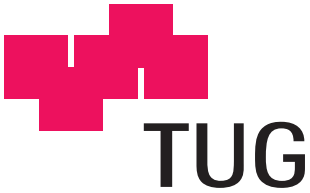

Elias Tappeiner

Urban Pointing: Browsing Situated Media using accurate Pointing Interfaces

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Computer Science

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr. techn. Tobias Langlotz
Assoc.Prof. Dipl.-Ing. Dr. techn. Holger Regenbrecht
University of Otago - New Zealand

Univ.-Prof. Dipl.-Ing. Dr. techn. Dieter Schmalstieg
Graz University of Technology - Austria

Graz, Austria, Oct. 2015

Abstract

Nowadays, digital media can be linked in 3D to physical objects or locations and is often referred to as situated media. Situated media allows new ways to browse and access digital information. The purpose of this thesis is to build interface concepts to browse accurately placed situated media, on mobile devices. First, as a reference interface, we developed a prototype of an Augmented Reality browser. Augmented Reality browsers blend the media content accurately registered onto the camera stream, where the camera stream represents a live vision of the physical world. Second, inspired by Science Fictions movies, we implemented a novel pointing interface for situated media. The situated media is queried by pointing with the mobile device instead of the camera. Third, we introduced the novel reflected Augmented Reality browser interface combining the interface concept of an Augmented Reality browser with the ergonomics of the media pointer.

Besides conceptualization, we implemented prototypes for all introduced concepts by adapting state-of-the-art vision-based tracking technology for outdoor environments, initially developed for Augmented Reality. We highlight the changes that are required for working with the different interfaces and conducted a technical evaluation to provide insights into the performance. We conclude this work by presenting hypothesis targeting usability, ergonomics and social implications of our interfaces for an upcoming user study.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

The text document uploaded to TUGRAZonline is identical to the presented master's thesis dissertation.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Ort	Datum	Unterschrift

Acknowledgments

Als Erstes möchte ich mich bei meinen Betreuern Tobias Langlotz und Holger Regenbrecht bedanken, die mich während meiner Zeit in Neuseeland geleitet, motiviert und gefordert haben. Sie waren auch abseits der Universität immer hilfsbereit. Wann immer das Wetter gut war und es zeitlich passte, war Tobias auch für eine gute Surf-Session zu haben. Für die unbürokratische und zeitnahe Betreuung an der Universität in Graz möchte ich Dieter Schamlstieg danken. Vor allem aber auch dafür, dass es mir ermöglicht wurde, eine unvergessliche Zeit in Neuseeland zu verbringen. Neben meinen Betreuern danke ich natürlich meinen Studienkollegen, Freunden und Freundinnen, national und international, die mich schon mein ganzes Leben lang begleiten und auch bei denen, die ich während meiner Zeit in Graz, Portugal und Neuseeland kennen lernen durfte. All jene, die zu zahlreich sind, um sie namentlich zu nennen, haben mich immer motiviert und unterstützt. Ich möchte mich auch bei den Geschäftsführern und Mitarbeitern der Firma Codeflügel bedanken, bei der ich fast zwei Jahre lang arbeiten durfte. Alles, was ich in dieser Zeit lernen durfte, hat zum Erfolg und Abschluss dieser Arbeit beigetragen.
Anschließend möchte ich mich auch bei meinen Eltern Astrid und Siegfried bedanken, die mir dieses Studium nicht nur ermöglicht haben, sondern mir auch immer eine große Hilfe und Unterstützung waren. Dabei möchte ich mich besonders auch bei meinem Bruder Lukas bedanken, der bei dieser Arbeit, aber auch bei fast jedem mathematischen Problem, das mir während meines Studiums begegnet ist, weiter geholfen hat.
Zuletzt möchte ich noch meinem Opa danken. Seiner Zeit voraus, wusste er immer über den Wert von Bildung und eines Studiums Bescheid. Seine Finanzspritzen für den armen Studenten, aber vor allem seine unglaubliche Weisheit werde ich nie vergessen.
Danke an alle.

Contents

List of Figures

List of Tables

1 Introduction

Contents

1.1 Situated media

Nowadays the most common method to access digital information or media is by using a web browser. A web browser allows us to browse any public data in the web by just searching for it. Other methods to query this huge amount of information is by accessing it over a hyper link or directly, using the Uniform Resource Locator (URL) to the information. Exactly this method of querying data over a unique public file path, the URL, is the most spread form of information retrieval, nowadays.
However, in the last years, much information is not just linked to an URL, but enriched with additional meta data. A commonly added meta data type is the GPS location of the information. In many cases, the location from a GPS sensor is immediately added when the information is generated on a smartphone using the integrated sensors. This kind of information is in the literature often referred as situated media [13]. Basically, the access to the digital information or media is not an URL anymore but a location or a physical object on a location. Due to the diversity of mobile devices over the last years, finally, many commercial services exist that use such GPS- or geo-tagged information. Examples are Flickr[1] and Panoramio[2] allowing to display GPS-tagged images, YouTube[3] supporting

[1] https://www.flickr.com (20.08.2015)
[2] http://www.panoramio.com (20.08.2015)
[3] https://www.youtube.com (20.08.2015)

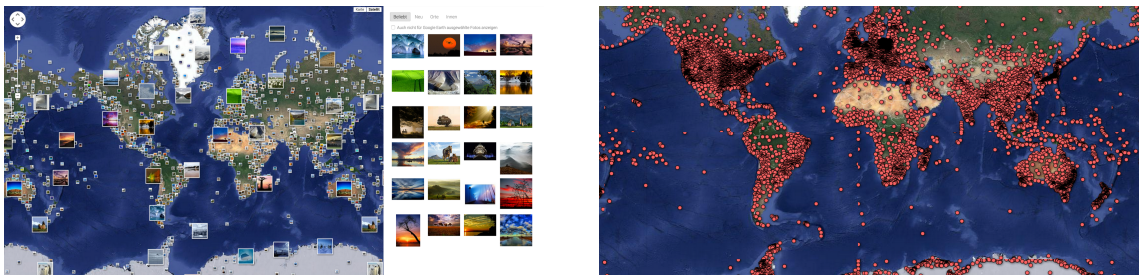

Figure 1.1: Samples of digital map applications used to browse situated media in 2D. Left, a screen shot of the GPS-tagged images available on Panoramio. Right, the world map presenting all GPS-annotated Wikipedia articles.

GPS-tagged videos or location attached Wikipedia articles[4]. Equivalently, all the named services allow to place digital media like text, audio, image and video information into the physical world.

1.2 Interfaces for situated media

Similarly to sticking a physical note to a physical object, situated media can be used to place digital media on a physical object. However, we also need interfaces to access the information.
In figure 1.1, we show digital maps as a common method to access GPS-annotated media. Maps are often used interface to browse situated media. Modern digital maps allow us to zoom to different scales, they give us an overview, and we are fast by finding objects and places on them. Nonetheless, digital maps are limited to two dimensions. E.g. the GPS-tagged Wikipedia article of the Clocktower in Graz and the article about the bell inside the Clocktower are overlapping on a map, as it is not possible to browse the media in 3D.
In order to browse the situated media in 3D, attaching the media on its global position in a 3D map, such as Googles Street View[5], is a known solution. Similarly, the media can also be placed on a representation of the environment, such as panoramic images [45]. The information is accessible by browsing through the static representation of the environment.

Another important method to name is Augmented Reality (AR) [4]. AR accesses situated media placed in three dimensions and overlays it graphically onto a live vision of the physical world, e.g. a camera stream. While initially demonstrated only for research on "bulky" custom hardware [10], AR eventually bridged the gap from research to a commercial product, running on of-the-self devices such as smartphones. AR is often used for advertisements and games, however, accessing situated media in outdoor environments

[4] https://en.wikipedia.org (20.08.2015)
[5] http://www.google.de/intl/de/maps/streetview/ (20.08.2015)

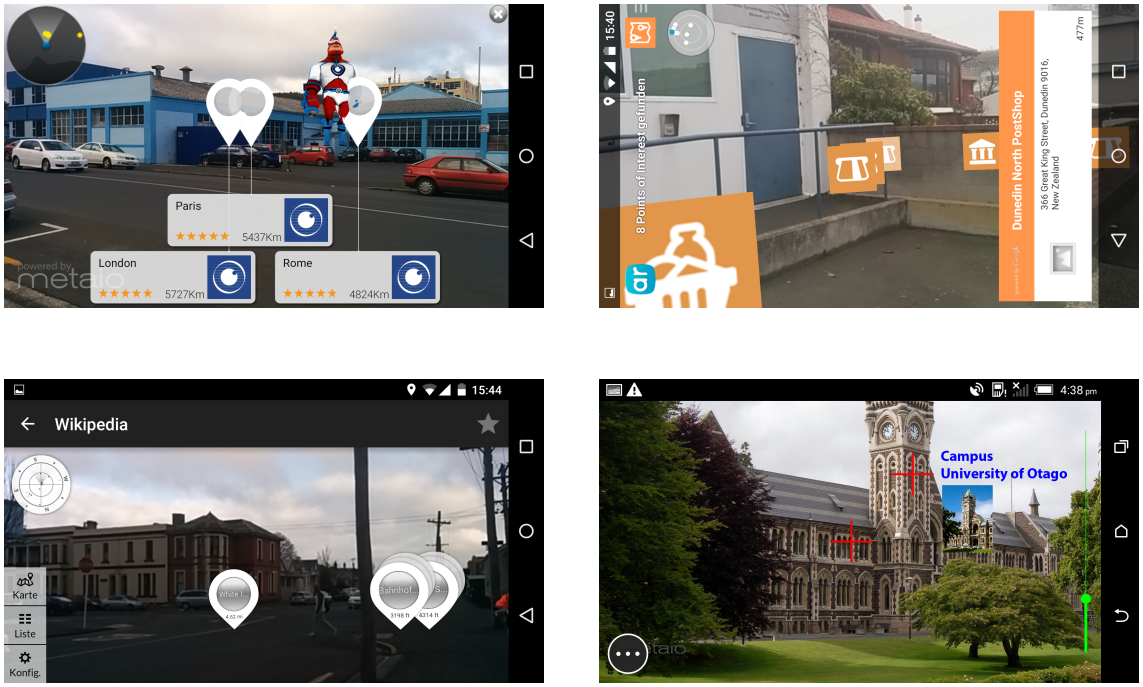

Figure 1.2: Screen shots of existing commercial AR browser. (top left) Metaio, (top right) Layar, (bottom left) Wikitude compared to our prototype (bottom right) of an AR browser.

is still one of the most promising application scenarios [22]. Based on the idea of web browsers for desktop computers, these outdoor AR applications are often referred to as AR browsers [12][21]. In figure 1.2, we demonstrate the concept of different AR browsers. We point the camera towards an object to query the situated media from the physical world. The situated media is correctly blended onto the camera stream as a virtual representation of the object.
Nowadays, many different applications using AR are available, most of them for mobile devices. Layar[6], Wikitude[7] and Metaio[8] are commercial examples of AR browser, combined they have over 40 million downloads.

1.3 Urban pointing

Methods like 3D maps allow the user to navigate through a static representation of the physical world and explore the digital content inside this world. AR browser, instead, blend the situated media on a live representation of the environment. The live updating

[6] https://www.layar.com (20.08.2015)
[7] http://www.wikitude.com (20.08.2015)
[8] http://www.metaio.com (20.08.2015)

Figure 1.3: Concept of the urban pointing interface for situated media browsing [35]. A mobile pointing device, hold like a remote controller, is pointed towards a physical object of interest to query the attached digital media.

of the physical world requires the user to be co-located to the position of the situated media. The digital media is queried by pointing the camera towards its position in the physical world. Even though AR browser are already a widely used method to access the rising amount of digital media, placed in the physical world surrounding us, many challenges remain. For example, we know from the literature that AR browsers have problems in usability and social acceptance [12]. However, there are other interfaces than AR browsers that access situated media in 3D. One of those interfaces is pointing. Tough the concept of pointing is well known and studied for indoor environments [8], it is relatively under-researched for outdoor environments. Similar to AR browsers, the user and the digital content are co-located when using pointing. In order to query the digital information in the physical world, the mobile device aims in direction of interest. For example, to query the GPS-tagged Wikipedia article of an interesting historical building, we can simply point a smartphone as a pointing device towards the building to query the article. In figure 1.3, we illustrate such a pointing device and its interface concept to query situated media. In that example we would use it like a remote controller or the fictional "tricorder" (Star Trek).
While pointing in outdoor environments was already conceptually introduced, to our best knowledge, no pointing interfaces with high accuracy exist. Current implementations rely on sensors to identify pointing actions [32][35]. Sensors such as GPS or magnetic compass are often affected by the environment, making it hard to implement accurate pointing. However, the concept of a pointing promises natural ergonomics and an unobtrusive interface to browse situated media.

1.4 Outline and contributions

In this work, we introduce the concepts and implementations of accurate urban pointing interfaces. We do so by firstly discussing the AR browser as an existing concept for accessing situated media. Furthermore, we review the concept of situated media pointing and a novel AR browser based pointing interface. Many different pointing interfaces for well defined indoor environments are known, but not many implementations for outdoor environments exist. One of the main reasons might be the accurate device position and orientation that is needed to realize pointing interfaces. Tracking in controlled indoor environments is easier than in uncontrolled outdoor environments. We apply state-of-the-art methods to track position and orientation and achieve accuracy comparable to current AR browser, allowing us to study media pointing as an alternative interface to AR browsing. Finally, we present the reflected AR browsers, combining the one-handed ergonomics of pointing with the interface concept of AR browser, augmenting the camera stream with the situated media content. After discussing the concepts and presenting the technical details of the prototypes, we give an outlook of their usability and ergonomics. We argue the proposed interfaces are more natural to use, therefore, more socially acceptable and of higher usability, compared to the known AR browser. We summarize the interface discussion into three hypotheses that we want to accept or reject in an upcoming user study. Summarizing, the significance of this work is two-fold. Firstly, it has significance for the mobile human-computer interaction research community, as it presents a new approach for accurate interaction with mobile devices in outdoor environments, of scientific and commercial interest. Secondly, this work also has significance for the AR community, as it will add novel insights into existing approaches and use-cases in AR browsers.

2 Related Work

Contents

In this work, we present our idea and implementation of known and novel situated media browsing interfaces. The systems we suggest involve two main research areas: (1) situated media interfaces, (2) accurate location and orientation awareness. In this chapter, we put our work in relation to the current research covering the stated topics. In the first section of this chapter we introduce the terminology of situated media. In the following, we discuss the research field of situated media interfaces and systems regarding the taxonomy introduced by Hansen [15]. In the second section, we focus on the practical problems of such interfaces. Mainly, we describe accurate orientation estimation and localization in urban environments on mobile devices.

2.1 Situated media systems and interfaces

With the rising number of mobile devices, the information produced and consumed by such devices has changed. Increasingly more information gets enriched by taking advantage of the mobility of such devices. Numerous services add the geographic position of the user to their generated data, e.g. GPS-tagged videos, images, articles or even services. In 2006, Hansen [15] generalizes systems producing and consuming such information with the terminology of ubiquitous annotation systems. Ubiquitous systems are basically mobile systems, annotating means adding notes to objects. Ubiquitous annotation systems summarize all systems allowing the user to add digital information to the physical world (e.g. objects, locations, or even persons). In his work, Hansen categorizes such systems and names the main challenges to build them. A system can be

designed for on-location or off-location purposes. On-location systems require the user and the system to be co-located. Off-location systems allow the user to interact with the system from any location. Beside the location of the system and the user, Hansen, categorizes the connection of the information and the object as attached or detached. The digital information is attached if it is placed onto a physical object or location. In case the information is detached, there exists a conjunction between the object and the information. Considering the taxonomy, our presented work can be classified as on-location attached ubiquitous annotation system. The user and the physical object are in close distance, and the annotations are placed on the object.
Hansen describes four challenges to consider for the creation of an ubiquitous annotation system: anchoring, structure, presentation and editing. Anchoring is the linking of the physical and digital world. Thereby, the technique used for the anchoring also affects the accuracy of the annotation placement. Linking digital information with GPS-locations is a common type of anchoring. The structure is the technical implementation of the different anchoring technologies. An often used structure technique is to encode the link into the digital information. For example, one may store the GPS-location within a photography. The presentation characterizes the type of the selected annotation, such as text, image, audio or video content. Finally, the editing, according to Hansen, is the manipulation of the digital content.

The term "situated media" is first mentioned by Guven et al. [13] in the same year. Similar to Hansen's on-location concept [15], Guven et al. explain situated media, as media (e.g. hypermedia or multimedia) that is attached into a close physical environment of the user. The term is mentioned in the context of a user-friendly authoring system for mobile Augmented Reality (AR). The idea of the work is to build a system to allow users, without programming knowledge, to create their own situated media content inside a mobile AR application. The common definition of the term "Augmented Reality" is given by Azuma [4] as an application that fulfills the criteria of: combining the real and virtual world, being interactive in real-time and registering three dimensional information in the physical world.

An earlier vision of situated media interfaces can be dated back to 1999. Jim Spohrer publicizes his vision of the Worldboard in that year [39]. The Worldboard is a global infrastructure that allows to attach digital information to the physical world. The information is provided to the user in digital or augmented form on handheld devices or head-mounted displays. Although not all of his visionary ideas became true, yet, his input is leading the way to the current situated media research.

After exploring the definition of situated media, we focus on the interfaces to query the media. The taxonomy for ubiquitous annotation systems by Hansen [15] can be adapted to classify situated media interfaces. The commonly used interface to

	Attached	**Detached**
On-location	AR browser [10][23][21][31] QR codes [17] Marker (Image) tracking [19][43] Pointing [9][32][36][35][47]	Situated aware maps [1][28][44] Indirect AR [45] Situated Simulation [25] Location and ID Sensors [14][31][39]
Off location	VR annotations [37]	Maps Street view

Table 2.1: Situated media interfaces using the taxonomy introduced by Hansen [15]. The situated media is attached, when it is present on the physical object, whereas detached media is in conjunction with the physical object. A situated media interface is considered to be on-location if the user and the physical object need to be co-located. The interface is off-location in case the user can browse the information location independently.

browse situated media are digital map applications. Applying the taxonomy of Hansen, information presented in map applications can be classified as detached media. The digital object on the map is in conjunction with the physical object it represents.
One of the first digital map applications is the Cyberguide, presented by Abowed et al. [1] in 1997. Their mobile tour guide uses the global position of the user to navigate the person to location annotated objects on an image-based map. The Cyberguide is able to globally localize the user, outdoors using GPS and indoors with the use of an infrared network. Even though there are problems with the accuracy of the system, like positioning on the image-based map, their idea of interacting with the situated media on a map using the current location of the device, is widely used, nowadays.
To put the situated media on the map in relation to the relative position of the user, Pradhan et al. [28] added in 2001 the orientation of a digital compass to the user location. Pradhan's work provides an interface to explore hyperlinks attached on a two dimensional digital map. With the use of a GPS equipped hand-held device, the user of the system is able to browse hyperlinks in the physical world by pointing with the device. In practice, due to the inaccuracy of the consumer GPS and the digital compass, the authors face offset problems while pointing. To overcome the offset issue, the authors introduce an algorithm correcting the angle of view of the estimated pointing direction. The algorithm simply adjusted the angle of view according the uncertainty of the sensors. Unfortunately, a wider viewing angle leads to an inaccurate object selection. Nevertheless, the addition of the compass allows a new form of situated media browsing on a map, by using a two-dimensional pointing gesture from the current user position in the direction of the information. Applying the pointing gesture, the user is able to choose the information of interest registered on the map.

Up to this point, we have discussed on-location services by means of user location-aware map interfaces. Thereby, the situated media was present as

detached information on the map. Simultaneous, there exist numerous off-location map interfaces. In the early 2000s, location based systems got the interest of the industry. In 2003, Google begins with the development of Google Earth[1] and later, the web-based Google Maps[2] map application. Around the same time, other services, like Microsoft's Bing Maps[3] and the community-based OpenStreetMap[4], appeared. In 2012, Apple presented their own digital map application Maps[5].

The idea of all this services is to allow the user to browse and search off-location, geo-annotated media on a digital map. Having the map in digital form on any device with a web browser and an internet connection, the user can interact over the web browser with the presented media. By considering the position of the user, all these map applications also provide on-location services (e.g. navigation). Instead of using the web browser to interact with the media on the map, handheld devices can use the position and the device's bearing to access digital content in an area around the user.

However, map applications are limited to two dimensions for the media presentation. To overcome this issue, many projects focus on different forms of Augmented Reality (AR) to attach situated media information to the physical world.
By definition, AR applications provide an interface to explore situated media. The pioneering work in the field of outdoor AR is done 1997 by Feiner et al. [10]. They use a desktop computer in a backpack with an external GPS module and a compass for outdoor localization. Using this information, Feiner et al. attached digital labels on real buildings around their campus, visible through a head-mounted display. Similar to the on-location map applications, the two-dimensional location and orientation of the user is used to attach the digital, three-dimensional labels around the user to physical objects.
Following Feiner's work of an augmented labeling system, the term AR browser appeared. Identical to the concept of web browser, used to access hyper-referenced media, AR browser can be used as an interface to access situated media. Grubert et al. [12] describe AR browser as a "generic augmented reality application proposing to display geo-located multi-media content using a digital representation augmented on the vision of the physical world (i.e. a camera-image in the context of smartphone technology)".
Often, AR browser are used in combination with authoring systems, allowing the user to register different types of digital media in the physical world. Like the work of Rekimoto et al. [31], allowing the user to place digital Post-its in form of photographs or audio notes into the physical world. The media is explored by using wearable hardware, for the localization of the user and the presentation of the media. Another example is the

[1] https://www.google.com/earth/ (17.03.2015)
[2] https://www.google.com/maps/ (17.03.2015)
[3] http://www.bing.com/maps/ (17.03.2015)
[4] http://www.openstreetmap.org/(17.03.2015)
[5] https://mapsconnect.apple.com/(17.03.2015)

user authored audio information system by Langlotz et al. [23]. The audio information is browsed using a mobile phone. Later in 2013, Langlotz [21] presented the concept of Augmented Reality 2.0. Similar to the Web 2.0 provides an infrastructure for user generated hypermedia content, Augmented Reality 2.0 is the concept of providing infrastructures and interfaces for the user to generate situated media content. In his work, Langlotz presents authoring systems as well as existing browsing systems (e.g. AR browser).
The work of Langlotz also shows the capability of mobile phones as a platform for AR browsers. Such a phone-based interface is presented in 2006 by Kähäri et al. [18]. The project MARA is an outdoor AR browser using a mobile phone with attached sensors, for the pose estimation, as well as a hand-held visual see-through for the augmentation. Due to the technological progress on computation power, application development, sensor integration, mobile phones become the most used platform for AR browsers. With the technological progress and the diversity of mobile phones, the concept of AR browsers also lead to a variety of commercial products during the last years.
Beside the interface type, the media presentation is important for the usability. It is important to avoid overloading the vision of the physical world with information, because the linkage between digital content and physical object can get lost. Lee et. al. [24] cover the concept of information layers while browsing digital information in the physical world. In dependence of the media type, the media is organized in presentation layers. This layer system allows the user to selectively browse information of interest. Other works are investigating the placement of two dimensional or textual labels. They consider view management as an essential part of accessing situated media with the use of AR browser [5]. An intelligent label placement is important to keep the interesting physical world information visible. Rosten et al. [33] used visual features to identify areas with information. Following, labels are placed in areas with the least visual information.

While map applications are a two-dimensional interfaces and AR browsers are tree-dimensional interfaces for situated media browsing, indirect AR can be described as a combination of those two interfaces [45]. Generally, indirect AR uses an offline panorama with the registered situated media, instead of the live camera preview. The situated media is offline placed in the panorama as a representation of the physical world. However, the panorama selection depends on the position and orientation of the user. Compared to AR browser systems, this method allows an accurate offline placement of the augmented media, as the whole panorama view is affected by 3D registration errors, but not the media placement. Using Hansen's taxonomy [15], we speak of an on-location interface for detached media.
The photographer Ratana presents an app using indirect AR to show the sun altitude at different day times[6]. Ratana blends the panoramas of Google's Street View[7] with the

[6] http://www.sunsurveyor.com/(24.03.2015)
[7] https://www.google.com/maps/views/streetview(24.03.2015)

position of the sun. Similar to map applications, this method is a three-dimensional off-location, detached situated media interface. The situated is registered in the 3D map data of Google Street View.
In 2011, the same year of the presentation of indirect AR, Listol [25] takes the 3D pose of a phone, as known from AR systems, to select the point of view for pure digital renderings. E.g. a virtual camera in a digital world is moved by the physical movement of the phone camera. Similar to indirect AR, the situated simulation considers the on-location physical environment of the user to explore pure digital 3D content attached to a physical position.
Generally, most AR methods use a vision system with a camera pointing into the scene to determine a pose usable to set a digital camera. As hand-held devices are the common platform for AR solutions, the ergonomics for such media browser is defined by the physical placement of the camera onto the device.

Alternatively to use AR to access three-dimensional situated media, two-dimensional pointing, considering position and orientation of the user [28], can be extended to browse situated media in three dimensions. The idea of identifying geographic objects by pointing is introduced in 1999 by Egendorfer [9]. The pointing device is called a Geo-Wand, it is equipped with a GPS receiver and a gyroscope to identify to position and orientation of the device. Direction and position are matched with the knowledge base of the system to identify the digital object the user is pointing at. The Geo-Wand retrieves basic information from the knowledge base of the system, but a connection to the internet should allow the user to browse additional information about the detected object.
Later, in 2007, with the growing number of mobile phones and the raising consumer market for advanced navigation devices, such as GPS sensors, accelerometer, gyroscopes or magnetic compasses, Simon et al. [36] construct the Point to Discover system. They describe the Point to Discover as a prototype of the Geo-Wand introduced by Egendorfer. The system combines a mobile phone, a magnetic compass, a GPS and an accelerometer-based tilt sensor to a location and orientation aware pointing device.
One year later, Simon et al. [35] demonstrate the performance of the Point to Discover system under real world conditions. They individually evaluate the performance of the localization and the pointing in four representative areas: low-density urban, park, urban and urban canyon environment. To evaluate the localization, the GPS offset to a designed path on a map is measured. The pointing quality is measured by the horizontal offset of the estimated pointing direction from fixed positions to target objects. Even though good results are achieved in the park environment, the known inaccuracy of the used consumer GPS as well as the unprecise magnetic compass are the main problems of the system. Beside the weak performance of the consumer hardware, small positioning errors leading to large pointing offsets are challenges to solve.
The next year, in 2009, Robinson et al. [32] use location and orientation aware devices

to implement the sweep shake interface. Instead of receiving visual feedback, the user is notified by vibrations of the pointing device, when situated media is detected. To browse the selected annotated media, the user points to one of the four defined haptic areas to choose between image, text, audio or video media. The evaluation of the system shows that haptic feedback is less accurate than the described three-dimensional visual pointing systems (e.g. the Point to Discover system [35]), nonetheless, the user consider the system generally as easier to use.
Also in 2009, Zhang et al. [47] used a Geo-Wand implementation on a mobile phone to study their interface for searching two-dimensional annotated information. Rather than pointing directly to the annotated physical world object, they use spacial scanning gestures to find these annotated objects. By introducing the wide angle scanning gesture, they overcome the problem of the pointing offset, due to the low performing sensors in the phone. The study participants still consider the system as "cool to use" and successfully performed searching tasks.

Later, in 2012, Zhao et al. [48] compare different outdoor pointing techniques. They compare pointing with the camera to pointing using the mobile phone like a remote controller. Near and far pointing accuracy of targets in a 180 degree field of view are studied. For the study, they define three different conditions. In the pointing condition, participants hold the phone to the target, arm outstreched. In the focus condition, users point the center of the phone screen to the target without visual indication (similar to an AR browser). Finally, in the hold condition, participants hold the phone flat (like a compass) between waist and chest to be able to see the screen. Finally, results for the focus and the hold condition are presented. In both cases, gestures are identified as pointing, if for a given time threshold, no point exceeds 10 degrees of distance around the target. Focusing is generally slower than holding (2.4 ms versus 1.5 ms). However, the mean absolute error over all test targets reaches for the hold condition 9.9 and 3.9 degrees for the focus condition.
Despite their pointing method accuracy evaluation, the authors mention a GPS and sensor based prototype for an outdoor pointing system. The outlined prototype uses an Android mobile phone as a data generating server and a client computer, analyzing the data to find probable object selections done by the phone.

Vision-based on-location technologies require close distances to the physical object to request its annotated information. The probably most known example is the QR code [17], a standardized black and white pattern used as a visual data storage. The QR code annotates a physical object with digital information by attaching the code onto the object. Using a QR code scanner, the annotated media can be read out, thereby, the distance to browse the information depends on the size of the code.
In 1999, the early days of vision-based AR, fiducial markers are used to estimate the relative pose of the camera detecting the marker [19]. Fiducial marker are artificial

images, easy to detect in natural environments. With the increased calculation power of mobile phones, Wagner et al. [43] presented in 2008 a pose tracking on mobile phones, based on planar natural features. Unlike QR codes, fiducial marker or natural feature based tracking allow to track the pose of the camera that is used for the detection. Given the pose of the camera, the situated media can be augmented into the physical world.
The advantage of the visual detection and pose estimation is an accurate 3D registration of the digital content in the physical world. However, to interact with the attached digital media, close distance, relatively to the size of the marker is required of the user.

Later, in 2004, Hansen et al. [14] introduce the context aware hypermedia framework HyCon as an other on-location technology. The framework associates locations of RFID and bluetooth-tagged objects with images, websites and other situated media. All of these areas send their location information to a mobile HyConExplorer within their range. The system requires the user to be close to an associated area, even though the information is not attached to a certain object. With the HyCon, Hansen et al. [14] introduced a prototype of the Worldboard concept of Jim Spohrer [39]. Similar, the digital Post-its presented by Rekimoto et al. [31] use wireless personal area signals to access on-location detached media. These two works both show the idea of accessing situated media through wireless personal area technologies. The media is detached from the object, and the situated information is automatically queried in an area around the physical object.

In contrast to the discussed interfaces Sinclair et al. [37] present in 2002 a method for off-location but, attached situated media browsing. They put labels on parts of a digital airplane augmented on a fiducial marker. More generally, browsing annotated digital information on digital objects can be classified as situated media browsing as well.

2.2 Location and orientation awareness

Following the first implementations of situated media interfaces, a lot of effort was put in the accuracy optimization of outdoor localization and orientation estimations. Those two measurements are the key for a stable and accurate technical realization of a situated media interfaces. Early works, e.g. Feiner et al. [10] or Pradhan et al. [28], rely on GPS localization and the two dimensional orientation of a digital magnetic compass. Both works claimed too inaccurate GPS sensors and magnetic compasses as main reasons for their unstable media browsing. However, compared to modern sensors in mobile phones, they used standalone sensors with antennas and less interference. Still, they are not able to reliable annotate fine details in the physical world.
Zandberg et al. [46] studied, in 2011, the localization error of the integrated GPS sensors in mobile phones. They measure an average error of five to eight meters in their outdoor test scenarios, more than two times as much than for standalone sensors.

To overcome the inaccurate positioning using GPS sensors, stable and precise computer vision approaches are used to determine the position of the device in the world. The fiducial marker tracking by Kato et al. [19] allows to track the position and orientation of a camera in the physical world. The integration of the camera into the moblie phone extends the position tracking to the mobile phone. Unfortunately, marker-based tracking requires to change the reality by placing the marker into the physical world.
Wagner et al. [43] extend the idea of marker tracking to natural feature tracking. Instead of using an artificial marker, the position of the camera can be extracted by matching natural features, on a plane, with a known set of features. The work of Wagner et al. allows to remove the marker from the scene, but requires scene knowledge in form of a natural features set.
The improved performance of mobile phones allow Klein et al. [20] to implement the Simultaneous Localization and Mapping (SLAM) approach on a camera phone. During the movement of the camera, feature points are extracted of the camera stream and triangulated to three-dimensional world points. The generated point cloud forms a rough model of the physical world and allows to track the position of the camera relative to the model. The SLAM algorithm does not require scene knowledge, nor do artificial markers need to be placed into the scene. Nonetheless, the triangulation of the three-dimensional points requires translations of the camera position, since the reconstruction is similar to using a stereo camera with a fixed baseline. Furthermore, the positioning is relative to the generated point cloud and not a global position estimation, which is necessary to browse situated media registered in the physical world.
In 2010, Wagner et al. [42] use natural feature tracking on a mobile phone to estimate the orientation of the phone by tracking features on a panorama representation of the environment. The method is similar to the SLAM method, but instead of creating a three-dimensional map of points a two dimensional map is generated. New points are extracted from a panorama representation of the environment generated on the fly. Therefore, no pre-knowledge of the scene is required. Since a two-dimensional representation of the physical world is created, the camera does not need to be translated for a three dimensional point triangulation. However, the depth information is not extracted, and the algorithm tracks the 3 Degree of Freedom (DoF) orientation of the camera and not a full 6DoF pose including the relative distance to the point cloud. The evaluation of the method results in a horizontal orientation offset of less than one degree, after the full 360 degree panorama is created.

One notable drawback of these vision-based 6DoF and 3DoF tracking approaches is the missing global localization. A second issue is the reliability. Vision-based methods have to deal with occlusion, illumination changes, different view angles, scales and rotations.

In 2007, Reitmayer et al. [29] presented an approach to visually get a 6DoF pose in urban environments on mobile devices, supported by orientation sensors. The combination of a visual edge detection and a known textured model of the environment, supported by sensors allow a stable and reliability pose estimation. The same year, Reitmayer et al. [30] used the relative outdoor pose estimation to faster initialize and improve the global GPS localization. The method improves the GPS location by statistically combining the tracked position in the known textured model and the current GPS location. Measurements on known ground truth locations showed a mean distance of less than 0.2 m from the estimated position to the ground truth position.
Later, in 2009, Arth et al. [3] presented a method to determine a 6DoF camera pose, by comparing the camera image to an offline generated sparse 3D point cloud of the environment. Starting with an initial position retrieved by GPS, the system determines the camera position from the point cloud. Basically, the cameras center of projection can be estimated by matching the image against the point cloud. The division of the point data-set into cells reduces the memory consumption of the system, to run it on a mobile phone.
Chen et al. [6] introduce, in 2011, a method to determine the global position by matching the camera view against the panoramic information retrieved from Google Street View. In a similar way, Ventura et al. [40] [41] show a Street View independent localization and tracking method in urban environments on mobile phones. They prepared an offline model from captured panoramas of the environment. By utilizing a client-server model, the localization from the offline model can be done on the server, while a patch tracker runs on cached panoramas on the mobile device. The tracker combines a visual patch search inside the offline model with a known pose prior to estimate the current pose. The prior is provided by the server localization or from the previous tracking round. In order to estimate a global pose, Ventura et al. present a scaling and orientation method to register the offline model in a global reference frame. Finally, the system reaches an average error of less than 25 cm in translation and less than 0.5 degrees in orientation.
Recent work in the field of global pose detection and tracking is done by Arth et al. [2] in 2015. The method proposed has a similar concept as the work of Reitmayer et al. [30]. However, Arth et al. are able to estimate and track a global camera pose using untextured models of the environment. They combine a line segment detection with a vision-based segmentation method to identify planes in the camera view. The determined planes are matched against the model of the environment to estimate the pose of the view.
Summarizing, the combination of vision and sensors offers promising methods for global pose estimations. However, the presented approaches have their drawbacks, wheather some pre-knowledge of the scene is required or the pose estimation is not global and limited to a small environment. Overall, the methods show accurate positioning results usable for situate media browsing.

Similar to the 6DoF pose estimations, there exist 3DoF global orientation estimations, performing without scene knowledge. A global 3DoF orientation estimation is registered in a north-centered reference frame and can be used for situated media browsing from known positions. The common method to get the north-centered orientation of the device is using the internal sensors, summing up the angular moments measured by the gyroscope or combining the gravity vector from the accelerometer and the north vector from the magnetic compass. However, the gyroscope orientation is relative to the orientation of its initialization and introduces a drift over time, due to the summation of the measured angular moments. The acclerometer-compass orientation is measured in a north-centered reference frame, but contains the noisy north vector of the magnetic compass. Lawitzki[8] fuses the sensor orientations to a stable north-centered orientation. He achieves the stable orientation estimation by removing the compass noise with a low-pass filter and the gyroscope drift with a high-pass filter, before fusing both signals to one orientation.
Schall et al. [34] introduce, in 2010, a north-centered orientation by combining the stable, but relative vision-based orientation of Wagner et al. [42] and the north aligned acclerometer-compass orientation. This is achieved by estimating the average offset between the visual orientation and true north, over time, with a Kalmanfilter. While the method of Lawitzki relies on a calibrated compass, the approach of Schall et al. dynamically calibrates the compass. Over time, the evaluation of the system results in an offset of less than three degrees to true north.

2.3 Discussion

In this chapter, we defined the terminology of situated media and classified existing situated media interfaces using the taxonomy of Hansen [15]. The study of different situated media interfaces gives us an overview of the current systems and the main problems in that particular field. Beside two-dimensional map applications, AR browser are the preferred method to browse digital media annotated in the physical world. Generally, map applications do not suffer from inaccurate positioning, since they are designed to browse detached media. AR browsing systems, designed to interact with attached three dimensional situated media, require an accurate global pose estimation. Recent methods achieve good results by supporting the inaccurate sensors with vision-based approaches. We reuse selected state-of-the-art tracking methods to build mobile device based, situated media interfaces. Specifically, we combine the global localization of Ventura et al. [41] and the north-centered orientation estimation of Schall et al. [34]. Our prototypes are the basis to study different situated media interaction methods. So far, no pointing interfaces with comparable orientation and localization accuracy as current AR browser exist.
During the study of the related work we skipped the wide field of indoor pointing systems,

[8] http://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutorial(18.04.2015)

mostly in relation to large screen pointing systems. In the survey work of Dang [8], the common techniques are presented and classified. We are aware of this related field, however, this thesis deals with outdoor pointing interfaces, especially in urban environments. For further information related to the field of pointing, we refer to the survey work of Dang.

3 Concepts

Contents

Situated media is digital information attached to locations or physical objects. To be able to access situated media on-location, we require mobile interfaces. Basically, we are interested in the technical implementation and, further, in usability, performance, ergonomics and social implications of such interfaces. Consequently, in this chapter, we discuss the concepts for 3D situated media interaction, such as AR browser and media pointing interfaces.

The common concepts to interact with situated media are digital maps, Augmented Reality (AR) and media pointing. Digital maps are widespread and often used to interact with media, linked to physical locations. Maps are designed to access the media in 2D. 3D maps are rarely used to browse situated media because of missing 3D map content for many areas. AR methods aim to process live representations of the physical world and are designed to browse digital media registered in 3D. Similar media pointing allows to browse media in 3D, however, not much work covering media pointing in outdoor environment exists. Identically, both methods, AR browser and media pointer, require the user to be co-located with the situated media of interest.

The concepts are the foundation for accuracy oriented prototypes as implementations of the introduced interaction methods. Having, the prototypes allows us to study the usability, performance, ergonomics and simultaneous social implication of media pointing as an alternative situated media interface to AR browsing. From a technical point of view, the concepts do not differ fundamentally. the mobile device implementing the concepts need to be location- and orientation-aware. E.g. in order to be able to identify the digital media

attached to the Colosseum in Italy, the media browser needs to identify the location of the user in Italy and its orientation with respect to the Colosseum. Though well studied for AR browsing, the global pose problem is still a hard challenge to solve in many fields. Especially existing media pointing prototypes [32][35] suffer from poor location and orientation awareness, affecting the interface performance. Many AR browser implementations are intended for urban environments [2][30][41]. Urban environments have the advantage to be relatively static and mostly provide additional information in form of 3D models. As we use existing technologies to implement our prototypes, we limit our applications to to urban environments.
In the following we discuss the concept of AR browser, media pointing and a newly introduced reflected AR browser.

3.1 AR browser

First we discuss the AR browser concept. By definition an AR browser is a "generic augmented reality application proposing to display geo-located multi-media content using a digital representation augmented on the vision of the physical world" [12]. Basically an AR browser is the common concept to access situated media in 3D. The AR browser is therefore the reference to our newly introduced interaction methods. Nowadays, AR browser are mainly implemented for smartphones or tabled devices. The general idea of an AR browser is to record the physical object or location of interest with the mobile device. Onto this vision of the physical world the 3D registered situated media is blended correctly placed. From an ergonomic perspective the camera is pointed towards the physical world of interest and the media is viewed on the display of the device. In figure 3.1 we illustrate the concept of an AR browser. First, on the left, we show the ergonomics of the AR browser. The device is hold at head height, used as a video see through to explore the environment. Second, on the right, we present the idea of an AR browser interface. The virtual object representation of the Colosseum in Italy is blended with a label naming the object. In this case the label is the situated media, similarly we could show any media attached to the Colosseum, e.g. a Wikipedia article.
AR browser are a known method to access 3D situated media. However, we argue that they are obtrusive to use in social environments, because of the unusual, head height usage of the mobile interface. Additionally, AR browsers copy the already visible world to build the linkage between digital media and the physical world. However, the tight linkage between digital and physical world favors an accurate selection of the digital media. To select the situated media attached to a physical object, the center of the camera, usually indicated by a cross-hair, is pointed towards the object. The literature teaches us that aiming the center of the camera to an object can achieve pixel accurate object selections. Differently, pointing with a mobile device to select an object is a harder task [48].

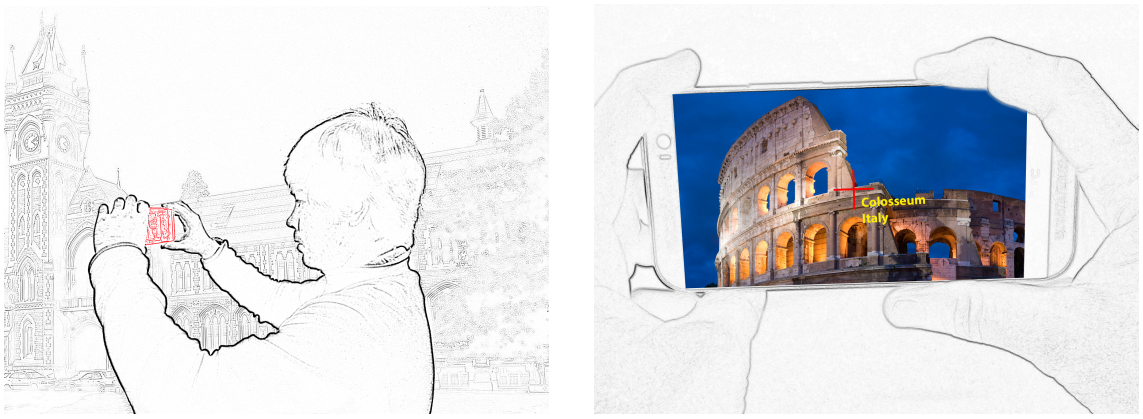

Figure 3.1: Sketch of the concept of an AR browser. On the right, the ergonomics of the concept is presented. The device is used as an optical see-through, scanning the scene, held statically at head height. On the right, the interface is presented, displaying the situated media content (the description of the object in this case) blended correctly over the camera stream.

3.2 Media pointing

A different situated media browsing concept is pointing. Media pointing is generally a well-known concept, though it is mostly studied for controlled indoor environments [8]. The concept of AR browsing accesses the situated media using camera aiming. In contrast, the idea of outdoor media pointing is to point with the mobile device, similar as a remote controller, towards the physical object of interest. On the left side of figure 3.2, we illustrate the concept of pointing. The device is held one-handed at waist level. We argue that the natural gesture of using a mobile device is unobtrusive and socially more acceptable than an AR browser.
On the right hand side of figure 3.2, we demonstrate a mock-up of the pointing interface. Due to the concept of pointing instead of camera aiming, the camera image is not rendered onto the screen of the device. The unused screen space can now be fully occupied with the situated media content, allowing to display more content than on an AR browser. E.g. blending the whole content of a news article over the camera stream of the AR browser makes the concept of camera focusing to select the media impractical. By occupying too much of the screen space, there is not enough of the camera stream visible to accurately browse media. Therefore, many AR browser implementations place labels into the environment and make the content accessible by clicking the label. However, to access the full situated media content, a view switch is needed. In our opinion, the different object selections of the media pointer, allowing to occupy more screen space to present information of interest, can increase the usability. Nonetheless, as shown in the mock-up of figure 3.2 (right), not the whole screen space is occupied with situated media content. The user still requires feedback for a successful media selection. As a possible solution of

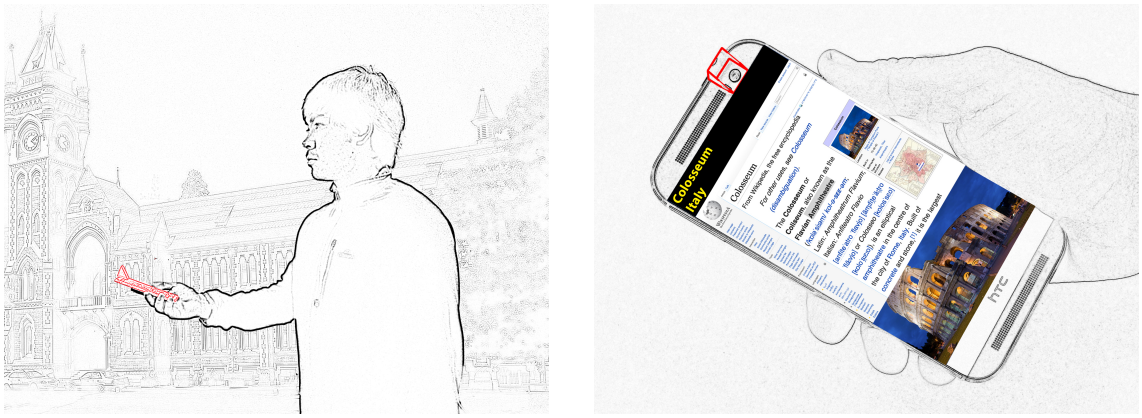

Figure 3.2: Sketch of the concept of the situated media pointing interface. The illustration on the left demonstrates the ergonomics of the concept. The device is used similar to a remote controller to select the situated media of interest (left). The situated media is selected by pointing towards the physical object of interest with pointing gesture at waist level. Right, the envisioned interface layout of a pointing interface is presented. Most of the screen space is occupied by the situated media content. A static representation of the selected physical objects connects digital and physical world.

connecting the digital world with the physical pointing gesture, we can think of placing a static representation of the selected physical object next to the media content.

3.3 Reflected AR browser

Finally, while working on the prototypes of the AR browser and the media pointing concept, we came up with a third, new interaction method, combining the existing concepts. We named the new concept reflected AR browsing. It combines the ergonomics of the media pointer with the accurate camera aiming of the AR browser. Other interface aspects are similar as for the AR browser. The camera records the physical scene in front of the user, is rendered on the screen and blended with the digital media registered in the scene. Simultaneous, the device is held similar to the media pointer, one-handed at waist level. We sketch the idea of the newly introduced situated media pointing concept in figure 3.3 (left). We envision the user to focus on the physical world instead of the visual representation on the browsing device. However, pixel accurate object selection by camera aiming is still possible. In figure 3.3 (right), we illustrate the interface layout. The layout is basically an AR browser in portrait mode.
Subsequently explained in chapter 6, we implemented the reflected AR browser by mounting a prism on the camera of the device. Using the prism, the light rays are bend by 90 degrees and the scene in front of the device can be captured, while keeping the media pointing gesture.

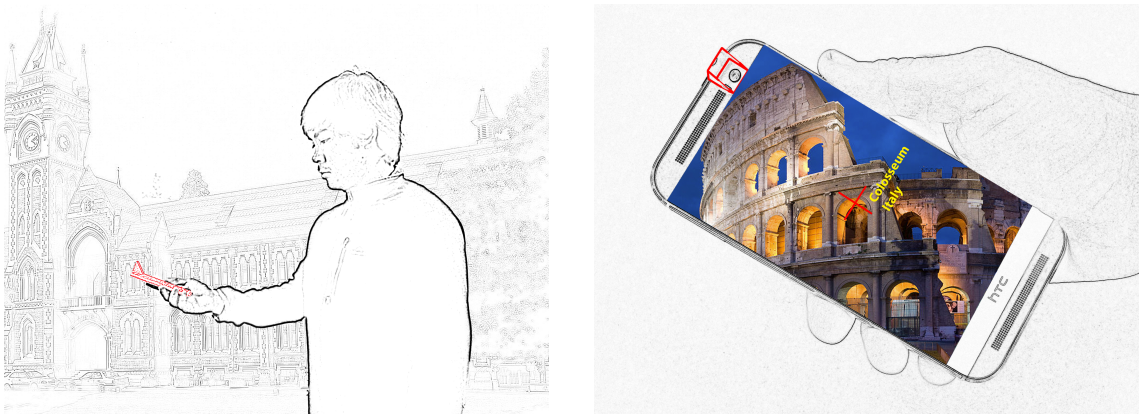

Figure 3.3: Sketch of the concept of the reflected AR browser. On the left the envisioned ergonomics of the reflected AR browser is presented. The ergonomics is identical to the media pointer, though the media selection is done by camera focusing. On the right, the interface layout of the concept is visible, a hand held optical see through reflecting the environment, with blended situated media content.

3.4 Summary

The AR browser is the current, well studied method to browse situated media in 3D. In this chapter, we presented the media pointer and the reflected AR browser, alternative interface concepts to browse 3D situated media. Both methods differ from the AR browser in terms of usability, ergonomics and consequently in social acceptance. In the following, we discuss the technical implementation and the performance of the presented prototypes.

4 Localization

Contents

Previously, we introduced the idea of situated media as digital information, annotated in the physical world. We discussed existing interface concepts for accessing situated media and the technical challenges of situated media browsing. The main problem to solve for any interface is the global pose problem. The global pose is the combination of the global positioning and a north-centered orientation estimation. In order to solve the global pose problem, we adapt existing solutions of the global positioning and the north-centered orientation estimation. Finally, we combine both solutions to a global pose. The estimated global pose is further the foundation to implement prototypes demonstrating the concepts of an AR browser, a media pointer and a reflected AR browser. In the following, we describe our solution for the global positioning problem.
The device for the implementation of our interface prototypes is a standard hardware, Android based smartphone. Naturally, the integrated consumer GPS sensor would be the basic method to localize the smartphone. Though, an accuracy of less than eight meters within a confidence interval of 95% [46] is not accurate enough for the purpose of situated media browsing. The possible uncertainty radius of eight meters only allows to annotate digital content to objects with 16 meters in diameter. We could not reliable detect smaller objects, as we can not identify our correct position within a circle of 16 meter diameter. However, the literature teaches us that the combination of vision-based methods and sensors can yield accurate global localization [20] [30] [40] [41] methods. The localization and tracking method of Ventura et al. [41] is a recent work, and the code for the method

is the only reviewed method publicly available. The code is published under BSD license at GitHub and carried on as open source project after the publication of the work in 2012. Therefore, we decided to use the proposed method to get a vision-based global localization on our smartphone. Even though the system can handle arbitrary environments, the best performance is achieved in a static urban environment. This fact leads to the decision of implementing our pointing interfaces for urban scenes.
In this chapter, we explain the components of the "Wide-area scene mapping for mobile visual tracking" of Ventura et al. [41] and our contribution to the open source project. The components of the system are: (1) the reconstruction, (2) the global registration, (3) the online localization and (4) the mobile tracking system. Basically, the method implements a client-server architecture. After an offline reconstruction and global registration step, the mobile client requests its position from the server. Therefore, an image is sent by the client to the server which determines the global position of the image by matching it against the reconstruction. Compared to local Simultaneous Localization and Mapping (SLAM) approaches, the client-server architecture allows to localize in large environments. The reconstruction is generated offline and stored on the server, accordingly not limited in size. Furthermore, one image is enough to get the position from the server, whereas mobile SLAM needs translational movements to triangulate the environment. The evaluation by Ventura et al. demonstrates a translational error below 25 cm and a rotational error of less than 0.5 degrees for over 80% of images tested. However, a feature based vision approach is always sensitive against environmental changes like illumination, occlusion or general appearance changes due to different seasons.

4.1 Reconstruction

We describe the necessary steps to perform global localization requests using the selected client-server method of Ventura et al. [41]. In order to process visual localization requests of the client, the server needs to be aware of the environment. In the original work, the content creation for the reconstruction is done by taking a video with an omnidricetional camera, while walking through the urban environment. From the video, panoramic images are sampled and warped into eight overlapping perspective views. For our reconstructions, we take multiple images using the wide angle camera of the mobile phone and slight translations between each view. We reconstruct buildings of our campus area, where we also evaluate our interface. The Structure from Motion pipeline described by Snavely et al. [38] with modifications for wide angle views is applied to estimate a 3D point cloud of the environment.

First, Scale-invariant feature transform (SIFT) descriptors of each view are extracted [26]. The descriptors of the SIFT image features are matched between each image pair and, with the use of the Random Sample Consensus set analysis algorithm (RANSAC), the fundamental matrix of the pair is estimated. The fundamental matrix

Figure 4.1: Sample images used for the reconstruction of the commerce building (campus of the Otago University in Dunedin, left) and the resulting point cloud (right).

is a 3x3 matrix relating the corresponding points of two stereo images and can be used to reduce the search space for point correspondences. In each iteration of RANSAC, one possible fundamental matrix is calculated by the eight-point algorithm [16]. Consequently, the fundamental matrix with the most inliers is chosen and outliers are removed. If the number of inliers is greater than twenty, the image pair and its fundamental matrix are considered for further processing. However, we have consecutive photography of the environment, therefore, we can reduce computation time by just considering successive image pairs and avoid matching of all images. The matches of the remaining geometrically consistent image pairs are further organized into tracks. A track is a connected amount of matched keypoints, visible in multiple views. Tracks are considered to be valid, if the contained keypoints do not appear in the same image. All tracks with more than two keypoints are kept for the next processing steps.
The keypoints of each track are used to extract camera parameters and triangulate the 3D location of the visible point, such that the reprojection error of each 2D point in the track is minimal. Snavely et al. formulated the minimization as a non-linear least squares problem and solved it with the Levenberg-Marquardt [27] method. In order to avoid bad local minima, the parameter for all cameras and tracks are not calculated at once, but incrementally. The optimization is initialized by estimating the parameter of a well-conditioned pair of cameras. Therefore, the image pairs with the most matches and the largest baseline are chosen. Sequentially, the remaining cameras are added to the optimization, and their extrinsic parameter are estimated. The order depends on the number of already estimated 3D points visible by the camera. Originally, the algorithm estimates the extrinsic and intrinsic parameter $\mathbf{K}$ of the camera with a combination of the Direct Linear Transform [16] method (DLT) and RANSAC, whereby the intrinsic camera parameter $\mathbf{K}$ are initialized with the EXIF tags of the images. The EXIF tag of an image provides meta-information about the image, e.g. aperture, exposure, focal,

length, GPS coordinates. Since we know the camera used for the reconstruction, we calculate the intrinsic parameter **K** of the camera by a previous calibration step, instead of relying on the inaccurate EXIF information of the image. For the calibration, we use the OpenCV camera calibration tool [1] and perform it on the 9x6 tiles square chessboard. As required by the reconstruction framework, we extract the principal point and the focal lengths, radial distortion and skew are ignored.
Next, with the estimated camera parameter, the new tracks of the camera image can be triangulated and added to the optimization, given that the track is visible by at least one other recovered camera. The process is repeated, until no reconstructed points are visible by any remaining camera. After each iteration, a bundle adjustment on the generated sparse point cloud is processed. Due to performance improvements for large data sets, we iteratively detect keypoints with large reprojection errors, remove them and rerun the optimization, as recommended by Snavely et al.
In figure 4.1, sample images used for the reconstruction and the resulting sparse point cloud are depicted. With a standard laptop (Macbook Pro 2015), we could do the feature extraction, image pair matching, track generation, camera parameter estimation and point triangulation for the given and other sample reconstructions (50 to 150 images with 2600x1520) in less than 5 minutes of calculation time.

4.2 Global registration

By having the sparse point cloud of the environment, new relative camera positions can be determined by matching the features of the camera image to the point cloud. However, we are interested to get global positions from the reconstructed environment. Ventura et al. [41] describe the registration process to request global camera parameter from the reconstruction, which we reuse in our work.
The first step to globally register the reconstruction is to estimate the model's up vector. The model is searched for vertical line segments to estimate the common vertical vanishing point. By knowing the vertical vanishing point, the models can be rotated to match its y-axis to the gravity vector. Next, the ground plane of the point cloud needs to be found. In a simple heuristic, the distribution of the point cloud's height value is collected. A selectable percentile value of the distribution determines the ground height. E.g. the ground plane is identified at a height where 0.7 percent of the 3D points lie. The reconstruction is translated along the gravity vector, so that the estimated ground plane is set to zero height.
The last step is the geographic placement of the model in order to estimate its orientation and scale. Accordingly, the point cloud is rendered superimposed on an alignment image and the user can translate, scale and rotate reconstruction, until it is globally correctly placed. As the up vector is already estimated, the user needs to place the model in 2D, a fairly easy task in urban environments with many reference points, such as mapped

[1] http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html (18.05.2015)

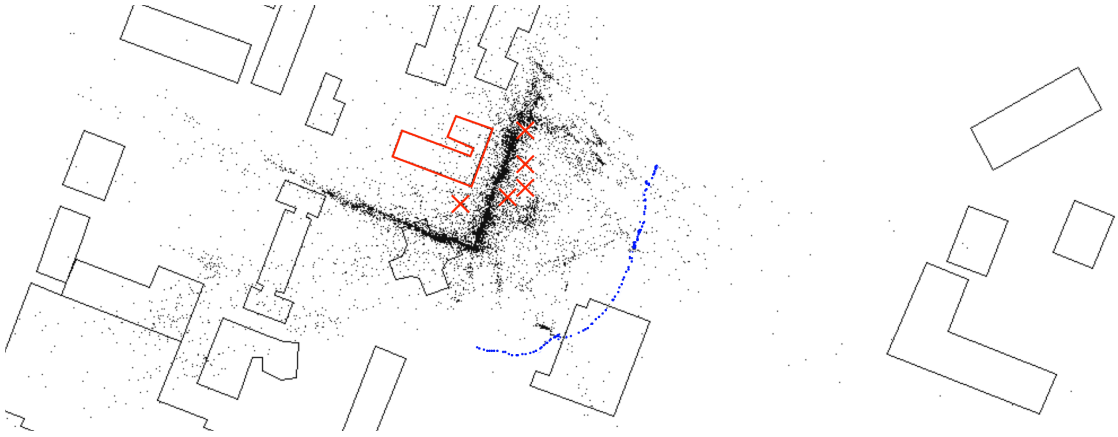

Figure 4.2: The global alignment of the reconstructed point cloud, showing, in red the Allen Hall building on the alignment image of OpenStreetMap and the known ground truth reference points, in blue the camera views of the reconstruction.

buildings. In the work of Ventura et al. the mapping material of OpenStreetMap is used for the alignment. Figure 4.2 shows a superimposed point cloud on the alignment image of OpenStreetMap.
After the global registration, we end up with an aligned reconstruction that allows us to determine global position information in metric unit. The model is placed in a right handed world reference frame with the y-axis aligned to gravity and the z-axis pointing to north. The position information is maintained in the UTM format and defines the global position of the used reference frame.

4.3 Online localization

So far, the reconstruction of the environment and the global placement is done as an offline preprocessing step on a server. The next step is the visual online localization of the camera-equipped mobile client using the preprocessed reconstruction. We use the keyframe-based initialization step of Ventura et al. [41], but rely on our own mobile orientation tracking system.
To visually localize the client, Ventura et al. proposes a feature-based method, it matches SIFT feature descriptors from the current client view against the triangulated 3D points of the reconstruction. With a parallel brute force search for each 2D SIFT descriptor, the closest reconstructed 3D point according to the Euclidean distance is found. By using the 2D-3D point correspondences and the Progressive Sample Consensus (PROSAC) [7] procedure, the Location Determination Problem (LDP) [11] can be solved and the location of the camera from the current camera frame is calculated. The LDP describes the geometrical task of determining the center of projection from given 3D world to 2D

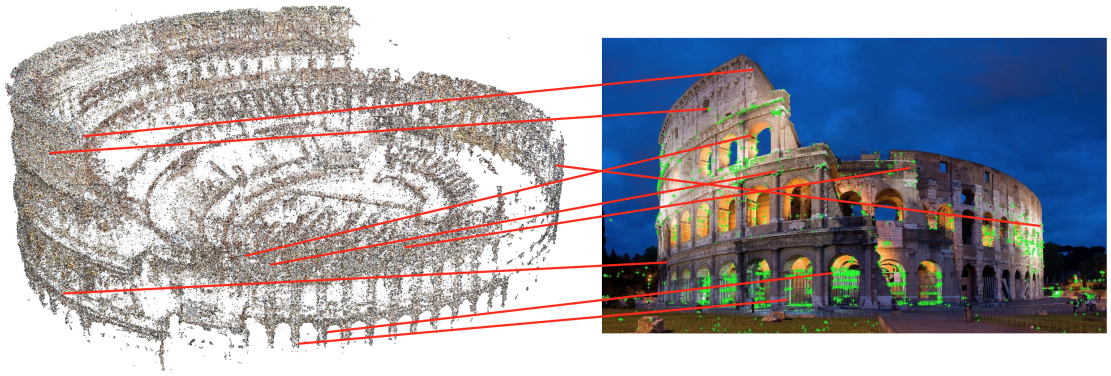

Figure 4.3: The matches between the SIFT features of the 3D point cloud and an arbitrary image of the environment, used to estimate the camera position of the image.

projection plane correspondences. The PROSAC algorithm speeds up the computation, since it considers the euclidean distance of the matches and converges, for this special case, faster than RANSAC. After the PROSAC run, 2D-3D matches are determined for a robust LDP global pose estimation. Figure 4.3 illustrates the 2D-3D matches used for the pose estimation.

Although the global location estimation can be performed on the mobile client itself, it runs faster and scales better for large reconstructions on a remote server. For performance reasons, we decided to run the approach in the client-server mode. Therefore, we implemented an Android-based mobile client, which sends compressed camera images over a wi-fi network to the server. Since the server requires the intrinsic camera parameter of the image to perform the LDP algorithm, we pre-calibrated the client's camera with the compressed images. After processing the received images the server responds with the global pose. To prevent numerical calculation problems on the mobile client, the position is send as (latitude, longitude, altitude).
For stability reasons, the position is filtered by a Kalman filter with a dynamic, linear movement model. A second static Kalman filter deals with the altitude.

The approach of Ventura et al. is designed to estimate the pose for an AR browsing system. As the currently most spread 3D situated media browsing interface, an AR browser uses the display of the mobile client as a video see-through with the situated media augmented onto it. Consequently, the camera of the mobile phone points to the urban scene of interest. Therefore, the images to localize can be sampled from the camera stream and sent to the server. However, for the pointing interface, the back facing camera of the mobile phone is pointing to the ground, and contrary the front facing camera, in up direction. In order to capture the scene for the visual localization, we mount a 90 degree

Figure 4.4: The mounted prism on the mobile client for the media pointing interface and the reflected AR browser. The image shows the prototype of the reflected AR browser, while testing the situated media browsing in front of the reconstructed environment on the campus of the University of Otago.

prism on top of the front facing camera. Beside a known 3DoF rotation, the modified client allows us to use the localization approach for our pointing interface similar as it is used by the AR browsing interface. The prism bends the light rays by 90 degrees, so that the up directed front camera captures the urban scene perpendicular to it. Figure 4.4 shows the mounted prism on the pointing client. As we use the front facing camera with the mounted prism, it changes the optics of the camera, therefore we apply a separate calibration for the pointing interface.

4.3.1 Camera readout

The major reason to implement the pointing interface under the Android platform is the variety of mobile phones running Android. As described in this section, we implement a vision-based server-client localization system requiring to read out the camera stream of the phone. In chapter 5, we describe a vision supported local orientation tracking. However, our pointing interface processes two camera streams simultaneously: one for the localization and a second one for the local orientation tracking. iPhones running iOS do not allow to access the front and back facing camera at the same time. Therefore, we

decide to build our interface under Android and use the HTC M8 [2] capable of accessing both cameras simultaneously.
Since Android relies on Java, it is a known issue to handle the camera stream in real time. Nonetheless, a fast camera access is essential for the orientation tracking as well as for the AR and reflected AR browsing interface using the camera stream as video see through. The common method is to use a native library to access the camera natively and bypass the slow Java Virtual Machine. For the following orientation tracking, we include Metaio [3] as a commercial AR tracking SDK. Using Metaio we access the camera through their native API, though Metaio can only access one camera at the time. The front facing camera with the mounted prism is responsible for the client-server localization, when the phone is in pointing mode. However, the latency of the client-server positioning allows us to access the front camera, with 12 to 15 frames per second, by the slower Java interface of OpenCV.

Figure 4.5: Sample images used for the reconstruction of the Allen Hall (campus of the University of Otago in Dunedin, left) and the resulting point cloud (right).

[2] http://www.htc.com/us/smartphones/htc-one-m8/ (18.05.2015)
[3] http://www.metaio.com/ (18.05.2015)

4.4 Evaluation

In order to test the functionality and accuracy of the presented localization approach, we perform a technical evaluation of the method. We evaluate the plain localization results against a newly introduced sliding window averaging and a linear Kalman filter. Both methods intend to smooth the localization results in order to avoid noisy positioning values, leading to inaccuracies and jitter, when browsing the situated media. We discuss the results on the basis of representative graphs and values of the evaluation.

For the evaluation of our visual localization estimation, we reconstructed the area around the Allen Hall building of the campus of the University of Otago in Dunedin. The environment around the Allen Hall is also the place for the planed user study on our interfaces. The building (figure 4.5) provides well distributed features, with a moderate amount of repetitive structures. Furthermore, the surveying department of the University provided us with differential GPS reference points around the building. Differently to the original reconstruction pipeline, we globally align the reconstruction against the known reference points, as the OpenStreetMap outline of the building is not correctly placed. Therefore, we locally add the reference points into the OpenStreetMap data and align the reconstructed camera positions, used to generate the point cloud, to its known corresponding ground truth position. Additional, we added multiple camera views taken from the known ground truth into the reconstruction. Adding more views avoids degenerated pose estimations for the evaluation, due to missing or inaccurate triangulated 3D points around the reference points. Finally, we record geo-tagged image test sequences from different reference points, taken at the same daytime as the images for the reconstruction. In order to get statistically more meaningful results, we generate three different test sequences. Our test sequences contain about 60 to 80 images in each sequence. While running the test sequence, we measured a latency of 1-3 seconds for each pose request. With this special setup, we compare the raw framework data to different refinement methods, for each test sequence.

4.4.1 Evaluation results

The localization evaluation of each test sequence results in a mean error and a standard deviation (std). The mean error is the average distance of each localized image to the ground truth position. The std value is the standard deviation of the distance value, helping to identify noise in the data. The localization values and the error are given relative to the reconstruction in meters as are the height values. In figure 4.6, 4.7 and 4.8, the results of the test sequences are presented. The color gradient of the data plot encodes the time behavior of the sequence, yellow for the first and blue for the last evaluated view.
In table 4.1, the evaluation results of the different refinement methods for all three test

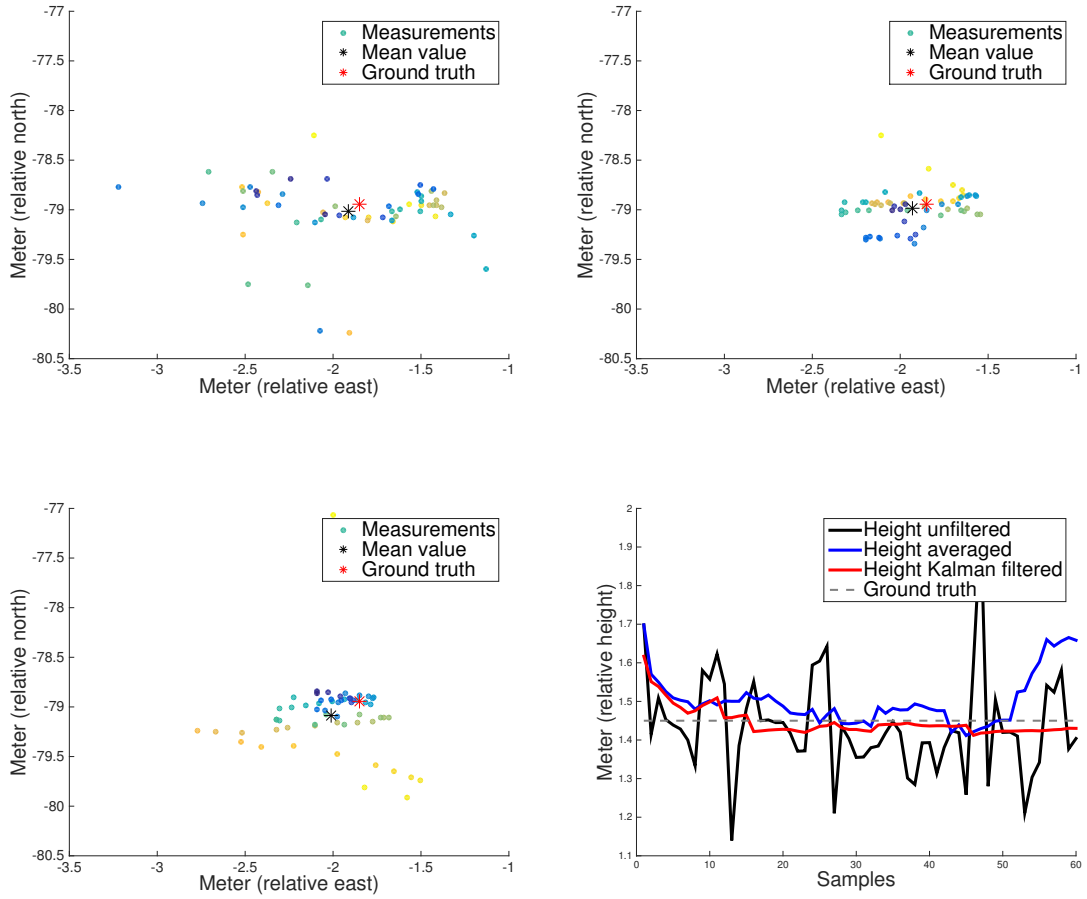

Figure 4.6: The unfiltered location samples (top, left) of test sequence 1. The sliding window averaged location samples (top, right). The Kalman filtered location data (bottom, left). The height evaluation (bottom, right). The locations are given in meters north and east relative to the origin of the point cloud used for generating the data points. The time sequence of the points is mapped to colors, yellow for the first and blue for the last position estimation.

sequences are summarized. E.g the measured mean error of the unfiltered localization in test sequence 3 is 0.99m, with a std of 0.74m. The averaging of the locations with a sliding window approach over 10 data points results in 0.54m mean distance to the ground truth and a std of 0.33m. Finally, the Kalman filtered locations end up with a mean error of 0.66m and 0.57m std. The GPS sensor recorded a position with 4.63m distance to the reference point. As the graph of the height evaluation shows, the Kalman filtered height value is the closest to the ground truth height with a mean distance of 0.08m and a std of 0.069m. The analyzed raw height value has a mean error of 0.3m and a std of 0.36m.

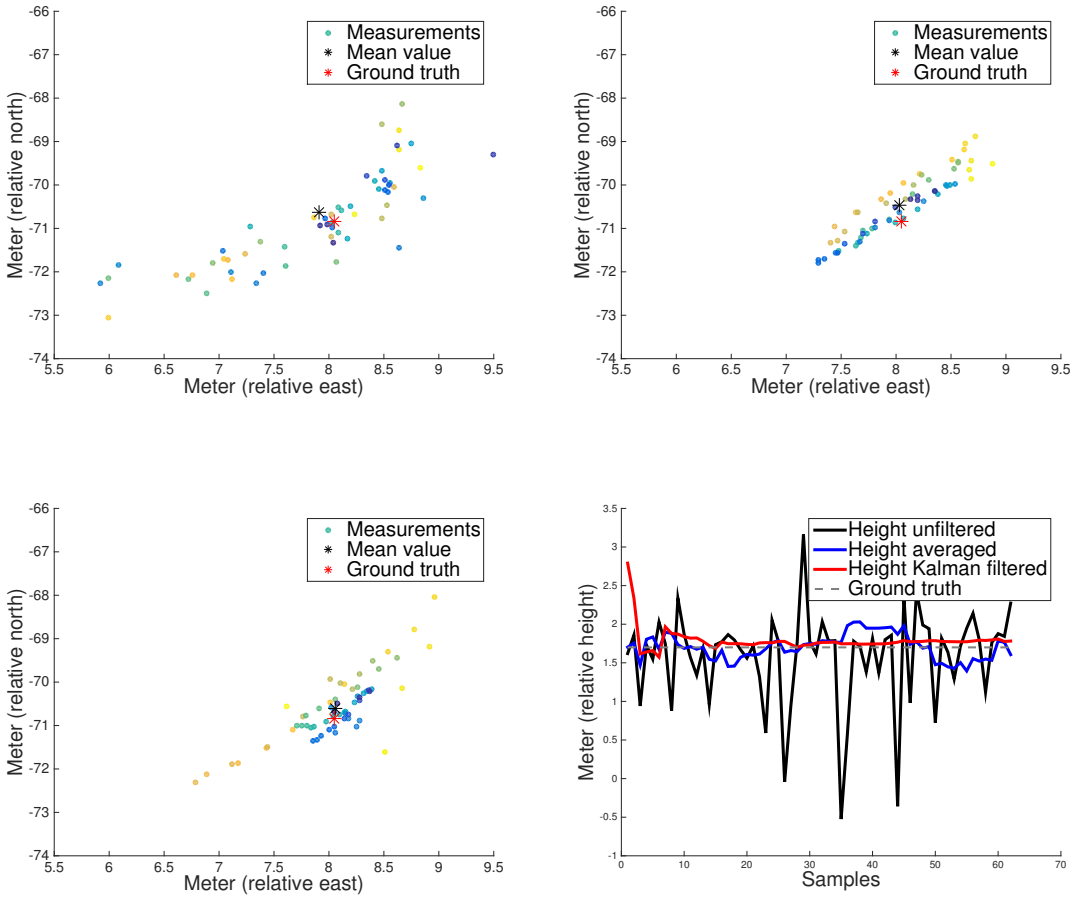

Figure 4.7: The unfiltered location samples (top, left) of test sequence 2. The sliding window averaged location samples (top, right). The Kalman filtered location data (bottom, left). The height evaluation (bottom, right). The locations are given in meters north and east relative to the origin of the point cloud used for generating the data points. The time sequence of the points is mapped to colors, yellow for the first and blue for the last position estimation.

Over all test sequences, we achieve sub-meter positioning accuracy. Using any method we significantly outperform the GPS sensor. For the height evaluation, the Kalman filtered values are the most promising, as they significantly reduce the noise of the raw data, as well as the mean error. In case of the localization, we get the best results using the sliding window averaging. However, by considering the time behavior, the Kalman filter stabilizes the positioning better than the simple averaging approach. The worse results of the Kalman filtering are explained by the initialization time of the filter, generating outliers in the beginning. After converging, the Kalman filter generates smooth position values close to the reference point. To conclude, we choose the Kalman filter as a method to prefilter the localization for the pointing interfaces.

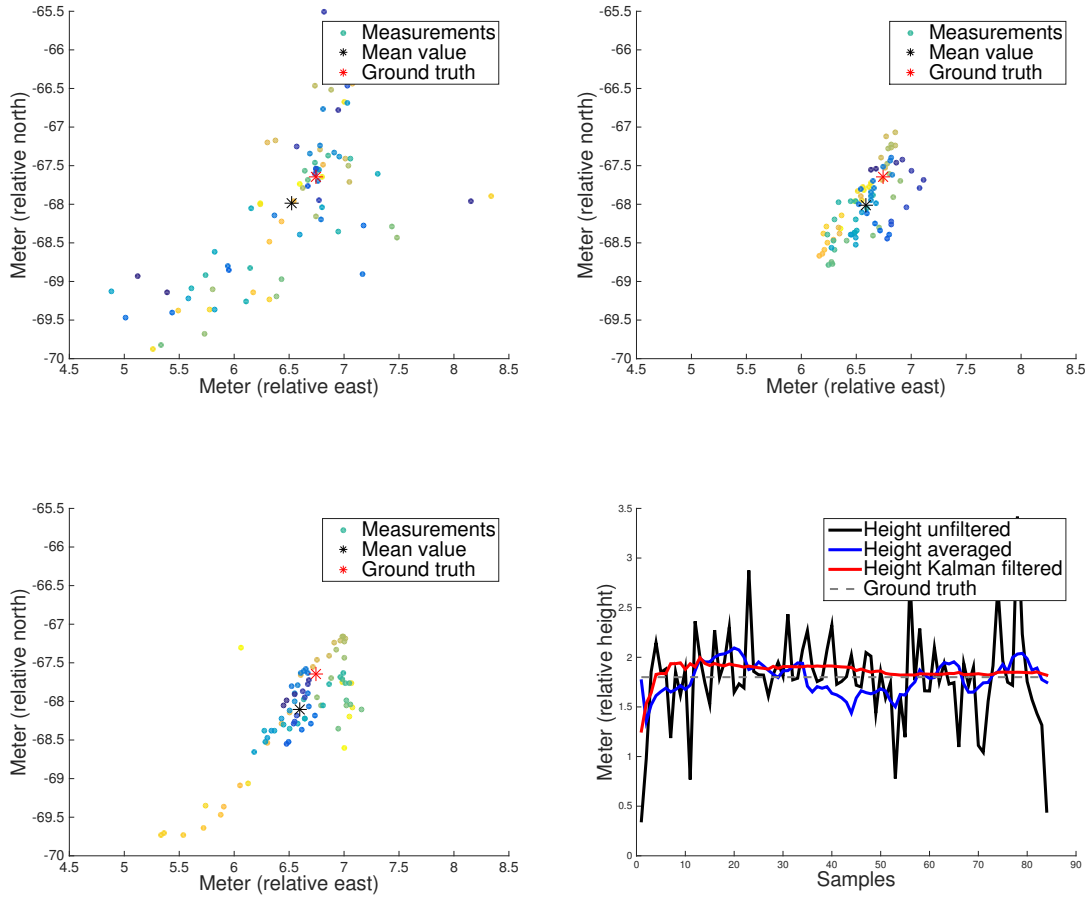

Figure 4.8: The unfiltered location samples (top, left) of test sequence 3. The sliding window averaged location samples (top, right). The Kalman filtered location data (bottom, left). The height evaluation (bottom, right). The locations are given in meters north and east relative to the origin of the point cloud, used for generating the data points. The time sequence of the points is mapped to colors, yellow for the first and blue for the last position estimation.

Additionally, we can use the evaluation to determine the object size, so that our situated media interfaces can reliable point at. In the second test sequence, we measure with 0.68m the highest mean error of the Kalman filtered positioning. Assuming a perfect orientation estimation, our localization framework limits the pointing interface to detect, on average, annotated objects of 1.36m in diameter. The value is computed by doubling the worst case mean error. With no error, the system can technically identify arbitrarily small objects. Considering the error, the object size needs to be extended by 0.68m on both sides, allowing for to detect 1.36m sized objects.

Seq. 1	**Mean pos. err.**	**STD pos. err.**	**GPS dist.**	**Mean height err.**	**STD height err.**
No filter	0.488	0.293	15.751	0.095	0.088
Averaging	0.27	0.138		0.057	0.06
Kalman	0.368	0.337		0.027	0.025
Seq. 2					
No filter	1.296	1.004	7.518	0.399	0.523
Averaging	0.796	0.457		0.155	0.158
Kalman	0.679	0.587		0.074	0.14
Seq. 3					
No filter	0.99	0.742	4.628	0.305	0.356
Averaging	0.538	0.325		0.138	0.082
Kalman	0.658	0.566		0.082	0.566

Table 4.1: Evaluation results of the visual positioning and the GPS positioning of all three test sequences. The presented error is the mean distance of the server generated positions to the ground truth, of each test sequence. The sensor values are the measured GPS offsets to the ground truth while recording the test sequences. Additionally, the height values of the visual position approach are presented and the mean distance to the ground truth height is calculated.

4.5 Summary

In this chapter, we presented the online localization approach by Ventura et al. [41]. We described its concept as well as the global alignment method to query global localization information. We evaluated the localization accuracy and compared it to the raw sensor results. Finally, we discuss the relation between localization error and the size of detectable physical objects.

5 Orientation tracking

Contents

Beside an accurate localization, the orientation of the mobile device needs to be known to browse the digital information placed in our physical environment. Contrary to the localization, the orientation has to be detected as fast as possible to build an interactive interface. To browse situated media, the mobile device is rotated to select the digital content of interest, but the position while browsing is constant except for small translational gestures. In order to reach the desired interactive speed, the method we describe runs locality on the device. We implement and adapt the method introduced by Schall et al. [34]. Their approach is to combine the internal sensors with a relative vision-based orientation tracker. Similar to the presented localization method, we evaluate our approach.
After having an accurate but low latency localization and a fast orientation tracking, we demonstrate how both methods are combined to a global north-centered pose, defining the pointing gesture of our interfaces. Finally, in the end of this chapter, we discuss the affect of the pose error to the implementation of the media pointing prototypes.

5.1 Mobile tracking system

The next technical challenge to an accurate situated media browsing interface is a north-centered orientation estimation. Ventura et al. [41] keep track of the pose received from the reconstruction request with a visual patch tracker on the reconstruction images. In contrast to the work of Ventura et al. our pointing system just requires a stable 3DoF orientation tracking to identify the users pointing gesture. The low

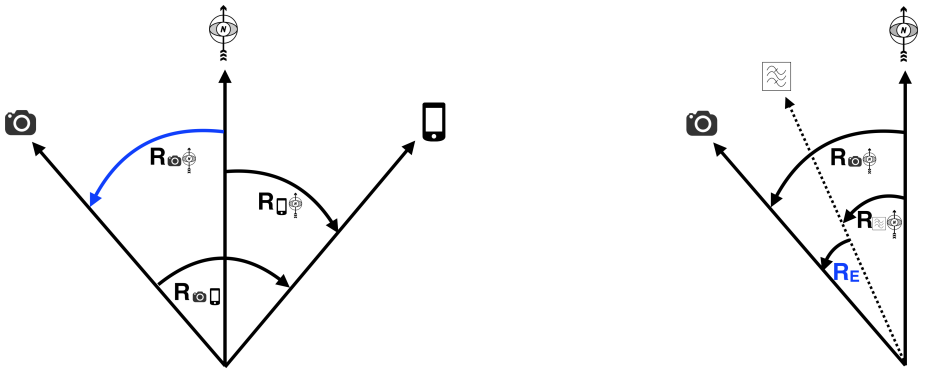

Figure 5.1: The combination of relative vision and north-centered sensor based orientation to a visual north-centered orientation estimation (left) and the Kalman filter applied on the combined orientation (right).

latency location update from the localization server is enough to keep track of the global position, whereas the local orientation tracking needs to be fast enough for user interactions. The basic approach for a local north-centered orientation tracking is to use a digital magnetic compass in combination with accelerometer or gyroscope, as described by Simon et al. [36] and Robinson et al. [32]. However, the authors claimed unstable orientation estimations, leading back to the inaccurate heading of the magnetic compass. Once more, the addition of a stable visual orientation tracking can stabilize the sensor result and allows a dynamic online calibration of the compass. We use a updated version of the north-centered orientation estimation introduced by Schall et al. [34] to get fast and reliable local orientation updates.

Combining the gravity vector of the accelerometer and the heading vector of the magnetic compass is a common method to estimate the orientation of the phone in a north-aligned reference frame. As the situated media we want to query with our pointing interfaces are placed in the real world, a north-centered orientation estimation of the pointing device is required. The magnetic compass needs to be pre-calibrated and still introduces noise into the estimation (heading error s = 1.31 degree between measurements [34]). A different approach to retrieve the relative 3DoF orientation of a mobile phone is the work of Wagner et al. [43] using the phones camera and visual features. They create a panorama of the environment on the fly and tracks 2D feature points in it. The underlying mathematical model assumes a single homography between consecutive images and calculates the 3DoF orientation around the center of projection of the camera from 2D features. Schall et al. stabilize the noisy north-centered sensor orientation by using the accurate vision-based orientation estimation.

Figure 5.1(left) shows the given orientation estimations: the sensor-based device to north orientation (R_{DN}) and the vision-based relative device orientation (R_{DV}). The first step Schall et al. propose is to combine the given orientations to get a vision-to-north estimation $R_{VN} = R_{DV}^{-1} R_{DN}$. The vision-to-north estimation is, ideally, a constant rotation, relating the sensor orientation and the relative vision orientation. By knowing the rotation R_{VN} and the relative device orientation R_{DV}, the mobile phone's north orientation can be calculated: $R_{DN} = R_{DV} R_{VN}$. As the orientations are in their own reference frame, all the involved coordinate systems need to be synchronized for these calculations.
However, the vision-to-north estimation contains the noise introduced by the sensor. To reduce the sensor noise in the repeated measurements of the vision-to-north orientation, an extended Kalman filter is applied on the orientation. The 3DoF orientation (R_E) relates the Kalman estimate ($\hat{R}_t$) at time t to the measured vision-to-north orientation $R_{VN} = R_E \hat{R}_t$, as it is demonstrated in figure 5.1(right). The three rotation angles of the innovation value R_E are Gauss-distributed with covariance P. As R_{VN} is constant, the mean value of R_E, is constant and the covariance grows through the noise introduced into the system by the sensors. The measurement equations of the Kalman filter at time t are:

$$\tilde{R}_t = I \hat{R}_{t-1} \tag{5.1}$$

$$\tilde{P}_t = I \hat{P}_{t-1} + w. \tag{5.2}$$

Because a constant is filtered, the state transition is the identity. To account for long-term system changes, a small normal distributed process noise w is added to the covariance of the innovation value. After the measurement, the predicted Kalman values are used in the update step to generate the Kalman estimate of the rotation R_{VN}:

$$R_E = R_{VN} \tilde{R}^{-1} \tag{5.3}$$

$$K = \tilde{P}(\tilde{P} + M)^{-1} \tag{5.4}$$

$$\hat{R} = K R_E \tilde{R}. \tag{5.5}$$

K is the Kalman gain, and M, the measurement covarinace of R_{VN}. Finally, the filtered vision-to-north orientation can be used to calculate the device to north orientation $R_{DN} = R_{DV} R_{VN}$ by combining it with the stable visual orientation R_{DV}.

5.1.1 Vision tracker

Schall et al. [34] refine the north orientation with the visual orientation tracker introduced by Wagner et al. [43]. The visual tracker generates a panorama of the environment on the fly and simultaneously tracks the orientation of the camera around its center of projection. Restricting the rotation around the center of the camera reduces the relation between consecutive images to 2D homographies. However, translational movements

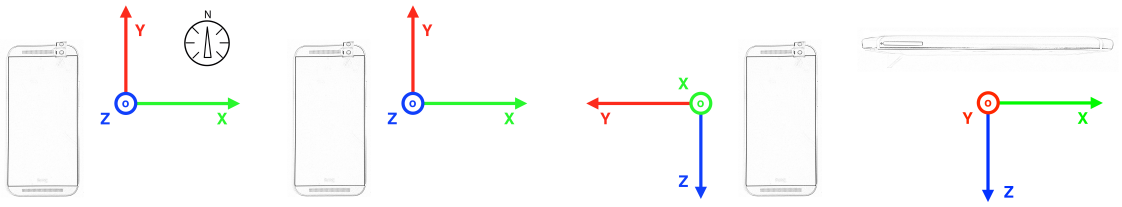

Figure 5.2: The world reference frame based on the phones sensors (left). The vision reference frame of a pointing like interface using the back-facing camera (left center). The reference frame of a rotated AR browser using the front-facing camera through the mounted prism (right center). The reference frame of a standard AR browsing interface using the back-facing camera in AR browsing mode (right).

introduce errors into the orientation tracking. As for any natural pointing gesture, translations of the device will occur. Therefore, we use a full 6DoF visual pose tracker instead of the panorama tracker proposed by Schall et al.
The recent development of commercial products in that area allow us the fast integration of such a tracker. We decided to use Metaio as a tracking SDK, since they deliver a so called instant tracking in their SDK. The instant tracker uses an initial image of the environment as a planar tracking marker and estimates the relative camera pose relative to the image. The assumption of a planar environment again reduced the mathematical model to 2D homographies. Therefore, the full 6DoF pose of the camera can estimated reliably, as long as the initial image is visible in the camera stream. To overcome the issue of the reduced field of view, we combine the instant tracker with Metaio's extended tracking feature. Running the extended tracker initializes the tracking with the planar tracking image from the instant tracker and additionally triangulates new 3D points around the tracking image. The synthesis of the instant tracker with a SLAM approach allows us to track the orientation of the phone for a wider angle of view, covering pointing scenarios. Further, the orientation tracking performs reliable, even if transational movements are introduced while tracking, since a full 6DoF pose is tracked. For the implementation of the approach, we use then the orientation component of the pose. The introduced tracker cannot track full 360 degree panoramas and assumes to be initialized in planar or far distance environments that can assumed to be planar. In order to hide those drawbacks from the user, the tracker is restarted automatically, if the tracking is lost. Restarting the vision tracker affects the introduced Kalman filter, as it needs a few frames to adapt the filter to the new relative position of the visual orientation. The filter smoothly adapts to the change and hides the reset from the user. Metaio is commonly used for AR applications and runs in real time. Critical components, like the camera readout and the tracking, run natively on the Android platform and are accessible over the JNI interface.

As mentioned in section 5.1, the Kalman filter requires all orientations to be in synchronized reference frames. Therefore, the pose estimation generated by Metaio needs to be transformed into the coordinate frame of the sensors. The sensor orientation reference frame is at the same time the world reference frame of the pointing devices. Basically, the pose returned by Metaio is a ready to use OpenGL model matrix. Accordingly, the vision tracker coordinate system is right handed, with Y being the up-vector and -Z pointing into the scene. Figure 5.2 shows the world reference frame and the vision tracker's coordinate systems for the different interface prototypes. An exception is the reflected AR browser, as the optics of the camera image used for the tracking are altered by the mounted prism. Simple coordinate system transformations can be applied to synchronize the systems for the Kalman filter.

At this point we want to mention that during the work on this thesis, Apple has acquired the Metaio GmbH. The reproducibility claim of this thesis, therefore, just holds till the end of 2015, as the free Metaio SDK used will be available till then. However, the freely available Vuforia SDK provides a similar combination of extended tracking and on-the-fly marker generation. The use of the Vuforia SDK should lead to the same results as presented in section 5.2 of this thesis.

5.2 Evaluation

Similar to the evaluation of the original work from Schall et al. [34], we measure the rotation around the Z-axis of the sensors reference frame. The rotation around the Z-axis is the north bearing of the mobile device, defined by the digital compass. As the compass is the main source of inaccurate orientation estimations [35], evaluating its behavior defines the quality of the north-centered orientation estimation. In order to measure the quality of the northing, we mount the pointing device onto a tripod capable of measuring angles. Since our tracker is designed to track 6DoF, the device does not need to rotate around the center of projection of the tracking camera. In order to measure the error, we compare the rotation to known ground truth bearings. Therefore, we initialize the measurement by pointing the device to survey marks with known bearings. After the initialization, we rely on the measured angles of the tripod as ground truth and compare these known orientations to the tracker data. The survey reference points are pillars on the rooftop of the surveying building of the University of Otago. With this setup, we evaluate the bearing of the pointing interface, using the back-facing camera in direction of the ground for the vision tracker. Similarly, we measure the bearing of the AR browser interface using the back-facing camera in direction of the scene as tracking camera. Finally, the reflected AR pointing interfaces rotation is measured, tracking the scene using the front-facing camera through the mounted prism. The interfaces are all evaluated independently, as the different camera and optics affect the visual tracking system, which further affect the presented hybrid tracker. The differing reference frame transformations of each interface

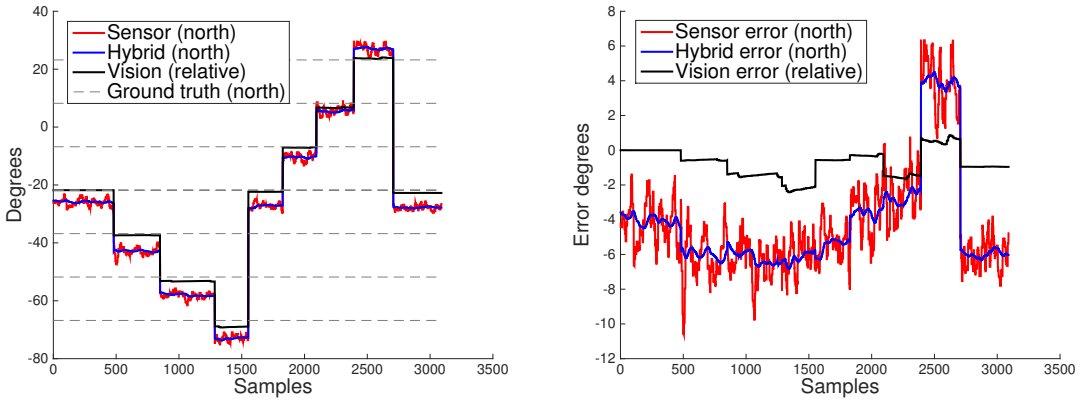

Figure 5.3: The ground truth data, the logged north bearing of the raw sensors and the hybrid, as well as the relative vision tracker (left) of the pointing interface. The performance of the error over time (right). The x-axis of the graphs is in degrees, while the y-axis encodes the time behavior.

prototype can be tested simultaneously.
Different to the earlier work of Schall et al. [34], we adapt the evaluation to the newly introduced 6DoF tracker and a pointing scenario. Instead of comparing the bearing to ground truth covering a full 360 degree rotation, we reduce the field of view to a realistic pointing scenario of 90 degrees. This restriction reduces tracker restarts, forcing the Kalman filter to adapt to the new relative visual orientation. The tracker restarts occur as the used tracker is designed for 6DoF pointing instead of full 360 3DoF orientation tracking. Due to a different mathematical model, the 6DoF tracker assumes the environment to be planar and can, therefore, track a full pose relative to this static environment. However, if the assumption of a planar environment is violated, the pose tracking fails and is restarted. The original 3DoF orientation tracker assumes rotations around the center of projection of the camera and does not allow for translational movements, which is usually not the case for pointing gestures.

5.2.1 Evaluation results

In figure 5.3 we present the behavior of the relative vision tracker, the raw sensors and the combined hybrid tracker compared to the ground truth orientation of the pointing interface. The error evaluation shows a mean error of 4.16 degrees for the raw sensors north bearing and 4.14 degrees for the hybrid sensor. The std of the raw sensor data is 3.22 and 2.98 degrees for the hybrid. The relative vision tracker is plotted into the graph but as its value is in a relative reference frame and possible restarted any time, no error evaluation is presented. Even though the evaluation produces similar results for the raw sensor bearing and the hybrid tracker, a small correction of the bearing is recognizable.

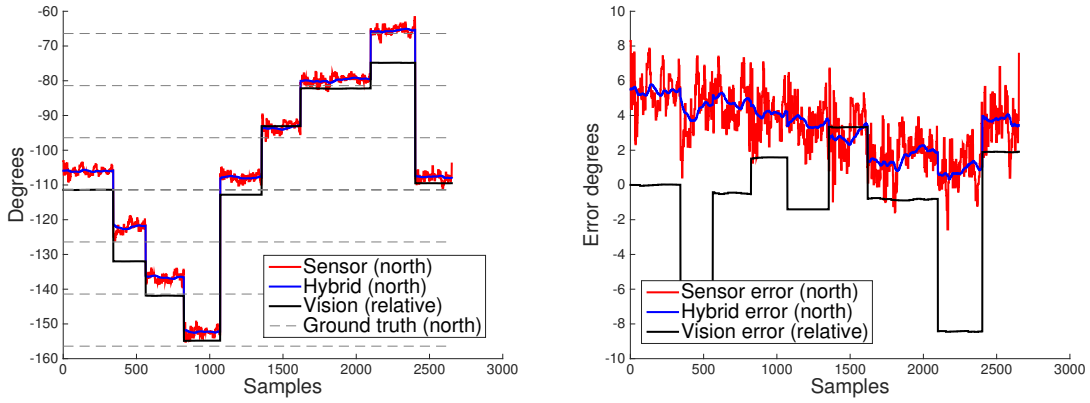

Figure 5.4: The logged north bearing of the reflected AR browser (left). The performance of the error over time (right). The plots show that the use of the reduced viewing angle of the front camera and the changed optics of the prism affects the performance of the vision tracker. However, the tracker is still precise and can reduce the noise of the sensors.

	AR browser	**Reflected AR browser**	**Media pointing**
Sensors mean error	-1.435	3.331	4.165
Hybrid mean error	-0.903	3.316	4.149
Sensors std. error	1.917	1.894	3.222
Hybrid std. error	5.435	1.519	2.985

Table 5.1: Evaluation of the north bearing error, of the raw sensor data and the introduced hybrid tracker.

Nonetheless, the noise of the sensors can be successfully removed by the Kalman filter combining the raw data with the visual tracker.
The graphs for the AR browser interface are shown in figure 5.4 and the reflected AR browser interface in figure 5.5. Finally the evaluation results are combined in table 5.1. The values consistently show an improvement of the hybrid tracking approach. Even though the mean sensor orientation offset is not always severely outperformed by the hybrid, the std. as well as the plotted graphs show a noticeable noise reduction.

The results demonstrate improvement over the raw sensor data. However, the real online calibration happens while comparing the sensor orientation to the relative vision orientation. Our evaluation method does not allow us to generate ground truth data, while rotating the device. Figure 5.6 shows the behavior of the tracker for rotations around all axis, while pointing. The plotted graphs demonstrate how the hybrid tracker is generated as a combination of the visual orientation tracker and the sensor data.
Finally, by assuming a perfect localization, the measured orientation error affects the size of the physical pointing targets. The required object size to distance ratio in dependence

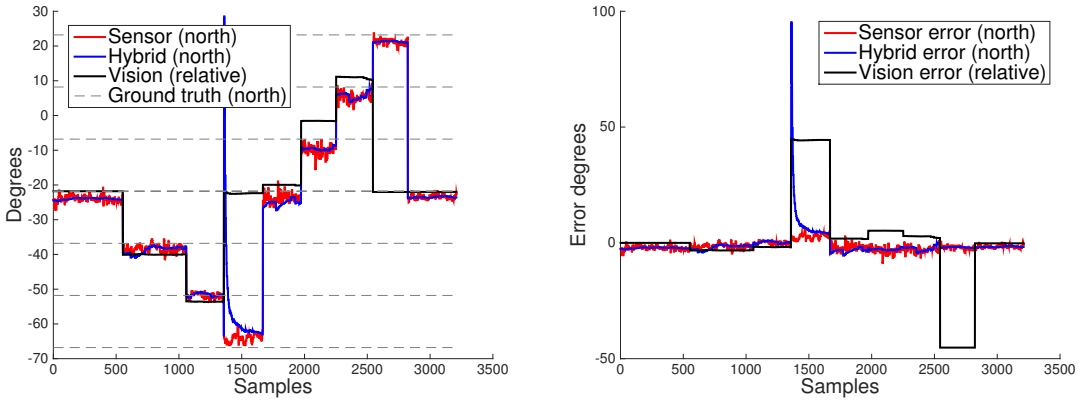

Figure 5.5: The logged north bearing of the AR browsing interface (left). The performance of the error over time (right). The outlier of the hybrid tracker is explained in a vision tracker restart. The restart occurs twice during the evaluation as the curve of the vision tracker jumps back to a near zero bearing.

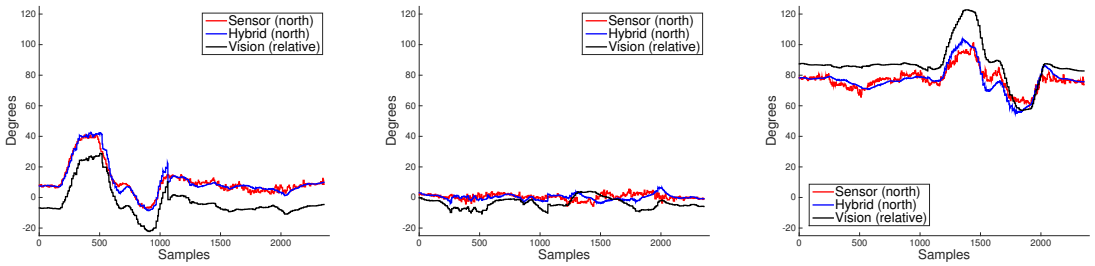

Figure 5.6: The time behavior of a natural pointing gesture for the X-axis (left), Y-axis (middle), Z-axis (right).

of the error is given as 0.7m for 5m of distance, 1.4m for 10m distance, 7.2m for 50m distance, 14.5m for 100m and 29m for objects of 200m distance. The values are samples of the orientation error function $o(\alpha, r) = 2(r \tan(\alpha))$, with α being the worst case orientation error of 4.14 degree, and r, the distance of the pointed target. The value in the error function is doubled to get the expected minimum size of the physical pointing target.

5.3 Localization and Tracking

After having a global position estimation [41], as presented in chapter 4, and the real-time north orientation tracking [34] we explain the combination to a global north oriented pose,

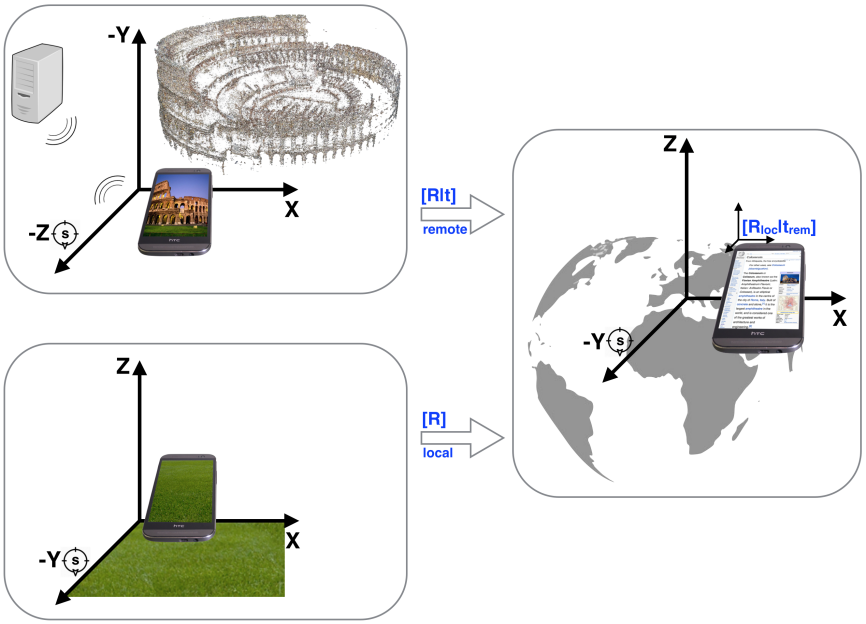

Figure 5.7: The combination of the remote global localization framework with the local orientation tracker to a global north-centered pose used pointing.

further used in our prototypes. We demonstrate the combination in figure 5.7. The mobile device sends images of its current position to the localization server. The server matches the images against the globally aligned reconstruction of the environment and returns the pose of the image to the client. Simultaneously, the client estimates its orientation in real-time by combining the phones sensors with the visual tracking approach. In a final step the real time orientation is combined with the server localization to the final pose, defining the pointing gesture. As demonstrated in figure 5.7 the localization framework has its own coordinate system, which needs to be converted to the world coordinate system defined by the sensors of the client. The local tracker runs already in the world coordinate system as it is supported by the sensors of the phone. The orientation of the localization approach is ignored, its latency is unsatisfying for interactive media selection. The location update is fast enough, as the browsing gestures are done from a static position.

5.4 Summary

Knowing the technical details and the evaluation results of the global pose generation, finally, we discuss the effects of the results to the concept of situated media browsing. Regarding the evaluation, we achieved, in worst case, sub-meter position accuracy (0.68m) and an expected orientation error of 4.14 degrees in our well-controlled evaluation environment.

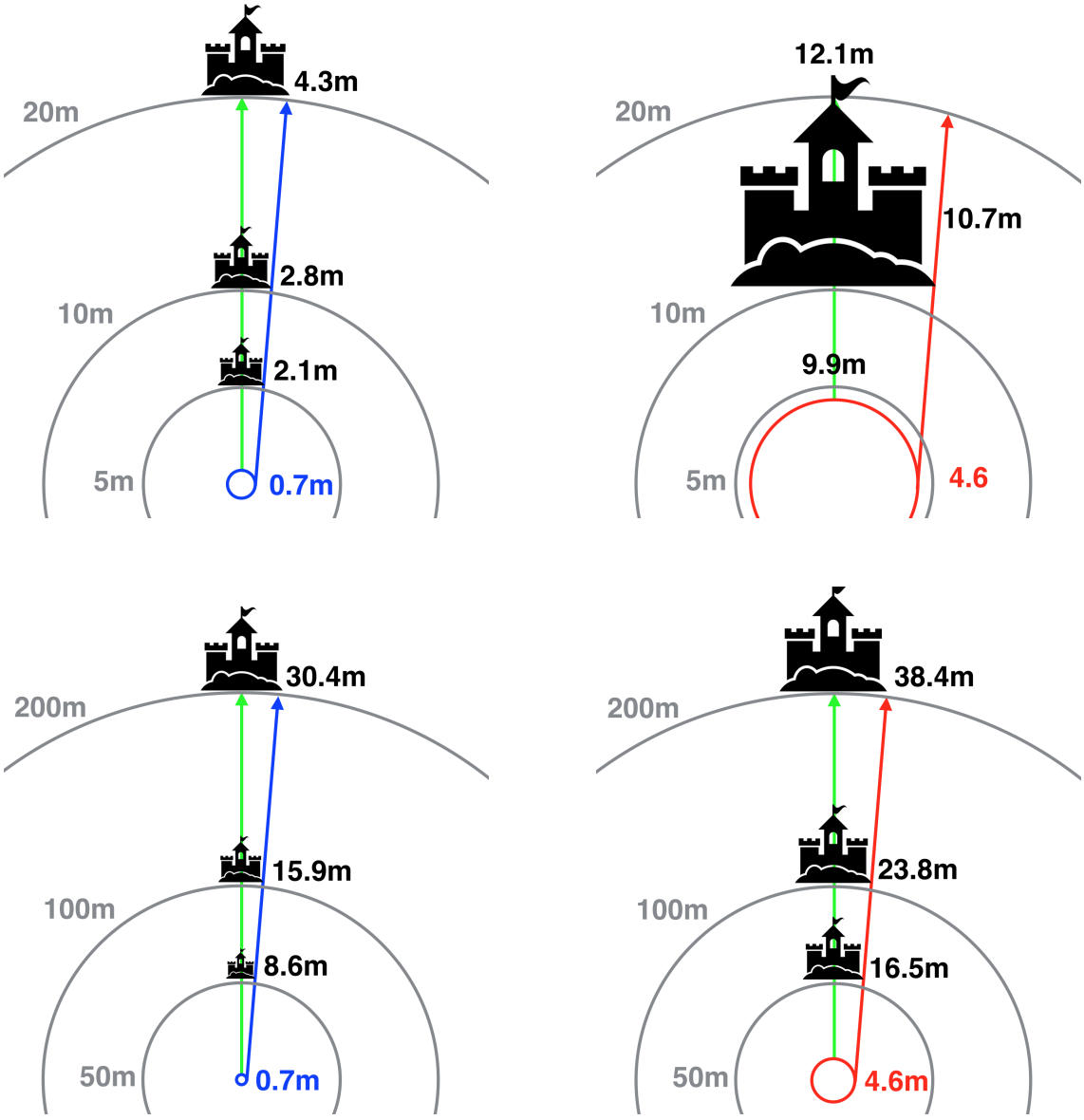

Figure 5.8: The expected size to object distance ratio, by taking the technical limits of the localization and orientation framework into account. On the left side, the diagram shows the values of our method. On the right side, the values of the raw sensors are demonstrated.

In figure 5.8 (left), we combine the evaluation results and show the technical limits of our approach. In figure 5.8 (right), the results using the raw sensor data are illustrated. We demonstrate the relation of the individual technical limits to the distance and size of real world pointing targets. The green arrow is the ground truth orientation, while the blue and red arrow shows the orientation error. The circle is presenting the localization uncer-

tainty. By combining both average error estimates, we can calculate the size of objects we can statistically expect to detect from different distances. These technical limits are further important for the interface design. Considering the illustrated results, we observe a clear accuracy improvement of our method compared to the naive sensor usage.
From a mathematical point of view the values are rounded samples of the equation:

$$l(e) = 2e \tag{5.6}$$

$$o(\alpha, r) = 2(r \tan(\alpha)) \tag{5.7}$$

$$s(e, \alpha, r) = l(e) + o(\alpha, r) \tag{5.8}$$

combining the error function of the localization estimation and the orientation estimation. The used variable values are the expected worst case values, deduced from the technical evaluations of the raw sensor data and our combined approach.

6 Prototypes

Contents

This chapter addresses the technical aspects of our prototypes. After presenting accurate location and orientation estimation methods, we discuss the technical details, needed to implement the prototypes. One aspect to consider first is the environment and type of situated media browsed by the prototypes. Following the definition of the media type, placement and accessibility in the physical world, we discuss the level of detail. The level of detail is our method to compensate for technical and interaction related inaccuracies. Furthermore, we explain the developed prototype of each individual interface concept. Technically, the prototypes are implemented as application for the Android operating system. We describe the observed interface of the prototypes, as well as the usage and the ergonomics. Additionally, we state our expectations with respect to the social acceptance of the interfaces, in form of hypotheses for an upcoming user study. We summarize by discussing the contribution of our interfaces as novel method of browsing situated media.

6.1 Situated media

In order to build the AR browser, the reflected AR browser and the media pointing prototype, we need to define a test scenario and the situated media content browsed by our prototypes. Because of the technical implementation of the global pose estimation,

it is required to provide environment information in form of a reconstructed point cloud. The reconstruction of the selected urban environment is hosted on a server, allowing our prototypes to query accurate localization information. For the performance evaluation of the global localization method [41], we reconstructed the area around the Allen Hall building at the campus of the University of Otago. We reuse the reconstruction as the urban test environment for our prototypes.
The Allen Hall is the home of the Theater studies of the University of Otago. We generate related situated media content. We annotate the campus of the university, the Allen Hall building, the floors and the windows of the Allen Hall building with 3D digital information. We collect a title and a brief description of the objects. We also add an object related image and, for the pointing interface, a picture of the object itself. All together, the situated media content is attached to the physical objects. For example the situated media content attached to a window of the Allen Hall contains, the name of the lecture hall behind the window, a summery of the given lectures, an image of the lecture hall and a picture of the window itself. The content and its position in the physical world is stored as a JSON-serialized object structure directly on the mobile device, parsed when the prototype is initialized. Similar as a configuration file, the JSON-file allows an easy editing of the situated media.
In order to query the situated media of a physical object, we also need a representation of the object inside the application. Because of their mathematical simplicity to define and intersect with, we choose spheres to represent the physical objects. We want to select the object of interest by intersecting it with an imaginary ray, originating inside our smartphone. In the previous chapters 4 and 5, we described how we accurately estimate the physical pose of the smartphone. The pose estimate can be used to define this imaginary ray. In dependence of the prototype, we get a vector pointing in the same direction as the camera for the AR browser, or along the axis of the phone for the pointing prototypes. Finally, to access the situated media inside our application, we intersect the vector defined by the physical pose of the smartphone with the spheres representing the physical objects.

6.2 Level of detail

For an accurate localization, we use an outdoor tracking framework for urban environments, likewise, our prototypes are designed for urban areas. Consequently, we are able to attach information to different sized objects in our environment. E.g. we can annotate whole city blocks, buildings, parts of a buildings such as storyes and finally, arbitrary objects. Representing the different sized objects of an urban environment we define four selectable level of detail (LOD): block, building, floor and object level. The objects of the same level are not intersecting, but can contain objects of smaller size. Buildings stand next to each other, however, a building can contain floors or windows with attached situated media. One level is selected and therefore, all prototypes are able to display exactly one situated media content at a time. In case more objects standing in front of each other

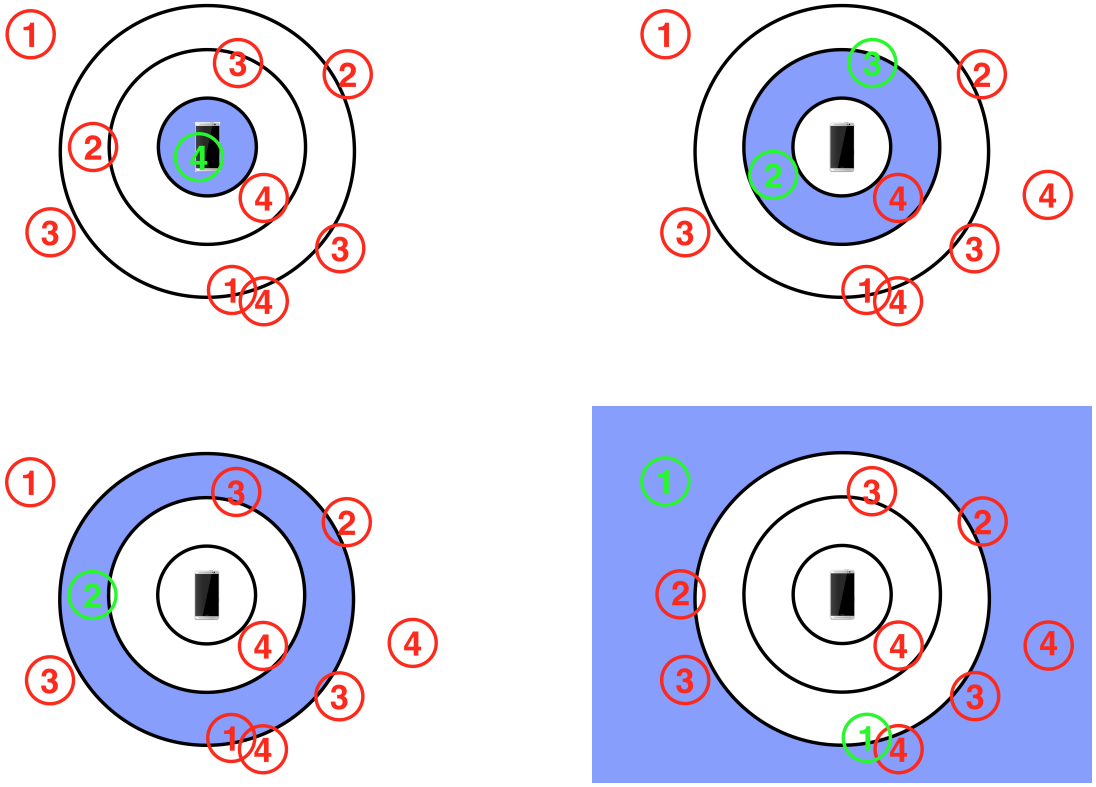

Figure 6.1: The level of detail selection scheme to automatically chose between block(1), building(2), floors(3) or object(4) level, based on the users location. Starting from the window level zone, closest to the user, the zone is filtered for window level objects. When objects of the current zone are detected, the level selection ends with the current zone selected (indicated in green).

are detected, the content of the object closest to the user is displayed.
The LOD concept is introduced to first, compensate for technical and interaction method based inaccuracies of the system and second, reduce the interface layout to present just one media content at a time. E.g. from the technical point of view, it is not possible to query the situate media attached to a window on a distance of 100m, however, information attached to a building can be accessed. The LOD is chosen and updated automatically by the system, based on the users location. In figure 6.1, we demonstrate the used scheme. First, we divide the area around the user into zones. Next, the zones are iteratively scanned for annotated objects, starting with the zone closest to the user. If in the first zone, annotated objects of the object level are detected, we set the LOD to object level and terminate. Otherwise, we continue with the next bigger zone and search for annotated floors if a floor within the range is detected, we terminate. We continue with the same procedure for the building level. We terminate by choosing the block level, the most gen-

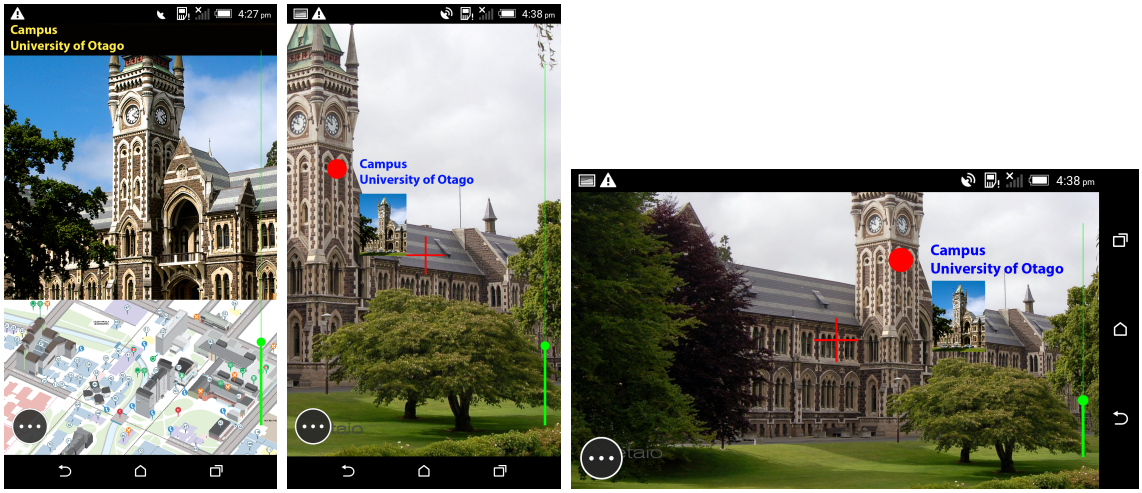

Figure 6.2: Layout of the prototypes implementing the pointing interface (left), the reflected AR browser (middle) and the state of the art AR browser (right). All prototypes contain the level of detail slider as well as a button to access the settings menu.

eral level, if no floors in the third zone are detected. This procedure is designed to select a LOD that contains at least one reliable detectable objects inside the accuracy range of the interface.
However, the LOD concept affects the interface of the prototypes, as we allow the user to manually readjust the selected level. The readjustment can be done by changing the slider position available in each interface (figure 6.2). If the user selects a level above the automatically selected likely ending up in inaccurate browsing, the slider changes its color to orange and red, indicating possible erroneous behavior.

6.3 Settings

Opposite to the LOD slider, each prototype also contains a button to access the settings menu. In figure 6.3, we show screen shots of the settings menu. The menu is mainly designed to select between the different prototype implementations. However, different adjustments of technical nature are accessible using the menu:

- the prototype selector, allowing to switch between the different prototype implementations
- the prefix name for the generated files of an evaluation run
- the switch to display visual on-screen debug information
- the switch to use a simple sensor fusion instead of the presented visual supported orientation estimation

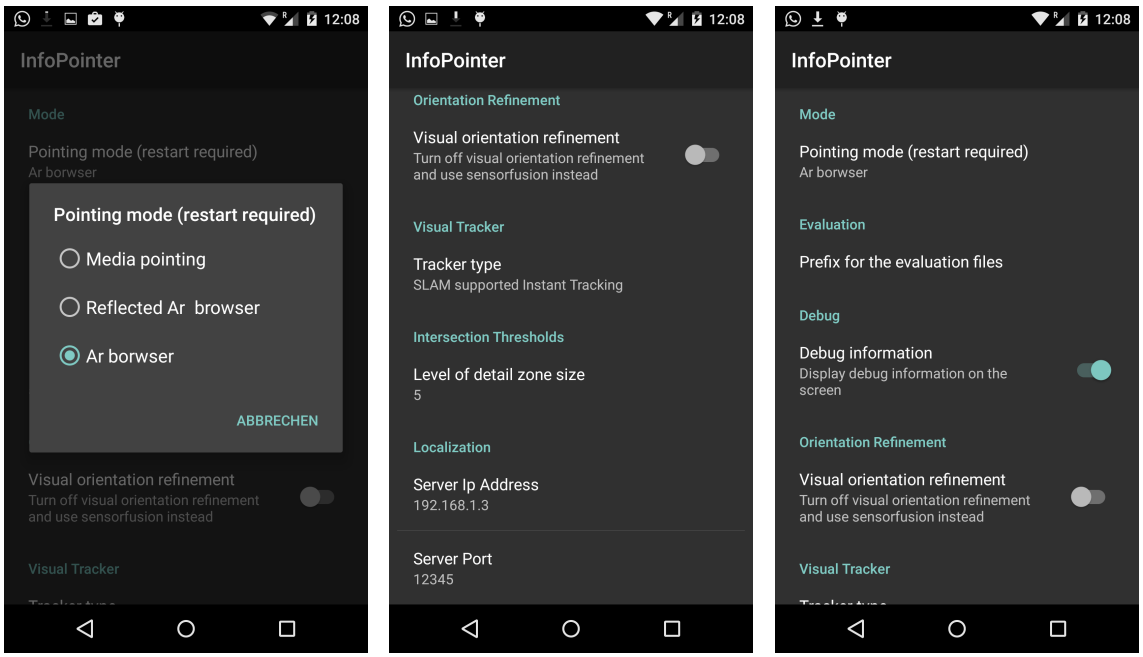

Figure 6.3: The settings menu to adjust different technical details. The selection menu of the different prototypes on the left, the list of the full settings menu is presented in the middle and on the right.

- the selector for different visual tracking methods
- the size of the level of detail selection zones
- the server ip-address and port to the localization server.

6.4 AR browser

The first interface we discuss is our implementation of a hand held video see-through AR browser. In the following we refer to it as AR browser. We describe the technical implementation as well as the observed ergonomics of the AR browser. Considering the definition of AR browsing by Grubert et al. [12], our interface augments situated information onto the vision of the physical world on a smartphone. The vision of the physical world is created by rendering the back-facing camera view on the smartphone's display. The camera view is blended with the situated information the user is pointing at. We demonstrate the basic idea of the interface in figure 6.2 (right). A central cross-hair is placed onto the display to allow accurate pointing, as it eases selecting the situated media on physical objects or locations. To indicate objects with attached situated media, they are marked with dots on the camera stream. Following, we browse situated media by

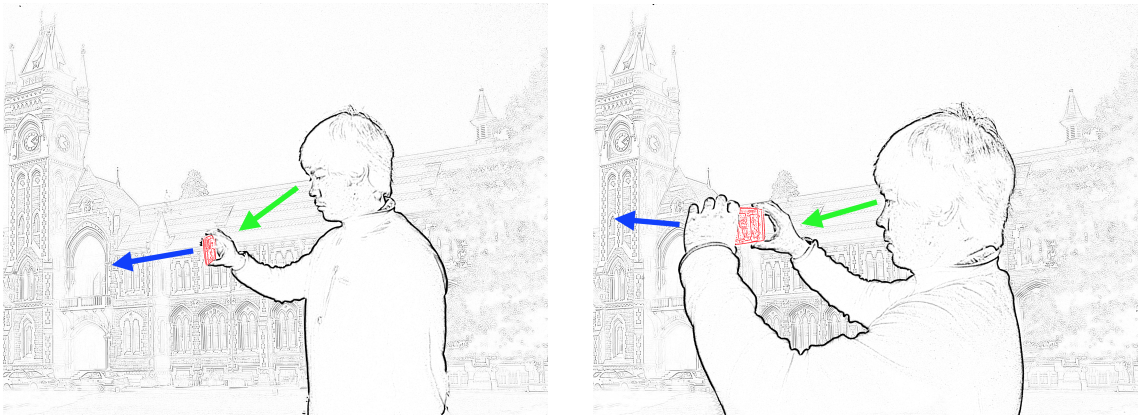

Figure 6.4: Interaction and ergonomics of our AR browser. Holding the device one-handed, shoulder height, able to see the physical world and the augmentation of the situated media on the display (left). The device is statically held two-handed in front of the head, exploring the physical world by looking to the screen (right).

pointing the center of the camera towards the physical object we are interested in. The intended pointing orientation describes a vector which intersects with the physical object. Finally, we render the situated media onto the selected object.
We design the interface for the landscape mode, maximizing the filed of view of the camera. To render the camera stream, we use the Medaio SDK that we already integrated into our prototypes for the visual tracking (chapter 5). To render the camera in full-screen, we use OpenGL ES[1] and an Android SurfaceTexture. The Metaio SDK allows us to access the stream data in real time using their native Android library. The situated media of a selected object is correctly placed on the virtual object of the camera stream using standard Android GUI elements. To find the position of the object on the screen we also use OpenGL. We can take the known physical pointing pose to define a virtual camera for OpenGL. This camera allows us to identify the position of the object on the screen. Technically we back-project the location of the visible object through the camera to the screen. On the known screen position we display the situated media.

According to our observations there exist two interaction modes while using the AR interface. Interacting with the device one-handed, shoulder height, able to see the physical world as well as the situated media augmented onto the device. Users focus on the physical world and use the device to point and see the situated media. Pointing is done dynamically out of the wrist (figure 6.4 left). Alternatively users interact holding the device statically in front of the head. Pointing is done by moving the head and the device simultaneous (figure 6.4 right).
Grubert et al. [12] investigate usage pattern and social implications of AR browsers. The

[1] https://developer.android.com/guide/topics/graphics/opengl.html (12.09.2015)

common usage pattern is rotational pointing from a still standing position. The survey of Gruber et al. demonstrates that almost half of the user experienced occurrences of social issues while using the AR interface. Generally, we did not experience usability issues, while working with the prototype. The handling to query the situated media content is intuitive and clear. However, we think the concept of using an optical see through to browse situated media is not ideal. AR methods require to render the camera stream as a copy of the physical world. The screen is fully occupied buy the camera stream, which needs to be visible to browse the situated media by camera aiming. When selected, the situated media is augmented on the camera stream. Consequently, we need to reduce the size of the situated media, to allow enough space for the camera stream. A different solution is to augment labels indicating the situated media content and opening additional views to present the full digital content. The required screen space for the camera stream affects the situated media content presentation and the usability of the interface.

6.5 Media pointer

The media pointer is different from the AR browser. The media pointer is not an optical see through, as it follows a different browsing concept, but is still designed to access situated media in 3D. The idea of the media pointer is to browse the situated media by pointing the device towards the object of interest instead of aiming the camera. Similar interfaces are already known in the literature [9][32][36][35][47], but suffer from poor accuracy so far. The presented prototypes are based on sensors, which are often affected by the environment and ,therefore, hard to control. Considering the technical performance analysis of chapter 5, we present a pointing interface with comparable accuracy as the known AR browser.
In figure 6.2 (left), we depict the prototype of the pointing interface. We illustrate a screen shot of the device, while pointing it towards the university campus. On top of the screen, we display the title and the description of the situated media being accessed. Under the description, two third of the available display space are used to present the browsed media information, e.g. an image of the campus. On the bottom of the display, we place a small representation of the physical object, as a user feedback. In this example, we show the campus map, as the map is stored as part of the situated media attached onto the campus. In contrast to the optical see through interface, the pointing interface does not provide a copied vision of the physical world. Therefore, the static view of the object adds the missing connection between physical and digital world. We found out that this connection improves the usability of the interface, as it helps the user to identify the aimed object. Because of the static layout, we realize the GUI using standard Android GUI elements.
The selection of the digital content is technically similar to the AR browser. We display the situated media content, when the vector along the axis of the phone intersects a physical object with attached situated media. We calculate the vector over the physical

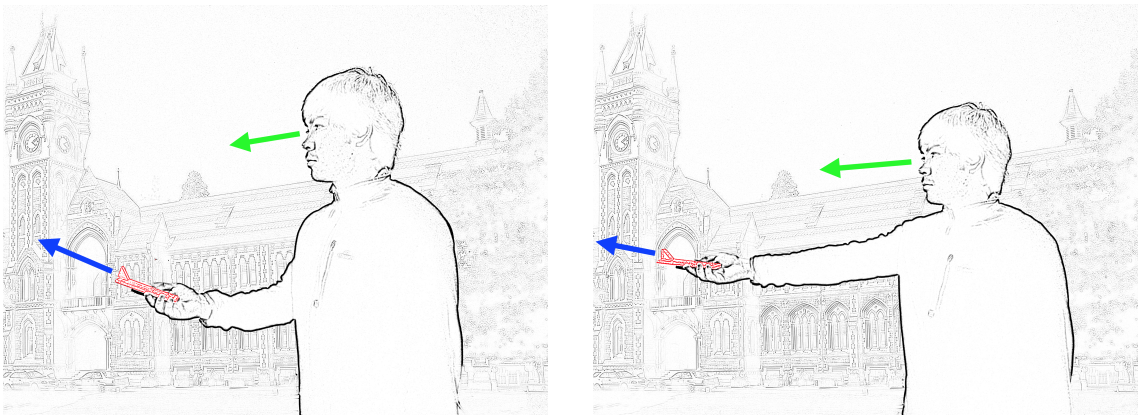

Figure 6.5: The situated media pointing interface. The arm at waist level (left) and the arm outstretched (right). The user looks to the object of interest, but is also able to see the situated media displayed.

pointing pose of the device.

Similar to the AR browsing interface, the digital media is available as long as the physical object is selected by pointing. The pointing is executed by holding the mobile device in portrait mode towards the physical object of interest. Generally, pointing is less accurate by a magnitude of more than two, compared to pointing by aiming a camera [48]. Therefore, we identified two use cases of our presented situated media pointing interface. In figure 6.2 (left), we present one use case. The user holds the phone one-handed at waist height, points and looks towards the physical object. Due to the reduced connection of the virtual and physical world, the attention of the user is forced to be in the physical world.
The other use case we identify is pointing with outstretched arm. Similarly as for the first use case, the user focuses the physical world, but is still able to identify situated information on the screen. We show the observed use case in figure 6.2 (right). The users achieve a higher degree of accuracy with outstretched arm. We think that the changed eye-to-arm position allows a more accurate distant object selection. The use case can be compared to pointing a handgun with outstreached arm using the ironsights of the gun. We expect users to change between both use cases in dependence of distance to the object and social environment.
Generally, we expect less social issues, when using one-handed user interfaces. In terms of usability, the presented media pointing interface provides more screen space for the situated media content. Further, the situated media content is placed statically on the screen, while the AR browser adjusts to position of the content with the pose of the phone (section 6.4). Summarizing, we argue that pointing is an intuitive and easy to use 3D

situated media browsing interface.

6.6 Reflected AR browser

The last interface we developed, to our knowledge, is not found in the literature yet. We introduced it as reflected AR browser in this thesis. The interface is basically combining aspects of the AR browser and the media pointer. It is identical to the described AR browser, a hand held video see-through. The layout of the interface is occupied with the vision of the physical world, and the situated media augmented on it. Therefore, the browsed information is placed in the same way next to the central cross-hair, as we demonstrate in figure 6.2 (middle). However, we browse the situated media by pointing the device similar as the media browser towards objects. Following, we use the reflected AR browser in portrait mode. In order to do so, we mount a prism on the front-facing camera.The prism bends the camera view by 90 degree and allows us to render the scene in front of the user, while holing the device like the media pointer in portrait mode. Finally, the interface's ergonomics is pointing based, while the object selection achieves the same accuracy as camera aiming [48].

In figure 6.6, we show the use of the reflected AR browser. The ergonomics is similar as for the media pointer. The user focuses the physical world, while accurately pointing using the camera stream and the cross-hair on the screen. However, because of the one-handed ergonomics, we argue that the reflected AR browser is less obtrusive than the AR browser. On one hand, we achieve a higher pointing accuracy because we use camera aiming, while on the other hand, we have reduced screen space for the media content, because of the rendered camera stream.
Summarizing, the interface is clean and easy to use we did not observe usability problems, while using the interface. Technically the interface is implemented identical to the AR browser explained in section 6.4. The only difference is a static coordinate system transformation in the OpenGL rendering, because the layout is in portrait instead in landscape mode.

6.7 Interface discussion

Based on the presentation of the different prototypes, we discussed the newly introduced interfaces in comparison to the well studied AR browser. We compared usage, media selection accuracy, ergonomics and social implications.
Grubert et al. [12] demonstrate in user studies that almost half of the participants experience social implications, while using AR browser. By reviewing the ergonomics of the interfaces, it turns out that the AR browser is the only two-handed interface presented. We argue one-handed interfaces are more naturally to use and socially more accepted.
Next, we focus on the layout of the interfaces. The AR based interfaces differ in their

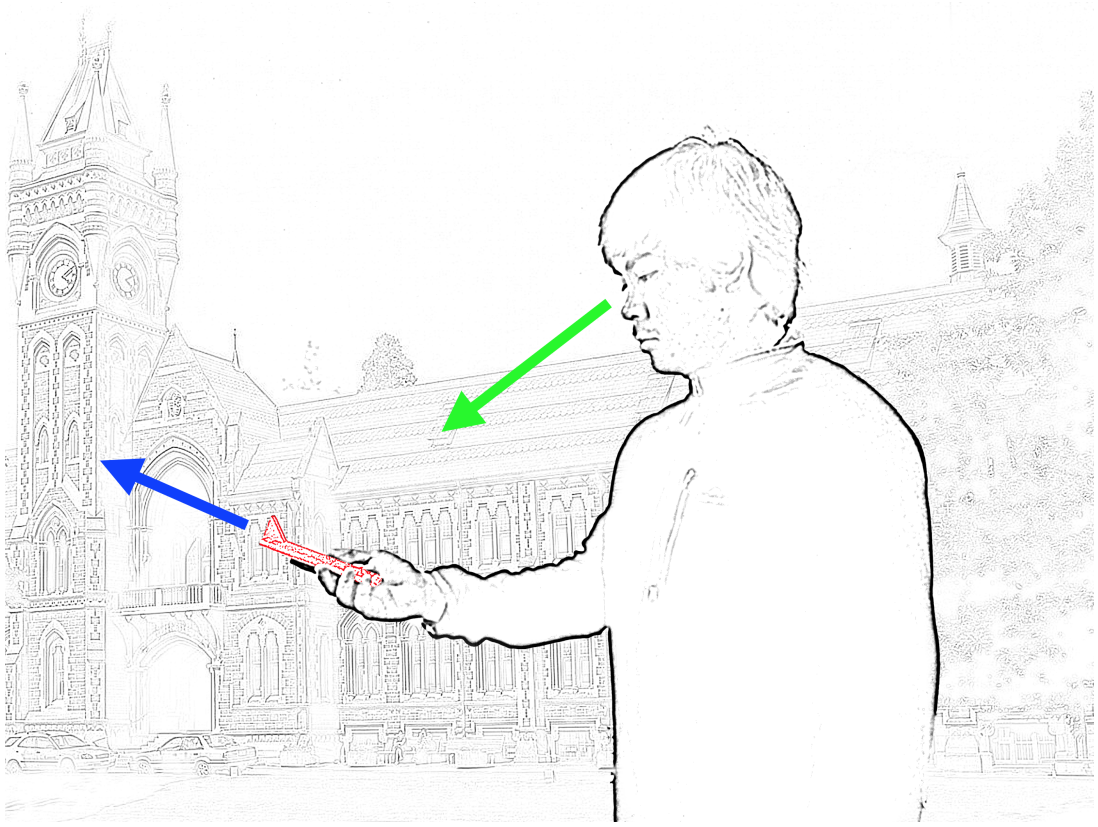

Figure 6.6: The ergonomics when using the new reflected AR browser interface. One-handed at waist level, with the user able to see the physical object, media browsing is done by camera aiming using the rendered cross-hair on the screen.

layout orientation. Both interfaces occupy the screen space with a copy of the physical world, by means of the camera stream. The AR browser shows a wider section of the physical world, as of the wider horizontal field of view of the back-facing camera in landscape mode. However, both interfaces display enough content to accurately aim towards objects of interest. As of the fact that the vision of the physical world is needed to utilize pointing, it can not get overloaded with the situated media content. Therefore, the amount of screen space to display the situated media content is limited. The media pointing interface overcomes the reduced media space issue, by minimizing the connection to the physical world. The dynamic, visual representation of the physical world is removed and replaced by a static representation of the physical object. This concept allows to occupy most of the screen space with the situated media content. Further, the user is forced to face the physical world for pointing. Nonetheless, the reduced connection of digital and physical world reduces the accuracy of the pointing interface. We hope to overcome this issue by reducing the LOD or by using the interface for closer distant pointing.
As a summery of this chapter, we introduce the following hypothesis for a future user

study:

1. The reflected AR browser has a higher usability compared to the traditional AR browser.
2. The media pointer has a higher usability than AR approaches.
3. The one-handed reflected AR browser and the media pointer are socially more accepted than the known AR browser.

7 Conclusion and Future Work

Contents

7.1 Conclusion

Augmented Reality (AR) browsers are an often used method to browse 3D situated media, their technology is well studied. In the last years, the amount of situated media rapidly increased, mostly due to the diversity of powerful mobile devices. The access to location information on mobile devices allows an easy enrichment of digital media with location information. We believe that digital media is soon not just available on the web through an internet browser, but can be browsed by accessing its location in 3D. Concluding, in this thesis, we present the implementation of accurate prototypes of the media pointing concept, using state-of-the-art tracking technologies known from AR browser. Even though the concept is known, current prototypes suffer from poor accuracy, affecting the concept of pointing [32][35]. Beside of the novelty of the prototype, achieving state-of-the-art accuracy, the prototype builds the foundation to study pointing as an alternative concept to situated media browsing. Additional, we introduce the reflected AR browser as a combination of the pointing interface and an AR browser.
Technically, the accuracy of any situated media browsing concept depends on the solution of the global pose problem. The global pose problem is the global location and north-centered orientation estimation used to detect situated media in the physical world.

In this work, we use an adapted version of the approach of Ventura et al. [41] to solve the global localization. We describe in detail how we modify the approach to be used

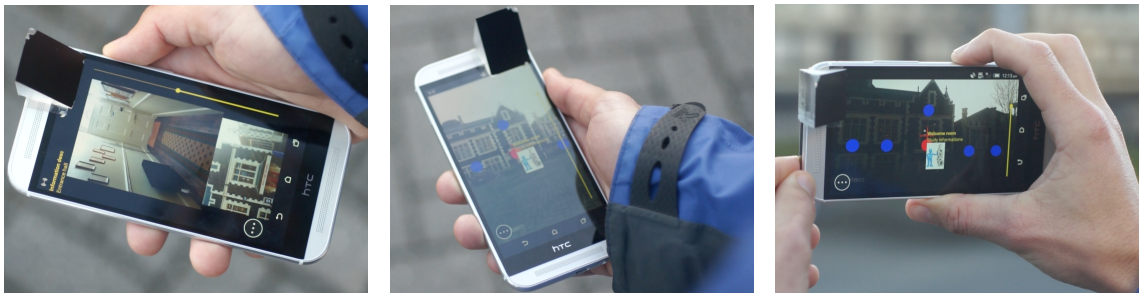

Figure 7.1: The interface prototypes, while testing them in front of the Allen Hall at the campus of the university of Otago. On the left the media pointer, the reflected AR browser in the middle and the AR browser on the right.

in our interface prototypes. The localization is done by sending image samples of the environment to a localizer server. The server hosts a point cloud representation of the environment. The point cloud is used to identify the exact global location of the image. For the reflected AR browser and the situated media pointer, we mount a prism onto the front facing camera. The prism bends the light entering the camera by 90 degree allowing to generate the images of the environment needed for the localization, while holding the phone like a remote controller. By evaluating the approach, we achieved sub-meter accuracy, outperforming the raw GPS localization by a magnitude of eight.
The north-centered orientation estimation is done by locally combining an accurate visual orientation tracker with the north-centered sensor orientation of the mobile device. The method is introduced by Schall et al. [34], however, we adapt the approach for pointing gestures, extending the original visual orientation tracker to a full pose tracker. The general idea is to calibrate the sensors on-the-fly with the accurate vision tracker. Our evaluation of the method results, in worst case, in an average north error of 4.15 degrees. Finally, we combine both methods to a solution of the global pose problem. The resulting global pose is similarly used in the media pointer, the reflected AR browser and the AR browser prototype, illustrated in figure 7.1. Summarizing, the used technologies allowed us to implement different, but comparable interfaces to browse situated media. The reflected AR browser is a novel interface, combining accurate camera aiming with the ergonomics of a pointing interface. The media pointer is the first of its kind, not entirely being based on internal or external sensors. We improved the accuracy of the raw sensors: first, by replacing the GPS sensor with a pure visual localization approach and second, by improving the orientation sensors with a visual tracker. The increased accuracy of the introduced media pointer allows to study pointing as an alternative situated media browsing concept. Additionally, the AR browser prototype, based on the presented technologies, is an existing situated media interface concept we would like to use as a reference.
Generally, we hope our prototypes can have an influence in the field of mobile human-

computer interaction, as we introduced novel methods to accurately interact with outdoor environments using pointing. Further, we belief accurate pointing to be an alternative method to explore unknown surroundings, also of commercial interest.

7.2 Future Work

Inaccuracies in the global localization technically affect the size of physical objects that can be annotated with digital media. By not detecting the global position accurately, pointing gestures can be erroneous by the magnitude of the offset to the real position. Concluding, bigger sized physical targets to compensate for the offset error are required. Similarly, the object size is affected by an inaccurate orientation estimation. To minimize the error of both, the location and the orientation, we decided to use two different available methods. The method of Ventura et. al [41] used for the localization, also provides a local, north-centered, vision based orientation tracking. Even though it is not designed for pointing, we plan to access this information to further reduce the orientation error. The additional visual north orientation can help to detect sensor errors and is may used as a fallback method. In a further step we plan to replace the current relative vision tracker with the north-centered vision tracker presented in the work of Ventura et al. However, the technical changes need to be evaluated before adapting the prototypes. Generally, it is unclear if the orientation tracking component of Ventura et al. performs well for pointing gestures.
Additional, to test the integration of a different visual tracking, we also plan to improve the raw sensor orientation by pre-filtering it before feeding it to the Kalman filter. Lawitzki[1] presents such a sensor fusion, combining a low-pass filtered magnetometer-accelerometer orientation with a high-pass filtered gyroscope orientation. Similar, as the integration of the visual tracker, an evaluation of the method is required before integration.
Beside the technical improvements, we could think of a different use case than browsing situated media. In future, we would like to extend our prototypes to browse situated services. Instead of querying information, we would like to get access to location based services or interact with the targeting physical object. In figure 7.2 we illustrate mock-ups of our vision of situated services. For example when pointing to our car, we would like to be able to lock and unlock it, or pointing to the park meter, we would like to display the interface to pay the parking fees. Basically, we hope to bring our prototypes one step closer to the "tricorder" known from the Star Trek movies. Our vision is a general-purpose device helping to scout and interact with unfamiliar areas.

Finally, we also plan to perform the user study mentioned in chapter 6 of this thesis. We are interested to verify or reject the introduced hypothesis. We want to find out if

[1] http://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutorial(26.08.2015)

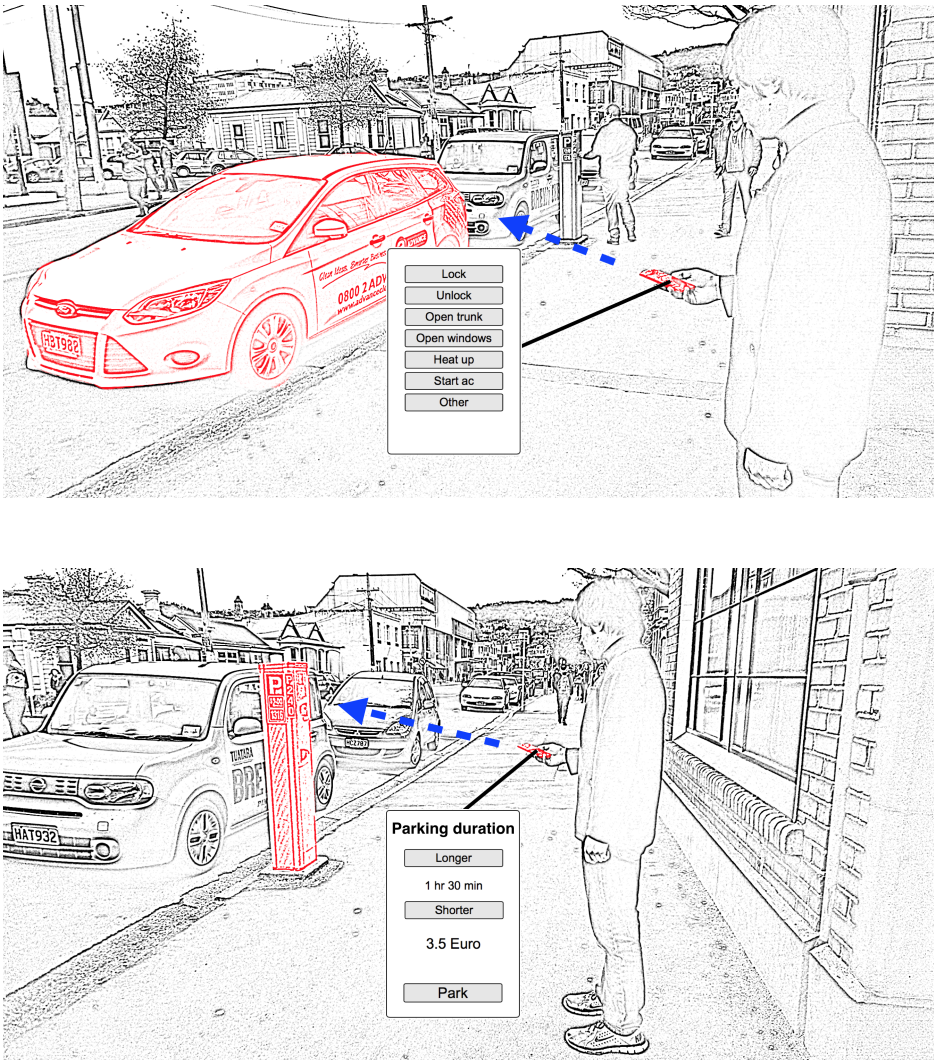

Figure 7.2: Illustrations of our vision of situated services. In future we plan to actively interact with the physical object annotated with a situated service. Examples are to interact with a car (top) or directly get access to a service by pointing at it (bottom). E.g. paying our parking fees by just pointing towards the park meter.

accurate media pointing in urban environments can be an alternative interface to AR browsing. As a result of the user study, we also hope to improve the interfaces of our prototypes. Especially, we hope to get feedback regarding the layout. We are interested in optimizing slider and button placement, as well as the layout of the media pointer.

Bibliography

[1] Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, pages 421–433. (page 9)

[2] Arth, C., C., P., J., V., and V., L. (2015). Global 6dof pose estimation from untextured 2d city models. (page 16, 20)

[3] Arth, C., Wagner, D., Klopschitz, M., Irschara, A., and Schmalstieg, D. (2009). Wide area localization on mobile phones. In *International Symposium on Mixed and Augmented Reality*, pages 73–82. (page 16)

[4] Azuma, R. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, pages 355–385. (page 2, 8)

[5] Bell, B., Feiner, S., and Hollerer, T. (2002). Information at a glance [augmented reality user interfaces]. *Computer Graphics and Applications*, pages 6–9. (page 11)

[6] Chen, D. M., Baatz, G., Koser, K., Tsai, S. S., Vedantham, R., Pylvanainen, T., Roimela, K., C., X., Bach, J., Pollefeys, M., Girod, B., and Grzeszczuk, R. (2011). City-scale landmark identification on mobile devices. In *Conference on Computer Vision and Pattern Recognition*, pages 737–744. (page 16)

[7] Chum, O. and Matas, J. (2005). Matching with prosac-progressive sample consensus. In *Computer Society Conference on Computer Vision and Pattern Recognition*, pages 220–226. (page 29)

[8] Dang, N. (2007). A survey and classification of 3d pointing techniques. In *International Conference on Research, Innovation and Vision*, pages 71–80. (page 4, 18, 21)

[9] Egenhofer, M. J. (1999). Spatial information appliances: A next generation of geographic information systems. *Technical Report, University of Maine*, pages 1–4. (page 9, 12, 57)

[10] Feiner, S., Macintyre, B., Höllerer, T., and York, N. (1997). A touring machine : Prototyping 3d mobile augmented reality systems for exploring the urban environment. *Personal and Ubiquitous Computing*, pages 208 –217. (page 2, 9, 10, 14)

[11] Fischler, A. and Bolles, C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pages 381–395. (page 29)

[12] Grubert, J., Langlotz, T., and Grasset, R. (2011). Augmented reality browser survey. *Technical Report, Graz University of Technology*, pages 1–30. (page 3, 4, 10, 20, 55, 56, 59)

[13] Guven, S. (2006). *Authoring and Presenting Situated Media in Augmented and Virtual Reality*. PhD thesis, New York, NY, USA. (page 1, 8)

[14] Hansen, A., Bouvin, N. O., Christensen, G., and Gronbak, K. (2004). Integrating the web and the world: Contextual trails on the move. In *Proceedings of the Conference on Hypertext and Hypermedia*, pages 1–7. (page 9, 14)

[15] Hansen, F. A. (2006). Ubiquitous annotation systems: technologies and challenges. *Proceedings of the Conference on Hypertext and Hypermedia*, pages 121–132. (page 7, 8, 9, 11, 17)

[16] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press. (page 27)

[17] ISO (2000). Qr code. ISO ISO/IEC 18004, International Organization for Standardization. (page 9, 13)

[18] Kähäri, M. and Murphy, D. J. (2006). MARA - Sensor Based Augmented Reality System for Mobile Imaging Device. *International Symposium on Mixed and Augmented Reality*, pages 180–180. (page 11)

[19] Kato, H. and Billinghurst, M. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system. *International Workshop on Augmented Reality*, pages 1–10. (page 9, 13, 15)

[20] Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone. In *International Symposium on Mixed and Augmented Reality*, pages 83–86. (page 15, 25)

[21] Langlotz, T. (2006). *AR 2.0 : Social Media in Mobile Augmented Reality*. PhD thesis, Graz, Austria. (page 3, 9, 11)

[22] Langlotz, T., Nguyen, T., Schmalstieg, D., and R., G. (2014). Next-Generation Augmented Reality Browsers: Rich, Seamless, and Adaptive. *Communications of the ACM*, pages 155–169. (page 3)

[23] Langlotz, T., Regenbrecht, H., Zollmann, S., and Schmalstieg, D. (2013). Audio stickies: Visually-guided spatial audio annotations on a mobile augmented reality platform. In *Proceedings of the Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, pages 545–554. (page 9, 11)

[24] Lee, R., Kitayama, D., Kwon, Y., and Sumiya, K. (2009). Interoperable augmented web browsing for exploring virtual media in real space. In *Proceedings of the International Workshop on Location and the Web*, pages 1–7. (page 11)

[25] Liestol, G. (2011). Learning through situated simulations: Exploring mobile augmented reality. *Research Bulletin*, pages 1–14. (page 9, 12)

[26] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, pages 91–110. (page 26)

[27] Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer series in operations research and financial engineering. Springer. (page 27)

[28] Pradhan, S., Brignone, C., Cui, J. H., McReynolds, a., and Smith, M. T. (2001). Websigns: Hyperlinking physical locations to the Web. *Computer*, pages 42–48. (page 9, 12, 14)

[29] Reitmayr, G. and Drummond, T. W. (2007a). Going out: Robust model-based tracking for outdoor augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*. (page 16)

[30] Reitmayr, G. and Drummond, T. W. (2007b). Initialisation for visual tracking in urban environments. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*. (page 16, 20, 25)

[31] Rekimoto, J., Ayatsuka, Y., and Hayashi, K. (1998). Augment-able reality: situated communication through physical and digital spaces. *Digest of Papers. Second International Symposium on Wearable Computers*, pages 1–8. (page 9, 10, 14)

[32] Robinson, S., Eslambolchilar, P., and Jones, M. (2009). Sweep-shake: finding digital resources in physical environments. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–12. (page 4, 9, 12, 20, 40, 57, 63)

[33] Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-Time Video Annotations for Augmented Reality. *Advances in Visual Computing*, pages 294–302. (page 11)

[34] Schall, G., Mulloni, A., and Reitmayr, G. (2010). North-centred orientation tracking on mobile phones. *International Symposium on Mixed and Augmented Reality*, pages 267–268. (page 17, 39, 40, 41, 43, 44, 46, 64)

[35] Simon, R. and Fröhlich, P. (2008). GeoPointing: evaluating the performance of an orientation aware location based service under real-world conditions. *Journal of Location Based Services*, pages 24–40. (page 4, 9, 12, 13, 20, 43, 57, 63)

[36] Simon, R., Fröhlich, P., Obernberger, G., and Wittowetz, E. (2007). The Point to Discover GeoWand. *International Conference on Ubiquitous Computing*, pages 1–4. (page 9, 12, 40, 57)

[37] Sinclair, P., Martinez, K., Millard, D., and Weal, M. J. (2002). Links in the Palm of your Hand: Tangible Hypermedia using Augmented Reality. In *Proceedings of the Conference on Hypertext and Hypermedia*, pages 127–136. (page 9, 14)

[38] Snavely, N., Seitz, M., and Szeliski, R. (2006). Photo tourism: Exploring Photo Collections in 3D. *Transactions on Graphics*, pages 835–846. (page 26)

[39] Spohrer, J. C. (1999). Information in places. *IBM Systems Journal*, pages 602–628. (page 8, 9, 14)

[40] Ventura, J. and Höllerer, T. (2011). Outdoor mobile localization from panoramic imagery. pages 247–248. (page 16, 25)

[41] Ventura, J. and Höllerer, T. (2012). Wide-area scene mapping for mobile visual tracking. *International Symposium on Mixed and Augmented Reality*, pages 3–12. (page 16, 17, 20, 25, 26, 28, 29, 37, 39, 46, 52, 63, 65)

[42] Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. (2010). Real-time panoramic mapping and tracking on mobile phones. *Transactions on Virtual Reality*, pages 211 –218. (page 15, 17)

[43] Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2008). Pose tracking from natural features on mobile phones. *International Symposium on Mixed and Augmented Reality*, pages 125–134. (page 9, 14, 15, 40, 41)

[44] Want, R., Schilit, B., Adams, N., Gold, R., Petersen, K., Goldberg, D., Ellis, J., and Weiser, M. (1995). An overview of the parctab ubiquitous computing experiment. *Personal Communications*, pages 28–43. (page 9)

[45] Wither, J., Tsai, Y. T., and Azuma, R. (2011). Indirect augmented reality. *Computers and Graphics*, pages 810–822. (page 2, 9, 11)

[46] Zandbergen, P. A. and Barbeau, S. J. (2011). Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation*, pages 381–399. (page 14, 25)

[47] Zhang, L. and Coulton, P. (2009). A mobile geo-wand enabling gesture based POI search an user generated directional POI photography. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, pages 392–395. (page 9, 13, 57)

[48] Zhao, Y., Chakraborty, A., Hong, K. W., Kakaraddi, S., and St. Amant, R. (2012). Pointing at responsive objects outdoors. *Proceedings of the International Conference on Intelligent User Interfaces*, pages 281 –284. (page 13, 20, 58, 59)