Konzepte mit einer Vielzahl von Technologien realisiert werden können, wird in der vorliegenden Arbeit speziell eine projektionsbasierte Realisierung betrachtet.

In unserem theoretischen Framework definieren wir die Konzepte *Display Continuum*, *Virtual Display*, *Dynamic Peephole* und *Ubiquitous Cursor*. Für die Repräsentation und Visualisierung eines Display-Kontinuums haben wir ein 3D-Modell erstellt, das nicht nur potentielle Displayflächen enthält, sondern auch Unregelmäßigkeiten, wie Hindernisse, Schatten und Diskontinuitäten. Außerdem stellen wir ein theoretisches Modell für Dynamische Ubiquitäre Virtuelle Displays vor, welches die Basisparameter beschreibt, mit denen DUVDs definiert werden können, und wir zeigen auf, wie diese Parameter diskret oder kontinuierlich modifiziert werden können, um bestimmte Effekte zu erzeugen. In diesem Zusammenhang untersuchen wir ein breites Spektrum von Benutzerschnittstellen für DUVDs.

Die Interaktionsmodule, die im Zuge dieser Arbeit implementiert wurden, umfassen 3D-Interfaces und diverse Methoden zur Gestikinteraktion. Benutzerinteraktion in der realen Umgebung wurde auf unterschiedliche Arten umgesetzt: bildbasierte Interaktion wurde mit verschiedenen Kamerainstallationen realisiert, und Gestikinteraktion basierend auf Beschleunigungssensordaten wurde prototypisch anhand eines kommerziell erhältlichen Geräts (Wii Remote) implementiert. Die vorgestellten Interaktionskonzepte umfassen sowohl explizite als auch implizite Benutzereingabe.

Schließlich werden mehrere Beispielapplikationen vorgestellt, die die Anwendungsmöglichkeiten und Vorteile der DUVD-Konzepte für komplexe Präsentations- und Interaktionsaufgaben aufzeigen. Diese Applikationen verwenden sowohl benutzergesteuerte als auch systemgesteuerte Virtuelle Displays in verschiedenen Büro- und Supermarkt-Szenarien.

Contents

1 INTRODUCTION

With the increasing number of personal computers and other electronic devices, which are becoming part of our everyday lives, people are getting more and more used to human-computer interaction. Being surrounded by computers, mobile phones, video games, MP3 players, video cameras and numerous other types of digital technology, which are constantly emerging, a new generation of people is growing up, who have been familiar with technology from the very beginning of their lives. To describe this new generation, the writer and game designer Marc Prensky has coined the term *digital natives* ([Prensky, 2001]), which refers to persons who were born during or after the introduction of digital technology and who have an especially good understanding of its concepts through interacting with it from an early age on.

For these upcoming digital natives, modern technology is nothing intimidating, like it might have been for previous generations; quite the contrary, they are eager to gain the newest gadgets, and do not only work with them but also seek out opportunities to improve and enhance them. Due to this change in attitude, currently, novel interaction modes, such as (multi-)touch screens and voice or gesture control, rise in popularity and are becoming more and more accepted.

The novelties concerning the human-computer interaction also include an ongoing development of innovative output devices and concepts. Head-mounted displays, 3D screens, holographic and immersive displays are some of the recent technologies, which aim at improving the presentation of visual data. In this context, projection also plays a prominent role. Although its history can be traced back to the 15th century, projection has proved to be a very flexible means of displaying visual output, which has been constantly developed.

In the following pages, we will present a short history of the development and usages of projection throughout the centuries. Then, we will motivate the topic of the present work by means of a fictional scenario including some of the concepts and tools introduced later in this work; and finally, this chapter will conclude with a specification of the research questions and technical challenges addressed in the present work.

1.1 The Magic Lantern and the Early History of Projection

The *magic lantern* (also referred to as *laterna magica*) is a primitive projection device, which first appeared in the middle of the 17th century and is regarded as the early predecessor of the contemporary slide and overhead projectors. It consists of a concave mirror behind a light source, which directs the light rays through a transparent slide with an image painted onto it. Through a magnifying lens or a lens system, an enlarged version of the slide image is projected onto a screen in front of the apparatus. Figure 1.1 illustrates the working principle of an early magic lantern prototype with a candle as light source in front of the reflecting mirror and an image slide placed upside down between the candle and the optics resulting in an upright image. In order to increase the mystery of the device, in this illustration, the displayed figure seems to be projected on or appear out of mist.

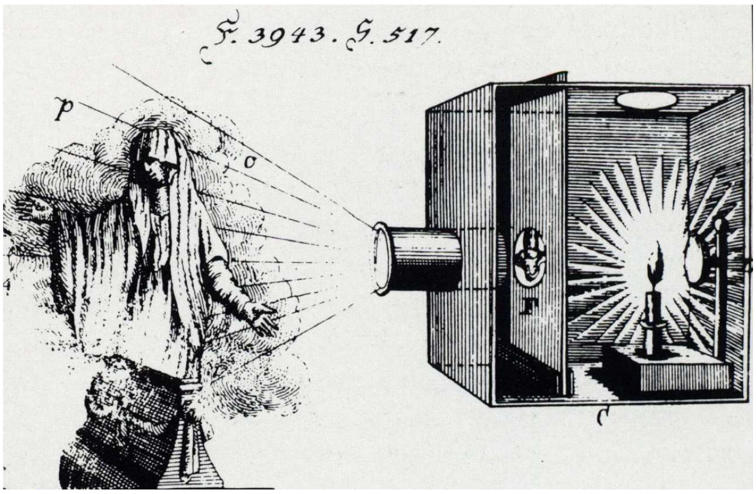


***Figure 1.1**: Illustration of the working principle of a magic lantern originating from a German economic-technological encyclopedia dating back to about 1800 (source: J. G. Krünitz, Oekonomisch-technologische Encyklopädie[2], volume 65, page 516)*

The light sources of the first magic lanterns were candles or oil lamps, which were quite inefficient and produced weak projections. A variation of the magic lantern using the sun as a light source is referred to as *solar microscope*. It uses a mirror for reflecting the sun rays through an optical lens system to produce magnified images of small translucent objects placed in it. With the invention of the *Argand lamp*[3] in the 1780s and the *limelight*[4] in the 1820s, the projection of brighter images was possible. Later, the inventions of the electric *arc*

[2]The encyclopedia started by Johann Georg Krünitz appeared in 242 volumes between 1773 and 1858; it represents one of the most significant scientific sources of that time.

[3]The Argand lamp was invented by Aimé Argand in 1780. It produces a light equivalent to about 6 to 10 candles.

[4]Limelight is a type of stage lighting that was used in theaters and music halls in former days. By directing an oxyhydrogen flame at a cylinder of quicklime (calcium oxide), an intense illumination is achieved.

lamp[5] in the 1850s and then the *incandescent electric lamps*[6] further improved the projected image of the magic lantern. A commercial example of a magic lantern produced at the beginning of the 20th century is illustrated in Figure 1.2.

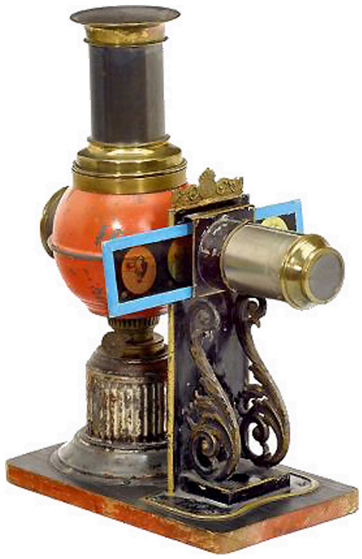


Figure 1.2: *A magic lantern manufactured by Ernst Plank & Company around 1900 in Nuremberg, Germany (source: http://babyloner.blogspot.com/2009/12/de-surnaturelles-apparitions.html)*

It is unclear who the inventor of the first magic lantern was. An early reference to a kind of very primitive projection instrument is found in *Liber Instrumentorum* by *Giovanni de Fontana* from about 1420. According to [Hankins and Silverman, 1999] (pp. 43-48), the German Jesuit priest *Athanasius Kircher* and the prominent Dutch physicist *Christiaan Huygens* both published descriptions and illustrations of magic lantern devices in the middle of the 17th century, and Huygens was most probably the one who established the term *laterna magica*. Although at that time, the optics of Huygens' magic lantern were already essentially identical to those of modern projectors, the early laternists did not yet have a clear idea of what to use this device for.

As the name *laterna magica* already implies, the first magic lanterns were mostly exploited by self-appointed magicians and showmen, and their purpose was to project fuzzy images of devils, skeletons, ghosts and goblins in order to frighten and fascinate people. However, there are also reports of the use of projection instruments to change the appearance of subjects' clothes, which reminds very much of the modern idea of *Augmented Reality* (see Section 2.1). Later, in the 18th century, the optics and mechanics of the magic lantern were improved, so that the French physicist *Étienne Gaspard Robertson* could develop his famous and for those days very impressing projection show *Phantasmagoria*, in which he introduced

[5]Arc lamps produce light by an electric arc (also called a voltaic arc), which appears between two electrodes placed in a glass bulb containing a noble gas.

[6]The incandescent lamp is a source of electric light invented by Thomas Edison, which works by heating a metal filament wire to a high temperature until it glows.

such novelties like back-projection onto a translucent screen or even a simple movement of the projected image by mounting a magic lantern on wheels. In this way, it was possible to move the lantern toward the screen and thus create the impression that the projected object was heading for the audience, which is a primitive realization of zooming.

Figure 1.3*: The Laterna Magica (about 1760), a painting by Paul Sandby. (British Museum, London, England; source: http://www.lib-art.com/artgallery/3986-the-laterna-magica-paul-sandby.html)*

Magic and entertainment were, however, not the only purposes magic lanterns have been used for. At the end of the 18th century, magic lanterns and solar microscopes slowly started finding their way into education and science, where they were used to demonstrate physical phenomena, astronomical diagrams and other educational contents. For this purpose, not only pictures on glass slides were projected but also actual physical objects and phenomena, like e.g. the blood circulation of a frog. A painting showing how a magic lantern presentation was performed in the middle of the 18th century can be seen in Figure 1.3.

The history of the magic lantern shows that our ancestors were also fascinated by the idea of enhancing and modifying their world by projection. With technical progress, the magic lantern principle resulted in the development of the contemporary overhead projector and slide projector, and with some further technology in the movie projector and the cathode ray tube of television screens and computer monitors. Nowadays, projectors are, on the one hand, more and more miniaturized and integrated into handheld devices, like mobile phones, watches and cameras. On the other hand, the light intensity and the image resolution of standard video projectors is constantly increasing. In addition to the traditional DLP and LCD projector technologies, new approaches, such as LED and laser projectors, are emerging. Projections in public spaces are becoming more and more common, e.g. in form of advertisements on floors and buildings, art installations, like in the Berlin *Festival of Lights* (see Figure 1.4 [left]) and spherical projected displays, like the one developed in the *Science on a Sphere* project[7] (see Figure 1.4 [right]).

[7] Science On a Sphere is a large projection system used to display animated data onto the outside of a sphere.

***Figure 1.4**: Illuminated Berlin Cathedral as part of the Festival of Lights in 2009 [left]; Science on a Sphere: projection of planetary data on a spherical display (sources: http://de.wikipedia.org/wiki/Festival_of_Lights, author: Michael F. Mehnert; http://sos.noaa.gov/)*

Almost any kind of surface and material is to some extent suitable for projection – even clouds are often used as projection screens for advertising purposes, and there are also systems using water and fog curtains for projection-based entertainment installations, such as the Flowscreen[8] or the FogScreen[9]. With a special transparent multilayer projection screen (TransScreen[10]), it is even possible to display 3D holographic animated images floating in space. Finally, also the retina of the human eye can be used as a projection surface, e.g. with the monocle-like EyeTap[11] device applying an extremely miniaturized projector to display a computer-generated image superimposing the original imagery the wearer is seeing by projecting it directly onto his retina.

All the above examples show that, although in modern society, the principle of projection is an everyday occurrence and not a mystery anymore, like it was at the time of its invention, the fascination of the *laterna magica* remains, and "modern magicians" still continue to experiment with new approaches of exploiting projection to entertain, educate and support people in their daily lives.

1.2 Application Scenario and Motivation

The following future scenario is intended to illustrate how projection technologies and interaction concepts, some of which are actually available today, can be used to support our everyday lives in the near future. It describes an ordinary weekday of the working mother Mrs. Smith.

Mrs. Smith wakes up in the morning to the sound of her alarm clock. She opens her eyes and sees the current time projected on the ceiling straight above her. It's 6:30 in the morning – time to get up and prepare for work. While entering the kitchen, she notices a projected

[8] http://www.technifex.com/pages/products/products_flowscreen.html
[9] http://www.fogscreen.com
[10] http://www.laser-magic.com
[11] http://eyetap.org/

note appearing beside the fridge. It's from her husband asking her to fetch a bottle of wine on her way home from work for tonight's dinner. Mrs. Smith makes herself a cup of coffee and takes a seat at the kitchen table for a short breakfast. While she is eating, she reads the latest news projected right in front of her on the table. She can browse through the different news articles using simple hand gestures. Shortly before leaving the table, Mrs. Smith lets the projection system display her schedule for the day and sees that she has a meeting with her boss at 2 p.m. in his office.

When Mrs. Smith enters her office, the ubiquitous system recognizes her presence and sets up her projected desktop at the usual place in front of her desk. For interaction with the projected interfaces, Mrs. Smith uses a virtual keyboard projected on the table and hand gestures instead of a mouse. After a while, she leaves her office to have a cup of coffee and discuss a current problem with a colleague in the employees' lounge. While they are sitting there, the tracking system recognizes Mrs. Smith's current location, and when she receives important emails (according to the preferences in her user profile), she is notified by projected messages appearing on the table in front of her. In this way, Mrs. Smith can keep track of her email account.

At 1:45 on this afternoon, while she is busy finishing a report, Mrs. Smith receives a projected reminder of the upcoming meeting with her boss, Mr. Mayer. She interrupts her current work and heads for Mr. Mayer's office. Arriving there, she finds the door closed, but a projected note appearing at her presence informs her that Mr. Mayer will be back in 10 minutes. Although, Mr. Mayer left the note for Mrs. Smith half an hour ago, the projected message has automatically adapted to the current time. In order to take advantage of the delay, Mrs. Smith makes her desktop appear projected on the table in front of Mr. Mayer's office and she continues her work while she is waiting. After a while, Mrs. Smith and her boss are sitting in Mr. Mayer's office, and Mrs. Smith requests the display of her desktop, which is then projected in front of her, so that she can show Mr. Mayer the current state of her work.

After the meeting, Mrs. Smith wants to make copies of some documents using the lab's new copier. When she approaches the device, the assistance system recognizes that Mrs. Smith is not yet familiar with this copier and so she is proactively offered usage instructions on a projected display above the device. Furthermore, the display also shows Mrs. Smith's current expenses for copies for this month.

In the meantime, Mrs. Smith's daughter Mary is on a school excursion to a museum of natural history. On the way there, Mary observes their bus driver using a head-up display projected onto the windscreen providing driving instructions and other important hints. While they are driving through a tunnel, Mary is captivated by the imagery projected on the walls, turning the inside of the tunnel into an art installation. When Mary's school class arrives at the museum, they are welcomed by a projected character which guides them from exhibit to exhibit. The character moves along the walls and tells the children interesting facts about the animals and plants that they see on the way. If there are specific parts of the exhibits that should be given special attention, the character morphs into a bright ball and moves onto the respective area thus highlighting it. As Mary is highly fascinated by the giraffes, she remains longer in front of their exhibit while the other kids walk on following the character. Thanks to the tracking system, however, the projected character recognizes that Mary was left behind, and a clone of him is sent to fetch her and guide her back to her group.

After work, Mrs. Smith goes to the nearby supermarket to buy some things for dinner and especially the wine which Mr. Smith has asked for. As she is in a hurry, at the entrance, she registers for the newly installed ubiquitous guiding system, which she has already used a few times before. As her customer profile is already stored in the supermarket's system, Mrs. Smith just has to use her fingerprint for registration. The system now already knows the products which Mrs. Smith has put on her electronic shopping list and guides her on the shortest path through the market projecting arrows on the floor in front of her which show her the way to the desired products. As soon as Mrs. Smith approaches a shelf with a product she is searching for, this product is highlighted by a projected spot, so that Mrs. Smith finds it immediately. As the assistance system recognizes that Mrs. Smith is in a hurry, she is not presented projected hints about new products that she might be interested in according to her customer profile. Finally, Mrs. Smith arrives at the wine department where a digital assistant helps her choose the appropriate wine for dinner by displaying information about the wines she is taking out projected onto a free area on the shelf.

Back at home, Mrs. Smith starts cooking while Mary talks about her day at the museum. Later, Mary would like to learn more about the giraffes, so she selects a children's program about these animals on television. However, she does not want to watch it alone in the living room while her mother is cooking in the kitchen, so she moves the projected television screen from the living room to the kitchen and places it above the dinner table.

At about 9 p.m., while Mr. and Mrs. Smith are still sitting with their dinner guests in the living room, Mary is already lying in her bed, eagerly listening to her bedtime story teller projected on the wall beside her bed. She knows that she can move in any position in her bed and the projected buddy will always follow her view, and he'll keep telling her stories until she has fallen asleep.

This is an imaginary scenario which might become reality by means of the projected Dynamic Ubiquitous Virtual Displays concept presented in this work. The theoretical background and some approaches for their implementation in a few exemplary prototypes will be described in the following chapters. The main motivation behind this approach is to offer users the possibility to have ubiquitous access to visual information in a natural and intuitive way. One way to achieve this goal is the exploitation of steerable projection, which allows the display of visual content on any suitable surface in the line of sight of the steerable projection device. A critical view of the advantages and disadvantages of steerable projection is given in Section 2.5. The next two sections define the formal research questions and technical challenges which this work aims to answer and solve.

1.3 Research Questions

This section presents an overview of the main research questions concerning the development of (projection-based) ubiquitous display systems, which are discussed in this thesis.

- Which functionalities have to be supported by a ubiquitous display system?

  Ubiquitous displays are more embedded into the physical environment than traditional screens and, depending on their technical realization, they can be more flexible in terms of positioning, dimensions, form and content. In the present work, we investigate how a ubiquitous display system can be established in an appropriately equipped environment and which functionalities such a system should offer in order to support a user working in and with it. Particular attention has to be devoted to the possibility of dynamically modifying the positions of ubiquitous displays in the environment. Furthermore, functionalities like spontaneous creation, deletion and storage, which are also uncommon for conventional screens, have to be taken into account when working with ubiquitous displays.

- Which methods and interfaces are suitable for interaction with ubiquitous displays?

  As mentioned above, ubiquitous display systems are supposed to offer novel functionalities, which have to be supported by appropriate user interfaces. Therefore, an investigation as to which common interaction techniques can be adapted for use with ubiquitous displays must be made. Moreover, new interaction metaphors and approaches have to be developed in order to enable interaction in the physical space, in which ubiquitous displays are embedded, and to cope with the special characteristics and limitations of the respective technical realization.

- Which theoretical models can be used to describe ubiquitous display systems?

  In order to be able to develop applications involving ubiquitous displays as an output modality, we need a method of describing the properties and capabilities of these displays in a theoretical model. As ubiquitous displays represent a new concept of output interfaces with characteristics that are not common for traditional screens, conventional models are insufficient to describe them. Therefore, new models have to be developed, which take into account the special characteristics of ubiquitous displays and the functionalities which a ubiquitous display system is supposed to offer.

- How can projection-based ubiquitous displays be combined with physical screens?

  In the context of interaction, particular attention can be payed to the junction between projection-based ubiquitous display visualization and conventional monitors. It is important to investigate if and how these two different display types can be combined. The difficulty lies in the fact that, though from the perceptual point of view both display types are very similar, there is an immense difference concerning their technical and conceptual realizations. Thus, it is a particular challenge to find a way of seamlessly combining projected and conventional displays, which appear natural and intuitive to the user.

- Which contextual knowledge is needed when working with ubiquitous displays?

  In any human-computer interaction, context plays an important role. When a user performs a primitive action, like e.g. a mouse click, it is crucial to know in which situation and under which conditions this interaction takes place in order to find the right interpretation of the user's intention. The use of context knowledge enables the triggering of a high number of actions with a relatively small set of input parameters, which allows for a more intuitive and simple interaction.

In a traditional desktop scenario, the main context knowledge which is considered is the current position of the mouse pointer and the locations of application windows on the desktop. When ubiquitous displays are considered, both knowledge about the user as well as knowledge about the environment in which the interaction is performed has to be taken into account.

- Which projection-based display systems are offered by other research projects and how can they be classified?

  When developing new approaches, it is crucial to have an overview of previous and ongoing research in the corresponding field in order to be able to improve already existing methods or rather develop new innovative solutions. For this purpose, a systematic analysis and classification of related projects is beneficial.

1.4 Technical Challenges

Aside from the conceptual research questions concerning the development of ubiquitous display systems, there are also technical challenges which have to be considered regarding the realization of such systems. As already mentioned, in the present work, we concentrate on the technical realization of a projection-based ubiquitous display system, which involves a steerable projector unit and a variety of interaction devices.

- How can a steerable projector system be installed in the environment?

  In order to be able to augment an environment with projection-based ubiquitous displays, a steerable projector has to be installed in the environment and calibrated to it, which encompasses the adjustment of both extrinsic and intrinsic parameters. Beside the appropriate hardware, the steerable projection system must encompass a corresponding software framework enabling a straightforward manipulation of the system.

- Which input devices can be applied in order to implement the new concepts for interaction with projection-based ubiquitous displays?

  Apart from the classical mouse and keyboard devices, innovative input methods like acceleration-based or vision-based interaction devices can be adapted for the manipulation of projection-based ubiquitous displays. This technical challenge is closely related to the research question of the conceptual methods and interfaces appropriate for interaction with ubiquitous displays addressed in the previous section.

- Which components/macros are needed in order to build projection-based ubiquitous display scenarios?

  In order to enable the development of application scenarios involving the use of projection-based ubiquitous displays, suitable development and simulation tools have to be provided. These tools should offer the ability to manipulate relevant parameters of the implemented ubiquitous displays and to adapt them to the respective application setup.

1.5 Organization of the Thesis

After the introduction and motivation of the present work, the remainder of this thesis is structured as follows. In the first part of this work, the conceptual and technical background, which is crucial for the understanding of the presented concepts and approaches, will be discussed. After that, Part II will give an overview of projection-based display systems spanning a broad range of different technologies encompassing immersive projection setups, large-scale static multi-projector displays and several steerable projection systems. After this survey, in Part III, we will present the concepts behind the introduced Dynamic Ubiquitous Virtual Displays, and we will take a closer look at the realization of the proposed concepts based on a number of implementations. Finally, we will conclude with a summary of the developed concepts and achieved results, and we will give an outlook on possible future development based on the present work.

Part I

Background and Basic Concepts

2 Conceptual Background

This chapter addresses the background knowledge which is needed for the understanding of the concepts and implementation of the system presented in this work. First of all, the topic of the present work will be embedded into the various research fields it is related to. After that, some further basic concepts which are relevant to the topic of this work will be outlined.

2.1 Embedding into the Research Context

With the miniaturization of computing devices and their increasing distribution in more and more areas of our working and everyday lives in the last decades, new research areas in computer science have emerged which aim at investigating different aspects of this ongoing evolution. This process requires the development of new models and metaphors for these newly emerging computing environments and the user interaction in and with them.

In the following, some of these research areas are illuminated and related to each other, and we outline how the subject of this thesis fits into these particular research fields.

The term **Virtual Reality (VR)** refers to computer-generated environments which can simulate parts of the real world as well as represent completely fictitious worlds which are supposed to appear as realistic as possible to the user. In his book "Virtual Reality" from 1991 ([Rheingold, 1991]), Howard Rheingold, one of the pioneers in this research area, describes VR is an experience in which the user is "surrounded by a three-dimensional computer-generated representation, and is able to move around in the virtual world and see it from different angles, to reach into it, grab it, and reshape it". The inventors of the CAVE Automated Virtual Environment (see Section 4.1.1), Cruz-Neira et al., propose a more technical definition which is also more confined to the visual domain: They describe a VR system as one "which provides real-time viewer-centered head-tracking perspective with a large angle of view, interactive control, and binocular display" ([Cruz-Neira et al., 1993]). In either case, the user is supposed to immerse himself – virtually or even physically – into a simulated, synthetic environment while being completely isolated from the real world. In order to achieve such immersive impressions, VR environments can be displayed using special devices, like closed head-mounted displays (HMDs, without see-through capability) or complex physical setups encompassing several monitors or projection screens surrounding the user, as in the previously mentioned CAVE.

In contrast to the VR approach, which is isolating the user from the real world, in recent years, research focuses more and more on combining virtual and real environments, in order to have the benefits of both worlds. Such approaches aiming at enhancing the user's environment with virtual objects are often referred to as **Augmented Reality (AR)** or **Mixed Reality (MR)**. In their paper [Milgram and Kishino, 1994], Milgram and Kishino define a **Virtuality Continuum (VC)**, which encompasses each possible variation of mixing real and virtual worlds – from a completely real environment to an entirely virtual world (see Figure 2.1). Thereby, they use the term Augmented Reality to describe environments which are closer to the real world (i.e. real environments augmented with virtual objects) and Augmented Virtuality (AV) to denote environments which have more virtual characteristics (i.e. virtual environments augmented with real objects). In this case, the term Mixed Reality encompasses every variation of AR and AV, whereby a clear distinction between both is not always possible. Typical AR examples are applications using optical or video see-through HMDs, augmented video streams on handheld devices or projected augmentation. In the first two cases, the user is supposed to wear or hold a device in order to see the augmentation, whereas in the latter case, the augmentation hardware is separated from the user. This special type of AR is often referred to as **Spatial Augmented Reality (SAR)** ([Bimber and Raskar, 2005]) as a contrast to the more traditional body-attached AR.

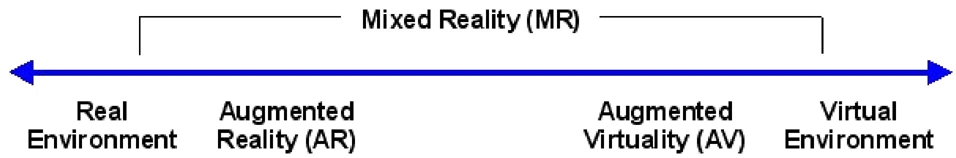


Figure 2.1: *Milgram's Virtuality Continuum (source: [Milgram and Kishino, 1994])*

An alternative view on AR and MR is proposed by Lifton and Paradiso in [Lifton and Paradiso, 2009]. In their environmental taxonomy, they establish the following definitions:

- **Reality** is simply life in the absence of virtual representations of the world;
- **Augmented Reality** has all aspects of reality, as well as an "information prosthetic" which overlays normally invisible information onto real objects;
- **Mixed Reality** would be incomplete without both its real and virtual components (e.g. a television studio with a blue screen installation, which is only "complete" when virtual background is overlaid);
- **Virtual Reality** contains only elements generated by a computer in an attempt to mimic aspects of the real world.

These different reality variations represent different states along the real-virtual axis, which indicates how much reality and virtuality is contained in the respective environment (see Figure 2.2 [left]).

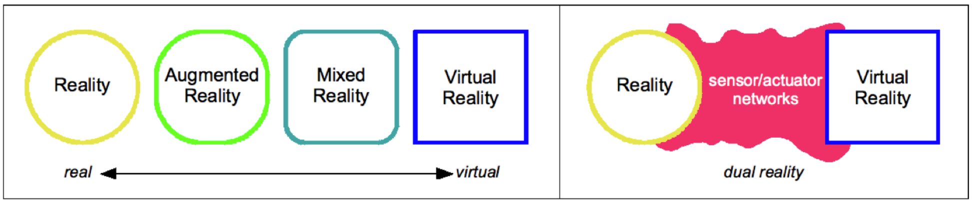


***Figure 2.2**: Lifton's environmental taxonomy [left] and the Dual Reality concept [right] (source: [Lifton and Paradiso, 2009])*

Lifton and Paradiso complement their taxonomy by a declaration of the concept of **Dual Reality (DR)**, which they define as an environment resulting from the interplay between the real world and the virtual world, mediated by networks of sensors and actuators (see Figure 2.2 [right]). In this sense, a DR environment comprises a complete reality and a complete virtual reality space, whereby both enhance each other by their ability to mutually reflect, influence and merge into each other. Ideally, this process is bi-directional, i.e. sensed data from the real world can be used to enrich the virtual world and vice versa.

Recently, Stahl et al. have extended Lifton's Dual Reality paradigm by introducing the concept of **Synchronized Realities** ([Stahl et al., 2011]). It generalizes the Dual Reality idea to any combination of real and virtual worlds that mutually influence each other. This particularly means that remote physical environments can be connected to each other and synchronized, e.g. in order to achieve a feeling of social connectedness between geographically distant family members.

The term **Ubiquitous Computing (UC)** describes a post-desktop paradigm of human-computer interaction. It was coined by Mark Weiser around 1988, who at that time was Chief Technologist of the Xerox Palo Alto Research Center (PARC). In his famous forward-looking and much-cited article [Weiser, 1991], which has inspired many researchers since its first publishing in 1991, Weiser presents his revolutionary ideas of the development of computing technology at the beginning of the 21st century. According to his vision, the computing devices would rapidly shrink in size while their performance and quantity will significantly increase, which has proved true thus far. The most important point in Weiser's UC concept is the seamless embedding of hundreds of miniaturized, context-aware and networked computing devices in the everyday environment. In this context, Weiser also uses the term **Embodied Virtuality** referring to "the process of drawing computers out of their electronic shells" thus bringing the "virtuality" of computer-readable data into the physical world.

In a Ubiquitous Computing (or Embodied Virtuality) environment with a variety of devices integrated into everyday objects, the users will be unobtrusively supported by the embedded instrumentation while scarcely being aware of working with computers. Despite the high amount of instrumentation, however, people working in a UC environment should not be mentally overloaded, rather, they should be enabled to accomplish their tasks faster and with less strain than without the support by the Embodied Virtuality. This computational model is often also described as **Pervasive Computing** or **Ambient Intelligence**, where each concept focuses on slightly different aspects. While the former term emphasizes the diffusion of computation into the physical environment and is mostly used in an industrial context, the latter phrase pays particular attention to the calmness and unobtrusiveness of the embedded instrumentation.

Recently, the terms **Hybrid Reality** or **Multiple Reality** describe an environment in which, based on special instrumentation, different users can perceive individual augmentation according to their current context. In [Castronovo et al., 2011], e.g., the authors present a multimodal conspicuity-enhancement system for e-bikes, which allows motorbikes to be highlighted in the visual and acoustic perception of car drivers in the vicinity, while at the same time, the appearance of the motorbikes remains unmodified for other traffic participants.

One main aspect of Ubiquitous Computing, on which Weiser particularly focuses in his article, is the appropriate deployment of visual displays. Although computers are supposed to disappear, users should still have the opportunity to interact with them and receive appropriate visual feedback. Ubiquitous computing displays should, however, not be restricted to static computer desktops, and interaction with them should go beyond traditional keyboard and mouse manipulation. Weiser proposes the use of physical Ubiquitous Computing displays, which he classifies in three different types according to their size, namely *tabs*, which are inch-scale machines (similar to today's smartphones), *pads*, which are about the size of a sheet of paper (analogous to the iPad) and *boards*, which are yard-scale displays (the predecessors of today's electronic whiteboards). The idea behind these display devices was to free the user from the locally bound computer desktops and thus to enable the display of information and interaction with it in the physical environment immediately where needed.

An alternative way of achieving this aim of bringing visual information and interaction with it into the environment is the use of projected displays. This idea is also picked up by Raskar et al. in their famous article about the *Office of the Future* ([Raskar et al., 1998]), in which they propose the instrumentation of an office environment with "smart" projectors and computer-controlled cameras in order to create "spatially immersive display surfaces" at designated locations in the user's surroundings.

An intriguing suggestion for an interaction metaphor for Ubiquitous Computing is proposed by James Scott in [Scott, 2005]. In order to achieve user interaction in a UC environment which should be intuitive for the user, he proposes to develop interaction metaphors which resemble the skills of superheroes, like telekinesis (action at a distance), teleathesia (sensing at a distance), telepresence and precognition/postcognition. According to Scott, one of the enabling techniques needed to accomplish this approach is the realization of ubiquitous display, e.g. using steerable projectors in order to allow a whole room to be used as an interaction environment. In this way, e.g., Superman's X-ray vision through walls and objects could be realized.

One further research filed which is related to the area of Ubiquitous Computing is referred to as **Nomadic Computing**. As the term already suggests, this research area focuses on the mobile and nomadic character of current and future computing technology. In contrast to the previously common – and still largely spread – view on computers as associated with their desktop peripherals (in general monitor, keyboard and mouse), the Nomadic Computing paradigm envisions an environment which offers highly flexible access to computational services and data with automatic adjustment to the currently available processing capabilities. Kleinrock – one of the pioneers of Nomadic Computing – defines *Nomadicity* as "the system support needed to provide a rich set of capabilities and services to nomads as they move from place to place in a transparent, integrated and convenient form" ([Kleinrock, 1995a]). In order to enable Nomadic Computing, highly adjustable, and user- and resource-adaptive systems have to be developed, which require a common system architecture and generic protocols for nomadicity. The area of Nomadic Computing is multidisciplinary, and one of its most important disciplines concerns advanced visualization applications including nomadic visual output opportunities ([Kleinrock, 1995b]).

In conclusion, it can be stated that the main focus of the present work lies in the area of Ubiquitous Computing and Augmented Reality as it aims at offering an almost seamless instrumentation of the user's environment (UC), which enables the deployment of so-called Virtual Displays using projected augmentation (AR) (see Section 5.2). Since the augmentation in this case is realized by devices integrated in the environment, it is a form of Spatial Augmented Reality. Furthermore, as the system presented in this work combines a real environment with virtual objects forming a virtual model of the former, enabling interactivity in both directions, it can also be seen as a Dual Reality system. Finally, this work contributes to the area of Nomadic Computing as it presents approaches to create visual user interfaces for nomadic use.

2.2 Smart / Instrumented Environments

Smart Environments (SEs) or **Instrumented Environments (IEs)** are physical spaces with embedded computation and instrumentation, like sensors, actuators and displays. They build the technical background and can act as testbeds for Ubiquitous Computing, Augmented Reality and other research areas mentioned in the previous section, thus enabling the transfer of novel technologies into everyday life.

Cook and Das define a *Smart Environment* as "one that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment" ([Cook and Das, 2004], [Cook and Das, 2007]). According to them, a SE is composed of four layers:

- *physical layer*: sensors, actuators, other physical devices and their corresponding interfaces;
- *communication layer*: infrastructure for data exchange and remote device control;
- *information layer*: databases, data mining, user modeling, etc;
- *decision layer*: e.g. rule engine, decision maker, etc.

Michael H. Coen describes *Intelligent Environments* as "spaces in which computation is seamlessly used to enhance ordinary activity" ([Coen, 1998b]). Coen argues that Intelligent Environments differ from the Ubiquitous Computing approach in that their instrumentation should not be embedded into everyday objects (like e.g. chairs), but the interaction-enabling devices should be integrated into the environment itself in the form of unobtrusive cameras and microphones. This instrumentation in combination with computer vision and speech recognition approaches allows for the creation of implicit human-computer interfaces, with which people can interact with the environment in natural ways without being aware of the fact that they are interacting with computers. Instead of making computer interfaces for people, Coen's vision is to create people interfaces for computers.

Coen has realized some of his ideas in the *Intelligent Room* project at MIT ([Coen, 1998a], [Brooks, 1997]). It is a physical room comprising several monitors and projected displays, computer controllable lights, curtains and audio system, and a large array of video cameras for recognizing human interaction. With these cameras and special computer vision and tracking techniques, the Intelligent Room can observe the user's location, recognize his or her current context and make assumptions about his or her intents. The Room then tries to react in an appropriate way to the user's behavior. If e.g. a person enters the room and lies down on the sofa after shutting the door, the Intelligent Room assumes that this person wants to relax and automatically dims the lights, closes the curtains and plays soft, calm music in the background. Besides having this implicit interaction, the user is also provided the opportunity to interact explicitly with the Intelligent Room by pointing with his finger or a laser pointer. As interaction surfaces, there are two projected screens and a table, which is rendered interactive only by camera observation. In addition to the vision-based interaction, the user can also use spoken language to interact with the room.

During the development and deployment of the Intelligent Room, its creators were faced with difficulties regarding e.g. the sensitivity of the camera tracking system and the trade-off between a large recognition grammar and the accuracy of speech recognition. The greatest challenge, however, turned out to be the development of a truly natural interaction opportunity with the embedded room instrumentation, especially for multimodal conditions. The developers of the system observed that, in some cases, it was very difficult for the users to remember what type of utterances were accepted for interaction or in which order some complex actions had to be performed in order to achieve the desired result.

A formal definition of an *Instrumented Environment* is provided by Michael Schneider in his PhD thesis ([Schneider, 2010]). He defines an Instrumented Environment as a quadruple (E, P ,D ,I), which represents the four central elements that characterize each IE with:

- *E*: *spatial extension* of the IE in the physical world;
- *P*: *purpose* of the IE;
- *D*: set of *digital items* like data and applications (with no physical appearance) which comprise the *virtual layer* of the IE;
- *I*: set of *physical items* like sensors, actuators and communication infrastructure components (all with physical appearance) which comprise the IE's *instrumentation*.

Furthermore, Schneider's definition constitutes that *D* and *I* together support the realization of *P*, and it restricts *E*, *D*, and *I* to the necessary minimums needed to realize the purpose *P*.

Instrumented Environments can range from private to public and from fixed to mobile, and there is a diversity of possible IE settings, like homes, offices, supermarkets, hospitals, public spaces, vehicles, etc. Accordingly, the technology embedded in these different environments can vary in quantity, type and complexity depending on the purpose of the corresponding IE. The system presented in this work has been deployed in two different types of IEs: an office setting and a retail environment.

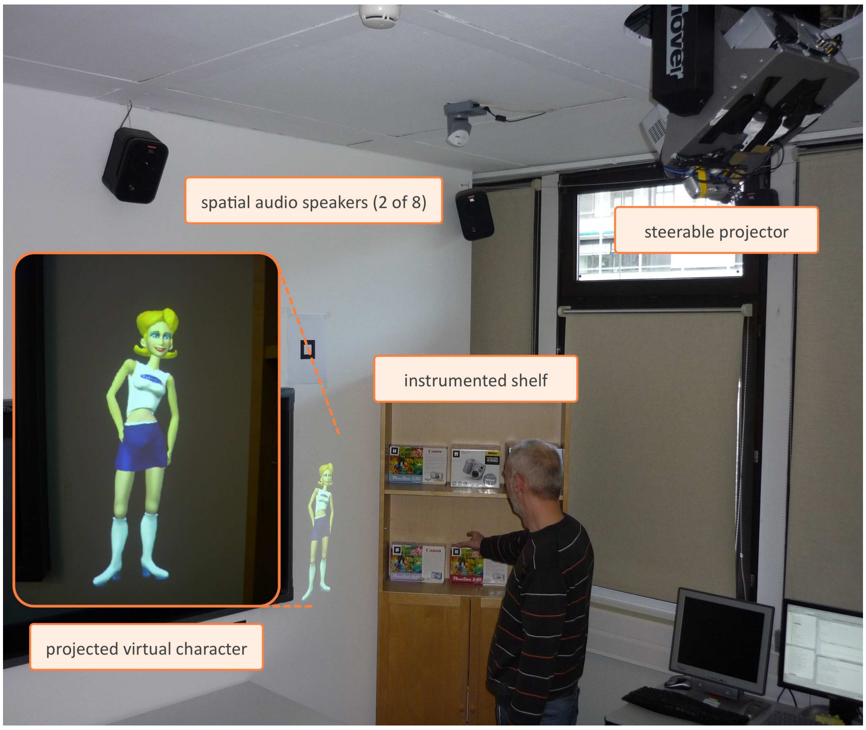


Figure 2.3: *Instrumentation of the SUPIE environment with projected virtual character*

The *Saarland University Pervasive Instrumented Environment (SUPIE)* is an example of an office-like IE (see Figure 2.3). In addition to a steerable projector unit (see Section 3.1), this Instrumented Environment also encompasses a spatial audio system (SAFIR) ([Schmitz and Butz, 2006]), an RFID-instrumented shelf containing objects fitted with RFID tags and/or visual markers, and also several cameras and displays. Moreover, there is an infrastructure of active RFID tags mounted on the ceiling of the room and infrared beacons placed at strategic locations, which are used for egocentric user localization and tracking with the Always Best Positioned (ABP) localization system ([Schwartz et al., 2005]), which allows the fusion of directional information from different sensor types using geo-referenced dynamic Bayesian Networks.

The *Innovative Retail Laboratory (IRL)*[1] is a supermarket-like IE of the German Research Center for Artificial Intelligence (DFKI)[2] (see Figure 2.4), in which application-oriented research is conducted ([Spassova et al., 2009], [Krüger et al., 2010]). Among others, it encompasses several instrumented shelves fitted with RFID antennas and RFID-tagged products, an instrumented shopping cart designed to assist customers during their shopping and an easy checkout system. Beside the simple identification of individual product items, the RFID labels attached to the products at the IRL are exploited as carriers of so-called Semantic Product Memories (SemProM, [Kröner et al., 2009], [Kröner et al., 2010]), which contain product-related information. A steerable projector unit has been mounted on the ceiling of

[1] http://www.innovative-retail.de
[2] http://www.dfki.de

the IRL, in order to enable the display of visual information for customers and supermarket staff.

***Figure 2.4**: Instrumentation of the IRL with projected navigation hint*

Aside from the IRL, the DFKI has established four more so-called *Living Labs* representing different types of environments, in which researchers can test, evaluate and demonstrate their technologies: in the *SmartFactory*, novel approaches for achieving flexible, self-organizing, and user-oriented industrial automation are being developed; the *Bremen Ambient Assisted Living Lab (BAAL)* is an apartment equipped with assistant systems for the elderly and people with physical or cognitive impairments; in the *Robotic Exploration* lab, robotic systems are tested under controllable and reproducible conditions; and the *Virtual Office* lab offers a variety of hard- and software methods to support people in their knowledge-intensive work. Although, the approaches presented in this work have not been applied in one of these four labs so far, they represent potential application areas for ubiquitous projected displays.

2.3 Selective Visual Attention and the Peephole Metaphor

As the present work deals with the representation of visual content, it is important to provide a brief background of some basic concepts concerning the way people visually perceive their surroundings.

Visual perception is the ability to interpret visual information by the effects of visible light reaching the eye. The act of seeing does not only refer to the physical synthesis of an image onto the light-sensitive membrane in the back of the human eye, called the retina, when light emitted or reflected by an object reaches the eye, rather, it also encompasses the complex cognitive process of recognizing this particular object as such, which takes place in the human brain.

In this context, *attention* is referred to as the cognitive process of perceiving one aspect of the environment while ignoring others, and *selective attention* is the focusing of one's conscious awareness on one particular stimulus. In cognitive psychology, *visual attention* is regarded as a two-stage process. In the first stage, attention is distributed uniformly over the visual scene, while the perceived information is being processed in parallel. In the second stage, attention is focused on one particular area of the visual scene, and processing is performed serially ([Jonides, 1983]).

Sensory overload is a state in which one or more of the senses are strained, making it difficult to focus one's selective attention on one particular stimulus or task. When concerning the visual perception, a sensory overload can be caused by an overwhelming multitude of different forms and colors, by blinking or hectically moving objects. In order to avoid such a visual overload in Ubiquitous Computing systems, the visual output of these systems has to be designed in a calm and unobtrusive way, so that at a certain point in time, the user is confronted with as much as necessary and as little as possible visual information. One way to achieve this goal is offered by the peephole metaphor.

The *peephole* concept was primarily presented in [Yee, 2003], which builds upon prior research by Fitzmaurice et al. ([Fitzmaurice, 1993], [Fitzmaurice et al., 1993]) on situated information spaces and spatially aware handheld devices. In his article, Yee defines a "peephole" as a movable window on a flat virtual workspace, which is larger than the currently visible part shown by the peephole. The peephole display is realized using a handheld device which can be tracked in space, so that users can explore the workspace by moving the device. The approach allows two-handed pen interaction with the displayed information in a number of different applications like e.g. a drawing program, a map viewer and a calendar. When the user moves the device, the displayed visual content is panned accordingly in the opposite direction, which creates the impression of a spatially anchored virtual workspace. As a fundamental concept of his work, Yee points out the possibility of concurrent navigation and interaction. This approach mainly focuses on interaction with one interface at a time which is larger than the peephole display. In contrast, the present work examines the placement of and interaction with a number of visual objects in 3D space, which are in general smaller than the visualizing peephole.

In a later work, Butz et al. take up the peephole concept developed by Yee and specify a *generalized peephole metaphor* as a model of interaction in Augmented Reality and Instrumented Environments where the physical space is superimposed by a virtual layer on which virtual windows can be opened to display or gather information ([Butz and Krüger, 2003], [Butz and Krüger, 2006]). The authors distinguish between different types of peepholes depending on the type of media (e.g. visual or acoustic), the originator of the peephole (user- vs. system-initiated) and the direction of data flow (output vs. input peepholes). This approach forms the basis for the Display Continuum and Dynamic Peephole concepts described in the present work (see Sections 5.1 and 5.4).

2.4 Introduction to Human-Computer Interaction

Human-computer interaction (HCI) is the study of interaction between people (users) and computer systems. The link between users and computers – and thus the interaction means between them – is represented by the *user interface* (or simply *interface*), which in general encompasses both software and hardware aspects. Usually, an interface provides a means of *user input* offering the possibility for users to manipulate a computer system and/or *system output* allowing the system to indicate the effects of the user's manipulation or to proactively offer information to the user.

In this section, we give a short overview on the development of user interfaces for HCI, especially focusing on the ones relevant for the present work.

2.4.1 GUIs and WIMP Paradigm

In the late 1970s and the early 1980s, the Graphical User Interface (GUI) emerged as a fundamentally novel concept of human-computer interaction. By the use of graphic icons and a pointing device, GUIs offer users a comfortable and particularly easy and intuitive way of controlling computers. In the decades since their introduction, GUIs have undergone an incremental refinement built on some fundamental core principles. Several GUI pioneers have created their own windowing systems. However, they all have some basic elements in common that define the *WIMP paradigm*. The acronym WIMP stands for "window, icon, menu, pointing device" denoting a style of interaction using these elements.

The introduction of the GUI paradigm has opened up new possibilities for human-computer interaction especially for non-expert users, and in this way, it has made a significant contribution to the acceptance and thus increasing integration of computers in people's everyday lives. However, with the recent development of Ubiquitous Computing systems, the WIMP paradigm is reaching its limits. The increasing distribution of computation into the user's environment demands a shift of the user interfaces off the desktop screen into the physical world. This is the beginning of the Post-WIMP era of human-computer interfaces ([Nielsen, 1993]). Applications for which WIMP interfaces are not suitable are, e.g., those that work with devices that provide continuous input signals, show 3D models or simply visualize an interaction for which there is no defined standard widget.

2.4.2 Post-WIMP HCI

In a visionary article from 1993, Jakob Nielsen, one of the experts in HCI, proposed a new type of user interfaces, which he calls *noncommand-based interfaces* ([Nielsen, 1993]), which would be operable in an unobtrusive way. In this context, Nielsen distinguishes between four different types of noncommand-based interfaces, namely eyetracking-based, music-based, virtual characters and embedded help.

Tangible User Interfaces (TUIs) are physical interfaces with which a person can interact with digital information. Alternative names for such interfaces are *graspable interfaces*, *physical interfaces* and *embodied interfaces* among others. One of the pioneers in tangible user interfaces is Hiroshi Ishii, the head of the Tangible Media Group at MIT. He is the inventor of a special form of TUIs, called Tangible Bits, which are intended to give physical form to digital information, making it directly manipulable and perceptible ([Ishii and Ullmer, 1997], [Ishii, 2008]).

Gesture recognition aims at interpreting human gestures with computing algorithms in order to enable human-computer interaction in a natural and intuitive way. Gestures can originate from any part of the human body but, commonly, the face or hands are considered. Current approaches in this field include emotion recognition from the face, and hand gesture recognition. Hand gestures can be performed in the form of static postures or motion. The main techniques used for gesture recognition are either based on computer vision or on motion and location sensors, like accelerometers or gyroscopes. Ideally, a gesture interface avoids an instrumentation of the user, thus letting him move freely during interaction. Some approaches, however, make use of wearable devices, mainly in the form of data gloves, with which the user's hand position and orientation, and the movement and postures of the fingers can be tracked very precisely (e.g. CyberGlove[3], P5 Glove[4], Power Glove[5] and many others).

Both tangible user interfaces and gestural interfaces can be regarded as special means for performing *3D interaction*. This is a newly emerging form of HCI where users are able to move and perform interaction in 3D space. In 3D interaction, the physical positions of elements of both human- and machine-issued information in the 3D space is relevant. In this context, the 3D interaction space can be either a physical one or a virtual model.

In [Jacob et al., 2008], the authors propose a framework for *reality-based interaction*, which focuses on the following four themes from the real world:

- *Naive physics*: people have common sense knowledge about the physical world.
- *Body awareness and skills*: people have an awareness of their own physical bodies and possess skills for controlling and coordinating their bodies.
- *Environment awareness and skills*: people have a sense of their surroundings and possess skills for negotiating, manipulating, and navigating within their environment.
- *Social awareness and skills*: people are generally aware of others in their environment and have skills for interacting with them.

According to Jacob at al., interface design concerning these four aspects is supposed to result in reality-based interfaces which should give users the impression of interacting with real-world objects and not with digital ones, which is supposed to increase their ease and joy of use.

[3]http://www.cyberglovesystems.com/
[4]http://www.vrealities.com/P5.html
[5]http://en.wikipedia.org/wiki/Power_Glove

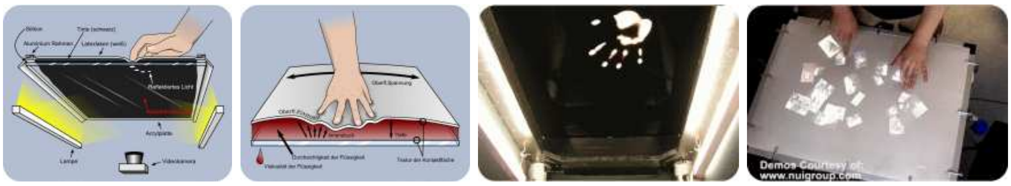


Figure 2.5*: Black Magic malleable projection surface: hardware setup, working principle, hand interaction observed by the camera, and projected visual feedback [from left to right] (source: http://www.fluidum.org/projects_blackmagic.shtml)*

Influenced by the emerging Ubiquitous Computing systems, there is a gradual movement from human-computer interaction toward *Human-Environment Interaction (HEI)*, with which users do not operate with individual computing devices but rather with a complex Instrumented Environment ([Encarnação, 2007]). While the first HCI systems have offered more explicit interaction methods with the corresponding devices, HEI provides more implicit interfaces, which can be operated with less cognitive load. In this context, *multimodal interfaces* offer different forms of processing input and output, such as text, images, speech and gestures. In a simple form of multimodal interfaces, the different modalities can be used alternatively, and in more elaborate interfaces, several input or output modalities can be combined in order to resolve ambiguities, which might occur during interaction, like e.g. in the SmartKom system, which combines speech, gesture and facial expression for input and output as described in [Wahlster, 2002]. This approach is referred to as *modality fusion*. In contrast, *modality fission* denotes the inverse functionality of modality fusion, since it refers to a mapping of the communicative intention of a system onto coordinated multimodal output. Finally, if all input modes of a multimodal system are also available for output and vice versa, we speak of *symmetric multimodality* ([Wahlster, 2003], [Wasinger and Wahlster, 2006]).

Another innovative development in HCI and HEI is the design of interfaces with animal- and human-like characteristics. These so-called *anthropomorphic interfaces* can range from simple talking objects to complex human-like characters ([Schmitz, 2010]). This concept is based on the observation that people often tend to treat objects and complex systems as if they were humans.

Organic interfaces are defined as such “with non-planar displays that may actively or passively change shape via analog physical inputs” ([Vertegaal and Poupyrev, 2008]). The term encompasses not only flexible electronic paper displays but also tangible physical interfaces with transformable and thus naturally adaptable shapes. One interesting example of an organic touch display is the so-called Black Magic[6] surface, which detects touch and pressure by liquid displacement inside a malleable surface ([Hilliges et al., 2008]). The deformation of the surface is observed with a camera placed under the table. In this way, the deformable surface allows the recognition of touch gestures with multiple fingers, whole hands and other object outlines. A projector mounted above the interactive surface displays system feedback on the table (see Figure 2.5).

With the ongoing development of *brain-computer interfaces* (or *neural interfaces*), possibly, in the near future, traditional physical interaction devices will become obsolete, as these

[6] http://www.fluidum.org/projects_blackmagic.shtml

interfaces will allow control of computer systems by thoughts. Recently, the first commercial brain-computer interface (EPOC[7]) was released, which was designed to be used as an input interface for gaming.

2.4.3 Interaction with Projected Displays and Widgets

After the overview of possible current and future interaction devices, in this section, we take a closer look at interfaces and interaction techniques used for working with projected displays.

In an early prototype of an augmented environment comprising projected table and wall displays, Rekimoto et al. propose a method for extending the displays of users' laptops to the adjacent projection surfaces ([Rekimoto and Saitoh, 1999]). Using camera-based object recognition, a laptop placed on a table is automatically integrated into the environment, so that the user can drag visual content from the laptop's display to the projection surface next to it. This interaction technique called *hyperdragging* uses a so-called *anchored cursor*, which is visualized as a projected line between the originating laptop and the current cursor position, as soon as the cursor of the integrated laptop is moved to the projection surface. In this way, the user is supported in keeping track of his current cursor location in the environment.

A similar approach to extending the visual space of laptops is pursued in the *Bonfire* system ([Kane et al., 2009]), where two projector camera pairs attached on both sides of the laptop provide two additional display and interaction surfaces on the underlying table to the left and right of the laptop. Using camera vision techniques (supported by input from the laptop's acceleration sensor), Bonfire enables four types of manual gestures: *tapping*, *dragging*, *flicking* and *crossing*. Moreover, simple color histogram-based object recognition is implemented, which enables both direct and indirect object interaction. The indirect object interaction in this case consists in capturing indications of the current user context and generating appropriate system reactions to its changes.

In conjunction with the rear-projection-based *Stanford Interactive Mural* (see Section 4.2.2), Guimbretière et al. have developed and tested different pen-based interaction techniques for large projected wall-displays ([Guimbretière et al., 2001]). The *Flow and Go* interaction combines handwritten character input with *FlowMenu* selection and object motion. In this context, *FlowMenu* is a kind of pie menu, with which the user can specify complex input through a set of submenus using one continuous stroke ([Guimbretière and Winograd, 2000]). Another interaction metaphor, called *ZoomScapes* allows users to scale specific visual objects by simply moving them into an appropriate region on the screen.

An interactive large *Wall Display* developed by Ashdown and Robinson [Ashdown and Robinson, 2001] combines a projected interface delivered by a static projector and up to four gesture input devices in order to investigate direct interaction opportunities. The position and orientation of each input device (*crayon*) are detected by an electromagnetic motion tracking system (Polhemus FastTrak), and the device also offers two buttons for triggering system commands.

Aside from the described wall setups, desk installations are often used for realizing interactive projection. The *Limpid Desk* e.g. is a projection-based mixed reality installation on

[7] http://www.emotiv.com/

which physical documents can be virtually rendered transparent by a projected overlay of the underlying content after the user has placed his hand on the topmost document of a stack of papers. The authors also propose an alternative interaction technique to easily reveal the content of a stack: after touching the upper layer document, the thumbnail images of all lower layers are projected on top of the document ([Iwai et al., 2006], [Iwai and Sato, 2006]).

Another interactive desk is the *Escritoire* ([Ashdown and Robinson, 2003b], [Ashdown and Robinson, 2003a], [Ashdown and Robinson, 2005]), which comprises a surface augmented by two projectors and was developed as a follow-up of the previously described *Wall Display*. One of the projectors is placed behind the desk projecting upward, whereas the other one is mounted above the desk. Both projector beams are reflected down to the desk surface by two mirrors. In this way, a so called *foveal display* is created, which consists of a large, low-resolution image (*periphery*) created by the first projector and an overlapping small, high-resolution area in front of the user (*fovea*) delivered by the second projector, in which the displayed items (virtual sheets of paper) can be dragged in order to obtain a more detailed representation. The desk interface allows the arrangement and annotation of digital paper in the form of images and text documents alongside with VNC streams. The system offers two-handed interaction opportunities with a combination of a desk-sized digitizer and stylus for the dominant hand and an ultrasonic whiteboard pen for the non-dominant hand. Thus, the user can position items on the desk with the non-dominant hand, while the dominant hand performs more detailed tasks, like writing or drawing. A client-server architecture supports distributed multi-user collaboration. The displayed items have a z-order, which determines their order of appearance on the interactive desk, thus adding the notion of piles to the interface.

An intriguing approach to providing high-resolution visual content on a restricted projection surface is the use of so-called *Display Bubbles* ([Cotting and Gross, 2006]). They represent freeform shapes projected on a table, which can be freely defined using a laser pointer. The assigned visual content, which is normally rectangular, is warped in order to fit into the shape of the bubble. The warping function leaves the center of the image mostly undistorted (focus), while the image distortion is increased towards the boundaries of the shape.

Further approaches to interacting with projected displays encompass the use of tracked movable surfaces, e.g. the one described in [Lee et al., 2005], which is tracked by means of embedded light sensors; moreover, gaze interaction ([San Agustin et al., 2010]) and NFC-based interaction ([Hardy et al., 2010], [Broll et al., 2010]) with projected visual content has been proposed.

Aside from the static projector setups described before, there are numerous approaches to using handheld projectors for interaction, like e.g. the iLamps, used for projected augmentation ([Raskar et al., 2005]), or the tracked handheld projector units described in [Cao and Balakrishnan, 2006] and [Cao et al., 2007]. The Map Torchlight application allows the augmentation of physical maps with digital information using a spatially aware handheld projector ([Schöning et al., 2009]).

Furthermore, miniaturized projector-camera units can be attached to the user's body in order to enable interaction with the user's environment during everyday life or work, as for example with the *Sixth Sense* device ([Mistry and Maes, 2009]). In the military project called

Interactive Dirt, soldiers are fitted with body-worn projector-camera devices (attached to the shoulder), which can be used to display e.g. position plans during military operations. Different vision-based interaction techniques have been tested in this project, including IR emitter sticks, reflective tape sticks, reflective tape on fingers, and laser pointers.

2.5 Characteristics of Steerable Projection

In recent Ubiquitous Computing research, projection has increasingly been regarded as a powerful means of displaying visual content and system feedback. Several steerable projection systems using a fixed projector and a pan/tilt mirror ([Pinhanez, 2001a]), or a projector placed in a pan/tilt unit ([Yang et al., 2001a], [Borkowski et al., 2005], [Molyneaux et al., 2007], [Ehnes et al., 2004]) have been developed aiming at the realization of projection-based Augmented Reality in everyday environments (see also Section 4.4). In [Ashdown and Sato, 2005], Ashdown et al. define a steerable projector as "a digital projector whose beam can be moved under computer control to illuminate different objects in its environment". In contrast to head-mounted displays, steerable projection devices free the user of cumbersome instrumentation and thus do not distract him from his surroundings or disable him in his normal work.

Display Technology	Mobility	Occlusion Problems	Simultaneous Users	Space Requirement
Traditional Monitors	fixed	no	single	small, desktop
Rear-Projected Display Walls	fixed	no	multiple	large space, behind display
Front-Projected Display Walls	fixed / ad-hoc creation	(yes) (depending on projection angle)	multiple	large space, in front of display
Steerable Projection Systems	fixed hardware, but steerable image	(yes) (depending on projection angle)	multiple (?)	small space for hardware, line of sight
Mobile Projectors	portable	no	single	small

Table 2.1: *Comparison of ubiquitous display technologies (adapted from [Molyneaux and Kortuem, 2004])*

In [Molyneaux and Kortuem, 2004], Molyneaux et al. present an overview and comparison of several fixed and portable ubiquitous display technologies including different projection setups (front, rear, steerable and mobile) with traditional monitors (see Table 2.1). The big advantage of traditional monitors and rear-projection displays is that these technologies do not face any occlusion problems – at least not by the users themselves but possibly by other people or objects. However, traditional monitors only provide a spatially restricted visual space, which is mostly detached from the physical environment. Although there are efforts to embed visual displays in walls ([Guimbretière et al., 2001], [Li et al., 2000a]), floors ([Grønbæk et al., 2007], [Iversen et al., 2007]), furniture ([Streitz et al., 2002], [Prante et al., 2004]) and other everyday objects ([Siio et al., 2003]), these installations are still restricted to specific surfaces and mostly require a high amount of instrumentation. Even with the possible future invention of a thin, flexible and easy to handle e-wallpaper, which will allow the plastering of whole rooms with wall-sized displays, it is still far from reality to have e-paper displays on all kinds of simple objects.

Therefore, it appears reasonable to exploit steerable projection for the realization of ubiquitous displays, as most of the developed concepts, metaphors and interaction techniques will be adaptable to future e-wallpaper displays. Moreover, the projection technology itself offers a number of advantages that traditional monitors cannot provide so far. In the following, an overview of some important steerable projection characteristics is provided.

- **Overlay:** One of the most outstanding features of projected displays is the fact that, in contrast to traditional screens, they can be easily created as an overlay on top of almost any surface or object. These potential projection surfaces can range from interior or exterior walls, desk surfaces and furniture even to products in supermarket shelves.

- **Spontaneous creation:** Normally, it takes some effort to install traditional screens at some desired locations. In contrast, when working with projection, virtual projected screens can be created on the fly at any possible position as long as it lies within the range of a steerable projector. The only precondition is to have an appropriate projection system installed in the environment.

- **Non-planarity:** In general, the creation of projected display is not restricted to planar surfaces. In contrast to physical monitors, which are normally planar, projected visual content can easily be adapted to arbitrarily shaped surfaces by appropriately pre-warping the projected image.

- **Transparency:** Apart from their semi-transparency property, which allows for creating see-through displays that superimpose but do not occlude the underlying surface, projected displays can also contain regions of real transparency[8]. This property makes projected displays more flexible as it enables the visualization of more complex content.

- **Frameless:** Due to their transparency property, projected displays are in principle frameless. This means that projected displays can be created in any shape, and their shapes can be even changed spontaneously. Thus it is possible to visualize for example freeform virtual characters, which can be animated so that they change their posture, shape and position in space.

- **Movable:** When using a steerable projection unit or a steerable mirror, the projector beam can be directed at different locations and moved continuously through the

[8] As the projection of "black" is actually realized by the absence of light, in black projection regions, the underlying surface is left completely unmodified (apart from some residual light).

environment. In this way, with an appropriate distortion correction, the created projected displays can be moved along the projection surfaces. Thus, the position of the displayed visual content can be changed in space, so that it can be e.g. adapted to the current user location or to the location of an object it refers to. In particular, it is possible to let the projected content appear to be stuck on a tracked moving object.

Although projection offers a lot of flexibility as a means of displaying visual content, its advantages are contrasted with some drawbacks, which are summarized in the following.

- **Insufficient brightness:** Depending on the light intensity of the deployed projector, its distance to the targeted projection surface and lighting conditions in the surroundings, the displayed visual content can be rather dark and thus poorly visible. Especially with small handheld projectors, this is often a serious issue, which currently can only be overcome by reducing the projection distance or by darkening the light of the surroundings. With the ongoing improvement of projection hardware, however, in the future, this problem will become less and less severe.

- **Inappropriate projection surfaces:** Ideally, projection surfaces are white (or at least plain-colored), planar and orthogonal to the projector beam. Obviously, these conditions are only fulfilled in a very few special cases. Oblique projection on planar surfaces can be compensated for by an appropriate predistortion of the projected image, so that the resulting projected image appears undistorted to the viewers (see Section 2.6). When projecting on nonplanar surfaces, the image distortion can also be compensated for but only for a single point of view, and also colored and textured surfaces can, to a certain extent, be used for projection of an appropriately color-corrected image ([Bimber et al., 2005]). In a multi-projector setup, the luminance in the overlapping regions is multiplied by the number of overlapping projectors and results in an unnaturally high brightness in these areas. To overcome this problem, appropriate alpha masks can be applied to each of the projected images, which reduce the luminance in the respective regions (photometric correlation, [Raskar et al., 1999]). However, transparent and highly reflective surfaces, like e.g. windows and glossy desk surfaces, still remain unsuitable for projection.

- **Shadow casting / occlusion:** A problem limiting the usability of projection in many scenarios is the fact that there must be a line of sight between the projection device and the aimed projection surface. When applying front projection in an interactive setup where the user is standing in front of the projection surface, the projected image is very likely to be at least partially shadowed by the user, which would disturb the usability of the application. In order to avoid this occlusion, projectors are often mounted in such a way that the projector beam hits the projection surface in a very low angle of incident. In this way, the volume of the projector beam is minimized so that it becomes less likely that it interferes with the user. Despite this minimization, the impact angle must still remain greater than zero, so that occlusion cannot be entirely avoided in front projection setups.

- **Restricted display area:** The size of projected images is limited by the boundaries of the projector beam, which itself depends on the beam angle of the applied projector and its distance and orientation to the projected surface. The beam angle is a characteristic of the respective projection device and normally can be modified only to a certain extent by a possible zoom functionality of the projector. A larger beam angle leads to a larger projection. When working with steerable projection, the distance to the projection surface and the projection angle often vary depending on the adjustment of the projection device. A greater distance to the projection surface and a low angle of incident of the projector beam lead to a larger projection. In any case, the size of the possible projected image is limited by the boundaries of the projector beam on the projection surface. However, compared with traditional monitors, which have very strict boundaries, steerable projection devices are still more flexible in terms of display size.

2.6 Distortion Correction for Projected Images

As already mentioned in the previous section, one problem when working with steerable projection is the fact that in most cases the aimed projection surface will not be perpendicular to the projector beam, which would result in a distorted image. There are two basic approaches which can be applied in order to overcome this problem. The first one consists in pre-warping the image to be displayed by applying appropriate geometric transformations to the pixels of the original image. These transformations represent mappings of the pixels in the source image plane to corresponding points on the dedicated projection surface. In case of a planar projection surface, this mapping is a *homography*, which can be determined from only four pixel correspondences ([Sukthankar et al., 2001]).

For more complex setups, like e.g. curved surfaces or surfaces containing edges and bumps, more elaborate warping techniques are needed, like e.g. the pixel displacement maps proposed in [Bimber et al., 2005] or the approach using a mesh of feature points described in [Yang et al., 2001b]. In these cases, the correct image warping is computed using a camera which observes the projected image. An approach for camera-based capturing of pre-warping mappings for projected image correction using imperceptible structured light is proposed in [Raskar et al., 1998]. Other systems for camera-based image correction using visible structured light are described e.g. in [Grossberg et al., 2004], and for multi-projector setups in [Raskar et al., 1999], [Yang et al., 1999], [Yang et al., 2001a] and [Raij et al., 2003]. In [Borkowski et al., 2003], the authors describe an approach for projecting perspectively corrected images on a moving planar surface in real time. The presented system uses a camera to track the boundaries of the movable screen and adapts the homography matrix to the current position and orientation of the surface to the projector (see also Section 4.4.4).

A survey and discussion of different camera-based geometric as well as photometric registration techniques for multi-projector setups is presented in [Brown et al., 2005]. The article outlines approaches for planar and arbitrarily shaped display surfaces addressing their respective advantages and drawbacks.

However, when projecting on surfaces containing discontinuities, like edges or gaps, the image correction can result in an image that appears undistorted only from a single point of view (sweet spot), which is inappropriate for a setup in which multiple users can be present or for a user with a frequently changing viewing position who either cannot be tracked at all or only roughly tracked. In order to obtain images that appear perspectively correct from each user location, the images have to be projected at appropriate planar or curved surfaces. Only if a surface is appropriate to physically draw an image on it in the real world, would it also be possible to project a perspectively corrected image onto it. An approach for image correction which takes this fact into account is the virtual camera method presented in [Pinhanez, 2001a] (see also Section 3.1.2).

This method is also exploited for image correction in the Fluid Beam system, which is used as the basic hardware and software platform in this work. A more detailed explanation of the virtual camera method and a description of the hardware and software of the Fluid Beam system is presented in Section 3.1.

3 Technical Background

3.1 Fluid Beam System

The starting point of the system developed in the present work is the *Fluid Beam* system, which has been implemented in the course of a diploma thesis ([Spassova, 2004b], [Spassova, 2004a]) in order to enable distortion-free projection on different surfaces in appropriately instrumented environments. With this system, it is possible to create so-called projected displays, on which images, videos and video streams can be presented. These projected displays can be placed (i.e. stored) at fixed locations in the environment, where they appear when the projector beam is directed at them. The displays can also be moved over the projection surfaces in the environment and thus offer a basis for various ubiquitous applications.

3.1.1 Fluid Beam Hardware

The *Fluid Beam* system uses the steerable projector unit *beaMover*[1] (see Figure 3.1) with a high-resolution digital camera mounted on it. The moving yoke of the beaMover unit has 2 degrees of freedom (pan and tilt), hence the device can be directed at almost every position in its surroundings. The range of the yoke is 340° in pan direction and 270° in tilt direction. The projector of the beaMover unit uses LCD technology and its light intensity is, depending on the model, 3.300 or 6.500 ANSI lumens respectively, which is bright enough to create projected images that are clearly visible even in daylight conditions.

The attached camera can be controlled remotely via USB and has a maximum image resolution of 5 megapixels. The pictures are mainly used for optical marker recognition, like e.g. in the *SearchLight* application described in Section 6.4.1. In addition, the camera can also deliver a low resolution video stream (320 x 240 pixels), which can be used to recognize interaction with projected widgets ([Reiter and Butz, 2005]).

The movable unit is connected to a PC via a USB/DMX interface. The DMX512 standard is a communication protocol mainly used to control stage lighting. It offers up to 512 channels with an 8-bit resolution. When controlling the beaMover unit, 2 channels are used to define the pan and tilt angle respectively, which guarantees smooth motion of the device and high position repeatability.

[1] http://www.beamover.com/

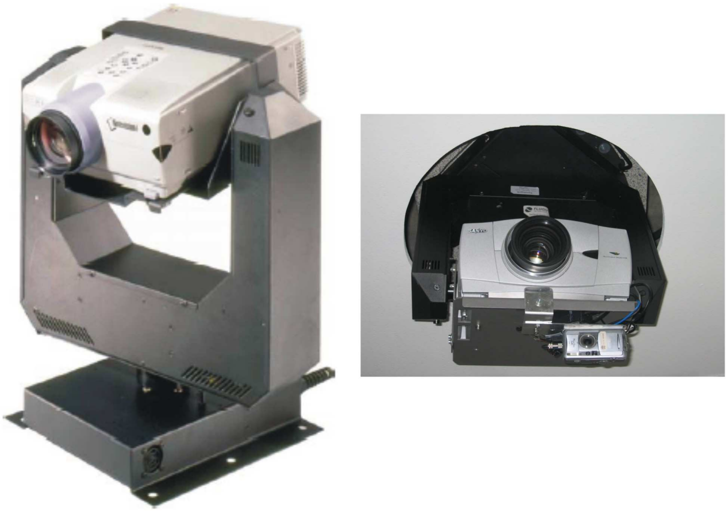


***Figure 3.1**: beaMover [left] (source: www.beamover.com) and the Fluid Beam device mounted on the ceiling [right]*

One instance of this *Fluid Beam* device has been mounted in the center of the ceiling of the Saarland University Pervasive Instrumented Environment (SUPIE) (see Section 2.2), where it is part of the ubiquitous instrumentation. There, it enables e.g. the projection of a virtual character serving as an assistant for this Intelligent Environment (see Section 6.3.3)

A second Fluid Beam device has been installed at the Innovative Retail Laboratory (IRL) (see Section 2.2), where it is used to project visual navigation hints and advertising at appropriate locations in the environment, like e.g. shelves and walls. One application that uses the *Fluid Beam* unit in a shopping scenario is the so called *micro navigation* (see Section 6.3.2).

3.1.2 Fluid Beam Software

The *Fluid Beam* software controls the positioning and movement of the *Fluid Beam* device and delivers a predistorted image in order to compensate for oblique projection. The predistortion method is based on the fact that, from a geometrical point of view, the process of viewing is the inverse of the process of projecting. This means that, a camera and a projector with the same optical parameters induce the respectively opposite distortion when viewing/projecting an image from the same position and orientation. Thus, an image distortion due to oblique projection can be compensated for by projecting this image as viewed by an appropriately positioned and oriented camera with corresponding intrinsic parameters. This can be realized using a simulated virtual camera in a virtual 3D model of the respective physical environment (see Figure 3.2).

In order to exploit this property, a 3D model of the environment in which projected displays are to be shown is created. This model contains all potential display surfaces, and it also includes a virtual camera which simulates the real-world *Fluid Beam* projector by mimicking its intrinsic and extrinsic parameters. This means that the optical parameters of the

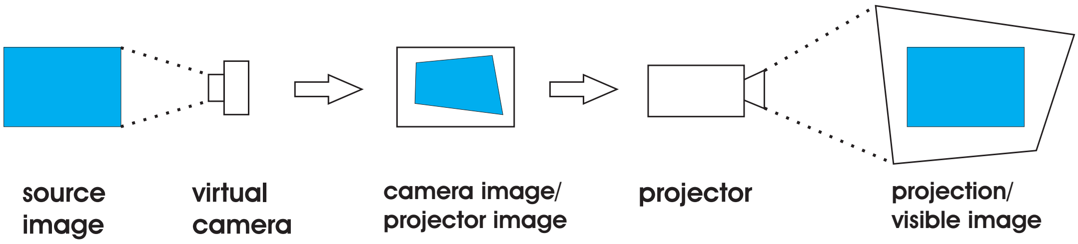


Figure 3.2: *Distortion compensation by means of a virtual camera*

virtual camera correspond to those of the projector, and its position and orientation in the 3D model is synchronized with the projector's position and orientation in the real world.

Using the method described above, it is possible to place visual content in the 3D model, which is displayed in the form of projected images at the corresponding positions in the real environment when the *Fluid Beam* unit is directed at them. The *Fluid Beam* software offers methods for adjusting the orientation of the steerable unit and also for defining the position and movement of the projected displays along the surfaces of the environment and for defining their visual content.

The *Steerable* interface provides a number of methods for defining the movement of the steerable projector beam (peephole) to a specified position in 3D space within a certain time interval. The aimed position can be specified either by the rotation angles of the steerable unit (pan and tilt) or by a 3D point in a reference coordinate system.

During the movement of the steerable unit, the focus of the projector is constantly being adapted according to its distance to the surface that it is currently directed at. In this way, the projected content always remains focused.

The *DisplayFactory* interface provides methods for creating projected displays at the surfaces of the environment and for specifying their size, position and orientation. It also offers the ability to define the visual content which is to be shown on a projected display, which can be either an image, a video or a video stream. Further, there are methods for discretely or continuously changing a projected display's position and/or orientation. The display's position change can either be synchronized with a corresponding projector movement or it can be independent of the projector orientation. In the former case, the projected display would stay visible during its movement as the projector beam would be constantly directed at the display while it is moving. Otherwise, if the steerable projector remains still, the display would possibly move out of the boundaries of the projection beam and become invisible at some point. The display will only become visible again if the projector is directed at its current position in 3D space.

In the initial version of the *Fluid Beam* software, there was no possibility for the user to interact with the projected displays in his surroundings. In order to enable user interaction, potential interaction techniques and respective devices had to be explored with respect to their suitability for working with projected steerable displays. A range of devices that we have tested and considered appropriate for the given task is presented in the appendix of this thesis (see Appendix A).

Part II

Related Work and State of the Art

4 Overview of Projection-based Display Technologies

In the last decades, a variety of display systems have been developed which aim at exploring and exploiting the capabilities of projection in order to create user-friendly and flexible displays. In this chapter, we present an overview of previous and current research in the area of ubiquitous and steerable projection, which ranges from large and technically complex immersive projection systems to multi-projector wall displays and steerable projection systems. The benefits and drawbacks of the different technologies and approaches are discussed and finally, the features of the presented steerable projection systems are compared to those of the DUVD system introduced in this work.

The projects described in this section constitute only a selection of all currently existing projection systems which are related to the topic of the present work. However, we assume that they build a representative cross-section of the relevant research.

4.1 Immersive Projection Environments

Spatially Immersive Displays (SIDs) enable users to explore and interact with virtual spaces while being in a physical environment which surrounds them with a panorama of imagery. Among the variety of possible display technologies for the technical realization of SIDs, projection has been one of the first approaches to be applied and has proved to be a suitable solution with regard to the large field of view it offers.

4.1.1 CAVE

The CAVE (Cave Automatic Virtual Environment, [Cruz-Neira et al., 1993]) is a room-sized, high-resolution 3D video and audio environment developed at the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago in 1991. This virtual reality theater is built of three rear-projection screens for walls and a down-projection screen for the floor, whereby the projector beams are usually folded by mirrors (see Figure 4.1 [left]). The typical interior of a CAVE is a cube with a side length of about 3m. The outside dimensions of the setup, however, cover a room of about 9m x 6m x 4m containing the projectors, the mirrors and the rest of the hardware including e.g. one computer per projection surface for controlling the corresponding image and several loudspeakers for audio output.

***Figure 4.1**: A model of a prototypical CAVE setup [left] and a user interacting in the CAVE at EVL [right] (sources: http://www.indiana.edu/∼rcapub/v21n2/p28.html and http://de.wikipedia.org/wiki/Cave_Automatic_Virtual_Environment)*

In order to see the projected graphics in stereo, the user has to wear active stereo glasses equipped with a location sensor with which the positions of the user's eyes are tracked. Thus, perspectively correct images are displayed in real-time, which results in a fully immersive experience for the user (see Figure 4.1 [right]). With some restrictions, it is also possible to experience the CAVE with several users, as long as no virtual objects appear between them, as this would destroy the stereo effect.

The main motivation for the development of the CAVE was to enable a three-dimensional immersive visualization of scientific data and thus offer researchers a new means for exploring complex information. Possible application areas encompass design engineering, training simulation, medical research, surgery and entertainment. Despite the impressive experience it offers, the deployment of the CAVE system is problematic because of the large space requirements for its setup and the enormous costs for the hardware.

Although being in a physical room, the user of the CAVE is given the impression of standing and walking in a virtual environment. In contrast to this approach, in this work, we present a method for enhancing a real environment with virtual information, which results in the coexistence of real and virtual objects.

4.1.2 blue-c

In the blue-c project at ETH Zürich, Gross et al. have developed a new generation immersive projection environment building on the concepts of the CAVE and extending them using novel technologies. In addition to a stereo 3D video output, the blue-c portal ([Gross et al., 2003]) also offers the ability to capture a 3D video representation of the user inside the environment from multiple video camera streams in real time. This is made possible by the use of special glass panels as projection screens containing liquid crystal layers, so that they can be switched from opaque to transparent, which allows video cameras placed outside the environment to "see through the walls". The system setup encompasses three of these see-through screens, two LCD projectors per screen, several cameras for video capturing and shutter glasses for the user (see Figure 4.2). Given a perfect synchronization between the shutters of the screens, the cameras, the projectors and the stereo glasses, this technology can be used not only for interacting in virtual spaces but also for telepresence applications, in which several remote users can meet in a virtual world. With the blue-c API ([Naef et al., 2004]), such collaborative immersive virtual reality applications can be developed in a straightforward way.

Figure 4.2: *Panoramic picture showing the setup of a blue-c portal (source: [Gross et al., 2003])*

Similar to the original CAVE technology, the blue-c setup is also very complex and costly. A less expensive alternative can be provided using a single fisheye projector for creating a surround screen, like in the approach described in [Johnson et al., 2007]. A significant advantage of the fisheye projector technology is that the projected image covers a very large surface of the immersive environment, and furthermore, the projector can be placed very close to the projection surface, thus avoiding shadow casting by users standing close to the display surface. However, the large field of view of the fisheye-lens projector might lead to a loss of brightness, especially towards the periphery of the projection, and to a lower resolution of the resulting image.

4.2 Large-scale Static Multi-Projector Displays

Multi-projector displays are increasingly being deployed in both research as well as everyday context. They are generated using rigidly mounted projectors, resulting in either front or rear projection setups. In contrast to the previously described immersive projection environments, these multi-projector setups are more lightweight in terms of both cost and space requirements.

In the following, we give some examples of research projects developing and exploiting multi-projector displays.

4.2.1 Office of Real Soon Now

In the “Office of Real Soon Now” at the University of North Carolina at Chapel Hill, researchers have set up a simple projector based office environment using off-the-shelf hardware components ([Bishop and Welch, 2000]). This environment builds a contrast to the

"Office of the Future", which is a project at the same university aiming at a more sophisticated setup ([Raskar et al., 1998]). Although, the Real Soon office does not provide as many capabilities as the Office of the Future, it offers the opportunity of being tested by real users over a long period of time. The developers themselves have worked in the Office of Real Soon Now for more than a year and were able to report about their practical experiences with this environment.

Figure 4.3: *Office of "Real Soon Now": panoramic images of the projector setup [top] and the resulting projected screens [bottom] (source: http://www.cs.unc.edu/∼welch/oorsn.html)*

The top image in Figure 4.3 shows the projector setup of the Office of Real Soon Now, which allows the creation of several spatially aligned static projections on a large, flat display surface in front of the user's office desk showing the desktop of the user's PC (see Figure 4.3 [bottom]). After having worked in this environment for a long period of time, the developers could report about the experienced advantages and problems. It turned out that the projected displays improve the social and technical interaction in the office, as they offer the opportunity for both researchers and visitors to view and work together on the content projected on the wall. For this purpose, the office is equipped with two wireless keyboards and two mice, which can be used simultaneously. Moving the PC desktop to the wall even made the physical office desk obsolete and thus lead to a more open office space.

Furthermore, according to the personal experience of the users, the distance to the projected display of about 2 to 3 meters is more relaxing for their eyes. Besides, the work in the modified office is overall less demanding for the whole body, as the users are given the opportunity to move freely through the office instead of sitting in front of a monitor.

The larger desktop space also offers the advantage of being able to open several applications side by side and thus to easily switch focus between them. By walking up to the projected screens, one can get a closer view at the opened windows, which represents a natural zoom capability.

Along with all these advantages provided by the novel office setup, the developers also experienced several problems with the system. The constant run of several projectors in a relatively small room leads to an extreme heating of the office and to a constant noise. Moreover, in order to project bright enough images, the lighting in the office had to be turned down. All of these hardware problems, however, can be avoided, or at least alleviated, when using more recent projector models.

Another issue, which goes beyond simple hardware problems, is privacy. When working on large projected screens, it is problematic to display confidential documents, as they will be visible to everyone in the vicinity. In order to have a more private display space, one of the users of the experimental office extended the hardware setup by a further projector creating a small screen in a bookcase which is not directly visible for visitors. Another way to maintain privacy could be to use additional small monitors for displaying private information.

Overall, the findings of the Office of Real Soon Now experiment confirm our opinion that projected displays are a suitable and maybe even better alternative to traditional monitors in an office environment.

4.2.2 Projected Display Walls

Already in the early 1990s, a research team around Paul Woodward at the University of Minnesota started developing a multi-projector display wall, which they have named *PowerWall*[1]. The PowerWall's display was a rear-projection screen of about 1.8 x 2.5 meters, illuminated by four video projectors. Each projector provided a resolution of about 2 megapixels, which resulted in an overall resolution of the PowerWall of about 8 megapixels. A first prototype was presented at Supercomputing'94[2].

Figure 4.4: *Scalable Display Wall: projector setup [left] and resulting projected screen [right] (source: http://www.cs.princeton.edu/omnimedia/)*

The main purpose of the PowerWall was the visualization of large sets of data, such as satellite images, meteorological or geological archives, or computer-simulated data, like e.g. the behavior of gases under different conditions. Taking advantage of this large screen,

[1] http://www.lcse.umn.edu/research/powerwall/powerwall.html
[2] http://sc94.ameslab.gov/

researchers could discuss and work together on the visualized data sets. Even though the PowerWall did not provide any direct user interaction capabilities on the projected image, the mere fact that it enables group interaction in front of the large screen was a significant achievement at that time.

Around 1999, a more sophisticated rear-projection wall was developed at Princeton University. This so-called *Scalable Display Wall* was built up by 24 projectors tiled together to create an image of 18 megapixels with a size of about 2.5 x 5.5 meters ([Li et al., 2000a]). In addition to the visual output, the Scalable Display Wall also offered spatialized audio created by a multi-channel sound server and 16 loudspeakers placed around the screen. An array of video cameras surrounding the wall was used for user and object tracking in order to enable interaction with the visual content (see Figure 4.4). Several input devices and modalities have been tested for interaction with the Scalable Display Wall, like e.g. voice control of the mouse cursor using a wireless microphone, pressure sensitive floor panels and a camera-tracked wand ([Li et al., 2000b], [Wallace et al., 2005]).

At about the same time, a similar tiled rear-projection display was built by the research group of Terry Winograd at Stanford University. The *Stanford Interactive Mural* comprises 12 DLP projectors, each one placed on an individually adjustable platform in front of a mirror. These mirrors fold the projector beams and direct them to the appropriate position on the screen. In this way, the size of the room needed behind the screen is reduced and the rear-projection wall becomes more compact (see Figure 4.5 [left]).

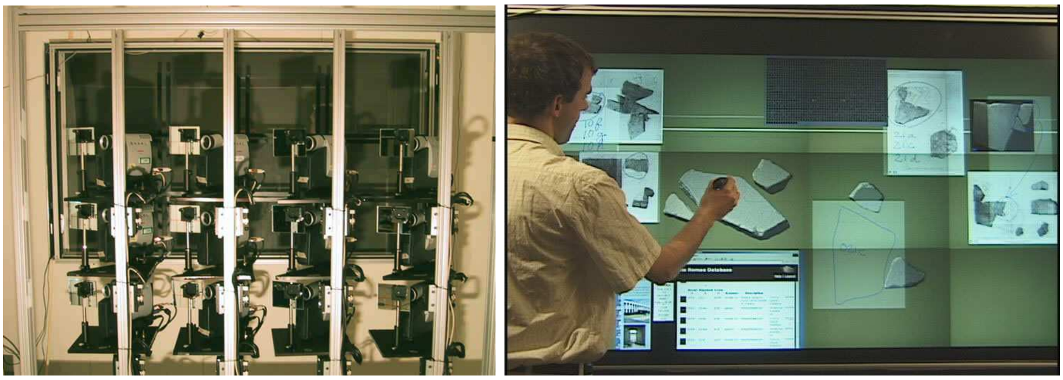


Figure 4.5: *Stanford Interactive Mural: projector setup [left] and resulting projected screen [right] (source: http://www.cs.umd.edu/∼francois/StanfordInteractiveMural.html)*

As the research focus of this project was not set at the optimization of the resulting image, some visual artifacts could be observed on the rear-projection screen resulting from the tiling (see Figure 4.5 [right]). As its name implies, the Interactive Mural served rather as a means for developing and testing new methods for user interaction with large screens ([Guimbretière et al., 2001]), which is realized using a wireless digital pen as input device (see also Section 2.4.3).

Recently, a research group at Fraunhofer-IGD in Darmstadt, Germany developed a high-quality, high-resolution tiled display wall with stereo capability. This so-called *HEyeWall* is stated to be the first stereo-capable multi-projector display worldwide ([Kresse et al., 2003]). It consists of an array of 6 x 4 tiles, each of which is illuminated by two projectors – one for each stereo image. Each one of the 48 digital projectors is driven by a dedicated PC. The display wall has a size of 5 x 2.5 meters and a resolution of 18 megapixels (see Figure 4.6 [left]). Because of the stereo capability, wearing special shutter glasses, the users can see the displayed high-resolution images in 3D. This representation is particularly suitable for applications using large 3D visual datasets, such as product and architectural design, city planning or the visualization of geographic data (see Figure 4.6 [right]).

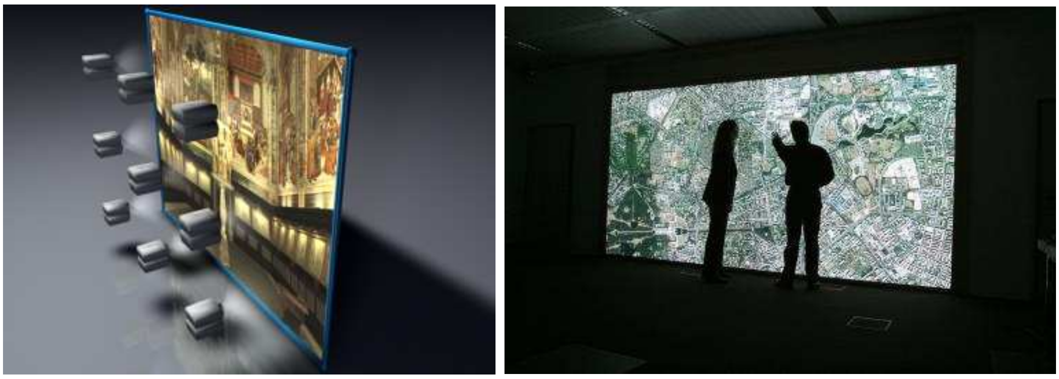


Figure 4.6*: HEyeWall: tiled rear-projection setup [left] and resulting projected screen [right] (source: http://www.christianknoepfle.de/projekt_heyewall.htm)*

In contrast to the previously described tiled display projects, the main objective of the HEyeWall project was to optimize the quality of the resulting display by reducing seams at the borders of the individual tiles and other irregularities in the projected image to the greatest possible extent. For this purpose, an elaborate photometric calibration technique was developed especially for the HEyeWall setup. In order to achieve a consistent coloring of the projected image, a common color gamut is identified for all of the used projectors, and for each projector, an individual color matrix is computed which converts its color gamut into the common gamut. Furthermore, also the light intensities and the black levels of the projectors are adjusted in order to avoid irregularities in the overall image. This complex calibration procedure enables an almost seamless integration of the individual projected tiles.

An early prototype of the HEyeWall, whose mere hardware setup costs amounted to ca. 800.000,- euros, was presented to the public at CeBIT 2004. Since then, the system has been under constant development, so that a new generation HEyeWall with improved graphics and a resolution of 35 megapixels could be presented in 2009.

4.2.3 e-Campus Project

The e-Campus project at Lancaster University focuses on developing and deploying new forms of interactive public displays ([Storz et al., 2006]). One of the resulting installations of this project consists of an array of three projectors installed in an underground bus station (called Underpass). The aim of this project was to enrich this dull, unfriendly space with visual interactive content. In order to achieve this goal, a mixture of artistic material, textual information and videos was presented on the projected displays to people waiting for the bus (see Figure 4.7).

Figure 4.7: *e-Campus installation using projected public displays in the Underpass (source: [Storz et al., 2006])*

The three displays could be accessed either individually or they could also be combined to build one large screen. Using external sensors, the projected content could be adapted to passing traffic, thus making the installation to some point interactive.

After having deployed this application for a certain period of time, the e-Campus researchers reported several important lessons which they learned during the development and deployment process. In this context, they found that it is important to have the ability to remotely monitor the output of the system as it is perceived by the user during deployment and to incorporate tools which provide information about the current state of the system. From the user's point of view, it is crucial to avoid inconsistencies in the system output, taking into account the user expectations.

4.2.4 Projected Light Displays

A research group at the Georgia Institute of Technology has developed configurable Projected Light Displays, which offer various features that help in adapting to the user's needs ([Rehg et al., 2002], [Summet et al., 2004]). First of all, it is possible to define a projected display on the fly by using coin-sized fiducial markers which are placed on the wall specifying the corners of the desired display. These markers are captured by a camera and their positions are detected. After that, the projected content is predistorted to fit within the area defined by the markers (see Figure 4.8 [top left]).

Furthermore, the authors propose a method for creating multi-planar displays on adjacent surfaces. In this case, structured light is used to detect the boundaries of the projection planes. A resulting multi-planar display is shown in Figure 4.8 [top center and right].

Finally, the Projected Light system offers the possibility to eliminate shadows and occlusion, which might be caused e.g. by the user in a front projection setup. For this purpose, a setup of two overlapping projectors is calibrated in such a way that one of the projectors can compensate for the image pixels which are occluded in the image provided by the other one.

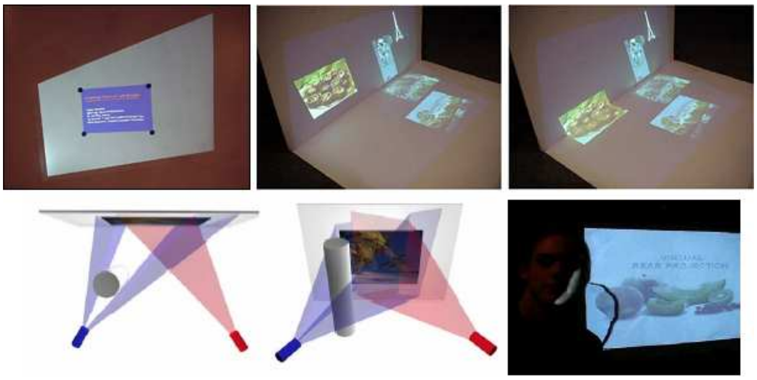


Figure 4.8: *Projected Light Displays: display definition using fiducial markers [top left]; multi-planar display [top center and right]; virtual rear projection [bottom] (source: http://www.cc.gatech.edu/∼mflagg/)*

This is realized using an adaptive alpha mask for each projector. Several methods have been implemented for detecting the current image occlusion: with a single video camera, with multiple video cameras or using an infrared source and a camera. This approach is called *virtual rear projection*, as by compensating for occlusions, the user is given the impression of working with a rear-projection screen. The authors claim that their shadow-compensating system operating at 10Hz is nearing the speed needed for interactive applications.

4.3 Augmented Objects

The project Shader Lamps aims at augmenting real world objects using projection ([Raskar et al., 2001], [Bandyopadhyay et al., 2001]). The idea is to take a neutral (e.g. white) object of a given shape and adapt its visual appearance according to a certain context. For the object illumination, the Shader Lamps system uses a pair of rigidly mounted projectors. The main challenge in this project is the spatial mapping of the projected image on non-trivial three-dimensional geometry. This is realized by initially creating a 3D model of the physical object using a 3D touch probe scanner. Subsequently, a set of key points on the physical object are manually calibrated by adjusting a projected cross on them. As a result, the appearance of the object can be changed by a projected pattern. As an example, the developers have illuminated a white 3D model of the Taj Mahal with an appropriate wall texture to achieve a realistic appearance (see Figure 4.9).

A similar work on object augmentation has been carried out by a group at Columbia University ([Grossberg et al., 2004]). Here, a projector-camera pair is used to achieve a geometric mapping of the projected image on a complex geometry, e.g. a ball. After that, radiometric analysis of the object is performed in order to compensate for a non-uniform coloring of the underlying object geometry. In this way, possible patterns on the object can be compensated for, which means that the augmented object does not necessarily have to be of uniform white color.

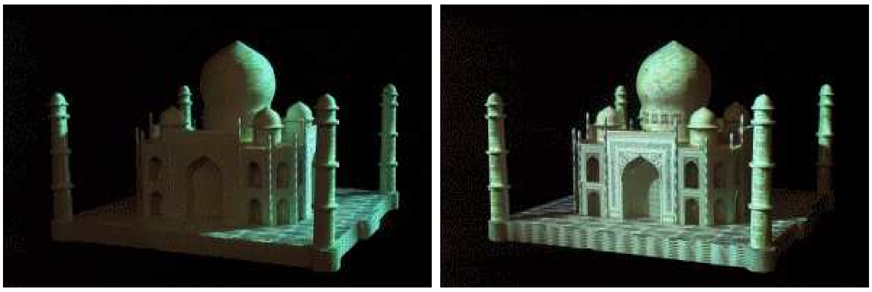


Figure 4.9: *Shader Lamps: two different illuminations of a model of the Taj Mahal (source: http://web.media.mit.edu/∼raskar/Shaderlamps/)*

4.4 Steerable Projection Systems

The research projects which are most relevant to the present work are situated in the area of steerable projection. Steerable projection systems make use of computer-controlled pan-tilt units either by carrying the projection device itself or by placing a mirror in front of the projection beam, thus directing it to a desired location in the surroundings. In the following, we describe several steerable projection systems of both types.

4.4.1 PixelFlex

The spatially reconfigurable multi-projector system PixelFlex combines 8 ceiling-mounted projectors with computer-controlled pan-tilt mirrors placed in front of each device ([Yang et al., 2001a], [Raij et al., 2003]). In this way, the images of the different projectors can compose a single projection screen, which can vary in pixel density, size and shape, thus adapting to the user's needs. In a multi-user scenario for example, a large projected display with lower resolution might be desirable (see Figure 4.10), while a single user might prefer a smaller and brighter display.

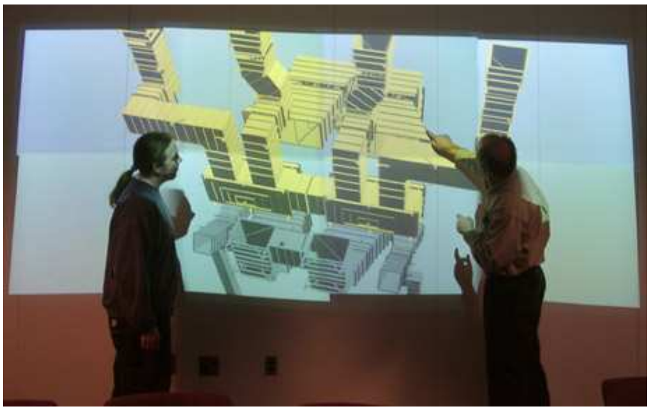


Figure 4.10: *PixelFlex: tiled display configuration (source: [Yang et al., 2001a])*

It is possible to adjust the orientations of the pan-tilt mirrors as well as the focus and zoom settings of each projector using an interface on the configuration-control PC. In this way, different screen layouts can be created and saved, so that later the system can be switched between these settings.

In order to achieve a geometrically correct image, a single wide-angle camera observes the composed projection display. Using fiducial markers and structured light the camera and each projector are spatially registered in respect to a global coordinate system. In this way, given a predefined system setting, each projector image can be predistorted appropriately to build a part of the resulting projected display.

Overlapping regions of higher pixel intensity on the resulting projected screen are compensated for by a photometric calibration. After a detection of the overlapping regions using a spectroradiometer, corresponding alpha blending masks are computed for each projector image. In this way, irregularities in the pixel intensity of the resulting image can be mostly corrected.

Overall, it can be stated that although using steerable projection, the resulting displays created by the PixelFlex system are configurable but mostly static. Location and size of these projected displays can only be switched using a number of predefined settings. To our knowledge, user interaction with the projected image has not been realized in the PixelFlex project. The created projected displays are not mainly designed to obtain an augmentation of the environment but rather to provide more flexible desktop displays.

4.4.2 Everywhere Displays Project

Probably the most prominent work in the field of steerable projection has been carried out by Claudio Pinhanez et al. in their Everywhere Displays (ED) project ([Pinhanez, 2001a], [Pinhanez, 2001b], [Pingali et al., 2003]). They have developed a device consisting of a rigidly fixed projector, a steerable mirror to direct the projector beam and a steerable video camera (see Figure 4.11 [left]). While the projector-mirror setup is similar to the ones used in the PixelFlex project, the steerable camera allows the realization of user interaction with the projected image.

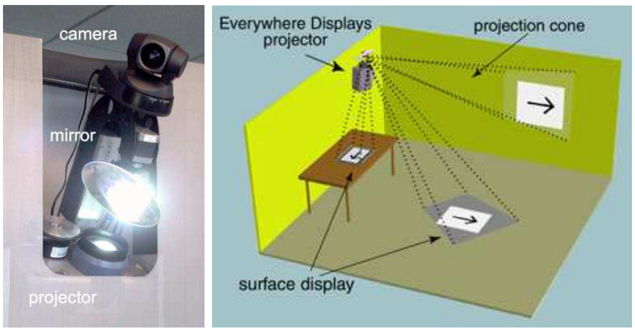


Figure 4.11: *Everywhere Displays projector: hardware setup [left] and schematic representation of the functionality [right] (source: http://www.research.ibm.com/ed/)*

The ED device enables the creation of projected displays on different surfaces in its surroundings as depicted in Figure 4.11 [right]. The projected images are predistorted to compensate for oblique projection using the virtual camera approach, which has also been used to correct image distortion in the Fluid Beam project deployed in the present work (see Section 3.1.2).

In contrast to the PixelFlex project, the ED-projector has been especially developed for the realization of Augmented Reality applications. One of the goals from the very beginning of the project was to bring visual information into the physical environment and to spatially assign it to the object or location it refers to.

In [Pinhanez et al., 2003], the authors describe a method for detecting direct user interaction with the projection. Using hand and finger detection with the ED-camera, the system can recognize clicking on a projected button or moving of a projected slider widget. Instead of the user's finger, uniformly colored objects can also be detected and used for interaction. One of the example applications includes an interactive menu for color selection. The user can select a particular color by pointing at it with his finger (see Figure 4.12 [left]).

An application example deploying the ED-device in a retail scenario is presented in [Sukaviriya et al., 2003]. The setup encompasses three different interaction spaces: a product directory projected on a wall or on a table, an interactive shelf and an interactive table.

The product directory is a projected menu containing a list of products. With a virtual slider widget on the left side of the directory, the user can scroll through the list and select a specific product. Subsequently, an arrow pointing in the direction of the selected product is projected on a physical signage board hanging from the ceiling.

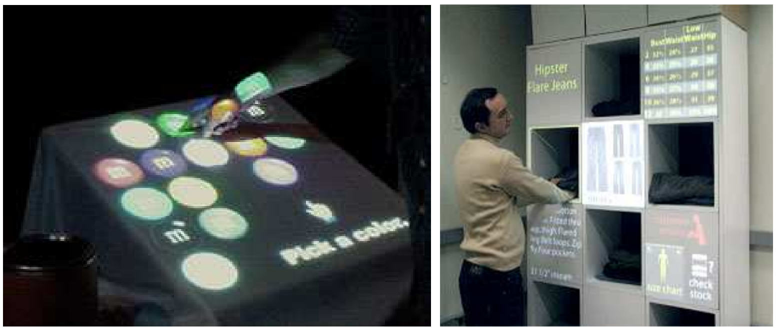


Figure 4.12: *Everywhere Displays applications: interaction with projected widgets [left] and interactive projected displays in a retail scenario [right] (source: http://www.research.ibm.com/ed/)*

The interactive shelf consists of several clothing bins combined in a rack with some plain white surfaces next to the bins on which projected information can be displayed. This information is automatically adapted by the application according to the proximity of the user and his interaction with the clothes in a specific bin (see Figure 4.12 [right]).

On the interactive table several products are placed, to which information is projected on the edge of the round table top. The product information changes while the user walks around the table, highlighting products related to different categories.

While the color selection menu and the product directory make use of explicit user interaction, the product shelf and the interactive table are examples of implicit user interaction with the projected information.

In order to be able to select appropriate projection surfaces in a particular situation, the ED system has been extended by a camera-based user tracking system ([Pingali et al., 2002]). In an initial step, surfaces suitable for projection can be defined and calibrated manually by the user. These surfaces build a set of so-called display zones, among which the system can chose when creating a projected display. An appropriate display zone is selected depending on the current position and head orientation of the user. Moreover, the position of the user is also taken into account in order to avoid occlusion of the projection surface by the user himself. When the user is detected standing between the projector and the projection surface, the system selects another display zone.

The ED system realizes real-world augmentation and offers various methods for user interaction. However, similar to the PixelFlex system, the projection surfaces are limited to a set of predefined options, between which the projection can be discretely switched. None of the presented application examples incorporates continuously moving projected displays.

4.4.3 Cooperative Augmentation of Smart Objects

At Lancaster University, David Molyneaux et al. have built and deployed a steerable projector-camera system similar to the one used for augmentation in the present work ([Molyneaux et al., 2007]). The device is mounted on the ceiling for a good overview of the environment and it is used for augmentation of smart objects.

In order to enable projected augmentation on a specific object, the system needs knowledge about the location, orientation, geometry and appearance of this object. For this purpose, the authors propose that each object keeps a model of its current state containing the desired information, which is referred to as *Object Model*. Some of this information can be static, like e.g. the geometry model of the object; other values have to be detected using internal and external sensors. The Object Model is stored on a Smart-It node attached to the object, which is also fitted with e.g. light and movement sensors ([Decker et al., 2005]). The smart object itself can require projected augmentation on its surface, sending a corresponding request and relevant information from its Object Model to the projector-camera system. In turn, the projector-camera unit can be used to locate and track a specific smart object and send the corresponding position and orientation information to its Object Model.

One example scenario described in [Molyneaux et al., 2007] uses projected augmentation to display warning messages on chemical containers. A more elaborate application example is presented in [Molyneaux and Gellersen, 2009], where a physical photograph album is augmented by projected images. Using the camera, the system can detect the current location of the album and project the appropriate cover image onto it. With the built-in Smart-Its light sensor, it can be detected when a user opens the book, so that the projected image is adapted to the "book open state". Furthermore, the album interface offers two simple projected buttons which can be triggered using vision-based touch detection. They are used to browse through albums or single photographs.

4.4.4 Interactive Surfaces in an Augmented Environment

A further steerable projector-camera pair was built and deployed by Borkowski et al. at INRIA Rhône-Alpes ([Borkowski et al., 2003]). The developed system uses computer vision techniques to automatically detect potential projection surfaces. For this purpose, projected patterns are captured with a distant video camera installed in the environment. Afterwards, a 2D map of the planar surfaces is generated and stored together with the corresponding surface characteristics needed for image pre-warping. Similar to the Everywhere Displays system, in this project, the user can only switch between different predefined projection screens. Furthermore, a white cardboard with a black border can be detected and tracked by the devices camera, so that an image can be projected on it while it is being moved. This Portable Display Screen (PDS) has been designed to enable a transfer of visual content between the different static projection screens. For this purpose, the user has to hold the PDS within a sensitive area on the projection screen, which triggers a transfer of the projected image from the screen to the PDS.

Apart from the steerable projector-camera device, the environment has been further instrumented with five steerable cameras, a fixed wide angle camera and a microphone array. This Instrumented Environment enables further user interaction methods with the projected displays, which are presented in [Borkowski et al., 2005]. The first one is a projected interface showing a list of all available projection screens. The user can select one of the screen locations using a projected button. The button click is detected using vision-based touch detection. In the second interaction mode, the user can interact with a projected interface from a distance using a laser pointer. The location of the laser spot on the screen is detected with one of the cameras installed in the environment. Similar to the previously described touch-based selection menu, the laser-based interface enables the selection of different projection screens. As soon as the user places the laser spot on the representation of one of the screens in the projected interface, the projection device is moved to the corresponding screen.

4.4.5 LumEnActive

LumEnActive is a commercially available steerable projection system, which has been developed based on scientific research ([Rapp and Weber, 2005], [Rapp and Weber, 2010]). The hardware consists of a computer-controlled rotatable mirror in front of a digital projector similar to the setups used in the projects described in Sections 4.4.1 and 4.4.2. The application concept, however, inherently differs from the ones described above. Instead of switching the projection between predefined settings as e.g. in the Everywhere Displays project, in LumEnActive the user is given the impression of a continuous workspace which is aligned with the surfaces of the room, of which only a part is made visible at a time. While the projector beam is moving over a surface, the visual content seems to stay stationary (see Figure 4.13). In this respect, the LumEnActive application is very similar to our approach presented in this thesis, as in both projects, the concept of a partially visible virtual layer covering the environment with digital information has been realized. In contrast to our work, however, the visual content in the LumEnActive application remains stationary, while only the currently visualized portion is changing. There is no way to move one visual element from one position in the room to another.

The LumEnActive software allows the placement of images, videos and VNC streams in the environment using a computer mouse or another pointing device. In a similar way, it is also possible to define movement tracks of the projector spot, which can be saved and played back in a loop afterwards. The software offers various interfaces for coupling with different

Figure 4.13: LumEnActive: schematic illustration of the system [left] and an exemplary scenario [right] (source: [Rapp and Weber, 2010])

input technologies e.g. to enable user interaction.

The system is designed to be used in a variety of applications and scenarios applying Augmented Reality, such as advertising in retail environments, trade fair installations, office installations or information systems for public spaces.

4.4.6 Projected Augmentation

In the Projected Augmentation project, Ehnes et al. work with a steerable projector device to which a video camera has been attached (see Figure 4.14 [left]). Similar to the LumEnActive approach, the system allows projected content to be spatially "attached" to objects or surfaces in the environment. This coupling between visual and physical objects is realized using visual markers attached to the corresponding physical objects (ARToolKit), which are tracked with the camera mounted on the steerable projector unit ([Ehnes et al., 2004], [Ehnes et al., 2005]).

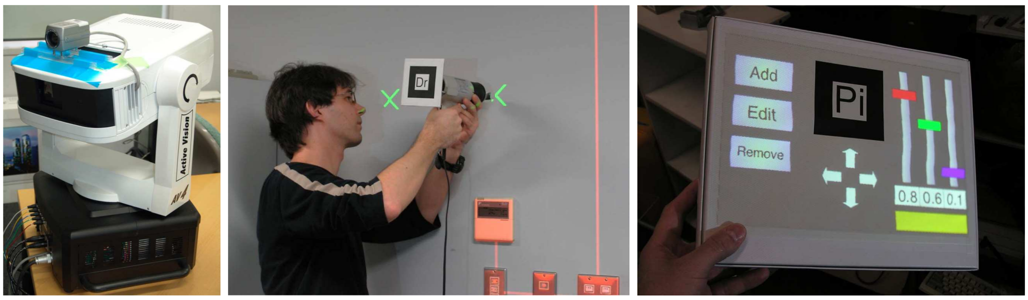


Figure 4.14: Projected Augmentation: steerable projector with attached camera [left], drilling application [center] and projected interface [right] (source: [Ehnes et al., 2004])

As an example application, the authors have implemented a system supporting the user in drilling holes. Using the X-ray vision metaphor, the system visualizes the locations of electrical wires and marks the positions where holes have to be drilled in the wall (see Figure 4.14 [center]). A further scenario which has been proposed in [Ehnes et al., 2004] is the display of an interactive menu on a movable cardboard (see Figure 4.14 [right]). This Personal Interaction Panel is supposed to be used e.g. to adjust certain application parameters. According to the paper, however, its interactive functionality has not been implemented yet.

In [Ehnes et al., 2005] and [Ehnes and Hirose, 2006], the authors present the concept of projected applications whose visual interfaces are interlinked to physical objects and displayed on them using the Projected Augmentation system. As soon as a specific visual marker has been recognized using the camera on the steerable device, the current state of the corresponding application interface is retrieved from the application repository and projected on the object.

An example application illustrating the projected application concept is the Guiding Ticket system. It is an assistant system supporting passengers in a public transportation scenario using projected information. The Guiding Ticket itself is a train ticket printed on a piece of paper, enhanced by an ARToolKit marker. When this marker is detected by the Projected Augmentation system, the latest information concerning the departure of the train is projected directly on the ticket, including the platform and coach number, the time until departure and an arrow pointing in the direction to walk in order to get to the departure platform.

Finally, the steerable projection system has been combined with a spatial audio setup in order to realize a projected virtual character, which can move along the surfaces of the environment ([Ehnes, 2010]). Although, this character is not human-like, the approach is similar to the one described in Section 6.3.3 of this work.

4.5 Synopsis

In this section, we have presented a variety of previous and present research projects related to the topic of projected displays, encompassing immersive virtual environments like the CAVE, multi-projector displays using rear or front projection, and finally, steerable projection systems. As the systems in the latter category are the most related to the Dynamic Ubiquitous Virtual Display system developed in the present work, in Table 4.1, we present an overview of the features of the different steerable projection systems in comparison with those of the DUVD system.

Although of a high technical complexity due to the combination of several steerable projectors, PixelFlex offers only a few opportunities to generate projected displays in the environment. By assembling the calibrated images of the different projectors into one combined image, it is possible to adjust the sizes and light intensity of the resulting projected display. However, the position of this display in the environment is restricted to only a relatively small region of the room in front of the projector setup. Furthermore, there is no known opportunity to interact with the projected image.

The Everywhere Displays projector was the first one to use the virtual camera approach for image distortion correction. It offers several interfaces for explicit and implicit user interaction. However, in the implemented applications, the projected displays are only switched among predefined fixed positions. A continuous movement of projected displays along the surfaces of the room is not considered in this project. Similarly, in the Interactive Surfaces project, fixed projected display locations are predefined. The user can move a projected display from one location to another using a tracked cardboard serving as a Portable Display

Screen (PDS).

In the Cooperative Augmentation project, smart objects are visually detected as soon as they appear in the scope of the projection device and visual information is projected onto them, which can be regarded as implicit interaction. The possible display locations are restricted to the surfaces of these smart objects, which means that all other surfaces in the room are not considered for projection at all. Direct user interaction is possible using projected buttons.

Similar to the DUVD system presented in this work, the LumEnActive system uses the peephole concept to visualize a portion of a larger virtual display layer. It does, however, not offer the ability to continuously move the graphical objects on this layer. Instead, the visual objects remain statically aligned with the geometry of the environment, while the beam of the steerable projector represents a moving spotlight revealing the underlying image.

In contrast to the LumEnActive system, in the Projected Augmentation project, continuously moving graphical objects have been implemented in the from of a virtual character capable of moving along the surfaces of the room. Furthermore, indirect user interaction is realized using visually tagged objects, e.g. a drilling machine for which drilling marks are projected at appropriate locations on the walls. For explicit interaction, a Personal Interaction Panel in the form of a cardboard with a projected interactive menu has been proposed, which however has not been implemented according to the corresponding paper.

	Hardware setup	Distortion correction approach	Continuous movement of graphical objects	User interaction via 3D interface	Direct user interaction via projected widgets	Direct user interaction via gestures	Implicit user interaction
PixelFlex	several fixed projectors with pan-tilt mirrors	homography	no	no	no	no	no
Everywhere Displays	fixed projector with pan-tilt mirror	virtual camera	no	no	yes	no	yes
Cooperative Augmentation of Smart Objects	pan-tilt projector-camera unit	homography	no	no	yes	no	yes
Interactive Surfaces in an Augmented Environment	pan-tilt projector-camera unit	homography	yes, with PDS	no	yes	no	no
LumEnActive	fixed projector with pan-tilt mirror	homography	no	no	only with extension	no	only with extension
Projected Augmentation	pan-tilt projector-camera unit	virtual camera	yes	no	(yes)	no	yes
Dynamic Ubiquitous Virtual Displays	pan-tilt projector-camera unit	virtual camera	yes	yes	yes	yes	yes

***Table 4.1**: Comparison of the features of the presented steerable projection systems to those of the DUVD system*

Part III

Dynamic Ubiquitous Virtual Displays

5 THEORETICAL FRAMEWORK FOR CREATING A DISPLAY CONTINUUM

Following the Ubiquitous Computing paradigm of integrating the means for human-computer interaction into the users' natural environment (see Section 2.1), we propose the creation of spatially flexible visual interfaces in 3D space, in order to enable visual output for Ubiquitous Computing systems. These visual interfaces should have the capability to easily adapt to the physical environment in which they are displayed. In order to achieve this goal, we introduce the concept of a *Display Continuum* as a novel approach to off-the-desktop visualization.

5.1 Display Continuum

In [Raskar and Low, 2001] and [Raskar, 2002], the term *interactive display continuum* is used to refer to the variety of possible projection screen shapes in a static projector setup, in which virtual objects are overlaid on physical geometry using spatial registration of the projected image to the given projection surface. In contrast, in this work, we introduce the concept of an imaginary layer enabling the display of visual information on surfaces in a physical environment and denote it as *Display Continuum*.

Definition: A *Display Continuum (DC)* is a continuous virtual layer (partially) covering the surfaces of a physical environment, on which visual content can be displayed and manipulated.

This concept allows not only the static placement and spatial storage of visual information in the physical environment, but it also offers the ability to continuously move or discretely reposition this visual content on the virtual DC layer. In this way, visual content can be distributed over the surfaces of the physical environment in a similar way as, for example, different application windows are positioned on a traditional computer desktop. The Display Continuum can thus be regarded as a kind of ubiquitous desktop integrated into the real world.

There are different possibilities for the technical implementation of the Display Continuum concept. On the one hand, mobile devices, like mobile phones and PDAs, or also head-mounted displays, allow the visualization of virtual objects as overlays on the physical world. However, with these devices, only an indirect view of the Display Continuum can be realized, as they are not part of the environment and have to be carried by the user. On the other hand, if we want to enable a direct view of the Display Continuum, the visual information has to be displayed directly onto surfaces in the physical world. One way to achieve this is to embed physical screens into each surface of the environment. This could be accomplished e.g. using thin OLED displays, which are currently being developed, and which might someday make the realization of digital wallpaper displays possible. Another approach to visualizing the Display Continuum for direct view is to use projection. The advantage of this method is that it is unobtrusive, i.e. no permanent modification of the display surfaces is needed. As already discussed in Section 4.4, steerable projector units have the ability to transform ordinary surfaces in their vicinity into visual displays without the need of any further instrumentation.

Ideally, the Display Continuum of a given environment would entirely cover all physical surfaces, and it would be directly and completely visible. However, due to the technical limitations of the potential visualization methods, it is practically not feasible to create such an ideal Display Continuum in an ordinary environment. As a consequence, we have identified four different aspects, which help us to characterize the various Display Continuum realizations concerning their limitations.

- **Spatial coverage**: Ideally, a Display Continuum is a closed layer covering the entire physical environment. However, depending on the technical realization, it might be impracticable to visualize some parts of the DC layer. For example, when using projection as the visualization means, it is very likely that certain surfaces may be shadowed by others so that projection onto them is not possible. Similarly, when the DC layer is built by embedded physical displays, there are probably also surfaces which are not overlaid by the DC. The number and size of these continuum gaps (representing a kind of blind spot) characterize the spatial coverage of a given DC.

- **Visual concurrency**: Although a DC with a maximum spatial coverage (as defined above) encompasses all surfaces in an environment, so that visual content can be placed at any location, the DC layer does not necessarily need to be entirely visible at any time. In some cases, e.g. when using mobile devices or projection for visualizing the DC, only certain segments of the layer can be visualized at the same time. Depending on the visualization device, these visible parts can be larger or smaller. Ideally, all parts of the DC layer can be visualized simultaneously, which means a hundred percent visual concurrency. Otherwise, we obtain only a partially visualizable DC layer, where the visualized windows – the so called visual peepholes (see Sections 2.3 and 5.4) – can be dynamically adapted in space.

- **Immediacy**: A DC layer can be visualized either immediately in the environment, so that users can see it directly, or it can be made visible by looking through a visualizing device, e.g. a head-mounted display, a handheld or another user-worn device. We can thus distinguish between DCs which can be perceived directly and those that can be seen only indirectly by looking through a mediatory device. The immediacy property is thus a binary attribute. Both types of DCs – the directly and the indirectly visualizable – can have their advantages and disadvantages in particular scenarios. A directly visible DC, for example, has the advantage of being more embedded into the physical room. However, in general, the realization of a directly visible DC needs a more complex setup. In contrast, indirectly visible DCs can potentially be created by means of off-the-shelf handheld devices, like mobile phones, and offer a high spatial coverage, but their very restricted visual concurrency (limited to a small screen) can be a drawback.

- **Homogeneity**: A homogeneous Display Continuum is one which is realized using only one technology, e.g. a set of large physical displays embedded in the environment. The latter setup can provide a DC with a high level of visual concurrency but a relatively low spatial coverage, depending on the size and amount of embedded displays. In order to increase the spatial coverage of this DC, one can for example combine this embedded display solution with a location-tracked mobile device, which would allow the visualization of the virtual layer at those surfaces which are not covered by embedded displays. In this way, we would obtain a heterogeneous Display Continuum combining two different visualizing approaches. As each enabling technology has its own benefits and drawbacks, by combining several types of visualizing devices, the characteristics of the resulting DC can be improved and adapted to particular application needs.

	Spatial coverage	Visual concurrency	Immediacy	Homogeneity
Embedded physical displays (e.g. OLED)	low/medium	high	direct	homogeneous
User-worn devices (e.g. HMD, handheld)	high	low	indirect	homogeneous
Steerable projection (e.g. Fluid Beam)	medium/high	medium	direct	homogeneous
Steerable projection with embedded physical displays	medium/high	medium/high	direct	heterogeneous

***Table 5.1**: Classification of possible DC-enabling technologies*

Table 5.1 gives an overview of the introduced characteristics of a Display Continuum when realized with different enabling technologies. On the one hand, a DC realized using embedded physical screens has the advantage of a high level of visual concurrency, as every part of the DC can be made visible at the same time. This approach also offers the benefit of a direct view without the need of instrumenting the user. On the other hand, at the current stage of technology, it is very hard to achieve a DC with a high spatial coverage using only physical screens. In contrast, with user-worn devices, DCs with maximum spatial coverage can be achieved. However, their disadvantage is that the devices have to be carried by the user, and in most cases, they visualize only a very small portion of the DC at a time, i.e. their level of visual concurrency might be quite low.

In this work, we concentrate on projection-based DCs with a high spatial coverage and a medium visual concurrency, which are realized with steerable projectors. As they are projected directly onto the surfaces in the physical world, these DCs are immediately visible for the user. Primarily DCs visualized by steerable projection are homogeneous. However, later in this work, we show how stationary physical displays can be embedded in a projection-based Display Continuum, in order to achieve a heterogeneous DC with an increased level of visual concurrency (see Section 6.2.2.2).

5.2 Virtual Displays

Given such a large display surface as provided by a Display Continuum, there are two conceptually different modes to interact with it. On the one hand, the user can be given the opportunity to address every single pixel of the virtual layer and to interact with it by changing its color and intensity. On the other hand, interaction with the Display Continuum can refer to closed units similar to the traditional desktop interaction with windows and icons. The former type of interaction is similar to the one in a drawing application on a desktop where, depending on the current settings, the user can modify individual pixels or pixel groups by moving the cursor over them. This interaction mode offers a very high degree of freedom in enabling access to the primary units of a DC (namely its pixels), which the user can manipulate individually. However, the flexibility of this interaction mode also implies a high level of complexity. Usually, when working on a traditional desktop, users do not want to interact with single pixels but with virtual objects, like windows, buttons, icons, etc. If, for instance, a new icon has to be created, the user would usually not draw it by hand but let the system create it using certain templates. Besides, if the user is only able to address individual pixels, he would not have the ability to easily address bigger units, e.g. icons, whole images and frames. Therefore, we propose the second interaction mode for working with the Display Continuum, in which whole visual units can be referred to and interacted with. We denote these units as *Virtual Displays*.

Definition: A *Virtual Display (VD)* is a spatially defined unit of visual content, which can be referenced and manipulated as a part of a Display Continuum.

In principle, projected Virtual Displays can have any given geometric shape. However, for the sake of simplicity concerning their creation, in the current realization, Virtual Displays only have rectangular shapes. Despite this simplification, projected Virtual Displays can represent arbitrary visual shapes, as due to the special characteristics of projection, the projected displays can be borderless and without visible background, so that only the visual shapes displayed on them are perceived as a closed visual entity and the actual rectangular shape of the display frame remains invisible.

5.3 3D Model of the Physical Environment

In order to realize a Display Continuum in a given environment, information about the geometry of this environment with its surfaces must be provided. This information can either be gathered automatically using e.g. optical geometry reconstruction with structured light ([Raskar et al., 1999], [Salvi et al., 2004]), laser scans ([Surmann et al., 2003]) or a combination of laser and image data ([Sequeira et al., 1999], [Tokuda et al., 2003]), or the geometry of the environment can be modeled using a 3D graphics editor, like Blender[1], Google SketchUp[2], AutoCAD[3], etc. Automatic geometry reconstruction requires special and mostly expensive hardware, and depending on the geometry measuring approach, the resulting model might be of low accuracy. Moreover, when automatically scanning the geometry of an environment, it is not possible to differentiate between individual parts of the environmental geometry, e.g. different walls, tables, windows, doors, etc. In order to receive such a semantically enriched 3D model, the measured geometry data has to be structured and edited, or the model can be created entirely in an appropriate editor.

In the next chapter of this work, we present two different approaches to modeling the geometry of an environment in order to obtain a Display Continuum. One way is offered by the map modeling toolkit Yamamoto, which has been extended in the course of this work, so that it provides methods for modeling steerable projectors and Virtual Displays (see Section 6.2.1.2). As an alternative approach, we have developed a user-assisted model acquisition toolkit using visual markers, which is presented in Section 6.1.

As, in general, a Display Continuum does not offer full spatial coverage, this has to be reflected in the corresponding 3D model. Interruptions in the Display Continuum can be caused by objects and surfaces on which placement of Virtual Displays is not possible – either due to their inappropriate coloring and structure, or because they are in some way occluded by other objects in the environment.

Definition: An *obstacle* is an interruption in the Display Continuum in the form of a surface, which is unsuitable for the placement of Virtual Displays.

When working with projection, *obstacles* can be e.g. pieces of furniture, windows, doors, etc. In general, such surfaces do not offer an appropriate projection space because of their intensive color or pattern, uneven structure, reflectance, specularity or transparency.

Further interruptions can also occur in a Display Continuum, when its underlying surfaces, which are potentially suitable for the placement of Virtual Displays, are occluded by other objects in the environment. Especially concerning projection-based DCs, such occlusions may occur not only when an object is located in the visual line between the user and the projection surface, but also when the line of sight between the projector and the aimed projection surface is interrupted by another object. In the latter case, the corresponding part of the DC might potentially be visible for the user, but nevertheless, it does not allow the placement of Virtual Displays, as it is shadowed by some other object.

[1] http://www.blender.org/
[2] http://sketchup.google.com/
[3] http://usa.autodesk.com/adsk/servlet/pc/index?id=13799652&siteID=123112

We denote such interruptions of a DC as *shadows*.

Definition: A *shadow* is an interruption in the Display Continuum caused by an occlusion of a surface by another object.

Although a Display Continuum is regarded as one continuous layer, it is usually built of several surfaces with different orientations. In this work, such interruptions concerning the surface orientation are denoted as *discontinuities*.

Definition: A *discontinuity* is an interruption in the Display Continuum, at which the normal vector of the underlying surface is changing.

Discontinuities mostly occur at the borders of adjacent surfaces, e.g. at room corners, and they require special treatment, especially when creating or moving Virtual Displays. Depending on the realization of the respective DC, the change of the surface normal in the Continuum might require an adaptation of the shape and orientation of a Virtual Display that is placed at this position.

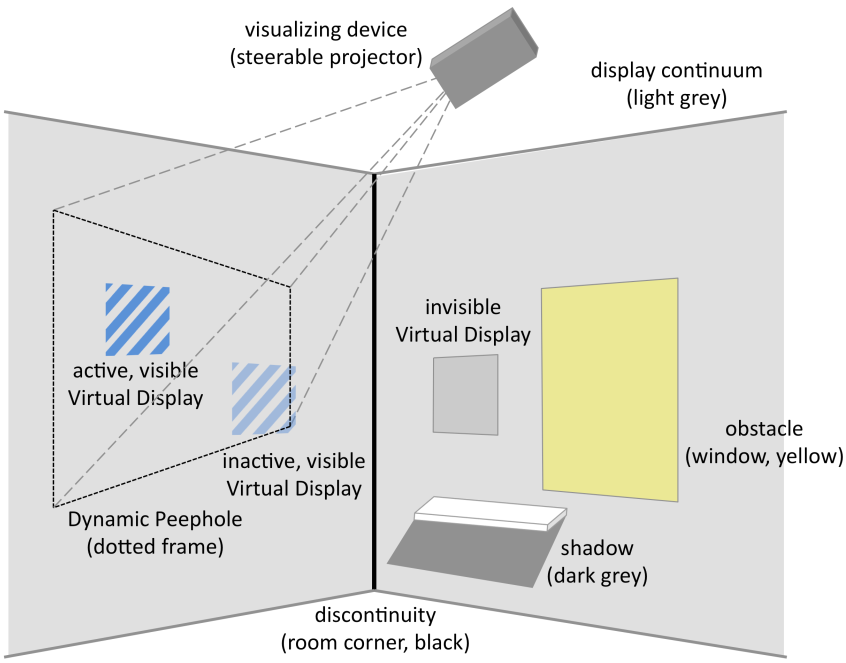


***Figure 5.1**: Exemplary 3D model with visualization of the basic DUVD concepts*

Figure 5.1 illustrates the basic concepts of Virtual Displays and Display Continuum in an exemplary 3D model. The Display Continuum overlays the walls of a room (light grey), so that Virtual Displays (active/inactive and visible/invisible) can be placed on it. The visible active display is visualized by a projector, which creates a DC with restricted visual concurrency. The currently visible part of the DC is denoted as a Dynamic Peephole (see Section 5.4). Exemplary interruptions of the DC in this illustration are a window (representing an obstacle), a room corner (representing a discontinuity) and a shadow cast by a shelf mounted on the wall.

In [Ashdown et al., 2004], an approach is presented which allows the calibration of a camera-projector system in such a way that it enables the placement and interaction with projected images across two adjacent surfaces placed at an angle of 90°. The corner between these two surfaces represents an interruption of a DC according to our definition. A typical example of such a setup is a horizontal desk pushed against a vertical wall. If an image is moved from one surface to the other, it is bent around the corner when it reaches the edge of the surface. This creates the realistic impression that a highly flexible piece of paper is smoothly slipped along the surfaces (see Figure 5.2 [left]).

***Figure 5.2**: Illustration of different discontinuity handling approaches: image bent around a corner [left], image bent along a curve [center] and Virtual Display switching orientation at a corner [right]*

A similar application is also presented in [Weiss et al., 2010], which is called BendDesk. In contrast to the former approach, the horizontal and the vertical surfaces of the BendDesk are connected by a curve, i.e. there is no discrete switch of the surface normal but a continuous transition, which allows a smoother interaction. When moving an image from one surface to another on the BendDesk, it is also bent along the curve, which creates the illusion of a virtual layer covering the curved surface of the BendDesk, on which the image is dragged (see Figure 5.2 [center]).

In contrast to the approach presented in this work, both previously described applications represent very locally restricted setups using fixed projectors, which require specific complex calibration in order to deliver spatially correct results. If the DC is visualized by a device with lower spatial accuracy (like the one used in this work), the bending of the moved Virtual Display might not be exactly aligned with the underlying discontinuity. Such slight inaccuracies, which are hardly noticeable on a planar surface, lead to unsatisfying results when they appear at a discontinuity, because the discontinuity itself represents a spatial reference point, which reveals very clearly even small irregularities in the positioning of the visual content. Thus, it would appear unnatural to an observer if the bend of a moved Virtual Display is shown not exactly in the according room corner but slightly beside it.

For the movement of Virtual Displays across discontinuities on DCs visualized by devices with mechanical inaccuracies, we propose to simply switch the orientation of the entire VD according to the normal of the underlying surface as soon as the midpoint of the VD crosses the discontinuity (see Figure 5.2 [right]). Additionally, a more realistic movement effect can be achieved by overlaying the discontinuity with a static virtual object. In this way, the

Virtual Display can disappear behind this masking virtual object shortly before reaching the discontinuity and then reappear on the other side with an adapted orientation. A masking virtual object for a room corner can be, for example, a virtual pillar.

5.4 Dynamic Peephole and Ubiquitous Cursor

If a Display Continuum offers only a low level of visual concurrency, most of its visual content is invisible, while only small parts of it can be visualized at the same time. In this case, the concept of Dynamic Peepholes can be applied, in order to render particular parts of the DC visible at a certain point in time. This concept is derived from the generalized peephole metaphor presented in Section 2.3.

Definition: A *Dynamic Peephole* is a spatially adjustable virtual window revealing the visual content of a Display Continuum with restricted visual concurrency, which can be controlled either by a user or by a system.

Metaphorically speaking, a restricted Display Continuum can be regarded as an unlighted layer, which is partially made visible by directing a virtual torch on it. The light beam of this virtual torch produces an island of visibility on the layer, which we call a Dynamic Peephole, so that the underlying content is visualized.

In fact, according to the Dual Reality paradigm (see Section 2.1), a Dynamic Peephole represents a mediating window in a Dual Reality world, which allows the transition of visual data from the virtual world to the real world. In our case, the real world is the physical environment and the virtual world is represented by the 3D model of this environment. Virtual Displays, which are created and manipulated in the model, appear in the physical world only if they lie within a peephole. Conversely, user manipulations on the visualized Virtual Displays in the real world are reflected on the corresponding virtual objects in the 3D model.

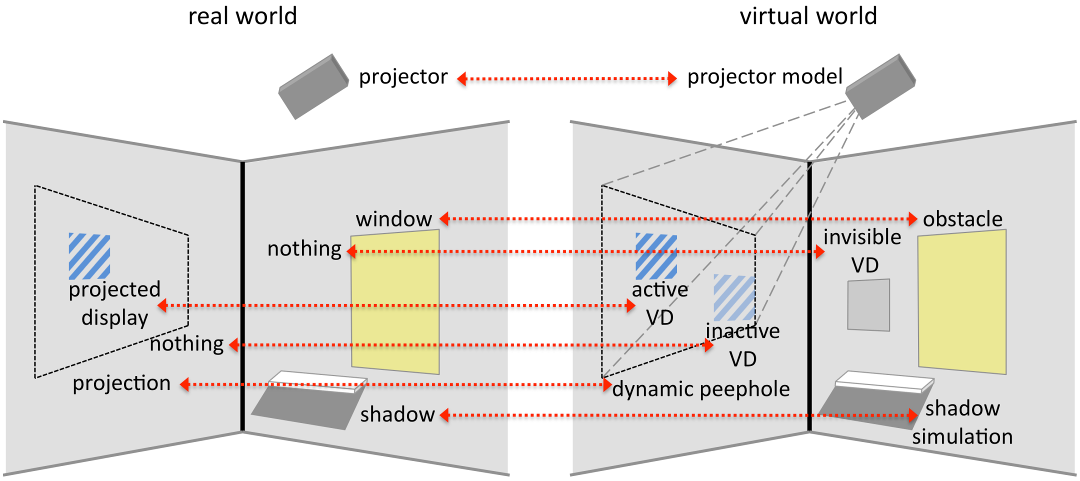


***Figure 5.3**: Exploitation of the Dual Reality concept*

To clarify the relation between real world and 3D model, Figure 5.3 illustrates the correspondences between objects and phenomena in the real world and their counterparts in a virtual model.

Virtual Displays are called *visible* if they lie within a peephole, i.e. they are currently rendered visible in the physical environment; otherwise, if they lie on the invisible part of the Display Continuum, they are called *invisible*. Additionally, a Virtual Display can be in an *active* or *inactive* state, which describes its potential visibility property. If a VD is inactive, it does not appear visible to the user even if it lies within a peephole. On the other hand, an active VD only appears visible if it is located within a visualizing peephole or if a Dynamic Peephole is moved over it.

As the definition states, the position of a Dynamic Peephole – and hence the currently visible part of the DC – can either be determined by the user in order to access particular Virtual Displays or create new ones at desired locations, or the peephole position can also be steered by a system in order to provide visual feedback, reveal particular information to the user or draw the user's attention to a particular location.

When the user intends to interact with objects on the Display Continuum, it is important to make clear where the current interaction focus of the system lies. In traditional desktop systems, the mouse cursor marks the location on the screen which is currently in focus and will respond to potential user input. The cursor concept provides the user visual feedback concerning his desktop interaction and defines the location of the current interaction focus.

A similar approach to providing feedback on the interaction focus can also be applied to a Display Continuum. As an equivalent to the mouse cursor on a desktop, we propose the concept of a *Ubiquitous Cursor*, which is a visual mark displayed in a Ubiquitous Computing environment indicating the position currently aimed at for interaction.

Definition: A *Ubiquitous Cursor* is a visual mark indicating the location of the current interaction focus on a Display Continuum.

This Ubiquitous Cursor is placed on the Display Continuum and can be moved along it. In order to keep it visible, the Ubiquitous Cursor should always be located on the visible part of the DC (e.g. within a visual peephole). In the projection-based implementation presented in this work, the Ubiquitous Cursor is bound to the center of the projected Dynamic Peephole and thus can be directed at different positions on the underlying Display Continuum.

In a heterogeneous Display Continuum setup, where the DC is visualized using a combination of different device types, it must be possible to transmit the Ubiquitous Cursor between different visualizing technologies. An example realization of such a focus switch between steerable projection and stationary physical screens is described in Section 6.2.2.2.

In the original peephole implementation ([Yee, 2003]), one main problem, which has been observed in user studies, is the loss of orientation on the virtual workspace (which represents a small Display Continuum). The adoption of the Ubiquitous Cursor in combination with the alignment of the Display Continuum with the physical environment could counteract this drawback.

5.5 Theoretical Model of Dynamic Ubiquitous Virtual Displays

Similar to traditional screens, ubiquitous displays have a number of basic parameters, which have to be defined for each ubiquitous display instance. These parameters encompass the *display name* (which is needed for referencing), its *size* (width and height), its *location* in physical space (defined by its midpoint as the reference point) and its *orientation* (which can be defined by two vectors, e.g., the normal and the down vectors). Instead of defining the display size, location and orientation separately, it is also possible to characterize a display by the positions of its corners, from which the previously mentioned parameters can be inferred. In some cases, the definition of a ubiquitous display by its corners might be more comfortable for the user. However, when changes in the display location and orientation have to be defined, then the use of the midpoint and orientation vectors is more appropriate.

Apart from these shape and location parameters, a visual display can also be characterized by its resolution. Traditional monitors normally have variable resolution settings, whereby a technically determined maximum resolution cannot be exceeded. Projection devices also have a technically fixed resolution, defining the number of pixels that they are capable of projecting. The actual resolution of a projected image, however, depends not only on the projector resolution but also on the distance between the projector and the screen surface. In this context, a more relevant parameter is the *pixel resolution* “pixels per inch”, which defines the density of pixels on a display. A projection surface placed relatively close to the projector results in a small projected image with a high pixel density, which appears sharper then an image created with the same projector on a screen which lies farther away (independent from the projection focus, which has to be adapted accordingly). Apart from the projection distance, the pixel density – and hence the resolution of the projected image – also depends on the zoom setting of the projector. A higher zoom factor leads to a smaller projected image with a higher pixel density.

In photometry, *illuminance* is the total amount of light incident on a surface per unit area. It is a measure of the intensity of the incident light or, informally speaking, the perceived brightness of an illuminated object. Illuminance is measured in *lux* (lx) or *lumens per square meter* (lm/m^2), and its value is wavelength-weighted by the *luminosity function* to correlate with human brightness perception. The resulting value can be used to describe the perceived brightness of a Virtual Display.

The *transparency* attribute, which can vary between 0% and 100%, defines the translucency of a Virtual Display, where 0% results in a fully opaque display, and 100% transparency leads to a fully transparent (and hence imperceivable) display. As already discussed, the perceptibility of a projected Virtual Display in our case also depends on its activity state and on the current focus of the Dynamic Peephole (see Section 5.4).

Another property which is essential for the definition of ubiquitous displays is the *display content*. The approach presented in this work allows the presentation of both static images and video data as ubiquitous display content. Although the visual content is not a necessary specification for the creation of a display, we also regard it as a basic parameter, because the main purpose of a ubiquitous display is the presentation of visual output.

Summarizing, we define the following *basic parameters* of Dynamic Ubiquitous Virtual Displays:

- Size (width and height)
- Location (3D position of the display midpoint)
- Orientation (normal and down vectors)
- Resolution
- Brightness / Illuminance
- Transparency
- Activity / Visibility
- Visual content (image, video or live stream)

In addition to these basic parameters, which are similar to those of traditional physical screens, ubiquitous displays also possess properties which arise from the special characteristics of steerable projection (see Section 2.5). Due to the fact that the application of the Display Continuum approach allows the modification of the location and orientation of ubiquitous displays, they can be assigned a *movement* property. The basic properties size, transparency and visual content can also be modified accordingly.

All these dynamic modifications of the basic DUVD parameters are then dependent on *time constraints*, which means that a specific change of a basic parameter takes place in a predefined period of time. If this time period is zero, the parameter modification is discrete, otherwise, it is continuous. A ubiquitous display can, for example, be made to move continuously from position A to position B within 10 seconds or, in contrast, it can be made to disappear from position A and immediately reappear at position B, which would represent a discrete location change. Similarly, orientation, size, transparency, and visual content of ubiquitous displays can also be modified continuously or discretely.

As a result of the above consideration, we define the following types of *dynamic parameter modifications* of DUVDs:

- *continuous modification* of a basic parameter (with a given duration)
- *discrete modification* of a basic parameter

Of course, dynamic modifications of different parameters can also be performed in parallel, or they can overlap each other in time. A ubiquitous display can e.g. move along a predefined path, while its content is changed appropriately.

Although, dynamic parameter modifications could be defined to appear without any dependencies, typically, there would be some constraints, which would trigger a dynamic parameter modification. At the same time, a basic parameter itself can be a trigger for the modification of another basic parameter. The location of a ubiquitous display e.g. can influence its visual content, or the displayed visual content can have an influence on the current display size. This means that each basic parameter of a ubiquitous display can be used in a constraint for modifying the values of other basic parameters. We refer to these parameters as *intrinsic constraint parameters*.

Further parameters, like the identity and location of a user or a physical object in the environment, or the reaching of a specific point in time, can also act as triggers for an adaptation of certain ubiquitous display parameters. These parameters are usually provided by external events, and we refer to them as *extrinsic constraint parameters*.

In the following, we give some examples of extrinsic constraint parameters, which can influence and trigger changes in the basic ubiquitous display parameters:

- Point in time
- User identity and location
- Object identity and location
- Direct user interaction

A typical example of a time-dependent ubiquitous display is a reminder message. In this case, a ubiquitous display can be created at a desired location, initially showing no content. In order to make it display a reminder message, it has to be assigned a "point in time" constraint, which would have an influence on the visual content of the display. In this way, the ubiquitous display can show a message about an approaching appointment at a specified date.

If the reminder message concerns only individual users or user groups, the ubiquitous display can be assigned a "user identity and location" constraint defining that the specific message should be displayed only if a dedicated person is present in the environment. In this case, the user identity and his current location have an influence on the visual content of the display.

In order to make the reminder message even more adaptive, the user location can also be used in a constraint influencing the location of the ubiquitous display showing the message. In this way, the message can be displayed at a location in the vicinity of a dedicated user, depending on the current user location at the previously specified point in time. Another scenario, in which the user location can be exploited in constraints influencing ubiquitous display parameters, is a museum guide application. In this context, the ubiquitous display location can be bound to the current user location, so that the display is "following" the user. Moreover, the displayed visual content can be adapted to the current display location in order to show appropriate information corresponding to the exhibits in its vicinity, and the displayed content can further be dependent on the user identity in order match the user's interests.

Later in this work, some approaches for explicit and implicit user interaction with ubiquitous displays will be presented (see Section 6.2). This interaction should of course have an impact on the basic parameters of the aimed ubiquitous display. Thus, events issued from the user interaction can also deliver extrinsic constraint parameters, which can influence certain basic ubiquitous display parameters.

Of course, the list of proposed extrinsic constraint parameters is just an excerpt, as there are numerous other factors which can be taken into account when defining ubiquitous display behaviors. Consequently, the characterization of ubiquitous displays must be flexible enough to allow the definition of further extrinsic constraint parameters.

Figure 5.4 presents an overview of the concepts presented in this chapter and their relations to each other.

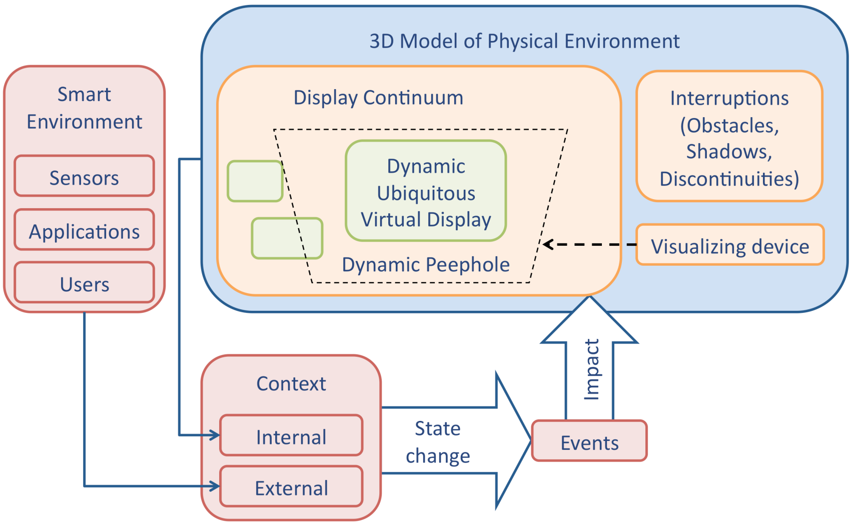


***Figure 5.4**: Illustration of the concept of Dynamic Ubiquitous Virtual Displays*

5.6 Interaction Methods for Dynamic Ubiquitous Virtual Displays

When a novel medium for visual information presentation is developed, one has to consider how users can interact with it in order to access and manipulate its content. Usually, before inventing entirely new means of interaction, a common approach is to identify well-established interaction metaphors and methods, and adapt them in such a way that they can be used with the novel medium. In this way, users can grow more easily accustomed to the new interaction method, when they recognize slightly modified but familiar interaction structures.

Based on the previously developed concepts and parameters, we have identified the following functionalities which a ubiquitous display system should support:

- Virtual Display creation

- Virtual Display deletion
- Defining/redefining the content of Virtual Displays
- Adjusting the basic parameters of Virtual Displays (size, rotation, etc.)
- Moving Virtual Displays along the Display Continuum
- Moving the Dynamic Peephole along the Display Continuum

5.6.1 Interaction via 3D Model

As the currently most popular metaphor in human-computer interaction is still the WIMP metaphor on desktop computers, our first idea for an interaction approach with the Display Continuum was to visualize the 3D model of the environment and use it to remotely manipulate the Display Continuum via desktop interface. This interface consists of a window showing a view of the environmental model, which can be manipulated using a common computer mouse. The visualization of the model shows the surfaces of the Display Continuum and its interruptions (obstacles, shadows), where each type of surface is characterized by a different color. By clicking and dragging, the user can rotate the 3D model in order to access a desired location on the Display Continuum. Mouse interaction can also be used for the creation and manipulation of Virtual Displays.

Two implementations of this 3D interface concept are presented in detail in Section 6.2.1. An advantage of this interaction approach is the decoupling of the interaction space (desktop) from the application space (physical environment), in which the actual effect of the interaction takes place. In this way, the user can interact with the Display Continuum remotely, without needing to be present in the corresponding environment. On the other hand, in case the user is in the environment he is working with, the decoupling of interaction and application spaces can be regarded as a drawback, because in this case, the user has to constantly switch his attention between the interface on the desktop and the real world in order to observe the effects of his manipulation.

5.6.2 Interaction in Physical Environment

In order to overcome the attention switch problem, which arises when interacting with a Display Continuum using a 3D desktop interface, we decided to offer users the ability to interact with the DC directly in the physical environment in which it is visualized. This type of interaction is more direct, as the user can be given the impression of immediately manipulating its physical surroundings without any apparent mediatory interface.

Moreover, the transition of the interaction space into the user's physical surroundings enables the implementation of system-controlled DUVDs, with which the user's focus of attention can be guided to relevant information by the ubiquitous system. One problem which occurs in this case is the loss of focus when the user is not aware of system-issued output on a DUVD, e.g. when a Virtual Display appears outside the user's current field of view. This issue can be handled by adding further output modalities to the DUVD system, such as sound. Auditory cues can be used as spatial hints to newly appearing system messages, which can be realized using a spatial audio system (e.g. SAFIR [Schmitz and Butz, 2006]). Another approach to counteracting the focus loss problem is the tracking or estimation of the user's viewing direction. In this way, the visual information on the Display Continuum can be adapted to the user's current field of view by either displaying the respective ubiquitous display in front of the user or by visually guiding the user's view to the displayed content if it is bound to a certain location.

In the next chapter, we present several implementations of real-world user interfaces for DC interaction. These interface modules offer various gesture interaction approaches, in which the gestures are recognized using different sensing techniques, including vision-based and other sensor-based approaches.

5.7 Synopsis

In this chapter, we have presented the basic concepts concerning the realization of a Display Continuum. In this context, we have defined the terms *Display Continuum*, *Virtual Display*, *Dynamic Peephole* and *Ubiquitous Cursor*. We have established a set of characteristics for classifying a Display Continuum, and we have proposed a concept for modeling the physical environment in order to obtain a 3D model which represents a Display Continuum. Further, we have explained how a Dynamic Peephole and a corresponding Ubiquitous Cursor can be applied for visualizing relevant information on a Display Continuum and for supporting the user during interaction.

Finally, we have proposed a theoretical model of Dynamic Ubiquitous Virtual Displays taking into account a number of basic parameters and their potential modifications based on intrinsic and extrinsic parameter constraints. Further, we have outlined the two basic types of interaction with a Display Continuum, which can be realized either using a 3D interface in a desktop application or real-world interaction in the physical environment.

6 Architecture, Realization and Applications

In this chapter, we present the architecture and realization of the *Dynamic Ubiquitous Virtual Display (DUVD)* system based on the previously described concepts. Figure 6.1 shows an overview of the DUVD system architecture with its main modules.

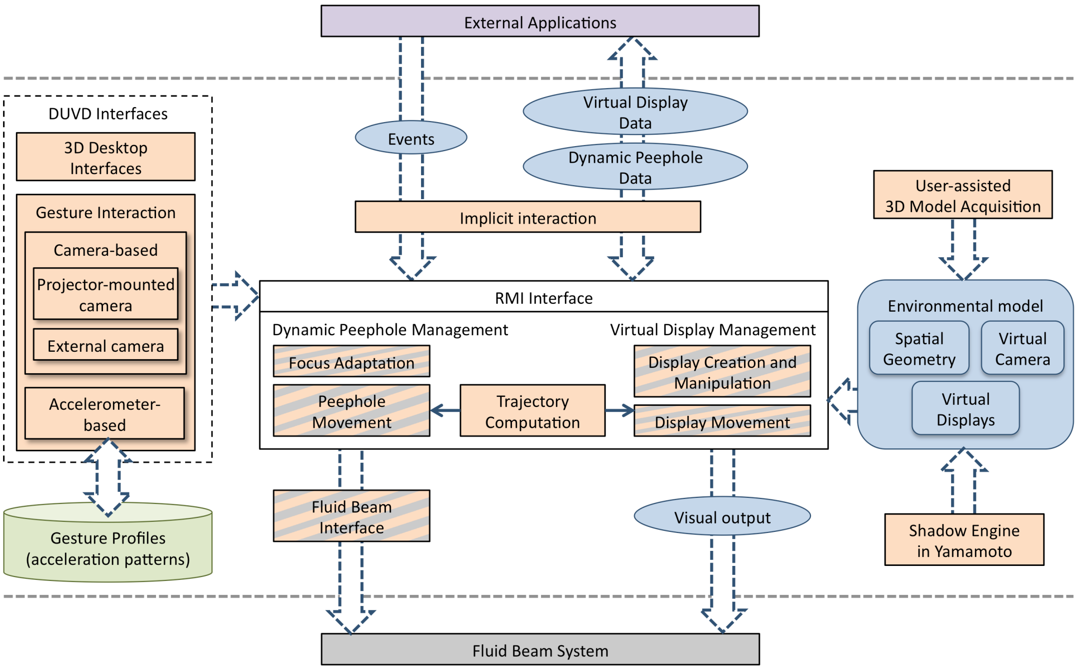


***Figure 6.1**: DUVD system architecture*

The core components needed for a distortion-free projection of visual content on appropriate surfaces in the environment and for the control of the steerable projector device are provided by the Fluid Beam software (see Section 3.1.2), which has been adapted and extended for use in the DUVD system (depicted as orange striped rectangles). Exploiting these core modules, the different DUVD interfaces enable users to work with the Display Continuum in various ways. These interfaces and further main DUVD modules (depicted as orange rectangles) are described in detail later in this chapter. In addition to the explicit interaction opportunities, implicit interaction has also been realized using sensor data provided by external applications in form of events. The 3D model representing the Display Continuum can either be generated with a semi-automated modeling tool (see Section 6.1), or it can be retrieved from the modeling toolkit Yamamoto, which has additionally been extended by a shadow-computing module (Shadow Engine) (see Section 6.2.1.2). In the overview graphic, the environmental model and data which is generated by the DUVD system or required as input from external applications is represented by blue oval elements. Finally, user data needed by the accelerometer-based interaction module is obtained from a gesture profile pool (depicted as green cylinder).

6.1 User-assisted 3D Model Acquisition

As discussed in the previous chapter, for creating a Display Continuum, the DUVD system needs an appropriate 3D model of the environmental geometry. In order to facilitate the user in generating such a model, we have developed a tool for user-assisted 3D model acquisition using the steerable projector device with an attached camera.

Projector-camera calibration

In order to enable the correct detection of surfaces in the environment, the steerable projector-camera unit has to be calibrated appropriately. The initial calibration process encompasses the calibration of the projector-mounted camera and subsequently a computation of the spatial relation between projector and camera.

For calibrating the intrinsic camera parameters, we apply Zhang's algorithm ([Zhang, 1999], [Zhang, 2000]), which uses a planar checkerboard pattern with a known size as calibration tool. The camera parameters are calculated by semi-automatically matching the reference points on the checkerboard in images taken from different perspectives.

After the camera has been calibrated, the same algorithm is exploited for the computation of the spatial relation between projector and camera, taking into account the previously determined intrinsic camera parameters. For this purpose, we need a reference surface with an attached checkerboard pattern defining the world coordinate system. In a first step, the position of the camera in the world coordinate system is computed, consisting of a translation (T_{wc}) and a rotation (R_{wc}) component (see Figure 6.2 (a)). Subsequently, we project the checkerboard pattern on the reference surface, capture an image of it with the projector-mounted camera (see Figure 6.2 (b)) and detect the reference points of the projected pattern in the camera image. Taking into account the previously computed position of the camera in the world coordinate system, we can compute the positions of these reference points in the same coordinate system (see Figure 6.2 (c)). Hence, applying again Zhang's algorithm, we can obtain the position of the projector in world coordinates (T_{wp} and R_{wp}) (see Figure 6.2 (d)).

Finally, the relation between projector and camera can be computed as follows:

$$R_{pc} = R_{wc} * {R_{wp}}^{-1}$$
$$T_{pc} = T_{wc} - R_{pc} * T_{wp}$$

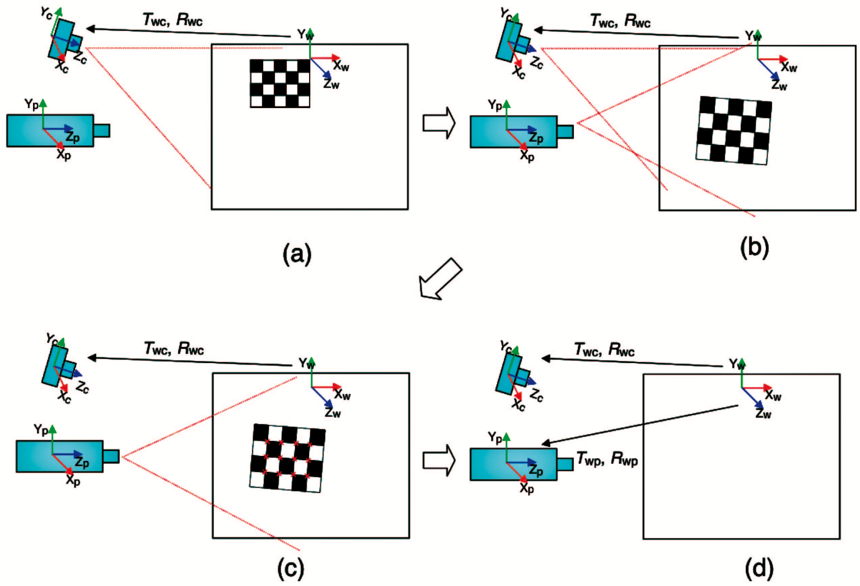


Figure 6.2: *Calibration of projector-camera setup: (a) computing camera position in world coordinate system defined by physical checkerboard pattern; (b) capturing an image of a projected checkerboard pattern; (c) detecting reference points of projected pattern and computing their positions in world coordinate system; (d) deriving translation and rotation components for computing projector position in world coordinate system*

Projector-Camera Calibration Toolbox

For the implementation of this calibration algorithm, we have developed a user interface which builds on the *GML C++ Camera Calibration Toolbox*[1] provided by the Graphics and Media Lab of the Moscow State University. This tool is a stand-alone application providing methods for camera calibration (including the previously introduced algorithm by Zhang). In addition to a set of pattern detection and calibration algorithms, the Camera Calibration Toolkit provides a graphical user interface allowing the user to easily access the different functions.

We have extended this GML C++ Camera Calibration Toolbox to a *Projector-camera Calibration Toolbox* implementing the previously described calibration algorithms. Figure 6.3 shows a screenshot of the modified user interface of the toolbox. The buttons (a), (b) and (c) have been adopted from the original toolkit version: button (a) triggers the pattern recognition algorithm for individual images, button (b) starts the same algorithm for a list of images, and (c) triggers the camera-calibration algorithm after the checkerboard pattern has been detected in at least four images.

With buttons (d) – (f), the additional functions for camera-projector calibration can be triggered. The transformation of (projected) reference points from camera coordinates to points in world coordinates given a predefined reference surface is triggered by button (d). Subsequently, the calibration of the projector parameters can be initiated by pressing button

[1] http://graphics.cs.msu.ru/en/science/research/calibration/cpp

(e). Finally, button (f) starts the computation of the projector-camera transformation.

The output window in Figure 6.3 shows an example of the projector calibration step, in which the reference points of a projected checkerboard pattern have been detected.

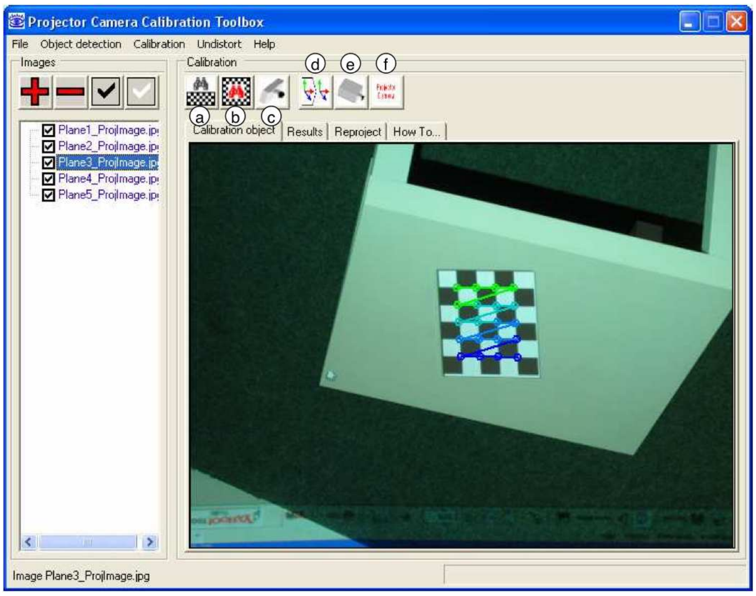


***Figure 6.3**: Projector-camera Calibration Toolbox as a modification of the GML C++ Camera Calibration Toolbox: buttons (a), (b) and (c) adopted from the original interface; buttons (d), (e) and (f) triggering additional functions*

Acquisition of planar surfaces with attached visual markers

As auxiliary means for detecting the position and orientation of surfaces, we use optical markers (ARToolKit[2]), whose location can be detected in respect to the camera using the corresponding software library. The marker is attached to the surface which is to be modeled, and it is captured with the camera (see Figure 6.4). An analysis of the marker image with the ARToolKit software provides a transformation matrix (M_{mc}) representing the relation between marker and camera. With the previously computed projector-camera transformation ($M_{pc} = R_{pc} * T_{pc}$), we can compute the transformation matrix representing the position and orientation of the marker in projector coordinates ($M_{pm} = {M_{mc}}^{-1} * M_{pc}$). This knowledge is used to project a Virtual Display onto the visual marker, with a cross label denoting the center of the display. Now the user can define the margins of the surface that is to be modeled by moving the cross along the surface plane to the corners of this surface. Finally, the positions of the surface corners can be computed in world coordinates using the following equation: $M_{ws} = M_{ps} * M_{wp}$, with M_{ps} characterizing the location of a surface corner in projector coordinates and M_{ws} denoting its location in world coordinates.

[2]http://www.hitl.washington.edu/artoolkit/

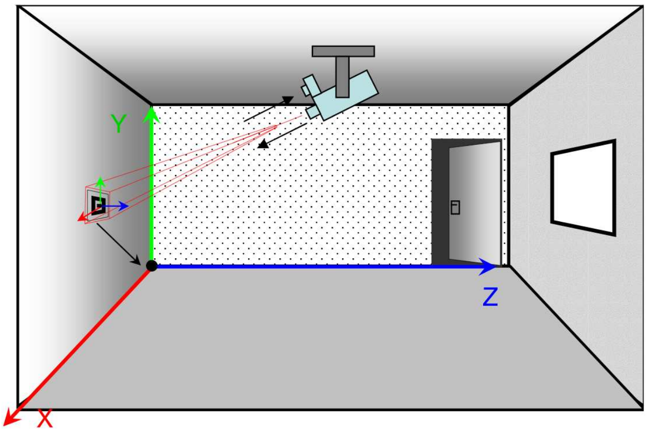


Figure 6.4*: Illustration of the surface geometry acquisition algorithm*

User interface for model acquisition
For enabling a user-assisted model acquisition with the previously described algorithm, we have developed a graphical user interface allowing the user to define individual surfaces and add them to a 3D model. Figure 6.5 shows a screenshot of this interface, which provides the following main functionalities:

- Global configuration of a 3D model
- Detection of the position and orientation of a visual marker attached to a surface
- Translation of a projected label along the detected surface/plane
- Acquisition of surface geometry in a 3D model

The components involved in the global configuration of the 3D model can be found in the frame *Room Configuration*. The button *Create New Model* initializes a new empty model and *Save Model* enables the storage of a model in an xml format. With *Load Model* a previously created 3D model can be loaded in order to be extended by further surfaces. *Initialize World CS* enables the definition of a world coordinate system.

In order to determine the position of a visual marker relative to the predefined world coordinate system, the elements of the *Surface Determination* frame can be applied. *New Marker* initializes the marker detection in a certain area, specified by the pan and tilt values of the steerable unit. The computation of the marker location in the reference coordinate system is triggered by the button *Get Marker Transform*. In order to reduce a possible error in the marker position detection, this computation can be performed several times with different marker positions on the same surface. Finally, *Compute Orientation* triggers the computation of the plane on which the surface is located as an average of the values computed for each marker.

The frame *Surface Reconstruction* contains elements for defining the margins of a surface in a previously detected plane by moving the projected cross label to its corners. For initializing the creation of a new surface, the button *Configure New Surface* has to be pressed.

The button *Create Virtual Display* triggers the display of the cross label as a Virtual Display on the computed plane. With the keys <x>, <y> and <z>, the user can specify the axis on which the cross should be moved to the surface corners. Then, with the arrow keys, the cross label can be moved along the currently specified axis. As soon as a surface corner has been reached, the *Get Corners* button captures the corresponding position in the reference coordinate system. Finally, with *Add Surface To Model*, the current model is extended by the newly specified surface.

Optionally, the surface can be assigned a specific *type* (e.g. wall, obstacle), a *name* and a *reachable from* attribute, which allows the definition of adjacent surfaces. This enables the creation of a connected 3D model, which allows a movement of Virtual Displays across surface borders.

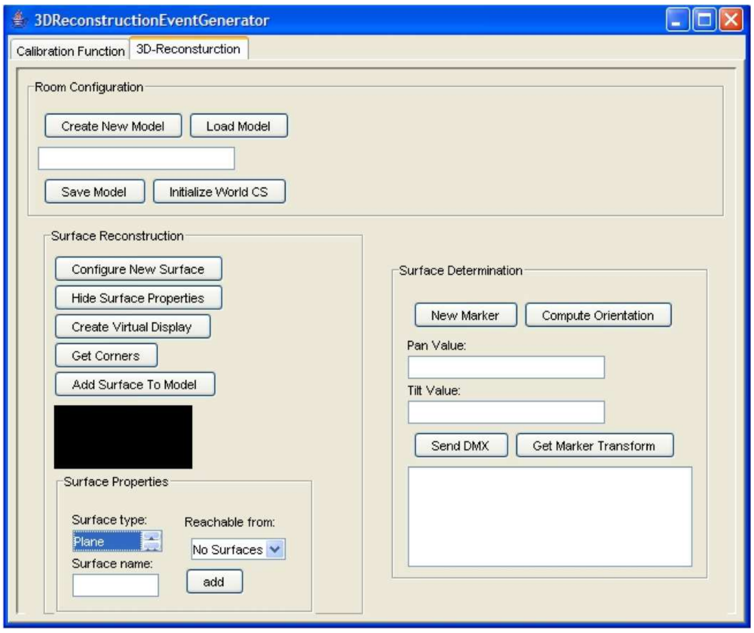


***Figure 6.5**: 3D model acquisition tool*

6.2 DUVD Interfaces

In the course of this work, we have applied various types of interaction techniques for creating and controlling Virtual Displays and Dynamic Peepholes on a Display Continuum. In this section, the developed interfaces are presented in detail, starting with desktop interfaces through to system components for interaction in the physical world.

6.2.1 Interfaces for Desktop Interaction

Keyboard, mouse and monitor are currently still the standard peripherals used with computers. Most people who regularly work with computers are used to handling these devices.

Hence, developing a Display Continuum interface using these input and output devices appears to be an obvious solution. In this way, people can acquaint themselves with the new spatial output concept while applying adapted familiar interaction metaphors.

6.2.1.1 3D Desktop Interface

Our first approach to facilitating the creation and placement of Virtual Displays in a given environment aims at creating a desktop interface which is intuitive and easy to use ([Spassova, 2007]). The proposed solution is based on a visualization of the 3D model[3] of the physical environment, which is primarily exploited for the realization of the Display Continuum (see Section 5.3).

Beside the surfaces that are suitable for projection (like walls, desk surfaces, etc.), the 3D model also contains obstacles (like windows, doors, etc.), on which no projection is possible, according to the concepts described in Section 5.3. Using this model, we have also implemented an algorithm which computes trajectories for the movement of Virtual Displays avoiding collision with obstacles. This path-finding approach allows for example to make a projected virtual character move through the environment following the user in real time (see Section 6.3.3).

In the 3D model of the developed interface, potential projection surfaces are represented in grey and obstacles are rendered in yellow, so that they can be easily distinguished. For creating the 3D model, the DUVD system provides a *Surface* class[4], which can be used to specify individual surfaces of the environment in a predefined coordinate system. An instance of the *Surface* class is defined by the corners of the corresponding surface (as Point3d objects) and the surface type, which can be one of the following:

- *PROJECTION_SURFACE*: representing a surface suitable for projection;
- *OBSTACLE*: representing an obstacle surface, which is not suitable for projection;
- *SHADOW*: representing a surface which is shadowed by another object and thus not suitable for projection;
- *STATIONARY_DISPLAY*: representing a stationary physical display (monitor), which builds a static peephole in the Display Continuum.

According to the concepts formulated in 5.5, each model surface which builds a part of the Display Continuum is assigned a normal vector and a down vector, which are needed for a correct placement of Virtual Displays on it. The direction of the normal vector results from the specified surface corners, which must all lie in the same plane. When a Virtual Display moves across the boundaries of adjacent surfaces of the DC, the normal vector, i.e. the orientation, of the VD is adapted to the one of the underlying model surface. For vertical surfaces, such as walls, the down vector can also be assigned automatically. This parameter is important for determining the initial orientation of a newly created Virtual Display. When a VD is created on a certain surface in the model, its down vector is automatically aligned with

[3]The model is built using Java3D and is displayed in a JFrame programming construct.

[4]The *Surface* class extends of the standard Java3D class Shape3D.

the down vector of this surface. For non-vertical model surfaces, e.g. desks, the down vector, specifying the default VD orientation on them, has to be manually defined by the modeler.

Using the *Surface* class, the 3D model of the desktop interface can be built either manually or it can be generated automatically from a previously created Yamamoto model using a custom *YamamotoToJava3D* converter. The latter approach allows also the incorporation of computed shadow surfaces, which have been determined by means of the *ShadowEngine* module in Yamamoto (see Section 6.2.1.2).

In the visualizing frame, the 3D model can be rotated horizontally and vertically by clicking and dragging with the left mouse button. Clicking with the right mouse button on a projection surface initializes the creation of a Virtual Display in the model. A subsequent dragging, while the mouse button is still kept pressed, opens a rectangular frame representing the outlines of the Virtual Display to be created (see Figure 6.6 [left]). When releasing the right mouse button, the user creates a Virtual Display with the given shape on the Display Continuum, and a representation of it is visualized in the 3D model of the interface (see Figure 6.6 [center]). At the same time, if the interface is connected to a corresponding Fluid Beam device, the Virtual Display is also projected at the equivalent position in the physical environment (see Figure 6.6 [right]). In the 3D model, the display is created as a *VirtualDisplay* object with the defined corner points and it is automatically assigned a unique ID, which is used for referencing.

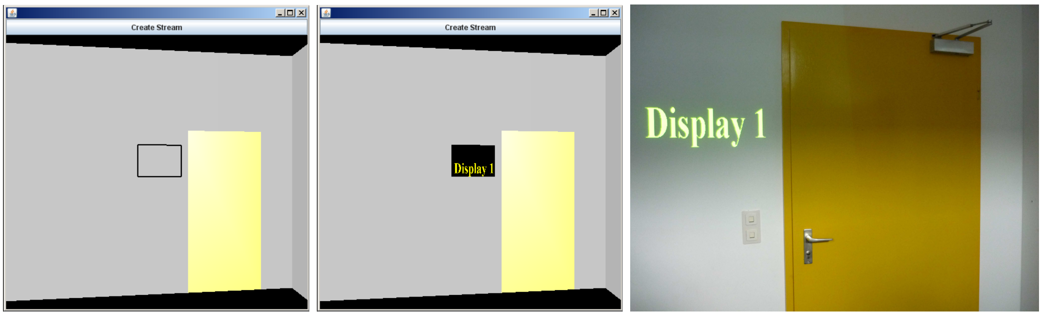


Figure 6.6: *Virtual Display creation: frame representation during mouse drag [left], representation with display ID [center] and corresponding projected Virtual Display in the physical environment [right]*

At this stage, the display does not have any visual content yet, so by default, it shows its own ID (see Figure 6.6 [center] and [right]). The user can now define the display content by drag-and-dropping images or videos from the desktop on the representation of the Virtual Display in the 3D model of the interface (see Figure 6.7). Then, the chosen image appears on both the projected Virtual Display and the corresponding representation in the interface. When a video is shown on the projected display, the representing display in the interface then shows only a movie symbol. The visual content of a Virtual Display can be re-defined at any time. By dropping a new image or video file on the display representation, the old content is replaced.

Additionally, the 3D interface offers also the possibility to assign a video stream from the desktop to a Virtual Display. Clicking on the *Create Stream* button of the interface opens a

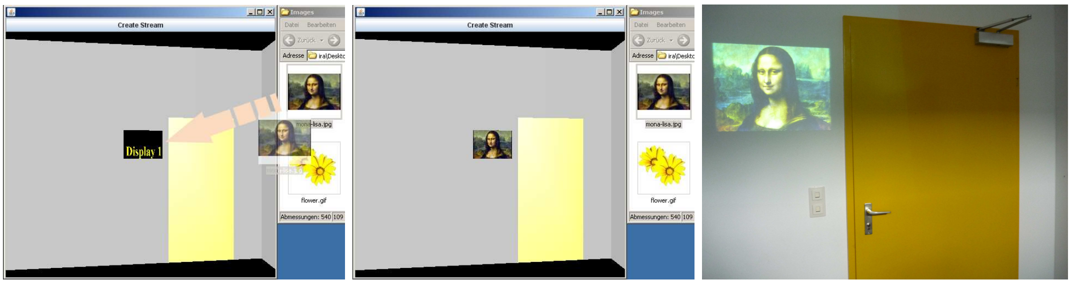


***Figure 6.7**: Defining display content using drag and drop: image file being dragged [left], image set as display content in the interface [center] and projected on the corresponding Virtual Display in the physical environment [right]*

new frame, with which the user can specify the area on the desktop which is to be streamed to a corresponding display. After an optional adjustment of the frame's boundaries to the desired desktop content, the user can click on the *Stream to* drop-down menu, which contains the IDs of all currently available Virtual Displays, and assign the video stream to one of them. Alternatively, for simplification, there is also the possibility to stream the entire desktop as Virtual Display content independent from the stream-defining frame, using the *Full screen to* menu, which also contains all current Virtual Display IDs.

An extended version of the stream creation interface offers an additional *Show On Display* button, which crates a new Virtual Display showing the selected part of the screen. This Virtual Display, which is initially located on a stationary screen, can then be dragged to the adjacent projection-based Display Continuum (see Figure 6.8). This embedding of physical screens into the projection-based DC is further used in the accelerometer-based gesture interaction module described later in this chapter (see Section 6.2.2.2).

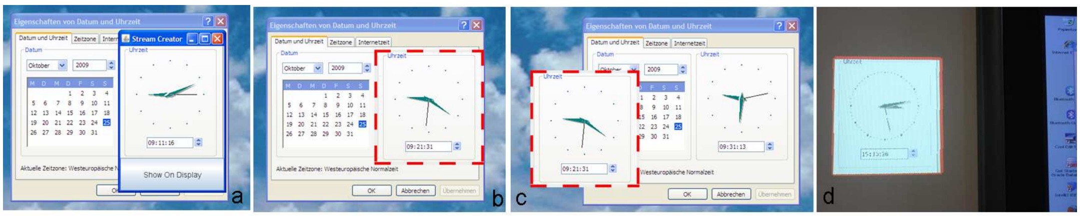


***Figure 6.8**: Virtual Display creation using a live stream form a stationary screen: (a) stream creation frame specifying a location on the screen showing a clock application, (b) Virtual Display created on the screen showing the specified stream, (c) Virtual Display moved on the screen, (d) Virtual Display projected on the Display Continuum beside the physical screen*

After a Virtual Display has been created, it can be moved to a new position by clicking and dragging with the left mouse button on the corresponding representation of the display in the 3D interface. The projected Virtual Display in the physical environment and the corresponding display in the 3D model then move in parallel and change their orientations according to the orientation of the surface they are currently placed on, i.e. in particular, the displays flip around their vertical axes as soon as they move across a room corner, which represents a discontinuity in the Display Continuum.

Furthermore, the 3D interface offers also the possibility to discretely change the size and the position of a Virtual Display in a menu that appears after a left click on the corresponding display representation in the 3D model (see Figure 6.9). This allows a more exact adjustment of the parameters than with the mouse dragging interaction.

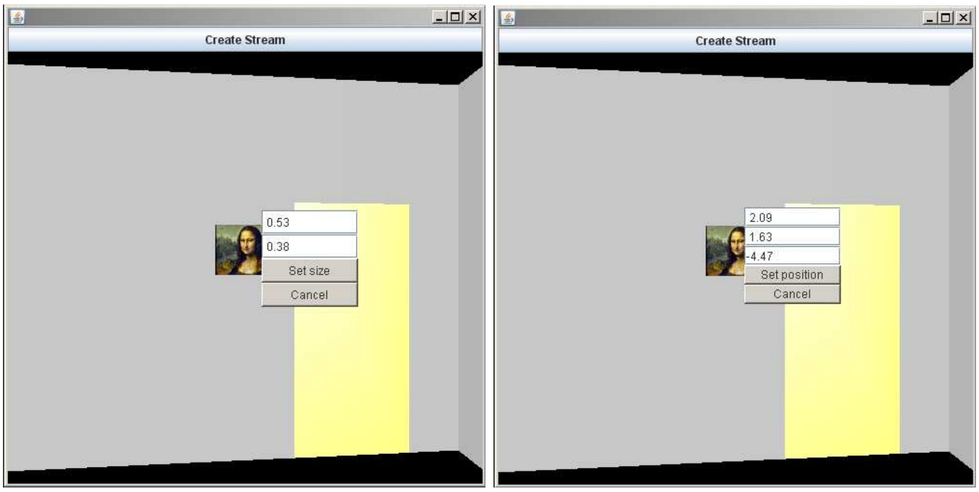


***Figure 6.9**: Menus for size [left] and position adjustment [right]*

6.2.1.2 Yamamoto Extension with Shadow Simulation

The Java-based 3D interface described in the previous section allows a comfortable manipulation of Virtual Displays in a desktop application. However, it does not provide an opportunity to interactively build or modify the 3D model itself, which represents the Display Continuum. In order to overcome this limitation, we decided to take an existing graphical modeling framework, which offers appropriate 3D modeling capabilities and extend it through specific modules in order to enable the modeling of steerable projection devices and the corresponding Virtual Displays.

The framework which was chosen for this purpose is the modeling toolkit *Yamamoto* (Yet Another MAp MOdeling TOolkit) ([Stahl and Haupert, 2006], [Stahl, 2009], [Stahl and Schwartz, 2010]). It has been developed to support the modeling, design and development of user assistance systems in Intelligent Environments. The focus of this tool is on the geometric modeling of physical environments and their instrumentation with sensors and actuators in 3D. Figure 6.10 shows a typical view of the Yamamoto editor with the 3D visualization of a currently chosen model (here: IRL, see Section 2.2) and an editing menu for the adjustment of certain model parameters on the right side of the interface.

In the course of the present work, the Yamamoto framework was extended by several classes in order to realize the modeling of steerable projectors and Virtual Displays. Additionally, the module *ShadowEngine* was developed, which realizes the computation of shadowed surfaces on a given Display Continuum according to the definition in Section 5.3. The extension also establishes a connection between Yamamoto and the Fluid Beam software of

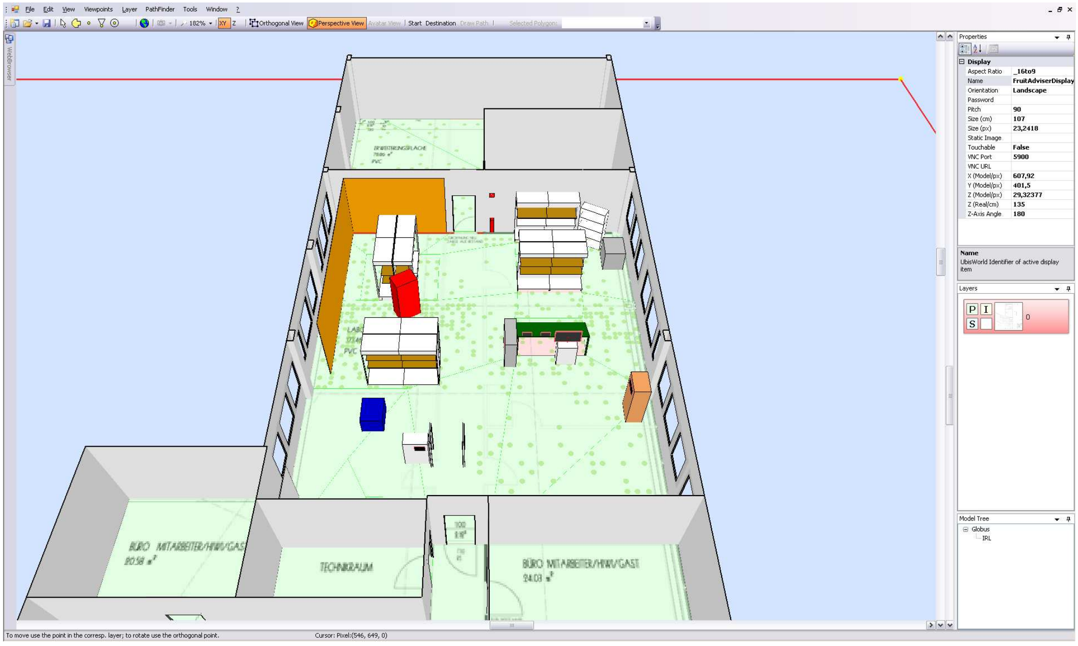


***Figure 6.10**: Yamamoto editor*

the corresponding steerable projector, so that the Yamamoto editor can be used as a further 3D interface to control the projector and the Virtual Displays similarly to the 3D interface described in the previous section.

Furthermore, a *YamamotoToJava3DConverter* is provided, with which models stored in yml-format[5] can be transformed into Java3D models for use in the previously introduced 3D interface.

Although Yamamoto allows a full 3D visualization of the modeled geometry, the actual modeling process and the stored yml-models are only in 2,5D. This means that e.g. the walls of a room are defined in the form of 2D edges on a floor plan and a global parameter defines the wall height. This representation has originally been chosen by the Yamamoto developers in order to facilitate the modeling of building geometry out of corresponding architectural floor plans. However, for the geometric computations which have to be performed by the *ShadowEngine* module, this representation is unsuitable.

For this purpose, we have developed a new class hierarchy for the representation of wall surfaces. It consists of the classes *WallUnit*, *WallBox* and *WallPlane*, where

- *WallUnit* is the basic wall element, which represents a continuous wall section, i.e., it encompasses only concatenated Yamamoto wall edges which all lie in the same plane;
- *WallBox* contains WallUnits which lie sufficiently close together in the same plane;
- *WallPlane* encompasses all WallUnits lying in the same plane.

[5] Yml is a proprietary xml-based storage format developed for the representation of Yamamoto models.

An element of type WallBox actually represents what people intuitively perceive as a complete wall, even if it might be interrupted by a narrow gap or obstacles in the form of doors or columns. The elements of type *WallPlane* are necessary for the computation of shadows which do not entirely fall on one single wall. In general, the outlines of a thrown shadow are computed on a particular *WallPlane* and subsequently cropped at the boundaries of the corresponding *WallBox* objects.

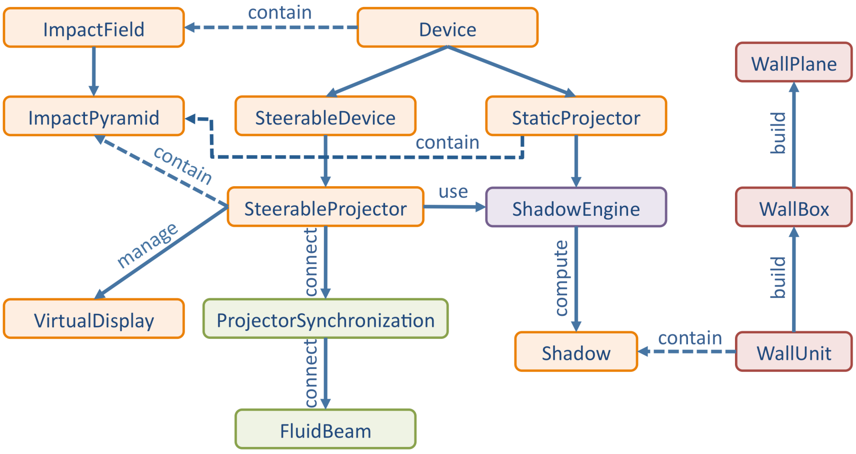


Figure 6.11: *Architecture of the Yamamoto extension for modeling and control of steerable projectors and Virtual Displays and for shadow simulation*

Figure 6.11 illustrates the architecture of the Yamamoto extension. In order to provide a generic interface for the modeling of a variety of interactive devices, which might be integrated in an Instrumented Environment, the superclass *Device* has been defined. In addition to some position parameters, it contains an instance of type *Box* representing the device's *body* and an instance of type *ImpactField* representing the physical range of the device. Objects of type *ImpactField* can have different shapes, e.g., the electromagnetic field of an RFID antenna can be ideally represented as a sphere, the scope of an infrared beacon is typically conical, and the beam of a projector has the shape of a pyramid.

As in the present work, we mainly focus on projectors as interacting devices, we have developed the class *ImpactPyramid* as a subclass of the generic class *ImpactField*. The *Device* subclasses *StaticProjector* and *SteerableProjector* both contain a parameter of type *ImpactPyramid* representing the corresponding projector beam taking into account the aperture angle, the projection range and the aspect ratio of the respective projector. In order to facilitate the future incorporation of further steerable devices, like e.g. steerable cameras, into the Yamamoto framework, we provide the generic class *SteerableDevice* as a subclass of *Device* and a superclass of *SteerableProjector*. It contains parameters and functions concerning the spatial adjustment of steerable devices.

For the representation of Virtual Displays, we have implemented the Yamamoto class *VirtualDisplay*, which corresponds to the respective display class in the DUVD core system. Figure 6.12 shows an example of a Virtual Display represented in Yamamoto. In the properties menu of the Yamamoto editor, the parameters of the selected Virtual Display can be

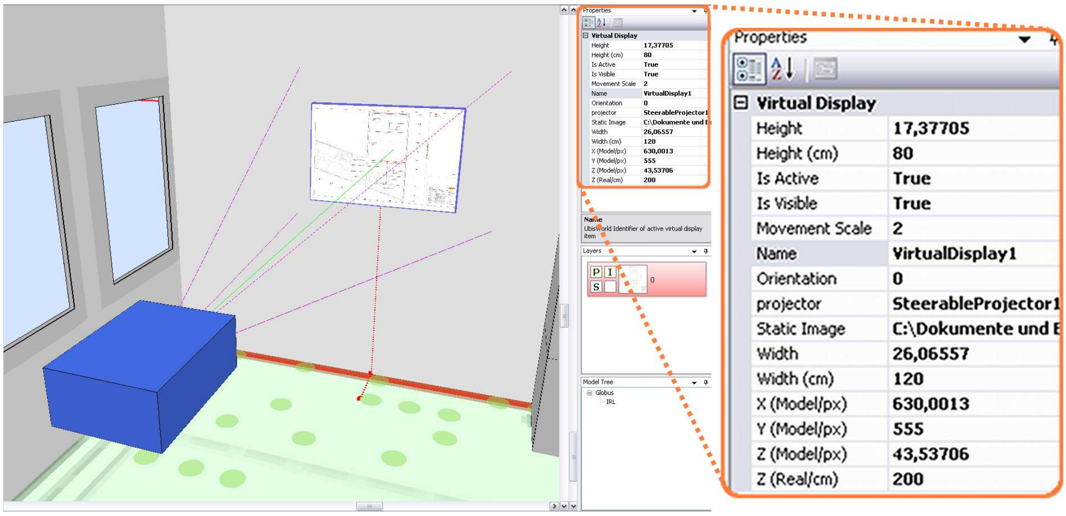


***Figure 6.12**: Virtual Display in a Yamamoto model*

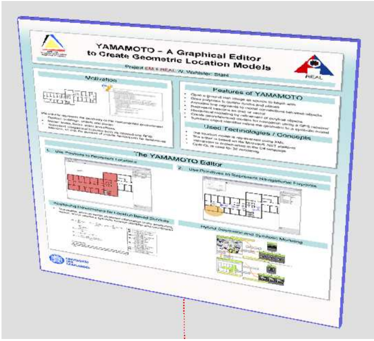
accessed and modified if needed.

As already described in Section 5.3, Virtual Displays can exist in four different states depending on their activity status (active/inactive) and their visibility status (visible/invisible). In its Yamamoto counterpart, the current state of a Virtual Display is denoted by minor changes in the visual representation. An inactive Virtual Display is rendered slightly transparent, and the content of an invisible Virtual Display is grayed out to a certain degree. The combination of the activity and visibility states results in four different possible representations, which are illustrated by way of example in Figure 6.13.

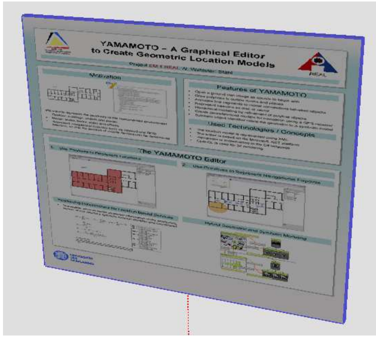
Furthermore, the *ProjectorSynchronization* represents an interface for a direct propagation of user interactions in Yamamoto to the corresponding steerable projection system in the physical world. Similar to the approach of the previously described 3D desktop interface, this synchronization of virtual model and real world elements implements the Dual Reality concept proposed in Section 5.3.

The module *ShadowEngine* uses the hierarchical wall structure built by *WallUnits*, *WallBoxes* and *WallPlanes*, and the *SteerableProjector* objects provided by a given model in order to compute the regions on the Display Continuum which cannot be reached by the projector beam of this steerable projector (*shadows*). The algorithm computes individual shadows for each object present in the projection area of the involved steerable projector, and finally, these shadow fragments are merged to connected regions, which can be visualized in the Yamamoto editor.

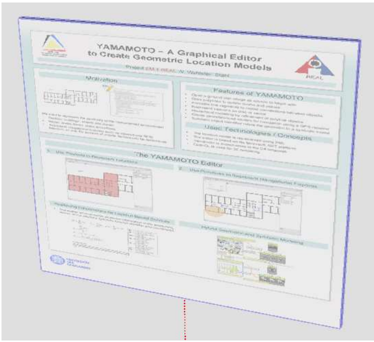
The shadows computed by the *ShadowEngine* are visualized as semi-transparent surfaces in the 3D model in the Yamamoto editor. The color of the shadow surface matches the one of the corresponding projector representation. They can be displayed in the editor in a *static view* or in a *live view*. In the former case, the shadow representation is updated after the user has finished the movement of a *SteerableProjector* or another object in the 3D model. In contrast, the live view implements a continuous update of the projector shadows during the movement of the corresponding steerable projector device. In this way, a modeler can be



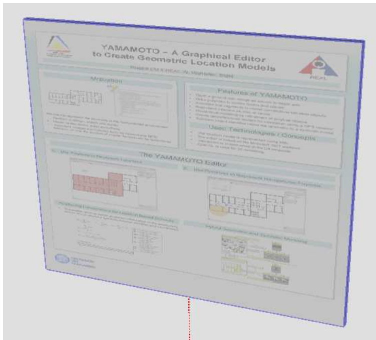
(a) active and visible



(b) active and invisible



(c) inactive and visible



(d) inactive and invisible

***Figure 6.13**: Visualization of Virtual Display states in Yamamoto*

supported in the selection of an appropriate mounting location for a steerable projector in a newly established Instrumented Environment. By moving the *SteerableProjector* object in Yamamoto, the user can observe the current shadow state and thus find an optimal location for the installation of the steerable projector in the physical environment. While searching for an appropriate mounting location for a steerable projector, potential goals can be, for example, a minimization of the resulting shadow surfaces or an optimal illumination of important surfaces, on which projection is necessary. The static view mode, on the other hand, is less computation-intensive and thus more efficient for complex 3D models.

A screenshot of the Yamamoto editor with shadow visualization for a steerable projector can be seen in Figure 6.14.

The *ShadowEngine* module allows also the simulation of several steerable projectors in the same environment. In order to be able to distinguish the shadow surfaces corresponding to each *SteerableProjector*, the computed shadows are displayed in different colors, matching those of the respective projector objects (see Figure 6.15 [left]). When several steerable projectors have to be installed in one room, it is often interesting to see which surfaces cannot be reached by neither of the projectors in a given setup. For this purpose, the *ShadowEngine* module offers the opportunity to switch to a *combined shadow* mode, in which the shadows of all projectors are intersected resulting in an overall shadow representation (see Figure 6.15 [right]).

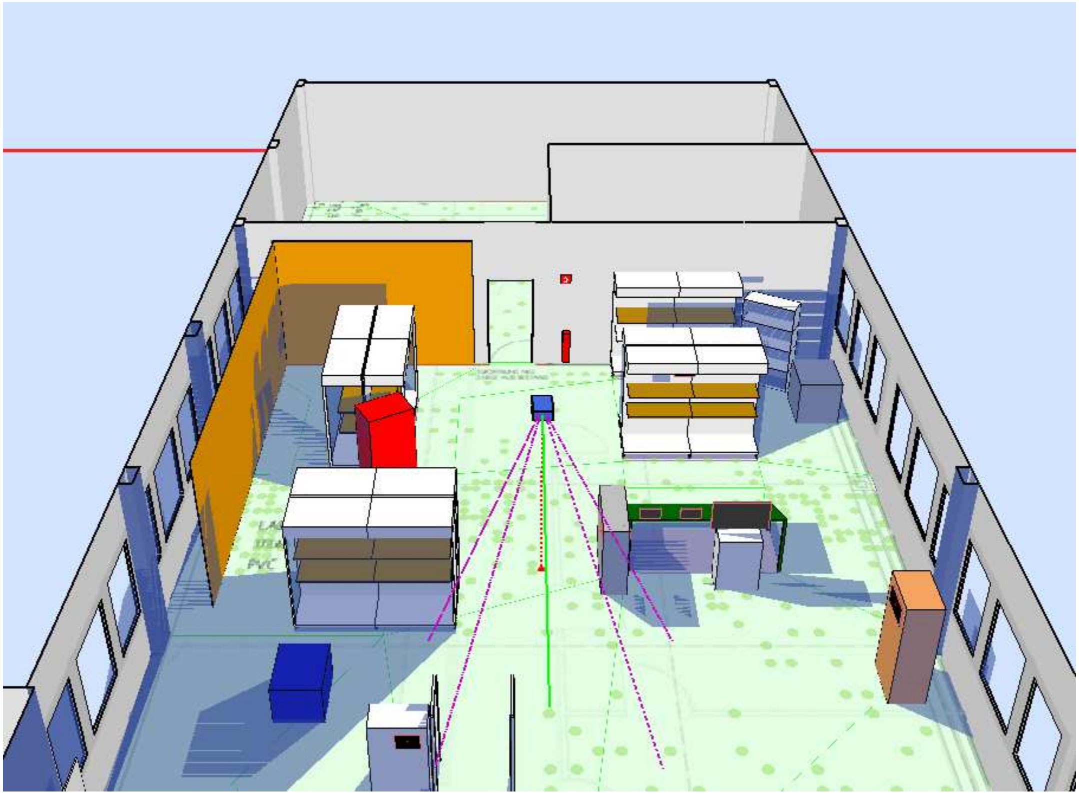


Figure 6.14: *Shadow representation in Yamamoto*

6.2.2 Real-world Interfaces for Gesture Interaction

As the output of the DUVD system is presented in the physical surroundings of the user, it seems natural to provide also input interfaces for interaction in the real environment instead of working with a visualized model. In this way, the user is not restricted to working only on the limited space of a traditional desktop monitor, but he should be enabled to interact with the system while moving freely through the Instrumented Environment.

For the development of the real-world interfaces introduced in the following sections, our aim was to attach as little instrumentation to the user as possible. Instead, the user should be given the opportunity to interact using common objects and gestures, which are familiar to him.

6.2.2.1 Vision-based Interaction

One way to observe user behavior in the environment and thus to detect user-issued commands to the DUVD system is provided by the exploitation of vision-based techniques. For this purpose, appropriate cameras have been installed at certain locations in the user's surroundings. On the one hand, camera sensors are mounted directly at locations where user interaction is expected, which are used to observe single interaction areas. On the other hand, cameras are attached to the steerable projector unit, thus taking advantage of its steer-

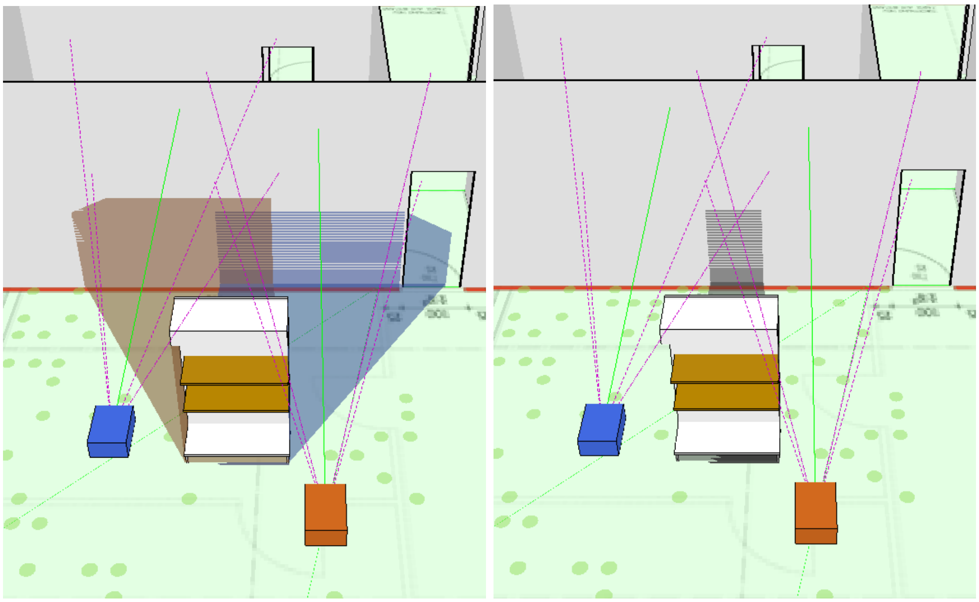


***Figure 6.15**: Individual shadows for each projector [left] and combined shadow for both projectors [right]*

ability property in order to direct the camera view at different areas of interaction. In both cases, the captured user interaction is identified and interpreted in a specific way.

6.2.2.1.1 Interaction Using Projector-mounted Camera

This interaction module allows users to control the DUVD system by means of ordinary colored objects. The user is free to select an interaction object at the beginning of an interaction sequence, however, the chosen object has to comply with certain conditions concerning its from and color in order to be suitable for interaction. After an object has be recognized and accepted as an interaction object by the system, the user can perform certain gestures with this object on the Display Continuum in the physical environment in order to create and manipulate Virtual Displays and adjust the location of the Dynamic Peephole.

A similar interaction approach with a device called “magic wand” is presented in [Ciger et al., 2003]. It is designed for user interaction with back-projection-based Virtual Environments. In this case, the magic wand is an elongate stick which is tracked using an electro-magnetic tracking system (Ascension Flock of Birds[6]). In contrast, the recognition of the interaction object in the present work is realized by analyzing the video stream delivered by a camera mounted below the projector of the steerable unit. In this setup, the camera captures the whole projection area of the steerable projector device, which in this case builds the Dynamic Peephole visualizing the Display Continuum.

[6]http://www.ascension-tech.com/realtime/RTflockofBIRDS.php

Detecting and tracking colored objects

As already mentioned, not every object is suitable for interaction in the present approach. Interaction objects must have the following characteristics:

- *Sufficient size*: The interaction object must be large enough to be recognized in the camera image. In order to compensate for image noise, it must be possible to distinguish between the interaction object and other small regions with a similar color value.

- *Uniform color*: The interaction object should have one base color, which is registered for recognition; otherwise, depending on the portion of other colors and the orientation of the object, the visible part containing the registered color might not be large enough for being recognized in the camera image.

- *Unique color*: The color of the interaction object should differ as much as possible form other colors present in the environment in order to achieve a clear identification.

- *Clear orientation*: The shape of the interaction object should allow a recognition of the object's orientation, i.e., one of its dimensions must be larger than the others. For this reason, stick-like objects are preferred.

In order to detect if a specific object is suitable for interaction, we have implemented a number of methods for automatically checking the relevant object parameters during registration. The size of the interaction object e.g. is estimated by the size of the bounding box of the recognized color region. This size value is compared to a predefined reference value, and the object is accepted only if the size of the bounding box exceeds this value.

In order to guarantee the uniqueness of the interaction object color, we build a color signature of the background in the form of a color histogram representing the occurrence frequency of each hue value (in the HSV color space). After the recognition of a potential interaction object, its hue value is looked up in the histogram, and the object is accepted only if the corresponding frequency value is sufficiently low. If the recognized object has an unsuitable color, it is possible to recommend some alternative colors to the user by projecting spots with the least frequent hue values on the wall.

The suitability of a detected object regarding its shape can be estimated by analyzing its outlines after applying a Hough transform[7] ([Duda and Hart, 1972]). The Hough-transformed image of an object with a clear orientation will result in a high number of lines with similar angles (parallel or almost parallel lines). In contrast, if we take a round or arbitrarily shaped object, the Hough transform will detect lines in various different directions. Hence, the number of different line angles can be taken as a measure for the suitability of a shape for interaction.

The uniformity of an object's color as such is difficult to detect. However, the more different colors an object has, the greater is the probability that one of the other constraints is violated. If we have a number of small surfaces with different colors, the object will be probably rejected due to the size constraint. Moreover, a complex color pattern on one object

[7] The classical Hough transform is concerned with the identification of lines in images.

will lead to the detection of many different line angles after Hough transform, so that such an object is also likely to be discarded due to the size constraint.

According to the above constraints, everyday objects like e.g. a uniformly colored stylus, a closed umbrella or even a cucumber can be used for interaction (see Figure 6.16).

Figure 6.16: *Possible interaction objects for real-world interaction*

In the following, we will describe the algorithms used for the registration, recognition and tracking of an interaction object, and we will specify the set of gestures used for interaction.

In order to be able to use a certain object as an interaction device, it has to be registered as such before starting the interaction process. In general, there are different ways to register an object in a vision-based system. One way is to make the user actively specify the object which has to be recognized by the system, e.g. by placing it in the camera view and choosing one representative pixel in the image, e.g. by clicking on it. This approach leads to relatively robust object recognition results, however, it requires additional interaction devices, e.g. a computer mouse.

In order to simplify the object registration process, we have implemented an automated interaction object recognition. It allows the registration of an interaction object by simply holding it in a predefined region in front of the projection surface before starting the interaction. At this point, the system performs a constant image subtraction and thus recognizes changes in the image if a new object is placed in the dedicated region. As soon as a new object is recognized, the system compares its color (hue value) with the colors appearing in the background image. In case the new color sufficiently differs from the background colors and the other previously described conditions are fulfilled, the corresponding object is accepted and automatically registered as interaction object. This algorithm performs best in front of a uniformly colored wall.

In order to indicate where the interaction object is expected to be placed, the corresponding area on the wall is marked by a projected square. As soon as an interaction object has been recognized and accepted, the user is provided visual feedback in form of a projected OK sign, which disappears again after a few seconds.

After a specific object has been registered as interaction object, it has to be constantly identified and tracked in the camera image in order to be able to recognize user interaction with it. For this purpose, it must be possible to identify uniformly colored areas (also called "blobs") in a video stream. The Blob Detection Library[8] offers appropriate functionality for recognizing areas of similar brightness in video streams. However, as this library allows only

[8]http://v3ga.net/processing/BlobDetection/

a blob detection depending on the brightness of the image areas and not the color, the image that we receive from the projector-mounted camera has to be preprocessed in several steps:

1. In order to reduce image noise, a Gaussian filter is applied on the camera image.
2. The image is converted from RGB to HSV color space, so that the hue value, which is invariant to illumination brightness, can be used for color segmentation.
3. Using a thresholding algorithm, all pixel which have a hue value similar to the one of the previously registered interaction object are segmented from the rest. Relevant pixels are marked white and the others are set to black (see Figure 6.17).

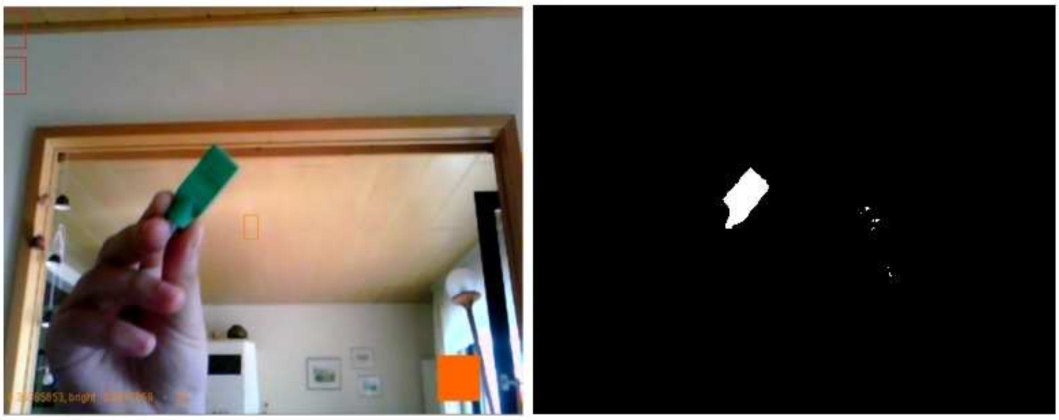


***Figure 6.17**: Image segmentation: original image with green interaction object [left] and result after segmentation [right]*

In order to speed up the preprocessing of the image, the above steps are not applied to the whole image delivered by the camera but only to a certain rectangular area surrounding the previously detected location of the interaction object. If this area contains several blobs which might potentially represent the interaction object, then the one which appears closest to the previously detected object location is chosen.

Interaction method

The interaction paradigm implemented in this module is based on the location and posture of the interaction object detected in the camera image. In [Kendon, 2001], Adam Kendon defines gestures as “excursions”, which means that sequences of “action recognized as ’gesture’ move away from a ’rest position’ and always return to a rest position”. We use this concept in order to distinguish intended user gestures from random movements of the interaction object. Interaction gestures are initialized by keeping the interaction object still in a specific posture for a certain period of time. Depending on the spatial context and the posture, the subsequent movement of the object is interpreted as a certain gesture, which is then transformed into a corresponding command. The gesture is regarded as finished as soon as the interaction object has been again kept still for a certain period of time.

In order to avoid that any standstill of the interaction object in the camera image leads to a (possibly unintended) gesture initialization, we trigger a new gesture only if the object has been held either in a horizontal or in a vertical position. We could observe that these two postures occur very rarely when the user is just holding the object without the intention to interact with the system.

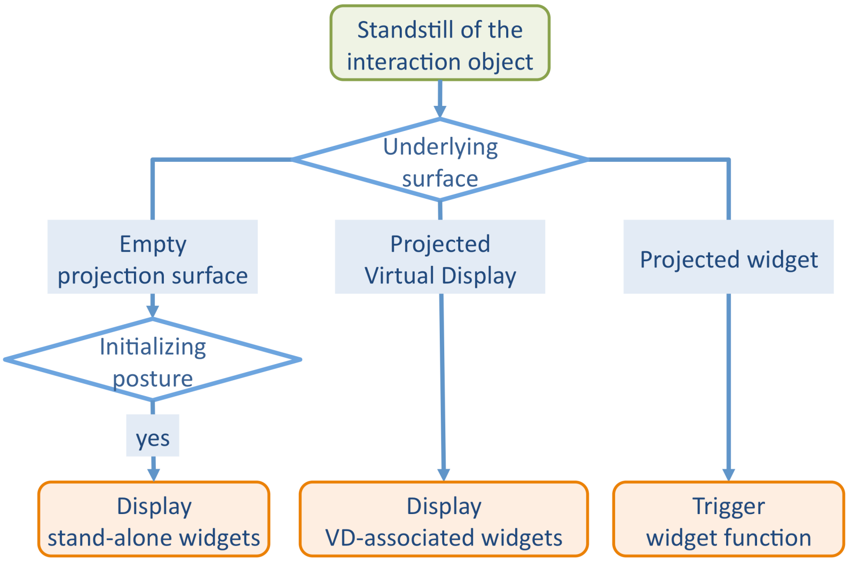


***Figure 6.18**: Gesture initialization and widget activation flowchart*

A further possibility to initialize a gesture is provided by holding one part of the interaction object in front of a projected Virtual Display or in front of a projected widget associated with a specific function. Depending on its type, a projected widget can be either linked to a Virtual Display or stand-alone. If the interaction object is standing still, the system checks if at least two corners of the interaction object detected in the camera image lie inside the boundaries of a Virtual Display or of a projected widget surface as they appear in the camera image frame. In the latter case, the user interaction is interpreted as a pointing gesture on the widget, and thus the corresponding function is triggered. If the interaction object is detected in front of a Virtual Display, the VD-associated interaction widgets are displayed. If the interaction object appears in front of an empty projection surface in an initializing posture, the stand-alone widgets are projected. An overview of this program flow is illustrated in Figure 6.18.

In our module, we have implemented the following interaction widgets:

- *Stand-alone widgets*: These widgets are displayed when the interaction object is detected in an initializing posture and it is not being held in front of a Virtual Display or a widget. In the current version two stand-alone widgets have been implemented (see Figure 6.19 [left]).
  - *DP movement widget*: initializes a movement of the Dynamic Peephole; after

the widget has been activated, the visual peephole is moved along the Display Continuum, following the movement of the interaction object.

  - *VD creation widget*: initializes the creation of a new Virtual Display with a fixed size.

- *VD-associated widgets*: These widgets appear beside a Virtual Display if the user is pointing on it with one tip of the interaction object for a certain period of time (see Figure 6.19 [right]).

  - *VD movement widget*: initializes a movement of the associated Virtual Display; similar to the stand-alone DP movement widget, after an activation of the widget, the Virtual Display is moved along the Display Continuum according to the movement of the interaction device.
  - *Rotation widget*: initializes a rotation of the associated Virtual Display; in this case, the rotation angle of the Virtual Display corresponds to the rotation angle of the interaction object detected in the camera image.

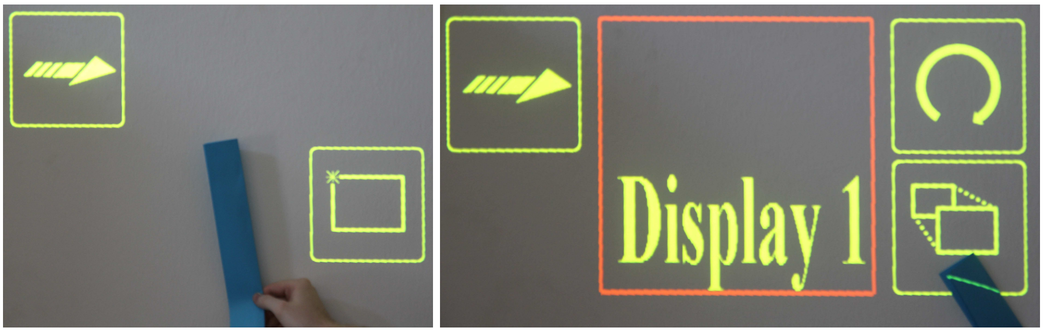


Figure 6.19: *Projected widgets for interaction with colored objects: stand-alone widgets [left] and VD-associated widgets [right]*

6.2.2.1.2 Interaction Using External Camera

In some cases, when user interaction is expected to take place at a specific location, it is more appropriate to detect the interaction process using an external camera, which completely covers the interaction area.

In the course of this work, we have developed and implemented a module for recognizing pointing gestures performed by the user (deictic gestures). The interaction module PEG (Pointing Extra Gesture) detects the user's hand in 3D space and hence deduces the object at which the user is pointing. For capturing the user's hand, we apply a PTZ (pan-tilt-zoom) network camera by Axis[9], which is mounted on the ceiling above the interaction area. The camera image can be captured and its orientation and zoom can be controlled remotely over internet. In order to be able to detect the hand and its position in space, the user has to wear a colored glove with a visual marker attached to it (see Figure 6.20).

[9] http://www.axis.com/products/cam_2130/

Figure 6.20: *Colored glove with ARToolKit marker for recognizing pointing gestures*

In this setup, the colored glove is tracked by the PTZ camera using a color detection algorithm similar to the one which is used for detecting the interaction object described in the previous section. The additional visual marker enables the detection of the 3D position and orientation of the hand. In our implementation, we use an ARToolKit[10] marker for this purpose, which comes with a software library including tracking functions in 3D space in relation to the camera position ([Kato and Billinghurst, 1999]).

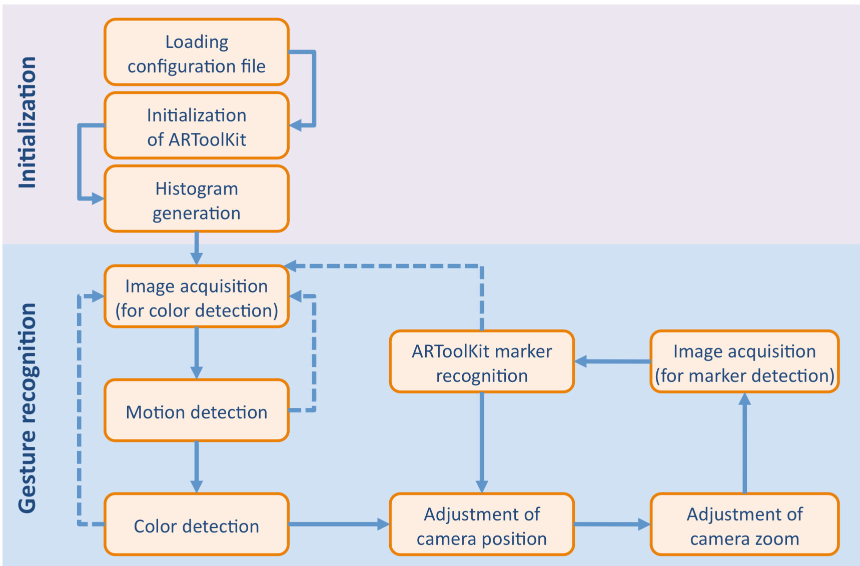


Figure 6.21: *Process flow of the PEG module: continuous arrows specify the next step after a successful completion of the current phase; dashed arrows indicate the fallback case*

Figure 6.21 gives an overview of the gesture recognition process of the PEG module. After an initialization phase at the beginning of the application, the camera image is constantly captured and analyzed. As soon as motion is detected in the image (using background

[10]http://www.hitl.washington.edu/artoolkit/

subtraction), the concerned image region is scanned for the currently used glove color. If an appropriately colored object has been found in this region, the camera position is adjusted so that this object appears in the center of the image. After that, the camera zooms in on the presumed colored glove in order to recognize the attached ARToolKit marker and thus to detect the exact hand position and orientation in 3D space. If the user's hand is moving, the orientation and the zoom of the camera have to be constantly readjusted. In case the user's hand is lost during this process, so that the marker or color detections do not provide satisfying results, the system returns to a previous step (indicated by dashed arrows in Figure 6.21).

Some exemplary results of the motion detection process performed as part of the recognition of a gloved hand are presented in Figure 6.22. The upper left image is the result of the pixelwise background subtraction. This image is binarized using a predefined threshold value in order to compensate for possible changes in the lighting conditions, which results in the upper right image. In a next step, this image is used as input for a blob detection process in order to identify the regions containing movement. In the lower left image of Figure 6.22, the single blobs are marked with bright blue rectangles; blobs which are sufficiently close together are combined to motion regions (dark blue rectangles). In this step, motion regions which are smaller than a given threshold are discarded. The lower right image of Figure 6.22 shows the binary image after the motion region filtering.

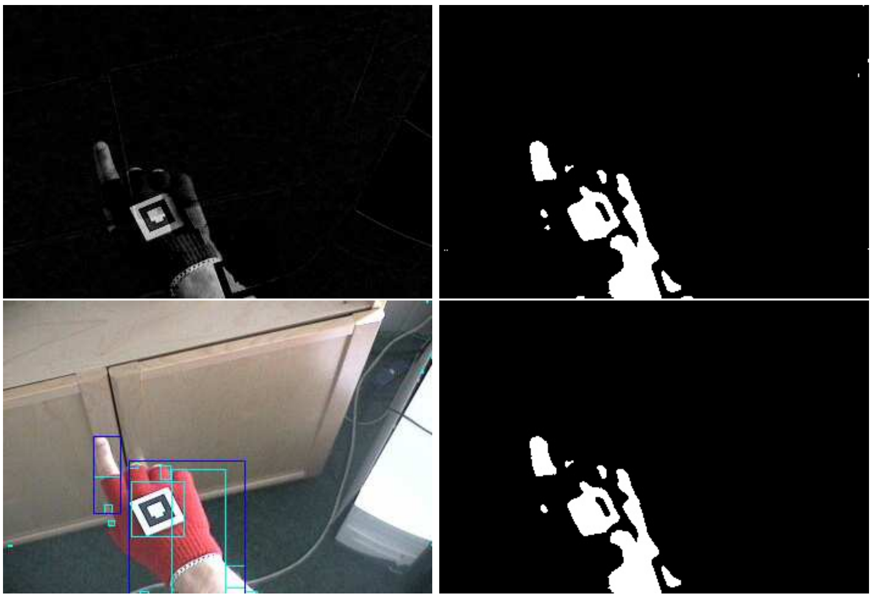


Figure 6.22: *Motion detection with the PEG module: background subtraction [upper left]; binarized image [upper right]; detected motion blobs (light blue) and combined regions (dark blue) [lower left]; filtered binary image [lower right]*

The PEG module serves as an input modality for the DUVD system and provides a POINTING_GESTURE event every time a gloved user hand is recognized and localized. This event contains the 3D position and orientation of the hand in the DUVD coordinate system. Using this knowledge, the system can create a virtual ray in the 3D model with the given origin and direction and thus spot the virtual representation of the object the user is pointing at. In this way, respective system feedback can be provided in form of a projected Virtual Display at an appropriate location. The process of computing the targeted location out of the detected marker position in the camera image is outlined in Figure 6.23.

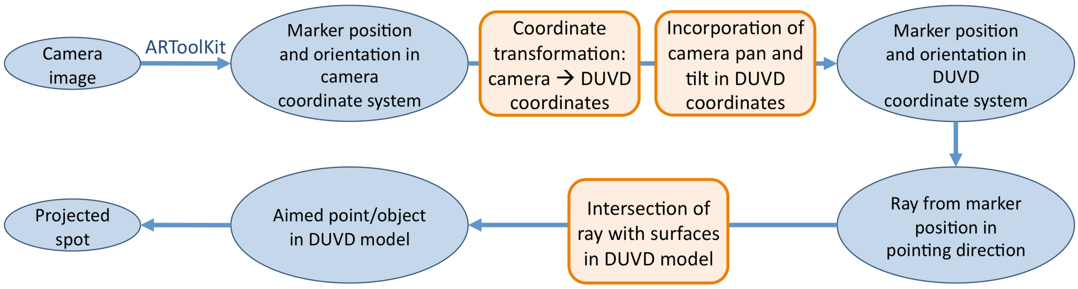


***Figure 6.23**: Coordinate transformation for detected ARToolKit marker from camera coordinates to DUVD coordinates in PEG module*

This interaction method is applied as one possible user input modality for the Mobile ShopAssist (MSA) ([Wasinger et al., 2005], [Wasinger, 2006], [Kahl et al., 2008]) in conjunction with the DUVD system. The MSA is a multimodal shopping assistance system running on a handheld device, which offers customers the opportunity to request information about products using different input modalities (speech, writing and clicking) or a combination of those. For output, the MSA also provides several modalities, such as speech and written text. The connection to the DUVD system offers further input and output modalities for the MSA in the physical world, namely the Pointing Extra Gesture (as input) and projection in form of Virtual Displays (as output). In this context, the term *extra gesture* is used in contrast to the pointing *intra gesture*, which denotes a selection click on the touch screen of the handheld device on which the MSA is running.

Using the PEG module with the MSA, the customer can point on a product he is interested in, while standing in front of a shelf. The DUVD system holds virtual representatives of the corresponding products in the 3D model, and thus the aimed product can be identified as described above. System feedback is carried out in form of a projected spot on the product in the shelf, which indicates that the system has recognized the pointing correctly. Taking advantage of the multimodality of the MSA, more specific input and output can be accomplished by adding e.g. the speech modality. A typical interaction sequence can then look as follows: If a customer wants to ask for the price of a product, he can point (PEG) at it and utter “How much does it cost?”; the system response consists of a spot projected at the recognized product and a speech utterance, like “This camera costs 99 euros”.

The PEG module offers also the possibility to recognize several users depending on the different ARToolKit markers attached to their gloves. Two examples of different user input and system output are presented in Figure 6.24. The users are assigned different spot colors in order to be able to distinguish, which system feedback (projected spot) is intended for whom. In this way, the PEG module supports multiuser interaction to a certain extent, i.e. within the scope of the external camera and the current projection beam of the steerable unit.

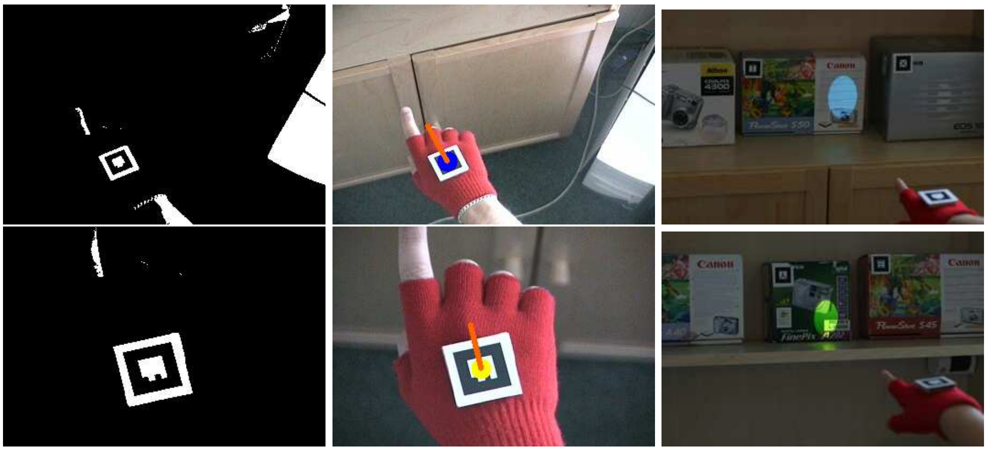


Figure 6.24: *Pointing gesture recognition with different users (yellow and blue): binary image generated during hand recognition [left], visualization of the recognized hand position and orientation [center] and projected spot at the corresponding product in the shelf [right]*

6.2.2.2 Accelerometer-based Interaction

For many applications, vision-based human-computer interaction offers an appropriate means for user input as, ideally, it can be realized without any technical instrumentation on the user, e.g. by applying depth-sensing cameras. However, the deployment of vision-based techniques shows some technical limitations concerning especially the need for a line of sight between the camera and the interaction object/hand, which cannot always be guaranteed, and the image quality, which depends on different factors, like camera resolution and lighting conditions, which are often not optimal.

In order to be able to provide a further approach for user interaction with the DUVD system as an alternative to the vision-based techniques, we decided to investigate other sensors with which user gestures can be captured for system input. In this context, accelerometers offer suitable capabilities as they can provide direct feedback about the movement of the object they are attached to.

In the appendix of this work, two accelerometer-based interaction devices are presented, namely the TZI SCIPIO Gesture Band and the WInspect Glove, which have been developed in the wearIT@work project (see Appendix A.1). These wrist band and glove-like devices are fitted with 3D accelerometers, with which the relative movement of the user's hand in space can be captured. Although we intended to apply one of these devices as an input interface for controlling the DUVD system using gestures, it turned out that the SCIPIO devices have been developed especially for the wearIT@work project and have stayed only prototypes which are not commercially available.

An alternative, commercially available interaction device providing sensing functionalities similar to those of the SCIPIO devices has been launched by the Nintendo corporation in conjunction with their gaming console Nintendo Wii. It uses s wireless motion-sensing controller (*Wii Remote* or *Wiimote*) based on infrared light detection and 3D accelerometers (see Appendix A.2). Shortly after the release of the gaming console, several open source APIs were provided for connecting the Wii Remote to a computer via Bluetooth. In this way, the acceleration data detected by the Wii Remote can be retrieved and used as system input. As the Wii Remote device is commercially available at reasonable costs, it can be applied as an interaction device for the development and testing of accelerometer-based gesture interaction techniques instead of a specialized glove device.

Recently, location and movement sensors, like accelerometers, gyroscopes and digital compasses, are increasingly incorporated in modern smartphones. These devices represent a further platform, to which the acceleration-based interaction approaches presented in this section can be adapted (see Appendix A.3).

wiigee

In order to be able to define and recognize certain movement sequences of the Wiimote as gestures, we apply the open source gesture recognition library *wiigee*[11] ([Schlömer et al., 2008]). This library has been specifically developed for capturing and analyzing the data delivered by the Wii Remote controller. It offers methods for defining (training) arbitrary movement patterns as gestures in an initialization phase, which are later used to classify user gestures with a certain probability according to these predefined patterns.

In the *training phase* of wiigee, each gesture which is supposed to be identified in the later process has to be defined by the user. For this purpose, the movement sequence representing the gesture has to be performed repeatedly by the user holding the Wiimote. In order to achieve a training set allowing feasible recognition results, the wiigee developers recommend to repeat each gesture for at least five to ten times (or up to fifteen times as a matter of experience). During each performance of the gesture, a *TrainButton* must be pressed. After a sufficient number of training iterations, pressing a *CloseGestureButton* ends the training process for this particular gesture. Several gestures can be trained by repeating the whole procedure for each gesture.

After all desired gestures have been recorded, the wiigee application builds up an internal *gesture model*. In the subsequent *recognition phase*, the user can reproduce the previously trained gestures while pressing a *RecognitionButton*. The wiigee system tries to identify each movement sequence as a specific gesture according to the previously trained gesture model and fires a *GestureEvent* containing a list of the gesture patterns which best match the detected movement sequence along with their calculated recognition probabilities. Out of this list, generally, the gesture with the highest probability is selected as input.

Wiigee is applied in the present interaction module for both defining as well as recognizing user gestures. As in its original version, wiigee does not provide any method for storing the user's training data, in the course of the present work, it has been extended with an xml-based storage model. In this way, each user can generate his own gesture set, save it in an individual gesture profile and reload it every time when he uses the Wiimote interaction module. Each gesture profile is assigned a specific user-defined gesture, with which it can be loaded.

Interaction method

While the Wiimote device provides numerous input possibilities, in this work, its gesture

[11] http://www.wiigee.org/

input capability by means of the 3D accelerometer is the most interesting one. The Wiimote constantly sends its acceleration values via Bluetooth to a connected computing system. If the Wiimote is being moved, the measured values reflect the acceleration of the movement combined with the gravitational acceleration of the Earth. This means that when the controller is held still, the accelerometer is only affected by the Earth's gravity. In this case, the acceleration values indicate the rotation angles of the Wiimote around its X-, Y- and Z-axes.

When using the acceleration-sensing capabilities of the Wii Remote, we distinguish between two types of gestures:

- *continuous gestures*: gestures that have an immediate and continuous effect while they are being performed, e.g. rotation about any of the Wiimote's axes.

- *discrete gestures*: the moving path of the Wiimote characterizes a special three-dimensional form, e.g. a letter, a digit, a rectangle; performing a discrete gesture can trigger an event.

In continuous gesture mode, the acceleration output of the Wii Remote is immediately mapped to system output, e.g. resulting in a movement of a Virtual Display or of the Dynamic Peephole. In our application, continuous gestures consist in rotating the Wii Remote around one of its three axes (see Figure A.3 [right]). In the following, we denote a rotation around the Y-axis as a left/right rotation and a rotation around the X-axis as an up/down rotation. Continuous gestures are especially suitable for triggering continuous functions like movement, in-/decreasing of numbers or rotation. In this way, the movement of the Wiimote can be continuously mapped to the movement or modification of the interaction target ([Spassova and Guo, 2009]).

In the Wiimote interaction module, we use continuous gestures for the following functions:

- *Movement of Virtual Displays*: left/right rotation results in corresponding left/right movement of the target Virtual Display; up/down rotation results in an up/down movement of the Virtual Display.

- *Rotation of Virtual Displays*: left/right rotation is mapped to a corresponding rotation of the Virtual Display around its normal vector.

- *Movement of the Ubiquitous Cursor/Dynamic Peephole*: analogous to the mapping for Virtual Display movement but with the Ubiquitous Cursor as interaction target.

- *Navigation through projected menu items*: in this module, we have implemented several pie and list menus (which are described later in this section); left/right rotation is used to navigate through the pie menus and up/down rotation for the list menus.

- *In-/decrease of numbers*: used for the numerical adjustment of the size and rotation angle of a Virtual Display through the projected menus; up/down rotation in-/decreases a number and left/right rotation leads to a switch to the next digit.

Discrete gestures, on the other hand, are used to trigger command functions like the creation of a display or the opening of a projected menu. They consist in symbolic "air drawings" and are recognized using the previously described wiigee toolkit. In contrast to continuous gestures, the function triggered by a discrete gesture is not invoked until the gesture has been completely finished.

Discrete gestures are used to trigger the following command functions:

- *Creation of a Virtual Display*: After a triggering gesture is performed, a Virtual Display is created at the location indicated by the Ubiquitous Cursor.
- *Switch of the interaction target*: In the context of the present interaction module, an interaction target can be either the Ubiquitous Cursor or one of the Virtual Displays; a Virtual Display is automatically selected as interaction target (marked by a red border) as soon as the Ubiquitous Cursor is placed on it; in this case, performing a specific gesture switches the interaction focus back to the Ubiquitous Cursor in order to be able to move the Dynamic Peephole to a new location without moving the previously selected Virtual Display at the same time.
- *Opening of projected menus*: In the present module, we have implemented three different projected menus (display menu, display list menu and gesture profile menu, described later in this section), which can be opened by different gestures.

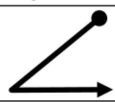
Function	Gesture
Creation of Virtual Display	
Switch of the interaction target	
Opening of display menu	
Opening of display list menu	
Opening of user profile menu	

Table 6.1: *Default discrete gesture patterns: onset marked by point, arrow indicating the direction and end of the movement sequence*

Altogether, five different discrete gestures are needed for triggering these command functions. In pursuit of more reliable recognition results, the default discrete gestures have been chosen to be as distinct as possible. Table 6.1 shows an overview of the default gestures, which were not only drafted to be distinct in respect to each other, but their shapes also mostly correlate with the meanings of their functions. This design was chosen in order to provide a set of gestures which are easy to memorize for users. Alternatively, instead of using the default gestures, each user is also given the opportunity to define and train his own set of discrete gestures, which are stored in a personal gesture model.

The display creation gesture has the form of a rectangle, which resembles the outlines of a Virtual Display. The gesture used to switch the interaction target is defined as a circle which is performed while holding the Wiimote upright, pointing to the ceiling. This gesture can be well distinguished from the other gestures, as due to the particular orientation of the interaction device, it results in notably different acceleration values. In order to open the display menu for editing a Virtual Display, the gesture "M" (for "menu") is performed. The gesture for opening the display list menu looks like a squeezed "L" (for "list"). It has been defined in this way, because it can be better recognized by the wiigee toolkit than a real "L"-shaped movement. The last gesture, "P" (for "profile"), is used to open the gesture profile menu, which enables the loading of personal gesture sets. The arrows in each figure imply the direction of performing the gestures, and the point denotes the starting point of the movement.

Interaction options

Although the Wiimote device applied with this interaction module offers a number of buttons for input, our aim is to provide an interaction method which mainly uses the motion data delivered by the accelerometer with as few auxiliary buttons as possible. In this way, the interaction method can be easily adapted for use with other accelerometer-based devices, like the glove and wristband devices described in Appendix A.1, which offer only a limited number of buttons. However, with accelerometer-based gesture recognition, it is difficult to detect the beginning and end of a certain gesture, so we decided to make use of at least a minimum button input for indicating the on- and offsets of movements which are to be recognized as gestures.

To be able to experiment with different interaction methods, we have implemented two gesture interaction options, which differ in the usage of buttons. The first option involves the usage of two different buttons during gesture performance. One of the buttons (button A) is pressed to indicate a discrete gesture. This button is to be held down while a discrete gesture is being performed. This means that pressing A indicates the beginning of a discrete gesture and releasing A marks the end of this gesture. In a similar way, button B is used to specify the performance of a continuous gesture.

The alternative interaction option applies only one single auxiliary button (button A), which is used in various ways. On the one hand, similar to the previous approach, button A is pressed and held down to indicate when a gesture is being performed (for both discrete and continuous gestures). On the other hand, a short click on the button switches the current gesture mode between discrete and continuous. This means that depending on the currently selected gesture mode, a movement of the interaction device is interpreted either as a discrete or as a continuous gesture. In order to make the user aware of the current gesture mode, we provide visual feedback in form of a projected Ubiquitous Cursor with additional gesture mode symbols (see Figure 6.27), which will be discussed later in this section.

Finally, we have also implemented an interaction option which is only button-based and does not make use of any accelerometer-based gestures. In this case, the Wii Remote interaction is very similar to the use of a common remote control. This interaction option is used as a baseline for a comparative user study, as described in Appendix B.

Projected menus

Not every single function needed for interaction with the Display Continuum can be assigned a specific gesture, as this would lead to a large set of interaction gestures, which would be difficult to memorize. To keep the gesture set small, we provide the following projected menus as auxiliary means for triggering certain functions:

- *Gesture profile menu*: This menu leads the user step by step through the process of setting up an individual gesture profile. It also offers the possibility to load a previously created gesture profile.

- *Display list menu*: This is a list menu which helps navigating through the Display Continuum. It shows a list of all currently available Virtual Displays. By selecting one of the entries, the Dynamic Peephole is moved to the position of the corresponding display. Additionally, this menu allows the selection of all displays or of all currently visible displays, so that this selection can be manipulated (e.g. moved) simultaneously. The display list menu can be opened either on a Virtual Display or on an empty surface on the DC. If it is opened on a Virtual Display, the menu contains an additional entry *Define content*, which triggers a movement of the Ubiquitous Cursor to a stationary screen for content selection (see Figure 6.25).

- *Display menu*: In contrast to the previous two menu types, the display menu is always associated with a certain Virtual Display. It is used to adjust some of the display parameters (size and orientation) and to delete a display if needed (see Figure 6.26).

***Figure 6.25**: Display list menu*

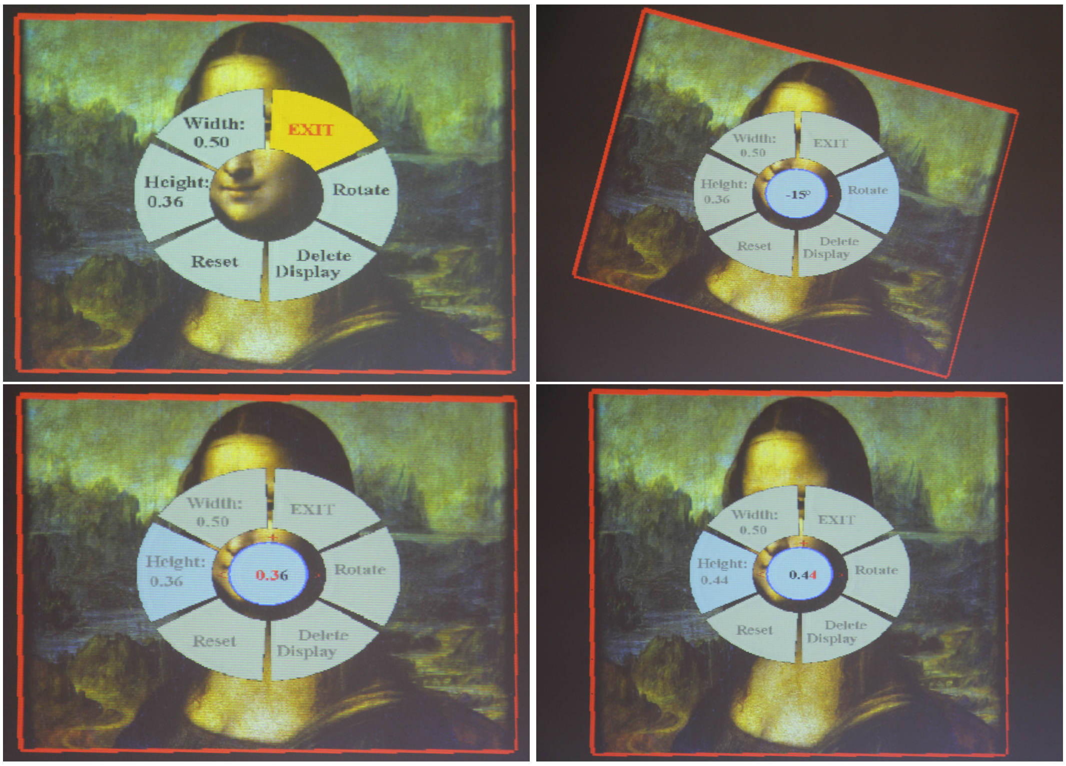


***Figure 6.26**: Display menu with submenus: main menu [upper left], rotation submenu [upper right], submenu for hight adjustment [lower left] and image with modified height [lower right]*

Ubiquitous Cursor and interaction focus

According to the concept introduced in Section 5.4, we have developed the Ubiquitous Cursor as an equivalent to the mouse cursor on a computer desktop. This Ubiquitous Cursor is realized in the form of an arrow resembling the appearance of the standard desktop cursor. It is projected in the environment indicating the position currently aimed at for interaction (see Figure 6.27 (a)). The Ubiquitous Cursor is always displayed in the center of the Dynamic Peephole. In this way, it can be moved along the surfaces of the Display Continuum together with the Peephole. Using this pointer metaphor, the user is aware of the current location of the Dynamic Peephole, which marks the interaction focus. This projected feedback is supposed to support the user in selecting a desired location, e.g. for creating a new Virtual Display.

In order to indicate the current interaction mode, the Ubiquitous Cursor can be enhanced either with a cross symbol for the continuous interaction mode (see Figure 6.27 (b)) or with a tilde symbol for the discrete interaction mode (see Figure 6.27 (c)).

As soon as the Ubiquitous Cursor is moved on a Virtual Display, this display is automatically selected for interaction. In order to indicate the switch of the interaction focus, the Virtual Display is marked by a red border and the Ubiquitous Cursor is hidden. Now the user has the opportunity to interact with the selected display. For switching the interaction

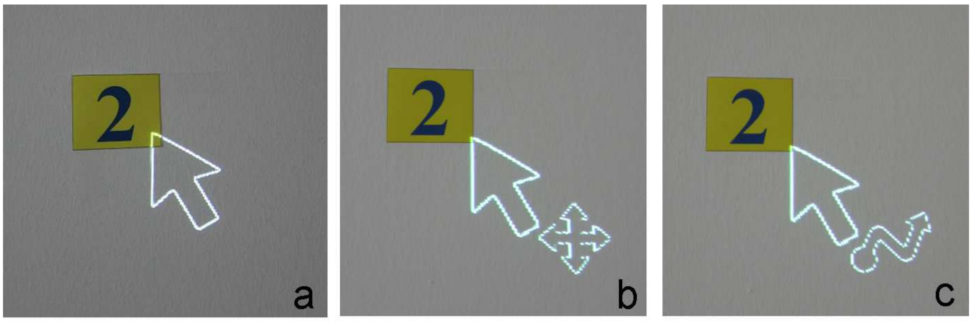


***Figure 6.27**: Ubiquitous Cursor projected on a wall indicating (a) neutral state, (b) continuous gesture mode and (c) discrete gesture mode*

focus back to the Dynamic Peephole, the user can perform a special discrete gesture (see Table 6.1). After that, the Ubiquitous Cursor reappears and can be moved along the Display Continuum using the previously described continuous gestures.

Integration of stationary screens into the Display Continuum

In order to be able to define the content for a projected display (image, video or live stream), interaction with stationary screens must be possible. This is realized by automatically switching the interaction focus from the projected Display Continuum to a stationary screen as soon as the Ubiquitous Cursor reaches its border. Thus, it is possible to create Virtual Displays on a stationary screen and drag them to the Display Continuum, where they appear as projected displays (see Figure 6.28).

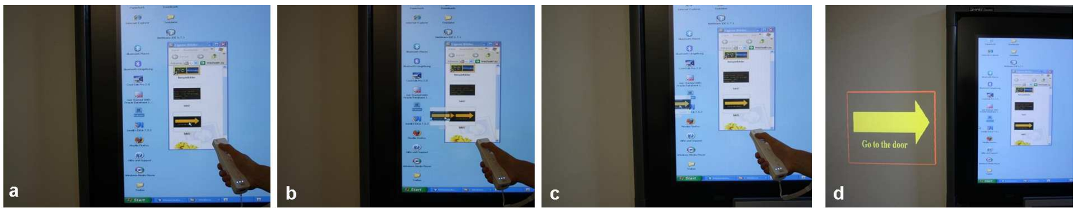


***Figure 6.28**: Virtual Display creation by dragging an image from a stationary screen to a projected Display Continuum*

Interaction context

Because of gesture similarities, the larger a gesture library is, the more difficult it is to obtain reliable recognition results. In order to achieve an easy to use interaction module, the set of used gestures has to be kept as limited as possible. This results in function overloading, when the same gesture can refer to different system commands. In this case, interaction context plays an important role as it helps in resolving the ambiguities. Especially in the case of discrete gesture interaction, the context is taken into account in order to limit the number of possible interpretations of the received acceleration data and thus reduce the risk of recognition errors of the wiigee toolkit.

As described above, the gesture interaction employs five different discrete gestures to trigger five command functions. Additionally, identifying gestures can be specified, which are used as a kind of password to load personal gesture profiles. However, not all gestures make sense in each system state. Therefore, in certain cases, the gesture library used by the wiigee recognition can be restricted to a limited set of potentially possible gestures according to the current interaction context. The restriction of the gesture set is supposed to improve the gesture recognition process.

Here, context describes the state of the system in which a gesture is being performed. We have identified four context states which are relevant for gesture recognition. Each of them is unique and in each one only certain gestures, which fit the particular context, are included in the current gesture library.

In the following, the four context states are listed with their descriptions and the possible gestures building the respective gesture library.

- *VD context*: The interaction focus lies on a Virtual Display.
  Possible discrete gestures:
  - Switching of the interaction target
  - Opening of a display menu
  - Opening of a display list menu
- *DC context*: The interaction focus lies on the Ubiquitous Cursor, which is placed at a location on the Display Continuum where no Virtual Display is currently present.
  Possible discrete gestures:
  - Creation of a Virtual Display
  - Opening of a display menu list
  - Opening of a gesture profile menu
- *display menu context*: A display menu or a display list menu is currently open.
  Possible discrete gestures:
  - No discrete gestures are expected.
- *gesture profile context*: The gesture profile menu is currently open.
  Possible discrete gestures:
  - Identifying gestures for loading gesture profiles.

6.2.3 Real-world Interfaces for Implicit Interaction

The previously described DUVD interfaces have been designed for explicit user interaction with the Display Continuum. This means that the user intentionally issues a command to the system using an appropriate interface. This type of human-computer interaction implies that the user knows exactly what he wants the system to do and how to issue this wish to the system. Implicit HCI has originally been developed for working with desktop computers, and it assumes that the user has at least a certain expertise regarding the interface commands. With the development of Graphical User Interfaces (GUIs) (see Section 2.4.1), HCI has become more intuitive and applicable even for non-experts as these interfaces require less expert knowledge by mimicking the real world and limiting the interaction space to a very restricted set of standardized commands, which are interpreted according to the current system state.

With the increasing dissemination and embedding of computation into everyday life, there is the need to push the naturalness of HCI to a further level, where explicit command issues should become obsolete. This approach is referred to as *implicit human-computer interaction* and was defined in [Schmidt, 2000] as "an action performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input". In oder to enable such implicit user input, a HCI system is supposed to exploit knowledge about the user activities and his environment, which has to be observed using appropriate sensors.

Using implicit HCI, upcoming Ubiquitous Computing systems can react proactively to the user's needs. Simple examples of implicit interaction in everyday life are automatic light switches, which turn the lights in a house on and off depending on the presence or absence of a person. In a supermarket scenario, implicit interaction can take place when the customer handles products in a shelf. The Digital Sommelier described in [Schmitz et al., 2008], for example, displays relevant product information as soon as the customer takes a wine bottle out of the shelf. The information can also be adapted to the way the customer is holding the wine bottle, assuming that he is interested in more details when turning the product and looking at its back side.

In the course of this work, we have developed several DUVD applications involving implicit user interaction, which are presented in the next section. We denote the involved DUVDs as *system-driven*, as they are created and controlled by the respective system according to the observed user behavior. In contrast, we also present examples of *user-driven* DUVD applications, which are controlled explicitly by the user.

6.3 System-driven DUVDs

The DUVD system has been applied and tested in various applications. In the following, we illustrate some examples of applications using system-initiated Virtual Displays. This means that the locations, the movement and the displayed content of the Virtual Displays are controlled by the underlying system depending on the user and environmental context. Although, in these cases, the user does not consciously control the system output, it is in fact influenced by the user behavior, which is observed by the DUVD system and results in implicit interaction as described in Section 6.2.3.

6.3.1 Product Associated Displays

In this section, we introduce the concept of *Product Associated Displays* (PADs, [Spassova et al., 2005], [Wahlster et al., 2010]) as a way of providing visual feedback to users implicitly interacting with physical objects in an Instrumented Environment. PADs are projected Virtual Displays created at locations that can be intuitively associated with the objects they show information about. The concept is illustrated in a shopping scenario.

This application is mainly concerned with the following question: How can a customer be supported in his shopping process by the deployment of adaptive Virtual Displays?

Our application scenario, an instrumented shop, consists of the following components among others: instrumented shelves, some public screens and a PDA for each user (see also MSA in Section 6.2.2.1). The shelves are fitted with RFID antennas and allow for sensing implicit user interactions with RFID-labeled objects, such as picking up a product or putting it back into the shelf. The walls, the floor and especially the shelf surfaces of the instrumented shopping environment build a Display Continuum and thus provide physical spaces that can be used for information visualization through Virtual Displays. Since the human mind locates information and concepts spatially, such an environment allows for a mapping between physical space and abstract information, where the physical space is enriched by digital information, and the digital information can be made more accessible and understandable by mapping it to physical space.

In the original version of the shopping assistance system MSA ([Wasinger et al., 2005]), visual information about the products taken out of the shelf is displayed on a public screen beside the shelf and on the user's PDA. In both cases, the user's attention has to be directed away from the object he is interacting with to the location of the displayed content. In this context, we distinguish between stationary public screens, which are bound to fixed locations, and projected Virtual Displays, which can be created on arbitrary surfaces within the Display Continuum. Product Associated Displays represent a special case of projected Virtual Displays and offer a more intuitive way to provide visual feedback to the user interacting with products from the shelf than stationary public screens.

If the customer takes the last product out of a shelf, we exploit the space left empty in the shelf to project relevant information about this product. Although in the process of taking an object out, the user focuses his attention on the product itself, the former location of the object is still in his peripheral view. So if a change like the appearance of a new projected Virtual Display occurs in this area, it is very likely to be recognized by the customer and thus to draw his attention to the projected information. Following this approach, the user's attention does not have to be directed to a new display location, as the relationship between the physical object and the displayed information arises automatically. In fact, a spatial mapping between a physical space and the digital information is established, and supports the user's ability to process and interpret information about where objects are in space: *visuospatial perception*. This represents the relation between the physical space around the user and what the user sees. As human-computer interaction moves from the computer screen to the environment, this aspect becomes fundamental and can be exploited by mapping content and relational information to the space around a person.

The technical implementation of the PAD approach is realized in the following way:

- When a product is taken out of the instrumented shelf, it is recognized by the shelf's RFID reader and a corresponding event is generated and sent to the DUVD system.

- If the system detects a PRODUCT_DISAPPEARED event, a Product Associated Display is visualized at the corresponding location, showing the name and a picture of the removed product (see Figure 6.29 [left]) indicating that the system has recognized the user's implicit interaction.

- Now the user can explicitly ask for information about the chosen product by applying the different MSA modalities, which have already been mentioned in Section 6.2.2.1.2. The visual feedback is displayed on a projected PAD in the shelf (see Figure 6.29 [right]). In this example the user has asked for the price of the chosen camera. With the displayed text "price: 499€", the system not only delivers the required information but also implicitly (by the word "price") indicates that the request has been recognized correctly.
- If the product is placed back in shelf, the system generates a PRODUCT_APPEARED event, which denotes the end of the interaction sequence. This causes the PAD to be switched to an inactive state, so that the projected display disappears. Otherwise, in case the specific product item has not reappeared in the instrumented shelf after a certain period of time, the PAD returns to its initial state, i.e. any specific information which might have been shown on the PAD is removed, as it can be assumed that the customer has decided to purchase this product and proceeded with his shopping tour.

Figure 6.29: *Example of user interaction via PADs: initial image of the product indicates the recognized interaction [left]; visual feedback to the user's request for the price of the product [right]*

All PADs used in this application are initialized at startup. Their locations are calibrated according to the positions of the corresponding products in the shelf. Initially, all PADs are in an inactive state, so that they are invisible to the customer.

The PAD concept provides the opportunity for several users to interact with a product that is currently in the possession of another user. In this case the projected PAD plays the role of a placeholder and enables interaction with a product even if the physical object is actually not available. Alternatively, the PADs could also show hints on how the empty shelf has to be refilled, which are intended for the supermarket staff. In this case, the PADs can be configured to be implicitly controlled by the staff's movement through the marker, so that the refilling hints are shown when a staff member is currently passing by.

In [Sukaviriya et al., 2003], the authors describe and evaluate a steerable interface system using the Everywhere Displays projector, allowing interaction with projected interfaces on arbitrary surfaces in a retail store scenario (see also Section 4.4.2). Their paper presents three types of user interaction, one of which consists in projecting information about products on a surface right beside the products' bins, which is similar to the PAD scenario. In contrast to our approach, however, the interaction there is based on the user's position and is sensed using computer vision. In their evaluation, the authors elaborate on the problem of many test subjects not being able to associate the displayed information with the products because of the spatial distance between them. Often subjects were not even aware of any displayed feedback because their attention was drawn away by other activities. These results particularly encouraged us in our belief in the effectiveness of Product Associated Displays.

6.3.2 Micro Navigation and User-adapted Advertising

A further application which deploys the DUVD system in a retail context is the so-called *micro navigation* service. It enables the display of visual navigation hints in form of projected Virtual Displays in order to guide the customer's attention to products he is searching for.

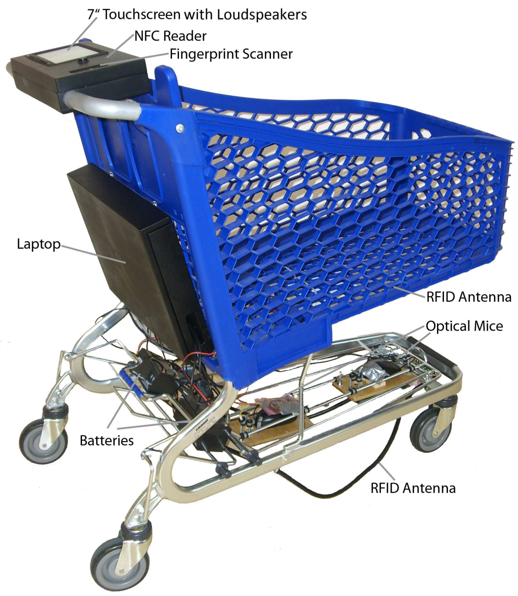


Figure 6.30: *IRL SmartCart hardware and instrumentation*

The micro navigation service is connected to an instrumented shopping cart – the so-called *IRL SmartCart* ([Kahl et al., 2011], [Kahl et al., 2009]) –, which acts as an input and output interface for assisting the customer during his shopping tour. Figure 6.30 shows the hardware and instrumentation of the SmartCart. It is equipped with a touchscreen integrated in its handle, and it uses RFID technology for recognizing RFID-tagged products placed in its basket. The customer can identify himself at the SmartCart using a built-in fingerprint sensor or an NFC card. After the customer is identified, the SmartCart's system loads the corresponding user profile, which contains the customer's current shopping list among others.

Furthermore, the instrumented shopping cart is capable of recognizing its own location in the shopping environment, which in combination with a 3D model of the supermarket is used for the realization of a navigation service. The self-localization of the SmartCart is realized by means of a second RFID antenna mounted on the lower part of the cart, which recognizes RFID tags placed in a grid under the flooring of the shopping environment. The current location of the cart is calculated using the Always Best Positioned algorithm ([Schwartz et al., 2005], [Brandherm and Schwartz, 2005]).

The ability of the SmartCart to locate itself in the environment together with the knowledge about product placements in the supermarket and the customer's user profile enable the generation of user-adaptive advertisement and product recommendations ([Kahl et al., 2010]). Aside from the opportunity to show these hints on the cart's built-in display or on stationary displays placed at strategic locations in the supermarket, there is also the opportunity to use the steerable projection system in order to show the navigational information on projected Virtual Displays in the environment. As soon as the customer approaches the location of a product or a product group, which he is searching for, a projected arrow is displayed at an appropriate location at the corresponding shelf or on the floor, giving a visual hint to the position of the searched product (see Figure 6.31 [left]).

The vicinity to a specific product or shelf is defined by spatial zones in the 3D model. The customer's location in the supermarket is detected indirectly through the tracking of his SmartCart. As soon as the customer (or rather the customer's shopping cart) enters a specific zone of interest, the SmartCart sends a corresponding event to the DUVD system, which activates the appropriate Virtual Display and shows a navigation hint according to the currently searched product. When the customer finally reaches the product of interest or otherwise leaves the zone of interest, the Virtual Display is deactivated and disappears.

In combination with the navigation service offered by the SmartCart and the user profile information, the DUVD system also enables the creation of location-based user-adaptive advertisement. If the customer has, for example, the entry "muesli" on his electronic shopping list and he is approaching the shelf with the cereals, he is proactively informed about a new sort of muesli which, according to his preferences, he might be interested in. Figure 6.31 [right] shows such a user-adaptive projected advertisement displayed at the top part of a shelf.

6.3.3 Virtual Room Inhabitant

Intelligent Environments often provide a variety of different devices and services which are embedded in the physical space (see Section 2.2). On the one hand, these setups have the advantage of not overstraining the users with too much obvious instrumentation. On the other hand, especially novice users are often unaware of all devices and services in an Instrumented Environment and the abilities they offer.

To counteract this drawback, the deployment of virtual characters has been proposed as

Figure 6.31: *Micro navigation hint projected on the floor in front of the SmartCart [left] and in combination with customer-adapted advertising on top of a product shelf [right]*

a new way to improve the usability of complex hardware setups in Instrumented Environments. By introducing a virtual character, we aim at facilitating intuitive interaction with the Instrumented Environment. The Virtual Room Inhabitant (VRI, [Kruppa et al., 2005], [Kruppa, 2006]) is a life-like virtual character, which is capable of "freely moving" along the walls of the room. In this way, it can offer situated assistance to users within the environment. The concept of a virtual character "living" within the Instrumented Environment and thus playing the role of an assistant, allows both novice and advanced users to efficiently interact with the different devices integrated within the environment. The character is capable of welcoming a first time visitor and its main purpose is to explain the setup of the environment and to help users while interacting with it. A further scenario in which the VRI has been applied is a museum, where it plays the role of a visitor guide ([Stock and Zancanaro, 2007]).

The VRI implementation is a combination of three components: a character engine, a spatial audio system (SAFIR, [Schmitz and Butz, 2006]) and projected Virtual Displays delivered by the DUVD system. This enables the virtual character to appear and move along the surfaces of the Display Continuum, while its visual appearance is spatially synchronized with its audio output through the character engine. In this way, the character's voice and sounds appear to originate from the location where it is currently being visualized, even while it is moving.

The visual appearance of the VRI is realized as a live video stream on a projected Virtual Display. Thus the character can be animated in real time by the character engine. As the

***Figure 6.32**: Virtual Room Inhabitant (Cyberella) beside a stationary screen*

Virtual Display is borderless and the character animations have a transparent (i.e. black) background, the character does not appear to be placed in a display frame but it is smoothly embedded in the environment (see Figure 6.32).

The first version of our VRI, who was named Cyberella, was integrated within a shopping and navigation demonstrator. In this scenario, users were given a PDA and asked to perform a combined indoor/outdoor navigation task. The idea was to lead the users to an airport ground and upon entering the airport facilities to guide them towards certain duty free shops until their departure time approaches. In our demonstration setup, these shops are represented by different rooms, one of them being the instrumented room (see Section 2.2). There the VRI plays the role of a host, welcoming visitors and introducing the components of the instrumented room to them.

In this scenario, the behavior and movement of the VRI is controlled by a predefined script. As soon as a user enters the room (detected by an infrared beacon), the character appears on the wall nearby the entrance to welcome the user. After that, the character moves from one device in the Instrumented Environment to another following a predefined path. It stops at each device and shortly explains its functionalities. The demo involves an instrumented shelf recognizing RFID-labeled products, an instrumented shopping cart similar to the IRL SmartCart (see Section 6.3.2) and a wall-mounted screen, on which product information and recommendations can be presented. While moving from one position to another the character appears as a rotating ball. This form of visualization has been chosen as the applied Cyberella character does not provide any walking animation sequence.

To conclude the shopping demo, the VRI is finally triggered to notify the user about the immediate boarding of his flight. For this purpose, the character appears alongside the exit of the room, points to it and instructs the user to proceed to the boarding gate.

Path generation algorithm for VRI

In order to be able to make the VRI application more flexible, we have developed an algorithm for ad hoc computing of appropriate pathways given the start and end point of a movement. The path generation takes into account the geometry of the Display Continuum (especially discontinuities and obstacles), the size of the Virtual Display on which the character is presented and some other particularities concerning the movement of a virtual character, which are explained later in this section.

The 3D model of the Instrumented Environment builds the basis for the path generation. It provides information about all relevant projection surfaces and obstacles present in the environment. When moving a Virtual Display from A to B on the Display Continuum, it has to be taken into account that there might be obstacles on the way, which should be avoided in order to maintain the undisturbed visibility of the projected character during the entire movement. This means that obstacles have to be bypassed if they lie on the direct path between A and B. Moreover, it must be guaranteed that the computed path is wide enough, i.e., it must offer enough space for the visualization of the whole Virtual Display showing the character. The reference point for a Virtual Display movement is the midpoint of the display, and depending on the display size, the computed path must lead far enough from the surrounding obstacles in order to enable an undisturbed visualization.

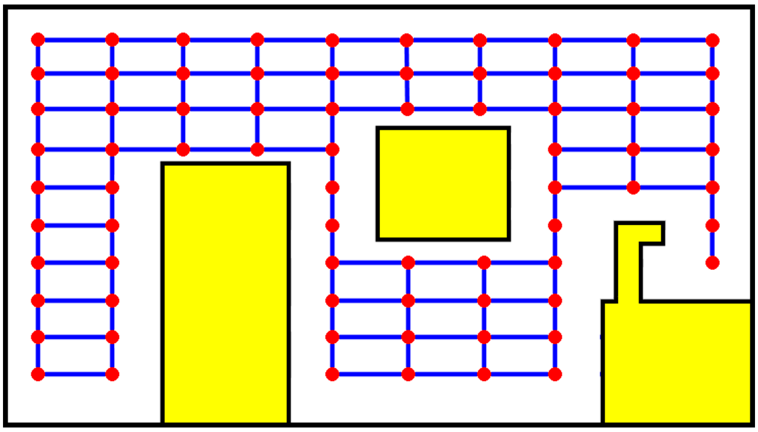


Figure 6.33: *Display Continuum with obstacles and path network: red dots mark PathNodes and blue lines represent PathEdges*

Taking these conditions into account, we have developed and implemented a path generation algorithm which operates on a path network (class *PathNetwork*). This network is a mesh of nodes (*PathNode*) distributed in a regular grid over the surfaces of the Display Continuum which are suitable for projection. These PathNodes represent locations on the Display Continuum at which the VRI can rest and interact with the user. Neighboring PathNodes are connected by edges (*PathEdge*), which represent possible path segments for a Virtual Display movement. An example of a wall surface with obstacles and a path network is illustrated in Figure 6.33.

Aside from their location in 3D, PathNode objects contain also information about the surface they lie on, and they can be marked as points of interest which lie in the vicinity of a special device. In this case, these nodes mark potential locations at which the VRI can be placed in order to explain the functionalities of the corresponding device. Every PathEdge which connects two PathNodes is assigned a specific weight value. These weights specify the quality of the corresponding edges and are taken into account by the path-finding algorithm in order to generate optimal movement paths for the VRI display.

The weight values assigned to the PathEdges – and thus their quality – depend on the position and orientation of the corresponding path segments on the Display Continuum. In our particular case of a moving virtual character, horizontal path segments are preferred over vertical ones, as it appears more natural when the character walks along a horizontal line than when it hovers vertically. This is especially the case, when the character performs a walking sequence while it is moving. Therefore, in our PathNetwork, horizontal edges are assigned smaller weights than vertical ones as a small weight denotes a high quality path.

Edges which connect adjacent surfaces are also not preferred, as in most cases, they pass across discontinuities on the Display Continuum. Therefore, such PathEdges are assigned high weights. In order to have a PathNetwork in which every node can be reached from any other node, in some cases, it is inevitable to include edges which pass across obstacles. These path sequences should be used only if there is no other possibility to reach the corresponding node, therefore these edges are assigned the highest weights.

The PathNetwork is generated automatically given the corresponding 3D model of the Display Continuum. The network generation algorithm proceeds in several steps. First of all, a local mesh of nodes and edges is generated for each individual surface of the Display Continuum. Subsequently, the edges which connect adjacent surfaces are added to the PathNetwork. When generating the local mesh, the algorithm starts at the top left corner of the surface and computes the nearest possible position for a PathNode depending on the size of the VRI display. After that, the system proceeds in incremental steps to the right and downwards until the whole surface has been explored. If during this search, the algorithm hits an obstacle, it first tries to find a bypass around it. If this is not possible, then an edge with a high weight is added which passes the obstacle.

After the PathNetwork has been generated, it can be used to compute the "shortest" (i.e. the optimal) path between two given nodes. The *PathFinder* implements an adaptation of the well-known Dijkstra algorithm, which is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge weights (i.e. edge costs), producing a shortest path tree ([Dijkstra, 1959]). For a given source node in the PathNetwork, the algorithm finds the path with lowest cost between that node and every other node. As in our case, we only need to find one optimal path from a single source node to a single destination node, the algorithm can be stopped once an optimal path to the destination vertex has been determined.

VRI behavior

Finding an appropriate movement path through the environment is not the only task which has to be solved when developing an interactively moving character. If our character were a simple object displayed in the environment, it would be sufficient to compute an optimal path as described above and move the Virtual Display showing the character on this path from A to B. In this case, the image of the simple object would always remain the same.

In our case of a life-like character, however, it makes a difference if the character is walking in a horizontal direction (to the right or left) or in a vertical direction (up or down). While in the former case, the character can perform a walking sequence to one of both directions (see Figure 6.34, top rightmost), it would appear unnatural if the same walking animation is displayed while the character is moving upwards or downwards. In the latter case, a more natural appearance can be achieved if the character stays in a neutral position as if it uses an elevator to move in the vertical direction, or we can use a morphing ball animation which has already been deployed for the character movement in our first VRI version (see Figure 6.34, bottom).

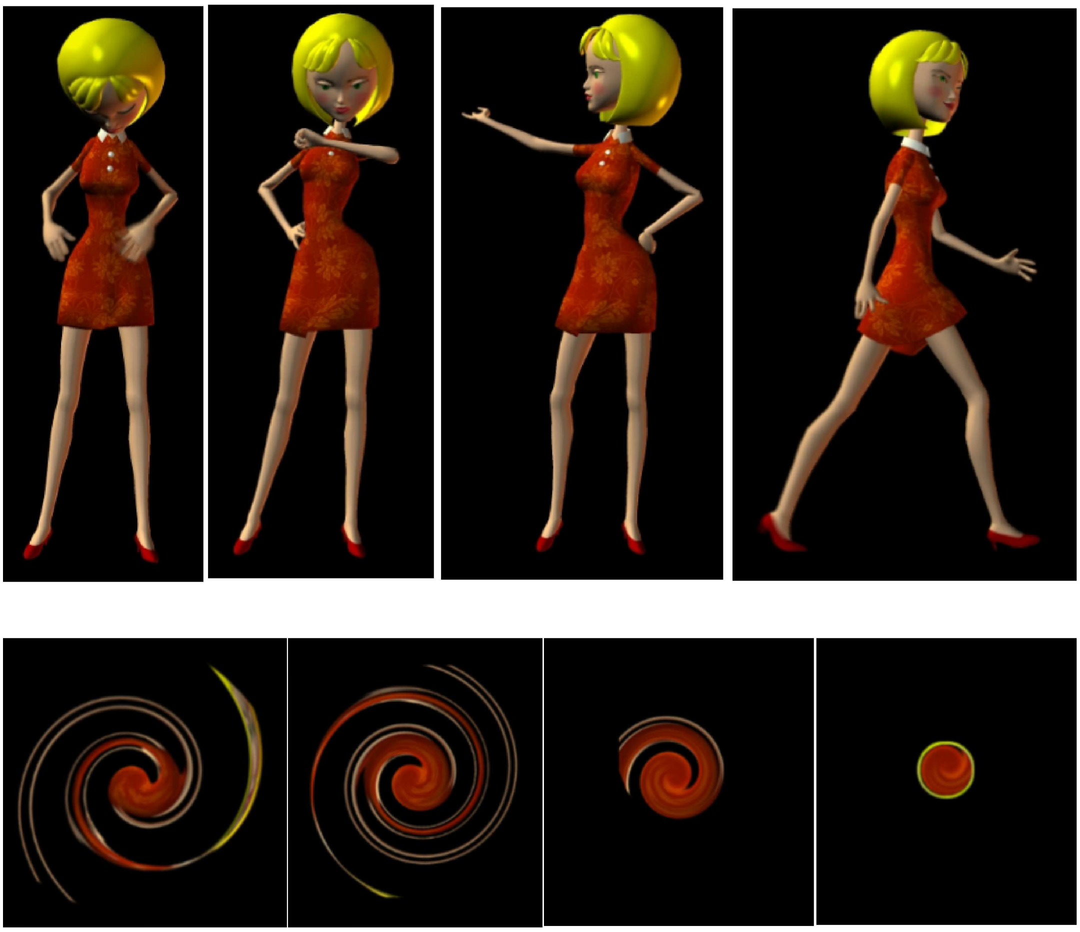


Figure 6.34: *Gesture examples of the VRI character Minnie: idle gestures "skirt cleaning" and "looking at the watch"; "showing to the right" and "walking to the left" [top, from left to right]; character morphing into a ball [bottom]*

Furthermore, in order to make the VRI more interactive, we have implemented a behavior module, which adapts the character's performance to the user's position and orientation in the room, which is detected using the Always Best Positioned algorithm ([Schwartz et al., 2005], [Brandherm and Schwartz, 2005]) running on a handheld device. The behavior of the VRI is implemented in the form of a deterministic automaton with the following five character states:

- *wait*: The character is displayed at one location and performs idle gestures while waiting for user input.
- *talk*: The character talks to the user, e.g., it explains the functionalities of a nearby device.
- *follow*: The character moves along the Display Continuum (either walking or in a ball shape) following the user.
- *call*: The character tries to attract the user's attention, e.g. when it detects that the user is not noticing or following it.

During runtime, the VRI behavior module constantly monitors the user's position and orientation, which are computed by the *PositionFinder* module. Any change in the user's state is reported by the PositionFinder in the form of events. We distinguish between the four following user event types, which can trigger transitions between the character states of the VRI and thus implicitly influence its behavior:

- *present*: A user is detected in the vicinity of the VRI.
- *watching*: The user is looking in the direction of the VRI.
- *looking away*: The user is currently not looking in the direction of the VRI.
- *moving away*: The user is moving away from the current position of the VRI.

The transition graph describing the character reactions to the user's behavior is presented in Figure 6.35.

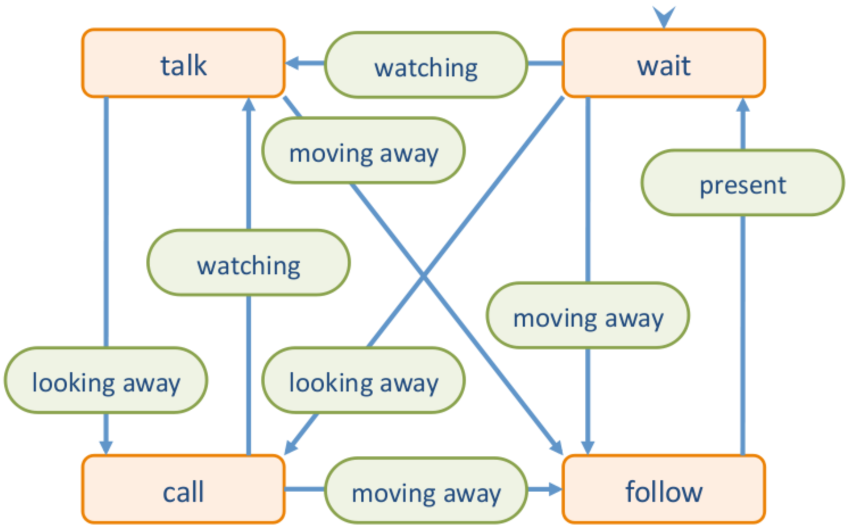


Figure 6.35: *Transition graph defining the VRI behavior depending on implicit user interaction*

The content which is presented in the talking state of the VRI depends on the current position of the character, which in turn is adapted to the detected user location. In this way, when the user is approaching some points of interest (e.g. devices in the environment), the character is placed at a position nearby the corresponding device, and the VRI's information script associated with this device is presented, i.e., the character tells the user about the functionalities of the device in front of him.

According to the defined transition graph, an example scenario could take place as follows: The VRI is waiting in an idle mode at the entrance of the Instrumented Environment until a user enters the room. If the user stops and looks at the character, he is being welcomed in the Instrumented Environment. Otherwise, if the user is passing by and approaching a certain device, e.g. the instrumented shopping cart, the VRI is following him by moving over the walls of the room, until it reaches the position defined as point of interest for the cart. After that, the character waits until the user is looking at it. As soon as the user's attention is directed at the character, it starts explaining the instrumented shopping cart's capabilities. If during the talk, the user looks away, the character tries to regain his attention. If the user is not interested in the explanation about the device and moves away, the character follows the user to the next device he might be interested in.

In some situations, several user events can occur at the same time. In this case, the behavior module needs an additional conflict solving mechanism in order to be able to react in a deterministic way. For this purpose, we have assigned priority values to the different event types, so that in case of simultaneous occurrence of different events, the one with the highest priority is preferred. In our scenario, it is probable that the events *looking away* and *moving away* occur at the same time. In this case, the *moving away* event is regarded as the more significant one and thus, it is assigned a higher priority. The *looking away* event is regarded as the second significant one, and the events *present* and *watching* are not in conflict with each other, so that they have the same lowest priority.

6.4 User-driven DUVDs

Beside the applications involving implicit interaction, we have developed some services using explicit user interaction. These user-driven DUVDs are controlled mainly by the user, especially in terms of their positions and content. In these applications, the user can intentionally enforce system feedback or also create individual Virtual Displays with customized content.

6.4.1 SearchLight

The SearchLight service implements a physical search function for Instrumented Environments ([Butz et al., 2004]). It can be regarded as an analogy to the file search functionality on a common PCs, which represent virtual environments. SearchLight transfers this interaction paradigm into the physical environment, where the targets of the search are physical objects, and accordingly, the response to the search query also occurs in the physical world.

Instrumented Environments as discussed in this work offer the ability to extend our physical surroundings by a computational layer providing new functionalities. One such functionality can be the capability of objects to make themselves known in order to be noticed or found by humans. This functionality was already proposed in Weiser's Ubiquitous Computing vision ([Weiser, 1991]). A search function for physical environments would alleviate the need to keep track of all of the things in our environment. One possible application is keeping track of books in an office, a library or a book store. In our exemplary scenario, we consider a library with ubiquitous display capabilities and a conventional inquiry terminal to find out about books. The inquiry interface on the computer terminal in a library could then just provide an additional *show me* button for the selected book, which sends an event to the SearchLight service initiating a highlight of the corresponding book position in the shelf.

For the implementation of the SearchLight service, first of all, the system needs knowledge about the locations of the searched objects. Similar to the hand-tracking approach used in the pointing gesture module (PEG) described in Section 6.2.2.1.2, the visual marker library ARToolKit is deployed for the location detection of objects which are considered as potential targets for a search request. However, in contrast to the pointing gesture approach, the SearchLight module uses a projector-mounted camera to detect the visual markers.

The SearchLight service operates in two phases: In a preparation phase, it scans the room for optical markers and sets up a repository of the detected markers and the corresponding locations. This information is used in the operating phase in order to display projected hints onto the searched objects. In the following, the processes performed in both phases are presented.

***Figure 6.36**: SearchLight* scan*: images taken during the scanning process [left] and an example of an AR Toolkit marker [right]*

Scan: The environment is scanned by taking slightly overlapping pictures in all horizontal and vertical directions using the projector-mounted camera (see Figure 6.36 [left]). Each picture is analyzed using jARToolKit[12], a Java version of the original ARToolKit library. The IDs of all detected markers are stored in a repository together with their computed locations in 3D space. These locations are derived from the marker position and orientation in the camera image and the orientation of the steerable projector unit at which these image has been taken. The jARToolKit library computes the marker position in the camera coordinate system. A combination of this displacement with the current transition matrix of the steerable unit in the room coordinate system provides the location and orientation of the visual marker

[12] http://sourceforge.net/projects/jartoolkit/

in the coordinate system of the DUVD system.

Show: After the room has been initially scanned, users can search for marked objects, e.g. books in a shelf. If the user submits a search query using a common library interface, the result of this query can be visualized in the physical environment. For this purpose, each library entry contains the ID of the ARToolKit marker attached to the corresponding book. If this marker has been identified during the scan phase, its location is looked up in the generated repository and a Virtual Display is created and activated at the computed location. This display shows a projected spot highlighting the searched book (see Figure 6.37) similar to the approach which has been realized in the previously described PEG module for direct user interaction (see Section 6.2.2.1.2).

Figure 6.37*: SearchLight* show*: a highlighted book in the shelf [left] and another one on the window sill [right]*

With our experimental setup (room size 5m x 6m, shelf on the wall, 4 megapixels camera with 3x optical zoom mounted on the steerable projector unit), we were able to reliably recognize markers down to a size of 10mm in the whole room. In the current demo, scanning is done only once when SearchLight is started. In an advanced version, idle times of the projection system could be deployed to systematically rescan the environment for possible changes in order to adapt the search repository. This process can also prioritize regions where changes are more likely, which could be identified using additional sensors, such as RFID tags, optical recognition (e.g. FibreShelf [Krüger et al., 2011]) or motion detection with additional cameras. Even when the projector unit is used for other tasks, the image of the projector-mounted camera can be analyzed for possible visual markers, and thus the scan process of the SearchLight service can be performed as a side effect of other applications.

In theory, existing bar codes on many products could be used for the recognition process. In the case of books, optical character recognition (OCR) or image recognition (e.g. Google Goggles[13]) could even completely eliminate the need for markers, since book spines and covers are designed to clearly identify books.

[13]http://www.google.com/mobile/goggles/

6.4.2 Beam-Its

Sticky notes, better known by the brand name Post-it, are an important tool for many people to organize their daily lives. They provide a convenient way of attaching small amounts of information, such as a few words or a sketch, to objects and places in our physical environments. They can be used as reminders of duties and appointments and their ubiquity and convenient form factor is hard to match.

The Beam-Its service is an elaborated user-driven application implementing a virtual version of sticky notes, which can be placed in the physical environment. Beam-Its can be created by the user on a PDA and placed on the surfaces of the Display Continuum in the environment, where they are visualized when needed or required. Beam-Its can contain handwritten text and sketches, just as the common physical Post-it notes. Alternatively, Beam-Its can also be created as typed text, using the virtual keyboard of the PDA.

In keeping with our research aims, we have investigated how a popular concept from the physical world can be extended to an Instrumented Environment setting, where the physical and digital worlds mix. The Beam-Its service illustrates what a virtual version of the traditional Post-it concept could look like and which advantages it would have over the physical version. In an environment providing a large Display Continuum, virtual sticky notes can appear or remain invisible in the environment depending on specific context parameters, such as time or identities of people present. They can also contain multimedia content in addition to simple text or sketches. Figure 6.38 shows a simple scenario demonstrating an obvious advantage in terms of privacy.

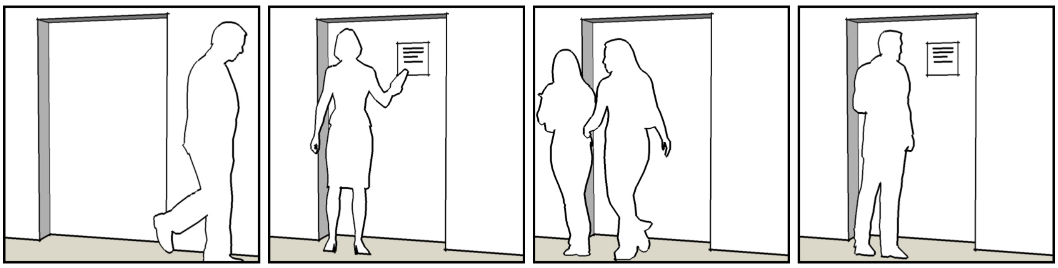


Figure 6.38: *Beam-Its example scenario: Mr. Smith leaving his office; Mrs. Smith creating a private virtual message; people passing the door without seeing Mrs. Smith's message; Mr. Smith finding the Beam-It left for him [from left to right] (author: Andreas Butz)*

In the example scenario, Mr. Smith leaves his office to get a cup of coffee. While he is away, his wife passes by and finds his door closed. As she cannot not talk personally to her husband, Mrs. Smith leaves him a personal virtual sticky note on the door. When other people pass Mr. Smith's door, it appears empty to them because the created Beam-It remains invisible. Only when the system detects the presence of Mr. Smith upon his return, the virtual note left by Mrs. Smith is visualized at the door.

With the Beam-Its module developed in this work and implemented at the SUPIE Instrumented Environment (see Section 2.2), this and other scenarios can be realized.

A number of desktop applications implement electronic sticky notes, including Post-it

Digital Notes[14], Stickies[15], Hott Notes[16], PtiMémo[17], NoteZilla[18] or Quick Notes Plus[19]. They allow users to create digital messages which closely mimic conventional sticky notes and to place them on their virtual desktops or attach them to documents or web pages. Some of these digital notes can also contain pictures or alarms which remind the user of an upcoming appointment, but all of them live purely on virtual desktops and do not appear in the real world.

Two recent approaches to virtual messages in the real world are the Digital Graffitti Service[20] and the Place-Its application [Sohn et al., 2005], with which users can leave digital messages anywhere in the environment using their mobile phones. With the former, tourists can mark interesting locations and share their experiences with others visiting the same place, and with the latter, users can place reminders for themselves at predefined locations (e.g. home, work), so that they receive these messages when they arrive at these particular places. A user study conducted with the Place-It application shows that location-based reminders are in general considered useful and enjoyable, although some of the participants asked for time-constrained reminders, which were not offered by the application. These results encourage us in our belief that virtual messages that can be location- as well as time- and user-dependent might be an enrichment for people's daily lives. In contrast to the approaches described above, in our work, we are trying to seamlessly augment the physical environment to the bare eyes of the user deploying a projection-based immediately visualized Display Continuum.

When we first designed the Beam-Its, we considered a visual appearance very close to the physical Post-It version, i.e., a clearly marked yellow square with black or colored pen strokes. In preliminary tests, this design exhibited considerable disadvantages for projection. The brightness of the typical Post-It yellow reduces the usable contrast range for the projected ink strokes and the design also interfered with physical objects onto which the notes were projected. Therefore, we decided to implement a very simple background-less and frameless design, which maximizes the usable contrast and minimizes graphical elements. Slight modifications, such as individual pen colors or borders, could also be used to identify authors or other properties of a note, such as priority. If future versions were to use a different base technology, such as electronic wallpaper, these design decisions would have to be reconsidered.

In our prototype implementation, the user can create virtual sticky notes with a tracked PDA, using the stylus and a specific note-taking application (see Figure 6.39 [left]). Instead of tearing a paper note from a pad, he then taps a *Beam-It* button on the PDA interface, and the created message appears as a projected note shown on a Virtual Display on the Display Continuum in front of the user's current location (see Figure 6.39 [right]). Like in the previously described VRI application (see Section 6.3.3), the user position is detected by means of the indoor location system Always Best Positioned using active RFID tags and infrared

[14] http://www.3m.com/us/office/postit/digital/digital_notes.html

[15] http://www.zhornsoftware.co.uk/stickies/

[16] http://www.hottnotes.com

[17] http://ptimemo.lynanda.com

[18] http://www.conceptworld.com/NoteZilla

[19] http://www.conceptworld.com/qnp

[20] https://www.ct.siemens.com/en/technologies/se/beispiele/graffitis.html

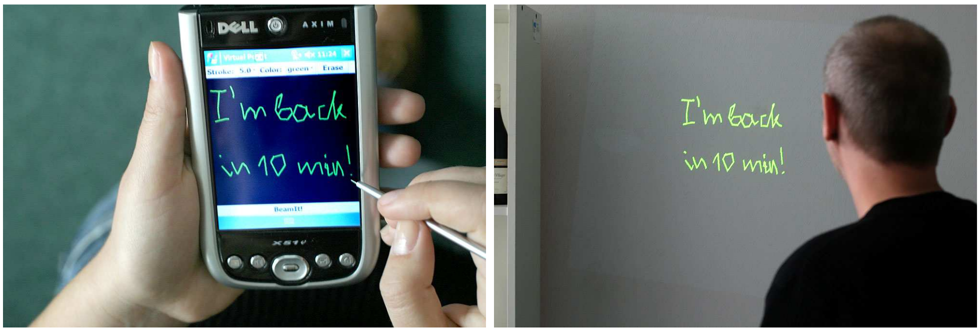


***Figure 6.39**: A Beam-It as it is written on the PDA [left] and displayed on the wall [right]*

beacons ([Brandherm and Schwartz, 2005], [Schwartz et al., 2005]), and his current orientation is additionally determined by an electronic compass, which can be either attached to the user's PDA or integrated into his clothes. This location system offers an accuracy of about 2m and a precise detection of the user's orientation, which allows only a rough determination of the final Beam-It position. This accuracy turned out to be mostly sufficient for our purposes and we expect it to be even improved by further development of the indoor location system.

After it is placed in the environment, the Beam-It virtually "sticks" to this location and is displayed there or hidden as appropriate, depending on the situation. This basic interaction scheme closely mimics the physical Post-Its as we know them, which allows the transfer of a widely familiar mental model and makes the application easily understandable.

In contrast to conventional paper sticky notes, Beam-Its can exhibit certain additional properties. They can be *time-dependent* and/or *personal*. On a technical level, the display of the Beam-It message is controlled by events, such as the detection of the presence of a particular person or the approaching of a certain date.

Time-dependent Beam-Its act as reminders by appearing at a predefined date. In an office environment, these virtual messages can be used to remind users of meetings or other events. A time-dependent Beam-It can be specified to appear either at a predefined fixed location, or if the reminder is assigned to a certain user, it can be shown near the current location of this user, which is detected by the user tracking module.

Personal Beam-Its are specifically addressed to a certain person or a group of pepople and provide a privacy mechanism which is impossible to achieve with physical sticky notes. The personal Beam-It is only displayed when the respective person is present. In order to detect this presence, users are tracked with the previously introduced indoor localization module.

The simple Beam-Its described so far just contain electronic ink (i.e. polygonal strokes) and digital text. They could therefore be implemented with minimal network traffic, resulting in a very good interactive behavior. We have also experimented with a number of extensions using different types of data, such as photos taken by the PDA's camera and sound recordings using the built-in microphone.

We have also developed and advanced version of the Beam-It application for Android smartphones (see Appendix A.3). In addition to the hand-written Beam-Its, it also offers the opportunity to create typed text Beam-Its using a virtual keyboard on the phone (see Figure

6.40).

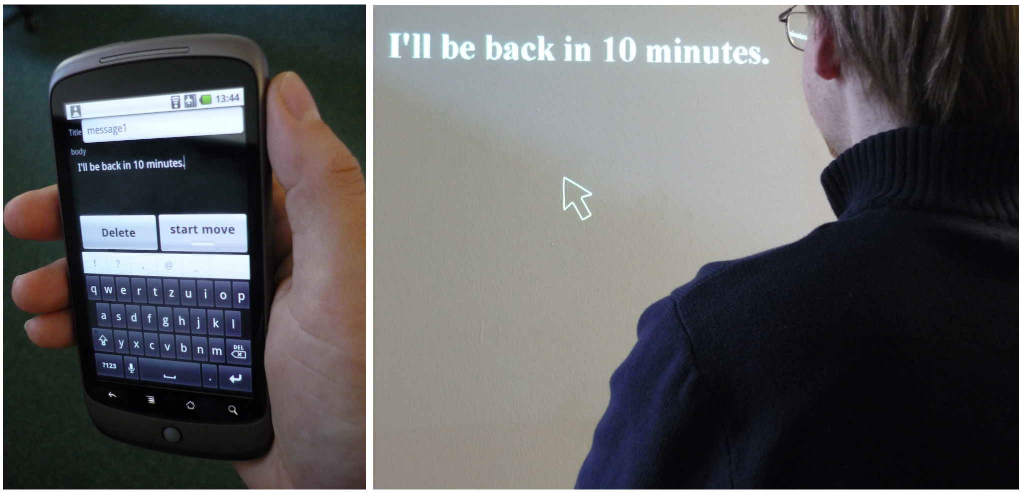


Figure 6.40*: A Beam-It typed on the smartphone [left] and displayed on the wall [right]*

Moreover, the built-in orientation sensors of the smartphone (3D accelerometer and digital compass) are used to control the position of the Dynamic Peephole. For this purpose, the accelerometer-based gesture interaction module developed for the Wiimote device (see Section 6.2.2.2) has been adapted for the Android smartphone. The absolute orientation obtained through the phone's digital compass allows to adjust the movement of the Ubiquitous Cursor in such a way that it follows the current orientation of the phone.

Alternatively, for a more fine-grained adjustment, the Ubiquitous Cursor can also be controlled through a kind of cross pad which is displayed on the phone's touchscreen. It consists of two circular zones, whereby the outer circle triggers a faster movement of the cursor than the inner one. While the user presses a certain part of the cross pad, the Ubiquitous Cursor moves in the corresponding direction in the room (see Figure 6.41).

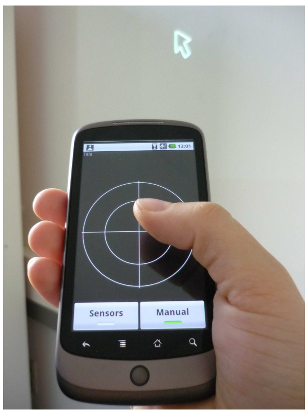


Figure 6.41*: Cross pad provided by the Beam-It application for adjusting the Ubiquitous Cursor*

6.5 Synopsis

In this chapter, we have presented the various interaction modules of the DUVD system and a number of applications using both explicit or implicit interaction approaches. Table 6.2 shows an overview of these modules and applications regarding their interaction type, the applied interaction devices and the corresponding interaction methods.

The proposed desktop interfaces use a visualization of the Display Continuum as a 3D model, which the user can work with using traditional keyboard and mouse input. These interfaces allow an exact and – if needed – remote manipulation of Virtual Displays and Dynamic Peepholes. However, when working at a stationary desktop, the user might be distracted from his surroundings, where the actual system output is presented.

Real-world interfaces use the physical space for both interaction input and system output. In this chapter, we have presented different real-world interfaces based on computer vision, motion sensors, and user and object tracking. In order to support the input capabilities of certain interaction devices, we have also developed a number of projected menus and widgets.

	Interaction type	Interaction devices	Interaction methods
3D desktop interfaces	explicit	Physical monitor, mouse and keyboard	Clicking, dragging, buttons and menus
Interaction via colored objects	explicit	Projector-mounted camera, colored objects	Postures and movement with colored object, projected widgets
Pointing Extra Gesture (PEG)	explicit	External camera, tagged colored glove	Pointing; (further input modalities by MSA)
Accelerometer-based interaction	explicit	Device with 3D accelerometer and at least one button (e.g. Wiimote, smartphone)	Continuous and discrete gestures (arm movement), projected menus
Product Associated Displays (PADs)	implicit	RFID-labeled objects, instrumented shelf	Object placement; (further input modalities by MSA)
Micro Navigation	implicit	IRL Smart Cart	Entering zones of interest
Virtual Room Inhabitant (VRI)	implicit	User tracking (e.g. infrared beacons)	User movement
SearchLight	explicit/ implicit	Visually tagged objects	Search query
Beam-Its	explicit/ implicit	Mobile device (PDA, smartphone)	Typing, drawing, writing, buttons, menus on mobile device

Table 6.2: *Overview of DUVD interaction modules and applications*

Part IV

Conclusion and Further Research Opportunities

7 Conclusion

In this work, we have presented a conceptual design and a prototypical implementation of a framework for Dynamic Ubiquitous Virtual Displays (DUVDs), which allow the creation and manipulation of visual content on the surfaces of an appropriately equipped Instrumented Environment. Although the theoretical concepts can be applied to a variety of enabling technologies, the present work, focuses on a projection-based realization.

In our theoretical framework, we have introduced and defined the new concepts *Display Continuum*, *Virtual Display*, *Dynamic Peephole* and *Ubiquitous Cursor*. We have developed a 3D model for the representation and visualization of a Display Continuum, which includes not only potential display surfaces but also irregularities like *obstacles*, *shadows* and *discontinuities*. Further, we have presented a theoretical model of Dynamic Ubiquitous Virtual Displays outlining the basic parameters with which DUVDs can be defined, and we have pointed out how these parameters can be discretely or continuously modified in order to achieve a desired effect. In this context, we have investigated a broad range of interfaces paradigms for DUVDs.

Several interaction modules have been implemented in the course of this work, including 3D interfaces and diverse gesture interaction methods. Real-world interaction has been realized in various ways: vision-based interaction has been developed using different camera setups, and accelerometer-based gesture interaction has been prototypically implemented by deploying a standard, commercially available device (Wii Remote). The presented interaction techniques encompass both explicit and implicit input concepts.

Finally, we have presented a number of exemplary applications demonstrating the capabilities and advantages of the DUVD concepts for complex presentation and interaction tasks. These applications employ both user-driven as well as system-driven Virtual Displays in various scenarios in the office and retail context.

7.1 Scientific Contributions

At the beginning of this thesis, we have specified a number of research questions and technical challenges, with which the present work has been concerned. In the following, we will return to these questions and point out how the concepts and approaches developed in the present work have made a contribution towards solving the problems that have been addressed.

- Which functionalities have to be supported by a ubiquitous display system?

  In the present work, we have introduced the concept of Dynamic Ubiquitous Virtual Displays which represent a generalized metaphor for presenting visual output in Augmented Reality applications. We have proposed a general definition of Virtual Displays and specified a number of basic display parameters which can be modified discretely or continuously in order to adapt the ubiquitous visual output to some specific intrinsic or extrinsic constraints. In Section 5.6, we proposed a number of basic functionalities which a DUVD system should support, and in Chapter 6, we presented a number of interaction modules and applications implementing these functionalities.

- Which methods and interfaces are suitable for interaction with ubiquitous displays?

  In order to enable interaction with the projection-based ubiquitous displays developed in the present work, we have designed and implemented a variety of interfaces, ranging from 3D interfaces for remote desktop applications, to vision-based and accelerometer-based gesture interaction modules. The latter have been designed for both implicit and explicit user interaction. The proposed interaction concepts have been prototypically implemented in a number of different applications. The main contribution in this context is the development of real-world interfaces for projection-based ubiquitous displays, whose interaction space is aligned with the user's physical environment.

- Which theoretical models can be used to describe ubiquitous display systems?

  In our theoretical framework, we have defined the new concept of a Display Continuum, which represents a virtual layer in a physical environment on which visual content in terms of Virtual Displays can be presented. We have identified a number of characteristics for classifying different types of Display Continua. For visualizing specific parts of a Display Continuum with restricted visual concurrency, we have proposed the concept of Dynamic Peepholes which represent virtual windows into the visual content of a Display Continuum. They can be both user- and system-driven, where the latter type can be applied in order to guide the user's attention at a specific location and to provide system feedback. In order to indicate the current interaction focus, the Ubiquitous Cursor has been introduced as an equivalent to the traditional mouse cursor transferred into the physical environment.

- How can projection-based ubiquitous displays be combined with physical screens?

  In conjunction with the accelerometer-based gesture interaction module, we have prototypically embedded a physical screen into the projection-based Display Continuum. In this way, the Ubiquitous Cursor concept can be applied across the technology border.

This allows the transfer of Virtual Displays from the Display Continuum visualized by the physical screen to its projection-based part and vice versa.

- Which contextual knowledge is needed when working with ubiquitous displays?

  In our theoretical framework of DUVDs, we have identified a number of extrinsic constraint parameters which can have an impact on certain Virtual Display parameters. These extrinsic parameters can be, for example, certain points in time, the identity of a present user and his current location, or the location and identity of a certain object. Using these parameters, it is possible to specify certain constraints concerning, for example, the movement, activity and content of a Virtual Display, which enables the realization of various applications, such as the creation of Product Associated Displays, projected navigational hints, user-adaptive advertisement in supermarkets and virtual Post-its.

- Which projection-based display systems are offered by other research projects and how can they be classified?

  In Section 2.5, we outlined some special characteristics of steerable projection systems, and we discussed their benefits and limitations in providing visual output in Augmented Reality applications in contrast to other projection-based technologies, head-mounted displays and traditional monitors. The first systematic and comprehensive overview of projection-based display systems was presented in Chapter 4 encompassing technically complex immersive environments, large-scale multi-projector displays using front- or rear-projection and several steerable projection systems with different hardware setups and conceptual fundamentals. A closer look was taken at the latter group of systems, whose characteristics were contrasted with the ones of the DUVD system developed in the course of this work.

In the following, we summarize the technical challenges which have been solved while developing the DUVD system presented in this thesis.

- How can a steerable projector system be installed in the environment?

  In order to facilitate the creation of a 3D model of the environment, which is needed for enabling a distortion-free projection with the Fluid Beam system, we have developed a new approach to user-assisted calibration of potential projection surfaces using visual markers. As a further alternative for modeling the physical environment, we have extended the functionality of the modeling toolkit Yamamoto to provide methods for modeling steerable projection devices and Virtual Displays. Following the concept of Dual Reality, the Display Continuum in the physical environment and the corresponding model in Yamamoto have been synchronized so that they can mutually influence each other.

- Which input devices can be applied in order to implement the new concepts for interaction with projection-based ubiquitous displays?

  For the technical realization of the developed interaction concepts, we have tested and explored a variety of devices in terms of their usability concerning the interaction with ubiquitous displays in the physical environment. Some of these devices were prototypes, while others were off-the-shelf appliances which had to be adapted to our specific requirements.

- Which components/macros are needed in order to build projection-based ubiquitous display scenarios?

  As a further extension of the Yamamoto toolkit, we have implemented a module for determining the regions lying within the potential projection area of a steerable projection unit in order to identify possible display locations for projection-based Virtual Displays. This functionality enables the simulation of steerable projector setups in a virtual model in order to optimize the coverage of the resulting Display Continuum. In this way, it is possible to determine the appropriate positions for installing steerable projection units in a specific environment using the simulation in the 3D model prior to the actual mounting of the devices in the physical environment.

7.2 Opportunities for Further Research

Based on the concepts proposed in this thesis and the implemented DUVD system, further research in the area of ubiquitous displays can be conducted. In the following, we propose some possible topics for future development and extension of the presented work.

Development and deployment of further interaction technologies and methods
In addition to the interaction technologies applied in the present work, the DUVD system can be extended by further interaction modules, enabling for example speech and other acoustic input. A natural language user interface using spoken language would be a considerable improvement for the interaction with virtual characters realized on ubiquitous displays. Further acoustic interfaces could use common sounds, such as finger snapping or whistling, for controlling the ubiquitous display system.

In order to achieve a more robust and user-friendly vision-based interaction, 3D cameras providing additional depth information can be applied for gesture recognition. In this way, visual markers, which we currently use as auxiliary means to enable vision-based hand recognition, will become obsolete.

Finally, a closer look can be taken at possible opportunities for multimodal and multi-user interaction with ubiquitous displays in order to increase the naturalness of the interaction with the environment. In this context, combining speech and gestures appears to be a promising approach.

Applying the DUVD concepts to further DC technologies
As already pointed out, steerable projection is only one possibility to realize ubiquitous displays. We assume that ongoing and future research in the area of novel display technologies will lead to the development of highly flexible and easily configurable physical display foils in the near future. This technology would enable an effortless embedding of physical display

surfaces in common objects. If digital wallpaper covering the walls of a room could be realized in this way, the DUVD concepts could be applied to this new technology and possibly extended to its specific features and limitations.

Aside form the considered combination of projection-based ubiquitous displays and stationary physical screens, further technology combinations can be investigated, e.g. steerable projection and mobile projection, steerable projection and handheld devices, or stationary screens and handheld devices.

Combination of several steerable projector units

In the present work, the combination of several steerable projector units in the same physical environment has been considered only in terms of the developed shadow simulation module. Another important topic is the interplay of several steerable projection units in the same room with partially overlapping projection areas. In this case, a mechanism for synchronizing the individual units is needed in order to guarantee an optimal exploitation of the corresponding Dynamic Peepholes. The technical challenge in this context is to achieve a correct representation of ubiquitous displays moving across the overlapping projection areas of the various devices. Further, a resource distribution concept and a media allocation approach are needed in order to enable simultaneous projection of several ubiquitous displays at spatially distant locations.

Adding further degrees of freedom to the steerable projector unit

The steerable projector unit applied in the current implementation of the DUVD system has only two degrees of freedom (pan and tilt rotation). In cluttered environments, this hardware setup results in large regions which cannot be reached by the projector beam because of occlusion. This limitation can be counteracted by developing a hardware setup providing further degrees of freedom. This can be achieved for example by mounting the steerable pan-tilt unit on an appliance whose position can be adjusted using motorized winches, such as the one of the so-called Spidercam[1], which is used to freely position video cameras in a given physical environment.

A further opportunity to add more spatial flexibility to the steerable projector is to mount the pan-tilt unit on a mobile robotic platform. In combination with a localization system, the steerable projector could autonomously move through the environment and thus reach the desired display surfaces.

[1]http://spidercam.org/

Part V

Appendix

A INTERACTION DEVICES

A.1 TZI SCIPIO Gesture Band and WInspect Glove

In the course of the wearIT@work project[1], a general-purpose wearable input device called TZI SCIPIO Gesture Band was developed that integrates a 3-axis acceleration sensor and an RFID reader in a one-size-fits-all elastic wristband (see Figure A.1). The measured sensor data is sent to a PC via a Bluetooth module and a Li-Ion battery pack guarantees up to 8 hours operation time. An additional input modality is provided by integrated buttons. Status LEDs and programmable audio can be used for output.

For host systems with a Java VM and a JSR82 Java-Bluetooth-API, a device interface class is available. It can establish the connection to a specific TZI SCIPIO Gesture Band and deliver its sensor data to an application program. Possible host systems are Linux and Microsoft Windows and java-enabled mobile devices such as PDAs and mobile phones.

The WUI (Wearable User Interface) Toolkit, which is a part of the European Wearable Computing Framework contains a driver for the device. It allows gesture-based interaction with the application programs using the device.

***Figure A.1**: TZI SCIPIO Gesture Band (source: http://matrix.wearlab.de/scipio/)*

Another input device developed in the same project is the WInspect Glove ([Lawo et al., 2006]). It contains basically the same hardware as the Gesture Band but it is built in the form of a fingerless glove with minimal covering on the inside of the arm

[1] www.wearitatwork.com

and wrist (see Figure A.2). Contrary to the Gesture Band, the Glove has 3 textile buttons that fit around the fingers and do not interfere with manual tasks. It is designed for hybrid gesture-based or direct selection interaction.

***Figure A.2**: TZI SCIPIO WInspect Glove (source: http://matrix.wearlab.de/scipio/)*

Regarding their motion sensing capabilities, both devices described above are suitable for gesture-based interaction. In fact, the WInspect Glove is more comfortable than the Gesture Band as it allows for purely one-handed gesture interaction thanks to the cleverly placed finger buttons, which can be used to trigger different actions.

A.2 Wii Remote (Wiimote)

Another interaction device using acceleration sensors is the Wii Remote controller. It is originally developed and distributed as a controller for Nintendo's Wii gaming console but was soon discovered by the research community as a powerful tool for interaction with various applications. Beside a 3-axis accelerometer, the Wii Remote Control disposes of an infrared camera, which can be used to calculate the relative motion of the device with respect to an infrared light source. Additionally, the Wii Remote Control offers six input buttons and a four-way digital cross (D-pad) on its front side and a larger trigger button on the back side (see Figure A.3 [left]). Output can be provided by four blue LEDs, a loud speaker and a vibration motor (rumble function).

The built-in acceleration sensor can measure linear acceleration in three directions (see Figure A.3 [right]). If the device is not being moved, it delivers an upward acceleration value (+Z, when horizontal) equal to the force of gravity g (approximately 9.8 m/s^2) but in the opposite direction. This fact can be used to recognize tilt movements with the Wiimote.

Several Application Programming Interfaces (APIs) are available for the Wiimote, which allow a connection of the device to a computing system. In this way, sensor input can be received from the Wiimote and its output (LEDs, rumble and sound) can be controlled remotely via a Bluetooth connection. APIs are provided for all main programming languages, e.g. the Wiimote API[2] for C, Wiim[3] for C++, and the following APIs are Java-based: WiiuseJ[4], motej[5] and jwiimote[6].

[2] http://code.google.com/p/wiimote-api/

[3] http://digitalretrograde.com/projects/wiim/

[4] http://code.google.com/p/wiiusej/

[5] http://motej.sourceforge.net/

[6] http://code.google.com/p/jwiimote/

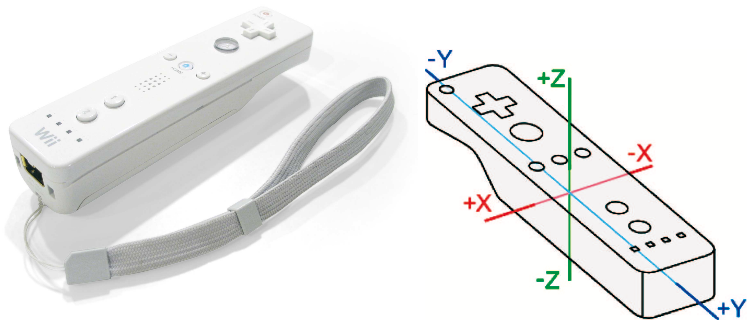


***Figure A.3**: Wii Remote controller device [left] and coordinate system of the acceleration sensor [right]*

A.3 PDA with Digital Compass / Smartphone

In order to enable mobile interaction with projected Virtual Displays, a PDA has been equipped with an RFID antenna and a digital compass. By means of an indoor location system using active RFID tags and infrared beacons ([Schwartz et al., 2005], [Brandherm and Schwartz, 2005]), the position of the user holding the PDA can be tracked. With the attached digital compass, the user orientation can also be measured precisely. In this way, the system can estimate which surfaces the user is currently looking at and use them for the placement of Virtual Displays. With the touch display of the PDA, the user has the opportunity to create individual content for the projected displays.

In a recent update of our system, we have applied a modern Android smartphone (Nexus One[7]) for interaction with Virtual Displays, which provides more elaborate features than the previously described PDA. Aside from a built-in digital compass, the smartphone contains a 3-axis accelerometer similar to the one of the Wiimote device. With the data provided by these sensors, it is possible to determine the absolute orientation of the smartphone in 3D space.

Figure A.4 shows some screenshots of the extended Beam-It application running on a Nexus One smartphone.

[7] http://en.wikipedia.org/wiki/Nexus_One

Figure A.4: *Screenshots of Beam-It application on Nexus One smartphone: initial window showing a list of previously created Beam-Its, menu for color selection and menu for stroke width adjustment [top row, from left to right]; typed Beam-It content, hand-drawn Beam-It content and the cross pad for position adjustment of the Ubiquitous Cursor [bottom row, from left to right]*

B Experimental Evaluation of Wiimote Interaction with DUVDs

Evaluating newly developed systems by means of controlled user studies is a common practice in order to investigate the usability of new approaches. To set up such a usability study, an appropriate scenario has to be prepared, wherein a number of preferably unbiased users are asked to perform a number of tasks using the system being tested. Usually, the test persons are observed during the performance of the experiment, and additionally, they are asked to fill in some pre- and post-test questionnaires in order to gather feedback on the system being evaluated.

Among the interaction techniques implemented in the course of this work, the accelerometer-based gesture interaction described in Section 6.2.2.2 was selected for evaluation by means of a usability study, as one of the novel real-world interaction techniques developed in the course of the present work. As we have implemented two slightly different variations of the accelerometer-based interaction concept using the Wii Remote controller, the aim of the performed user study was to compare these two interaction options with each other in terms of usability. Additionally, the accelerometer-based interaction options have been compared against the solely button-based interaction technique with the Wii Remote controller, which has also been described in Section 6.2.2.2.

B.1 Participants

As test subjects for our user study, we recruited twenty participants (eleven male and nine female) mainly from the local university campus (Saarland University, Germany). They ranged in age from 19 to 39 years with an average age of 26 years. In a demographic questionnaire, which the participants were asked to fill in after the experiment, we gathered general information about the test subjects, like gender, age and profession. Furthermore, the questionnaire contained some questions concerning the participants' general acquaintance with computers and their experiences with the Nintendo Wii console and gesture interaction in particular (see Figures B.3 and B.4).

It turned out that 9 of the participants have had no experience at all with the Wii console, 7 test subjects stated to be somewhat familiar with it, and 4 participants assessed themselves as being very familiar with the Wii console. Concerning their experience with other gesture-based interaction devices (beside the Wii Remote), 17 of the participants stated that they have never used such devices and only 3 have used gesture interaction before.

B.2 Experimental tasks

In order to give the participants the opportunity to apply a broad set of input commands in each interaction option, we have prepared a fixed list of tasks which the test subjects were asked to perform with each of the interaction options respectively (gestures and one button, gestures and two buttons, only buttons). To avoid skewed results due to learning effects, we counterbalanced the order of the three interaction options, while in each interaction option, the order of the performed tasks was maintained. The tasks were designed to range from simple to complex ones. The completion of a simple task requires only one simple action (e.g. the movement of the Ubiquitous Cursor or the opening of a menu). A complex task involves several simple actions, which can possibly be performed in various ways (e.g. the creation of a Virtual Display with specific properties). In some cases, there are even different alternatives (different sets of simple actions) for the completion of a complex task.

The tasks which the participants were asked to perform in the study were the following:

1. Move the Ubiquitous Cursor to *position1*.
2. Create a display (*display1*) which is 0.42m high and 0.63m wide at this position (*position1*).
3. Show the image *message* as content on this display (*display1*).
4. Create a live stream showing the area of the clock application on the plasma screen.
5. Show the live stream on a new display (*display2*) to the left of the first display (*display1*).
6. Move all visible displays to *position2*.
7. Create a new display (*display3*) with the following properties:
    a. It is located at *position1*;
    b. It has a 25 degree rotation;
    c. It shows the image *flower*.
8. Go to *display1* (which shows the image *message*).
9. Delete all visible displays.

Out of these nine tasks, the tasks 2, 6, 7 and 9 are complex tasks. The locations *position1* and *position2* were labeled on the walls in the Instrumented Environment with numbered yellow markers. The images *message* and *flower* could be obtained from two image files, which were shown on the plasma screen. The digital clock window involved in task 4 was also displayed on the plasma screen.

B.3 Procedure

Every participant was asked to perform the previously defined tasks using each of the three interaction options respectively. Prior to the beginning of the experiment, the participants were given a brief general introduction to the application and the overall procedure of the experiment. They were informed that they will be asked to perform a number of tasks in three rounds, using three different interaction options respectively.

More specific instructions concerning the respective interaction option were given prior to each interaction round. The participants were instructed how to control the application with the respective interaction option, and they were given the opportunity to test it in a short trial session. Additionally, at the beginning of the first gesture-based interaction round, the participants were asked to create their own user profiles by training the predefined interaction gestures. After each interaction round, the participants rated the applied interaction option by filling in an experiment questionnaire (see Figures B.5 – B.8). It comprises eleven statements, where the last two (10 and 11) are only relevant for the gesture-based interaction options (and not for the solely button-based interaction). Each of these statements could be rated on a rating scale from 1 (totally disagree) to 6 (totally agree). The rating scale was built with an even number of points in order to avoid the possibility of a neutral rating.

The statements of the rating questionnaire look like the following (originally in German):

1. The movement of the Ubiquitous Cursor was intuitive.
2. I was always aware of where the Ubiquitous Cursor was.
3. I always knew on which object the interaction focus was currently lying.
4. I knew exactly what to do (which button to click, which gesture to perform) in order to accomplish a task.
5. The navigation through the menu items was intuitive.
6. The increase/decrease of numbers (e.g. when resizing or rotating a projected display) was intuitive.
7. The Wiimote was an easy to use interaction device for controlling the Ubiquitous Cursor as well as the Virtual Displays.
8. I was able to complete all tasks successfully.
9. I felt comfortable with the interaction.
10. I knew exactly which gesture to perform in order to trigger a desired action.
11. I knew exactly when and on which button to click in order to trigger a desired action.

After the completion of all three interaction rounds, the participants were asked to give a general rating comparing the three interaction options with each other under several aspects and to specify a personal ranking of the interaction options reflecting their experiences during the user study (see Figures B.9 and B.10).

The following statements were to be rated for the direct comparison of the interaction options:

1. With the following interaction option, I could easily complete the given tasks.

2. In my opinion, the following interaction option is intuitive.

3. I would like to use the following interaction option in everyday life or at work.

Again, for each interaction option, each statement could be rated from 1 (totally disagree) to 6 (totally agree).

At the end of each rating questionnaire, the participants were given the opportunity to add some personal comments concerning their experiences during the user study.

Table B.1 shows a schematic overview of the experimental procedure described above.

General introduction	Brief general introduction to the application and the overall procedure of the experiment
Instruction 1	Introduction to the first interaction option with trial
Interaction round 1	Task completion using the first interaction option
Questionnaire 1	Evaluation of the first interaction option
Instruction 2	Introduction to the second interaction option with trial
Interaction round 2	Task completion using the second interaction option
Questionnaire 2	Evaluation of the second interaction option
Instruction 3	Introduction to the last interaction option with trial
Interaction round 3	Task completion using the third interaction option
Questionnaire 3	Evaluation of the last interaction option
Questionnaire comparing all three interaction options	Direct comparison of all three interaction options
Demographic questionnaire	Demographic questions on age, gender, experience with Wiimote, etc.

Table B.1: *Schematic overview of the experimental procedure*

B.4 Results

In order to analyze and interpret the results of the user study, the data gathered through the different questionnaires was evaluated using statistical methods[1]. Figure B.1 shows an overview of the most interesting results of the evaluation. In all diagrams, the blue bars/sections refer to the average rating results concerning the solely button-based interaction option (BI), the green ones refer to the gesture interaction option with two buttons (GI2) and the violet ones refer to the gesture interaction option with one buttons (GI1). A table showing a distribution of corresponding scale groups is added below each diagram (except for the pie chart).

Comparison of the three interaction options

As it can be seen in the pie chart of Figure B.1 a., the button-based interaction outperformed both gesture-based interaction options according to the overall user ranking, with 70% of the participants declaring it to be their favored interaction option. 20% of the participants voted for the gesture-based interaction with two buttons, while only 10% preferred the gesture-based interaction with only one button. This ranking can be interpreted with respect to the complexity of the respective interaction options: generally, people are accustomed to operating a system using buttons, e.g. when working with a remote control. In contrast, gesture-based interaction is currently by far less common as means for controlling electronic devices. When comparing the complexity of the two gesture-based interaction options, it can be stated that using two buttons to specify the respective (continuous or discrete) gesture mode is cognitively less demanding than using only one button to switch between the two gesture modes, especially when the same button also has to be pressed to indicate that a gesture is currently being performed. This overloading of the button functionality possibly makes the GI1 interaction too complicated for unexperienced users.

Although statement 4 ("I knew exactly what to do (which button to click, which gesture to perform) in order to accomplish a task.") has been rated similarly for all three interaction options, the evaluation of statement 7 ("The Wiimote was an easy to use interaction device for controlling the Ubiquitous Cursor as well as the Virtual Displays.") shows a significantly higher rating for the BI option, which means that the participants felt much more comfortable using the Wiimote as an ordinary button-based remote control than as a gesture interaction device.

The average ratings for statement 8 ("I was able to complete all tasks successfully.") support the popularity rating of the different interaction options. Since, in general, humans enjoy the feeling of success, it appears consistent that interaction options which are perceived as more successful are preferred by the users. This is also confirmed by the results for statement 1 ("With the following interaction option, I could easily complete the given tasks.") of the direct comparison questionnaire.

The rating results for statement 9 ("I felt comfortable with the interaction.") support the results for statement 8, which means that the ease of use of the Wiimote device in a certain interaction option is related to the user comfort with the respective interaction.

Diagrams f. and g. illustrate the reported user confidence in using gestures and buttons with the two gesture-based interaction options. Surprisingly, the results show no significant

[1] The results were evaluated using SPSS version 16.

difference between the two options, which does not support our previous assumption that the gesture-based interaction with two buttons is less complex than the one with only one button.

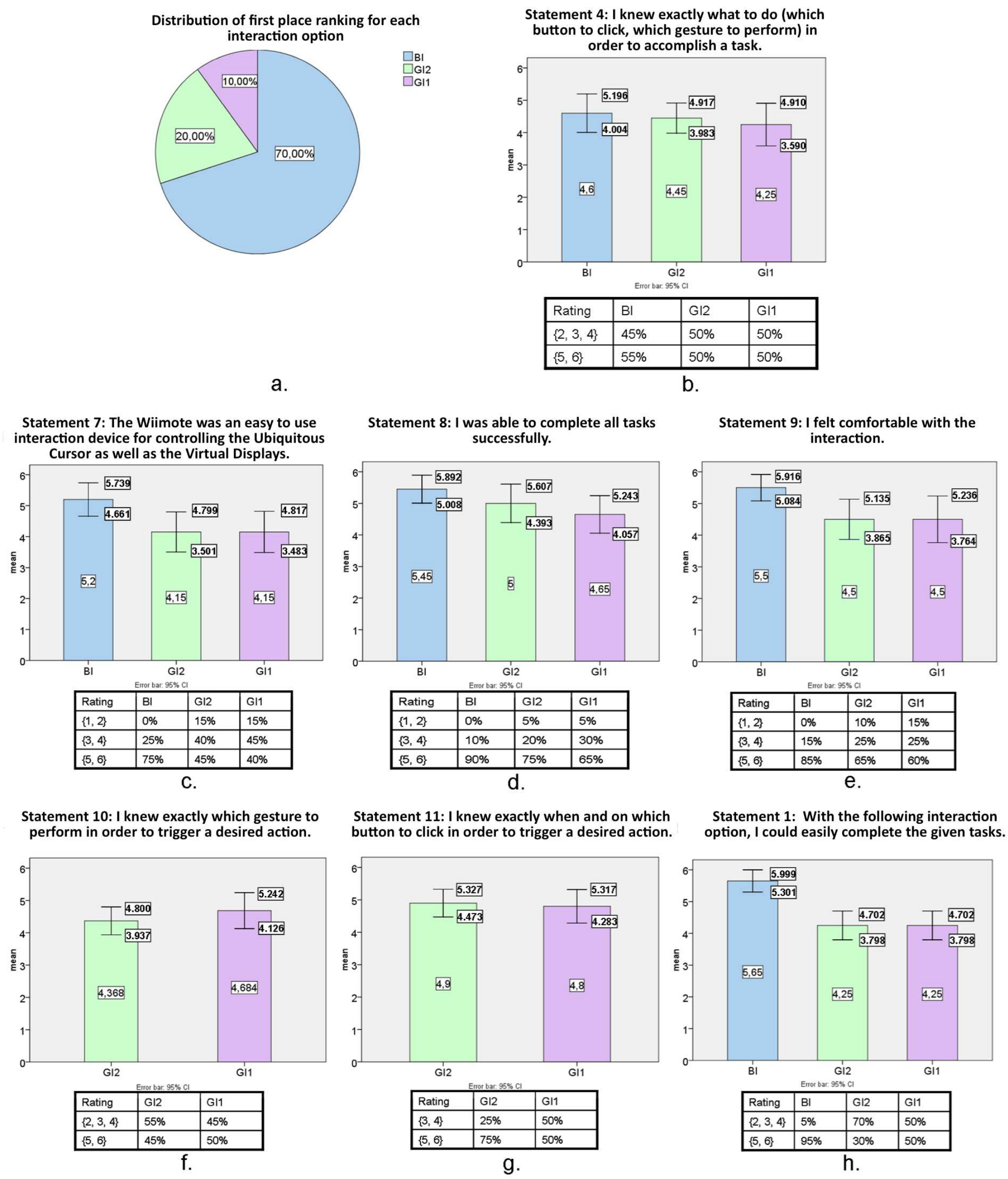


Figure B.1*: Statistical diagrams showing the evaluation results: a: distribution of first place ranking; b: statement 4; c: statement 7; d: statement 8; e: statement 9; f: statement 10; g: statement 11; h: statement 1 of the questionnaire comparing the three interaction options (b – h: with denoted 95% confidence intervals)*

As commented by some participants as well as observed in the course of the user study, the participants often had to repeat the performed gestures once or twice before they could be recognized correctly be the system. In rare cases, some gestures had to be repeated up to 5 to 7 times. These technical shortcomings obviously lead to user frustration in some cases. Furthermore, it could be observed that sometimes participants confused the usage of the buttons in the different gesture-based interaction options. Often, the second performed gesture-based interaction option was confused with the previously performed one.

One participant rated most statements for both gesture-based interaction options with values from 1 to 3, as he often had to repeat the gestures several times before they were recognized correctly by the system. Another participant, who rated each interaction option with 4 to 6 on average, provided the value 1 for the statement “I would like to use the following interaction option in my everyday life or at work” for all interaction options. As a reason for this, he commented that he could not imagine a possible scenario in which he could use these kinds of interaction.

In the demographic questionnaire, all participants acknowledged that they use computers either for professional or for recreational reasons – 15 out of 20 stated that they are working with computers more than 25 hours per week. The majority of the participants (16) had never or only occasionally played with a Wii console, and only 3 of them have had experiences with other gesture input devices beside the Wiimote. Therefore, we can assume that our test subjects were more accustomed to use buttons-based controls in their everyday lives, and the gesture-based approach was a rather unfamiliar concept to them.

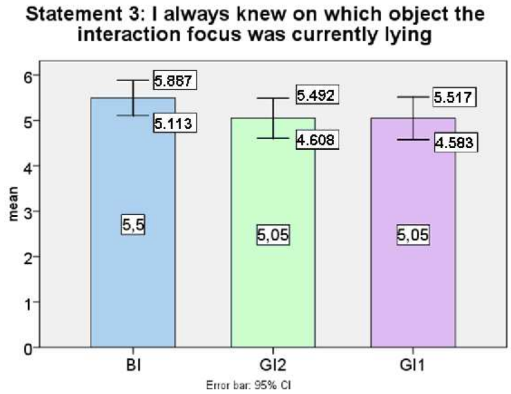
Benefits of visual feedback during interaction

The results presented in the diagram in Figure B.2 show that the participants were quite aware of the interaction focus during all interaction options. For statement 3 (“I always knew on which object the interactions focus was currently lying.”), the average ratings for all three interaction options are higher than 5, and the table with the corresponding scale group distribution shows that, in all conditions, the statement was rated with a value of at least 3, with the majority of participants assigning a 5 or 6.

As, in all interaction conditions during the experiments, the main indicator of the current interaction focus was the visual feedback provided by the Ubiquitous Cursor and the red border indicating a selected Virtual Display (see Figures 6.27 and 6.28 (d)), these results support the assumption that the provided visual feedback is appropriate for guiding the user’s focus during interaction in the physical space. This assumption is further confirmed by some user comments provided in the questionnaires, saying that both the Ubiquitous Cursor as well as the red border were perceived to be helpful for successfully completing the tasks.

Final conclusions

As an overall conclusion of the results derived from the performed usability testings, it can be stated that occasional technical problems concerning the gesture recognition and the resulting difficulties in completing a given task have had a negative influence on the user acceptance and the usability of the gesture-based interaction. This encourages us in further improving the accelerometer-based gesture recognition process in order to achieve a more robust gesture interaction.



Rating	BI	GI2	GI1
{3, 4}	10%	30%	25%
{5, 6}	90%	70%	75%

***Figure B.2**: Statistical diagrams showing the evaluation results for statement 3 with denoted 95% confidence intervals and a distribution of corresponding scale groups,*

Due to the only slight differences between the two gesture-based interaction options, during the experiments, the participants often confused the actions of these two interaction types. We assume that if users are given the opportunity to practice using only one of the gesture interaction options, they will be able to understand the interaction concept and memorize the applied actions more easily.

Further, we could observe a user preference for button-based interaction, which can be explained by a probably greater experience in working with button-based devices in their daily lives. We hope that with an increasing popularity of gesture controls for example in gaming applications, gesture interaction will also gain in popularity in other areas of people's everyday lives.

Regardless of the applied interaction options, the provided visual feedback indicating the target of the current interaction focus has proved to be appreciated by the participants, and it will be considered in future interaction modules.

Questionnaire number:

Personal information:

Age: **Gender:**

Profession:

1. How many hours per week do you spend in front of a computer?

- ○ 0 – 1 hours weekly
- ○ 1 – 10 hours weekly
- ○ 10 – 25 hours weekly
- ○ 25 – 40 hours weekly
- ○ > 40 hours weekly

2. For which purpose do you normally use the computer? (multiple selections allowed)

Professional reasons

- ○ Surfing the Internet / reading emails
- ○ Work with office suite
- ○ Software development
- ○ Image editing
- ○ Video / sound editing
- ○ Other:

Recreational reasons

- ○ Playing games
- ○ Surfing the Internet / reading emails
- ○ Work with office suite
- ○ Software development
- ○ Image editing
- ○ Video / sound editing
- ○ Other:

***Figure B.3**: Demographic questionnaire (1/2)*

3. How familiar are you with the Nintendo Wii game console?

   ○ Not at all familiar (never played)
   ○ Somewhat familiar (played a few times)
   ○ Familiar (often played)
   ○ Very familiar (played very often and well)

4. Do you have any experience with gesture-based interaction apart from the one used by the Nintendo Wii game console?

   ○ No
   ○ Yes, namely:

   ____

   ____

Figure B.4*: Demographic questionnaire (2/2)*

1. The movement of the Ubiquitous Cursor was intuitive.

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree totally agree

Comments / Suggestions:

2. I was always aware of where the Ubiquitous Cursor was.

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree totally agree

Comments / Suggestions:

3. I always knew on which object the interaction focus was currently lying.

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree totally agree

Comments / Suggestions:

Figure B.5*: Questionnaire on each interaction option (1/4)*

4. I knew exactly what to do (which button to click, which gesture to perform) in order to accomplish a task.

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

____

____

____

5. The navigation through the menu items was intuitive.

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

____

____

____

6. The increase/decrease of numbers (e.g. when resizing or rotating a projected display) was intuitive.

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

____

____

____

***Figure B.6**: Questionnaire on each interaction option (2/4)*

7. The Wiimote was an easy to use interaction device for controlling the Ubiquitous Cursor as well as the Virtual Displays.

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

---

8. I was able to complete all tasks successfully.

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

---

9. I felt comfortable with the interaction.

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

Figure B.7*: Questionnaire on each interaction option (3/4)*

Statements concerning only the gesture-based interaction options:

10. I knew exactly which gesture to perform in order to trigger a desired action.

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

____

____

____

11. I knew exactly when and on which button to click in order to trigger a desired action.

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6

totally disagree totally agree

Comments / Suggestions:

____

____

____

***Figure B.8**: Questionnaire on each interaction option (4/4)*

1. With the following interaction option, I could easily complete the given tasks.

only buttons

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree — totally agree

gestures and one button

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree — totally agree

gestures and two buttons

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree — totally agree

2. In my opinion, the following interaction option is intuitive.

only buttons

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree — totally agree

gestures and one button

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree — totally agree

gestures and two buttons

☐ ☐ ☐ ☐ ☐ ☐
1 2 3 4 5 6
totally disagree — totally agree

***Figure B.9**: Questionnaire comparing the three interaction options (1/2)*

3. I would like to use the following interaction option in my everyday life or at work.

only buttons

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree … totally agree

gestures and one button

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree … totally agree

gestures and two buttons

☐ ☐ ☐ ☐ ☐ ☐

1 2 3 4 5 6

totally disagree … totally agree

4. Please give a ranking of the three interaction options in the order of your personal preference.

1: ___________________________________
2: ___________________________________
3: ___________________________________

Comments / Suggestions:

____
____
____
____
____
____
____
____
____

Figure B.10: *Questionnaire comparing the three interaction options (2/2)*

List of Figures

List of Tables

Bibliography

[Ashdown et al., 2004] Ashdown, M., Flagg, M., Sukthankar, R., and Rehg, J. M. (2004). A Flexible Projector-Camera System for Multi-Planar Displays. In *Proceedings of Computer Vision and Pattern Recognition*, pages 165–172.

[Ashdown and Robinson, 2001] Ashdown, M. and Robinson, P. (2001). The Writing's on the Wall: Large, Remotely Controlled Displays. In *Proceedings of Euro-CLS*, pages 83–88, Maastricht, Netherlands.

[Ashdown and Robinson, 2003a] Ashdown, M. and Robinson, P. (2003a). Experiences Implementing and Using Personal Projected Displays. In *IEEE Workshop on Projector-Camera Systems*, Nice, France.

[Ashdown and Robinson, 2003b] Ashdown, M. and Robinson, P. (2003b). The Escritoire: A Personal Projected Display. In *Journal of the Winter School of Computer Graphics*, volume 11, pages 33–40, Plzen, Czech Republic.

[Ashdown and Robinson, 2005] Ashdown, M. and Robinson, P. (2005). Escritoire: A Personal Projected Display. *IEEE Multimedia*, 12(1):34–42.

[Ashdown and Sato, 2005] Ashdown, M. and Sato, Y. (2005). Steerable Projector Calibration. In *IEEE International Workshop on Projector-Camera Systems (PROCAMS 2005)*, San Diego, California, USA.

[Bandyopadhyay et al., 2001] Bandyopadhyay, D., Raskar, R., and Fuchs, H. (2001). Dynamic Shader Lamps: Painting on Movable Objects. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, pages 207–216, Washington, DC, USA. IEEE Computer Society.

[Bimber et al., 2005] Bimber, O., Emmerling, A., and Klemmer, T. (2005). Embedded Entertainment with Smart Projectors. In *ACM SIGGRAPH 2005 Courses*, page 8, New York, NY, USA. ACM.

[Bimber and Raskar, 2005] Bimber, O. and Raskar, R. (2005). *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Ltd., Natick, MA, USA.

[Bishop and Welch, 2000] Bishop, G. and Welch, G. (2000). Working in the Office of ”Real Soon Now”. *IEEE Computer Graphics and Applications*, 20(4):76–78.

[Borkowski et al., 2005] Borkowski, S., Letessier, J., and Crowley, J. L. (2005). Spatial Control of Interactive Surfaces in an Augmented Environment. *Engineering Human Computer Interaction and Interactive Systems*, 3425/2005:228–244.

[Borkowski et al., 2003] Borkowski, S., Riff, O., and Crowley, J. L. (2003). Projecting Rectified Images in an Augmented Environment. In *ICCV Workshop on Projector-Camera-Systems*. IEEE Computer Society.

[Brandherm and Schwartz, 2005] Brandherm, B. and Schwartz, T. (2005). Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems. In *Proceedings of the International Workshop on Location- and Context-Awareness (LoCA)*, volume 3479 of *Lecture Notes in Computer Science*, pages 223–234. Springer Berlin / Heidelberg.

[Broll et al., 2010] Broll, G., Reithmeier, W., Graebsch, R., Holleis, P., and Wagner, M. (2010). Dynamic NFC-Displays as a Prototyping Platform for Direct Mobile Interactions with Public Displays. In *UBI Challenge Workshop 2010 at Ubicomp 2010*, Copenhagen, Denmark.

[Brooks, 1997] Brooks, R. A. (1997). The Intelligent Room project. In *CT ’97: Proceedings of the 2nd International Conference on Cognitive Technology (CT ’97)*, page 271, Washington, DC, USA. IEEE Computer Society.

[Brown et al., 2005] Brown, M., Majumder, A., and Yang, R. (2005). Camera-Based Calibration Techniques for Seamless Multiprojector Displays. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):193–206.

[Butz and Krüger, 2003] Butz, A. and Krüger, A. (2003). A Generalized Peephole Metaphor for Augmented Reality and Instrumented Environments. *International Workshop on Software Technology for Augmented Reality Systems (STARS)*.

[Butz and Krüger, 2006] Butz, A. and Krüger, A. (2006). Applying the Peephole Metaphor in a Mixed-Reality Room. *IEEE Computer Graphics and Applications*, 26(1):56–63.

[Butz et al., 2004] Butz, A., Schneider, M., and Spassova, M. (2004). SearchLight – A Lightweight Search Function for Pervasive Environments. In *Proceedings of the 2nd International Conference on Pervasive Somputing*, pages 351–356, Vienna, Austria.

[Cao and Balakrishnan, 2006] Cao, X. and Balakrishnan, R. (2006). Interacting with Dynamically Defined Information Spaces Using a Handheld Projector and a Pen. In *UIST ’06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 225–234, New York, NY, USA. ACM.

[Cao et al., 2007] Cao, X., Forlines, C., and Balakrishnan, R. (2007). Multi-user Interaction Using Handheld Projectors. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST ’07, pages 43–52, New York, NY, USA. ACM.

[Castronovo et al., 2011] Castronovo, S., Endres, C., Del Fabro, T., Schnabel, N., and Müller, C. A. (2011). Multimodal Conspicuity-Enhancement for E-Bikes: the Hybrid Reality Model of Environment Transformation. In *Proceedings of the 16th international conference on Intelligent user interfaces*, IUI ’11, pages 433–434, New York, NY, USA. ACM.

[Ciger et al., 2003] Ciger, J., Gutierrez, M., Vexo, F., and Thalmann, D. (2003). The Magic Wand. In *Proceedings of the 19th spring conference on Computer graphics*, SCCG ’03, pages 119–124, New York, NY, USA. ACM.

[Coen, 1998a] Coen, M. H. (1998a). A Prototype Intelligent Environment. In *CoBuild ’98: Proceedings of the First International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, pages 41–52. Springer Berlin / Heidelberg.

[Coen, 1998b] Coen, M. H. (1998b). Design Principles for Intelligent Environments. In *AAAI ’98/IAAI ’98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 547–554, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Cook and Das, 2004] Cook, D. and Das, S. (2004). *Smart Environments: Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing)*. John Wiley Sons, Hoboken (New Jersey).

[Cook and Das, 2007] Cook, D. J. and Das, S. K. (2007). How Smart are our Environments? An Updated Look at the State of the Art. *Pervasive and Mobile Computing*, 3(2):53–73.

[Cotting and Gross, 2006] Cotting, D. and Gross, M. (2006). Interactive Environment-Aware Display Bubbles. In *UIST ’06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 245–254, New York, NY, USA. ACM.

[Cruz-Neira et al., 1993] Cruz-Neira, C., Sandin, D. J., and DeFanti, T. A. (1993). Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. In *SIGGRAPH ’93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142, New York, NY, USA. ACM.

[Decker et al., 2005] Decker, C., Krohn, A., Beigl, M., and Zimmer, T. (2005). The Particle Computer System. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN ’05, pages 443–448, Piscataway, NJ, USA. IEEE Press.

[Dijkstra, 1959] Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15:11–15.

[Ehnes, 2010] Ehnes, J. (2010). An Audio Visual Projection System for Virtual Room Inhabitants. In *Proceedings of the 20th International Conference on Artificial Reality and*

Telexistance (ICAT 2010), pages 118–125, Adelaide, Australia. The Virtual Reality Society of Japan (VRSJ).

[Ehnes and Hirose, 2006] Ehnes, J. and Hirose, M. (2006). Projected Reality – Content Delivery Right onto Objects of Daily Life. In *Advances in Artificial Reality and Tele-Existence*, volume 4282/2006 of *Lecture Notes in Computer Science*, pages 262–271. Springer Berlin / Heidelberg.

[Ehnes et al., 2004] Ehnes, J., Hirota, K., and Hirose, M. (2004). Projected Augmentation – Augmented Reality using Rotatable Video Projectors. In *ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 26–35, Washington, DC, USA. IEEE Computer Society.

[Ehnes et al., 2005] Ehnes, J., Hirota, K., and Hirose, M. (2005). Projected Augmentation II — A Scalable Architecture for Multi Projector Based AR-Systems Based on 'Projected Applications'. In *ISMAR '05: Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 190–191, Washington, DC, USA. IEEE Computer Society.

[Encarnação, 2007] Encarnação, J. L. (2007). HEI! – The Human Environment Interaction. In *Proceedings of the 2007 conference on Human interface: Part I*, pages 623–631. Springer Berlin / Heidelberg.

[Fitzmaurice, 1993] Fitzmaurice, G. W. (1993). Situated Information Spaces and Spatially Aware Palmtop Computers. *Communications of the ACM*, 36(7):39–49.

[Fitzmaurice et al., 1993] Fitzmaurice, G. W., Zhai, S., and Chignell, M. H. (1993). Virtual Reality for Palmtop Computers. *ACM Transactions on Information Systems (TOIS)*, 11(3):197–218.

[Grønbæk et al., 2007] Grønbæk, K., Iversen, O. S., Kortbek, K. J., Nielsen, K. R., and Aagaard, L. (2007). Interactive Floor Support for Kinesthetic Interaction in Children Learning Environments. In *INTERACT'07: Proceedings of the 11th IFIP TC 13 International Conference on Human-Computer Interaction*, pages 361–375. Springer Berlin / Heidelberg.

[Gross et al., 2003] Gross, M., Würmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-meier, E., Svoboda, T., Gool, L. V., Lang, S., Strehlke, K., Moere, A. V., Oliver, Zürich, E., and Staadt, O. (2003). blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence. In *ACM Transactions on Graphics*, pages 819–827, New York, NY, USA. ACM.

[Grossberg et al., 2004] Grossberg, M. D., Peri, H., Nayar, S. K., and Belhumeur, P. N. (2004). Making One Object Look Like Another: Controlling Appearance Using a Projector-Camera System. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:452–459.

[Guimbretière et al., 2001] Guimbretière, F., Stone, M., and Winograd, T. (2001). Fluid Interaction with High-resolution Wall-size Displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 21–30, New York, NY, USA. ACM.

[Guimbretière and Winograd, 2000] Guimbretière, F. and Winograd, T. (2000). FlowMenu: Combining Command, Text, and Data Entry. In *Proceedings of the 13th annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 213–216, New York, NY, USA. ACM.

[Hankins and Silverman, 1999] Hankins, T. L. and Silverman, R. J. (1999). *Instruments and the Imagination*. Princeton University Press, Princeton, New Jersey, second printing, and first paperback printing edition.

[Hardy et al., 2010] Hardy, R., Rukzio, E., Holleis, P., and Wagner, M. (2010). Mobile Interaction with Static and Dynamic NFC-based Displays. In *Proceedings of the 12th international conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 123–132, New York, NY, USA. ACM.

[Hilliges et al., 2008] Hilliges, O., Kim, D., and Izadi, S. (2008). Creating Malleable Interactive Surfaces using Liquid Displacement Sensing. In *Proceedings of the 3rd IEEE International Workshop on Tabletops and Interactive Surfaces (Tabletop 2008)*, pages 157–160.

[Ishii, 2008] Ishii, H. (2008). Tangible Bits: Beyond Pixels. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages xv–xxv, New York, NY, USA. ACM.

[Ishii and Ullmer, 1997] Ishii, H. and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA. ACM.

[Iversen et al., 2007] Iversen, O. S., Kortbek, K. J., Nielsen, K. R., and Aagaard, L. (2007). Stepstone – An Interactive Floor Application for Hearing Impaired Children with a Cochlear Implant. In *IDC '07: Proceedings of the 6th international conference on Interaction design and children*, pages 117–124, New York, NY, USA. ACM.

[Iwai et al., 2006] Iwai, D., Hanatani, S., Horii, C., and Sato, K. (2006). Limpid Desk: Transparentizing Documents on Real Desk in Projection-Based Mixed Reality. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, page 319, Washington, DC, USA. IEEE Computer Society.

[Iwai and Sato, 2006] Iwai, D. and Sato, K. (2006). Limpid Desk: See-Through Access to Disorderly Desktop in Projection-Based Mixed Reality. In *VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 112–115, New York, NY, USA. ACM.

[Jacob et al., 2008] Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., and Zigelbaum, J. (2008). Reality-Based Interaction: A Framework for Post-WIMP Interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI ’08, pages 201–210, New York, NY, USA. ACM.

[Johnson et al., 2007] Johnson, T., Gyarfas, F., Skarbez, R., Towles, H., and Fuchs, H. (2007). A Personal Surround Environment: Projective Display with Correction for Display Surface Geometry and Extreme Lens Distortion. *IEEE Virtual Reality Conference*, 0:147–154.

[Jonides, 1983] Jonides, J. (1983). Further towards a model of the mind’s eye’s movement. *Bulletin of the Psychonomic Society*, 21(4):247–250.

[Kahl et al., 2009] Kahl, G., Leichtenstern, K., Schöning, J., Spassova, L., and Krüger, A. (2009). Contextual Learning Game for Toddlers Installed on an Interactive Display Attached to a Shopping Cart. In *Proceedings of PerED 2009: Workshop on Pervasive Computing Education (PerED-09)*, pages 1–8, Orlando, Florida, USA.

[Kahl et al., 2010] Kahl, G., Spassova, L., and Krüger, A. (2010). User-adaptive Advertisement in Retail Environments. In *Workshop on Pervasive Advertising and Shopping*.

[Kahl et al., 2011] Kahl, G., Spassova, L., Schöning, J., Gehring, S., and Krüger, A. (2011). IRL SmartCart – A User-adaptive Context-aware Interface for Shopping Assistance. In *Proceedings of the 16th international conference on Intelligent user interfaces*, IUI ’11, pages 359–362, New York, NY, USA. ACM.

[Kahl et al., 2008] Kahl, G., Wasinger, R., Schwartz, T., and Spassova, L. (2008). Three Output Planning Strategies for Use in Context-aware Computing Scenarios. In *Proceedings of the AISB 2008 Symposium on Multimodal Output Generation (MOG-2008)*, number 10, pages 46–49.

[Kane et al., 2009] Kane, S. K., Avrahami, D., Wobbrock, J. O., Harrison, B., Rea, A. D., Philipose, M., and LaMarca, A. (2009). Bonfire: A Nomadic System for Hybrid Laptop-Tabletop Interaction. In *UIST ’09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 129–138, New York, NY, USA. ACM.

[Kato and Billinghurst, 1999] Kato, H. and Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, Washington, DC, USA. IEEE Computer Society.

[Kendon, 2001] Kendon, A. (2001). An Agenda for Gesture Studies. *Semiotic Review of Books*, 7(3):7–12.

[Kleinrock, 1995a] Kleinrock, L. (1995a). Nomadic computing. In *Keynote Address: International Conference on Mobile Computing and Networking (MOBICOM)*.

[Kleinrock, 1995b] Kleinrock, L. (1995b). Nomadic Computing – An Opportunity. *ACM SIGCOMM Computer Communication Review*, 25(1):36–40.

[Kresse et al., 2003] Kresse, W., Reiners, D., and Knöpfle, C. (2003). Color Consistency for Digital Multi-Projector Stereo Display Systems : The HEyeWall and The Digital CAVE. In *Proceedings of the workshop on Virtual environments 2003*, EGVE '03, pages 271–279, New York, NY, USA. ACM.

[Kröner et al., 2009] Kröner, A., Gebhard, P., Spassova, L., Kahl, G., and Schmitz, M. (2009). Informing Customers by Means of Digital Product Memories. In *Proceedings of the 1st international Workshop on Digital Object Memories (DOME-09)*, volume 4 of *Ambient Intelligence and Smart Environments, AISE*, pages 21–26. IOS Press, Amsterdam.

[Kröner et al., 2010] Kröner, A., Kahl, G., Spassova, L., Feld, T., Mayer, D., Magerkurth, C., and Dada, A. (2010). Demonstrating the Application of Digital Product Memories in a Carbon Footprint Scenario. In *Proceedings of the Sixth International Conference on Intelligent Environments (IE-10)*, pages 164–169. IEEE Computer Society.

[Krüger et al., 2011] Krüger, A., Schöning, J., and Olivier, P. (2011). How Computing Will Change the Face of Retail. *Computer*, 44:84–87.

[Krüger et al., 2010] Krüger, A., Spassova, L., and Jung, R. (2010). Innovative Retail Laboratory – Investigating Future Shopping Technologies. *IT - Information Technology*, 52(2):114–118.

[Kruppa, 2006] Kruppa, M. (2006). *Migrating Characters: Effective User Guidance in Instrumented Environments*. Akademische Verlagsgesellschaft Aka GmbH, Heidelberg.

[Kruppa et al., 2005] Kruppa, M., Spassova, L., and Schmitz, M. (2005). The Virtual Room Inhabitant – Intuitive Interaction With Intelligent Environments. In *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence (AI05)*, pages 225–234, Sydney, Australia.

[Lawo et al., 2006] Lawo, M., Witt, H., Kenn, H., Nicolai, T., and Leibrand, R. (2006). A Glove for Seamless Computer Interaction - Understand the WINSPECT. In Kenn, H., Glotzbach, U., and Herzog, O., editors, *Proceedings of the Smart Glove Workshop*, pages 3–8.

[Lee et al., 2005] Lee, J. C., Hudson, S. E., Summet, J. W., and Dietz, P. H. (2005). Moveable Interactive Projected Displays Using Projector Based Tracking. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 63–72, New York, NY, USA. ACM.

[Li et al., 2000a] Li, K., Chen, H., Chen, Y., Clark, D. W., Cook, P., Damianakis, S., Essl, G., Finkelstein, A., Funkhouser, T., Housel, T., Klein, A., Liu, Z., Praun, E., Samanta, R., Shedd, B., Singh, J. P., Tzanetakis, G., and Zheng, J. (2000a). Building and Using A Scalable Display Wall System. *IEEE Computer Graphics and Applications*, 20:29–37.

[Li et al., 2000b] Li, K., Chen, H., Chen, Y., Clark, D. W., Cook, P., Damianakis, S., Essl, G., Finkelstein, A., Funkhouser, T., Housel, T., Klein, A., Liu, Z., Praun, E., Samanta, R., Shedd, B., Singh, J. P., Tzanetakis, G., and Zheng, J. (2000b). Early Experiences and Challenges in Building and Using A Scalable Display Wall System. *IEEE Computer Graphics and Applications*, 20(4):671–680.

[Lifton and Paradiso, 2009] Lifton, J. and Paradiso, J. A. (2009). Dual Reality: Merging the Real and Virtual. In *Proceedings of the First International ICST Conference on Facets of Virtual Environments (FaVE)*, volume 33, pages 12–28. Springer Berlin / Heidelberg.

[Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, E77-D(12):1321–1329.

[Mistry and Maes, 2009] Mistry, P. and Maes, P. (2009). SixthSense: a Wearable Gestural Interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, page 1, New York, NY, USA. ACM.

[Molyneaux and Gellersen, 2009] Molyneaux, D. and Gellersen, H. (2009). Projected Interfaces: Enabling Serendipitous Interaction with Smart Tangible Objects. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 385–392, New York, NY, USA. ACM.

[Molyneaux et al., 2007] Molyneaux, D., Gellersen, H., Kortuem, G., and Schiele, B. (2007). Cooperative Augmentation of Smart Objects with Projector-Camera Systems. In *Proceedings of 9th International Conference on Ubiquitous Computing*, volume 4717/2007 of *Lecture Notes in Computer Science*, pages 501–518. Springer Berlin / Heidelberg.

[Molyneaux and Kortuem, 2004] Molyneaux, D. and Kortuem, G. (2004). Ubiquitous Displays in Dynamic Environments: Issues and Opportunities. In *Workshop on Ubiquitous Display Environments*, Nottingham, England.

[Naef et al., 2004] Naef, M., Staadt, O., and Gross, M. (2004). blue-c API: A Multimedia and 3D Video Enhanced Toolkit for Collaborative VR and Telepresence. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 11–18, New York, NY, USA. ACM.

[Nielsen, 1993] Nielsen, J. (1993). Noncommand User Interfaces. *Communications of the ACM*, 36:83–99.

[Pingali et al., 2003] Pingali, G., Pinhanez, C., Levas, A., Kjeldsen, R., Podlaseck, M., Chen, H., and Sukaviriya, N. (2003). Steerable Interfaces for Pervasive Computing Spaces. In *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, page 315, Washington, DC, USA. IEEE Computer Society.

[Pingali et al., 2002] Pingali, G., Pinhanez, C., Levas, T., Kjeldsen, R., and Podlaseck, M. (2002). User-Following Displays. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'02)*, volume 1, pages 845–848. IEEE Computer Society.

[Pinhanez, 2001a] Pinhanez, C. (2001a). The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. *Lecture Notes in Computer Science*, 2201:315–331.

[Pinhanez, 2001b] Pinhanez, C. (2001b). Using a Steerable Projector and a Camera to Transform Surfaces into Interactive Displays. In *CHI '01 extended abstracts on Human factors in computing systems*, CHI '01, pages 369–370, New York, NY, USA. ACM.

[Pinhanez et al., 2003] Pinhanez, C., Kjeldsen, R., Levas, A., Pingali, G., Podlaseck, M., and Sukaviriya, N. (2003). Applications of Steerable Projector-Camera Systems. In *IEEE International Workshop on Projector-Camera Systems at ICCV 2003*.

[Prante et al., 2004] Prante, T., Streitz, N., and Tandler, P. (2004). Roomware: Computers Disappear and Interaction Evolves. *Computer*, 37(12):47–54.

[Prensky, 2001] Prensky, M. (2001). Digital Natives, Digital Immigrants. *On The Horizon*, 9(5):1–6.

[Raij et al., 2003] Raij, A., Gill, G., Majumder, A., Towles, H., and Fuchs, H. (2003). PixelFlex2: A Comprehensive, Automatic, Casually-Aligned Multi-Projector Display. In *IEEE International Workshop on Projector-Camera Systems*.

[Rapp and Weber, 2005] Rapp, S. and Weber, I. (2005). LumEnActive: A Novel Presentation Tool for Interactive Installations. In *6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, Pisa, Italy.

[Rapp and Weber, 2010] Rapp, S. and Weber, I. (2010). LumEnActive and Pervasive Digital Signage Projection. In *Pervasive Advertising and Shopping Workshop*, Helsinki, Finland.

[Raskar, 2002] Raskar, R. (2002). *Projector-Based Three Dimensional Graphics*. PhD thesis, University of North Carolina at Chapel Hill.

[Raskar et al., 1999] Raskar, R., Brown, M. S., Yang, R., Chen, W.-C., Welch, G., Towles, H., Seales, B., and Fuchs, H. (1999). Multi-Projector Displays Using Camera-Based Registration. In *VIS '99: Proceedings of the conference on Visualization '99*, pages 161–168, Los Alamitos, CA, USA. IEEE Computer Society.

[Raskar and Low, 2001] Raskar, R. and Low, K.-L. (2001). Interacting with Spatially Augmented Reality. In *Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation*, AFRIGRAPH '01, pages 101–108, New York, NY, USA. ACM.

[Raskar et al., 2005] Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C. (2005). iLamps: Geometrically Aware and Self-Configuring Projectors. In *ACM SIGGRAPH 2005 Courses*, page 5, New York, NY, USA. ACM.

[Raskar et al., 1998] Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. (1998). The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *SIGGRAPH '98: Proceedings of the 25th annual conference* *on Computer graphics and interactive techniques*, pages 179–188, New York, NY, USA. ACM.

[Raskar et al., 2001] Raskar, R., Welch, G., Low, K.-L., and Bandyopadhyay, D. (2001). Shader Lamps: Animating Real Objects With Image-Based Illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 89–102. Springer Berlin / Heidelberg.

[Rehg et al., 2002] Rehg, J. M., Flagg, M., Cham, T.-j., Sukthankar, R., and Sukthankar, G. (2002). Projected Light Displays Using Visual Feedback. *Proceedings of the Seventh International Conference on Control, Automation, Robotics and Vision*, pages 926–932.

[Reiter and Butz, 2005] Reiter, D. and Butz, A. (2005). Design and Implementation of a Widget Set for Steerable Projector-Camera Units. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 216–217, Washington, DC, USA. IEEE Computer Society.

[Rekimoto and Saitoh, 1999] Rekimoto, J. and Saitoh, M. (1999). Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, New York, NY, USA. ACM.

[Rheingold, 1991] Rheingold, H. (1991). *Virtual Reality*. Summit Books.

[Salvi et al., 2004] Salvi, J., Pagès, J., and Batlle, J. (2004). Pattern Codification Strategies in Structured Light Systems. *Pattern Recognition*, 37(4):827–849.

[San Agustin et al., 2010] San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., and Hansen, J. P. (2010). Evaluation of a Low-Cost Open-Source Gaze Tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications*, pages 77–80, New York, NY, USA. ACM.

[Schlömer et al., 2008] Schlömer, T., Poppinga, B., Henze, N., and Boll, S. (2008). Gesture Recognition with a Wii Controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, TEI '08, pages 11–14, New York, NY, USA. ACM.

[Schmidt, 2000] Schmidt, A. (2000). Implicit Human Computer Interaction Through Context. *Personal and Ubiquitous Computing*, 4(2):191–199.

[Schmitz, 2010] Schmitz, M. (2010). *Tangible Interaction with Anthropomorphic Smart Objects in Instrumented Environments*. PhD thesis, Saarland University.

[Schmitz et al., 2008] Schmitz, M., Baus, J., and Dörr, R. (2008). The Digital Sommelier: Interacting with Intelligent Products. In *The Internet of Things: First International Conference, IOT'08*, Lecture Notes in Computer Science, pages 247–262. Springer Berlin / Heidelberg.

[Schmitz and Butz, 2006] Schmitz, M. and Butz, A. (2006). SAFIR: Low-Cost Spatial Audio for Instrumented Environments. In *Proceedings of the 2nd International Conference on Intelligent Environments*, pages 427–430, Athens, Greece.

[Schneider, 2010] Schneider, M. (2010). *Resource-Aware Plan Recognition in Instrumented Environments*. PhD thesis, Artificial Intelligence Group, Saarland University.

[Schöning et al., 2009] Schöning, J., Rohs, M., Kratz, S., Löchtefeld, M., and Krüger, A. (2009). Map Torchlight: A Mobile Augmented Reality Camera Projector Unit. In *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 3841–3846, New York, NY, USA. ACM.

[Schwartz et al., 2005] Schwartz, T., Brandherm, B., and Heckmann, D. (2005). Calculation of the User-Direction in an Always Best Positioned Mobile Localization System. In *International Workshop on Artificial Intelligence in Mobile Systems (AIMS)*, Salzburg, Austria.

[Scott, 2005] Scott, J. (2005). UbiComp: Becoming Superhuman. In Heidelberg, S. B. ., editor, *Proceedings of the UbiComp 2005 Workshop on UbiPhysics*, pages 1–3.

[Sequeira et al., 1999] Sequeira, V., Ng, K., Wolfart, E., Gonçalves, J., and Hogg, D. (1999). Automated Reconstruction of 3D Models from Real Environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:1–22.

[Siio et al., 2003] Siio, I., Rowan, J., Mima, N., and Mynatt, E. (2003). Digital Decor: Augmented Everyday Things. *Proceedings of Graphics Interface 2003*, pages 159–166.

[Sohn et al., 2005] Sohn, T., Li, K. A., Lee, G., Smith, I. E., Scott, J., and Griswold, W. G. (2005). Place-Its: A Study of Location-Based Reminders on Mobile Phones. In *Ubicomp*, pages 232–250, Tokyo, Japan.

[Spassova, 2004a] Spassova, L. (2004a). *Fluid Beam – Konzeption und Realisierung eines Display-Kontinuums mittels einer steuerbaren Projektor-Kamera-Einheit*. Diploma thesis, Artificial Intelligence Group, Saarland University.

[Spassova, 2004b] Spassova, L. (2004b). Fluid Beam - A Steerable Projector and Camera Unit. In *Proceedings of the Doctoral Colloquium at the 8th IEEE International Symposium on Wearable Computers*, pages 56–58, Arlington, USA.

[Spassova, 2007] Spassova, L. (2007). Interacting with Projected Displays by Means of a 3D Interface and an Instrumented Bracelet. In *IEEE Symposium on 3D User Interfaces*, Charlotte, NC, USA.

[Spassova and Guo, 2009] Spassova, L. and Guo, F. (2009). Concepts of Interaction with Projected Displays Using Gestures. In *European Conference on Ambient Intelligence (AmI-09)*, Salzburg, Austria.

[Spassova et al., 2009] Spassova, L., Schöning, J., Kahl, G., and Krüger, A. (2009). Innovative Retail Laboratory. In *European Conference on Ambient Intelligence (AmI-09)*, Salzburg, Austria.

[Spassova et al., 2005] Spassova, L., Wasinger, R., Baus, J., and Krüger, A. (2005). Product Associated Displays in a Shopping Scenario. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 210–211.

[Stahl, 2009] Stahl, C. (2009). *Spatial Modeling of Activity and User Assistance in Instrumented Environments*. PhD thesis, Artificial Intelligence Group, Saarland University, Saarbrücken.

[Stahl et al., 2011] Stahl, C., Frey, J., Alexandersson, J., and Brandherm, B. (2011). Synchronized Realities. *Journal of Ambient Intelligence and Smart Environments*, 3:13–25.

[Stahl and Haupert, 2006] Stahl, C. and Haupert, J. (2006). Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Environments. In *Proceedings of the International Workshop on Location- and Context-Awareness (LoCA)*, pages 74–85. Springer Berlin / Heidelberg.

[Stahl and Schwartz, 2010] Stahl, C. and Schwartz, T. (2010). Modeling and Simulating Assistive Environments in 3-D with the YAMAMOTO Toolkit. In *Proceedings of the 1st International Conference on Indoor Positioning and Indoor Navigation*, pages 1–6, Zürich, Switzerland. IEEE Xplore.

[Stock and Zancanaro, 2007] Stock, O. and Zancanaro, M. (2007). *PEACH - Intelligent Interfaces for Museum Visits (Cognitive Technologies)*. Springer Berlin / Heidelberg.

[Storz et al., 2006] Storz, O., Friday, A., Davies, N., Finney, J., Sas, C., and Sheridan, J. (2006). Public Ubiquitous Computing Systems: Lessons from the e-Campus Display Deployments. *IEEE Pervasive Computing*, 5(3):40–47.

[Streitz et al., 2002] Streitz, N., Prante, T., Müller-Tomfelde, C., Tandler, P., and Magerkurth, C. (2002). Roomware: The Second Generation. In *CHI '02: extended abstracts on Human factors in computing systems*, pages 506–507, New York, NY, USA. ACM.

[Sukaviriya et al., 2003] Sukaviriya, N., Podlaseck, M., Kjeldsen, R., Levas, A., Pingali, G., and Pinhanez, C. (2003). Embedding Interactions in a Retail Store Environment: The Design and Lessons Learned. In *Proceedings of IFIP Conference on Human-Computer Interaction - INTERACT*, pages 519–526.

[Sukthankar et al., 2001] Sukthankar, R., Stockton, R. G., and Mullin, M. D. (2001). Smarter Presentations: Exploiting Homography in Camera-Projector Systems. *IEEE International Conference on Computer Vision*, 1:247–253.

[Summet et al., 2004] Summet, J. W., Flagg, M., Ashdown, M., Sukthankar, R., Rehg, J. M., Abowd, G. D., and Cham, T. J. (2004). Robust Projected Displays for Ubiquitous Computing. In *Ubicomp Workshop on Ubiquitous Display Environments*.

[Surmann et al., 2003] Surmann, H., Nüchter, A., and Hertzberg, J. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45:181–198.

[Tokuda et al., 2003] Tokuda, Y., Iwasaki, S., Sato, Y., Nakanishi, Y., and Koike, H. (2003). Ubiquitous Display for Dynamically Changing Environments. In *CHI '03 extended abstracts on Human factors in computing systems*, CHI '03, pages 976–977, New York, NY, USA. ACM.

[Vertegaal and Poupyrev, 2008] Vertegaal, R. and Poupyrev, I. (2008). Organic User Interfaces: Introduction to Special Issue. In *Communications of the ACM*, volume 51, pages 26–30.

[Wahlster, 2002] Wahlster, W. (2002). SmartKom: Fusion and Fission of Speech, Gestures, and Facial Expressions. In *Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems*, pages 213–225, Kyoto, Japan.

[Wahlster, 2003] Wahlster, W. (2003). Towards Symmetric Multimodality: Fusion and Fission of Speech, Gesture and Facial Expression. In *In Proceedings of the 26th German Conference on Artificial Intelligence*, pages 1–18. Springer Berlin / Heidelberg.

[Wahlster et al., 2010] Wahlster, W., Feld, M., Gebhard, P., Heckmann, D., Jung, R., Kruppa, M., Schmitz, M., Spassova, L., and Wasinger, R. (2010). *The Shopping Experience of Tomorrow: Human-Centered and Resource-Adaptive*, pages 205–237. Cognitive Technologies. Springer Berlin / Heidelberg.

[Wallace et al., 2005] Wallace, G., Anshus, O. J., Bi, P., Chen, H., Chen, Y., Clark, D., Cook, P., Finkelstein, A., Funkhouser, T., Gupta, A., Hibbs, M., Li, K., Liu, Z., Samanta, R., Sukthankar, R., and Troyanskaya, O. (2005). Tools and Applications for Large-Scale Display Walls. *IEEE Computer Graphics and Applications*, 25(4):24–33.

[Wasinger, 2006] Wasinger, R. (2006). *Multimodal Interaction with Mobile Devices: Fusing a Broad Spectrum of Modality Combinations*. Akademische Verlagsgesellschaft Aka GmbH, Heidelberg.

[Wasinger et al., 2005] Wasinger, R., Krüger, A., and Jacobs, O. (2005). Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant. In *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive)*, pages 297–314, Munich, Germany.

[Wasinger and Wahlster, 2006] Wasinger, R. and Wahlster, W. (2006). The Anthropomorphized Product Shelf: Symmetric Multimodal Human-Environment Interaction. In Aarts, E. and Encarnação, J. L., editors, *True Visions: The Emergence of Ambient Intelligence*, pages 3–540. Springer Berlin / Heidelberg.

[Weiser, 1991] Weiser, M. (1991). The computer for the 21st century. *Scientific American Special Issue on Communications, Computers, and Networks*, 265(3):66–75.

[Weiss et al., 2010] Weiss, M., Voelker, S., Sutter, C., and Borchers, J. (2010). BendDesk: Dragging Across the Curve. In *ITS '10: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 1–10, New York, NY, USA. ACM.

[Yang et al., 1999] Yang, R., Brown, M. S., Seales, W. B., and Fuchs, H. (1999). Geometrically correct imagery for teleconferencing. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 179–186, New York, NY, USA. ACM.

[Yang et al., 2001a] Yang, R., Gotz, D., Hensley, J., Towles, H., and Brown, M. S. (2001a). PixelFlex: A Reconfigurable Multi-Projector Display System. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 167–174, Washington, DC, USA. IEEE Computer Society.

[Yang et al., 2001b] Yang, R., Yang, R., and Welch, G. (2001b). Automatic Projector Display Surface Estimation Using Every-Day Imagery. *9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*.

[Yee, 2003] Yee, K.-P. (2003). Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–8, New York, NY, USA. ACM.

[Zhang, 1999] Zhang, Z. (1999). Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Proceedings of the International Conference on Computer Vision*, pages 666–673.

[Zhang, 2000] Zhang, Z. (2000). A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334.