Linköping Studies in Science and Technology

Dissertations, No. 1145

Bringing Augmented Reality to Mobile Phones

Anders Henrysson

Department of Science and Technology
Linköpings universitet

Norrköping 2007

Bringing Augmented Reality to Mobile Phones

Anders Henrysson

Printed by LiU-Tryck, Linköping 2007

ISBN 978-91-85895-43-4 ISSN 0345-7524

Rationality is the recognition of the fact that nothing can alter the truth and nothing can take precedence over that act of perceiving it

Ayn Rand

When you make the finding yourself - even if you're the last person on Earth to see the light - you'll never forget it

Carl Sagan

Abstract

With its mixing of real and virtual, Augmented Reality (AR) is a technology that has attracted lots of attention from the science community and is seen as a perfect way to visualize context-related information. Computer generated graphics is presented to the user overlaid and registered with the real world and hence augmenting it. Promising intelligence amplification and higher productivity, AR has been intensively researched over several decades but has yet to reach a broad audience.

This thesis presents efforts in bringing Augmented Reality to mobile phones and thus to the general public. Implementing technologies on limited devices, such as mobile phones, poses a number of challenges that differ from traditional research directions. These include: limited computational resources with little or no possibility to upgrade or add hardware, limited input and output capabilities for interactive 3D graphics. The research presented in this thesis addresses these challenges and makes contributions in the following areas:

Mobile Phone Computer Vision-Based Tracking
The first contribution of thesis has been to migrate computer vision algorithms for tracking the mobile phone camera in a real world reference frame - a key enabling technology for AR. To tackle performance issues, low-level optimized code, using fixed-point algorithms, has been developed.

Mobile Phone 3D Interaction Techniques
Another contribution of this thesis has been to research interaction techniques for manipulating virtual content. This is in part realized by exploiting camera tracking for position-controlled interaction where motion of the device is used as input. Gesture input, made possible by a separate front camera, is another approach that is investigated. The obtained results are not unique to AR and could also be applicable to general mobile 3D graphics.

Novel Single User AR Applications
With short range communication technologies, mobile phones can exchange data not only with other phones but also with an intelligent environment. Data can be obtained for tracking or visualization; displays can be used to render graphics with the tracked mobile phone acting as an interaction device. Work is presented where a mobile phone harvests a sensor-network to use AR to visualize live data in context.

Novel Collaboration AR Applications

One of the most promising areas for mobile phone based AR is enhancing face-to-face computer supported cooperative work. This is because the AR display permits non-verbal cues to be used to a larger extent. In this thesis, face-to-face collaboration has been researched to examine whether AR increases awareness of collaboration partners even on small devices such as mobile phones. User feedback indicates that this is the case, confirming the hypothesis that mobile phones are increasingly able to deliver an AR experience to a large audience.

Acknowledgements

My first thanks go to my friend and collaborator Mark Billinghurst for his great efforts to inspire, enrich and promote my research. Great thanks also to my supervisors Anders Ynnerman and Mark Ollila for their guidance throughout these years.

Matthew Cooper, Morten Fjeld and Nguyen-Thong Dang for their much appreciated feedback on this thesis. Karljohan Palmerius for his help on LaTeX. Friends and colleagues at NVIS and HITLabNZ.

My financiers, Brains & Bricks and CUGS, for supporting my research and travels.

This research work was funded in part by CUGS (the National Graduate School in Computer Science, Sweden).

Contents

Chapter 1

Introduction

Augmented Reality (AR) is a grand vision where the digital domain blends with the physical world. Information not only follows a person, but also her very gaze: looking at an object is enough to retrieve and display relevant information, amplifying her intelligence. Though research on AR has advanced over the last several decades, AR technology has yet to reach the mass-market. The minimum requirement for AR is a display, a camera for tracking, and a processing unit. These are also the components of camera phones, predicted to account for more than 80% of total worldwide mobile phone sales by 2010[1].

Mobile phones, which were not long ago "brick-like" devices limited to phone calls, have evolved into digital "Swiss Army knives" and reached sales of more than one billion per year[2]. Web browsing, multimedia playback and digital photography are only some of their capabilities; with increasing storage, communication and computational resources, their versatility and importance will continue to grow. Realizing AR on mobile phones would make this technology available to millions of users and, in addition, provide a rapidly developing research platform.

This thesis studies AR on mobile phones, addressing some of the technical obstacles that must be overcome before mobile AR becomes commonplace. The research arises from the motivation that this range of devices is now increasingly capable of AR and is likely to become the dominant AR platform in the future. Research on AR intersects with mobile and Ubiquitous Computing in general and 3D interaction in particular. Opportunities in these areas were addressed as the research on AR progressed.

The remainder of this chapter introduces technologies and concepts upon which this thesis is based. Next, current mobile technology is surveyed to illustrate the versatility of modern mobile phones and to point out relevant trends. One such trend is positioning which, combined with orientation-sensing, enables AR. Another trend is short-range data communication, which enables mobile units to seamlessly connect to each other and with embedded devices. This actualizes the concept of Ubiquitous Computing, where an intelligent environment provides system input. AR fundamentals are then presented, followed by a brief introduction to the 3D input con-

[1] $www.gartner.com/it/page.jsp?id = 498310$

[2] $www.strategyanalytics.net/default.aspx?mod = PressReleaseViewer\&a0 = 3260$

trol terminology used later. The chapter finishes off with research challenges and contributions of this thesis. Chapter 2 then presents research threads joined in the contributions chapter.

1.1 Mobile and Ubiquitous Computing

Increasing battery power combined with decreasing power consumption and other advances in electronics design has resulted in a wide range of mobile computing devices. Laptops have been complemented with Tablet PCs, PDAs and Ultra Mobile PCs. Parallel to this untethering of computing resources, mobile phones have developed into versatile tools for mobile computing and communication as illustrated in Figure 1.1. There has been much progress in areas important for realizing AR on mobile phones:

Processing
Mobile phones now have sufficient processing power for simple computer vision, video decoding and interactive 3D graphics. Also featuring color displays[3] and ubiquitous network access[4], handsets are increasingly capable of streaming video, web browsing, gaming, and other graphic and bandwidth intensive applications until recently only found on stationary computers with wired connections. Many device manufacturers[5] are also fitting graphics processing units (GPUs) into mobile phones, providing faster graphics and hardware floating-point support.

Imaging
The late 1990s saw the first demonstration of a mobile phone camera. Since then, more than one billion camera phones have been sold and progress toward higher resolutions and better optics has been fast. Camera phones are also capable of video[6], using either the back camera for recording or the front camera for video phone calls. The tight coupling of camera and CPU gives mobile phones unique input capabilities where real-time computer vision is used to enable new interaction metaphors and link physical and virtual worlds.

Positioning
It is not only image sensors that have made their way into mobile phones. Many handsets are now equipped with GPS antennas to establish their location in global coordinates, enabling location-based services which provide specific information based on user location. Such services include finding nearby resources in unfamiliar environments and tracking objects, for example cars. Entertainment is another area for location-aware systems with Pervasive gaming - also known as location-based gaming - being a new breed of computer games which use the physical world as a playing field and therefore depend on positioning technology. Game scenarios include mobile players on street level, equipped with handheld devices positioned with GPS, and online players seeing the street players as avatars in a virtual world. To obtain more accurate positioning, and

[3]Display color depth often range from 16 to 24 bits per pixel at e.g. QVGA (320×240) resolutions
[4]WCDMA at 384 Kbps is common and emerging HSDPA currently supports up to 7.2 Mbps
[5]For a list of graphics-accelerated mobile devices, see for example: $mobile.sdsc.edu/devices.html$
[6]Typical resolutions range from QCIF (176×144) to VGA(640×480) at frame rates from 15 to 30 fps

Figure 1.1: Phone evolution. The left phone is a Nokia 6210 announced in 2000. It has a monochrome display that renders 6 lines of characters. The right phone is a Nokia N95 8GB announced in 2007. It features a 2.8" TFT display with 16 million colors. It also has hardware 3D graphics acceleration and GPS positioning. On its back is a 5 megapixel camera and a second camera is located on its front. (Photograph courtesy of Nokia)

also indoor gaming where GPS signals are blocked, radio beacons such as WLAN can be used. Positioning a user in a coordinate system makes it possible to identify the close vicinity and provide related information on a 2D map. Adding head orientation to position makes it possible to identify what the user is looking at and display information in 3D.

Interface

Mobile phones interfaces have evolved with their increasing functionalities. Early handsets - limited to making phone calls - featured a character-based user interface, only requiring number keys for input. As graphical user interfaces (GUI) became the norm due to the increase in processor speeds, availability of color raster screens, and the success of the GUI paradigm on PCs; 5-way joypads, joysticks and jogdials were introduced along with menu buttons. These additions enabled fast menu navigation and icon selection, necessary for GUI interaction. High-end smartphones adopted stylus interaction and miniature QWERTY keypads, though still supporting one-handed finger interaction - contrasting with PDAs' inherently bimanual interaction style. A stylus has the advantage of being handled with great precision, due to its pen and paper metaphor and small contact surface with the screen, but it is limited to one contact point. In contrast, some recent touch screens devices, for example the Apple iPhone, allow multiple-finger gestures; zooming is made by pinch gestures: *pinch open* to zoom in and *pinch close* to zoom out. Camera phones often feature a dedicated camera button for taking photographs, and many modern multimedia phones have media buttons such as `play` , `next` etc. However, there has been little or no development of mobile phone input dedicated to 3D interaction despite increasing 3D rendering capabilities on modern handsets.

Short-range Communication
Wireless networking is not only available via wide area cellular systems, but also via short-range communication standards such as Bluetooth and WLAN. These technologies are interesting because they enable data exchange with devices for, for example, tracking, context information, media output or database access. Computationally heavy tasks may seamlessly be distributed to surrounding computing resources. Devices scan the proximity for services and establish an ad-hoc communication channel with minimal configuration requirements. This is important when the user is mobile and new digital contexts must be mapped. Such service discovery and inexpensive short range communication are also of importance in Ubiquitous Computing.

Ubiquitous Computing
Ubiquitous Computing is a paradigm where computing is embedded in our environment, hence becoming invisible [Wei91]. It represents the third wave in computing, the first being mainframes (one computer serving many people) and the second personal computers (one computer serving one person). With many small computers - some being mobile - serving one person, one vision is to build intelligence into everyday objects. A fundamental property of intelligent objects is that they are able to sense and output relevant state information; hence, sensors and wireless communication constitute important technologies. In an intelligent environment, a mobile phone can seamlessly connect to embedded devices that provide services. Xerox Parc's Ubicomp project[7] included development of inch-scale tabs, foot-scale pads and yard-scale boards - devices working together in an infrastructure that recognized device name, location, usage, and ownership of each device. It is easy to see the parallels with today's inch-scale mobile phones and yard-scale interactive plasma screens. The concepts of a sensing environment and of connecting different-scale devices are once again becoming interesting as mobile phones obtain increased capabilities to communicate data.

1.2 Augmented Reality

In ubiquitous computing, the computer became "invisible". In AR, the computer is transparent and the user perceives the world *through* the computer. This means that a computer can mix impressions of the real world with computer generated information, in this way augmenting reality. The world being both three dimensional and interactive, requires an AR system to have the following three characteristics [Azu97]:

1. Combines real and virtual
2. Interactive[8] in real-time
3. Registered in 3D

[7] $www.ubiq.com/weiser/testbeddevices.htm$

[8] A frame rate of 5 fps is minimum for tolerable interactivity while 30 is minimum for smooth animation. See Tang and Isaac *Why Do Users like Video*

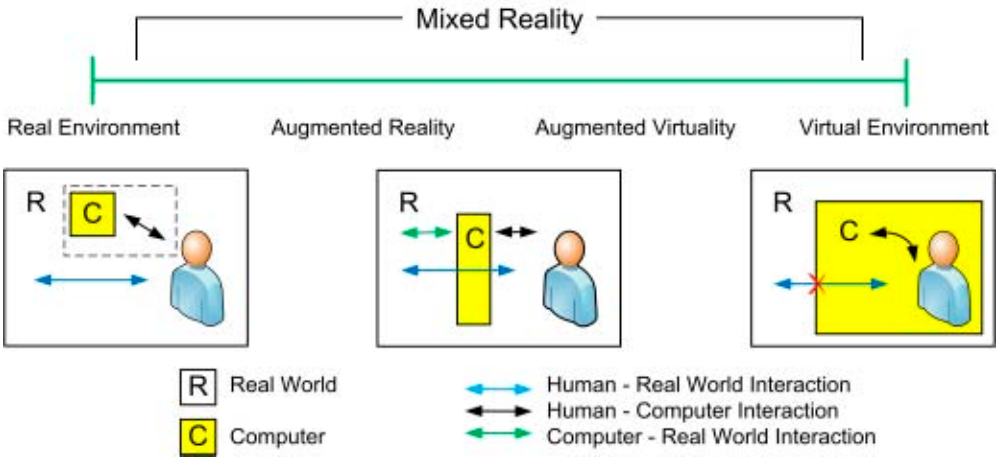

Figure 1.2: Milgram's Reality-Virtuality continuum and corresponding interaction styles. The upper part depicts Milgram's continuum, which range from the real (i.e. physical) environment to immersive virtual environments. Between these extremes, there is a mix between real and virtual, hence the term Mixed Reality. The lower part illustrates corresponding interaction styles. Interaction in a real environment requires a user to switch focus between computer (dashed box) and physical environment, whereas Mixed Reality interaction superimposes these domains. A Virtual Environment permits no real world interaction. (Adapted from [MK94] and [RN95])

The concept of AR applies to all senses but this thesis focuses on visual enhancements. This means that AR systems overlay the users' view of the real world with real-time 3D graphics. Change in view direction is immediately reflected by re-rendering of the virtual scene to preserve spatial relationships between real and virtual objects. In this way, virtual imagery can seem attached to real world objects.

It is illustrative to compare AR with Virtual Reality (VR) - where only virtual information is presented. While ubiquitous computing was intended to be the absolute opposite of VR, AR has a closer relationship to VR since sensory impressions are partially virtual. Milgram's continuum [MK94] (Figure 1.2) shows this relationship: the further to the right, the less real world information is perceivable. The middle ground between real and virtual environments is called Mixed Reality, which also includes Augmented Virtuality where most of the input, often the background, is computer-generated. Milgram's continuum highlights another AR advantage: There is no need to make an expensive digital version of a real world scene when visualizing new objects in an existing environment.

In many movies, part of the content is computer-generated to produce scenes that cannot be created with physical props. It is of crucial importance to register these objects in 3D so as to preserve the illusion of virtual objects existing in the physical world. However, in movies there is no requirement for interactive rendering. Considering AR as real-time movie effects gives a hint of the research problems but also some of its potential.

What makes researchers interested in AR is its overlay of the real world with context-related information, resulting in intelligence amplification [Azu97]. This can be further described as projecting the visualization domain onto the task domain, thus eliminating domain switching where reference points in the visualization domain must be matched to corresponding points in the task domain - a sometimes time-consuming task. Such elimination of spatial seams is believed to yield higher productivity[9]. For example, a physician could project an X-ray image (visualization domain) onto the patient (task domain) to avoid having to memorize the X-ray while looking at the patient and having to switch back to the X-ray to refresh memory or learn new details. This generalizes to most situations where context related visual information is used to perform real world tasks.

Analogous to task and visualization domains, in collaborative work one can speak of task and communication spaces. In traditional face-to-face collaboration, participants surround a table on which the object of interest, e.g. a product prototype, is placed. People can see each other and use expressive non-verbal communication cues - task and communication spaces coincide. If the object of interest is instead a CAD-model displayed on a vertical screen, participants can no longer perceive visual communication cues such as gaze or gesture while sitting side-by-side observing the screen - task and communication spaces do not coincide. With AR it is possible to get the best of both worlds: coinciding task and communication spaces and digital content. These features make AR interesting for computer supported cooperative work (CSCW).

There is a wide range of application areas for AR. Besides medicine [BFO92], palaeontology [BGW+02] and maintenance [FMS93], Augmented Reality gaming [TCD+00, CWG+03] is a logical extension to pervasive gaming, but perhaps more similar to first person computer games, now taking place in the real world. AR may also extend location-based services by having buildings highlighted in 3D instead of merely displaying a legend on a 2D map [FMH97]. This requires the location-specific information to be a graphical object with a certain position, and the object to be augmented to be in the line of sight.

There are three key technologies upon which an AR system is built:

1. **Tracking.** The system must know the user's viewpoint to retrieve and present related virtual content. More precisely, it must know the position and orientation of the system display in a physical coordinate system with known mapping to a virtual one. The establishment of position and orientation parameters is known as tracking.

2. **Registration.** Tracking is only a means to achieve registration - the final alignment of real and virtual information that is presented to the user. Registration must be made with pixel accuracy at interactive frame rates to preserve the illusion of real and virtual coexisting in the same domain.

3. **Display.** An AR system must be able to output a mix of real and virtual. The display must hence allow the user to see the real world overlaid with 3D graphics. It should also be trackable at interactive frame rates.

[9]This is one of the motivations behind Industrial Augmented Reality (IAR). See for example $www.arvika.de$

Type	Technology	Example
Mechanical	Armature	SensAble Phantom®
Source-based	Magnetic, ultrasonic	Polhemus FASTRAK®
Source-less	Inertial: gyroscope, accelerometer	InterSense InertiaCube™
Optical	Fiducial markers, natural feature tracking	A.R.T. ARTtrack, ARToolKit
Hybrid	e.g. Optical-Inertial	InterSense IS-1200 VisTracker™

Table 1.1: Tracking technology examples

1.2.1 Tracking

A mobile see-through display[10] must be tracked with 6 degrees of freedom (6DOF) in order to display the correct image overlay. Tracking is a general problem with application not only in AR but also in VR and robotics; it will therefore be treated briefly here, though tracking is significantly more difficult in AR due to registration requirements. Registration is directly dependent on tracking accuracy, making AR tracking a very demanding task. Ideally, the resolution of both the tracking sub-system and the display should be that of the fovea of the human eye.

There are two main tracking strategies: egocentric *inside-out* and exocentric *outside-in*. In inside-out tracking, the AR system is equipped with enough sensors to establish position and orientation parameters. Outside-in tracking takes place in an instrumented environment where fixed sensors track the mobile AR system from the outside and supply it with tracking data for registration.

In broad terms, tracking technology can be divided into mechanical, source-based, source-less and optical (Table 1.1). Mechanical tracking systems calculate the final orientation and position by traversing armature limbs, accumulating the relative position of each link. They are accurate but cumbersome and with limited motion range.

The principle behind source-based tracking is to measure the distance between sources and receivers. To achieve 6DOF tracking, three sources and three receivers must be used. Electromagnetic trackers emit a magnetic field while acoustic trackers transmit ultrasonic signals picked up by microphones. Both these trackers are unobtrusive and industry-proven but can only be used in controlled environments. GPS and WLAN sources can be used for positioning only.

Inertial trackers are source-less devices that measure change in inertia. Accelerometers measure acceleration and yield position data when their output is integrated twice. Gyroscopes measure angular motion and work by sensing the change in direction of an angular momentum. Gyroscopes require calibration while accelerometers use dead reckoning and are hence prone to drift over time. The big advantage is that both technologies can be miniaturized and also deployed in unprepared environments. Compasses give absolute heading relative to the Earth's magnetic field; however, they are vulnerable to distortion.

Optical tracking is based on analysis of video input and calculates either absolute camera pose relative to known geometries or camera motion relative to the previous frame from extracted

[10] In video see-through displays, it is often the camera that is tracked and assumed to be close enough to the display for the offset to be ignored

features. These geometries can either be 3D objects or 2D markers called fiducials. Alternatively, one can speak of marker-less and marker-based tracking. Optical tracking requires a clear line of sight and computer vision algorithms are computationally heavy. There are however several strengths: It is cheap, accurate and flexible and one single sensor provides 6DOF tracking. It is very well suited for video see-through displays presented in the next section. Cameras are readily available in mobile phones where coupled with a CPU for possible image analysis. It should be noted that optical tracking extends to non-visible wavelengths e.g. infra red. Very accurate range measurements can also be obtained by another form of optical tracking: laser beams.

The choice of tracker is a trade-off between mobility, tracking range, system complexity etc. Most setups requiring wide area tracking use hybrid approaches, combining different trackers respective strengths. This is the case for most outdoor AR configurations, to be described in Section 2.1.2. For a more in-depth treatment of tracking technologies, please refer to the survey by Rolland et al. [RDB01].

1.2.2 Displays

There are three display categories that are used for superimposing computer graphics onto a view of the real world: optical see-through, video see-through and projection-based.

Optical see-through displays are partially transparent and consist of an optical combiner to mix real and virtual. The main advantage is that the user sees the real world directly; however, having to reflect projected graphics, these displays reduce the amount of incoming light. Other problems stem from having different focal planes for real and virtual and the lag of rendered graphics.

Video see-through displays consist of an opaque screen aligned with a video camera. By displaying the camera images, the display becomes "transparent". The advantages are that the computer process the same image as the user sees (also introducing a delay in the video of the real world to match the tracker delay) and both real and virtual information are displayed without loss of light intensity. Applying image analysis to the captured video frames makes it possible to achieve correct registration even if tracking data is noisy. The downside is that eye and video camera parameters differ.

Projection-based systems use the real world as display by projecting graphics onto it. This makes them good at providing a large field of view. They place the graphics at the same distance as the real world object making eye accommodation easier. The big drawback is that they require a background to project graphics onto; hence, an object can only be augmented within its contours and might require special surface and lighting conditions to provide bright virtual information. Most projectors can however only focus the image on a single plane in space. This limits the range of objects that can be augmented.

There are three main configuration strategies for the above display categories: head-worn, handheld and stationary (Figure 1.3).

A head-mounted display (HMD) is worn as a pair of "glasses". This enables bimanual interaction since both users' hands are free; for many industrial and military applications, this property makes HMDs the only alternative. In mobile/wearable applications they are the dominant display type. Though there are technical challenges remaining for both optical see-through

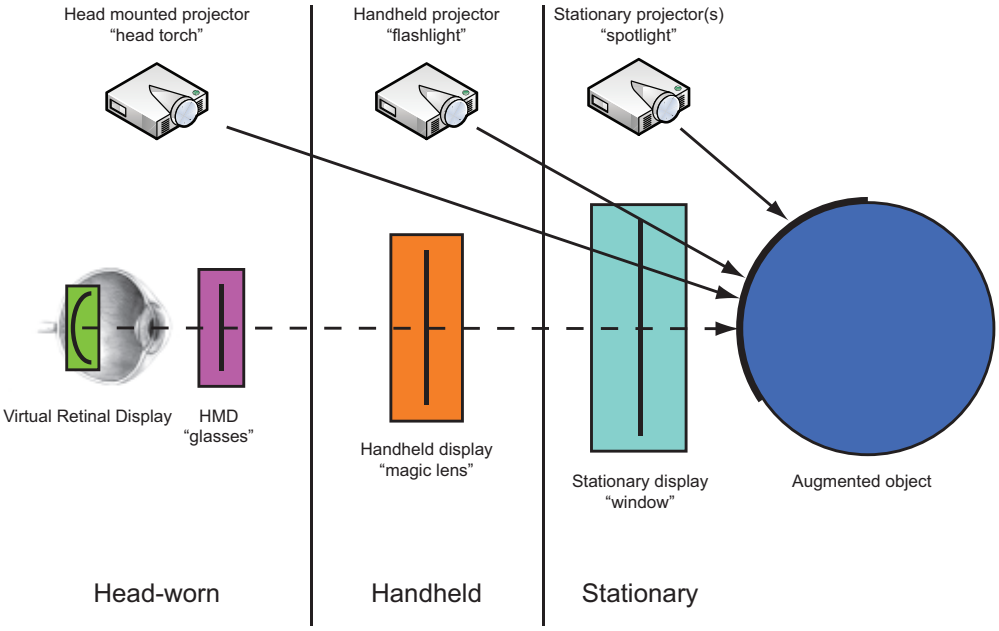

Figure 1.3: Display configurations and metaphors. An image plane with virtual imagery can be generated directly on the retina by a laser beam. At the other end of the scale, the image plane coincides with the augmented object. Between these extremes, a display is needed for the image plane. Such a display can be either optical or video see-through and be head-worn, handheld or stationary depending on tracking area, interaction requirements etc. (Adapted from image courtesy of Oliver Bimber and Ramesh Raskar)

and video see-through HMDs, their main problems are social and commercial. Wearing a salient HMD in public is not socially accepted and most systems are cumbersome and expensive. Consumer HMDs have been developed primarily for VR and personal big-screen television, but have failed to become popular in the mass market. It remains to be seen if increasing mobile storage and video playback capabilities will spur a new wave of affordable HMDs. Even if such display were commonly available, their usage for AR would be limited since they are not see-through and/or lack tracking capability. Fitting an HMD with calibrated cameras and tracking sub-systems is a daunting task. Head-mounted projectors are still at an experimental level and it remains to be seen if they will become competitive.

A handheld display is used as a magic lens [BSP+93] or a "looking glass", magnifying information content. As such, they have limited field of view and currently there is no support for stereo, resulting in less depth cues. Despite these drawbacks, handheld displays have emerged as an alternative to head-worn displays. The main reason is that widely available handheld devices have become powerful enough for AR. Using a mobile phone for handheld AR is similar to using

it for photography - a socially accepted activity. Handheld AR is a complement, rather than a replacement, to wearable configurations with HMDs. It is reasonable to believe that AR will develop in a manner analogous to VR, where immersive HMD and CAVE configurations used in industry and academia have been complemented by non-immersive VR experiences such as 3D gaming and Second Life, running on consumer hardware. Miniature projection systems is an emerging technology with possible impact on handheld AR. The idea is to embed a small, but powerful, projector inside a handheld device. If used for AR, such a device would be used as a flashlight "illuminating" the real world with digital information.

Stationary displays act as "windows" facing the augmented world. Since they are stationary, no tracking of the display itself is required. Stationary displays range from PC monitors to advanced 3D spatial displays. A simple approach to AR is to connect a web-cam to a PC and augment the video stream. Projectors can be used to augment a large area without requiring the users to wear any equipment. The problem of only having one focal plane can be remedied with multi-projector techniques [BE06].

1.3 3D Input

Since the real world is three dimensional, AR is inherently 3D in its mixing of real and virtual imagery. This means that AR must support 3D interaction techniques for users to be able to interact beyond merely moving the tracked display. To enable both translation and rotation in 3D, an input device must support at least six degrees of freedom (6DOF) interaction[11].

Human interaction capabilities and limitations have been studied for decades and a vast body of knowledge is scattered across domains such as human motor control, experimental psychology, and human computer interaction (HCI). However, all aspects of 6DOF interaction are not fully understood and there is no existing input device that suits all 3D applications. Different input devices have therefore been proposed for different 3D interaction tasks and user environments.

Input devices can be categorized according to different criteria. One important such criterion is the resistance an input device exercise on an operator's motions when handling the device. Resistance ranges from zero or constant resistance *isotonic*[12] devices to infinite resistance *isometric*[13] devices. Between these extremes there exist devices whose resistance depends on displacement (elastic devices), velocity (viscous devices) and acceleration (inertial devices). In reality it is common to regard it to be a continuum of elasticity ranging from mostly isotonic (e.g. the computer mouse) to mostly isometric (e.g. IBM's TrackPoint). In this thesis the binary 5-way re-centering joystick/joypad common on mobile phones will be labeled isometric, despite not being perfectly isometric. Though their mass makes the mobile phone and all other free-moving devices inertial to some extent, it will be considered isotonic when tracked in 3D.

Another criterion is the transfer function (TF) relating the force applied to a control device to the perceived system output. One important characteristic of this TF that maps human input

[11] For a more complete treatment of this subject, please refer to Zhai [Zha95] and Bowman et al. [BKLP04]

[12] From Greek *isos* and *tonikos* = constant tension

[13] From Greek *isos* and *metron* = constant length, non-moving

Task	Description	Real-world counterpart	Parameters
Selection	Acquiring or identifying a particular object from the entire set of objects available	Picking an object with a hand	Distance and direction to target, target size, density of objects around the target, number of targets to be selected, target occlusion
Positioning	Changing the 3D position of an object	Moving an object from a starting location to a target location	Distance and direction to initial position, distance and direction to target position, translation distance, required precision of positioning
Rotation	Changing the orientation of an object	Rotating an object from a starting orientation to a target orientation	Distance to target, initial orientation, final orientation, amount of rotation, required precision of rotation

Table 1.2: Canonical manipulation tasks for evaluating interaction techniques. From [BKLP04].

to object transformation is its order. A zero order, i.e. constant, TF maps device movement to object movement and the control mechanism is therefore called *position control*. A first order TF maps human input to the velocity of object movement and the control mechanism is called *rate control*. Higher order TFs are possible but have been found inferior. For example, the computer mouse uses position control while most joysticks use rate control. If the mapping is 1-to-1 between device movement and object movement, interaction is said to be isomorphic[14]. Isomorphic interaction is a direct form of manipulation where atomic actions, such as object translation and rotation, are identical to motion of human limbs controlling the input device. This means that both voluntary and involuntary device movements are mapped to object movements. A computer mouse is often isomorphic having a 1-to-1 mapping between hand movement and mouse pointer displacement. While being considered intuitive, isomorphic interaction is limited by human joint constraints.

Though any combination of resistance (isotonic and isometric) and transfer function (position control and rate control) is possible, two combinations have proven to be superior for general input: isotonic position control and isometric rate control. Rate control and position control are optimal for different tasks and ideally an input device supports both. Workspace size is one factor that is of importance when choosing control type. Since position control is limited by human joint constraints - especially in the isomorphic case - a large workspace might favor rate

[14]From Greek *isos* and *morphe* = same shape

control.

A 3D input device supports a set of 3D interaction techniques. The effectiveness of these techniques depends on the manipulation tasks performed by the user. Since it is not feasible to evaluate interaction techniques for every conceivable task, a representative subset of possible manipulation tasks is often chosen. A task subset can be either general or application-specific. In this thesis, a canonical set of basic manipulation tasks is used to evaluate 3D interaction techniques developed for mobile phone AR. These are summarized in Table 1.2.

An interesting approach to both 3D input in AR, Tabletop[15], and to some extent Ubiquitous computing interaction, is the tangible user interface (TUI). Instead of relying on a dedicated input device, a TUI consists of physical objects acting as widgets which can be manipulated, arranged spatially etc. to provide system input. This gives a tighter coupling between device and function and allows the use of size, shape, relative position etc. to increase functionality. An AR user sees both real and virtual and can therefore manipulate real world objects. Arranging tracked objects on a horizontal display affords persistent, multi-user interaction. Tracked in 3D, TUI components provide isotonic 6DOF input. With embedded sensors, communication and computing capabilities, TUI components merge with Ubiquitous computing devices and provide an interface for the vanishing computer.

1.4 Research Challenges

The AR Grand Challenge is tracking. Without it, registration of real and virtual is not possible. The pursuit of tracking has led to construction of setups with high complexity, in turn making it harder to introduce AR technology to a broader audience. The lack of accessible AR technology means that researchers know very little about social issues and what real users demand.

The main challenge addressed in this thesis is to bring AR to one of the most widespread and fast-evolving family of devices: mobile phones. This imposes a set of challenges:

- A tracking solution needs to be created for this considerably restricted hardware platform. Such a solution must be able to track the mobile phone with 6DOF, at interactive frame rates, with sufficient stability between frames, and with a range large enough to allow the user to move the device to interact. Tracking often utilizes high-end sensors and/or requires significant computational resources. Due to the limitations of current mobile phones, it would not make sense to develop new tracking algorithms on this platform. Instead, existing algorithms must be ported and made more efficient. This has also been the strategy for OpenGL ES, where a desktop API has been reduced, ported and optimized.

- Interaction needs to be addressed in order to take AR beyond simple viewing. The phone form factor and input capabilities afford unique interaction styles, different from previous work in AR interaction; hence, new AR interaction techniques must be developed and evaluated in this context. The limited input capabilities offered by the mobile phone's keypad must be worked around or extended to provide the required means for 3D interaction. The

[15] Refers to horizontal displays e.g. MERL DiamondTouch™ and Microsoft Surface™.

ability to use the phone motion as input, in particular, must be explored. Solving these challenges also creates new opportunities in mobile HCI since 6DOF tracking technology has not been available on mobile phones before and 3D interaction on these devices has not been formally studied.

- The main advantage of AR is the reduction of cognitive seams due to superimposed information domains. This results in, for example, increased awareness in face-to-face collaboration. Such advantages of AR over non-AR, proven for other platforms, must be confirmed or dismissed in the mobile phone environment for it to be meaningful to use it. Experiments must be designed for exploring this. This is important for motivating further research on mobile phone AR.

- Proof-of-concept applications must be developed to demonstrate the feasibility of mobile phone AR, also exploring new opportunities not easily pursued with other platforms, but opened up by this configuration; among the mobile phone's interesting features for this are its tangibility, multimodal display and short range connectivity. Of special interest is to explore how to interact with intelligent environments and also how to remedy the phone's limited output capabilities. In this respect, this challenge is without a limited scope and will rather serve to exemplify advantages.

One challenge that should be acknowledged to be as great as that of tracking is content creation. Since AR virtual content is context dependent, detailed geospatial knowledge about the user's physical environment is needed to design content that is registered with the real world. However, this challenge is not device dependent and research on it not included in this thesis.

1.5 Contributions

This thesis presents the first migration of AR technology to the mobile phone and a body of knowledge drawn from subsequent research conducted on this platform. The individual contributions are presented as papers appended to the thesis. The main contributions of each paper are as follows:

Paper I introduces the platform used in this thesis along with the first collaborative AR application developed for mobile phones. Results are obtained from user studies investigating awareness and feedback. Also design guidelines for collaborative AR applications are provided

Paper II presents the first formal evaluation of 3D object manipulation conducted on a handheld device. Several atomic interaction techniques are implemented and compared in user studies

Paper III extends prior interaction research by adding various gesture-based interaction techniques along with isotonic rate control. In the user studies, impact on performance from task dimensionality was researched

Paper IV further explores 3D interaction by focusing on scene assembly. Two strategies for 6DOF interaction are demonstrated

Paper V presents the first mesh editing application for mobile phones, for which local selection techniques were refined

Paper VI applies lessons learned in the above papers on interaction to collaborative AR. A platform based on a shared scene graph is presented, also demonstrating phones coexisting with large screens in a collaborative setup

Paper VII explores browsing reality with an AR enabled mobile phone as interface to sensor networks. An inspection tool for humidity data was developed and evaluated

Paper VIII marries 2D exocentric tracking with 3D egocentric tracking to provide a framework for combining three information spaces and provide near seamless transition between them using motion-based input

The author of this thesis is the first author and main contributor to papers **I**-**V**, main contributor of concepts to papers **VI** and **VII**, and joint contributor to paper **VIII**. Chapter 3 is written to reflect the author's contributions.

Chapter 2

Towards Mobile Phone Augmented Reality

This chapter describes two research paths leading to the realization of AR on mobile phones. The first section follows AR from early research configurations to recent handheld AR on consumer devices. The second section follows the development of camera-based input on handheld devices. Last, an introduction to 3D interaction design will be given.

2.1 Approaches to Augmented Reality

The first steps towards AR were taken in the late 60's when Sutherland and his colleagues constructed the first see-through HMD [Sut68] which mixed a view of the real world with computer generated images. Such displays were used during following decades in research on helmet-mounted displays in aircraft cockpits e.g. in the US Air Force's Super Cockpit program[1] organized by Furness III, where fighter pilot's views were augmented.

When portable displays became commercially available a couple of decades after Sutherland's experiments, many researchers began to look at AR and researched how it could be realized. This section takes an odyssey through AR history and presents selected efforts in areas fundamental to AR and of importance to the contributions of this thesis. First, fundamental research on AR techniques will be presented. These early works were, with few exceptions, based on HMDs and made important contributions to solve challenges in tracking, registration, interaction, and collaboration; at the same time demonstrating advantages of AR for a range of application areas. Technical directions included fusion of data from multiple trackers, i.e. hybrid tracking, and adoption and refinement of computer vision techniques for establishing camera pose relative features.

One goal has always been for the user to roam freely and benefit from AR in any conceivable situation. Next, how HMDs were connected to wearable computers to realize outdoor AR will be

[1] $www.hitl.washington.edu/people/tfurness/supercockpit.html$

presented. The challenges addressed include wearable computing and wide area tracking. Since earlier works on optical tracking are less applicable in an unknown environment, much focus has been on hybrid solutions, often including GPS for positioning. Such work on enabling and exploring outdoor AR is important because the need for superimposed information is bigger in unfamiliar and dynamic environments.

If outdoor AR set out on a top-down quest for the ultimate AR configuration, handheld AR - rounding off this section - embarked on a bottom-up track where off-the-shelf devices were exploited for visual overlays. Earlier sections have given accounts of most of the technical challenges of and motivations for this endeavor. The research directions have primarily been to provide optical tracking solutions to commonly available handheld devices, requiring little or no calibration, and with built-in cameras: that is with no additional hardware being required. These efforts constitute the foundation for this thesis.

2.1.1 HMD-based AR

The term Augmented Reality was coined[2] in the early 90's by Caudell and Mizell [CM92], two researchers at Boeing. They developed a setup with an optical see-through HMD, tracked with 6DOF, to assist workers in airplane manufacturing by displaying instructions on where to drill holes and run wires. This was feasible because the overlays needed only simple graphics like wireframes or text. A similar HMD-based approach was taken by Feiner and his colleagues with KARMA [FMS93], where maintenance of a laser printer was assisted by overlaid graphics generated by a rule-based system.

It was apparent from these and other results from the same time that registration was a fundamental problem. Azuma [AB94] contributed to reducing both static[3] and dynamic[4] registration error for see-through HMD AR. With a custom optoelectronic head tracker, combined with a calibration process, static registration was significantly improved. Prediction of head movements were made to reduce dynamic errors resulting from the fact that an optical see-through HMD can not delay real world information to compensate for tracking and rendering latency.

Augmenting objects with instructions is only one application area for AR. Another heralded capability of AR is its ability to give a user "X-ray vision" by visualizing otherwise hidden objects. This is of great importance in medical surgery where the incision should be kept as small as possible. With AR, a surgeon can see directly into the body. An early example of this was provided by Bajura et al. [BFO92] who registered ultrasound image data with a patient. The images were transformed so as to appear stationary within the subject and at the location of the fetus being scanned. Not only the HMD but also the ultrasound scanner was tracked in 3D. The HMD in this work was video see-through but the video images were not used for optical tracking. Bajura and Neumann [BN95] later demonstrated how registration errors in video see-through HMDs could be reduced by using image feature tracking of bright LEDs with known positions. Optical tracking was further advanced by Uenohara and Kanade [UK95], who demonstrated two

[2] Pierre Wellner used the term "Computer Augmented Environments"

[3] The sources of static errors are distortion in the HMD optics, mechanical misalignments, errors in the tracking system and incorrect viewing parameters

[4] Dynamic errors are due to system latency

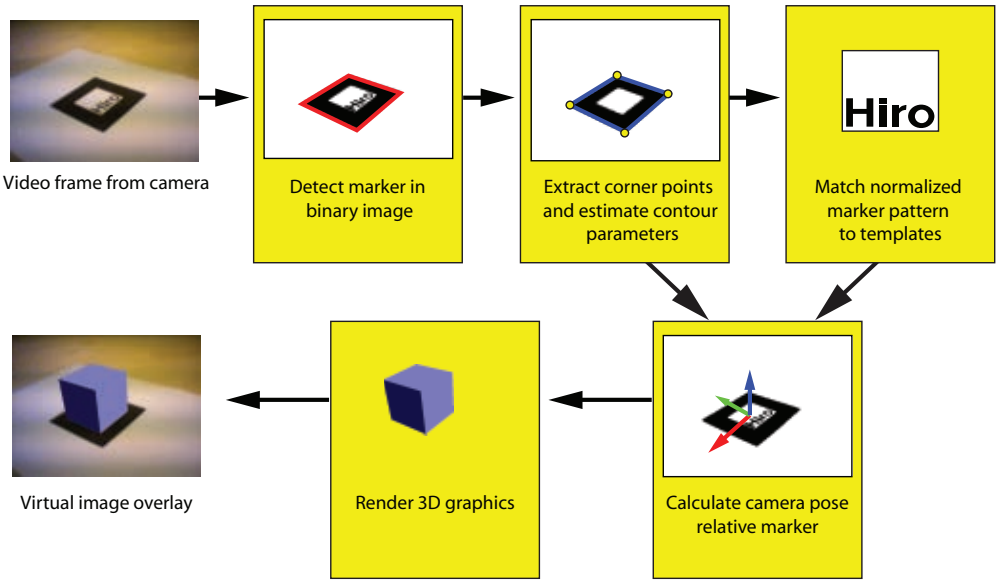

Figure 2.1: Overview of ARToolKit pipeline. A threshold value is used to binarize the input frame. The extracted marker is identified using template matching and its contour used for calculating the pose of the physical camera in marker coordinates. This pose is copied to the virtual camera rendering 3D objects in the same coordinate system.

computer vision-based techniques for registering image overlays in real-time. They tracked a camera relative to a computer box using a model-based approach, and also relative to a phantom leg with attached fiducial markers. Similar fiducial markers were used by State et al. [SHC+96] where they were combined with a magnetic tracker to provide improved tracking stability. The markers were color-coded rings to facilitate quick detection and easy calculation of center of mass.

The practice of using pattern recognition to identify objects had been around since the late 60's when the first barcode readers emerged[5]. Rekimoto's Matrix [Rek98] combined such object identification with 3D camera pose estimation by using a 2D barcode printed inside a black square. The system scanned a binary image to detect the squares and extracted the code that identified the marker. It also calculated the camera pose from the four coplanar marker corners. Multiple markers could be printed by an inexpensive black and white printer, associated with 3D objects based on their encoded ID, and tracked using software only. This successful approach allowed rapid prototyping of AR applications and an easy system setup compared to previous systems that used LEDs or hybrid tracking.

Several research groups developed similar concepts combining identification and pose esti-

[5] $www.nationalbarcode.com/History-of-Barcode-Scanners.htm$

mation based on inexpensive paper fiducial markers. One notable work was conducted by Kato and Billinghurst [KB99] in which they presented a fiducial marker tracking system that was to become one of the most widely adopted platforms for AR: ARToolKit. It demonstrated not only how multiple markers could be used to extend tracking range, but also how markers could be manipulated to provide an inexpensive method for tangible 6DOF interaction (further exploited by Woods et al. [WMB03] for both isotonic position control and isotonic rate control).

ARToolKit works by thresholding the video frame to obtain a binary image in which connected regions are detected and checked for square shape. The interior of a candidate square is then matched against templates to identify it and to obtain its principal rotation[6]. From the principal rotation and the corners and edges, the camera pose relative to the marker is calculated. Figure 2.1 illustrates the pipeline. In a single marker setup, the world coordinate system has its origin in the middle of the marker. Additional markers can be defined relative to the origin marker to extend tracking range or they can represent local coordinates of a scene subset or interaction prop. To be tracked, a marker must be completely visible and segmented from the background. One way to interact with the system is thus to detect if a marker has been obscured by e.g. a finger gesture in front of the camera [LBK04].

Another advantage with 2D fiducials is that they can be printed not only on a blank piece of paper but also in books and other printed media. This allows 2D printed media to be augmented with related interactive 3D animation. MagicBook [BKP01], developed by Billinghurst, Kato and Poupyrev, is one example of how printed content can be mixed with 3D animation. This transitional interface spans the Milgram continuum by not only augmenting a physical book with 3D content but also allowing the user to experience the virtual scene in VR mode. Digital content is viewed through handheld video see-through glasses, which resemble classic opera glasses.

Being able to do single-user AR, researchers began to explore collaborative AR and its hypothesized advantage with superimposed task and communication spaces. Multi-user configurations are challenging since correct registration must be achieved for all participants. Marker-based tracking turned out to provide a means to establish a common reference frame.

ARToolKit was developed for the Shared Space project, which researched interaction in collaborative AR. In [KBP+00], they described the Shared Space interface for face-to-face collaboration where users could manipulate virtual objects directly by manipulating physical objects with markers on them (Figure 2.2, left). Such tangible interaction could be used by novices without training. Users shared the same scene, viewed through HMDs. Collaborative AR using HMDs was also pioneered by Schmalstieg et al., who developed Studierstube [SFSG96, SFH+02] to address 3D user interface management in collaborative AR where users wore head-tracked stereoscopic HMDs. One developed interaction device was the Personal Interaction Panel (PIP) - a panel usually held by the user's non-dominant hand - onto which widgets were superimposed. A tracked pen allowed fine-grained manipulation of buttons and sliders, thus providing an interface not very different from a desktop. By being personal, the PIP acted as a subjective-view display assuring privacy of data. Reitmayr later brought Studierstube to a mobile platform [RS01] bringing together collaborative AR and outdoor AR.

[6]This template matching approach to marker identification limits a system to tracking only known markers; but, on the other hand, it makes it easy for a human to produce and identify markers.

Figure 2.2: HMD in indoor and outdoor AR. Left image shows a HMD-based AR system with marker-based tracking and tangible interaction. Right image depicts Tinmith, an outdoor AR setup, used for playing AR Quake. In both cases, a commercially available opaque HMD has been made video see-through by aligning it with a camera. (Photographs courtesy of University of Washington and University of South Australia)

2.1.2 Outdoor AR

Most systems presented so far have been spatially restricted to laboratory or small workspace set-ups. Progress in battery and CPU technology made it possible to carry computers powerful enough for AR. Research and development of robust wearable configurations is important for motivating adoption of AR in many workplaces where cables must be avoided, and also for bringing gaming back to the physical world.

Starner and his colleagues at MIT explored AR with wearable computers [SMR+97] and demonstrated applications that used computer vision techniques such as face recognition to retrieve information about a conversation partner. To prevent scene clutter, hyperlinks - indicated by arrows - were clicked to activate the display of video and 3D graphics. Feiner, MacIntyre, Höllerer and Webster demonstrated the Touring Machine [FMHW97], a wearable configuration for outdoor AR. In outdoor applications, it is not possible to rely on fiducials since the user can roam freely and cover an area not possible to prepare with markers. Instead, a hybrid tracking solution that combined differential GPS for positioning with a magnetometer/inclinometer for orientation was developed. It used an optical see-through display HMD connected to a backpack computer to display labels in a campus tour application and used a handheld computer with stylus for interaction. A later version displayed 3D models of buildings that had once occupied the campus, in addition to supporting remote collaboration [HFT+99].

A similar backpack configuration was Tinmith, constructed by Thomas et al. [TDP+98] for research on navigation where a user is presented with waypoints. This platform was later extended and used for ARQuake [TCD+00]. ARQuake was an AR version of the popular PC game

Quake[7] now taking place in the real world with virtual monsters coming out of real buildings. This was realized by modeling the game site in 3D and aligning it with physical buildings for occlusion. Correct registration was achieved by combining GPS and a magnetic compass with optical tracking provided by ARToolKit. ARToolKit was used near buildings and indoors where GPS and compass either were not accurate enough or could not operate due to blocked satellite signals. Players aimed the gun using head movements with a two-button gun device firing in the direction of the center of the current view (Figure 2.2, right). Piekarski and Thomas developed glove-based interaction [PT02] using pinch-sensing and fiducials for tracking. It allowed the Tinmith user to point and select using a virtual laser beam and also to enter text using gestures.

Another famous realization of a classic PC game in the real world using AR is Human Pacman, developed by Cheok et al. [CWG+03]. Players move around in an outdoor arena to collect virtual cookies displayed in their HMDs. They are also supposed to find Bluetooth-embedded physical objects to get special abilities in the game. Pacman players, tracked by GPS and inertia sensors, collaborate with *Helpers* who monitor the game from behind a PC and give advice, a role common to pervasive gaming. Pacmen are hunted by Ghost players who try to catch them by tapping on their shoulders, which are equipped with capacitive sensors to detect the catch event. Ghosts too have their own Helpers who see the game in VR mode. Human Pacman demonstrates AR gaming enhanced by ubiquitous computing and tangible interaction technologies.

Computer games are interesting for AR researchers, not only for their being appealing applications that make users more willing to explore innovative metaphors, hardware etc. and also make users more tolerant of system limitations [SLSP00]; as Schmalstieg discussed in [Sch05], many 3D games augment the virtual view with status information, radar screens, items lists, maps etc. This makes them a rich source of inspiration for future AR interfaces.

2.1.3 Handheld AR

In parallel with the development of outdoor configurations, researchers began to explore handheld devices for indoor AR tasks. Research started out on custom setups and later migrated to commercial devices with integrated sensing and computing, lowering the bar for realizing AR.

Among the first to experiment with handheld spatially aware displays was Fitzmaurice, whose Chameleon [Fit93] was an opaque palmtop display tracked in 6DOF and thus aware of its position and orientation. It allowed the user to interact with 3D information by moving the display itself without any need for complex gloves or other external input devices. Inspired by Chameleon, Rekimoto developed NaviCam [RN95], the first see-through handheld display. It consisted of an LCD TV tracked by a gyro and equipped with a CCD camera. NaviCam was connected by cable to a PC for 2D augmentation of objects that were identified in the video stream from detected color barcodes. Codes like these made it possible to track mobile objects such as books in a library. In an evaluation [Rek95], handheld AR proved to be superior to HMD-based AR for finding targets in an office environment. TransVision [Rek96b] extended NaviCam with two buttons and connected it to a graphics workstation for 3D overlays. This system allowed two users to collaborate sharing the same scene. Selection was made by pushing a button and was guided

[7] $www.idsoftware.com$

Figure 2.3: Handheld AR: Invisible Train. This application demonstrates marker-based tracking for video see-through AR on a PDA. Interaction is based on device motion for navigation and stylus input for selection and manipulation: in this case opening and closing track switches. (Courtesy of Vienna University of Technology.)

by a virtual beam along the camera axis. Objects were manipulated in an isomorphic fashion by being locked to the display while selected.

A concept similar to NaviCam was Mogilev's AR Pad [MKBP02]. It consisted of a handheld LCD panel with an attached camera, both connected to a desktop computer running ARToolKit for tracking and registration. In addition, it had a Spaceball input device attached to it, enabling not only isomorphic interaction, but also the 6DOF interactions supported by the Spaceball control. Users appreciated not having to wear any hardware, but found the device rather heavy.

The first handheld AR device that was not tethered to a PC was Regenbrecht's mPARD [RS00], which consisted of a passive TFT and camera combo that communicated with a PC via radio frequency communication. As PDAs became more powerful, researchers started to explore how to use them for handheld AR. Among the first PDA-based projects to use tracking was Batportal [NIH01] by Newman, Ingram and Hopper. It was used for 2D and 3D visualization and the display's view vector was calculated from two positions given by ultrasonic positioning devices, called Bat - one around the user's neck at a fixed distance from his eyes, and one on the PDA. Though not overlaying virtual objects on the real world, Batportal showed that a PDA screen could be tracked and display graphics streamed from a workstation.

AR-PDA, developed by Geiger et al. [GKRS01], was the first demonstration of AR on an off-the-shelf PDA. It was a thin client approach where a PDA with an on-board camera sent a video stream over WLAN to an AR-server for augmentation and displayed the returned stream. Similar client/server approaches were taken by other researchers e.g. Pasman and Woodward [PW03].

By porting ARToolKit to the PocketPC platform, Wagner realized the first self-contained AR system on a PDA [WS03], with optional server support for increased performance. To make it run natively at an interactive frame rate, they identified the most computationally heavy func-

tions and optimized them by rewriting them with fixed-point[8], replacing double precision floats. This operation tripled the number of pose estimations per time unit. The rewrite to fixed-points was necessary since PDAs lacked floating point hardware. In addition to the inside-out tracking provided by ARToolKit, an outside-in approach using ARTTrack[9] was implemented. An indoor navigation application was used to test the system. This platform was combined with a lightweight version of Studierstube and used to explore collaborative AR with PDAs. One application was AR Kanji [WB03], where participants were supposed to match icons depicting objects with corresponding kanji cards that had markers printed on their back for identification and augmentation. Flipping a card enabled the system to determine if the correct card was chosen. Invisible Train [WPLS05] was a collaborative AR game where a wooden railroad track was augmented with trains and switches (Figure 2.3). Players tapped the switches with a PDA stylus to change tracks and prevent the trains from colliding. Other applications included a collaborative edutainment application for learning art history called Virtuoso [WSB06]. The original ARToolKit port was further optimized and extended - the new version called ARToolKitPlus [WS07]. This platform has later been migrated to other handheld Windows devices such as mobile phones running Windows Mobile and Gizmondo.

When researchers turned to mobile phones, or more precisely camera phones, for handheld AR, a similar evolution from client/server setups to self-contained systems occurred. The lack of mobile phones with WLAN made it hard to deliver real-time AR in client/server configurations and few attempted to do so. The most notable work was NTT's PopRi[10] where video captured by the mobile phone was streamed to a server for marker detection and augmentation. The graphics was produced using image-based rendering, resulting in very realistic overlays with, for example, furry toys. Despite being used over a 3G network, the system suffered from latency and low frame rates. Sending individual images over Bluetooth [ACCH03] was also tried, but could not provide real-time interaction.

Parallel to the first contributions of this thesis, Möhring, Lessig and Bimber experimented with a prototype marker tracking solution on camera phones [MLB04]. They designed a 3D paper marker format onto which color-coded coordinate axes were printed. Using computer vision, they could identify the four non-coplanar axis end-points and reconstruct the coordinate system for the current viewpoint and use it to render 3D graphics with correct perspective and at interactive frame rates.

Due to lack of other built-in sensor technologies, most handheld AR configurations have been based on optical tracking, using a built-in or plugged-in camera. Kähäri and Murphy, researchers at Nokia, have recently begun to explore outdoor AR on mobile phones with source-less tracking. Their MARA[11] prototype uses accelerometers in all three axes to determine orientation, a tilt compensated compass for heading, and a GPS antenna for positioning - all sensors placed in an add-on box and communicating via Bluetooth. It can be used, for example, in highlighting a friend in a crowd based on their GPS position, or overlaying buildings with labels and providing real-world hyperlinks to related web pages. When tilted into a horizontal position, the phone

[8] Fixed-points use part of the integer data type to store decimals.

[9] $www.ar-tracking.de$

[10] $labolib3.aecl.ntt.co.jp/member\_servlet\_home/contents/U015.html$ Demoed at ART03 Tokyo, Japan

[11] $research.nokia.com/research/projects/mara/$ Demoed at ISMAR06 Santa Barbara, USA

automatically displays a 2D map with the user's position indicated. Future versions of MARA and other spatially aware mobile phones might be a perfect platform for browsing geospatial information spaces similar to the Real World Wide Web [KM03], envisioned by Kooper and MacIntyre.

Discussion
In previous research on handheld AR, there are several gaps that are addressed in this thesis. First, there is a lack of real-time tracking on mobile phones and proposed client/server approaches are costly if a user is billed per KB. Second, no interaction techniques beyond navigation and simple screen tapping have been developed for handheld AR on commercially available devices. Interaction techniques have been developed for custom configurations like NaviCam and AR PAD, but not formally evaluated. No work has shown advantages of AR on mobile phones. A mobile phone has an advantage over a HMD by being multimodal. It can be used for both see-through AR and web browsing without compromising the safety of a mobile user. Also having less social implications and configuration requirements than HMD-based systems, mobile phones are likely to be the platform that realizes the visions of the Touring Machine and brings AR to a broad audience.

2.2 Beyond the Keypad

Few handheld devices have interface components dedicated to graphics interaction. Even handheld game consoles support only a subset of the atomic actions necessary to perform the canonical 3D tasks presented in Section 1.3. Researchers have exploited the tangibility of handheld devices to extend their interaction capabilities for graphics applications and also for navigating an interaction space extending beyond the physical screen. This section discusses such efforts in extending phone interfaces and workspaces. Of special interest is the use of built-in cameras for continuous establishment of device position or relative motion. This research is important to fully utilize phones' increasing 3D rendering capabilities and the consequent ability to use the third dimension for compensating the limited 2D screen area. The main tracks are motion field estimation for 2D interaction and tracking of code markers for object identification and up to 6DOF device tracking. These latter efforts converge with the ones presented in the previous section on AR and the resulting research direction is the one to be advanced in this thesis.

Fitzmaurice's Chameleon inspired not only research on handheld AR, but also novel input techniques for small screen devices. Rekimoto explored tilt-based interaction [Rek96a] using a configuration similar to NaviCam. By tilting the device itself, users could explore various menus and navigate maps. Small and Ishii designed spatially aware displays [SI97] which used gravity and friction metaphors to scroll digital paintings and newspapers. The user put the device on the floor and rolled it back and forth while the painting appeared fixed relative the floor. Like Rekimoto they explored tilting operations to pan an information space extending beyond the display area.

Ishii introduced the concept of Tangible User Interfaces with Tangible Bits [IU97]. The vision was to provide a seamless interface between people, bits and atoms by making bits ac-

cessible through graspable objects and thus perceptible to other senses than sight - restricted to "painted bits" i.e. pixels. Their Tangible Geospace application demonstrated physical icons ("phicons") casting digital shadows in the form of a campus map on a tabletop display; a passive lens that tracked on the display provided a peep-hole into a second information space; and an active lens, consisting of an arm-mounted, mechanically tracked flat-panel display, provided a tangible window into a 3D information space registered with one phicon.

Influenced by the above works, Harrison, Fishkin et al. [HFG+98] built prototypes of devices with manipulative interfaces. They put combined location and pressure sensors in the top corners of a PDA to let the user flick pages in a digital book using thumb motion similar to flicking pages in a physical book. Other sensors detected when the device was squeezed or tilted and mapped these inputs to application controls. They called their approach Embodied User Interfaces and also developed a prototype handheld computer with built-in accelerometers [FGH+00]. Tilt-base scrolling was also demonstrated by researchers at Microsoft [HPSH00].

Peephole displays [Yee03] was another concept based on detecting device motion relative to an information space grounded in the physical world. Inspired by Bier's Toolglass [BSP+93], Yee tracked a PDA in 2D, allowing a user to move it to pan the digital space while drawing text and images covering an area larger than the screen. This isomorphic panning technique was applied to selection tasks and map viewing. With LightSense [Olw06], Olwal extended the peephole display by tracking a mobile phone on a semitransparent surface with printed media. Behind the surface is a camera that detects the phone's LED light and maps it to coordinates in a 3D information space. With the LightSense system it is possible to augment a subway map with several levels of digital street maps, browsed by moving the phone on the surface and lifting it to zoom out. Due to uncertainty in z-values, derived from the size of the filtered light source, height levels are discrete while the xy-plane is continuous.

Not long after camera phones became ubiquitous, researchers began to use them for computer vision-based input. There have been three main approaches which can be characterized by how much knowledge there is about what is being tracked. Not knowing anything about the background, it is still possible to track the frame-to-frame camera motion field - algorithms once developed to track objects in a video streams from a stationary camera are used here for the inverse problem: tracking camera motion relative to fixed features. Camera tracking can be made more robust by looking for objects with known properties like color or overall shape, for example of a human limb. With marker tracking, one knows the exact geometry looked for and not only full 3D tracking is possible, but also object identification using matrix codes or templates as mentioned in the previous section. Next follows a presentation of important works in each category.

2.2.1 Motion Field

Among the first applications to use vision-based input was Siemens Mozzies, developed for their SX1 camera phone in 2003. This game augments the camera viewfinder with mosquito sprites which appear registered to the background by compensating phone movements estimated using lightweight optical flow analysis. The player is supposed to swat the mosquitoes, aiming at them

with cross hairs in the center of the screen.

Several works have used similar techniques to transform a mobile phone into a 2D isotonic device, thus allowing motion of the device itself to be used as input. Hannuksela et al. [HSH05] implemented a block matching[12] algorithm for estimation of global 2D motion. They used a Kalman filter to smooth parameters and used fixed-points to reach interactive frame rates on a camera phone with 104 MHz CPU. This algorithm was able to detect translation along all three major axes and rotation around the z-axis i.e. rotation around the camera view vector. A similar block matching algorithm was implemented by Haro et al. [HMSC05]. In this work, they use motion of the phone to browse photo thumbnails and depending on motion magnitude, the interface zooms in on images. Instead of increasing scrolling speed in response to larger physical motions, the scrolling rate is kept constant while the zoom level is increased. To zoom in on an image, phone movement is slowed down. This concept of camera-based panning and zooming was also applied to map-browsing. Map navigation based on phone motion was also explored by Drab and Artner [DA05].

TinyMotion [WZC06] is yet another example of a block matching algorithm for motion estimation implemented on a mobile device. With it, Wang, Zhai and Canny confirmed that a tracked mobile phone followed Fitt's law. A key contribution of their paper was experimental validation of their approach to motion-based input against a wide variety of background scenes, showing that a TinyMotion controlled pointer was usable under almost all conditions. They evaluated the algorithm formally by benchmarking the detection rate of camera movements under typical conditions. An informal usability test was conducted by distributing camera phones installed with TinyMotion enabled applications and asking the users to test the system against any available background and illumination conditions.

2.2.2 Object Tracking

Paelke developed Kick-Up Menus [PRS04] demonstrating how foot tracking could be used for input. In this concept, a foot-shaped object was detected and its contour extracted using edge detection. The vectorized contour is tracked between frames within a region of interest to check for collision with menu labels superimposed on the video background. This allows the user to "kick" a menu to select items. This work is most known for the AR-Soccer game, later labeled Kickreal. In this game, the foot edge vectors are used to simulate kicking a virtual ball to try to score a goal in a football penalty shootout.

SymBall [HW05], a two-player table tennis game by Hakkarainen and Woodward, used phone motion to control virtual rackets. A player sees the ball approaching and, to hit it, tries to position the racket in screen space by moving the phone in physical space. Instead of detecting and tracking unknown features, users selected a color and directed the camera towards an object with that color. The algorithm then detected and tracked the largest object with the chosen color.

Detecting a printed or hand-drawn circle in the incoming video stream allowed Hansen, Eriksson and Lykke-Olesen to define a mixed interaction space [HELO05] spanned by the space in which the circle could be detected by the camera - an inverted pyramid if the phone is held

[12] Algorithms that track a feature between frames by comparing pixel neighborhoods

Figure 2.4: Data matrix formats. The left code is a SemaCode while the middle one is a QR Code. Both these codes encode the title of this thesis. The right code is a Visual Code that encodes the dissertation number of this thesis. All three codes have guidance bars and/or squares to identify corners and to facilitate both distortion correction and code scanning.

parallel to the circle plane. Motion of the phone inside the mixed interaction space provided simultaneous panning and zooming of maps and images; browsing layered pie menus works by the same principle. The mixed interaction space technique, also called MIXIS, has been deployed in multi-user applications for interaction with large public displays. In [EHLO07] they demonstrate how it can be used to arrange and share photos displayed on a public surface. Multiple cursors are tracked by assigning one color to each user. Exchanging circle tracking with ellipse tracking made it possible to track faces [HELO06] with the front camera and create an interaction space in front of the user's face. As with Kick-Up Menus, the benefit is that there is no need for an object to be placed in the environment. Another advantage is that it provides motion input even if the background is homogeneous or moving - cases where motion field approaches as those presented earlier fail.

2.2.3 Marker Tracking

Many matrix code formats have been developed to link physical and virtual domains. Placing a recognizable code on an object makes it possible to obtain further information and functionality by capturing an image of the code and extracting an ID. Via a central lookup service, the ID is used to retrieve information from a database. If the object is static, a code can be used to position the user and provide context related services. A high density pattern can be used to store data directly, not only an ID. Commercial matrix recognition systems such as QR Code[13], Semacode[14], Data Matrix and Up-code[15] have been used for more than a decade (Figure 2.4). Application areas include tagging electronic components, business cards, magazines, advertising

[13] $www.denso-wave.com/qrcode/$
[14] Semacode ($semacode.org$)
[15] $www.upc.fi/en/upcode$

flyers etc. Recently, researchers have begun to use animated codes on computer displays [LDL06, LB07] to provide a visual data channel that, unlike Bluetooth, requires no pairing.

To extract data in the first place, the code must be detected in the image. If done at interactive frame rates, code detection can be exploited for motion based interaction. Codes are printed relative to a known geometry, for example a square or a set of bars, and, after detecting the geometry, it is straightforward to extract the code. As has already been demonstrated with ARToolKit and Rekimoto's Matrix, known geometry can be used to calculate camera pose relative to the marker.

Rohs developed visual codes [RG04] with data capacity of 83 bits and two guide bars enabling calculation of camera parameters (Figure 2.4 right). Using correction for barrel distortion, adaptive thresholding and error detection, their first implementation needed more than one second to read the codes and calculate the mapping between code plane and camera image plane. This mapping could also be inverted to determine which coordinates in the code plane correspond to, for example, the central pixel in virtual crosshairs. The code recognition algorithm was extended with block matching to track phone motion in real-time. Five triples of x and y displacements and camera-axis rotation could be calculated each second making it useful for continuous interaction [Roh04]. The user could place virtual crosshairs over a printed table entry, with known offset from a code, to retrieve entry information; rotation and tilt relative code plane were used as additional input parameters. Visual codes could also be displayed on a digital display, allowing phone motion to be used as input over a Bluetooth connection (A similar code-based concept for interaction with large displays using mobile phones was developed by Madhavapeddy et al. [MSSU04]).

Since the visual code recognition algorithm did not run at interactive frame rates, continuous 3D interaction was not possible. Instead, interaction primitives were divided between two modes: static, requiring the camera to stay fixed for a moment; and dynamic, that is continuous [RZ05]. These modes are used for two interaction techniques for large displays: discrete *point & shoot*, where user focus is on device cross hairs, and continuous *sweep*, where focus is on the display cursor [BRS05]. In a multidirectional tapping test based on ISO 9241-9, point & shoot was significantly faster than sweep, which in turn had significantly smaller error rate. Neither technique, however, proved better than the phone joystick. A survey [BBRS06] including these and other techniques where the mobile phone was used as an input device concluded that there were no works on 3D interaction using mobile phones. Similar techniques were also compared to a combination of accelerometers and magnetometers for performance in motion based information navigation on a handheld device [RE07], with tracking relative to a grid of markers yielding best results.

Hachet et al. [HPG05] developed another grid tracking approach to provide 3DOF motion-based interaction for a camera-equipped PDA. Their tracking target is divided into 64 cells, each with a unique two-line pattern; the upper line being a 3 bit blue-green binary code for the x-coordinate in marker space and the lower line the same for the y-value. Starting from the central screen pixel, the borders of the closest cell are identified and 6 color samples are made to extract the x- and y-values. The z-coordinate is calculated from the cell size. This color-based approach is fast and allows the target to be held close to the camera for bimanual interaction.

In a recent project by Winkler et al. [WRZ07], a camera phone running ARToolKitPlus is held in front of a stationary marker allowing phone motion to be used to control a car in a racing

game. This interaction technique was preferred to traditional button control by users of the game. Another very interesting study was made by Hwang et al. [HJK06] and compared handheld displays with motion-based and button-based input to stationary displays of varying size and with mouse/keyboard input. In a test measuring the perceived field of view (FOV) relative to actual FOV, the perceived FOV was widened by 50% in the case of a handheld display with motion-based input - an effect not appearing in other configurations. Also such factors as presence and immersion improved significantly with motion-based interaction compared with button-based and reached a level comparable to desktop VR platforms with actual FOV ranging from 30 to 60 degrees (the handheld displays had a 15 degree actual FOV). In a qualitative usability test the handheld display with motion-based input was perceived to be superior in terms of naturalness and intuitiveness when compared with button-based handheld and desktop configurations that used mouse and keypad.

Discussion
Addressed gaps in previous work on motion-based input include lack of 3D interaction techniques, not possible before due to lack of real-time 3D tracking solutions for mobile phones. These need to be developed and studied formally. Tracking a phone with 6DOF enables it to be used, not only for mobile AR, but also for mobile VR, and - when connected to a PC via Bluetooth - as a large-screen 3D input device.

2.3 3D Input Devices and Interaction Techniques

Although many of the works presented so far demonstrate procedures for evaluating novel interaction techniques in both AR and motion-based interaction, there is much to learn from research in VR and desktop 3D interaction. It is relevant since this thesis is the first to explore 6DOF motion-based interaction for mobile phones, hence being different from both PDA-based AR and previous 3DOF and 4DOF camera-based interaction techniques.

Much research on multi-dimensional control has focused on designing new input devices, aiming to develop more natural, faster and easier-to-learn ways to map human intentions to computer applications. The human hand joints above the wrist involve 23 DOFs [Stu92] and, ideally, a single-hand input device should be able to sense each possible DOF - a very challenging technical problem. However, since most tasks concern rigid objects, only involving 6 DOFs, most multi-dimensional input devices aim to support 6DOF interaction.

Ware designed the Bat [War90], an isotonic device for 6DOF interaction allowing users to use hand motion to place virtual objects. Fröhlich designed two 6DOF devices: GlobeFish and GlobeMouse [FHSH06]. Both devices used a 3DOF Trackball[16]; combined with a SpaceMouse[17] in the GlobeMouse case, and mounted in a mechanically tracked frame to realize the GlobeFish. Both devices performed better than a standard 6DOF SpaceMouse in 3D docking tasks.

Sundin [Sun01] designed from scratch a position controlled elastic 6DOF input device, called SpaceCat, for desktop 3D applications. This included specifying parameters such as spatial

[16] Ball housed in a socket containing sensors to detect rotation of the ball
[17] Elastic 3D input device

and temporal resolution, short and long term steadiness; physical properties like mass, size, elastic forces and torques, sticking friction, viscosity etc. The realized product was compared to Spaceball[18] in qualitative and quantitative usability tests where both devices were used for the same tasks. From gathered data, SpaceCat was found to perform better in terms of completion times and ease-of-learning.

There are several papers that provide an overview of multi-DOF input devices and design issues. Zhai discussed user performance in relation to 6DOF device design [Zha98]. According to Zhai, isotonic devices have advantages in being easy to learn and fast, outperforming other devices. The downsides include limited movement range and low coordination if clutching is needed, due to human joint constraints. Fatigue is another issue together with lack of persistence in position when released. These characteristics can be expected to apply to mobile phones tracked with 6DOF. Zhai also concludes that the participation of fingers can significantly improve performance compared to devices operated by wrists and arms alone.

Subramanian and IJsselsteijn [SI00] surveyed spatial input devices and provided a classification based on naturalness, range of interaction space, DOFs, and atomic actions. Fröhlich [Frö05] used a taxonomy by Jacob et al. [JSMJ94] for classifying novel multi-DOF input devices according to their separable and integral DOFs. Quoted studies had shown that for docking tasks, integral DOFs performed better in the first phase, called the ballistic phase, where the users alter several DOFs at the same time to rapidly approach the target. In the last phase, called the closed-loop phase, separable DOFs performed better and allowed users to fine-tune one DOF at a time to complete the docking task. This suggests that an input device should support 12DOF where the 6 object DOFs can be altered all at once or one at a time. One such device is the Cubic Mouse [FP00], which is an isotonic device with a 6DOF + 3×2DOF design. It consists of a small box with three orthogonal rods running through it. The box is tracked with 6DOF using a magnetic tracker, and each rod controls two DOFs. For the ballistic phase, a user alters 6DOF in an integral fashion by moving the cube. In the closed-loop phase, each DOF can be altered separated by either rotating the rods or by pushing and pulling them for translation. Each rod thus corresponds to a local or global coordinate axis. Research on separable and integral multi-DOF input is interesting with respect to this work since a mobile phone can provide 6DOF input both in an integral fashion when tracked in 3D and in a separable manner using the keypad, as will be shown later.

In many works, evaluations have been made for specific applications and platforms so different from mobile phones and AR that their empirical results are not applicable. However, there have also been efforts towards understanding generic 6DOF interaction issues. Such papers are useful because they provide examples of well designed usability studies for object manipulation, particularly translation and rotation tasks.

Mine [Min95] described several interaction techniques and usability study results for immersive VR environments. Hand has produced an extensive summary of the literature for 3D interaction in desktop and VR applications and usability study results [Han97]. Hinckley et al. [HTP+97] analyzed the usability of four 3D rotation techniques. Virtual Sphere and ArcBall [Sho94], two 2D mouse-driven interaction techniques, were compared to isotonic input devices

[18] Isometric 6DOF input device

based on magnetic trackers - one with ball housing and one without any housing. In a within-subjects experiment with 12 male and 12 female participants, the task was to rotate a house model to match a randomly rotated one displayed on the other half of the screen. When each of the 15 tasks was finished, users clicked a foot pedal to get feedback. Statistical analysis of data for the 10 last tasks for each input technique showed isotonic techniques to be faster than mouse-based ones, but did not indicate difference in accuracy. A between-subjects test mirrored these results. Additional feedback in the form of user ranking of input techniques showed that the ball-shaped isotonic device was the most popular way to rotate 3D objects. Further insights came from observing users and interviewing them.

Bowman, Gabbard and Hix [BGH02] surveyed usability evaluation methods for VR environments and discussed what made such environments challenging to evaluate. The two major approaches presented are testbed evaluation and sequential evaluation, with the testbed approach being more generic and more suited for evaluating basic tasks such as selection and manipulation, but also being costly to perform.

3D interaction studies have also been conducted with AR interfaces, although there are not many that explore general object manipulation, which is of particular interest in this thesis. One example is Ellis [EBM+97] who conducted an experiment to explore the user's ability to move a virtual ring over a virtual wire in an AR display with different rendering latencies.

Discussion

A survey of user-based experimentation in AR by Swan and Gabbard [IG05] found that only 8% of AR publications had an evaluation study and only a portion of theses involved virtual object manipulation. None of the cited works were on handheld AR. This thesis addresses this gap by providing formal user studies and empirical data.

The methodology used for evaluating 3D interaction in this thesis employs a combination of quantitative measurements and qualitative feedback in the form of interviews and rankings, similar to Hinckley. To obtain sufficient data from a limited pool of subjects, a within-subjects approach is used. This means that all participants perform all tasks with all studied techniques. Observations are made manually and no video is recorded.

There are two fundamental differences between works on 3D input, like Sundin's, and that presented in this thesis:

1. Working with commercially available mobile phones implies that input device parameters, that is physical properties such as mass, ergonomics, viscosity etc., are constant. Instead, interaction techniques must be developed to exploit existing hardware design capabilities. This poses a different set of challenges compared to software and hardware co-design.

2. No prior work has been found where 3D objects are manipulated with 6DOF on a mobile phone; thus, there exist no best known practice against which a new approach can be benchmarked and hypothesis tested. Instead, a batch of viable interaction techniques will be developed and compared using standard evaluation methodologies to find out which techniques are best suited for 6DOF interaction in AR contexts.

Chapter 3

Realizing Mobile Phone Augmented Reality

This chapter presents contributions of the included papers. It is divided into four sections representing the focus areas of this thesis. The first section presents development of a marker-based tracking solution where the global coordinate system is defined by fiducial markers. Such markers can be printed on paper, for example in newspapers, to provide an additional, dynamic information space. Tracking the phone relative a marker allows phone motion to be used for 3D interaction as described in the second section. Tracking several connected phones in the same coordinate system enables collaborative AR, which the third section elaborates on. In the fourth section, phones are not only connected to each other but also to other devices that provide input and output. Each section presents example applications to further make clear the contributions and illustrate possible outcomes.

3.1 Mobile Phone Augmented Reality Platform

With a camera aligned with a screen and coupled with a CPU, mobile phones are capable of video see-through AR as indicated in the works on camera-based interaction presented in the previous chapter. The video stream captured by the camera is analyzed for features that are used to calculate the camera pose. While a range of physical features permit tracking with 6DOF, a general tracking solution benefits from standardized features that are easy to reproduce. Fiducial markers meet this requirement and development of optical tracking based on markers is described in paper **I**. For this, the ARToolKit tracking library was ported to the Symbian[1] platform for the first time. ARToolKit was the obvious candidate being open source, easy to use and used by many research groups even for PDA-based AR. It also allows rapid prototyping of applications.

The first challenge was to adapt it to the rather restricted computational environment provided by mobile phones. The initial port was realized by adding a C++ wrapper class to the open source

[1] *www.symbian.com*

ARToolKit C code developed for PCs. This was necessary to avoid initialized data prohibited by Symbian. From images captured by the camera, two new images are created by masking out RGB values: one image used as background texture to make the screen see-through and one image analyzed for markers by ARToolKit. If markers are found, corresponding corner and edge data will be used to calculate the camera pose in marker coordinates. The very first port [HO04] ran at 1 fps on a Nokia 6600 with a 104 MHz CPU, not enough for continuous interaction. To improve performance it was necessary to look at fixed-point arithmetic.

For 3D graphics rendering, OpenGL ES[2] is used, and the extrinsic[3] parameters, calculated by ARToolKit, are converted and used for the `MODELVIEW` transformation in the rendering pipeline to align the physical and virtual cameras. OpenGL ES is a subset of desktop OpenGL and provides low-level functionalities for rendering interactive 3D graphics on limited devices. While OpenGL ES supports floating point, better performance can be achieved with fixed-point.

3.1.1 Fixed-Point Library

Lacking floating point units (FPU), mobile phones must emulate floating point arithmetic in software, making such operations up to two orders of magnitude slower than integer equivalents. To increase speed of the ARToolKit camera pose algorithm, a fixed-point library based on ARM assembler routines was developed. In order to minimize information loss due to conversion between floating-point and fixed-point representations, variable precision is necessary. Trigonometric values, ranging from -1 to 1, use 28 bits for decimals while large numbers use only 8. This necessary variation of precision makes software development much harder compared to using 16 bit precision only, as with OpenGL ES, and much care must be taken to prevent overflow causing loss of the most significant bits. This is challenging because most bits are used (since both the integer and precision parts of a converted floating point number are stored in the same integer representation) and therefore overflow is likely to happen if care is not taken. There are some additional criteria for a fixed-point operation:

- It must have similar computational costs for all possible input. This is essential for predicting the speed of an algorithm rewritten with fixed-points.
- All results must have low error. For example, a Taylor expansion of a function may return a correct result for some input, but be inaccurate for others. A tabulated function will, on the other hand, often have a maximal error when values halfway between tabulated sample-points are used as input.

Fixed-point is based on the integer data type and can be added and subtracted as integers. Other arithmetic and trigonometric operations need to be implemented especially for fixed-points. The most important function is multiplication. This is also the routine most likely to overflow if a naive integer multiplication approach is taken. To prevent overflow and maximize speed, an ARM assembler routine that stored the result in two registers was used. This meant

[2] $www.khronos.org/opengles$

[3] position and orientation of the camera with respect to the world coordinate system here defined by markers

that no data was lost and guaranteed that only the least significant bits were thrown away when the register values were merged. This operation ran 27 times faster than the floating-point equivalent and only 1.6 times slower than pure integer multiplication. Division was implemented as a multiplication of the dividend and the inverted divisor. Inversion was implemented using a Newton-Raphson approximation algorithm where the initial guess was made using a reciprocal table. A similar algorithm was implemented for calculation of square roots.

CORDIC rotators[4] and Taylor expansions were implemented for trigonometric functions, but were rejected in favor of tabulated trigonometry. For example, using CORDIC rotators to calculate both sin and cos simultaneously was 6 times faster than using floating point, but 3 times slower than using tabulated values and linear interpolation. Also, the error increased for input values close to 1. Taylor expansion of sin[5] was 3.6 times faster than the CORDIC equivalent, but was discarded since using necessary high order terms caused overflow for some input unless precision for all input was sacrificed. Instead, 1024 arccos values, ranging from 0 to 1, and 1024 sin values, ranging from 0 to $\frac{\pi}{2}$, were tabulated. To retrieve a value, the array index was calculated from the input value.

Several algorithms were tested and compared for best performance and conformity to stated criteria, with fastest fixed-point operations running on average twenty times faster than floating point equivalents. Not only basic arithmetic and trigonometric functions were implemented, but also vector and matrix operations, such as matrix inversion, cross product, and dot product, optimized in assembler.

Based on [WS03], the most computationally heavy functions were rewritten to use fixed-point instead of double precision floating point. Most other camera pose functions were also rewritten. To estimate necessary precision, a webcam connected to a PC running ARToolKit was moved freely around a marker, moving in and out of tracking range. The largest and smallest values were stored and precision chosen to accommodate them.

This rewrite made the pose estimation run at 4-5 fps on the 6600 and later at 8-10 fps on the Nokia 6630 with a 220 MHz CPU. Exchanging double precision floating point routines for fixed-point equivalents introduced a small increase in tracking instability, but not more than what was considered acceptable. Though some mobile phones now have GPUs that support floating point calculations for graphics, fixed-point representations are likely to remain important for supporting low-end handsets.

3.1.2 Camera Calibration

The intrinsic[6] camera parameters are obtained in a calibration process and used to set the entries for the `PROJECTION` transformation in the OpenGL ES rendering pipeline. To calibrate the camera, the ARToolKit calibration application was modified to take still images instead of a live

[4]These hardware efficient functions use an arctan table, shift, addition, and subtraction operations to emulate a wide range of trigonometric, hyperbolic, linear and logarithmic functions. For details see Ray Andraka's *A survey of CORDIC algorithms for FPGA based computers*

[5]$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ...$

[6]Internal camera parameters such as focal length, skew and distortions

video feed. Five images containing 81 points each were used to calculate the intrinsic parameters. Using more data yields better calibration, but the manual process of identifying points in calibration target images is time consuming. Calibration is important for reducing static registration errors and good calibration can increase tracking performance since the algorithm converges faster if provided with correct camera parameters.

Zooming is not used, either in the calibration process or during tracking. With the mobile phone being a tangible device, the user is assumed to close up on details by moving the phone itself closer to the target. It would however be possible to use digital zoom since the current zoom factor is available through the API. For this to work, the camera needs to be calibrated at several zooming levels; by polynomial fitting, an expression for each intrinsic parameter can be derived.

3.1.3 Further Enhancements

To increase performance, frame-to-frame coherency thresholding was implemented. It works by calculating the accumulated length of marker corner difference vectors between two consecutive frames. If the length is below the defined threshold, the previous camera pose is used. This reduces computation time since the cost of marker detection is redundant compared to camera pose estimation.

ARToolKit tracking is inherently unstable and to remedy this, Double Exponential Smoothing-based Prediction (DESP) - an inexpensive alternative to Kalman filtering [Jr.03] - was implemented to smooth tracking data. The penalty is a small latency when moving the phone since old camera pose parameters influence the current ones. Even though this reduced jitter, the tracking algorithm is noticeably instable if the marker is tracked looking down the z-axis i.e. with a slant angle close to 0 degrees [ABD04]. This is a problem if markers are placed on a vertical wall.

Since lighting conditions may vary during usage, it is important to be able to adapt the threshold used for segmenting out the black marker square from the white background. Too high threshold value would label some white areas black, causing marker detection to fail. Too low threshold would, likewise, make black marker areas white in the binary image. For dynamic thresholding, a simple brute force approach was developed where the threshold value is incremented (a typical step is 20 gray levels) until the maximum is reached or a marker is detected. Dynamic thresholding is especially important when the front camera is used for tracking (as described in Section 3.2.3).

For ARToolKit to detect a marker, all four edges must be completely visible in the video frame. If the marker is partly obscured (such as with a finger) the tracking fails and nothing is rendered. To see how this limitation could be worked around, an experiment with motion flow was conducted. A pyramidal feature tracking algorithm based on patch differences, similar to the works presented in Section 2.2.1, and Harris feature detection were implemented and used to track marker corners (A pyramidal Lucas-Kanade feature tracker was also implemented, but was found inferior to the patch difference one for the purpose of marker corner tracking.). This allowed everything but three corners to be obscured. From the three corners, tracked between frames by the feature tracker, the position of the fourth corner was estimated using simple heuristics. From these four corners, edge information was calculated as usual by ARToolKit. Although

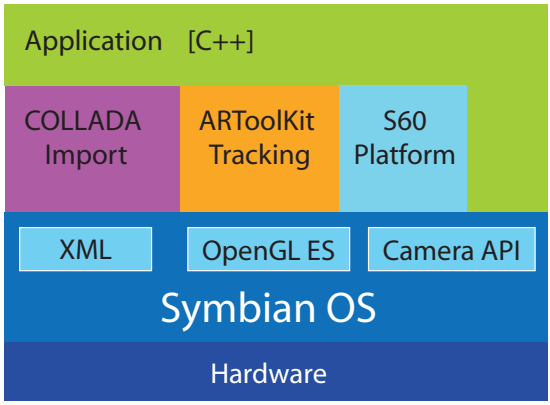

Figure 3.1: Software components. The main components are the ARToolKit tracking library, OpenGL ES rendering and the camera API. These are the necessary parts of any mobile phone AR application. A COLLADA scene importer was developed, based on Symbian XML functions, while other API:s were used for multi-sensory feedback in form of both audio and vibration.

the experiment was successful, this enhancement was not considered essential enough to be included in the platform. There was also a small risk of confusing a marker corner and an interior marker pattern feature, resulting in severe registration errors.

Later, a solution for importing COLLADA[7] files was developed, enabling import of basic scenes from SketchUp, Google Earth or 3D Warehouse[8]. The file ending was associated with the AR application, which made it easy to transfer and open 3D scenes via Bluetooth or as e-mail attachments. Figure 3.1 gives an overview of platform components.

3.1.4 Example Application: Wellington Zoo Campaign

One application area for AR is to fuse printed media with interactive 3D graphics. As an example of this, an ad campaign by Saatchi & Saatchi for Wellington Zoo was based on the platform presented in this section. An ad featuring a marker was printed in a local New Zealand newspaper and readers could SMS a code to a provided phone number. In the returned SMS, a URL pointing the users to a WAP site was provided. When entering the site, the phone model was identified and if found compatible, the user was instructed to download and install an AR application. After launching the application, the user needed only to point the camera at the printed marker for a 3D animal to pop up. The user could view the animal from an arbitrary view point as long as the marker was framed (Figure 3.2).

The models were created in a 3D animation package and exported from a 3D browser as OpenGL vertex lists. From this floating-point representation, fixed-point vertex lists compatible

[7] An XML-based interchange file format for 3D assets. See: *www.khronos.org/collada/*
[8] *sketchup.google.com/3dwarehouse*

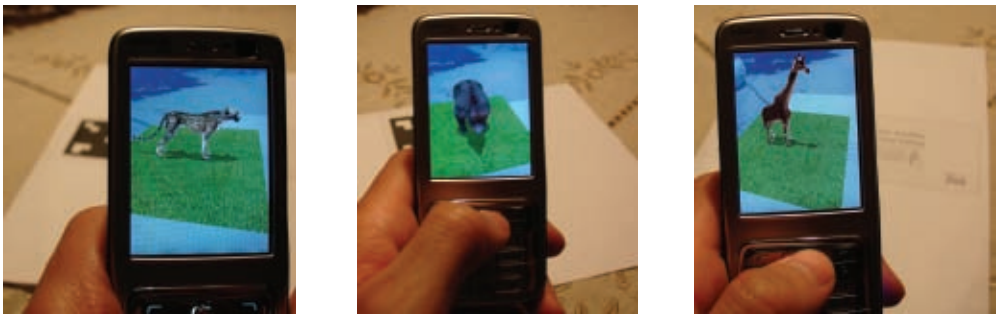

Figure 3.2: Three animal applications used in what is believed to be the first AR advertisement campaign. Readers sent an SMS to obtain the application and after installing it they simply started the application and pointed the phone to experience a virtual animal popping up from the printed ad.

with OpenGL ES were created, though a 1-to-1 mapping was not possible. Installation file sizes ranged from 163KB to 214KB assuming that the phone had native OpenGL ES support. Keeping file sizes small is essential for this kind of OTA delivery and for minimizing end-users' costs if billed per KB.

This example shows the strengths of the presented platform regarding ease of use and compatibility with existing delivery mechanisms. This is believed to be the first real world AR advertisement campaign and it competed in three categories at Cannes Lions 2007[9], though not winning any awards. However, it showed that mobile phone AR is now mature enough for the highly competitive advertising industry.

3.2 Mobile Phone as a 6DOF Interaction Device

Augmented Reality requires the device to be tracked with 6DOF in order to correctly register real and virtual information. 3D interaction requires the input device to sense both rotation and translation in 3D, each with 3 DOF. From these observations follow that an AR enabled mobile phone might be used as an isotonic 3D interaction device where motion of the phone itself is used as input. The challenge is to develop feasible 3D interaction techniques and design experiments to evaluate them. Papers **II**, **III**, **IV** and **V** present contributions to continuous 3D interaction in AR contexts.

3D interaction with multidimensional input devices involves a combination of the following requirements[10] [SD91]:

1. **Navigation:** The user should be able to navigate the scene by controlling the virtual camera.

[9] *www.canneslions.com*

[10] There are other similar lists, but for the purpose of structuring the contributions, this one has been chosen

2. **Global Selection:** The user should be able to select any object in the scene.

3. **Rigid Body Transformation:** This includes the usual requirements of translation and rotation: any transformation which changes the object's orientation and position in space but leaves its local geometry unchanged.

4. **Local Selection:** The user should be able to select part of an object which can be used to deform the object but not its overall position in space.

5. **Deformation:** This includes operations to manipulate the object's geometry.

Descriptions and evaluations of how these requirements are met on AR enabled mobile phones will be presented next. The scope is limited to exploring general 3D interaction in a small workspace and, contrary to practice in desktop CAD and VR environment applications, no additional graphics is rendered to guide the user during interaction. The most likely application scenario being games, dependence on coordinate axis visualization is preferably avoided.

3.2.1 Navigation

In AR, virtual objects are registered with the real world. Thus navigation of the virtual scene reduces to navigating the physical world. Control of the virtual camera is indirect and handled by the tracking system establishing the camera's position and orientation. Thus, the virtual camera is controlled by direct movement of the device. In the general mobile phone scenario this yields an *eyeball-in-hand* metaphor for viewpoint control: to zoom in, the phone is moved closer to the object. The challenges in providing navigation are thus identical to the ones for enabling tracking.

If the virtual scene is based on a single paper marker, navigation can be facilitated by manipulating the marker with the hand not holding the phone. Such interaction where the scene is made to move in correspondence with the marker employs a *scene-in-hand* metaphor. These two metaphors are merged when asymmetric bimanual interaction is used for manipulation of the marker combined with movement of the phone. Typically the paper marker is manipulated by the non-dominant hand while the dominant hand with its finer temporal resolution moves the phone. This division in macrometric and micrometric functions will be discussed further in the cases of isomorphic object manipulation.

3.2.2 Global Selection

To discriminate between different objects, each object must have a unique property obtainable through the selection interface. A simple approach is to use the alpha channel to encode an 8 bit ID. Sampling the color vector of the central pixel, indicated with virtual crosshairs, and extracting the alpha value returns the target objects' ID to the application. This approach is sufficient to select partially occluded objects, given that the following assumptions hold:

1. Not more than 255 objects need to be discriminated between.

2. The alpha channel is used only for selection, not for transparency.

When this is not the case, the full color vector and two render passes can be used as described in Section 3.2.4. Selection is made by moving the phone itself and positioning crosshairs over the object to be selected. Hence selection and navigation are tightly coupled and so are the challenges in realizing them.

3.2.3 Rigid Body Transformation

For interactive AR applications, it is essential to go beyond the looking glass metaphor and allow the user to manipulate the virtual content. In this thesis, two basic 3D manipulation tasks are considered: positioning and rotation. These tasks have previously been studied on other platforms, but the differences in form factor and interface motivate research on the mobile phone. For such 3D interactions, the challenge is to provide the user fast and accurate control over each DOF. Several manipulation techniques have been implemented and studied in order to identify the most applicable ones for this platform.

The following translation techniques have been implemented:

- **Isometric rate control using the joypad/button interface.** Each DOF is controlled by a pair of buttons, incrementing and decrementing the translation along the axis. The resulting object translation is continuous at a constant rate. Assuming a small workspace, no acceleration was considered necessary.

- **Isomorphic position control where the selected object is fixed relative to the phone.** In this case there is a one-to-one mapping between hand and object motion only limited by tracking range and joint constraints. The phone is used as a free-flying isotonic input device providing direct manipulation, which might introduce fatigue.

- **Bimanual isomorphic position control.** Here the user moves both the phone and the marker. The user's dominant hand, with its finer temporal resolution, applies micrometric control of the phone while the non-dominant hand applies macrometric control of the paper marker. This is similar to using pen and paper, where coarse positioning (macrometric control) of the paper is combined with finer control (micrometric control) of the pen.

- **Gesture input using the front camera**. This interaction technique requires bimanual operation with the phone held in the user's non-dominant hand. The movement of one finger is tracked by the front camera and is mapped onto object translation. 3D finger tracking is realized by attaching a small marker to the user's index finger. Running both cameras at the same time is still too slow and hence there are two modes: viewing mode with a live background and interaction mode with a frozen background. A user must activate interaction mode by pressing a button.

- **Isotonic rate control.** This technique maps rotation of the phone onto object translation. Tilting the phone forwards or sideways translates the object in the xy-plane. To translate

along the z-axis, the phone is rotated like a screwdriver. The speed by which the object translates scales with rotation angle. Similar to the gesture case, two modes are used. Rotation is relative to the current orientation when switching to interaction mode. Thresholding of movements provides a stable zone where no translation is performed. This prevents unintentional input caused by tracking instability and/or involuntary hand movements.

- **Physical prop.** The transformation of a physical object is mapped to a virtual one. For example, a cube with markers can be manipulated in front of the camera. The drawback is that at least one cube marker and one global scene marker must be visible at the same time. Another drawback is that a second object must be present hence compromising the ambition not to be dependent on external input devices. However, the low cost and complexity of a paper cube motivated an exception.

The following rotation techniques have been implemented:

- **Isometric rate control using the joypad/button interface.** Each DOF is controlled by a pair of buttons, incrementing and decrementing the rotation around the corresponding axis (object's local coordinate axes) at each frame update. The rotation is continuous at a constant rate. As with translation, no acceleration was implemented. Instead the increment step size was chosen to provide a reasonable trade-off between speed and accuracy.

- **ArcBall.** This well-known quaternion-based rotation technique encapsulates an object with a sphere that can be dragged using the mouse. It has been adapted to the mobile phone, where the central pixel, indicated by cross hairs, is used as mouse pointer and 2D motion of the phone itself is mapped to 3D rotation angles depending on where on the virtual sphere the user clicked and the subsequent device trajectory.

- **Isomorphic position control where the selected object is fixed relative to the phone.** This is the same as in the translation case. Limited tracking range and human joint constraints make clutching necessary for larger rotation angles, thus resulting in low coordination.

- **Bimanual isomorphic position control.** Similar to the translation case except that the non-dominant hand rotates the marker paper around the z-axis. For rotation around the z-axis, bimanual interaction effectively doubles the maximum rotation angle possible without clutching.

- **Isotonic rate control.** Similar to the translation case with phone rotation mapped to object rotation at each update. The rate is thus moderated by the rotation magnitude. This technique allows rotation around all three axes at the same time, though a threshold is used to prevent unintentional rotation.

- **Gesture input using the front camera.** Here finger motion is mapped onto rotation angles either using the ArcBall technique or 2D rotation along the motion field vector. The downside is the number of modes necessary in both cases, with interaction mode divided into two submodes.

Figure 3.3: Experimental setup and tasks. Participants sat at a table with a tangible paper marker. In positioning and rotation tasks, they were supposed to align a solid block with a wireframe one as fast as possible.

- **Physical prop.** The transformation of a physical object is transferred to the virtual one as with translation. Here the uses must rotate the cube using only one hand. This might be tricky, but at the same time it exploits finger interaction which, for the general case, is considered to provide superior control compared to wrist movements [Zha95].

All of the above techniques were first tested informally and some were deemed unfit for AR interaction on mobile phones. Interaction with a physical prop obscured too much of the camera FOV. Gesture-based ArcBall turned out to be too complex to be useful with at least three modes for viewing, positioning of finger and actual interaction. Also tilt-based translation (isotonic rate control) turned out not to be competitive.

Paper **II** presents the first user study on AR 3D interaction on mobile phones. This pilot user study compared translation techniques and rotation techniques separately. First, three translation conditions were compared: isomorphic position control, isometric rate control and bimanual isomorphic position control. Qualitative and quantitative feedback was obtained in addition to user comments and observations. Participants sat at a table with a paper marker on it (Figure 3.3 left). In the bimanual cases, they were allowed to manipulate the marker. They completed five tasks where a solid object was to be aligned with a wireframe one (Figure 3.3 middle). In each task, the goal was to get below an error margin as quickly as possible.

Timings showed that isomorphic position control where the object was fixed to the phone was significantly faster than isometric input using the keypad. Questionnaire answers showed that the users felt that they could position the object more easily in the isomorphic condition while the isometric condition was perceived as more accurate. Having the object fixed to the phone coordinate system was also perceived as quicker and more enjoyable. Subjects ranked isomorphic interaction highest with regard to ease of use. There was no significant difference between one-handed and bimanual isomorphic interaction.

The same subjects also participated in a comparison between four rotation conditions: isomorphic (one-handed and bimanual), ArcBall and isometric keypad input. Here the task was

Figure 3.4: The left image shows 2D finger interaction based on motion field estimation. The remaining two show 3D finger tracking realized by putting a marker on one finger. In both cases, the front camera is used to register input.

to align a solid object with a wireframe one, both having the same center (Figure 3.3 right). Timings indicated that ArcBall and keypad input are on average twice as fast as isomorphic input. No significant differences were found among survey answers and ranking despite significant differences in completion times.

The conclusions from this study are that isomorphic interaction is the best for translation while isometric input is the most suitable for rotation, rivaled only by ArcBall. Bimanual interaction does not seem to be important and half of the participants were observed not to utilize this option when available. There was also a tendency for users to use their non-dominant hand to stabilize the phone in the keypad input conditions. This highlights issues with non-separated navigation and interaction. User interviews indicated that in the bimanual cases users made gross movements with the phone and instead made finer movements with the paper.

In paper **III**, a second pair of user studies was conducted to research the potential of the front camera to enable 3D interaction. Figure 3.4 shows 2D and 3D finger tracking. The first study compared three different translation techniques: isomorphic position control, isometric rate control and gesture input with a finger marker. The tasks were the same ones as in the first user study presented above and in paper **II**. This allowed a direct comparison with previous results, giving a hint whether there were any flaws in the evaluation methodology.

This study also compares the effect increasing the dimensionality of the task has on the results. There were two tasks which could be accomplished by moving the block in only one direction. One task could be completed by moving the virtual block in two directions and the remaining two tasks required the user to move the object in all three directions. There were, however, no comparisons between individual DOFs.

Timings showed significant differences between task performance time both across the three interaction techniques and for the keypad condition, across tasks. In the keypad input condition, it took longer for the user to complete tasks when these required motion in more directions. In both the gesture input and isomorphic position control case, there was little difference in task time as the tasks required manipulation in more dimensions. The isomorphic position control case performed better than the other conditions except for the keypad input condition in the simple one degree of freedom case. For 3DOF translation, moving the phone itself is significantly faster

than the alternatives.

Analysis of the qualitative feedback from questionnaires and rankings found no statistically significant difference between conditions. The subjective test results were, however, interesting, as they demonstrated that, despite two of the interface methods - tangible and finger interaction, being novel to the users, they were "good enough" for them to use given the similarity in rankings. They were however instructed on how the interaction technique worked and were also allowed to practice before attempting the tasks.

A rotation study was conducted comparing three rotation techniques: isotonic rate control (tilt), gesture interaction based on finger tracking, and isometric rate control (keypad). As in the translation study, dimensionality varied between tasks. Phone tilt and finger tracking used motion flow algorithms instead of markers for interaction. Motion flow tracks only with 2DOF and thus requires an extra mode to support full 3DOF rotation. Despite this limitation, motion flow is interesting because commercial applications with finger-based input would most likely use it.

Analysis of the timing data showed keypad input to be fastest, followed by tilt input, and finger input being slowest. There was a significant difference in results across interface type and also across task complexity, meaning that increasing the number of DOFs resulted in longer task completion times.

Questionnaires showed one significant result: keypad input was perceived as being more accurate than finger input. Keypad input was also ranked best followed by tilt and last finger input with respect to both how easy, how accurate and how quick the interaction technique was perceived to be. In interviews, participants thought that the front camera field of view (FOV) was too small and that hand coordination was difficult. While it would be possible to experiment with modified optics for bigger FOV, results would not be applicable to commercially available models.

These studies confirm the findings in the previous studies, presented earlier and in paper **II**, that isomorphic position control can be more effective than isometric rate control for object translation, while isometric rate control should be chosen for object rotation. It should be noted that the purpose of the conducted studies has been to obtain a high-level understanding, eliciting which general categories work best for 3D AR interaction. Further investigation is needed to optimize each interaction technique. Especially rate control mappings could benefit from further research.

In the above user studies, translation and rotation have been studied separately. In a general 3D application it would be necessary to combine both transformation modes to provide full 6DOF interaction. For isomorphic position control, 6DOF is the default, but other combinations require the user to switch between modes. Paper **IV** shows how keypad-based rotation and translation can be combined using a simple menu-based approach. The rich set of buttons allows most combinations to be realized. For example, the ArcBall rotation technique can be combined with any translation technique given one button assigned for switching mode. The best combination for 6DOF path following has yet to be researched, but both studies indicate that a combination of isomorphic position control for translation and isometric keypad rate control for rotation is the best choice. This is the combination of choice for CMAR, presented in Section 3.3.2.

Since a mobile phone supports 6DOF interaction in both integral (isotonic position control

Figure 3.5: Selecting and manipulating several vertices to deform a polygon mesh. Selected vertices are locked to the phone and manipulated by phone movements. A group of vertices are selected by "painting" them and then framing one of them with the square located in the center of the screen.

using phone motion) and separable (isometric rate control using the joypad/keypad) fashion, it would be possible to explore 12DOF interaction where integral manipulation is used in the ballistic phase of a docking task and separable fine-tuning is used in the close-looped phase [Frö05].

3.2.4 Local Selection

Selecting individual vertices is essential for editing a mesh where the geometry is changed, but the overall position in space remains the same. The challenge is to discriminate between large numbers of vertices, also being densely projected onto screen space. This is achieved by assigning a color vector for each vertex based on its ID. Using a 32 bit RGBA color vector allows discrimination between $2^{32} - 1$ vertices. From a pixel sample, the ID can be retrieved and the corresponding vertex selected. Individually colored vertices are only rendered when the selection button is pressed and are overdrawn with uniform vertices in the final image.

For distant vertices it is hard to use a single pixel sample for selection. Tracking jitter combined with small unintentional movements makes it difficult to position the cross hairs over a particular vertex. To overcome this, the crosshairs were replaced by a square inside which five pixel samples were made. The central sampling point is ranked higher and thus unintentionally selecting two vertices can be avoided. Selection of multiple vertices as depicted in Figure 3.5 is made by entering a multi selection mode and then "painting" the vertices to be selected using the phone as a brush. For further details see paper **V**.

3.2.5 Deformation

To deform a mesh, selected vertices can be manipulated as objects, using the object manipulation techniques outlined above. Paper **V** describes the first mobile phone application that allows a user to manipulate individual vertices in a polygon mesh. Here an arbitrary sized grid is generated

procedurally, with each vertex having a unique ID mapped to an RGBA color vector for local selectability as described above.

When selected, a vertex is fixed in the phone camera space, ready for manipulation using the isomorphic position control technique (Figure 3.5). At each frame update, vertex positions are used to synthesize the triangle mesh and to calculate its normal vectors. To avoid sharp corners, the mesh is smoothed by distributing translations to neighboring vertices. A group of selected vertices can be manipulated with 6DOF without altering their spatial relationship. This is useful when several vertices are to be displaced by the same amount. Selecting multiple vertices is performed as described in the previous subsection, that is, by "painting" them.

While it makes little sense to do 3D mesh editing on a phone, such a task poses interesting challenges and might constitute a future testbed for 3D interaction techniques. From tests it was clear that mesh editing is a more complex task than scene assembly and research on mobile phone mesh editing using AR is likely to benefit general atomic actions such as local selection.

3.2.6 Usability Aspects

There are fundamental usability aspects that are of concern for all 6DOF interaction devices. Here will be discussed how some of them turn out in a mobile phone AR context based on experiences from the research on interaction.

Zhai [Zha98] lists six aspects to the usability of a 6DOF input device:

- Speed
- Fatigue
- Accuracy
- Coordination
- Ease of learning
- Device persistence and acquisition

A tracked mobile phone with motion-based input can operate as fast as the user can move the device. If higher speed is required, for example for a larger workspace, rate control interaction techniques can be deployed and adapted to yield high speed transformations. This can be achieved by increasing each increment and/or using acceleration.

Fatigue is an issue for most isotonic devices, including the mobile phone. With smaller and lighter models, fatigue is reduced but not eliminated. Strategies such as freezing the background, and so decoupling manipulation and navigation, can be adopted as discussed in Section 3.4.2. This makes it possible to use isotonic input to navigate and freeze a scene, then to assume a low fatigue position for manipulation using, for example, button input. Section 3.4.3 describes another strategy suitable when the data is 2D only: here the device is placed on a horizontal surface and operated like a 2D mouse.

As with speed, accuracy is limited by human motor skills in an isomorphic configuration. Also tracking instability affects accuracy when the object to be manipulated is locked to the phone and then released, thus being projected from camera space to marker space. If higher accuracy is required, keypad input, for example, can be scaled to yield sufficiently small increments.

Coordination can be measured by the ratio between the actual and the most efficient trajectory. To maximize coordination, a device must be able to alter all degrees of freedom at the same time and at the same pace. Isotonic position control scores high on coordination while other input techniques are suboptimal since they must switch between translation and rotation mode or be combined with other techniques to deliver full 6DOF interaction. This is, however, compromised if clutching is necessary, thus requiring the device to be moved in a direction that is opposite to the optimal trajectory. In such cases, isometric interaction might provide better coordination.

Device persistence is the extent to which the device stays in position when released. Traditional isotonic 3D input devices score low on this unless they are grounded perhaps with an armature or by being suspended with wires. In the case of a tracked mobile phone, the display is part of the interaction device and by freezing the scene, some temporary persistence can be obtained since the scene remains fixed relative to the phone until unfrozen.

Ease of learning for the different interaction techniques has not been studied explicitly. In the user studies, subjects were allowed to practice until they felt acquainted with the technique. For this there was a dedicated practice task which the users could repeat. Time spent on practicing could give a hint on ease of learning. This aspect is also related to the naturalness of the device.

From Aliakseyeu et al. [ASMR02], we have that the naturalness of an interaction device can be estimated by comparing it to how we interact with the real world. In general, a device meets the requirement for naturalness if the atomic actions provided by the device match the atomic actions required to perform a task. If the device fails to provide the atomic actions, for example the device requires the user to switch between pairs of DOF, it is considered unnatural. They [ASMR02] adopt five design guidelines for natural interaction devices:

- Two-handed interaction is preferred to one-handed interaction.
- Visual feedback is important for creating a feeling of spatial awareness.
- The action and perception space should coincide.
- Minimal use of intrusive devices, such as head-mounted displays, should be preferred.
- Wireless props preferred to wired ones.

From this it can be seen that a camera tracked mobile phone using AR should achieve a high level of naturalness. Since only one hand is needed to operate the device, the other hand can be used for interaction as described in the bimanual interaction conditions; hence, it supports two-handed interaction. It gives visual feedback and in an AR scenario, the action and perception spaces coincide[11]. There are no other devices required for either viewing or interaction. Furthermore, it requires no wiring.

[11]This is not entirely true when the keypad is used for input though the keypad and display are tightly coupled.

Figure 3.6: Virtual scene assembly using phone motion. Placing a crosshair and clicking a button allowed users to select pieces and manipulate them in 3D using phone motion. If the pieces were placed sufficiently close they were aligned and attached to each other.

3.2.7 Example application: AR LEGO

To demonstrate 6DOF interaction, a simple virtual LEGO© application was implemented (Figure 3.6). In this application the user can build structures by attaching virtual LEGO© bricks to each other in any configuration that would be possible with the physical counterpart. The virtual bricks form sub-structures when attached to each other. These sub-structures can be treated as a group by selecting the bottom brick.

When a brick is selected using crosshairs, it is detached from the brick below and can be moved freely (isomorphic position control). If other bricks are attached directly or indirectly to the selected brick, they will remain fixed in the local coordinate system of the selected brick. Once the brick is released, the application checks if the released piece is positioned within the margin of error to be attached to another piece. If so, the released brick, and the pieces for which it is a base, will be aligned with the new base brick. A grid restricts the transformations, making it easy to attach one piece on top of another as expected from the physical counterpart.

The keypad interface is used in parallel with motion-based input, with transformation increments and decrements adapted to the grid step size. The selected brick is rotated 90 degrees for each update and translation is made one grid step per frame update. After each update, there is a check for attachment.

The phone gives haptic feedback on detachment and attachment events by vibrating when bricks are joined or separated. No continuous collision detection has been implemented; otherwise, vibration could be used to indicate collision between the selected bricks and the rest of the scene. Audio feedback would also be useful. Research on such multi-sensory feedback will be presented in Section 3.3.1. Users found the virtual LEGO© application easy to use once they were informed about the crosshair and which button to press for a selected piece to remain fixed relative to the phone.

Figure 3.7: Playing AR Tennis. Players sit opposite each other with a set of markers between them. These markers define a common space in which ball is simulated and viewed through the phones. A virtual racket is defined to coincide with the phone and, to hit the ball, a user positions the phone so that the virtual racket intersects the ball's trajectory. Multi-sensory feedback is given to indicate a hit, and ball data is sent over Bluetooth

3.3 Collaborative AR

Tracking more than one device in the same coordinate frame enables collaborative AR where the same scene is viewed from different angles and updates to the virtual content are broadcast to all participating devices. Research challenges include sharing scene data and investigating potential benefits from using AR.

3.3.1 AR Tennis

Paper **I** describes the first collaborative AR application on a mobile phone. Here a tennis game was developed to research awareness and multi-sensory feedback. Tennis was chosen because it can be played in either a competitive or cooperative fashion, awareness of the other player is helpful, it requires only simple graphics, and it is a game that most people are familiar with. When playing tennis, hitting the ball is an essential event that needs to be perceived by the player; hence, it is also suitable for studying different feedback modalities.

The application is based on the platform presented earlier. In addition to tracking and graphics, a Bluetooth peer-to-peer layer was developed to transfer game data between two devices. Symbian API calls for vibration feedback were used together with sound playback capabilities to provide different feedback configurations for events such as hitting the ball or the ball bouncing off the ground or hitting the net.

A virtual tennis ball is simulated in marker space where a tennis court with a net constitutes the shared scene as can be seen in Figure 3.7. Interaction is 6DOF and based on a racket metaphor

Figure 3.8: Experimental setup. From left: face-to-face condition; upper image shows video feedback while in the lower image, video is turned off; non face-to-face condition where the players are separated by a black cloth.

where the phone itself is used to hit the ball. A virtual racket is defined in the x,y-plane of the camera coordinate system thus coinciding with the phone itself. If there is a collision between the ball and a racket, the ball direction is reversed and sent to the other player together with its position, and multi-sensory feedback is given simultaneously. The game is synchronized in this way every time the ball is hit or served. The game ran at about 8 frames per second with tracking, physics, graphics, audio, vibration and Bluetooth enabled. Due to this rather low tracking performance, ball speed is not affected and hence only the position and not the velocity of the phone is used as input. With considerably faster tracking or using accelerometers, a more accurate racket metaphor would be possible.

AR is interesting for face-to-face collaborative work because it allows the collaborators to be aware of each other while interacting with computer generated information. An experiment was conducted to see if this was also the case for mobile phone AR. The study compared three different configurations (Figure 3.8):

1. Face-to-face AR
2. Face-to-face non-AR
3. Non-face-to-face gaming

In the first configuration the users could see each other as well as having video see-through devices, while in the second configuration, only graphics was displayed. The third configuration provided no information about the other player's actions besides game state.

The task was for a pair of subjects to work together to achieve the highest number of consecutive ball bounces over the net during three minutes. Results showed no statistically significant difference between conditions; however, questionnaire survey results showed significant differences. While the configurations were perceived as equally enjoyable, users felt that is was much easier to be aware of what their partner was doing in the face-to-face AR condition with the live video background than in the other two conditions which had no video background. A ranking

of the three conditions in order of how easy it was to work together showed that almost all users preferred the AR condition. These results show that an AR interface does indeed enhance the face-to-face gaming experience on mobile phones and they confirm that AR increases perceived awareness compared to non-AR, though it does not seem to increase actual performance.

With its small field of view, a mobile phone can only provide limited visual feedback. This makes it interesting to investigate complementary feedback, primarily audio and haptic. Audio feedback consists of a racket sound being played when the ball is hit or served. The built-in vibration utility makes it easy to provide vibro-tactile feedback for the same event.

A second user study was conducted to compare the four multi-sensory configurations resulting from enabling and disabling audio and haptic. The task was as before to maximize ball bounces over one minute. Once again no statistically significant differences could be found across conditions for this quantitative approach. Qualitative feedback on the other hand showed that users felt that they were more aware of when they or their partner had hit the ball when both audio and haptic feedback was provided. Audio only was perceived to be more important than haptic only. This can be attributed to it being a shared cue, rather than haptics which can only be experienced by one person. Multi-sensory feedback also made the game significantly more enjoyable. A ranking of the four conditions in order of how easy it was to work together showed that almost all users preferred the condition where both audio and haptic feedback was provided. These results show that users feel that multi-sensory output is important in face-to-face AR gaming.

From observations it was clear that tracking was far from perfect; however, users adapted their behavior to reduce tracking problems, for example by limiting the rotation or by moving the phone very fast and then holding it still for a few frames. The latter behavior is suitable for marker-based tracking, which calculates absolute position in contrast to simpler motion flow algorithms.

At the 2005 International Mobile Gaming Awards[12], AR Tennis won both the Grand Prix and the Best Technical Achievement award. It has also been shown at several conference and museum exhibitions around the world and has been used by thousands of people. This success shows that a mobile phone game consisting of only a few polygons can be highly enjoyable if motion-based input is used in combination with AR and multi-sensory feedback. Motion-based input enables familiar metaphors to be used, in this case the racket metaphor, which might result in faster learning.

One important experience from exhibiting AR Tennis is the easy setup of a shared space, using the marker-based approach taken in this thesis. Simply placing a paper marker on a well lit table is enough to establish a common coordinate system. Realizing a similar game with natural feature-based tracking would require both devices to agree at start-up on how the scene space is defined. An instrumented environment would be much more expensive and time-consuming to configure.

[12] *www.imgawards.com*

Figure 3.9: Four devices viewing and manipulating the same scene. CMAR ViSION, a PC client for high quality output on large displays. A group collaborating on furnishing an apartment.

3.3.2 CMAR: Collaborative Mobile Augmented Reality

While AR Tennis showed the feasibility of collaborative AR on mobile phones, it was restricted to two users and interaction was limited to momentarily input. CMAR on the other hand was developed to support collaboration between up to eight participants and with continuous 6DOF interaction. The purpose was to demonstrate and study collaboration between several devices and with full 3D interaction capabilities. The main research challenge was how to efficiently share the scene representation between devices connected via Bluetooth and propagate 6DOF user inputs in real-time and without loss of data. Another challenge was to indicate and manage object ownership to prevent two participants manipulating the same object.

CMAR is based on a shared scene graph, which makes it easy to synchronize scene data across devices by sending updates to the scene graph nodes. There are several existing scene graphs for PCs, but these were considered too complex to port to mobile phones. Instead, a lightweight scene graph which uses OpenGL ES functionalities was implemented. In a CMAR network, up to eight participants are connected via Bluetooth, with one device acting as server, broadcasting updates made to the scene graph. Each node has a local copy of the scene graph and commits to all received updates. Updates included rotation, translation and selection events and must be visible on all connected devices in real-time. As a solution to enable this, a protocol containing a number of update packages was designed. To indicate ownership of objects, each user was assigned a color and an object selected by a user was highlighted by a semi-transparent bounding box of this color. Selection was based on sampling the alpha value of the central pixel, indicated by crosshairs. Once selected, an object could be translated using isotonic position control while keypad input was used for rotation. Input made by one device was broadcast and perceived in real-time on all nodes. For more details see paper **VI**.

The current version is based on BoomslangBT[13], a C++ Bluetooth library for mobile devices. It requires all devices to be present at startup. For future versions of CMAR, it would be interesting to add support for late joining of devices to an established network.

[13] *boomslangstudios.com/NewSite/boomslangbt.htm*

3.3.3 Example application: Collaborative Furnishing

As an example of the CMAR platform, a furnishing application was implemented where users can collaborate to furnish an apartment (see Figure 3.9). A piece of paper having the apartment plan and a marker was placed on a table around which the participants were seated. They were provided a set of 3D furniture that could be selected, translated along the ground plane, and rotated around the vertical axis.

One CMAR client ran on a laptop with a DLP projector connected to it. This PC client ran a scene graph implemented in OpenGL and featured more detailed objects than the phones. Running a client on a PC restricted the number of participating phones to seven. Further details on using a PC as group display are given in Section 3.4.1. This example application also served as a test case for the CMAR platform to obtain user feedback on its interaction styles and functionalities. An informal, qualitative evaluation showed that users thought it was easy to manipulate objects and collaborate with others.

3.4 AR in Ubiquitous Computing

It is not only mobile phones that have benefited from advancement in electronics design and production. Connected devices, such as sensors and displays, have become increasingly affordable and integrated into our environment. This section presents research where an AR capable mobile phone interacts with an instrumented environment for input and output. Main challenges include data transfer and device integration.

3.4.1 CMAR ViSION

A mobile phone screen is usually only a couple of inches in diameter and the 3D rendering performance far from what can be achieved on a PC, despite the introduction of GPUs. Connecting a mobile phone to a large display, such as a HD plasma TV or a projector, is hence an attractive solution for providing several people with a view of the 3D scene without requiring them to get very close to each other. Phones' relatively limited rendering capabilities also make it interesting to explore how the superior visual output of a PC can be exploited by phones.

As described earlier, CMAR allows a client to be run on a PC for output on a large display. A network with a PC, several mobile phones and a large display enables mobile phones and PCs to play different roles in a collaborative setup. This allows realization of a setup where mobile phones act as personal displays for private data and as interaction devices, while a screen connected to a PC CMAR client - called CMAR ViSION - acts as group display visualizing the scene using high quality rendering but no AR. The CMAR ViSION's virtual camera parameters can be set interactively to specify the scene view. Though CMAR ViSION only allows one person to interact with the camera view, manipulation and visualization of the scene is still possible for all participants through their phones.

Output on large screens connected to computers requires data to be shared between nodes as before, and it also requires different object representations to manage the big performance

Figure 3.10: A concept where a mobile phone is used to visualize humidity data derived from an embedded sensor network was realized by placing sensors in a marker coordinate frame. This allowed a resulting live image to be registered with the element of interest.

difference between a phone and a PC with a GPU. The proposed solution is that the PC client shares the same scene graph as the mobile phones but acts only as receiver of data. 3D objects on the PC have a higher level of detail in order to exploit the advantages in rendering performance offered by computers.

The purpose of this research was not to regard the mobile phone primarily as a an interaction device for large screens - as many others already have, but rather to see if such a group display was of advantage in a collaborative AR setup. In the informal evaluation, with a DLP projector, users were positive about the group display.

3.4.2 Visualization of Sensor Data

By placing sensors in the environment, it is possible to obtain data to synthesize "live" graphics. If AR is used, graphics can be registered with the elements inside which sensors are embedded. Arranging sensors in a 2D grid supports an approach where data from individual sensors emerges as pixels in an image when it is translated into color values.

Paper **VII** introduces a system that merges handheld AR, sensor networks and scientific visualization, providing a new approach for inspection of interior states within elements (Figure 3.10). Data collected from an embedded wireless sensor matrix is used to synthesize graphics in real-time and AR allows it to be viewed in its context through the phone. The technical challenge is to obtain the data and produce a graphical representation to provide insight into the interior state of the element. The main objective was to evaluate this novel concept and to research which visualization techniques users preferred.

The main components of the setup are a network of ZigBee enabled humidity sensors and a camera phone with 3D rendering capability. These phones are not yet equipped with ZigBee technology, which is why a temporary communication solution has been designed, consisting of

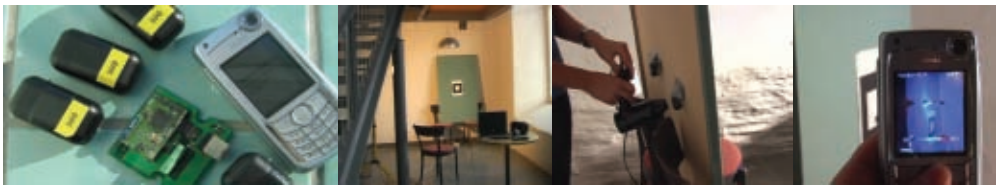

Figure 3.11: Proof-of-concept experiment using a steaming kettle and a hairdryer to affect sensors. From left: sensors, controller and mobile phone; experimental setup with test wall, marker and laptop; sensors being affected by a hairdryer; real-time visualization.

both ZigBee and Bluetooth combined using a laptop as router. Humidity as the sensed property was chosen because it is a major problem for the construction industry. Building maintenance could potentially benefit from new on-site inspection tools, like the proposed system.

The sensors measure the relative humidity at the location of the sensor, providing scalar data from a discrete set of coplanar measurement points in 3D. A continuous representation is obtained by interpolating measurement point values. The interpolated values are then mapped to corresponding RGBA color values using one transfer function for each channel. The resulting color coded texture provides the user with an overview of the humidity as well as its distribution. The alpha channel is necessary for making redundant pixels transparent and thus allowing the user to see the background.

An experiment was carried out using a Nokia 6680, a network of four ZigBee enabled humidity sensors, and a laptop to which a ZigBee coordinator unit was connected via USB-interface. The sensors were attached to the back of a test wall, forming a two by two grid positioned at the center of the wall and located in a marker coordinate system. When the sensors were influenced using a steaming kettle and a hair dryer, the response was instantaneous and the observer given an immediate visual feedback in the visualization. Figure 3.11 shows the components and experimental setup.

The experiment demonstrated the possibility of synthesizing an image from live sensor data and having it registered with the element in which the sensors are embedded. Further, it shows that sensor state changes can be perceived almost immediately by the system user, in both amplitude and location. The aim of the experiment was to explore the concept rather than to perform a formal evaluation of the system.

Several visualization techniques and interface functions were implemented and evaluated in a user study. One of these functions enabled the user to freeze the current view. This snapshot metaphor allowed the user to assume a more natural pose when analyzing the data and all of the participants found it easier to interact with the visualization when frozen. This result indicates that freezing the view is an important function to reduce fatigue in applications where analysis of a scene is of importance.

3.4.3 LUMAR

LightSense [Olw06] is a platform for outside-in tracking of devices on a semitransparent surface. With a camera mounted on the back of the surface, it is possible to locate a mobile phone's LED light and track phone movements on the surface. In paper **VIII**, LightSense has been combined with the AR platform presented in this thesis to form a hybrid tracking solution called LUMAR, where the mobile phone camera is used for egocentric 3D tracking above the surface. When the device is placed on the surface, and the ability to use the camera is disabled, the device is tracked in 2D by the exocentric system.

LUMAR allows a user to browse a three-layered information space consisting of print media, 2D multimedia and 3D interactive graphics. This combination is important since all three media types have their individual strengths and together contribute to empower our human perception. Such benefit from merging information spaces has previously been demonstrated in projects like MagicBook [BKP01] and Tangible Bits [IU97]. The main challenges are to merge the two tracking solutions as transparently as possible and facilitate easy transition between information spaces. There must also be a consistent interaction metaphor.

The hybrid tracking system is realized by placing markers on the LightSense surface and whenever a marker is detected, the application switches to 3D mode and the 3D information space is navigated. This works since markers are not detectable when the phone is placed on the surface and hence no accidental transition to 3D mode is possible. However, the LED can be detected when the phone is lifted, though rotations cannot be estimated. There will thus be a seam between modes unless the phone is lifted without being rotated. All interaction is based on phone motion: in the 2D mode, the phone is used as a tangible cursor in the physical information space that the print represents (in this way, 2D multimedia registered with the print can be browsed); lifting the device, the user moves into 3D mode where the tracking transitions into full isomorphic 6DOF interaction.

The general LUMAR application scenario consists of a high resolution print that gives an easily navigated overview and ensures that the tangibility of the media and its possible use as a traditional display is preserved. Subsequent 2D information layers are digital and viewed by placing the phone on the surface of the print. They can be either continuous, providing higher information density than the print for the same region of interest, or they can be segmented into areas, each associated with various kinds of 2D multimedia, such as videos or web pages. The 3D information layer serves as an augmentation of the print or a continuous 2D layer. Thus there is a close spatial relationship between 2D and 3D information. The phone acts as a looking-glass that magnifies information content.

3.4.4 Example application: Interactive Apartment Exploration

As an example application for the LUMAR system, an apartment ad scenario was implemented (Figure 3.12). This test case used an apartment plan and images of the interior obtained from an Internet real estate site. It demonstrates how three information spaces - each of the spaces containing valuable information - can be combined, thus fusing different media into a consistent representation.

Figure 3.12: LUMAR: Merging three information spaces. Printed material (left) is augmented by 2D media (center) and 3D graphics (right). Moving the phone across the surface loads relevant images and lifting it makes a transition to 3D space. This example demonstrates how three different, but highly relevant, media can be merged into a single model of an apartment, browsed using device motion only.

An enlarged print of an apartment plan was placed on the LightSense table and four fiducial markers were placed in empty spaces of the print, ensuring that they were not obscuring any details. One photograph was associated with each room, and a 3D model of the apartment was constructed based on the floor plan and images. In this case, no mapping between LightSense and ARToolKit coordinates was necessary; photos were associated with areas in LightSense coordinates, while the 3D model was registered with the 2D print.

Since the camera API did not provide sufficient control of the built-in flash, an external LED was attached for 2D tracking. Placing the phone over a room on the surface will load and display an image of the room. When the phone is moved across the surface, corresponding images are loaded automatically. When the phone is lifted and a marker is framed, a 3D model of the apartment is superimposed on the printed plan and registered with it in 3D. The user can now move the phone freely to view the 3D model from an arbitrary direction provided that at least one marker is fully visible. The registration makes it easy to identify which object in the 3D model corresponds to a certain legend in the plan. Putting the phone back on the surface makes the application switch to 2D mode, as soon as no marker is visible.

Chapter 4

Conclusions

Over the past two decades, significant research has been conducted on Augmented Reality. The rapid evolution of mobile phones with cameras and 3D rendering capabilities has made it possible to migrate AR technology to this platform and conduct research on how to best exploit its capabilities. This thesis has significantly contributed to advance the area of mobile phone AR by presenting research on the following key areas: tracking, interaction, collaborative applications and Ubiquitous Computing applications. This research is an important step to making mobile phones a generally available AR platform in the future.

Summary of Contributions

The following key research contributions were made:

Tracking
This thesis describes how 6DOF tracking technology was brought to off-the-shelf mobile phones. The resulting platform presented in Section 3.1 enabled development of applications where paper markers are used to register real and virtual information in real-time and in three dimensions - applications that can run on standard commercial devices. It should be noted that tracking fiducial markers is only one way to establish camera and hence display parameters and presented concepts are not limited to this tracking approach.

Interaction
Mobile phones have a different form factor from traditional devices used for AR. For this reason, previous research on AR interaction made on backpack configurations or on PDAs may not apply to the mobile phone platform. Section 3.2 presented research on 3D interaction where the motion of the device itself was of particular interest due to the tangibility of mobile phones. A broad range of techniques exploiting motion, finger and keypad input were implemented. From the user studies, it can be concluded that isomorphic position control, where a manipulated object undergoes the same transformation as the user's hand, was preferred when the task was to

translate objects. In the case of rotation, both user studies proved that isometric rate control, that is mapping joypad and button input to rotations, was the preferred interaction technique. Gesture interaction using the front camera turned out to be complex and limited by tracking range and performance. Although these studies are focused on AR, their contributions are not limited to this interface. Developers might also use these results for general mobile phone 3D applications that require 3D interaction although it should be noted that there is always a trade-off between general applicability of results and test realism.

Collaborative Applications
Research on face-to-face collaborative AR was presented in Section 3.3. A tennis game was developed to study users' awareness in a collaborative environment. Results from a user study showed that the AR interface did increase partner awareness compared to a non-AR interface. Further results showed that multi-sensory feedback also increased awareness, audio cues being the most important. This game further highlights the strengths of marker-based tracking due to the simple configuration requirements for setting up a shared space. The game used a unique object-based interaction metaphor where the phone was used as racket. For continuous interaction between up to eight participants, CMAR was developed based on a shared scene graph and a custom protocol. User study participants appreciated running a CMAR client on a PC to provide a VR view on a large screen.

Ubiquitous Computing Applications
Another research area identified to be important was to study mobile phone AR in an instrumented environment as described in Section 3.4. Sensors provided real-time data from which live images were synthesized and registered with a physical element. An experiment showed that change in sensor output was directly perceivable on the device. User study results and feedback indicated that this was a feasible concept tool for on-site inspection. Extending LightSense with marker tracking resulted in a three-layer information space where a mobile phone was used for browsing 2D multimedia and navigating 3D scenes. Motion alone is used for input.

Future Research

While the platform presented in this thesis has been successfully used for various experiments and applications, more research is still needed on core technology. Tracking is the key enabling technology for AR and consequently one of the hottest research topics. Many strategies have showed promising results, but no single approach suits all situations. The marker-based approach taken in this thesis has its limitations: markers need to be placed in the environment and must be visible at all times. On the other hand, it is very easy to print a marker and use it as a common reference frame as demonstrated in Section 3.3. Marker tracking will need to be assisted by other technologies to extend range and performance. Single camera SLAM[1] uses detected features to track the camera through space while simultaneously creating a 3D map of the environment.

[1] Simultaneous Localization And Mapping

Initiated by marker tracking, SLAM might be used to extend tracking range and also solve a major problem: occlusion of virtual imagery by real objects. Sensors such as accelerometers and gyros are steadily making their way into mobile phones. Such sensors might be used to stabilize marker tracking and reduce computational demands posed by computer vision algorithms.

For outdoor mobile phone AR, marker tracking is not an option due to unprepared environments and wide area use. Instruments such as GPS and gyros are necessary but not yet sufficient. While many mobile phones have GPS antennae, far fewer have gyros. Model-based optical tracking might provide a useful tool for outdoor mobile phone AR tracking. This approach matches video features to CAD models to calculate the camera pose. With the emergence of Earth browsers such as Google Earth, more and more of our cityscape has digital 3D equivalents. These might be used for large scale model-based tracking.

Information retrieval is another big challenge. AR content is often custom made for a specific application. For AR to be a useful tool, however, information must be retrieved from significantly bigger collections; ultimately the whole Internet. Information must be fetched based on the user's view and turned into graphical representations with a specific location in geospatial coordinates. So far, little research has been devoted to this since context aware information retrieval applications seldom use AR interfaces.

Content creation in AR is challenging because to register virtual content with the physical world, the content creator must have detailed knowledge about the consumer's physical environment. Research needs to be conducted to solve this fundamental problem. One approach will be to use Earth browsers to obtain the necessary knowledge. A digital equivalent of the real world can then be used for registration, removing the need for in-situ development. Work on this has begun [HA07], but was not considered mature enough for inclusion in this thesis.

All of the above research challenges apply to AR in general, indicating that this research area is still struggling with key enabling technologies. Looking specifically at mobile phones, there are several areas that must be addressed, including: the limited screen space, need for more evaluation studies, interaction metaphors, interaction in large workspaces etc. With adjustable zoom it would be interesting to study the relationship between software field of view (FOV) and physical FOV. The front facing camera allows the head to be tracked relative to the device. This might be exploited to optimize the AR experience. Important work on human perception and cognition in AR has been conducted on HMD configurations, but needs to be conducted for handheld AR platforms as well to gain more insight into human factors on this platform.

More research on interaction needs to be performed, optimizing existing interaction techniques and inventing new ones. Possible combinations of positioning and rotation techniques must be tested to identify the best interaction technique for 6DOF tasks. Also 12DOF interaction, where two sets of 6DOF interaction techniques support integral and separable input respectively, should be studied to further explore the potential of the mobile phone as a 3D input device.

It is hard to make predictions, especially about the future. But we have likely only seen the beginning of the mobile phone revolution. Cutting-edge mobile phones are expanding into traditional computer territories, and also into new areas where mobility and tangibility is essential. With its great potential, AR may very well be one of these areas and with major industry backing, new technologies and services will emerge and need to be explored.

Bibliography

[AB94] Ronald Azuma and Gary Bishop. Improving static and dynamic registration in an optical see-through hmd. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH'94)*, pages 197–204, 1994.

[ABD04] Daniel F. Abawi, Joachim Bienwald, and Ralf Dorner. Accuracy in optical tracking with fiducial markers: An accuracy function for artoolkit. In *Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 260–261, 2004.

[ACCH03] Mark Assad, David J. Carmichael, Daniel Cutting, and Adam Hudson. Ar phone: Accessible augmented reality in the intelligent environment. In *Proceedings of 2003 Australiasian Computer Human Interaction Conference (OZCHI'03)*, pages 232–235, 2003.

[ASMR02] Dzmitry Aliakseyeu, Sriram Subramanian, Jean-Bernard Martens, and Matthias Rauterberg. Interaction techniques for navigation through and manipulation of 2d and 3d data. In *Proceedings of the workshop on Virtual environments 2002 (EGVE'02)*, pages 179–188, 2002.

[Azu97] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[BBRS06] Rafael Ballagas, Jan Borchers, Michael Rohs, and Jennifer G. Sheridan. The smart phone: A ubiquitous input device. *IEEE Pervasive Computing*, 5(1):70–77, 2006.

[BE06] Oliver Bimber and Andreas Emmerling. Multifocal projection: A multiprojector technique for increasing focal depth. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):658–667, 2006.

[BFO92] Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi. Merging virtual objects with the real world: seeing ultrasound imagery within the patient. *ACM SIGGRAPH Computer Graphics*, 26(2):203–210, 1992.

[BGH02] Doug A. Bowman, Joseph L. Gabbard, and Deborah Hix. A survey of usability evaluation in virtual environments: classification and comparison of methods. *Presence: Teleoperators and Virtual Environments*, 11(4):404–424, 2002.

[BGW+02] Oliver Bimber, Stephen M. Gatesy, Lawrence M. Witmer, Ramesh Raskar, and L. Miguel Encarnação. Merging fossil specimens with computer-generated information. *IEEE Computer*, 35(9):25–30, 2002.

[BKLP04] Doug Bowman, Ernst Kruijff, Jr. Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004.

[BKP01] Mark Billinghurst, Hirkazu Kato, and Ivan Poupyrev. The magicbook - moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, 2001.

[BN95] Michael Bajura and Ulrich Neumann. Dynamic registration correction in augmented-reality systems. In *Proceedings of the Virtual Reality Annual International Symposium (VRAIS'95)*, pages 189–196, 1995.

[BRS05] Rafael Ballagas, Michael Rohs, and Jennifer G. Sheridan. Sweep and point and shoot: phonecam-based interactions for large public displays. In *CHI '05 extended abstracts on Human factors in computing systems*, pages 1200–1203, 2005.

[BSP+93] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93)*, pages 73–80, 1993.

[CM92] T.P. Caudell and D.W. Mizell. Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, pages 659–669, 1992.

[CWG+03] Adrian David Cheok, Fong Siew Wan, Kok Hwee Goh, Xubo Yang, Wei Liu, and Farzam Farbiz. Human pacman: a sensing-based mobile entertainment system with ubiquitous computing and tangible interaction. In *Proceedings of the 2nd workshop on Network and system support for games (NETGAMES'03)*, pages 106–117, 2003.

[DA05] Stephan Drab and Nicole M. Artner. Motion detection as interaction technique for games & applications on mobile devices. In *Proceedings of Pervasive Mobile Interaction Devices (PERMID 2005) Workshop at the Pervasive 2005*, pages 52–55, 2005.

[EBM+97] S. R. Ellis, F. Breant, B. Manges, R. Jacoby, and B. D. Adelstein. Factors influencing operator interaction with virtual objects viewed via head-mounted see-through displays: viewing conditions and rendering latency. In *Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS'97)*, pages 138–145, 1997.

[EHLO07] Eva Eriksson, Thomas Riisgaard Hansen, and Andreas Lykke-Olesen. Reclaiming public space: designing for public interaction with private devices. In *Proceedings of the 1st international conference on Tangible and embedded interaction (TEI'07)*, pages 31–38, 2007.

[FGH+00] Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrison, Thomas P. Moran, and Roy Want. Embodied user interfaces for really direct manipulation. *Communications of the ACM*, 43(9):74–80, 2000.

[FHSH06] Bernd Fröhlich, Jan Hochstrate, Verena Skuk, and Anke Huckauf. The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI'06)*, pages 191–199, 2006.

[Fit93] George W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7):39–49, 1993.

[FMHW97] Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers (ISWC'97)*, pages 74–81, 1997.

[FMS93] Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, 1993.

[FP00] Bernd Fröhlich and John Plate. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'00)*, pages 526–531, 2000.

[Frö05] Bernd Fröhlich. The quest for intuitive 3d input devices. In *HCI International*, 2005.

[GKRS01] Christian Geiger, Bernd Kleinnjohann, Christian Reimann, and Dirk Stichling. Mobile ar4all. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, pages 181–182, 2001.

[HA07] Anders Henrysson and Miroslav Andel. Augmented earth: Towards ubiquitous ar messaging. In *ICAT'07 (To appear)*, 2007.

[Han97] Chris Hand. A survey of 3d interaction techniques. *Computer Graphics Forum*, 16(5):269–281, 1997.

[HELO05] Thomas Riisgaard Hansen, Eva Eriksson, and Andreas Lykke-Olesen. Mixed interaction space: designing for camera based interaction with mobile devices. In *CHI '05 extended abstracts on Human factors in computing systems*, pages 1933–1936, 2005.

[HELO06] Thomas Riisgaard Hansen, Eva Eriksson, and Andreas Lykke-Olesen. Use your head: exploring face tracking for mobile interaction. In *CHI '06 extended abstracts on Human factors in computing systems*, pages 845–850, 2006.

[HFG+98] Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '98)*, pages 17–24, 1998.

[HFT+99] Tobias Höllerer, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. Exploring mars: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6):779–785, 1999.

[HJK06] Jane Hwang, Jaehoon Jung, and Gerard Jounghyun Kim. Hand-held virtual reality: a feasibility study. In *Proceedings of the ACM symposium on Virtual reality software and technology (VRST'06)*, pages 356–363, 2006.

[HMSC05] Antonio Haro, Koichi Mori, Vidya Setlur, and Tolga Capin. Mobile camera-based adaptive viewing. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia (MUM'05)*, pages 78–83, 2005.

[HO04] Anders Henrysson and Mark Ollila. Umar: Ubiquitous mobile augmented reality. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia (MUM'04)*, pages 41–45, 2004.

[HPG05] Martin Hachet, Joachim Pouderoux, and Pascal Guitton. A camera-based interface for interaction with mobile handheld computers. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games (I3D'05)*, pages 65–72, 2005.

[HPSH00] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. Sensing techniques for mobile interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST'00)*, pages 91–100, 2000.

[HSH05] Jari Hannuksela, Pekka Sangi, and Janne Heikkila. A vision-based approach for controlling user interfaces of mobile devices. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 71, 2005.

[HTP+97] Ken Hinckley, Joe Tullio, Randy F. Pausch, Dennis Proffitt, and Neal F. Kassell. Usability analysis of 3d rotation techniques. In *ACM Symposium on User Interface Software and Technology (UIST'97)*, pages 1–10, 1997.

[HW05] Mika Hakkarainen and Charles Woodward. Symball: camera driven table tennis for mobile phones. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '05)*, pages 391–392, 2005.

[IG05] J. Edward Swan II and Joseph L. Gabbard. Survey of user-based experimentation in augmented reality. In *Proceedings of 1st International Conference on Virtual Reality, HCI International 2005*, 2005.

[IU97] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '97)*, pages 234–241, 1997.

[Jr.03] Joseph J. LaViola Jr. An experiment comparing double exponential smoothing and kalman filter-based predictive tracking algorithms. In *Proceedings of the IEEE Virtual Reality 2003 (VR'03)*, pages 283–284, 2003.

[JSMJ94] Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen Jr. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction*, 1(1):3–26, 1994.

[KB99] Hirokazu Kato and Mark Billinghurst. 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 (IWAR'99)*, pages 85–94, 1999.

[KBP+00] Hirkazu Kato, Mark Billinghurst, Ivan Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR'00)*, pages 111–119, 2000.

[KM03] Rob Kooper and Blair MacIntyre. Browsing the real-world wide web: Maintaining awareness of virtual information in an ar information space. *International Journal of Human Computer Interaction*, 16(3):425–446, 2003.

[LB07] T. Langlotz and Oliver Bimber. Unsynchronized 4d barcodes. In *International Symposium on Visual Computing, 2007 (To appear)*, 2007.

[LBK04] Gun A. Lee, Mark Billinghurst, and Gerard Jounghyun Kim. Occlusion based interaction methods for tangible augmented reality environments. In *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry (VRCAI'04)*, pages 419–426, 2004.

[LDL06] Xu Liu, David Doermann, and Huiping Li. Imaging as an alternative data channel for camera phones. In *Proceedings of the 5th international conference on Mobile and ubiquitous multimedia (MUM'06)*, page 5, 2006.

[Min95] Mark R. Mine. Virtual environment interaction techniques. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[MK94] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12):1321–1329, 1994.

[MKBP02] D. Mogilev, K. Kiyokawa, M. Billinghurst, and J. Pair. Ar pad: an interface for face-to-face ar collaboration. In *CHI '02 extended abstracts on Human factors in computing systems*, pages 654–655, 2002.

[MLB04] Mathias Möhring, Christian Lessig, and Oliver Bimber. Video see-through ar on consumer cell-phones. In *Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 252–253, 2004.

[MSSU04] Anil Madhavapeddy, David Scott, Richard Sharp, and Eben Upton. Using camera-phones to enhance human-computer interaction. In *Sixth International Conference on Ubiquitous Computing (Adjunct Proceedings: Demos)*, 2004.

[NIH01] Joseph Newman, David Ingram, and Andy Hopper. Augmented reality in a wide area sentient environment. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, pages 77–86, 2001.

[Olw06] Alex Olwal. Lightsense: enabling spatially aware handheld interaction devices. In *Proceedings of Fifth IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'06)*, pages 119–122, 2006.

[PRS04] Volker Paelke, Christian Reimann, and Dirk Stichling. Kick-up menus. In *CHI '04 extended abstracts on Human factors in computing systems*, pages 1552–1552, 2004.

[PT02] Wayne Piekarski and Bruce H. Thomas. Tinmith-hand: Unified user interface technology for mobile outdoor augmented reality and indoor virtual reality. In *Proceedings of the IEEE Virtual Reality 2002 (VR'02)*, pages 287–288, 2002.

[PW03] Wouter Pasman and Charles Woodward. Implementation of an augmented reality system on a pda. In *Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, pages 276–277, 2003.

[RDB01] J.P. Rolland, L. Davis, and Y. Baillot. *A Survey of Tracking Technology for Virtual Environments*, chapter 3, pages 67–112. Lawrence Erlbaum Associates, Inc., 2001.

[RE07] Michael Rohs and Georg Essl. Sensing-based interaction for information navigation on handheld displays. In *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2007.

[Rek95] Jun Rekimoto. The magnifying glass approach to augmented reality systems. In *Proceedings of International Conference on Artificial Reality and Tele-Existence '95 / Conference on Virtual Reality Software and Technology '95 (ICAT/VRST'95)*, pages 123–132, 1995.

[Rek96a] Jun Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST'96)*, pages 167–168, 1996.

[Rek96b] Jun Rekimoto. Transvision: A hand-held augmented reality system for collaborative design. In *International Conference on Virtual Systems and Multimedia (VSMM'96)*, pages 85–90, 1996.

[Rek98] J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Proceedings of the Third Asian Pacific Computer and Human Interaction (APCHI '98)*, pages 63–69, 1998.

[RG04] Michael Rohs and Beat Gfeller. Using camera-equipped mobile phones for interacting with real-world objects, 2004.

[RN95] Jun Rekimoto and Katashi Nagao. The world through the computer: computer augmented interaction with real world environments. In *Proceedings of the 8th annual ACM symposium on User interface and software technology (UIST '95)*, pages 29–36, 1995.

[Roh04] Michael Rohs. Real-world interaction with camera-phones. In *Proceedings of 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, number 3598 in Lecture Notes in Computer Science (LNCS), pages 74–89, 2004.

[RS00] Holger T. Regenbrecht and R. Specht. A mobile passive augmented reality device - mpard. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'00)*, pages 81–84, 2000.

[RS01] Gerhard Reitmayr and Dieter Schmalstieg. Mobile collaborative augmented reality. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, pages 114–123, 2001.

[RZ05] Michael Rohs and Philipp Zweifel. A conceptual framework for camera phone-based interaction techniques. In *Proceedings of Third International Conference on Pervasive Computing (PERVASIVE 2005*, number 3468 in Lecture Notes in Computer Science (LNCS), pages 171–189, 2005.

[Sch05] Dieter Schmalstieg. Augmented reality techniques in games. In *Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '05)*, pages 176–177, 2005.

[SD91] M. Slater and A. Davidson. Liberation from flatland: 3d interaction based on the desktop bat. In *Eurographics '91*, pages 209–221, 1991.

[SFH+02] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L. Miguel Encarnação, Michael Gervautz, and Werner Purgathofer. The studierstube augmented reality project. *Presence: Teleoperators and Virtual Environments*, 11(1):33–54, 2002.

[SFSG96] Dieter Schmalstieg, Anton Fuhrmann, Zsolt Szalavári, and Michael Gervautz. Studierstube - collaborative augmented reality. In *Proceedings of Collaborative Virtual Environments '96*, 1996.

[SHC+96] Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, and Mark A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96)*, pages 429–438, 1996.

[Sho94] Ken Shoemake. Arcball rotation control. *Graphics gems IV*, pages 175–192, 1994.

[SI97] David Small and Hiroshi Ishii. Design of spatially aware graspable displays. In *CHI '97 extended abstracts on Human factors in computing systems*, pages 367–368, 1997.

[SI00] Sriram Subramanian and Wijnand IJsselsteijn. Survey and classification of spatial object manipulation techniques. In *Proceedings of OZCHI 2000, Interfacing Reality in the New Millennium*, pages 330–337, 2000.

[SLSP00] Thad Starner, Bastian Leibe, Brad Singletary, and Jarrell Pair. Mind-warping: towards creating a compelling collaborative augmented reality game. In *Proceedings of the 5th international conference on Intelligent user interfaces (IUI'00)*, pages 256–259, 2000.

[SMR+97] Thad Starner, Steve Mann, Bradley J. Rhodes, Jeffrey Levine, Jennifer Healey, Dana Kirsch, Rosalind W. Picard, and Alex Pentland. Augmented reality through wearable computing. *Presence*, 6(4):386–398, 1997.

[Stu92] David Joel Sturman. *Whole-hand input*. PhD thesis, Massachusetts Institute of Technology, 1992.

[Sun01] Martin Sundin. *Elastic computer input control in six degrees of freedom*. PhD thesis, ETH Zürich, 2001. Diss. Nr. 14134.

[Sut68] Ivan E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 757–764, 1968.

[TCD+00] Bruce H. Thomas, Benjamin Close, John Donoghue, John Squires, Phillip De Bondi, Michael Morris, and Wayne Piekarski. Arquake: An outdoor/indoor augmented reality first person application. In *Proceedings of the Fourth International Symposium on Wearable Computers (ISWC'00)*, pages 139–146, 2000.

[TDP+98] Bruce H. Thomas, Victor Demczuk, Wayne Piekarski, David Hepworth, and Bernard K. Gunther. A wearable computer system with augmented reality to support terrestrial navigation. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers (ISWC'98)*, pages 168–171, 1998.

[UK95] Michihiro Uenohara and Takeo Kanade. Vision-based object registration for real-time image overlay. In *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed '95)*, pages 13–22, 1995.

[War90] Colin Ware. Using hand position for virtual object placement. *The Visual Computer: International Journal of Computer Graphics*, 6(5):245–253, 1990.

[WB03] Daniel Wagner and Istvan Barakonyi. Augmented reality kanji learning. In *Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, pages 335–336, 2003.

[Wei91] Mark Weiser. The computer for the twenty-first century. *Scientific American*, 265(3):94–104, 1991.

[WMB03] Eric Woods, Paul Mason, and Mark Billinghurst. Magicmouse: an inexpensive 6-degree-of-freedom mouse. In *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE'03)*, pages 285–286, 2003.

[WPLS05] Daniel Wagner, Thomas Pintaric, Florian Ledermann, and Dieter Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Proceedings of Third International Conference on Pervasive Computing (Pervasive 2005)*, volume 3468 of *Lecture Notes in Computer Science*, pages 208–219, 2005.

[WRZ07] Stefan Winkler, Karthik Rangaswamy, and ZhiYing Zhou. Intuitive user interface for mobile devices based on visual motion detection. In Reiner Creutzburg, Jarmo Takala, and Jianfei Cai, editors, *Proceedings of SPIE Multimedia on Mobile Devices 2007*, volume 6507, 2007.

[WS03] Daniel Wagner and Dieter Schmalstieg. First steps towards handheld augmented reality. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC'03)*, pages 127–135, 2003.

[WS07] Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, 2007.

[WSB06] Daniel Wagner, Dieter Schmalstieg, and Mark Billinghurst. Handheld ar for collaborative edutainment. In *Proceedings of 16th International Conference on Artificial Reality and Telexistence (ICAT'06)*, volume 4282 of *Lecture Notes in Computer Science*, pages 85–96, 2006.

[WZC06] Jingtao Wang, Shumin Zhai, and John Canny. Camera phone based motion sensing: interaction techniques, applications and performance study. In *Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST'06)*, pages 101–110, 2006.

[Yee03] Ka-Ping Yee. Peephole displays: pen interaction on spatially aware handheld computers. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'03)*, pages 1–8, 2003.

[Zha95] Shumin Zhai. *Human Performance in Six Degree of Freedom Input Control*. PhD thesis, University of Toronto, 1995.

[Zha98] Shumin Zhai. User performance in relation to 3d input device design. *SIGGRAPH Computer Graphics*, 32(4):50–54, 1998.