Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

AR 2.0: Social Media in Mobile Augmented Reality

Tobias Langlotz

Graz, Austria, June 2013

Thesis supervisors

Prof. Dr. Dieter Schmalstieg

Institute for Computer Graphics and Vision

Assoc. Prof. Dr. Holger Regenbrecht

Department of Information Science at Otago University in Dunedin

TU Graz
Graz University of Technology
Senat

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

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 entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

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.

…………………………… ………………………………………………..
date (signature)

To Stöppi

Abstract

The goal of this thesis is to design, implement and evaluate novel techniques for integrating social media into handheld Augmented Reality (AR) applications. Augmenting the physical environment with information is an idea prospected for over twenty years within the AR research community. The availability of mobile phones as an affordable platform for mobile AR applications lead for the first time to a wide awareness of the public. For instance, AR browsers, applications that allow users to browse information registered to places or objects in the real world, are now installed on several millions of phones. Despite the fact that AR browsers have arrived in the mass market, we argue that they are solely used for browsing information while at the same time lacking in amount and variety of information that can be accessed. This is caused by the current authoring tools aiming at professionals and do not permit a spontaneous creation of content in place, while also focussing on rather simple content forms.

In the first part of this thesis, we want to shed light on the adoption of AR browsers to the public, by presenting results of two conducted studies. The results are later used for developing the concept of Augmented Reality 2.0 as the next evolution of Augmented Reality. Augmented Realities 2.0 utmost concern is to not only see users of AR applications as consumers, but also as a producers of content ("Prosumer"). In preparations for our implemented prototypes, we present fundamental technologies such as a precise tracking for handheld devices and an AR framework that can be used to implement AR 2.0 applications. In the second part of this thesis, we present novel prototypes integrating social media in various forms into mobile AR applications. Lastly, an assessment of these prototypes is done based on user feedback and observations made through user studies.

This thesis extends the current understanding on the adoption of AR browsers and gives requirements helping developers to make informed design decision when building next generation AR browsers relying on user-generated content.

Kurzfassung

Das Ziel dieser Arbeit ist das Design, die Implementierung, und die Evaluierung von neuen Techniken, welche die Integration von Sozialen Medien in mobilen Augmented Reality Anwendungen erlauben. Die Überlagerung der physikalischen Umgebung mit digitalen Informationen ist eine Idee, die seit mehr als 20 Jahren in dem Forschungsbereich Augmented Reality (AR) verfolgt wird. Die Verfügbarkeit von Smartphones als kostengünstige Plattform für mobile AR Anwendungen bewirkt dabei erstmalig eine starke Verbreitung von AR Anwendungen auch für Endnutzer. Speziell Augmented Reality Browser - mobile Browser, welche digitale Informationen anzeigen, die an physikalischen Objekten oder geographischen Positionen verankert sind - haben dabei mit ca. 50 Millionen installierten Anwendungen den grössten Marktanteil. Trotz dieses Erfolges argumentieren wir, dass AR Browser einzig zum konsumieren von Informationen verwendet werden, wobei wir feststellen, dass die verfügbaren Informationen nicht sehr zahlreich, und nur in kleiner Formvielfalt vorhanden sind. Das hat hauptsächlich seine Ursache in den aktuellen Autorenwerkzeugen, welche auf professionelle Anwender zielen und keine spontane Erstellung von Inhalten für AR erlauben. Darüber hinaus erlauben sie meist nur die Verwendung von einfachen Medienformen.

Im ersten Teil dieser Dissertation werden wir unsere Studien zur Akzeptanz von AR Browsern durch die Öffentlichkeit präsentieren. Die Ergebnisse dieser Studien halfen bei der Entwicklung unseres Konzeptes von Augmented Reality 2.0 (AR 2.0) als nächsten Evolutionsschritt von AR, welchen wir einführen möchten. Das Kernanliegen von AR 2.0 ist, Nutzer von AR Anwendungen nicht nur als Konsumenten zu sehen, sondern auch zu Produzenten von Inhalten zu machen ("Prosument"). Im zweiten Teil dieser Arbeit werden wir die technischen Grundlagen für AR 2.0 zusammen mit entwickelten Prototypen vorstellen, welche Aspekte unserer Konzeptes von Augmented Reality 2.0 demonstrieren. Abschliessend werden wir die entwickelten Prototypen aufgrund von Nutzerstudien und Beobachtungen diskutieren.

Diese Dissertation erweitert das aktuelle Verständnis der Akzeptanz von AR Browsern und gibt Entwicklern definierte Anforderungen, welchen ihnen helfen Designentscheidungen für AR Browser der nächsten Generation zu treffen um nutzer-generierte Inhalte zu benutzen.

Acknowledgments

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Augmented Reality

Today's everyday life is heavily affected by an increasing amount of digital information. While a few years ago desktop computers were the given choice for accessing information, nowadays, more and more information is consumed using mobile devices such as tablets or smartphones. This evolution in the use of devices coincided with the evolution of interfaces that can be used to access digital information. Traditional interfaces are not well adapted to the new form factor, additionally they do not take advantage of the new opportunities created by new computing devices yet.

Augmented Reality (AR) is one category of human computer interfaces that has the potential to change the way digital information is accessed and consumed. The core idea of Augmented Reality is to augment or overlay digital information onto the real world so that it is displayed right at the object or place it relates to. Based on this idea it is possible to create interfaces that allow users to experience complex information in a natural way and at the same time avoid the overhead of handling complex devices. As the digital information is referenced to places or objects in the environment, the user can take advantage of a rather natural way to retrieve information: looking at and visually browsing his environment.

The first prototypes which implemented an Augmented Reality interface were realized in the 1960s by Sutherland [137]. As part of this long this long evolution, there is not one but several definitions that characterize Augmented Reality interfaces. However, throughout the AR research community, two definitions of Augmented Reality are most commonly used to outline and describe the research field of AR.

While Caudell and Mizell coined the term Augmented Reality [17], the widest used definition of Augmented Reality was provided by Azuma [5]. He described Augmented Reality by a set of characteristics that should be fulfilled by applications when implementing an AR interface. Azuma says Augmented Reality applications

- combine the real and virtual,
- are interactive in real time,
- register the information in the real world in 3D.

Contrary to Azuma's definition, the second commonly used definition by Milgram [87] is less specific in terms of requirements as it describes Augmented Reality as one area in the space between Reality and Virtual Reality. Milgram used this Mixed Reality Continuum to outline AR as a medium where Reality is augmented with virtual information. Unlike Augmented Virtuality, Milgram notes that in Augmented Reality the reality is still the dominating medium.

Most of the early AR application prototypes were still experimental. The hardware was custom-made or expensive and the software stack was in an early stage, usually only developed for research. This is mostly due to the unique needs Augmented Reality has in terms of hardware, requiring a camera to capture the real world, tracking sensors (e.g., GPS, inertia sensors, magnetic trackers or a camera just to name a few), a display device (i.e., Head Mounted Display, integrated LCD displays, projecting device) and finally a processing unit with enough performance to process all data while also being capable to render and visually combine the real world information with the augmentations (often requiring a capable GPU). Consequently, the first applications were for research or industrial applications and targeted professional users only.

However, due to the success of smartphones, an affordable platform for Augmented Reality applications became available. The combination of hardware features needed for AR together with an efficient, yet powerful CPU, packed into a small, and even more important, affordable device opened the way for AR in the consumer market [126].

Nowadays, there are various use cases for AR within an increasing number of application domains. AR has been proven to be useful in the professional domain, including, but not limited to medical applications, applications for education, applications supporting assembly or inspection tasks and applications for digital design and architecture. Beside applications for professional use, AR also made its way into the consumer domain. There, Augmented Reality primarily takes the form of edutainment systems, AR games allowing novel forms of interaction, digital advertisement, and finally as a personal information system. The last category is mostly occupied by so called AR browsers, AR applications that augment the physical environment with digital information stored in a remote database and associated to geographical locations or real objects (see Figure 1.1).

With more than forty million downloads of AR browser applications since 2008, and some even pre-installed on smartphones from leading brands, AR browsers such as Wik-

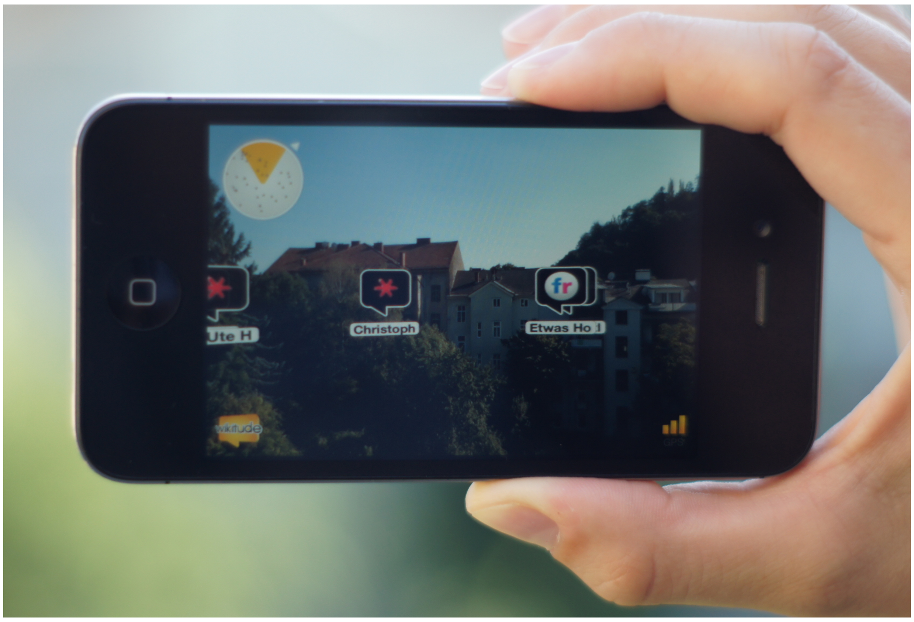


Figure 1.1: Wikitude, the first commercial AR browser application as an example of an AR-based personal information system running on a smartphone.

itude[1] and Layar[2] have become new players in the mobile application landscape and are by far the most successful AR applications in terms of consumer awareness.

The current generation of AR browsers supports mainly graphical augmentation based on location and viewpoint information delivered by the integrated sensors such as GPS, gyroscopes or compasses, while the camera is only used to display the information from the physical world. However, more recently some of these applications have also integrated vision-based approaches to augment physical objects (e.g., magazines, CD covers). The fast development of AR browsers can from our point of view be explained by four factors: (1) large adoption of smartphones that are capable to run AR applications, (2) establishment of mobile applications markets such as the App Store or Google Play, (3) recent media interest together with connected investment opportunities, and (4) creation of more accessible mobile APIs, leveraging the development of AR applications.

Despite the success in terms of consumer awareness of AR browsers that is reflected in the number of downloads, the general adoption is unclear and it seems digital maps and location-based services seem to be better adopted by users. This might be due to the fact that more existing applications making use of digital maps to geo-reference information. Furthermore, the interface of digital maps shares common ideas with physical maps. Consequently, people are more used to map-based interfaces. Apart from this, it is still widely unclear if AR browsers are gadgets that are only tried out a couple of times, or if AR browsers become important as a frequently used tool.

[1] www.wikitude.com

[2] www.layar.com

Within this thesis, we argue that AR browsers are not as well received as digital maps mainly due to the low quantity and quality of information that can be accessed with AR browsers. The accessible content is still widely created by professional content producers and it seems that the content producers can't provide the quantity and quality of information that is needed to convince users of the usefulness and applicability of AR browsers. In fact, the currently accessible content is very sparse, mostly of poor quality, and not very rich in terms of content forms. Furthermore, the content and information are not presented in a way that nurtures the unique possibilities offered by AR, therefore, adding no value over traditional ways of browsing and experiencing the information. The resulting questions are why this gap in quantity and quality exists and how the authoring process can be modified to target these problems while still utilizing the advantages AR has to offer.

1.2 Authoring for Augmented Reality

Augmented Reality as an interface to digital information relies heavily on visual data such as 3-dimensional renderings, graphical representations of textual content and images. Within Augmented Reality other representational forms of information (e.g., audio-based Augmented Reality[9]) have been explored, but are usually living a niche existence. Consequently, the existing authoring solutions for Augmented Reality target mostly the process of authoring digital information that is suited for visual representation.

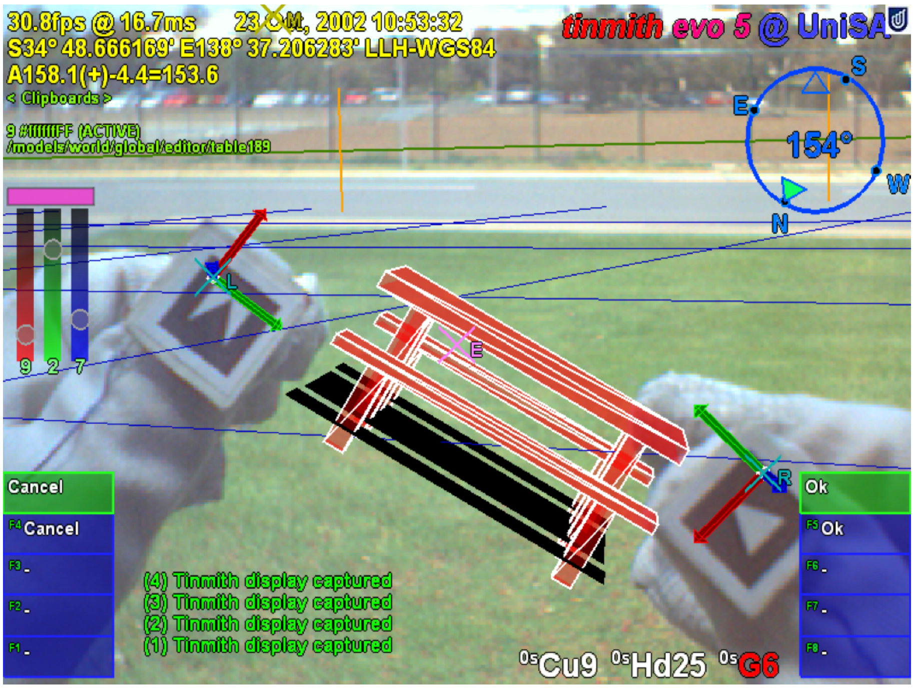
Despite the type of information and its representation, the authoring process in Augmented Reality can be separated into two steps. The first step is the creation of the content itself, while the second step is the arrangement of the content in space including the description of interactions possible within the created scenario [41].

Over the years, various authoring systems were created that usually originated either in the professional domain or in research. These systems often made use of existing authoring or modelling solutions together with their modelling pipeline used for the creation of graphical content. For Augmented Reality, these authoring solutions were extended by adding a step in their pipeline allowing the arrangement of the created content with respect to the real world or artificial markers. The Designer's Augmented Reality Toolkit (DART) is one of the best known works in this field [77]. It implements a plugin for the existing Macromedia Director[3] authoring solution and extends the functionality for Augmented Reality by adding AR authoring tools that allow it to specify the relationship between the physical and the virtual world. Until now, similar AR plugins have been presented for the most common modelling and authoring tools ranging from Adobe Flash to Google SketchUp[4] and Autodesk 3D Studio Max. All of these tools share the same common goal of supporting professionals within their preferred environment by maintaining their known workflow and a similar content creation pipeline when authoring content for Augmented Reality.

Other existing authoring works originated in research and explore approaches detached from existing authoring or modelling solutions. These approaches combine custom (and

[3] http://www.adobe.com/de/products/director

[4] http://www.inglobetechnologies.com



(4) Tinmith display captured
(3) Tinmith display captured
(2) Tinmith display captured
(1) Tinmith display captured

Figure 1.2: CAD inspired modelling in an outdoor AR environment using Tinmith by Piekarski et al. [104].

usually expensive) hardware with custom software solutions. An example for work in this domain is the Tinmith project for Computer-Aided Design (CAD), which inspired modelling in an outdoor AR environment [102] [104] (see Figure 1.2). There are other existing works such as the authoring of existing physical 3D models in AR using the Battlefield Augmented Reality System (BARS) [131] or the work by Wither et al. in the field of creating and placing annotations in AR using a head-worn laser range finder [156].

Despite the existence of the mentioned authoring approaches for Augmented Reality, none of these approaches seems to be the appropriate authoring solution for AR browsers or similar end-user focused mobile AR applications. Such approaches are only accessible to a small group of professional users due to both the price of the hardware and software, and the time needed to comfortably work with them. This results in an unbalanced and poorly scaling authoring process, where a small number of content producers has to create content as well as keep existing content up to date for a large and demanding user-base.

There are a few existing authoring tools that try to lower the requirements in terms of hardware and software, as well as reducing the experience required to author content especially for AR browsers. These authoring tools are mostly provided by AR browser companies, which specifically aim to support the authoring process for their own AR browsers. These tools assume the content (e.g., 3D models, pictures) to be created beforehand and outside of their authoring tools. Given this existing content, they usually employ web-based front ends relying on desktop interfaces that allow users to geo-reference the content by linking it to a GPS position (commonly via a map-based interface).

Some AR browsers such as Layar nowadays also support vision-based tracking of planar targets and created specific solutions (e.g., Layar Creator[5]) to support the different authoring process required to create augmentations for magazines and other types of printed publications. Despite the fact that they do not geo-reference the digital content, they employ a similar desktop interface to arrange the content on the printed marker, limiting a spontaneous authoring process. They also still assume the content to be created beforehand.

Independent of the type of augmentations, the authored content, either geo-referenced or referenced to an image, is saved in the custom format of the supported AR browser and is later uploaded to the back-end database of the corresponding AR browser.

The business models of existing companies in the field cause them to act as gatekeepers for the available content. This status is manifested by using proprietary formats that differ between the various AR browsers in contrast to desktop Web browsers that all access the same content encoded in standardized HTML pages[6]. From a content producer point of view, this requires users to author the content differently by creating specific packages targeting individual AR browsers.

Overall, existing authoring tools for creating content for AR browsers have lowered the use requirements compared to the more advanced AR authoring solutions. They do not use expensive hardware as they run on standard desktop computers, nor do they necessarily require expensive software as they are usually free. Although the learning curve for using these authoring tools is less demanding, as with more advanced tools the they still mainly attract professionals, rarely permitting creation of content by inexperienced users. The content is usually created beforehand and outside of the authoring tool as the existing authoring tools for AR browsers do not support the creation of content within the application. Instead, the content is expected to be available for being loaded into the authoring tool. The authoring tools focus mainly on the arrangement of the content. The original creation of the content, especially in cases where high quality content is required, still demands a lot of time and experience that usually can only be assumed within the professional domain.

Consequently, the majority of the available content consists of simple form and representation: mainly textual annotations and 2D images. In the unlikely case more complex content forms, such as 3D models, are used, they are usually a selection out of the few predefined 3D models that come as examples with the authoring tools.

While future generations of AR authoring tools might further reduce the requirements for content producers, other aspects of AR authoring are widely neglected in commercial solutions but are harder to solve. First, the content is still created using desktop interfaces making spontaneous authoring carried out in-situ difficult to realize. Secondly, the current interfaces for placing content are often derived from map-based interfaces. However, it is unlikely that these kinds of interfaces are a good choice for future versions of AR browsers using precise vision-based tracking, as they require an accurate placement of the content into the environment. Finally, social media, a term ”usually applied to describe the various forms of media content that are publicly available and created by end-users” [52], is not integrated or supported in current generation AR browsers. All these mentioned

[5] www.layar.com/creator

[6] www.w3.org

drawbacks hinder many anticipated use-scenarios of AR browsers such as social networks incorporating AR content. More importantly, they make a vast increase in the amount of AR browser-accessible content unlikely to happen.

Allowing users to create content within an AR application with an easy-to-use interface could allow content to be spontaneously created in-situ, which would help to exploit new application areas. Furthermore, it would open AR browsers to support social media, consequently turning users from consumers into *prosumers* [141] [115], users that consume and produce content. This shift would result in an increased content production. Finally, creating the content within the AR application would also allow an accurate placement using Augmented Reality as interface.

1.3 Hypotheses and Postulations

This thesis investigates content authoring solutions targeting mobile AR applications and in particular AR browsers by presenting novel solutions for authoring social media within mobile AR. As part of the investigations we raise the following hypotheses that are discussed in the remainder of this thesis:

- H1: The gap in content quantity and quality marks a major bottleneck in current mobile AR systems.

- H2: User-generated content in the form of social media can be consumed and produced on handheld AR system achieving acceptable usability.

- H3: There is a wide acceptance towards user-generated AR content beyond textual annotations and tagging, namely audio, video, graphical content and geometrical models.

- H4: System, user and content scalability in user generated, mobile AR can be achieved with appropriate framework architectures and implementations.

- H5: Reliable localization, registration and tracking are key to acceptance and usability of a mobile AR system.

Based on these hypotheses, we put the following postulations for future authoring solutions targeting end user mobile AR applications, but especially AR browsers:

- P1: New, domain-specific authoring solutions supporting laypersons as content producers are essential to assure that the amount of content grows with the user base.

- P2: These authoring solutions should enable people to create and place various types of content in AR, including textual and image augmentations, audio and video content and three-dimensional content.

- P3: The authoring process should support spontaneous authoring that is performed in-situ and does not require a prepared environment for the application to work.

- P4: The authoring application resides within a stable and robust application infrastructure consisting of precise tracking algorithms running on the client-side and a server bound back-end-infrastructure hosting the content and implementing a sharing functionality.

These postulations represent goals for the research conducted in the thesis but require a more detailed explanation to understand the scope of this thesis. This thesis aims to investigate the content gap in AR browsers, specifically the lack of available and useful content accessible through AR browsers. We present a conceptual model for increasing the amount of available content by involving inexperienced users in the content authoring process applying ideas of crowdsourcing [24]. Finally, we present several prototypes that investigate different aspects of authoring content for AR browsers. These prototypes are evaluated using best standards in technical evaluation as well as user studies to prove their applicability and usefulness.

Throughout this thesis we restrict ourselves to the use of accessible off-the-shelf hardware (mostly smartphones) to demonstrate our research. We apply this restriction to demonstrate the capabilities of current technology, but also to show the possibilities given current software and hardware. While in the future there might be different hardware form factors for AR browsers, such as the recently announced Google Goggles (also known as Google Project Glass[7]), their true capability and usability is only of speculative character. Consequently, it is unknown how future devices are received by the public. We also do not aim to extend any specific commercial AR browser for demonstrating our novel approaches but used our own AR browser prototype that was developed within this thesis. Our AR browser shares most aspects of existing AR browsers, as well as some new features such as improved tracking that are discussed within this thesis.

1.4 Contributions

This thesis contributes to the field of Augmented Reality, Human Computer Interaction and Pervasive Computing by investigating the use of social media within end-user mobile AR applications especially for AR browsers. By doing this, we also contribute solutions to the general issue of content creation in mobile Augmented Reality.

In particular, we contribute a concept and requirements for scalable content authoring that has the potential to increase the amount of accessible content for AR applications through user participation by means of explicit crowdsourcing [24]. Finally, we provide several novel application prototypes demonstrating the interface for creating media in AR and its usability. These prototypes cover content types ranging from textual annotations and images to audio and video content and the authoring of three-dimensional graphical content. The prototypes were evaluated and studied in the field with novice users to obtain empirical results to address the research questions involving usability and applicability of the presented approaches. The following overview outlines the main contributions:

- **A study investigating current generation AR browsers** showing how users use and employ current generation AR browsers. This study reveals flaws of AR

[7] https://plus.google.com/111626127367496192147/

browsers such as content availability and quality as some of the main limitation of the current generation of AR browsers.

- **A novel concept coined Augmented Reality 2.0 (AR 2.0) for integrating social media into AR**. We present our concept together with a requirement analysis for applications following the idea of Augmented Reality 2.0.

- **A stable and precise approach for tracking in large and unprepared environments** using panoramic maps of the environment. This approach serves as a technical foundation for Augmented Reality 2.0 applications.

- **A system supporting in-situ authoring of textual annotations and images** on smartphones.

- **Audio Stickies as a technique for creating and using audio annotations** within outdoor Augmented Reality.

- A technique demonstrating **situated compositing of video content in mobile AR**.

- **A system supporting the in-situ sketching of three-dimensional content** on mobile devices.

1.4.1 Study on current generation AR browser applications

Since its first appearance in 2008 (Wikitude[8] demonstrated on the Google G1 smartphone) AR browsers have become commercially successful with over 10 different commercial providers of AR browser applications and over 40 Million downloads in the main mobile app stores. Given the total amount of 20 Billion downloads of mobile applications from the app stores, AR browsers are only a small segment within the mobile app market. However, they are by far the most downloaded and used application-type in Augmented Reality.

Despite this success, the real-world usage of Augmented Reality browsers is still widely unexplored. Assumptions regarding the applicability and usefulness of AR browsers have remained largely untested. Furthermore, there are only few works investigating open issues of the current generation of AR browsers. We therefore conducted one of the first studies that investigated the real world usage and identified flaws within commercial AR browsers. We conducted the study in two steps. First, we downloaded accessible feedback of users in the mobile app stores. This includes several thousand ratings as well as the comments of thousands of users. Then we statistically analysed these ratings. We further clustered and analysed the user comments. This data gave us insights into end users view on AR browsers. Second, we did an online survey aimed at users of AR browsers. We used this tool to get additional feedback on topics that were not covered by the market analysis but also to get more details on specific topics.

The results of the study are presented in our technical report [34] and show that while the usage of Augmented Reality browsers is often driven by their novelty factor,

[8] http://www.wikitude.com

a substantial number of long-term users exist. The analysis of both quantitative and qualitative data showed that the low quantity and quality of existing digital information that can be accessed with AR browsers is a major reason for quitting the use of AR browsers. These results support our postulation of the need for new approaches to content authoring and are consequently one of the main motivations of this thesis. This work is discussed in more detail in Chapter 3 of this thesis.

1.4.2 AR 2.0 as a concept for integrating social media into Augmented Reality

The availability of a cheap, yet versatile platform for AR in the form of smartphones made it easy for developers to deploy AR applications to end users. AR browsers particularly benefited from this advancement in the mobile market. Despite the advances in technology and app distribution for end users, the authoring tools for AR are still aimed at professionals, limiting a vast expansion of content developed for AR applications. AR browsers are heavily affected by this situation commonly called the content gap. This is mostly due to their large scale in terms of working environment and to their massive user base.

With Augmented Reality 2.0 (AR 2.0) we present a conceptual model that overcomes these current limitations by applying concepts and techniques from the recent evolution of the Web into Web 2.0. We think that one of the main aspects of Web 2.0 is social networking technology combined with crowdsourcing. The development of social networks and platforms such as Facebook[9], Wikipedia[10] and YouTube[11] turned content consumers into content creators and allowed end users to participate in the content creation process. Within AR 2.0 we want to apply this large-scale infrastructure for collaboratively producing and distributing AR content. We think combining widely used mobile hardware capable of running AR applications with the concepts from Web 2.0 will allow for the development of a new type of AR platform that can be used on a global scale. At the same time, this combination will overcome current problems such as the content gap, by exploiting user generated content. The concept of Augmented Reality 2.0 was published by Schmalstieg et al. [125] and is presented in Chapter 3 of this thesis.

1.4.3 Tracking in large and unprepared environments using a panoramic map of the environment

Most outdoor tracking systems rely on inertial sensors to improve robustness. Even though some modern smartphones integrate a linear accelerometer, it is of little help in typical AR scenarios since it only delivers translational motion. We present a natural feature mapping and tracking method, which is sufficiently efficient and robust to allow three degrees of freedom tracking in outdoor scenarios on mobile phones. Assuming pure rotational movements, the method creates a panoramic map from the live camera stream. The conceptual approach is very similar to traditional simultaneous localization and mapping (SLAM). For each video frame, the camera is first registered based on features in the map. In a second step, the map is then extended with new features from viewing directions that have

[9] http://www.facebook.com

[10] http://www.wikipedia.org

[11] http://www.youtube.com

not been observed before. The contribution made lies in the description of a new method that can create and track panoramic maps in real time (30Hz) on a mobile phone. This work was presented by Wagner et al. [152] and discussed in detail in Chapter 4.

1.4.4 In-situ authoring of textual annotations and images on smartphones

Textual annotations and images are the most common content used within AR browser applications. Despite the simplicity of the content there are no existing solutions that allow creating and placing annotations within the AR browser. In our work on authoring textual annotations and images, we present a novel approach for creating and exploring annotations in place using mobile phones. The system can be used in large-scale indoor and outdoor scenarios.

Unlike the current generation of commercial AR browsers, we offer an accurate mapping of the annotations to physical objects that works beyond the accuracy of the integrated sensors. The system uses a drift-free orientation tracking based on panoramic images, which can be initialized using data from a GPS sensor. Given the current position and view direction, we show how annotations can be accurately mapped to the correct objects. For the re-detection, specifically after temporal variations, we first compute a panoramic image with extended dynamic range (which can better represent varying illumination conditions). The panorama is then searched for known anchor points, while orientation tracking continues uninterrupted. We then use information from an internal orientation sensor to prime an active search scheme for the anchor points, which improves matching results. Finally, global consistency is enhanced by statistical estimation of a global rotation that minimizes the overall position error of anchor points when transforming them from the source panorama in which they were created, to the current view represented by a new panorama. Possible applications range from Augmented Reality browsers to pedestrian navigation.

We investigate the potential of in-situ creation of AR annotation and images directly on the mobile phone and within the AR browser application, while previous authoring tools were mostly bound to desktop computers, or could operate only at the accuracy of the employed mobile sensors.

Our approach allows creating annotations in place and storing them in a self-descriptive way on a server, in order to allow a later re-identification. We use GPS information for efficient indexing, but identify the label positions using template matching against the panoramic map. This approach yields accurate and robust registration of annotations with the environment.

We tested our system using an AR campus guide as an example application and provide detailed results for our approach using an off-the-shelf smartphone. Results show that the re-detection rate is improved by a factor of 2 compared to previous work and reaches almost 90% for a wide variety of test cases. This research activities resulted in several publications [152][68][66] and we give more details on these works in Chapter 5 of this thesis.

1.4.5 Audio Stickies: In-situ authoring of audio annotations within outdoor Augmented Reality

Audio functionality is probably one of the most used functionalities on smartphone devices. In contrast, the use of audio as an additional information source for AR browsers has so far been widely ignored. Our research on Audio Stickies investigates the integration of spatially aligned user-generated audio annotations and visual augmentations into a single mobile AR system. Audio Stickies allow users to create and share precisely placed spatial audio annotations within a visual AR system. Our research demonstrates that the use of audio annotations that are positioned and oriented in augmented space successfully provides an additional, novel and enhanced mobile user experience. We also demonstrate the applicability and limits of Audio Stickies in noisy outdoor environments. The results of this work were submitted to [69] and are presented in Chapter 6 of this thesis.

1.4.6 Situated compositing of video content in mobile AR

Video capabilities are one of the essential features of today's smartphones and are not only used for AR but also to record high-quality videos. In this work we investigated a novel approach to recording and replaying video content in AR that is composited in-situ with a live view of the real environment. We show how the segmented video information can be precisely registered in real-time in the camera view of a mobile phone allowing a seamless integration into the user's view.

Our application is accompanied by a set of video post effects that can be applied in real-time on the video overlays. We further implemented a video layer functionality allowing users to combine several video augmentations running in parallel. We presented and evaluated our approach within the scope of a skateboard training application [69]. Our technique can also be used with online video material and supports the creation of augmented situated documentaries for AR browsers. We present this work in Chapter 6 of this thesis.

1.4.7 In-situ sketching of 3-dimensional content on mobile devices

Three dimensional content is an important element in AR applications, but usually this 3D content is created beforehand and outside of the target application. We developed a novel system that allows in-situ creation of 3D content for mobile Augmented Reality in unprepared environments. This system targets smartphones by allowing spontaneous authoring while in place. We describe two different scenarios, which depend on the size of the working environment, and consequently use different tracking techniques. A natural feature-based approach for planar targets is used for small working spaces, whereas for larger working environments, such as in outdoor scenarios, a panorama-based orientation tracking is deployed. Both are integrated into one system, allowing the user to use the same interaction for creating the content by applying a set of simple, yet powerful modelling functions. The results of this work were presented by Langlotz et al. [67] and are presented in Chapter 6 of this thesis.

1.5 Results

This thesis presents the research in the field of large-scale content authoring for mobile AR. The achieved results contribute to the field of Augmented Reality, Pervasive Computing, applied Computer Vision and Human Computer Interaction by investigating the effect of the content gap experienced in current generation AR browsers. It further presents the concept of user contributed social media for AR and demonstrates prototypes for implementing this concept. In particular, the community can learn the following lessons from this thesis:

- **Content availability and quality:** Content availability and quality are key for successfully establishing large-scale AR applications. We verified what other researchers described: "After a first 'wow-effect' wears off, users demand practical benefits which, in the case of AR, require a strong content creation pipeline" [149]. We also showed that the current content creation pipeline does not scale well with the number of AR browser users or with the global working environment.

- **User contributed content:** We determined user contributed content, or social media, to be a valid approach for guaranteeing content availability in dynamic scalable AR applications, particularly AR browsers. This aspect was already the driving force behind the evolution of the classical Web to the Web 2.0 and should therefore be applied as well to AR.

- **Support for a wide variety of content forms:** A wide variety of content forms increases the usefulness and applicability of applications such as AR browsers. Typically, AR applications work with either 3D overlays or textual annotations, but audio and video, both among the most popular forms of user generated content in the Web 2.0, are widely ignored. We show how integrating these content forms can extend existing AR applications, allowing a transition into new application areas. Besides demonstrating the use of textual annotations and 3D content, we also show in this thesis how to incorporate images, audio content and video overlays in use cases that were widely accepted by the users.

- **Spontaneous in-situ authoring:** Supporting user created content also requires supporting spontaneous authoring of content, which can happen while in-place. Within this thesis, we show that by carefully choosing the right interface and interaction for each type of content, we allow spontaneous in-situ creation of content within the AR browser application by inexperienced users and for various content forms.

- **Precise referencing of content:** Until now, AR browsers relied on sensor-based tracking leading to a rather error-prone system. This does not only limit the precision with which the content can be displayed, but also limits the resolution with which the content can be placed. While it seems that this problem is only recognized by a few users [34], we show that using a precise detection and tracking of the user's and content's position allows creation of new application scenarios which were impossible using a pure sensor-based tracking. While not applying a full six-degree-of-freedom tracking, our panorama-based orientation tracking supports the most common movement of the user while using an AR browser: having a static position and browsing the environment while rotating the device.

1.6 Selected Publications

This thesis consists of publications that are based on collaborations between various researchers from various institutions. The following list gives an overview about the publications and the people who were involved in the creation of them.

First, the following works summarize our investigations of the current generation AR browsers, especially in terms of usefulness, usage patterns, and limitations.

- **Tobias Langlotz**, Jens Grubert, Raphael Grasset, *Augmented Reality in The Real World:AR Browsers - Products or only Gadgets? Reflections on the initial adoption of AR technology by the general public*, Accepted for Communications of the ACM 2013.

- Jens Grubert, **Tobias Langlotz**, Raphael Grasset, *AR Browser Survey*, Technical Report, 2012.

  The author developed the questionnaire for the online survey, gathered the data from the mobile distribution platforms, and clustered as well as analysed the data from the mobile distribution platforms. The general reflections on AR browser technology mainly represent his viewpoint on the current state of the technology. ***Raphael Grasset*** *contributed to the questionnaire as well to the reflections and design considerations, whereas* ***Jens Grubert*** *contributed by analysing the results from the online survey as well as he gave valuable input to the discussions.*

Second, the concepts and requirements of Augmented Reality 2.0, as a concept for next generation AR applications incorporating social media, were firstly introduced in the following publication, but were further developed within the scope of this thesis.

- Dieter Schmalstieg, **Tobias Langlotz** and Mark Billinghurst. *Augmented Reality 2.0*, Springer Virtual Realities, Dagstuhl seminar proceedings (eds. Sabine Coquillart, Guido Brunnett, Greg Welch), 2010.

  The main author made significant input to the overall concept of AR 2.0 especially in terms of integrating social media by applying ideas from Web 2.0. He also contributed to identification of the requirements for an AR 2.0 environment. ***Mark Billinghurst*** *contributed to the general idea of AR 2.0 especially by integrating human centred design methodology and building the general concept on the results of past user studies researching Augmented Reality.*

The following papers are fundamental for this work, as they describe tracking technologies that are in general necessary to build an AR experience. Moreover, they are an technical enabler and requirement for an AR 2.0 environment.

- Daniel Wagner, **Tobias Langlotz**, Dieter Schmalstieg, *Robust and unobtrusive marker tracking on mobile phones.* Proceedings of the 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality 2008, IEEE Computer Society, 2008.

- Daniel Wagner, Alessandro Mulloni, **Tobias Langlotz**, Dieter Schmalstieg, *Real-time Panoramic Mapping and Tracking on Mobile Phones*, In Proceedings of IEEE Virtual Reality Conference 2010 (VR 10), 2010.

- Gerhard Reitmayr, **Tobias Langlotz**, Daniel Wagner, Alessandro Mulloni, Gerhard Schall, Dieter Schmalstieg, Qi Pan, *Simultaneous Localization and Mapping for Augmented Reality*, In Proceedings of the International Symposium on Ubiquitous Virtual Reality, Gwangju, South Korea, 2010.

  The tracking algorithms were mainly developed and implemented by ***Daniel Wagner.*** *The author made small contributions to the idea and optimizations of the implementation. He was further, together with* ***Alessandro Mulloni,*** *the main contributor to the final application part and the technical evaluation of the tracking algorithms.*

The following papers present first prototypes for an AR 2.0 environment displaying two-dimensional annotations. Both papers describe and discuss the user interface, but also have a strong focus on implementing novel approaches for precisely matching annotations anchor point. The author was the main contributor to nearly all stages of the work including design and implementation, designing and conducting user evaluations as well as the technical evaluation, and investigating the final results and performance.

- **Tobias Langlotz**, Daniel Wagner, Alessandro Mulloni, Dieter Schmalstieg, *Online Creation of Panoramic Augmented Reality Annotations on Mobile Phones*, IEEE Pervasive Computing, 2012.

- **Tobias Langlotz**, Claus Degendorfer, Alessandro Mulloni, Gerhard Schall, Gerhard Reitmayr, Dieter Schmalstieg, *Robust detection and tracking of annotations for outdoor augmented reality browsing*, Computers and Graphics, 2011.

  The author was the main contributor to the design and implementation. He further was responsible for the study design and the analysis of the results. ***Daniel Wagner*** *contributed to the template matching and especially the Walsh-Transforms used for optimizing the performance were added by Daniel.* ***Claus Degendorfer*** *contributed to the implementation for increasing robustness when matching annotations and helped to conduct the technical evaluation of the matching performance.* ***Alessandro Mulloni*** *contributed by adding the transitional interface, and together with* ***Gerhard Schall*** *and* ***Gerhard Reitmayr****, he contributed the north-aligned panorama tracking.*

Finally, the following publications extend the prototypes demonstrating Augmented Reality 2.0 by presenting interfaces allowing to use different media forms while also conducting studies demonstrating the applicability and usefulness in different scenarios. Similarly as to the previous work, the authors was the main contributor to design and implementations, designed and executed the studies, and was mainly responsible for the discussion of their results.

- **Tobias Langlotz**, Stefan Mooslechner, Dieter Schmalstieg, *In-Situ Content Creation for Mobile Augmented Reality*, In Proceedings of the ISMAR Workshop on Social Augmented Reality, 2009.

- **Tobias Langlotz**, Stefan Mooslechner, Stefanie Zollmann, Claus Degendorfer, Gerhard Reitmayr, Dieter Schmalstieg, *Sketching up the world: In-situ authoring for mobile Augmented Reality*, In Proceedings of the 2010 International Workshop on Smartphone Applications and Services (Smartphone 2010), 2010.

- **Tobias Langlotz**, Stefan Mooslechner, Stefanie Zollmann, Claus Degenorfer, Gerhard Reitmayr, Dieter Schmalstieg, *Sketching up the world: In-situ authoring for mobile Augmented Reality*, Springer Personal and Ubiquitous Computing, 2012.

- **Tobias Langlotz**, Mathäus Zingerle, Raphael Grasset, Hannes Kaufmann, Gerhard Reitmayr, *AR Record and Replay: Situated Compositing of Video Content in Mobile Augmented Reality*, In proceedings of OZCHI 2012.

- **Tobias Langlotz**, Holger Regenbrecht, Stefanie Zollmann, Dieter Schmalstieg, *Audio Stickies: Visually-guided Spatial Audio Annotations on a Mobile Augmented Reality Platform*, Submitted, 2013.

  The author was the main contributor to the design of the prototypes and was among the main contributors of the implementation. He was further the main contributor to the study designs and analysis, and conducted and majority of the studies. ***Stefanie Zollmann*** *contributed together with* ***Stefan Mooslechner*** *to the implementation for creating 3D models. She further helped implementing the scenario for evaluating the Audio Stickies.* ***Mathäus Zingerle*** *was with the author the main contributor to the implementation of the video augmentations.* ***Holger Regenbrecht*** *was involved in every part involving the design, implementation, and evaluation of the Audio Stickies.* ***Gerhard Reitmayr*** *and* ***Raphael Grasset*** *contributed to the design decisions of the prototypes as well as they were involved in the dissemination.*

The previously mentioned papers lead furthermore to the following patents that mostly resemble the presented ideas and were created with our project partner Qualcomm Inc.:

- Daniel Wagner, Alessandro Mulloni, Dieter Schmalstieg, and **Tobias Langlotz**. *Visual Tracking using Panoramas on Mobile Devices.*

- **Tobias Langlotz**, Daniel Wagner, Alessandro Mulloni, and Dieter Schmalstieg. *Online Creation of Panoramic Augmented Reality Annotations on Mobile Platforms.*

- **Tobias Langlotz**, Mathäus Zingerle, Gerhard Reitmayr. *Spatially Registered Augmented Video.*

1.7 Other collaborations

Some people have affected this thesis beyond the point of single publications. The following people need to be mentioned as their suggestions and feedback had a broader impact on this thesis:

- **Hartmut Seichter** and **Daniel Wagner** managed both the Christian Doppler Labor for Handheld Augmented Reality. They gave valuable input to general problems but also implementation details. Furthermore, they both contributed together with the other CDL members, most notably **Lukas Gruber** and **Alessandro Mulloni**, to Studierstube ES, the common shared framework for AR, which was used for implementing many of the produced prototypes.

- **Gerhard Reitmayr** and **Raphael Grasset** showed a strong interest in this work and shared a similar research interest. Their valuable input has affected and improved several of the listed publications, but goes beyond in a way that they gave valuable feedback to this thesis in general.

- **Holger Regenbrecht** and **Mark Billinghurst** allowed me to visit their research labs as a guest researcher. Their input and interest in my thesis, but also the critical questions, helped to frame the research questions as well as their provided feedback helped to significantly improve the quality of several works.

- **Reinhard Sainitzer** of Imagination Computer Services GesmbH was the main programmer of the Tomcat-based Servlet, which was used as a database to store the content.

Chapter 2

Related Work

Contents

Seen from the users' perspective, Augmented Reality is a relatively new technology. However, many commercial applications that are utilizing Augmented Reality build on knowledge gained from research and prototypes that date back to the mid 90s. Many AR applications that we see nowadays running on a smartphone are based on earlier prototypes that were demonstrated on bulkier hardware such as *backpack AR systems* - a combination of sensors and computers needed for AR that are carried, because of their size, in a backpack. But while commercial AR applications getting more attention from the public, the research community is also pushing further the limits of what is technically possible.

This chapter looks at existing work, giving background information that is central for understanding the context of this thesis. Since this thesis focuses on mobile Augmented Reality, we start in Section 2.1 by outlining remarkable points in the history of Augmented Reality, which led towards the development of *Mobile Augmented Reality* and finally *Handheld Augmented Reality.* We also consider important commercial applications that are worth mentioning. In Section 2.2, we give a brief introduction to the tracking technology that is available for building AR prototypes on mobile or handheld devices. Finally, in section 2.3 we show existing solutions targeting the creation and usage of content for Augmented Reality. In this section, we differentiate existing work depending on the form of media used - such as textual content, audio content or three-dimensional content. We continue to make these distinctions throughout this thesis.

2.1 Towards Mobile and Handheld Augmented Reality

While most existing definitions of Augmented Reality date from the mid 90s, the first AR application was created by Ivan Sutherland in 1968 [137]. His system, which is also known as "The sword of Damocles", is widely considered to be the first Augmented Reality system and at the same time the first *Head Mounted Display* (HMD). Ivan Sutherland's system allowed displaying a wireframe model that was augmented in the head mounted display.

After that, several years passed that were characterized by a continuous advancement in the field of computer science for both hardware and software. In 1992, Caudell and Mizell coined the term *Augmented Reality* (AR) [17]. They defined it as overlaying computer generated content on top of the real world and said, that AR could have performance advantages when compared to VR because less pixels need to be rendered. However they also stated that AR requires high registration accuracy to achieve convincing results. Around the same time, IBM presented the worlds first smartphone at the 1992 COMDEX (Computer Dealer's Exhibition). It took another year until the so-called IBM Simon Personal Communicator[1] became officially available. Even though this device was much more limited compared to today's smartphones - the main features aside from the phone capability were an integrated calculator, mobile fax and email functionality - it was the first step towards powerful and highly integrated mobile technology.

The same year the IBM Simon was released, Loomis et al. presented a research prototype to assist visually impaired people in navigating outdoor environments [74]. It was one of the first applications that used the Navstar GPS[2] (nowadays mostly referred to as Global Positioning System) to implement a *Location-Based Service* (LBS). The GPS receiver was attached to a notebook computer and combined with a head-worn compass. Based on label information from a *Geographic Information System* (GIS) database, the system gave navigational hints using a speech-synthesizer that spoke the labels.

At the end of 1993, the *Global Positioning System* (GPS) achieved initial operational capability. The GPS system's development started in the 1970s and was originally designed for military use only. But in 1983 an incident involving a Soviet jet which shot a Korean civil airplane because it accidentally (due to navigation errors) entered Russian airspace, prompted the US government to open the GPS system for civilian use. Even though its capability for civilian use was intentionally limited through *Selective Availability* (SA), which caused horizontal positional errors in the order of 100 meters, it became widely used for navigation but also in research as an enabler for location-based systems as well as Mobile Augmented Reality. Later development allowed for greater precision even for civilian users by using Differential GPS, which led to the decision to remove the Selected Availabilityfor civilian users in May 2000[3] .

Further steps, such as the Assisted GPS[4] introduced by Qualcomm in 2004, allowed a fast initialization of GPS within mobile phones by using additional position information to retrieve visible satellites. All of these developments, which originated in 1993, made GPS an essential part of today's generation of smartphones and a frequently used feature in current mobile AR applications.

In 1994, Paul Milgram and Fumio Kishino introduced their concept of the *Reality* -

[1] `http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/detail.aspx?id=40`

[2] `http://www.gps.gov`

[3] `http://ngs.woc.noaa.gov/FGCS/info/sans_SA/docs/GPS_SA_Event_QAs.pdf`

[4] `http://www.qualcomm.com/solutions/location/gpsone`

Virtuality Continuum (often also referred to as the Mixed Reality Continuum) [87]. The described continuum defines Augmented Reality as one technique within the Mixed Reality Continuum which is closer to Reality because only small portions are augmented, while reality still dominates the user's perception. In contrast, Augmented Virtuality describes a technique within the continuum which is closer to Virtual Reality because the user primarily perceives the virtual aspect and only small portions augment the reality on top of the virtual environment. Milgram's and Kishino's definition is nowadays widely used together with a later definition of Augmented Reality by Azuma.

In 1995, Jun Rekimoto and Katashi Nagao presented their NaviCam (Navigation Camera) prototype. Even though they used a tethered setup similar to the concept of position sensitive PDAs proposed by Fitzmaurice [30], it can be seen as one of the first mobile AR prototypes. The mobile client - a palmtop computer - had an attached camera and streamed data to a connected workstation. Using received video information the workstation computed visible objects based on detected coloured markers. The visible objects were then used to augment situation aware information onto the camera image. The combined image was sent back to the mobile client where it was displayed to the user.

Later in 1996, Rekimoto presented an improved version of the coloured barcodes used for detecting objects [111]. The system used 2D matrix codes - squared barcodes - which allowed computing the pose information of the camera with 6 *Degrees of Freedom* (DoF). This approach represents one of the first marker-based tracking systems. Similar approaches for tracking are used today and represent the foundation of many of AR applications.

In 1997, Ronald Azuma presented his survey of Augmented Reality [5]. Besides writing the first survey on Augmented Reality, he presented a new definition of Augmented Reality. According to Azuma, Augmented Reality is defined by three main characteristics: Augmented Reality (1) combines real and virtual, (2) is interactive in real time and (3) is registered in three dimensions. Azuma's definition of Augmented Reality is widely acknowledged and together with the earlier definition by Milgram and Kishino [87] the most used definition of Augmented Reality.

Also in 1997, Steve Feiner et al. [28] presented the first truly mobile Augmented Reality system. Unlike earlier systems, which used a mobile client together with a stationary workstation, Feiner et al. managed to integrate all components into a backpack system. The MARS (Mobile Augmented Reality System) system uses a head-worn see-through display, an orientation tracker, a differential GPS, a digital radio for wireless access of remote information, and a handheld device with stylus and touchpad as an interface with the system. The core of the system was a laptop computer carried in a backpack. Feiner et al. used this prototype to realize the Touring Machine, a mobile Augmented Reality system displaying campus information. The Touring Machine can be seen as the origin of today's mobile AR browsers in terms of application scenario, but also in terms of the technology used - such as GPS and compass for tracking.

In the same year, Thad Starner et al. [135] explored possible applications of wearable Augmented Reality systems. They created the Remembrance Agent, a personal assistant that "thinks" for users and displays context sensitive information to them. Starner et al. implemented information systems for offices which perform recognition of people and maintenance tasks. They also foresaw applications such as a dynamic, physically-realized extension to the *World Wide Web* (WWW) and potential problems like visual clutter.

Another important milestone relevant for the further development of mobile Augmented Reality was the invention of the camera phone by Philippe Kahn in 1997[5]. This invention was an important enabler; nowadays hundreds of millions of smartphones have integrated cameras, making them a platform for vision-based Augmented Reality applications.

In 1998, shortly after the Touring Machine was presented, Bruce Thomas et al. presented "Map-in-the-hat", another backpack-based mobile AR system [140]. Similar to the Touring Machine it includes GPS, an electronic compass, and a head-mounted display. Initially, this system was mainly used for demonstrating AR-based guidance in outdoor environments. This prototype was the starting point for the Tinmith system into which it eventually evolved. At the same time, Tobias Höllerer et al. presented a further evolution of the Touring machine prototype [46]. The previously used technology evolved further with the use of Real Time Kinematic GPS for higher precision in the position estimate and an inertial-magnetic orientation tracker for better orientation tracking. Höllerer et al. demonstrated a guided campus tour using hypermedia news stories overlayed onto the buildings of the campus. Later, this system was also extended by adding a indoor interface [47].

Inspired by the Touring Machine and other early mobile AR applications for outdoor scenarios, Jim Spohrer created the Worldboard concept [133]. The Worldboard presents a scalable infrastructure to support mobile applications from location-based systems to mobile AR applications. Spohrer also envisioned possible application cases for mobile AR for end-users and professionals (see Figure 2.1) as well as stating social implications of the Worldboard.

In 1999, Hirokazu Kato and Mark Billinghurst presented ARToolKit [53]. ARToolKit is a marker-based pose-tracking library working with 6DoF. ARToolKit was and still is a commonly used tracking library for Augmented Reality, which later made its way onto mobile phones as well as being ported to other programming environments such as Adobe Flash[6].

In 1999, the ongoing process of hardware miniaturization reached a point when GPS modules could be integrated into mobile phones. The Benefon Esc! NT2002[7], was the first GSM phone with a built-in GPS receiver and was released in late 1999. While the phone was not ready for Augmented Reality applications, as the screen was a low-resolution greyscale display and the processing power was not sufficient, the phone already demonstrated the first location-based services such as a friend finder. However, using these services required another phone of the same brand.

One year after the first mobile phone with GPS hit the market, Sharp released the first commercial phone with an integrated camera, the Sharp J-SH04[8]. The phone's camera resolution was 0.1 megapixels. While the phone was also not capable of running AR applications, mainly because of the low processing power, all main sensors were now available for mobile phones. This together with the price and the availability for end-users eventu-

[5] `http://www.computerhistory.org`

[6] `http://www.libspark.org/wiki/saqoosha/FLARToolKit/en`

[7] `http://www.benefon.de/products/esc/`

[8] `http://k-tai.impress.co.jp/cda/article/showcase_top/3913.html`

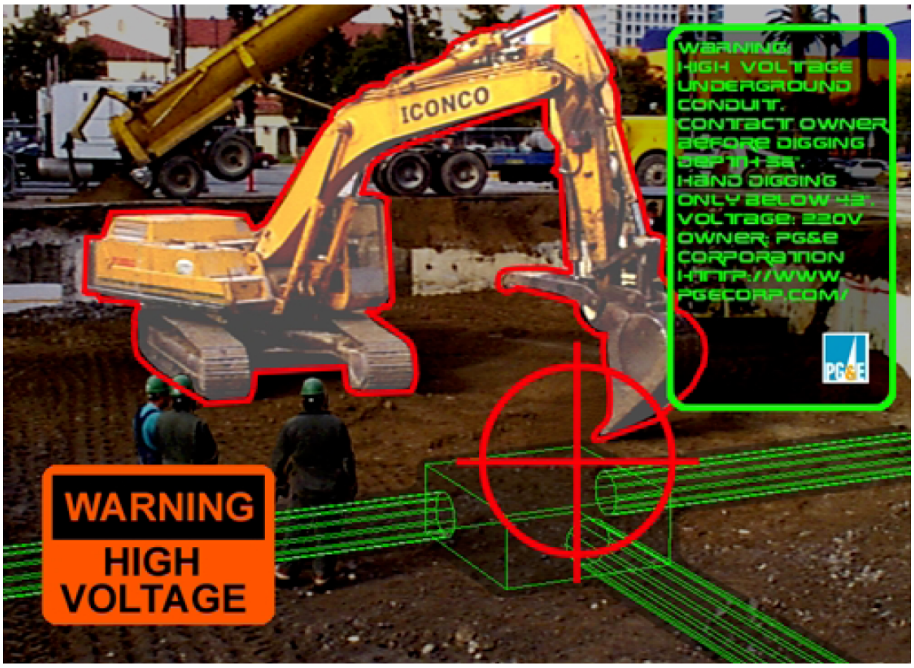


Figure 2.1: AR supported excavations as described in the Worldboard [133].

ally turned mobile phones into one of the most promising platforms for a wide range of Augmented Reality applications over the next years.

In 2000, the research community introduced several further application scenarios for mobile AR using backpack AR systems. Bruce Thomas et al. demonstrated AR-Quake, an extension to the popular desktop game Quake [139]. AR-Quake made use of a system similar to the previous Map-in-the-hat prototype [140] allowing the game to be played in an outdoor environment where the movements of the users in the real world are input into the game. Other use cases for mobile AR were presented by Simon Julier et al. with their Battlefield Augmented Reality System (BARS) [131]. The system also used a backpack AR system similar to the systems by Feiner et al. and Thomas et al. but demonstrated a military use-case, where a battlefield scene was augmented with additional information such as environmental infrastructure and enemy information.

In contrast to these systems, Regenbrecht and Specht presented mPARD, a different approach for a mobile AR system [107]. The demonstrated mobile unit can be reduced in terms of size and weight by using wireless video transmission to a desktop computer that performs the vision and graphics computations. The video feed is sent to the desktop computer, where the overlays are computed and applied to the video feed. The final frames are sent back over the wireless connection and are displayed on the mobile client. While the working environment is limited to within 200-300m to maintain the connection, the device size is smaller and lasts for up to 5 hours of operation.

In 2001, Satoh et al. showed an extension for existing backpack AR systems, which increased precision in the orientation estimate. They used a fiber-optic gyroscope for orientation tracking, combined with natural feature tracking to compensate for the drift

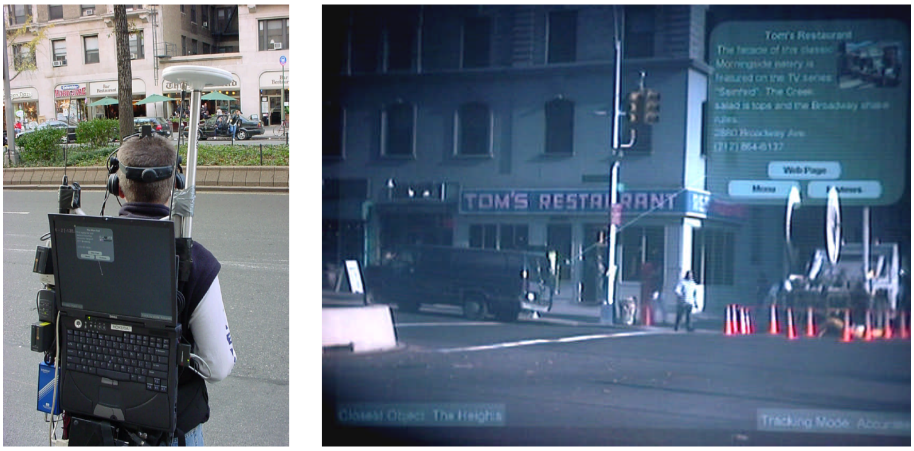


Figure 2.2: (Left) Touring Machine prototype presented by Höllerer and Feiner [45], (Right) An AR-based restaurant guide running on MARS

of the gyroscope [120].

Since 2000, Personal Digital Assistants (PDAs) gained more attention as their hardware became more capable of handling complex applications. In 2001, several research teams presented their work which relied on PDAs as a central component in their systems. Joseph Newman et al. presented the BatPortal [95] a wireless AR system running on a PDA. Newman et al. used an ultra-sonic pulse between a specially built device worn by the user, the so-called Bat, and fixed installed receivers deployed in the floors ceilings building-wide for localizing the users within the building.

In contrast to Newman et al., Vlahakis et al. presented Archeoguide, a PDA-based system for outdoor environments. The system was used to experience cultural heritage sites using AR as demonstrated in Olympia, Greece [148]. The system can be run on several devices including PDAs with GPS and was used to display 3D models of ancient temples and statues. Additionally, users could compete within an integrated game. Another system for cultural heritage sites was presented by Kretschmer et al. [61]. The so-called GEIST system can be used for interactive story-telling within urban or historical environments. Later in 2001, Fründ et al. presented a concept for building a wireless AR system using PDAs, called Augmented Reality Personal Digital Assitant (AR-PDA) [32].

The Real-Word Wide Web Browser (RWWW Browser) presented in the same year by Kooper and MacIntyre created one of the first mobile AR browsers [60]. Albeit not running on a mobile phone but using a head-mounted display instead, Kooper and MacIntyre used this system to access and display information from the World Wide Web. In 2008, Wikitude revised this idea, but this time on a mobile phone. Other earlier AR browser prototypes include the mobile AR restaurant guide demonstration by Feiner et al. at ISAR 2001, developed within their MARS project [45].

By 2003, PDA devices were available from different brands making them an interesting off-the-shelf platform for AR. Daniel Wagner and Dieter Schmalstieg ported ARToolkit to PDAs running Windows Mobile. They utilized it for an indoor AR guidance system running autonomously on a PDA [154].

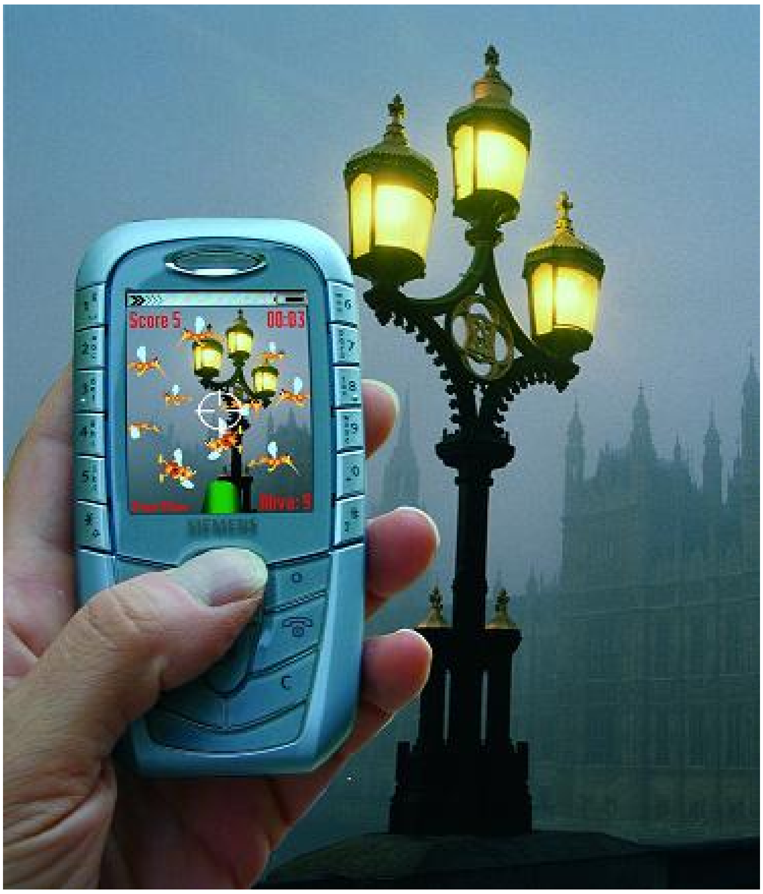


Figure 2.3: ”Mosquito Hunt” the first commercial AR game for mobile phones that made use of the integrated camera shipped with the Siemens SX1

PDAs were also used by Ramesh Raskar et al. when they presented iLamps [106] a first approach for *Spatial Augmented Reality* (SAR) on handheld devices. iLamps used a handheld projector-camera system to determine the geometry of the display surface. Once the geometry was known, the projector could then be used to augment the surface.

Slowly, mobile phones and especially the new generations of powerful mobile phone with multimedia capabilities, usually called smartphones, became capable of running simple AR applications. The Siemens SX1 smartphone[9] was released with an application which became known as the first commercial AR game for mobile phones using the integrated camera. Mozzies or Mosquito Hunt was a mosquito hunting game where mosquitoes are superimposed on the live video feed from the camera. Users could aim at the mosquitoes by moving the phone around so that the crosshair points at the mosquitoes (see Figure 2.3).

Other mobile AR games include Human Pacman, which was presented by Adrian David Cheok et al. [20] in 2004. In this version of the classic arcade game, Pacman and the ghosts

[9] `www.siemens.de`

were real human players that played the game in a physical world augmented with game artifacts that were perceived using an HMD. The users' movements were tracked using GPS and inertia sensors, similar to backpack AR systems, even though the components were smaller in size.

Up to this point, nearly all system were lab prototypes that required a lot of time and expert knowledge to produce the displayed content. To overcome this problem, Sinem Guven showed the first authoring system for mobile AR, which was aimed at people who are not experienced in programming [36]. The presented authoring system allowed users to create and edit 3D hypermedia narratives that were linked to their physical environment.

The following years brought a breakthrough for mobile AR applications running on smartphones. Unlike previous mobile AR systems that were sometimes quite heavy, all necessary computational power could now be packed into a small device that fit in one's hand. Therefore, the term *Handheld Augmented Reality* (HAR) was coined and is often used to distinguish this category from other mobile AR systems [154].

CPU intensive tracking using computer vision was the main bottleneck for AR applications running on smartphones. In 2004, Mathias Möhring et al. demonstrated for the first time real-time tracking of coloured 3D markers on mobile phones [88]. Running on off-the-shelf consumer devices, Möhring et al.'s prototype demonstrated detection and tracking of different coloured markers for video see-through augmented reality systems.

Visual Codes, another marker system for mobile phones presented by Michael Rohs and Beat Gfeller [85], were mainly used for interacting with real world objects. Two-dimensional Visual Code markers, attached to physical objects or displayed on screens, were used to retrieve information and interact with the information by pointing or moving the phone with respect to the marker. However, Rohs et al. did not implement a pose tracking with 6DoF, but mainly relied on the size of the marker in the camera image.

In 2005, Anders Henrysson first presented a tracking system supporting 6DoF on a mobile phone by porting ARToolKit to Symbian [43]. He used this system to implement an AR-Tennis game. Two users could play tennis against each other on a superimposed virtual tennis court. AR-Tennis was awarded the Independent Mobile Gaming best game award for 2005 and won a technical achievement award.

Later in 2005, Makri et al. presented within the Project ULTRA non-realtime *Natural Feature Tracking* (NFT) on PDAs [82]. The proposed application domains were maintenance and support of complex machines, as well as edutainment and cultural heritage by adding information overlays.

Most presented trackers solely relied on sensors such as GPS and compass, or on visual tracking. In 2006, Reitmayr et al. presented the results from investigating a model-based hybrid tracking for outdoor augmented reality in urban environments [108]. The presented system worked in real-time on handheld PC devices by combining an edge-based tracker for accurate localization, gyroscope measurements to deal with fast motions, and measurements of gravity and magnetic field to avoid drift.

In the following year, Klein and Murray presented a novel approach for *Simultaneous Localization and Mapping* (SLAM) by separating the tracking and mapping task into two separated threads [56]. Their approach called *Parallel Tracking and Mapping* (PTAM) is widely acknowledged in the AR community, and many later AR prototypes rely on this technology. The advantage of a SLAM approach is that after initialization no existing scene knowledge is required and the scene is mapped while running the tracking. Therefore it is an interesting solution for tracking in partially unknown environments. Still, a SLAM-based tracker requires an initialization by giving an initial pose from which it can start to track using SLAM.

In 2007, Apple joined the group of smartphone producers by announcing the iPhone[10]. While many features of the iPhone were seen in previous devices, the iPhone revolutionized the how they were integrated into one device. The form factor and interface of the iPhone affected all future generations of smartphones. Later, in 2008, Apples App Store opened, revolutionizing the how applications are distributed to customers' smartphones. As a consequence, smaller companies in particular reached a higher amount of end-users through the App Store than previously possible. The App Store also plays an important role for AR browser companies in distributing their applications.

The same year, the Human Interface Technology Lab in New Zealand, and Saatchi and Saatchi created the first mobile phone-based AR advertising campaign for the Zoo in Wellington[11]. The campaign used markers that were printed in the local newspapers which showed superimposed zoo animals seen through a phone's viewfinder.

In 2008, the first interactive tracking approach using only natural features on consumer mobile phones was developed. Wagner et al. modified the previously presented algorithms for tracking natural features such as SIFT and Ferns and optimized them for speed and memory footprint [153]. Both algorithms - the SIFT derivative and the Ferns variation - were running with frame rates of up to 20 Hz [153].

Later in 2008, the first commercial AR browser named Wikitude was presented by Mobilizy GmbH (now Wikitude GmbH)[12]. Wikitude uses GPS position together with compass data to display geo-referenced Wikipedia articles superimposed on a live camera feed (see Figure 2.4). Wikitude gained further popularity when named among the finalist at the 2008 Android Developer Challenge [13] targeted at developing innovative applications for the upcoming first Android-based device: The T-Mobile G1 (also known as HTC Hero). Later Wikitude was also ported to other platforms such as iOS.

Based on the early success of Wikitude, in 2009 SPRXmobile presented Layar as another competitor in the field of AR browsers. Similar to Wikitude, Layar uses GPS, compass and accelerometer for tracking the users' movement. It further consists of a content distribution model where content layers can be added. Each content layer contains information of a specific domain (e.g., GPS-tagged Twitter feeds, Flickr images) that is displayed onto the camera image. By announcing Layar as an AR browser, SPRXmobile also coined the term *AR browser*[14], which is nowadays used for many applications with a similar purpose.

Improvements in tracking on handheld devices continued in 2009. Georg Klein presented a version of his PTAM system which ran in real-time on a consumer smartphone [58], making it the first SLAM system to run on mobile phones at interactive frame rates. However, the presented system still shares characteristics with the original system such as

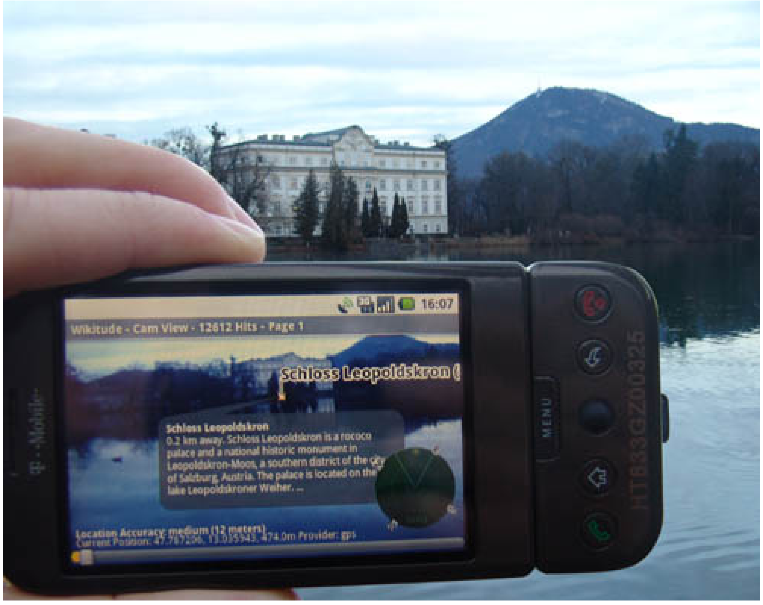
[10] http://www.apple.com/iphone/

[11] http://theinspirationroom.com/daily/2007/augmented-reality-at-wellington-zoo/

[12] http://www.wikitude.com

[13] https://developers.google.com/android/adc/adc_gallery/

[14] http://www.sprxmobile.com/we-launched-layar-worlds-first-augmented-reality-browser-for-mobile/


16:07
Wikitude - Cam View - 12612 Hits - Page 1
Schloss Leopoldskron (
Location Accuracy: medium (12 meters)
T - Mobile
MENU

Schloss Leopoldskron
0.2 km away. Schloss Leopoldskron is a rococo palace and a national historic monument in Leopoldskron-Moos, a southern district of the city of Salzburg, Austria. The palace is located on the lake Leopoldskroner Weiher. ...

Figure 2.4: Wikitude, the first commercial AR browser running on the T-Mobile G1

the small working volume. In 2010, Wagner et al. presented a SLAM-inspired approach for mobile phones the addressed some of these issues by using a panoramic map instead of a 3D model of the environment [152]. That approach aimed particularly at outdoor AR application, and though unable to track the camera in 6DoF, it still allows a precise tracking of the orientation and does not require any scene knowledge. The author of this thesis contributed to that approach and used it in many prototypes discussed later. A more detailed discussion of this approach can be found in Chapter 3.

It took until 2011 for SLAM trackers designed for use on mobile phones to become available commercially . At that time, 13th Lab presented Pointcloud, the first commercial SDK for creating applications using SLAM on mobile phones [15]. Pointcould made it possible for non-tracking experts to integrate SLAM-based tracking into end-user applications.

Despite the fact the SLAM-based trackers are described in research and are slowly becoming available as commercial solutions, SLAM-based trackers suffer from one main drawback: They need to be initialized with a pose. In 2011, Arth et al. presented a panorama-based approach together with a server hosting a reconstruction of the environment that can be used to determine an initial pose [2]. Using a panorama, which is created on the phone and sent to a server, a pose can be computed to initialize any kind of tracking such as SLAM.

Kurz et al. also presented several extensions to tracking approaches using natural features [63] [62]. Kurz et al. showed that by using the gravity vector of the integrated gyroscope or other sensor input natural feature tracking is more robust. This is achieved by tracking steeper angles with respect to the tracked target and also allows for track-

[15] `www.13thlab.com`

ing targets where certain features have a similar appearance but different orientations (repetitive structures). This could be used in future SLAM trackers as well as to stabilize tracking of planar targets.

While several research groups continued to work on improving tracking quality for AR applications and AR browsers, Blair MacIntyre et al. investigated content distribution for AR browsers. By relying on open standards, they tried to overcome the proprietary protocols and formats used by all AR browsers thus far [78]. The resulting Argon AR browsers used KARML, an open AR extension for KML (Keyhole Markupe Language)[16] to describe the content.

At the time of writing this thesis, Layar just recently announced that 25 million people have downloaded the Layar AR browser[17]. Together with the other AR browsers, for instance Wikitude, Junaio, Acrossair, just to name a few, AR browsers achieved between 40-50 million downloads. This makes AR browsers the most successful type of AR applications in terms of user awareness and marketshare. In addition, it showed that the success of smartphones in combination with an infrastructure such as mobile app stores created a viable platform for AR applications. However, despite the success of AR browsers in terms of user awareness, many challenges remain. Primarily there is not much existing work on how this commercial mobile AR applications are received by the public. This adds to the general lack of user studies investigating end-user AR applications as already pointed out by Dünser et al. and [26] and Schmalstieg et al. [125]. Additionally, there are many other open issues that are more technology related such as as the quality of tracking, rendering, the interface, but also the displayed information itself. Finally, there is not much existing work on novel ways of creating and authoring content for AR application in general, but also for AR browsers in particular. So far, the displayed content is mostly created using traditionally applications such as 3D editors or web-interfaces that are not designed for inexperienced users, nor do they make use of AR as interface when creating the content.

2.2 Tracking for Mobile Augmented Reality

The ability to track a camera's position and orientation is an essential requirement for creating a convincing integration of virtual content into the real world. This is particularly true because many other components of an AR system, such as interaction techniques, strongly depend on the characteristics of the used tracking [37]. For example, a camera-based pointing and selection metaphor is hardly useful if the tracking is prone to jitter or drift.

Having highlighted the importance of tracking approaches, we give here a brief summary of the current state of tracking for Augmented Reality. Given the context of this thesis, we will focus especially on tracking using mobile and handheld devices in outdoor environments.

Azuma gave a good overview of the requirements of tracking for Augmented Reality. He stated that tracking should be interactive in realtime (usually above 15-20 fps with 30fps as desirable), and for most applications a tracking with 6DoF is necessary to register

[16] `https://developers.google.com/kml/documentation/`

[17] `http://www.layar.com/blog/2012/09/17/thanks-for-25-million-downloads/`

the augmented information in 3D [5]. Furthermore, Azuma stated that tracking accuracy should be accurate within a small fraction of a degree in orientation and a few millimetres in position [4]. These remain challenging requirements, especially in outdoor environments.

The first outdoor AR systems created within the MARS project used a combination of GPS (position), magnetometers, and inertial sensor (orientation) for tracking cameras movement [28] [47]. This tracking setup became the preferred choice for most backpack AR systems [140] [139].

Currently, most outdoor AR applications for mobile phones and especially AR browsers (for example, Layar and Wikitude) rely on sensor-based tracking. Consequently, they suffer from the same drawbacks such as inaccuracy in terms of pose estimate and pose stability (jitter) as earlier systems. Feiner et al. [28] already stated in 1997 that due to inaccurate sensors, it is only possible to assign annotations to buildings and not to fine details (e.g., windows, advertisements). This precision was already not within the boundaries of Azuma's tracking requirements, but is even worse on mobile phones due to both, the lower quality of the integrated sensors and the limited size of the housing. While the backpack AR systems had only marginal constraints with regards to the placement of the device and antennas, the form factor of mobile phones arranges the sensors in way that they can easily interfere with each other and greatly constraints towards the antenna design.

Hardware and software manufacturers try to increase the accuracy of GPS in certain scenarios by incorporating knowledge about the cell ID/position of the currently used antenna or the position of the currently used WiFi cell[18]. Furthermore, most mobile phones use Assisted GPS to correct the position and increase the startup-time using additional information transmitted via the phone network.

Still, accuracy lags behind external dedicated sensors. Zandberg et al. determined the positional error of Assisted GPS sensors in mobile phones in outdoor tests. The position given by the mobile phone revealed a median error between 5.0 and 8.5m which is an increase by a factor of 2 to 3 compared to standalone GPS sensors. However, Zandberg et al. also stated that very large errors are uncommon and rarely exceed 30m [163].

Error in orientation from magnetometers is typically within a few degrees [66], but is heavily affected by external factors such as the proximity of ferromagnetic materials [121]. Additionally, the integrated sensor and especially the GPS sensor have an update rate which often does not reach interactive frame rates (1.5 Hz in the case of the iPhone 3GS GPS sensor). Overall, it seems not possible to fulfill Azuma's tracking requirements in outdoor environments by relying solely on current generation sensor technology.

Using computer vision approaches yields the possibility to combine a high update rate with higher precision, but often requires scene knowledge or a prepared environment.

The first systems used fiducial markers which were detected in the camera image and enabled computation of the position of the marker with respect to the camera [53]. While initially demonstrated on desktop PC, several groups including the authors own demonstrated marker tracking to run on mobile phones with a precision and update rate, fulfilling Azuma's requirements towards AR tracking [88][85][154][43][150].

Unfortunately, utilizing marker-based tracking requires inserting fiducial markers into

[18] `http://www.apple.com/pr/library/2011/04/`

the environment, which becomes increasingly problematic particularly for continuous tracking in large scenes. Natural feature-based tracking, such as presented by Wagner et al. [153], does not rely on artificial features, but on existing planar features within the environment. However, scene knowledge in form of a feature database is required to match the features visible in the camera image with the feature database. This, together with the assumption of planarity, constrains its usefulness in outdoor scenarios.

More recent research tries to overcome the necessity of existing scene knowledge by creating a map containing scene knowledge in form of 3D feature points while tracking from it [56]. These approaches are usually referred to as *Simultaneous Localisation and Mapping* (SLAM) approaches. First demonstrated on handheld devices by Klein et al. [58], SLAM-based approaches continue to pose several open challenges. First, they need to be initialized with an initial pose, and secondly, SLAM approaches usually require a translational movement to create the map, which is not the typical movement pattern of AR browsers [34]. Finally, storing and maintaining a growing map of the environment becomes increasingly computationally demanding and memory expensive.

In our work presented in Chapter 3, we show an approach that is inspired by SLAM, but uses a different approach to create and maintain the map to overcome some of the existing problems of SLAM approaches [152]. Instead of a using a 3D map, we create a panoramical representation of the environment (a 2D map). While this approach only permits tracking in 3DoF (orientation only), we can show that it possesses several advantages over existing SLAM systems, particularly when used in outdoor environments. Similar research was presented by DiVerdi et al. [23]. Their approach tracks camera orientation in real-time and simultaneously creates an environment map by calculating the optical flow between successive frames. These measurements are refined with more computationally expensive landmark tracking to avoid the drift introduced by frame-to-frame feature matching. While the results of this approach are similar to our approach, it relies heavily on the GPU and is barely able to run in real-time on mobile phones. Our approach uses a different algorithm and is optimized for running on mobile phones (see Chapter 4).

Other research aims to complement SLAM-based approaches by solving the problem of initialization. Chen et al. [19] showed how to determine the current position by matching the camera image against streetview images. Arth et al. [4] extended this idea by taking wide-angle pictures (panoramas) instead of a single camera image. The advantage is that panoramas hold more environment information that can then be used for matching. Arth et al. also demonstrated how this system can be combined with our panorama-based tracker [152] (also see Chapter 4) for precise localisation in world coordinates, while having real-time tracking on the device.

There is an increasing number of solutions that combine the precision of vision-based trackers with the general robustness of sensor-based tracking. These hybrid tracking solutions especially prove to be useful when large environments need to be tracked. Satoh et al. presented a high precision gyroscope for orientation tracking that is combined with natural feature tracking to compensate for drift [120]. Similar systems have been presented by You et al. [162], Jian et al. [50] and Schall et al. [122], while Ribo et al. [114] showed how to recover with tracking in 6DoF using hybrid tracking.

Reitmayr et al. demonstrated a system that combines a visual tracker tracking edges with sensor readings [109] [108]. The system is initialized via GPS but the vision-based tracker enables precise tracking at interactive frame rates. In case of occlusion or motion blur in the camera image, the inertial sensor is used to continue tracking.

Overall, tracking is still an emerging field of research, and proves especially challenging in outdoor environments. SLAM-based systems seem to be the preferred approach as they allow precision but do not require existing scene knowledge. Still, they must be accompanied by sensor-based tracking mechanism to allow initialization - if only to reduce the search area for vision-based localization - and to handle robustness in the case of fast movement or if occlusions occur.

2.3 Editing and Using Situated Media in Mobile AR

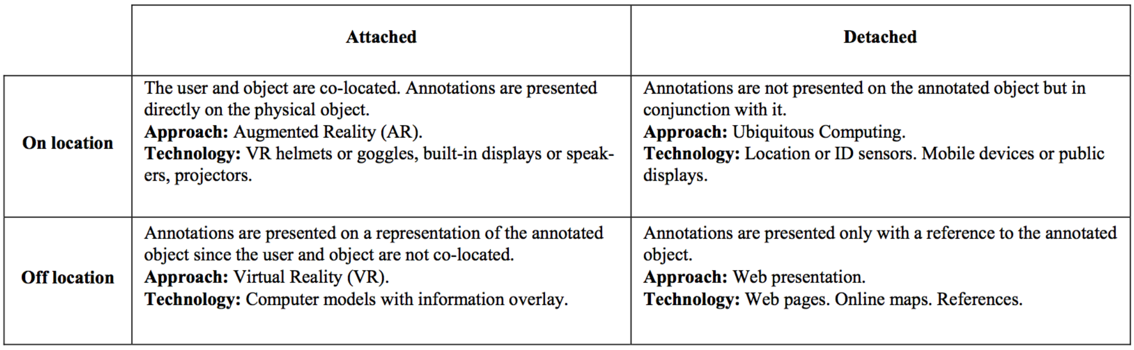
Associating information with places, objects or persons is a technique that is used in various research fields. Regardless of field, textual information is probably the most common form of information incorporated, but there are other types as well (e.g., graphical information or audio information). Often, information linked to a physical entity is referred to as an *annotation*. In the physical world, annotations could be comments or drawings made on newspapers or research papers to highlight key sections or to give feedback. Similarly, graffiti and tagging on walls and objects can be seen as a form of physical annotation.

With the rise of digital information, many aspects of linking or annotating digital information to physical entities were investigated. Hansen created a taxonomy for all kinds of systems utilizing digital annotations [40]. In his taxonomy, Hansen described four main challenges for any type of ubiquitous annotation system: *anchoring*, *structure*, *presentation* and *editing*. The way these challenges were addressed also characterises the created approaches.

According to Hansen, anchoring describes the linkage between physical entities (persons, places, and objects) and information. He also noted that the type of anchoring used affects the precision of “how well annotations can be placed in relation to the annotated resources“ [40]. The structure of an annotation describes the object relationship. The structure especially describes how the relationship is modelled and represented within the digital data. The presentation according to Hansen, describes the type of information that is presented and especially how it is presented in relation to the physical entities. Finally, editing describes how the annotation is edited or authored.

Guven et al. introduced a different terminology for digital annotations by introducing the concept of *situated media* [35]. Similar to Höllerer, who previously introduced the term *situated documentaries* as ”narrated multimedia documentary within the same physical environment as the events and sites that the documentary describes” [46], Guven defined the term situated media as referring to multimedia and hypermedia that are embedded in the surrounding spatial environment. However, neither Hansen nor Guven discuss social media as user contributed media [52] in their approaches.

In the following we use both situated media and annotation to describe the same content: digital information or media linked to physical entities (persons, places, and objects). Because the presented work will be mostly in the domain of Augmented Reality, a *augmentation* is defined to be an annotation or situated media presented using AR. We further extend these definitions by introducing the term *situated social media*, defining it



	Attached	**Detached**
On location	The user and object are co-located. Annotations are presented directly on the physical object. **Approach:** Augmented Reality (AR). **Technology:** VR helmets or goggles, built-in displays or speakers, projectors.	Annotations are not presented on the annotated object but in conjunction with it. **Approach:** Ubiquitous Computing. **Technology:** Location or ID sensors. Mobile devices or public displays.
Off location	Annotations are presented on a representation of the annotated object since the user and object are not co-located. **Approach:** Virtual Reality (VR). **Technology:** Computer models with information overlay.	Annotations are presented only with a reference to the annotated object. **Approach:** Web presentation. **Technology:** Web pages. Online maps. References.

Figure 2.5: Taxonomy for presenting annotations in relation to the annotated objects according to Hansen [40].

as situated media produced and shared by novice users by means of crowdsourcing.

In the following, we present related work in the field of Mobile and Handheld Augmented Reality by focusing on how different forms of annotations are used and edited in AR. We thereby follow when possible the taxonomy of Hansen to describe differences of the presented work with respect to anchoring, structure and editing. We reduce the presentation of annotations to Augmented Reality as interface (form of presentation) allowing an attached, on-location presentation (see Figure 2.5).

We extend Hansen's taxonomy with *media type*. While Hansen targets annotations in their general form, not differentiating between certain media types, we add the type of media used as a new classifier to distinguish existing AR systems, but also to show how the content is anchored, edited and modelled in a structure based on the media type. In the following, we differentiate between these media forms: text, audio, video and 3D models.

2.3.1 Editing and Using Textual Augmentations in Augmented Reality

Supporting textual augmentations is one of the key functionalities of most AR systems. An overview on the related works that investigated creation, use, and editing of textual annotations in the context of mobile AR applications is given in the following.

The Touring Machine [28] created within the MARS project [47] was the first prototypical mobile AR system to demonstrate the advantage of augmenting information over physical objects by using a backpack AR system and a head-mounted display to overlay textual annotations on real-world objects. As in most of the early outdoor AR systems, the anchoring is based on GPS position. The camera of the AR system is tracked using GPS in combination with compass and gyroscopes for tracking the orientation.

Later work by Kooper and MacIntyre [60] showed how geo-registered information within AR could be linked to online information sources - often of textual form - such as web pages. However, these prototypes suffered from the registration performance resulting from sensor-based tracking, and the bulky equipment or stationary infrastructure was not intended for daily or mobile use.

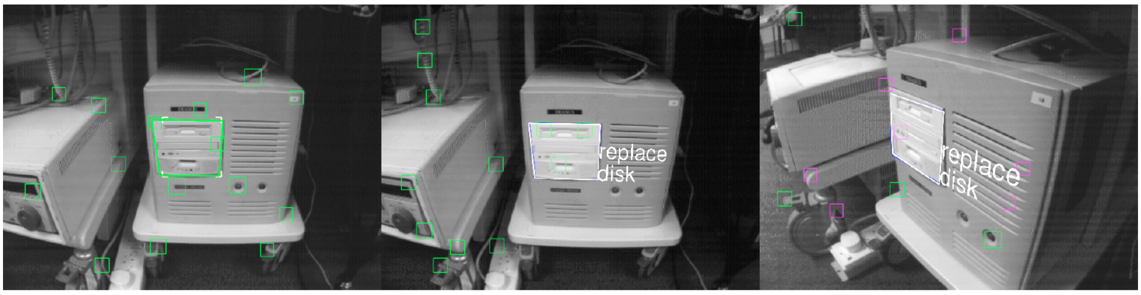


Figure 2.6: Semi-automatic placement of annotations. After selecting the annotation area (Left) and placing the annotation (Middle) the system calculates the 3D position of the annotation using the map of the used SLAM system (Right) [109].

Both approaches can be seen as the conceptual origins of the development of commercial AR-browser applications running on smartphones such as Wikitude or Layar. These commercial systems present annotations from databases that were created offline and anchored using a GPS position. Despite the fact that these AR browsers also support other media formats, we showed that textual content is the dominant media type used [34] (see Chapter 3).

Most of these AR browsers try to reduce the number of visible overlays through filters or layers to lessen the amount of visual clutter. However, usability is not only impacted by the number of textual annotations, but also their placement [11]. Bell et al. therefore stated that view management is essential to avoid ambiguity, as well as to ensure that important parts of the scene are not hidden [11]. Later, other research groups presented approaches for inserting labels, which required less prior information of the environment. These works include Rosten et al. [117], who used the concentration of visual features to identify areas that contain a huge amount of important information and therefore should not be occluded by labels. Later this idea was extended by Grasset and the author of this thesis by combining saliency computation and edge detection to identify areas that should be avoided when placing annotations. The created prototype was also demonstrated to run on mobile devices [33].

While most of the systems displaying textual augmentations rely on content prepared offline, only a few related works focus on creating annotations directly in an AR view. Early work on in-situ authoring targeted placing virtual objects in a real scene, and supporting users through triangulation from different views [104]. Rekimoto et al. presented Augmentable Reality, which allowed for annotating an environment prepared with barcode markers that refer to contextual information [112].

A number of approaches exist for this online annotating. For example, Reitmayr et al. [110] used an existing 3D model of the environment to calculate the exact position of an annotation by casting a ray into the scene. Later Reitmayr et al. described a set of techniques to simplify the online authoring of annotations in unknown environments using a simultaneous localization and mapping (SLAM) system [109] (see Figure 2.6).

More recently, Wither et al. presented several techniques aimed at solving the problem

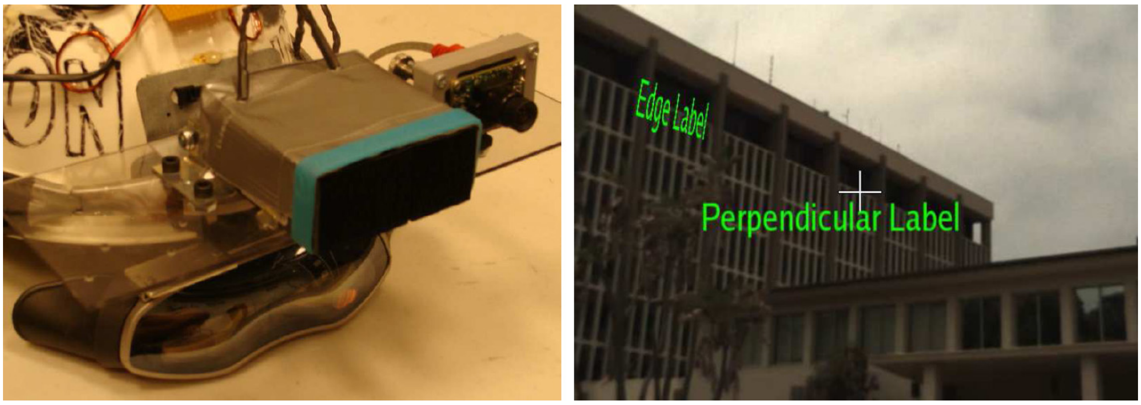


Figure 2.7: Placing annotations using a laser rangefinder. (Left) The setup which is based on a laser range finder installed on a HMD. (Right) Textual annotations are placed and aligned to objects by swiping the laser over the surface to be annotated. Perpendicular annotations are supported and are perpendicular to the initial viewing angle [156].

of assigning a depth value to annotations seen from a single viewpoint. Initial research showed that additional visual cues such as shadow planes, top-down view, or color-coded markers augmented in the user's view can support the user in assigning correct depth values [159]. The idea of using a top-down view was later refined by using aerial photographs to support the annotation process [157] [48]. After casting a ray in the direction of the object/position to be annotated in the AR view, a secondary view shows an aerial photograph, allowing the user to move the annotation along the ray. This process can be supported by applying computer vision to the aerial pictures to aid in identifying lines or geometric primitives.

Later Wither replaced this manual placement along a ray with a single-point laser rangefinder [156] to automatically calculate the depth information from a given position and orientation, which allowed better label placement (see Figure 2.7). A summary of this research can be found in [158].

The Envisor work by DiVerdi et al. [23] showed how textual annotation can be placed on a spherical representation of the environment and tracked using an orientation tracking algorithm. We demonstrated a more improved version of a similar idea [152], in which annotations are created and mapped to a panoramic representation of the environment. While we initially proposed to re-detect the annotations using template-matching [68], we later refined the re-detecting of the annotations using sensor fusion and global optimizations [66]. Chapter 5 will give a more detailed overview of this particular research.

With their work in *Indirect AR*, Wither et al. proposed another extension that not only uses the panoramic map for detecting and tracking textual annotations, but replaced the camera image with a current view into a panorama that also contained the textual annotations [160]. They demonstrated that using this approach, they trade the benefits of having a live camera view against a stable augmentation that is precisely registered into the panorama, as only the current view is affected by tracking errors but not the registration of the content in the user view.

2.3.2 Creating and Using Video Augmentations in Augmented Reality

The availability of inexpensive mobile video recorders and the integration of high quality video recording capabilities into smartphones have tremendously increased the number of videos being created and shared online. With more than 50 hours of video uploaded every minute on YouTube and billions of videos viewed each day[19] [20], new ways to search, browse and experience video content are highly relevant. Furthermore the availability of this community-created video data shows the high interest in this form of media.

The possibility to add meta-data, such as GPS position data, to images and later to videos allowed for the exploitation of this data by creating novel ways of browsing this content. Browsing images using their embedded GPS position was presented already by Kang et al. [51]. Later, this idea was picked up by image hosting platforms such as Flickr[21]. Similarly, online video hosting platforms started replicating these existing photo interfaces for video data and nowadays these features are also used in virtual globe application such as Google Earth[22] (or other map-based applications) to browse images and videos from Youtube or Panoramio[23].

More recently, efforts have been made to further explore the spatio-temporal aspect of videos. Applications such as Photo Tourism [132] inspired the work of Ballan et al. [7], which allows end-users to experience multi-viewpoint events recorded by multiple cameras. The system developed by Ballan et al. allows a smooth transition between multiple camera viewpoints and offers a flexible way to browse and create video montages captured from multiple perspectives. However, these systems require video footage not only from the same scene but also from the same time, which is not always feasible in practice.

These systems limit themselves to producing and exploring video content on desktop user interfaces (e.g., web, virtual globe), for the most part outside of the real context. Location-based video sharing, as for example realized by Vidcinity [24], extends the interface to explore geo-referenced videos on location, but does not overcome the problem of presenting the video tightly integrated into the environment.

Augmented Reality (AR) technology can overcome this issue, providing a way to place geo-referenced video content on a live, spatially registered view of the real world. For example, Höllerer et al., [46] investigated situated documentaries and showed how to incorporate video information into a wearable AR system to realize complex narratives in an outdoor environment. Recent commercial AR browsers such as Layar or Wikitude are now integrating this feature, and can support video files or image sequences, but still suffer from limited spatial registration due to the fact that the video is always screen-aligned and registered using GPS and other sensors.

Video augmentation has also been explored for publishing media. Red Bull[25] for ex-

[19] http://youtube-global.blogspot.co.at/2009/10/y000000000utube.html

[20] http://www.youtube.com/t/press_statistics

[21] www.flickr.com

[22] http://www.google.de/earth/

[23] http://www.panoramio.com

[24] http://www.vidcinity.com

[25] http://www.redbullusa.com

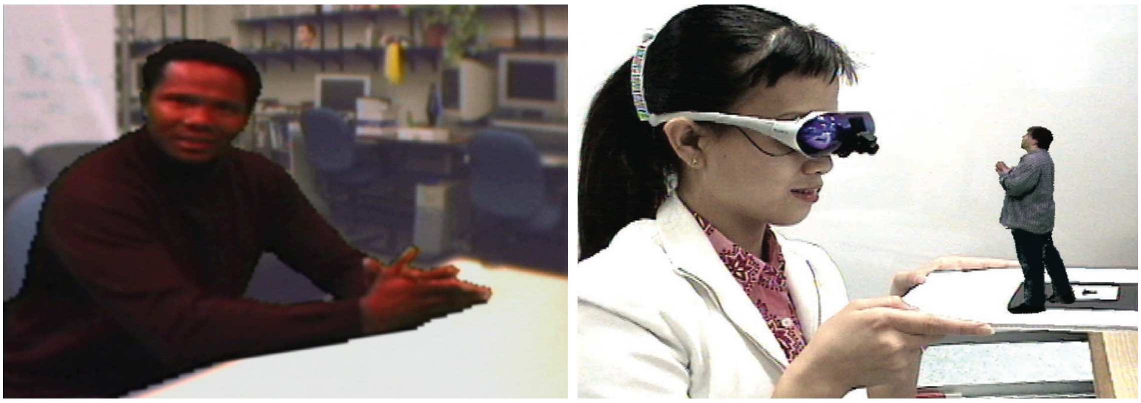


Figure 2.8: Examples of different techniques for integrating video in AR applications. (Left) A 3D video actor integrated in a 3D AR system [76]. (Right) A reconstructed 3D video actor augmented on a marker as part of a video conferencing system [105].

ample, presented an AR application that augmented pages of their *Red Bulletin* magazine with video material using Natural Feature Tracking (NFT). The application ran within a webpage as an Adobe Flash application, detected a magazine page and played the video content spatially overlaid on top of that page.

These projects generally present video on a 2D billboard type of representation, however other works have begun exploring how to provide more seamless mixing between video content and a live video view. MacIntyre et al. investigated within their Three Angry Men project the use of video information as an element for expressing narratives in Augmented Reality [76]. They proposed a system where a user wearing a head mounted display could see overlay video actors virtually seated while discussing around a real table (see Figure 2.8). The augmented video actors were pre-recorded and foreground-background segmentation was applied with their desktop authoring tool to guarantee a seamless integration into the environment [79] [**?**].

Whereas MacIntyre et al. used static camera recording of actors, the 3D Live [105] system extended this concept to 3D video. Prince et al. used a cylindrical multi-camera capture system, which allowed for capture and real-time replay of a 3D model of a person using a shape-from-silhouette approach. Their system supported remote viewing by transmitting the 3D model via a network and displaying the generated 3D video onto an AR setup at a remote location as part of a teleconference system (see Figure 2.8).

While these applications were proposed for indoor scenarios, Farrago[26], an application for mobile phones, proposed video mixing with 3D graphical content for outdoor environments. This tool records videos that can be edited afterwards by manually adjusting the position of virtual 3D objects overlaid on the video image, but requires the usage of 2D markers or face tracking. Once the video is re-rendered with the overlay, it can be shared with other users.

[26] http://www.farragoapp.com

2.3.3 Creating and Using Audio in Augmented Reality

Usually the term Augmented Reality is used to describe systems that visually augment the environment. However, there are also other modalities that can be augmented with digital information such as the haptic [97] or audition modalities [9]. While the former requires specific hardware, the latter can already be employed on today's off-the-shelf mobile hardware. In the following, we give an overview of existing work in the field of audio augmentations.

To date, most systems using audio augmentations involve two main approaches: (1) those that focus purely on audio augmentations to provide additional information, and (2) those that combine audio augmentations with visual augmentations. Especially for the latter, there are only a few works.

Similar to the early visual AR prototypes, early works in Audio AR relied on custom hardware to achieve the results. Bederson et al. [9] first introduced the term Audio Augmented Reality when they presented their museum tour prototype. The prototype was based on an mobile device (a Sony MD player with attached sensors) that automatically detected the position of users based on infrared emitters placed in the museum. Using the infrared-based anchoring, the MD player provided audio information that corresponded to the object being observed.

Similar projects such as Audio Aura [92] used active badges for tracking the position of individuals in office environments. Based on the position of the active badges, audio information was played about the corresponding people when a visitor entered the office. These early prototypes showed that for many audio AR applications a less accurate anchoring and tracking approach is sufficient to achieve convincing augmentation.

With the rise of mobile devices, Augmented Audio techniques were increasingly explored in outdoor environments. Usually GPS position is used for anchoring, while tracking relies on GPS for position and compass with accelerometers for orientation. Developing Audio AR applications is a focus of research especially for supporting visually-impaired people. Many application scenarios aim at supporting blind people with additional audio cues to assist their navigation in outdoor environments. Loomis et al. presented one of the first location-based systems that relied on GPS [74]. It was used to guide visually impaired people using GPS, GIS, and VR technologies.

Rozier et al. presented HearThere [119] an applications which plays audio information according to current GPS positions [119]. The overall setup was very similar to the Touring Machine [28], but allowed for editing of the audio objects in-situ using a map-based interface.

Audio augmentations were also used as a way to implement spatial narratives for location-based mixed reality presented by Dow et al. [25]. Their studies in the Oakland cemetery used an audio narrator to give historical information on selected graves and revealed that users acknowledged strong emotional experience.

Audio Nomad, another system in outdoor Audio AR was used for "seeding cities with sound" [161]. The system was based on static components like speakers that are combined with a HiFi system and GPS and placed on boats for harbour tours, but also on mobile clients such as mobile phones to play audio information. The audio information was a mix of oral histories, archival audio, site-specific historical information, field recordings, and music [161].

Similarly, McGookin et al. with their PULSE system aimed at an approach for an auditory display of geo-tagged social messages [83]. They used a text-to-speech engine to play messages for instance from Facebook or Twitter. McGooking et al. also discussed implicit and explicit notifications on the number of messages, their density in time, and their agreement.

Often audio augmentations are used for navigation and guiding. Stahl et al. presented the Roaring Navigator to guide tourist groups within a zoo environment by playing sounds indicating landmarks [134]. Similarly, McGookin et al. created Audio Bubbles, a system which uses augmented audio to assist in wayfinding for tourists [84]. The direction and distance to a Point Of Interest (POI) are mapped to a sound that surrounds the point of interest and assists in navigation towards that POI. Evaluation showed that Audio Bubbles outperformed maps and resulted a reduction in the time it took to find points of interest.

A major problem in audio augmentation is the issue of overlapping sound sources. Vazquez-Alvarez's user studies showed that two sound sources playing simultaneously could be perceived separately, albeit at the cost of an increased mental workload [145]. This workload is intensified in dynamic environments, where users or sound sources are moving or where more sound sources are playing in parallel.

Based on this evaluation Vazquez-Alvarez et al. presented approaches that deal with several audio sources that overlap. The first approach used the idea of proximity driven activation zones [144] [146]. Moving towards an audio source requires passing an activation zone that is represented by a symbolic sound indicating that the user is close to the audio annotation. If the user comes closer to the audio zone the audio augmentation is then played.

Magnusson et al. presented a system for non-visual orientation and navigation [81]. Unlike many other existing Audio AR systems that require users to physically move towards a position to experience the audio augmentation, Magnus et al. proposed using the pointing metaphor so that by pointing the phone in certain directions, audio augmentations are played to indicate directions. While previous Audio AR systems seemed to suffer little from poor sensor-based tracking, Magnus et al. stated that the error-prone orientation estimate from compass and accelerometers was a major drawback in their research.

Additionally, non-visual navigation and situated audio augmentation were also used in the more musically directed "Audio Grafitti" by Settel et al. [127]. This project allowed for the use of audio as graffiti by linking it to GPS position or places on walls, with users tracked via infrared. Further application scenarios also include location-based games, where it was possible to show that the use of augmented audio can increase the degree of immersion experienced [101].

Usually, systems use either audio augmentations or visual augmentations. Approaches combining both systems are rare. Behringer et al. have implemented a system for device diagnostics, which uses augmented instructions together with a speech interface and audio comments that give further instructions [10].

Haller et al. [38] use combined markers with 3D sound sources in a predefined setup for a more intuitive perception of the 3D sound in an indoor application. However, no user experience studies have been reported as part of this work.

Sundareswaran et al. [136] have improved object localization by playing sounds at the position of the object concerned. They use a wearable setup similar to the one presented by Feiner et al. [28] and extend it with sound capabilities. After a learning phase, users were able to detect objects based on sound more reliably.

Another approach using a backpack AR system that combines visual and audition modalities was presented by Tano et al. and used for informal communication [138]. The system allowed to create visual graffiti or audio notes that are linked to place via GPS or objects using RFID.

Similarly, Rekimoto and Nagao [113] implemented a prototype system, NaviCam, which detects color codes in the real environment as well as speech commands and synthesizes spoken messages. However, the content to be played is pre-defined.

To the best of our knowledge, systems that precisely place user created audio augmentations have yet to be investigated. While [119] and [138] involve user created content, they do not evaluate users' feedback or analyze the perceived usefulness of spatial audio comments. Furthermore they are restricted to rather imprecise tracking. The Audio Graffiti work by Settel et al. [127] allowed for the creation of audio graffiti in place but was also limited by the low accuracy of the used tracking system used. Furthermore, while it was possible to create the audio in place, this feature was primarily used by the creator of the art installation while setting up the installation. Instead, users usually focussed on browsing.

None of these systems allow users to place audio comments precisely in a sticky notes manner nor do they allow for comments on smaller objects. In our work on Audio Stickies (see Chapter 6) we present a new approach that allows for precise placement and selection of audio augmentations. We achieve this by using a vision-based tracking system for increased precision in tracking users' movements combined with using visual hints which indicate the position of audio augmentations to support browsing and selection.

2.3.4 Creating and Using 3D Media in Augmented Reality

Using 3D augmentations has a long tradition within AR - even the first AR prototypes were demonstrated with 3D renderings [137]. While 3D augmentations are among the most common media forms in research, they are rather uncommon in AR browser applications [34]. This might be due to the fact that they need more time for creation. In the following, we give an overview of related work in the field of 3D media in Augmented Reality. Due to the wide variety of applications and use cases involving 3D media in AR, we will focus on the creation of 3D content for Augmented Reality.

Overall, there are three main categories for approaches creating 3D content in AR: (1) Approaches that create content using desktop interfaces and do not necessarily have to be in place. (2) Approaches working in-situ which allow users to model and interact with the environment, (3) Approaches that support users in the modelling process by determining scene knowledge on the fly and incorporating this knowledge into the modelling process.

The Designer's Augmented Reality Toolkit (DART) by MacIntyre et al. is probably the best known example within the first category. It is implemented as a plugin for the Macromedia Director authoring solution, but implements functionalities that are unique

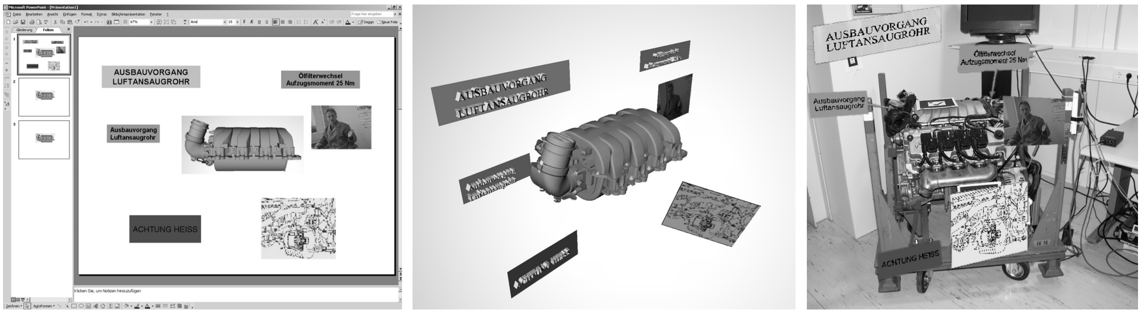


Figure 2.9: Steps for creating and layout content for AR using PowerSpace. (Left) An initial layout is made in 2D within MS Powerpoint. (Middle) The 2D layout is refined in 3D. (Right) The final scene seen in AR using the PowerSpace Viewer [41].

to AR [77]. It allows for the creation and placement of 3D content by using the existing function of Director, while also offering additional methods to specify the relationship with the real world (e.g., by incorporating markers).

While this work was intended for Macromedia Director, similar approaches exist for nearly all major modeling tools (including Google SketchUp [27]) which allows designers to continue using their known tools and workflows while creating content for AR.

A different approach was presented by Haringer and Regenbrecht. Their PowerSpace system created and positioned content first in 2D using Microsoft Powerpoint using a template for the physical object to be annotated later [41]. After finishing the initial 2D layout, the whole scene is exported in an XML format and read by the second editor within PowerSpace, which assists in aligning the objects in 3D based on the initial 2D layout (see Figure 2.9).

So far, the tools presented have not necessarily need to be used in place to work. However, working in-situ has some advantages. Lee et al. describe this process of authoring content in place as immersive authoring or: What You eXperience Is What You Get (WYXIWYG) [70], indicating that the modelling is done in the AR system within which the model will be used.

Within Tinmith Metro, Piekarski et al. demonstrated how to use Augmented Reality working planes - a set of infinite planes created from the user current perspective - to create existing geometry by intersecting the planes [102] [103]. This idea was later extended with more functions from 3D constructive solid geometry for constructing models within the Tinmith AR system [104].

A similar approach was presented by Baillot et al. to build complex geometry from points that are manually entered into the system and later used to create new points by intersecting rays with lines or surfaces. These sets of points can be further used to create lines, planes and 3D geometry by connecting several points to lines, combining lines to define a surface, and extruding a surface to create a 3D geometry [6]. Freeman et al. have implemented similar 3D modelling functions to model in AR with a free moving camera

[27] http://www.inglobetechnologies.com

[31].

One of the few examples of in-situ authoring conducted using a smartphone is presented in Henrysson et al., who built a simple scene using a smartphone as an interaction device to interact with augmented Lego blocks [44]. Later work showed how to manipulate an augmented mesh using a smartphone [42]. Similarly, the author of this thesis presented a work on generating 3D models through user-driven modelling on mobile phones [67]. In contrast to Henrysson et al., we presented a hybrid approach that works in indoor and outdoor environments, but does not need scene knowledge or existing markers in the scene (also see Chapter 6).

Creating virtual models of real world objects is a common goal when creating content for AR. These virtual models can be imported into a new scene or different context. Often the further aim is to precisely model the shape of real world objects in order to track position with respect to the real world object. In all these cases, vision-based methods can support the modelling process, as they reduce the number of parameters the user has to provide. In the following, we present several approaches that fall in this category of assisted content creation.

In 2007, Kim et al. presented a work [55] that is related to earlier work of Wither et al. [157] in that it combines aerial top-down views with the current camera view to model the scene. While Wither et al. focused on assigning depth values to textual annotations, Kim et al. aimed for a more complete 3D model of the environment. The modelling thereby also supported the lines in both camera image and aerial image to be detected using vision. Furthermore, the virtual duplicates of the physical objects can be used for feature-based tracking of the camera pose.

Similarly, Simon et al. presented an approach for modelling in an outdoor environment [130]. He identifies parallel lines and other information in the image to establish a coordinate system (see also *Single View Metrology* [22]). Based on the information of vanishing points and additional information given by the user, Simon et al. can determine a 3D model that can also be used for tracking the camera, while the user continues to model the environment.

Bastian et al. presented an approach that uses a set of camera images representing different views of the same object [8]. A user-assisted segmentations is then applied - supported by using the segmentation from the previous views - on all images to separate the object from the background. Using the extracted segments, a silhouette-based modelling approach can be applied to create a 3D model.

OutlinAR by Bunnun et al. is another approach for vision-assisted modelling. With OutlinAR, the user is able to build wireframe models by using a tracked handheld camera-mouse system - the wand - which is plugged into a standard computer [15]. Points are selected using this device and re-detected in different views using epipolar geometry. Using this approach, the user can create wireframe models, which can be used for tracking.

The initial research on OutlinAR was later extended to also run on smartphones [14]. In 2012, Bunnun et al. presented further research results gained from a comparison of running OutlinAR on two different devices: A touch-screen based device and a scroll-wheel wand [16]. This comparison also contained results for selecting points using both devices.

Many approaches for assisted modelling of physical objects use a variation of a SLAM

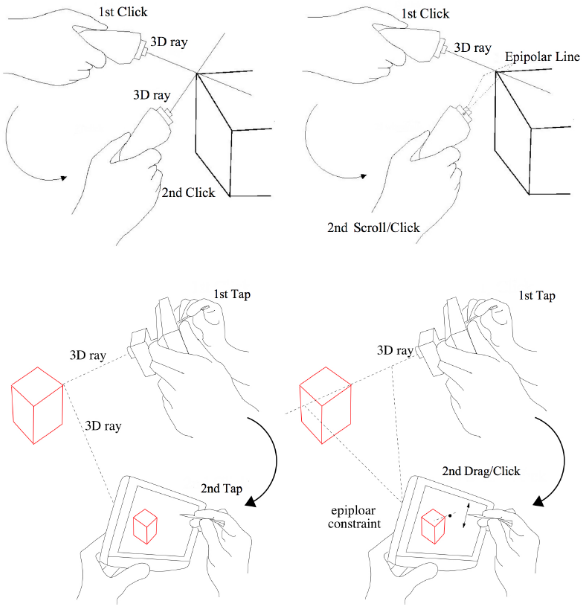


Figure 2.10: Pointing interactions using OutlinAR on two different devices. (Top) Selecting 3D features using the wand-like device. (Bottom) Selecting 3D features using the touch-screen device [16].

algorithm - most commonly PTAM [56] - to build a sparse map that is used for tracking. This map can then be made denser or at least used to support the modelling process by immediately having a tracked camera.

For instance, Hengel et al. [143] presented an AR extension of their previous work on VideoTrace [142]. VideoTrace uses real-time camera tracking, and high-level automated image analysis, for an immersive modelling process, which generates three-dimensional models of real objects, but relies on pre-recorded camera footage. Instead, the presented AR extension uses a live video feed and allows users to model in-situ rather than using existing videos in an offline step (see Figure 2.11).

Kim et al. showed a system that works similarly to the system of Hengel et al. However, instead of modelling a complete scene, it treats the created objects separately. This allows users to interact and move the physical objects once they are created, while still being able to track them [54].

Qi Pan presented ProFORMA, another system that allowed creating 3D models in an online manner [100]. To use the system a user has to rotate a physical object in front of a stationary camera, and 3D feature points are extracted using a structure from motion approach. These points are extended using Delaunay tetrahedralisation, and the resulting 3D mesh is refined using tetrahedron carving.

With Ninjas on a Plane, Chekhlov et al. presented another system that determines high-level scene knowledge using a SLAM system [18] by extracting physical planes from the sparse map created within the SLAM system.

Even a further step ahead is the system presented by Newcomb et al. [94] that creates

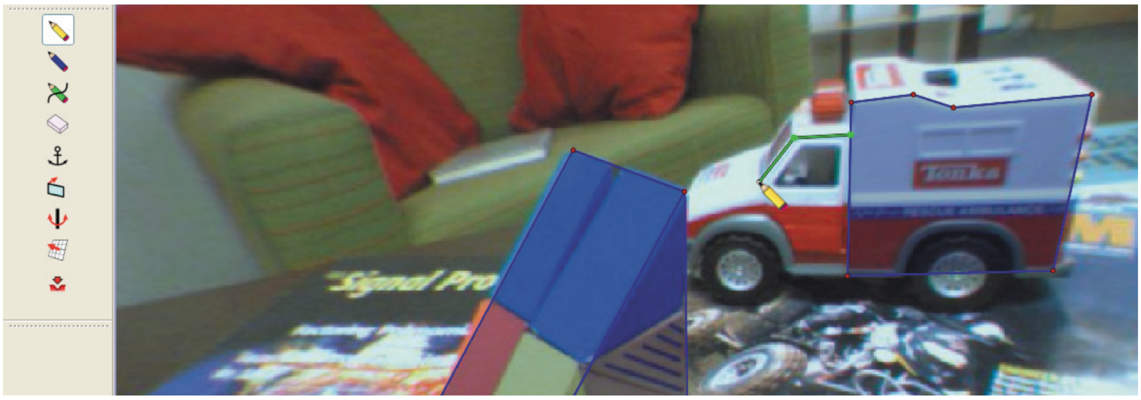


Figure 2.11: In-situ image-based modeling. The system allows automatic extrusion of selected image areas for creating 3D models of existing objects [143].

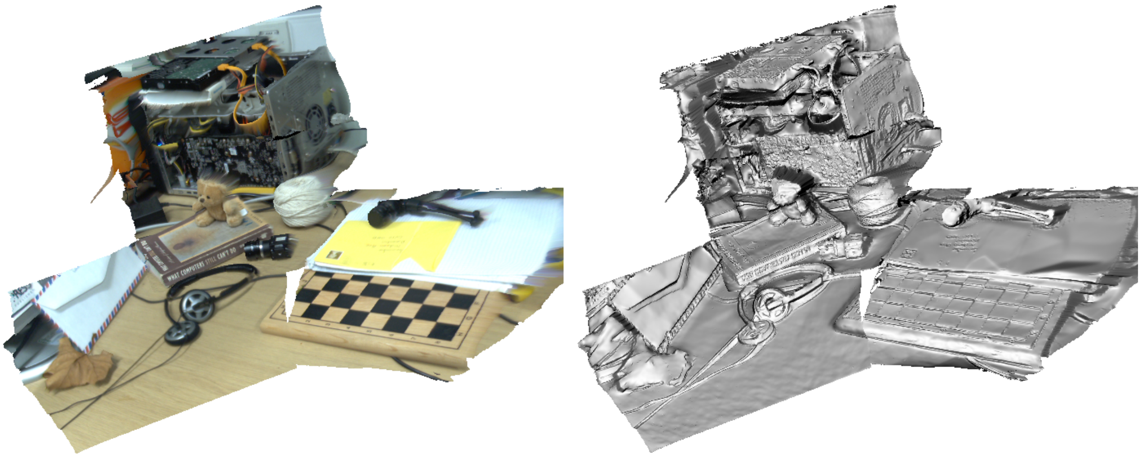


Figure 2.12: Partial reconstruction generated live from a single moving camera [94].

a dense reconstruction of the environment using a SLAM system at interactive frame-rates (see Figure 2.12).

By providing camera tracking or sparse (or in the case of Newcombe et al., even dense) scene knowledge, these systems assist the user in the modelling process. However, they have a major drawback: So far they only support the replication of existing physical models but do not provide assistance for the creation of 3D models that are not replicating the shape of existing models. In that case, the user must still rely on a manual modelling process.

2.4 Summary

In summary, while Handheld Augmented Reality and especially Mobile Augmented Reality already have a long track of research, longer than most end-users are aware of, there is not much research on how consumer oriented AR applications such as AR browsers are received by the public. This poses a major gap when identifying future research directions.

Furthermore, there is not much previous work allowing users to participate in the AR environment by creating and sharing digital items or informations. In particular, there is a lack of previous work on authoring content by end-users, nor is there much existing work successfully demonstrating authoring of content in place. The few existing applications demonstrating in-situ authoring target experienced users rather than novice end users.

We showed that depending on the type of media used, there is a different amount of existing work with very different characteristics. Textual and 2D annotations are for example commonly used in AR applications, but there is not much research in robustly and precisely tracking the annotations positions in outdoor environments (especially, if user-contributed 2D annotations are a further criteria). Similarly, audio annotations are so far only implemented using sensor-based tracking and rarely with user contributed audio information, thereby omitting use-cases that require precisely placed user-generated audio.

Video augmentations and their applications are so far not explored in outdoor AR, and there are only a small number of existing indoor applications using video information created in time-consuming offline steps. Finally, there are several works which allow for the modelling of three-dimensional content, but these approaches are usually aimed at professional users due to the experience required to use the necessary software or the custom hardware. The existing solutions for automatically reconstructing three-dimensional models, however, are limited to existing structures and cannot be used when the aim is to create new virtual content that does not resemble shape or structure of existing models.

Within the scope of this thesis, we want to investigate novel methods for authoring of multimedia content for AR applications aiming at inexperienced users. Our approach is horizontal in a way that we aim to cover several forms of media, as well as vertical, in a way that we also investigate fundamental problems such as an precise anchoring and tracking of the placed content.

Chapter 3

Towards Augmented Reality 2.0

Contents

In this chapter we present theoretical foundations for this thesis, investigating our hypothesis regarding usability of current generation AR browsers. In particular, we present a large study on usability and applicability of AR browsers in the field. In the second part of this chapter, we present our conceptual model of Augmented Reality 2.0 targeting next generation mobile AR applications, but especially next generation AR browsers. Our concept applies ideas of Web 2.0 to solve issues such as the availability of information, which we identified as one of the main issues in our study. The work presented in this chapter is not only important for our later research but can also be seen as a contribution beyond the scope of this thesis. This chapter is partially based on our technical report of Grubert et al. [34] and a revised version of our work presented by Schmalstieg et al. in [125] and Langlotz et al. [64].

3.1 A Study on Current Generation AR Browsers

As we showed in Chapter 2, the investigation of Augmented Reality as a new interface, together with its associated technical questions, already has a long history within research. Although first demonstrated in the 1960s, only recently have technologies emerged that can be used to easily deploy AR applications to many users. Camera-equipped cell phones with significant processing power and graphics abilities provide an inexpensive and versatile platform for AR applications. The concept of having an AR-based application to access and display digital information in users' environments has been explored within the research community for over 10 years. Steve Feiner's Touring Machine created within the MARS project [28], Spohrer's Worldboard [133] and the Real-World Wide Web Browser by Kooper and MacIntyre [60] are just earlier prototypes of what we nowadays call an AR browser.

Whereas the term *AR browser* was initially introduced by SPRXMobile for Layar in 2009, nowadays it is often used to describe a type of application. AR browser has come to signify a generic Augmented Reality application displaying geo-located multimedia content using a virtual representation augmented on the vision of the real world (i.e., a camera-image in the context of smartphone technology). AR browsers generally access remote resources through web protocols and services (e.g., HTTP, REST), index the content through media streams (termed *channels*, *layers* or *worlds*) and support a variety of MIME formats (usually HTML, text, image or 3D models).

Driven by the technical capabilities of current generation smartphones companies such as Wikitude, Layar, Metaio (Junaio)[1] or Acrossair[2] transported the ideas from the early AR browser prototypes presented in research to commercial applications and consequently made them accessible for end users.

Accelerated by the creation of mobile distribution platforms (e.g., iOS App Store or Google Play, formerly known as Android Market), which changed the way mobile applications are delivered to users, the awareness of this technology is spreading rapidly in the mind of the public. To date, AR browser applications have been downloaded approximately fifty million times, making them the most successful type of AR applications in terms of user awareness.

Despite the success of AR browsers which is reflected in the number of downloads and the fact the some AR browsers were factory installed on smartphones from leading brands, the usability and applicability of AR browsers has never been thoroughly analysed. So far, existing studies have been generally limited to the testing of some components and features (previously developed by academic research), in the context of lab-controlled studies [60] but their long-term use, the general applicability and usefulness remains mostly unknown (see Chapter 2).

In the following pages, we report on the results we obtained from a study investigating the usability and applicability of commercially available AR browsers. The study is based on two parts: Firstly, we report on an online survey of AR browsers that was actively advertised to users of AR browsers. Secondly, we also looked at the evolution and adoption of the technology that can be quantified from mobile distribution platforms, such as Google Play or Apple App Store, where AR browser applications can be downloaded, rated and commented on.

Thereby, we investigate and answer questions about the general usability and applicability of AR browsers in the field. In particular, we hypothesized that *content quantity and quality mark a major bottleneck in current adoption of mobile AR systems* and that *the majority of the consumed information is of simple form (e.g., textual information)* with both limiting usability. We also hypothesized that the *majority of current users are early adopters or technology-interested people*, who want to give the technology a try, without necessarily having a long-term interest.

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

[2] http://www.acrossair.com

3.1.1 Online Survey

In the following, we present the design and results of an online survey we conducted from May to July 2011 to gather user feedback on current generation AR browsers.

3.1.1.1 Method

For this exploratory study we decided to apply an online survey to gather feedback from users of AR browsers. We decided to use an online survey to increase the amount of people participating in the survey, as we were explicitly looking for people who tried an AR browser at least once.

We actively advertised our online survey through social networks such as Facebook, LinkedIn, Twitter and AR-related discussion boards and AR-related mailing lists. We also contacted the AR browser vendors and some of them advertised our survey in their blogs. We explicitly asked for users had tried an AR browser at least once. In total, 77 participants (14 female, 63 male) fully completed the survey, 118 partially answered the questions. For correctness, we only report on the results gathered from the completed responses.

Participants were informed about the purpose of the study and the approximate time needed to complete the survey (10 minutes). The data was collected completely anonymously; no incentives for taking part in the survey were offered. Participants were asked to answer 28 questions separated into three question groups (namely user background, type and applications, and benefits and drawbacks).

This section only reports on the questions and results that are relevant for this thesis, while the original survey was reporting on a wider field. The complete online survey can be found in the attachment of this thesis (see Appendix A) while a full discussion of all questions and results, including those that are outside the scope of this thesis, can be found in the report of Grubert et al. [34], on which this chapter relies.

The survey was created with LimeSurvey[3]. Statistical tests were conducted with R[4]. Coding of qualitative data was done in Nvivo 9[5] and Microsoft Excel.

3.1.1.2 Results

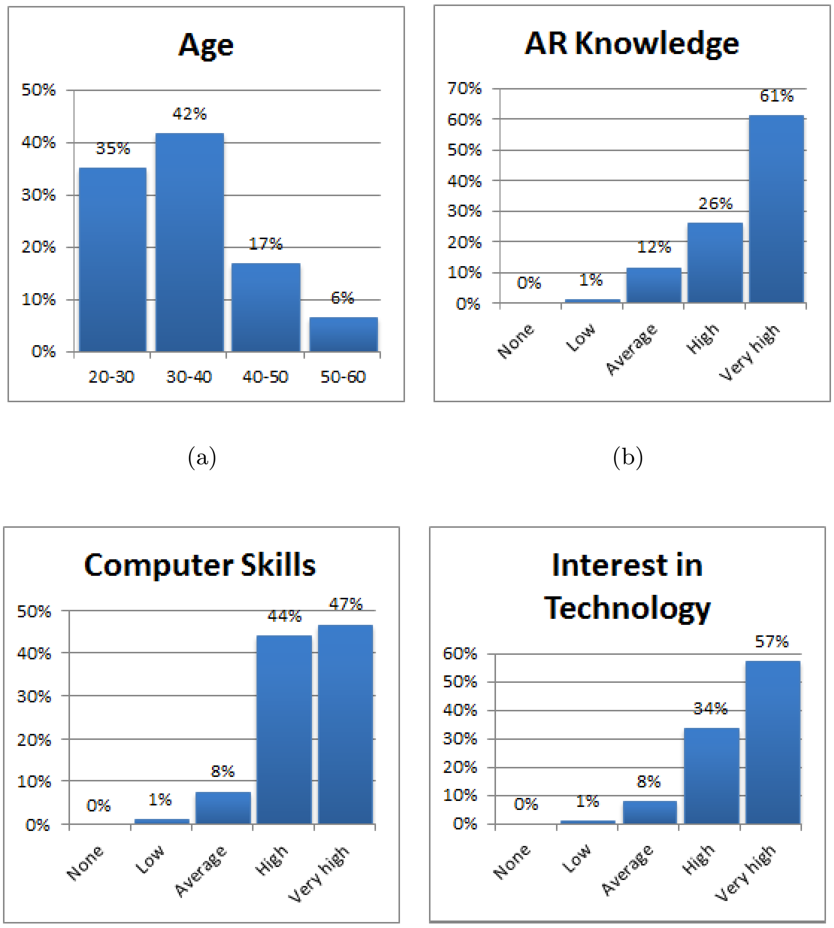
The survey was separated into several parts. The first part contained questions about the participants' background, helping us to categorize the users into groups such as novice user or technical expert. The obtained results were later used for analysing the given answers depending on the users' expertise. Later parts of the survey contained questions on usage behaviour, usage scenario, consumed media, feature quality, movement patterns, and reasons for discontinuing using AR browsers. In the following, we present selected questions and results of our survey that are relevant for this thesis.

Demographics The answers of the participants regarding their background indicate that most participants can be seen as tech-savvy people or early adopters of AR browser tech-

[3] http://www.limesurvey.org

[4] http://www.r-project.org

[5] http://www.qsrinternational.com/products_nvivo.aspx



(a) (b)

(c) (d)

Figure 3.1: Overview of participants' ages (a), knowledge of Augmented Reality technology (b), computer skills (c), and interest in technology (d).

nology. This could be caused by the recruitment channels used which included professional networks as well as the social networks of the authors. Most participants were aged between 20 and 40 years (Figure 3.1(a)) and show high computer literacy and interest in technology (see Figure 3.1). The professional status of the participants was gathered using an open form. We clustered the results in categories that also indicated the high technical profile of the participants, who were primarily from IT, engineering, Academic or even AR experts.

Application Background While writing this thesis, there are approximately 20 actively maintained AR browsers available in the mobile app stores. Three of them were noted as the most popular amongst the participants: Layar, Junaio and Wikitude (see Figure 3.2)and were all tried out by more than 50% of the people (multiple answers possible).

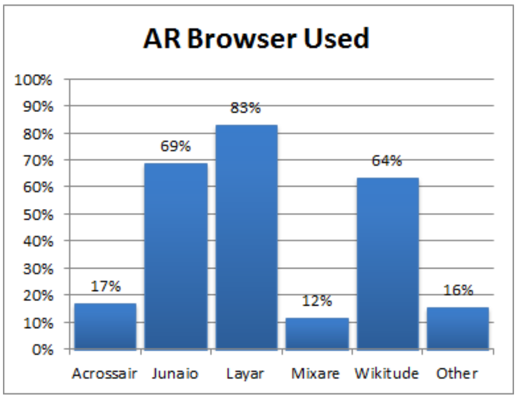
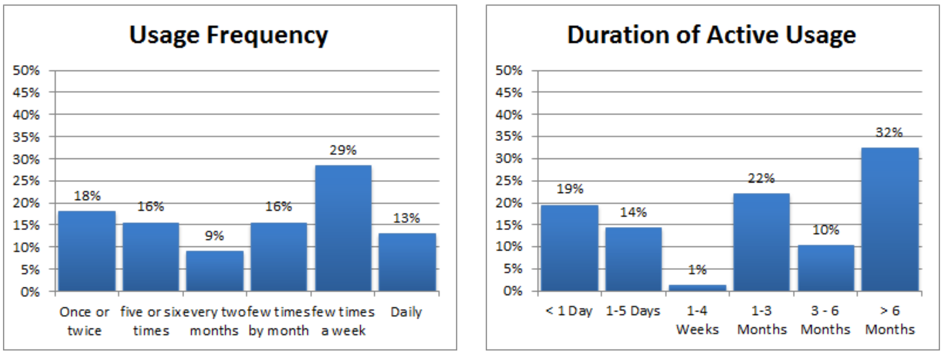


Figure 3.2: AR browsers used by participants.

SekaiCam, usually also being among the most popular AR browsers in terms of downloads, was under-represented. This is likely caused by the fact that it is mainly used in East Asian countries such as South Korea and Japan which are also likely under-represented within the participants of our study. The browsers were mainly used on iOS (54%) and Android devices (42%) ,with only a few using other platforms.

Usage Time For investigating the usage of AR browsers in the field, we were interested in the average duration of an AR browser session. The survey showed that typically an AR browser was used between one to five minutes. On the one hand, roughly a third of the participants (34%) tried out the browsers only a few times. On the other hand, 42% used the browsers at least on a weekly basis (see Figure 3.3(a)). The period of active usage was also split into two groups with a third of the participants (33%) using the browsers only for a few days and a third (32%) using them for at least half a year (see Figure 3.3(a)).



(a) (b)

Figure 3.3: Usage frequency (a) and duration of active usage (b).

We applied a Mann-Whitney U test to find differences in the usage time dependent on

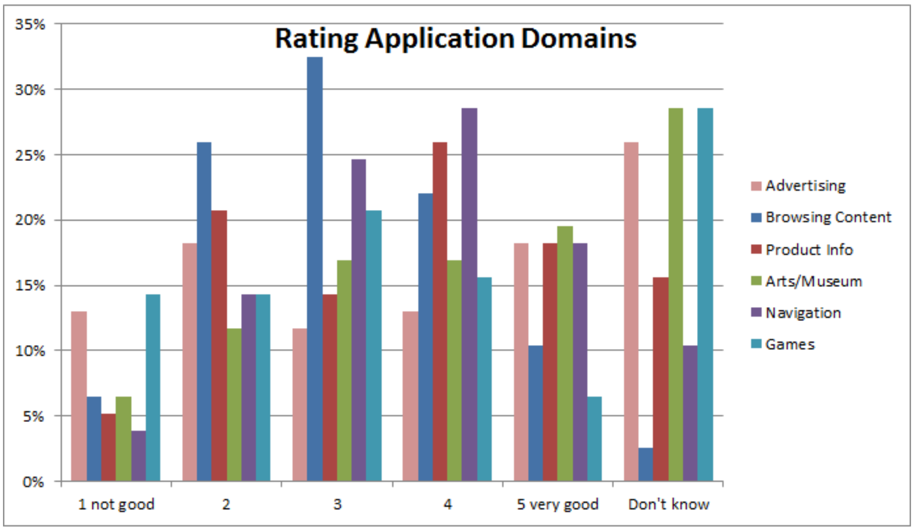


Figure 3.4: Rating of performance of current AR browsers for application domains.

the given expertise of the users. The test indicated that professional AR users (AR knowledge: very high, $n = 47, 61\%$) used AR browsers significantly more frequently (Mdn="few times a week") than novel users (AR knowledge low to high, $n = 30, 39\%$) (Mdn="5-6 times", "every two months"), $U = 924.5, p = .01$. This test also indicated that professional AR users use AR browser significantly longer (Mdn="3-6 Months") than novel users (Mdn="1-3 Months"), U = 924.5, p = .01.

These results give first evidence of our initial hypothesis that the majority of current users are early adopters or tech-savvy people with a high technical expertise. Furthermore, we showed that the general duration of active usage was rather short (if you take into consideration that AR browsers had been on the market for nearly four years by the time the survey was conducted) and was even shorter among users with less technical expertise.

Usage Scenarios Participants of our survey used AR browsers most often for general purpose browsing and navigation. Among all replies, 31% of the respondents also used the browsers for gaming, 39% in museum settings. The browsers were used outdoors by most (91%) and indoors by half (51%) of the participants (multiple answers possible).

Half of the responders rated browsers good to very good for accessing product information (44%) or guidance (47%), a third for browsing content (32%), advertising (31%) or museums (29%), but only 22% for gaming. However, a quarter to a third of the participants were still uncertain of their quality for advertising (26%), museums (29%), and games (29%). This might be explained by the relatively low number of participants who used AR browsers in these settings (see Figure 3.4).

Consumed Media Most participants viewed points of interests in textual form (77%), followed by 51% who viewed images, and 43% of the users who viewed 3D content. More complex web content (such as embedded webpages) and videos were experienced by only a third (27%) (see Figure 3.5). In summary, there is a strong trend towards simple forms

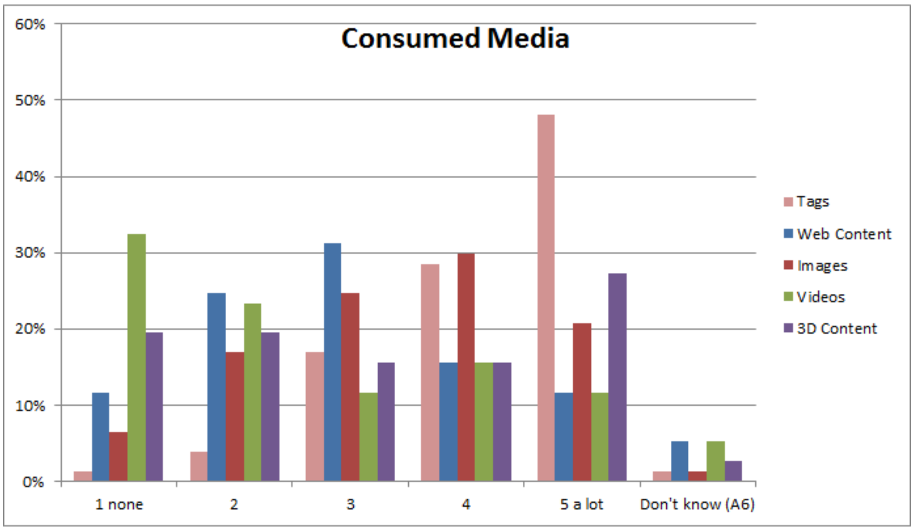


Figure 3.5: Type of media consumed with current AR browsers.

of media such as textual tags or pictures, which is in line with our stated hypothesis. The only exception here is the usage of 3D content. This could be explained by the technical background of the participants, many of them AR researchers, and the fact that 3D content traditionally has major significance in AR systems. Additionally, some AR browsers (e.g., Junaio) ship with a demo scenario containing 3D content. Unfortunately, due to the fact the many of our participants tried several AR browsers (e.g., most users of Junaio also tried Layar), we could not identify if the usage of 3D content was specific to one particular AR browser.

Feature Quality and Issue Frequency We further asked the participants several more explorative questions to identify general flaws in AR browsers. We therefore identified core areas we thought to be relevant to the usability and applicability of AR browsers. In particular, we asked the participants several questions about key features such as tracking (position accuracy, position stability), interface (design, content representation, clutter), performance (network, battery, general speed), displayed content (quality and quantity) and form factor (handiness and weight).

For the above mentioned features (except for device handiness and weight, which have a high rating with low issue frequency), low to modest ratings go along with modest to frequent experiences of issues.

A Kendall's τ test revealed moderate negative correlation between rating of feature quality and frequency of experienced issues for position accuracy, position stability.

A one-tailed Mann-Whitney U test indicated that professional AR users rated content representation significantly lower (Mdn=3) than novel users (Mdn=3, 4), $U = 511, p = .02$. The test also indicated that frequent users rated position stability significantly higher than non-frequent users (see Table 3.1), as well as content representation. Frequent users rated content quantity and content quality significantly higher and experienced issues with content quality less frequently than non-frequent users. In addition, issues with content

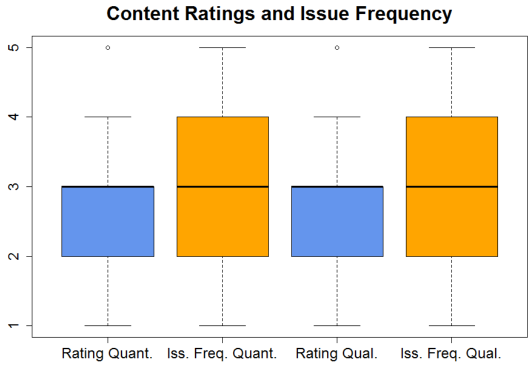


Figure 3.6: Content quality and quantity with ratings (blue) and issue frequency (orange).

quality did not appear as often for frequent users as they did for non-frequent users (Mdn=3 for both groups), U=538.5, p=.047. For other issues, no significant differences were detected.

Rating	n	Mdn f	Mdn nf	p-value	*U*
Position Stability	76	3	2,3	**.05**	854.5
Content Representation	76	3	3	**.01**	921
Content Quantity	75	3	2, 3	**.0026**	861.5
Content Quality	75	3	3	**.004**	925.5

Table 3.1: Significant differences in feature quality ratings for frequent (f) vs. non-frequent (nf) users according to Mann-Whitney U test. Interquartile range was two for all ratings.

Looking at the differences between frequent and non-frequent users, a one-tailed Mann-Whitney U test also indicated that long-term users rated position stability and content representation significantly higher than non-frequent users (see Table 3.2). For content quantity and content quality, there was only a weak significant difference. In addition, battery issues were experienced more frequent for long-term users ($Mdn = 4$) than for short-term users ($Mdn = 3$), $n = 70$, $U = 784.5$, $p = .018$, as well as device weight issues ($Mdn = 3$ for long-term, $Mdn = 2$ for short-term users), $U = 873.5$, $p = .023$.

Rating	n	Mdn lt	Mdn st	p-value	*U*
Position Stability	76	3	3	**.02**	897
Content Representation	76	4	3	**.008**	930
Content Quantity	75	3	3	.092	568.5
Content Quality	75	3	3	.07	556

Table 3.2: Differences in ratings in feature quality ratings for long-term (lt) vs. short-term (st) users according to Mann-Whitney U test. Interquartil range was two for all ratings.

Overall, the results indicate that for most key features of AR browsers professional or frequent users gave better ratings, which indicates, that based on their longer experience

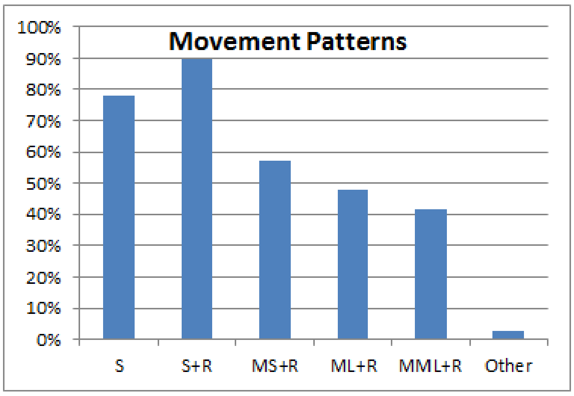


Figure 3.7: Movement patterns. S: standing. S+R: standing combined with rotation. MS+R: small (1-5m) movements combined with rotation. ML+R: larger movements ($> 5m$) combined with rotation. MML+R: multiple large movements ($> 5m$) combined with rotation.

with AR browsers they had a better chance of using an AR browser a couple of times with fewer or no issues. However, the general results are still in the average range for long-term users or below average in many cases for short-term users.

Movement Patterns With tracking being a major research direction in AR, and especially challenging in outdoor AR scenarios, we asked the participants of our study about the movement pattern when using AR browsers as the results could give valuable insights and help to align the tracking algorithms to the movement pattern. Most of the users were experiencing the application while standing at the same position (78%), combined with rotations (90%). Small movements ($< 5m$) were carried out by 57%. Large movements ($> 5m$) or multiple large movements were conducted by 48% respectively 42% (see Figure 3.7).

A Chi-squared independence test with Yate's continuity correction indicated significant differences between frequent and non-frequent users for standing combined with rotation, $\chi^2(1, n = 77) = 5.47, p = .02$ and multiple large movements ($> 5m$) combined with rotation $\chi^2(1, n = 77) = 5.94, p = .01$.

There was also a significant difference in multiple large movements ($> 5m$) combined with rotation between long-term and short-term users, $\chi^2(1, n = 77) = 10.05, p = .002$ and professional and novice AR users, $\chi^2(1, n = 77) = 5.55, p = .02$. Furthermore, between professional and novice AR users there were significant differences for small (1-5m) movements combined with rotation $\chi^2(1, n = 77) = 4.81, p = .03$ see, as well as a weak significant difference for larger movements ($> 5m$) combined with rotation $\chi^2(1, n = 77) = 3.35, p = .07$.

This analysis showed that while AR browsers were used by half of the participants also with large movements, frequent and long term users tend to restrict their movements more then non-frequent or short term users. While not fully proven by our survey, this could be seen as an indicator that the users adapt their movement pattern based on their experience, with better experience for rotational movements while keeping a fixed position.

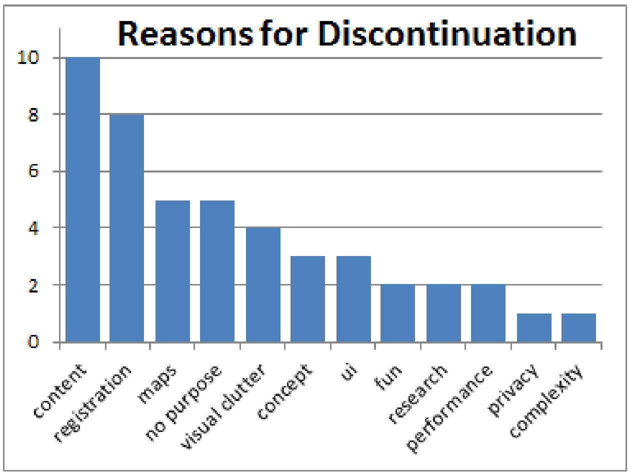


Figure 3.8: Reasons for discontinuation using AR browsers.

Qualitative Feedback Our initial hypothesis was that many users were short-term users or early adopters who briefly tried the applications instead of using them frequently over a longer period. Consequently, we were assuming that many people stopped using AR browsers, if not entirely then they stopped, or never started, to use them frequently. We therefore asked the participants of our study to provide reasons for withdrawing their usage of AR browsers if they did so; 31 (40%) of them provided free text answers. The answers were coded in a data-driven fashion [21] into 12 categories with 46 items. An overview about the reasons for discontinuation of AR browser usage can be seen in Figure 3.8.

The most mentioned category was related to content. Most of the free text answers focused on the general lack of interesting information ("Nothing interesting to see", "There's not much useful information"). The comments related to registration focused mainly on the imprecise registration of the virtual information with the physical environment ("Not useful as it was not spatially accurate") or even more specifically by saying that pure sensor-based tracking is not reliable, consequently reducing the usefulness ("It is not so reliable. Often the compass and the GPS do not work"). The comments regarding the registration issues were closely related to the group of answers that drew comparisons with existing map services such as Google Maps by saying that such map-based services are easier to handle ("I don't find it as convenient as just using something like Google Maps") or that AR browsers do not pose any advantages when compared to map-based services ("No advantage over Google Maps, less useful than Google Maps + internet recommendations for e.g. restaurants"). These comments are partially matched by feedback criticizing the general applicability and usefulness ("Not much real use-cases", "Generally I don't find them very worthwhile (to use privately)") and are clustered within the missing purpose group.

The last larger cluster of free text answers entirely focused on the visual clutter in terms of displayed information ("Too many POI one over the other") or the whole interface ("UI is always cluttered, information is not well structured").

In addition, subjects were asked to provide ideas for future features of AR browsers; 37 (48%) of them provided free text answers. The ideas for future features often matched the issues identified as reasons for discontinuing AR browser usage. In particular, the wishes for future features can again be clustered into content related future features ("Better designed content, more variety in regards to types of documents/files, more tools", "User-generated content", "Interesting stuff to see"), registration related features ("Need to find a way to calm down the jumpiness!!! Make it more exact", "Better location accuracy, robust POI display"), and visual clutter ("Well-arranged content, techniques for remove clutter", "More interactive, better filters").

Additionally, several wishes for future features can be clustered into interactivity, especially with the information presented in the AR browsers ("More interactive features (comments, rating, participating)", "3D interactivity").

Overall, the feedback given in form of free text answers matches the high technical profile of the survey participants and is uniform in the way that it targets previously identified issues such as content, tracking and registration, visual clutter and general applicability and usefulness, especially when compared to map-based services.

3.1.1.3 Summary

Our survey has mainly collected feedback from computer literate persons. Similar to other emerging technologies, like location-based services [86], users of AR browsers are early adopters, who have a high interest in technology.

On the one hand, a third of the participants used the AR browser just for a few days (five days or less: 33%) and less than six times (34%), indicating a large group of the participants merely tried out the browsers. On the other hand, 42% of the participants used AR browser for at least three months, and 42% at least weekly. Even though this might be considered a relatively short duration, especially when keeping in mind the fact that AR browsers had been on the market for over four years, it indicates that there is a regular crowd of users that use them for more than just 'trying out'. However, as our participants contain a large number of AR experts, we cannot exclude their professional interest in the technology being responsible for their usage frequency and duration, especially as the use frequency and duration were significantly shorter for novice users. Similar to the usage patterns of other mobile applications [12], AR browsers are typically used only for a few minutes per session.

Besides general purpose browsing, participants used AR browsers most frequently for navigation purposes. This could indicate that participants used the AR browser as an alternative to map-based navigation methods. However, the ratings of the current performance of AR browsers in domains such as browsing and navigation are rather mixed, while not many people have used AR browsers in other domains such as gaming, for exploring art or museums, or simply for product information.

Currently, the consumed content in AR browsers is mainly of simple form, such as textual tags (77%) or images (51%). Even if 3D content is available (as consumed by 43% of the participants), it is still mainly registered with 2 degrees of freedom (longitude, latitude). This can result in meaningless or cluttered overlay of content displayed on the screen. Our study results indicate that content and registration issues are a factor for discontinuing the use of AR browsers. But even frequent users rate the quantity or quality of available content only as average, which gives strong evidence of our initial hypothesis regarding content quantity and quality.

Another common issue with the use of AR browsers is the large power consumption that results in perceived issues with the battery life of mobile devices. Our study showed that registration, content and interaction with that content were among the most requested features for future versions of the AR browsers.

AR browsers were used by half of the participants also with large movements, but frequent and long-term users tend to restrict their movements more then non-frequent and short-term users. This is possibly due to long-term users adapting to the difficulties that arise from tracking inaccuracies and reading the information while moving. Previous studies investigated the reading performance of simple text while walking (e.g., [91, 123]) or automatic determined text readability over different backgrounds [72], but the impact of a changing camera image together with possibly jittering superimposed information while walking has not been investigated so far.

3.1.2 Mobile Distribution Platform Analysis

To complement our online survey, we analysed the customer feedback available from the two dominant mobile software distribution platforms: The Apple App Store and the Google Android Market. We looked at the ratings and user comments in both stores for some of the most popular AR browsers. As rating and commenting require users to authenticate, being limited to only one entry, this filtered information (no profanity, nominative) can provide some interesting insights into the popularity of these AR browser applications.

3.1.2.1 Method

To collect the data for the Apple App Store, we used the AppReviewsFinder software[6]. For Android Market, we used the data from the official Android Market homepage [7]. Data from both stores was gathered in June 2011 and represents the feedback given until then. Please note that the type and amount of information that can be retrieved from both distribution platforms is not symmetric. For example, one can access country-specific statistics for the Apple App Store, while there are no country-specific statistics available for the Google Android Market. It was also not possible to retrieve all user comments from the Android Market, limiting our analysis for this type of data to the Apple App store. Certain precise information is available to the developers of the software only (e.g., total numbers of downloads), and official information is only a rough indicator. We consequently decided to not evaluate some of this information. The presented numbers of downloads is also biased by the fact that some smartphone manufacturers have pre-installed certain AR browsers, and these copies are also included in the total number of downloads, despite the fact that users never explicitly downloaded them. We also restricted our analysis to solely focus on the current state of AR browsers on these distribution platforms at a specific

[6] http://www.massycat.co.uk/iphonedev/AppReviewsFinder/

[7] https://market.android.com

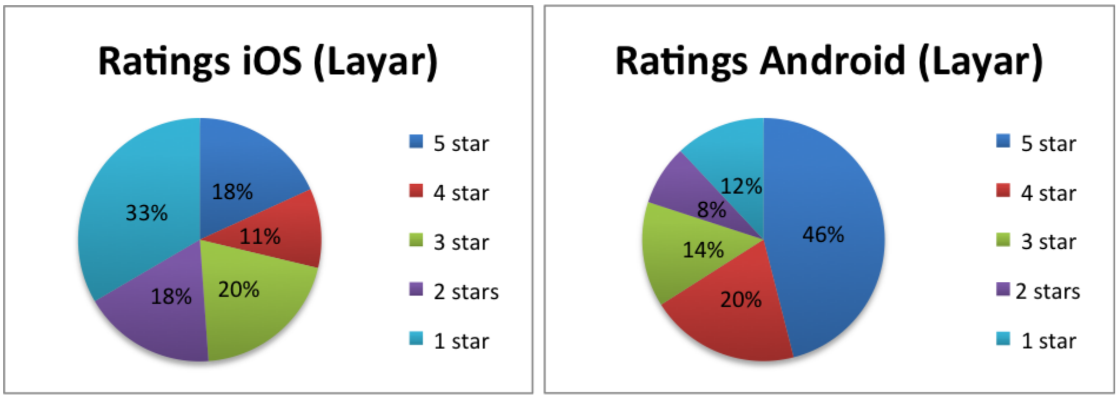


Figure 3.9: Difference of user ratings on both platforms based on Layar as example case (five stars are very good, while one star is very poor).

period of time, not considering the temporal aspect (e.g., trends over time for download, comments, adoption for specific countries).

3.1.2.2 Results

We describe here our review of the ratings in both distribution platforms and a deeper analysis of the comments in the Apple App Store for different AR browsers.

Ratings At the time of our study, we collected - for the different AR browsers - about 70.000 ratings for the App Store (multi-countries); about 30.000 ratings for the Android Market were available. Both mobile distribution platforms use a five star rating system (five stars being very good, while one star is very poor).

On the App Store, we identified five AR browsers that are prominent in terms of user-base and the number of countries in which they are available. Based on the numbers of ratings these are SekaiCam (27364 ratings), Layar (23385 ratings), Acrossair (9150 ratings), Wikitude (5443 ratings) and Junaio (3382 ratings). In contrast, there were only two AR browsers that achieved more than 1000 user ratings on the Android Market: Layar and Wikitude. For both, the number of ratings nearly matches those from the App Store.

The analysis of the gathered data showed that the average rating for all major AR browsers was very similar (overall average 2,49 stars) and also the differences in the average rating can be nearly ignored (Max: Layar 2,62 stars, Min: 2,39 stars Junaio). While examining the Android Market data, we observed that with the exception of SekaiCam all other applications rated significantly higher on the Android platform (average 3,65). This may be caused by stability problems on certain platforms or expectations that are platform dependent (see Figure 3.9). As an example, many iOS users have higher expectations regarding the implemented interface and the application quality, as both have so far been better for applications running on iOS.

The average rating is always the result of rather mixed ratings for all examined AR browsers as the standard deviation ranges from 1,38 (Wikitude) to 1,59 (Junaio), showing that many users gave very high or very low scores.

Based on the users' feedback in the Apple App Store it is also possible to analyse the difference in ratings between countries. In general, there is only a small deviation for all applications in the rating between the countries (Min. Layar SD = 0.38, Max. Junaio SD = 0.63). This is also reflected in the standard deviation of the ratings for each country, which are all nearly the same and showed that there are no significant effects that are based on cultural differences.

However, it is noticeable that for all countries with more than 100 ratings (to eliminate outliers), South Korea was always in the top group of top ratings, while France was always among the countries with the lowest average ratings. However, since the differences between the best and the worst ratings per country were only minor, this can only be seen as a weak trend. Furthermore, SekaiCam got on average lower ratings in German speaking countries (Germany and Austria), but again the difference was small (though noticeable) and could indicate content issues or poor localization. Based on the total number of ratings, most feedback came from users out from the USA, followed by Japan, UK, Germany, South Korea and France, with each application getting a relatively big number of ratings from the country of its origin (Acrossair/UK, Junaio/Germany, Layar/Netherlands, SekaiCam/Japan and Wikitude/Austria).

Comments We analysed 1135 comments from some of the most common western languages (English, German and French language) for all major AR browsers on the Apple App Store.

Analyzing the content of the comments, we categorized them in different groups. We filtered out the basic and rhetorical liking type of comments, and instead focused strongly on comments with a negative connotation or those that were arguing about a specific aspect of an AR browser. As a result, we obtained five major clusters (some with subgroups): applications crashes, content availability, user interface and visualization (contains comments about the graphical interface as well as the visualization of the content), tracking quality and general performance (contains comments regarding perceived performance, problems with network performance or comments regarding power consumption). An analysis regarding the occurrences in our dataset can be seen in Figure 3.10.

In the following, we present a deeper analysis of the clustered comments:

1. Application crashes: From the total 1135 comments, 225 comments contained complaints about regular crashes. This is by far the biggest category of complaints, which is also an indicator as to why the ratings were so mixed between one star and five stars. Most people with repeating crashes gave one star. User comments indicated that maintaining the proper software version for every new system or hardware update can be quite challenging.

2. Content availability: The second biggest category of complaints was regarding the availability of content. Many people expressed their disappointment with the amount and quality of available content. This ranges from no available content at all ("There were hardly any Points of Interest in Charlotte, NC") to very limited amount of content ("I looked for POI near me and all it came up with was a Post Box in the next street"). Furthermore, many users had certain expectations regarding content

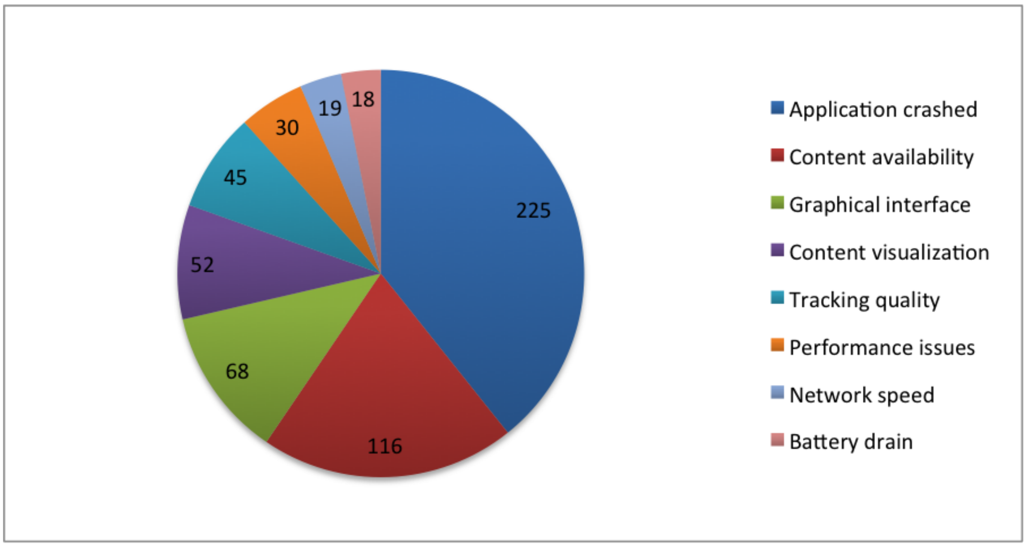


Figure 3.10: Result of clustering the total 1135 comments of the Apple App Store by focusing on negative connotations.

that were not fulfilled. Some users complained that the content was not up to date ("Then I tried supermarkets, and it found one non-existent supermarket in our town") or was not free.

3. User interface and visualization: Another problem that was raised in several comments was the quality of visual representation. Firstly, the graphical interface (menus and buttons) was highlighted several times as not very intuitive or not attractive enough as compared to other iOS apps. Furthermore, many people complained about the visualization of the displayed content (such as POIs), which can become unreadable if too many POIs are in close proximity ("It stacks up results until you need to point at the sky to read them"), or have generally low quality ("Can't wait until AR has real graphic experience").

4. Tracking quality: In their comments, some people addressed problems with positioning accuracy that are usually caused by a poor GPS signal or an inaccurate orientation estimate ("I played with this app near my hometown and it misidentified the location of our closest hospital - it was WAY off").

5. General performance: Only a few people had problems with the general performance or the speed of the network connections. However, some people suggested a caching mode. This would help users in foreign countries (e.g., tourists) to use the application even if they don't use an (expensive) 3G connection by pre-fetching and caching the results when a connection is available.

   To our surprise, only a small amount of users commented about the drain of battery caused by most AR browsers ("Tremendous drain on battery life. Actually causes my 3GS to heat up a lot", "But if it's gonna kill my battery, it has no place on my phone."). We think this originates from the fact that only a few people used an AR browser for a longer time, and consequently experienced such a sudden loss of battery power.

Beside the problems that were highlighted in the comments, many users also gave positive feedback that was often justified by the fact that most AR browsers are free to download. Many people also expressed their general interest in Ar browsers as they identified the potential of the technology. We often read comments stating that the current amount of content is small and there are still some bugs, but that the commenter will check back after some time as they think these applications have huge potential. This view is supported by positive ratings of the novel interface. Users stated how interesting it is, but only a very small number commented on how they made real use of AR browsers.

3.1.2.3 Summary

Overall, the data from the distribution platforms shows that the existing AR browsers perform similarly in term of user ratings. It also shows that there are no strong indicators for country specific or culturally specific effects with respect to the ratings. While the total number of ratings indicate that a large number of users at least tried AR browsers once, the real number of permanent users is still hard to estimate. The ratings suggest that the users' opinions are quite different; many gave a low score - and it is likely that they stopped using AR browsers - while another large group gave a high score. However, there is likely a novelty effect influencing the high scores of the second group. The comments also raised issues regarding the usefulness of the application, which in turn spotlighted the question of sustaining longterm use of AR browsers.

The comments from the Apple App Store show that the stability of AR browsers is one of the major issues that could be solved with better software quality management. Further problems are caused by the lack of content and the poor quality of the interface. Solving all these issues would resolve 75% of the users' complaints. A smaller group of users also pointed out problems with regards to content visualization and rapid battery drain. The importance of these two factors should not be underestimated. Particularly because the amount and density of the content will increase dramatically in the future. When end-users use AR browsers more frequently and more permanently, these problems may become a major issue.

3.1.3 Conclusion

We presented a first analysis of the adoption of Augmented Reality browsers by the public. Using two different evaluation tools - an online survey and an analysis of mobile distribution platforms - we reported on usage frequency, application scenarios, media consumed, subjective rating and general user comments.

During this analysis we identified similar patterns as an outcome of both tools, mainly with a population sample of technology enthusiasts. Firstly, a significant number of people tried AR browsers on their personal mobile devices and mostly responded positively on the technology; they also pointed out their interest in this type of application. Secondly, the tracking technology used - GPS for position, accelerometers and compass for orientation -

was not as much a limiting factor as we expected, especially concerning the feedback from the mobile distribution platforms. Thirdly, participants and end-users confirmed the high potential of this technology in the future, especially regarding some application areas such as content browsing and navigation.

The main issues, shown both in the survey and the mobile distribution platforms, were the scarcity of today's content on these platforms together with the quality of the displayed information. This is related to the simple form of the consumed information that showed up in the survey as well as in the comments from the distribution platforms.

The poor quality of the user interface and issues with battery life and performance (probably due to the tremendous energy consumption of the sensors involved in a standard AR browser) were among other frequently mentioned issues. From the analysis of the distribution platforms, comments indicated the lack of reliability and robustness of Augmented Reality browsers, which are also a common issue for other mobile applications.

The results of this study show that further research is needed to create a scalable approach for content authoring to create content for AR browsers of sufficient quantity and quality to fulfil the expectations of the users in terms of information richness as well.

3.2 Augmented Reality 2.0: Crowdsourced Content Creation for Augmented Reality

In this section, we describe a framework for overcoming these problems of current generation AR browsers by introducing the concept of *Augmented Reality 2.0* (AR 2.0): a combination of the terms *Augmented Reality* and *Web 2.0*. We describe the key ideas of the concept and its requirements for next generation Augmented Reality applications.

Like mobile phone AR, Web 2.0 is itself a recent development. O'Reilly mentions that the main difference between Web 1.0 and Web 2.0 technologies is that Web 2.0 enables end-user creation of web content, and thereby encourages social networking. Web 2.0 is characterized by open communication, decentralization of authority, and the freedom to share and re-use Web content. It is also driven by collaboration between users and provides a platform offering open APIs and applications that can be combined in sophisticated applications integrating information from multiple sources [99].

In contrast, original web technology was mainly used for one-way information retrieval. Only a few people made content, while most users accessed information without creating or modifying it. Web pages were mostly static and did not allow users to interact or provide additional information. This is very much the same situation for current AR browsers.

The advent of Web 2.0 substantially changed the way people use the Internet. Instead of only retrieving content, users are engaged in creating and modifying Web material. Two of the key innovations that can be supported through Web 2.0 are social networking and crowd-sourced content. Without revolutionary changes, the availability of the web has reached a point where the voluntary joint effort of literally millions of users can produce databases of a size and quality that has previously been considered impossible. For example, Wikipedia has already surpassed many traditional encyclopaedias in coverage and richness, and Flickr is one of the largest collections of digital images worldwide. Additionally, it was found that simple keyword tagging is powerful enough to replace sophisticated semantic web techniques as an organizational principle.

The open architecture of Web 2.0 services allows everybody to enrich these experiences with *Mashups* (information from different sources that can be combined to create a new value-added application), while advertising pays for the underlying infrastructure. It is important to note that all these results are based on previously existing technologies such as HTTP and Asynchronous JavaScript and XML (AJAX).

In a similar way, the goal of AR 2.0 is to provide widely deployable location-based mobile AR experiences that enhance creativity, collaboration, communication, and information sharing and rely on user-generated content. With an AR 2.0 platform a user should be able to move through the real world and see virtual overlays of related information appearing at locations of interest, and easily add own content.

Web 2.0 Characteristics	**AR 2.0 Characteristics**
Large number of users and web sites (already true for Web 1.0)	Large-scale in number of users as well as working volume
No clearly visible separation between accessing local data and remote data	No clearly visible separation between visualizing local data or remote data
Applications running in a browser behave like local applications, encouraging the user to interact with them	Applications locally running on the device can transparently download modules or new features from remote servers
A huge amount of non technical people retrieve data and contribute or modify it as well	Users can create or update the AR content at specific locations
Information from different sources can be combined and create a new value-added application, in so-called Mashups	Mashups access data from sources like traditional web services and combine them with AR content to display them in three-dimensional space

Table 3.3: Comparison of characteristics of Web 2.0 and AR 2.0.

This information overlay will be dynamically generated from a variety of sources and seamlessly fused together on the user's handheld display. In addition, users will be able to generate own location-specific virtual content that can then be uploaded to content servers and shared with others. Finally, the platform will provide support for social networking through synchronous and asynchronous context-sensitive data sharing. AR 2.0 as a user experience and networked medium has many parallel characteristics to Web 2.0 (see Table 3.3).

3.2.1 Requirements for Augmented Reality 2.0

If AR applications are going to be deployed on a massive scale in an AR 2.0 approach, there are several key areas of technology that are required:

1. A low-cost and mobile platform, combining a display, camera, sensors and processing power sufficient for tracking and rendering
2. Mobility to realize AR in a global space
3. Standardized content formats
4. Backend infrastructure for distribution of AR content and applications
5. Rich and interactive media accessible through AR
6. Authoring tools for creating AR content
7. Mobile interaction with the content
8. Large-scale AR tracking solutions which work in real time

In the remainder of this section, we briefly discuss these requirements for Augmented Reality 2.0.

A low-cost mobile AR platform To guarantee a critical mass of people participating in an AR 2.0 environment, many people need to have access to the technology. Similar to Web 2.0, where users need access to the information, the users within an AR 2.0 environment need an affordable platform capable of running AR applications. Fortunately, this platform is available: smartphones. As first shown by Daniel Wagner [149], smartphones are an affordable off-the-shelf platform for sophisticated AR applications. They combine a display, a camera, and various options for untethered networking combined with a processing power sufficient for tracking and rendering. In addition, they usually also house a set of sensors.

Usually, AR interfaces running on handheld devices such as smartphones or tablets are coined *Handheld Augmented Reality* [149] and are a sub-genre of *Mobile Augmented Reality* that also includes bulkier, but still mobile devices, such as back-pack AR systems. For simplicity, we will use both terms throughout this thesis interchangeably when talking about AR applications on mobile phones.

AR in a global space Another key requirement is to have AR working in a global space. Regarding the hardware, it means that it needs to be untethered, and the infrastructure for supporting global usage should be implemented. In addition, the software should support network connections to download content from remote sources in case they are not locally available. This also means that the software should be able to download additional modules and plugins on demand without additional stationary hardware. Fortunately, using smartphones as a platform for AR also primarily addresses these requirements. Their form factor allows a mobile usage that is usually only constrained by battery life. 3G communication infrastructure exists and allows, even in less developed countries, communication to remote servers. In addition, the sensors used such as GPS are designed to deliver pose information in global space.

Standardized content formats As we showed for Web 2.0, open standards for the commonly used content formats are essential for bridging the gap between different applications (e.g., different Web or AR browsers) as they allow re-using of the content from other applications domains (e.g., content created for the Web can be re-used within AR browser applications).

For an AR 2.0 environment, lessons should be learned from the rise of Web 2.0 that continuously fosters the development of open and standardized formats. Firstly, standardized formats, architectures and protocols should be developed for AR to describe the content structure. Standards organizations such as Khronos, X3D, OGC, and W3C[8] have already made an effort towards conceptualizing and defining AR, but this work is still in a preliminary phase. Some research groups have tried to establish formats such as ARML (Augmented Reality Markup Language) or KARML (an AR extension to KML) [78], yet there is no agreed-upon and widely implemented standard for describing content in AR. Secondly, the availability of accepted standards enables the combination of content from various sources (as demonstrated with Mashups in Web 2.0), forming a more valuable information product: the AR mashup. This would narrow the gap between the information on the Web and the content produced for AR browsers. However, we think that creating open standards can be guided by research, but ultimately needs to be picked up by the industry. This is still an open issue for AR related formats.

Content distribution for AR Besides open standards, an AR 2.0 environment needs a back-end infrastructure that supports hosting and sharing of content using open protocols and by also supporting different content sources and Mashups. The KHARMA architecture implemented in the Argon browser [78] is an early, but not yet commonly adopted, model for a possible open AR infrastructure, but requires further research.

Rich and interactive media accessible in AR The content accessible through current mobile AR applications, such as AR browsers, is still far from content accessible through standard Web browsers in term of richness. AR browsers are hindered by the basic nature of the information displayed: primarily text with a limited amount of pictures and even fewer 3D materials or audio-visual media. This is in sharp contrast to the current richness of the Web experiences in terms of media form, dynamics, design and quality.

End-user focused AR authoring One of the key aspects of Web 2.0 that needs to be transported into an AR 2.0 environment is end-user authoring of social media. Usually AR authoring is aimed at professional users. On the contrary, within the Web 2.0 there is a wide variety of options which allow users to create or integrate media content. Additionally, many of Web 2.0 applications target untrained users, offering interfaces that are simple to learn and use.

Mobile interaction with the content The Web showed that users not only browse content, but also interact with displayed information. In contrast, existing AR browser applications commonly focus on the browsing of information and offer few possibilities for

[8] www.w3.org/community/ar/

meaningful interaction. Future generations of AR applications, but especially AR 2.0 applications, need to integrate interaction. Allowing users to interact with the superimposed information should increase interest in AR, outside the scope of browsing.

Social networking for AR Social networks are an essential part of today's Web. They map real-world relationships between people onto virtual space, allowing users to sort contacts and friends in a novel way. This type of network can be used for communication but also for exchanging information. Nowadays, most bigger social or content platforms that gather people of similar interests (e.g., Flickr, YouTube) support the idea of contacts, friends or buddies and allow communication between the users and sharing content within groups. AR 2.0 applications should also reflect social network functions such as sharing of information or helping users to find people with similar interests.

Tracking for Augmented Reality in a global space Changes in tracking will severely affect the user experience. To precisely augment the environment with digital information, AR needs reliable tracking technology. Depending on the application, a wide range of different tracking methods has been developed. AR browsers currently support sensor-based tracking and, more recently and mostly indoors, vision-based tracking. Tracking in outdoor scenarios lacks the robustness and precision required for Augmented Reality [4]. Consequently, AR 2.0 applications need to improve tracking in large environments to support mobility.

3.2.2 Summary

We introduced the concept of Augmented Reality 2.0 that relies on ideas that were successfully in the Web 2.0 environment. We showed that user participation and crowdsourcing is a major option for scalable content creation. We further introduced several key aspects for the next iteration of Augmented Reality. In the remainder of the thesis, we will present approaches and prototypes that work towards the idea of Augmented Reality 2.0 by specifically addressing the problems of end-user authoring, interaction with the created content, and tracking in large spaces. We will demonstrate these outcomes with prototypes showing the usefulness of the proposed approaches.

3.3 Conclusion

Within this chapter, we presented results from an initial study on current generation AR browsers that combined results from an online survey with ratings and comments from the application distribution platforms. The results show that content quantity and quality are major issues identified by users. The poor quality of the displayed information is also caused by the simple form of information that is offered to the users.

Based on those main findings, we presented our concept of AR 2.0 that applies ideas from Web 2.0, such as social media, to Augmented Reality. Within the concept of AR 2.0, we also presented requirements that need to be fulfilled to support social media for mobile AR applications. While some of these requirements are at least partially fulfilled (e.g., smartphones as platform for AR, the ongoing work in standardization of AR content), others, especially tracking in outdoor environments and interfaces for creating social media in and for AR, are not available and represent a major gap in current AR research.

Chapter 4

Technical Foundations for Augmented Reality 2.0

Contents

In this chapter, we present technical foundations that are relevant for this thesis, and in general, for realizing the concept of Augmented Reality 2.0. These technical foundations are a brief overview of Studierstube ES - an AR framework specifically targeted at handheld AR devices, that was used, but also extended within the scope of this thesis - and a tracking approach called panoramic mapping and tracking that was created within this thesis and used for implementing prototypes described in the later sections of this thesis. Both the framework for AR, as well as the tracking approach, can be seen as fundamental technical enablers for Augmented Reality 2.0 that are necessary to create prototypes supporting the concept of AR 2.0. This chapter is based in parts on the system presented by Wagner et al. [150] [152] which was used and extended within this thesis and some parts were published by Langlotz et al. [66].

4.1 A Framework for Handheld Augmented Reality

Within the previous chapter (Chapter 3)) we named some of the key technologies that need to be provided to realize an AR 2.0 environment. Among others, these were the availability of a low-cost platform for deploying Augmented Reality. Fortunately, this platform is available in the form of smartphones or tablet devices. These devices combine a display with integrated sensors such as a camera and GPS, and also have enough computational power to run augmented reality applications. However, the key requirements for a low-cost AR platform are not fully satisfied by solely by addressing hardware requirements. A software framework that allows for the development of AR applications for these devices is also needed. Unfortunately, developing Augmented Reality applications is often a complex task because usually several components are involved: A tracking component for providing pose information to the application, a rendering component for creating the visual output, and at least one other component implementing the program logic. Additionally, in many cases the mentioned components are rely on more basic components (e.g., a tracking component often uses a computer vision component and a math component, while the rendering often has a renderer and a scenegraph). Moreover, all these components need to be optimized for handheld devices that have several constraints, in available CPU-power, battery lifetime or quality of sensor readings, and also in a wide range of different devices and, compared to desktop environments, a fragmented environment of different operating systems (e.g., Windows CE, Windows Phone, Symbian, Maemo, iOS, Android, Bada, Blackberry, WebOS, to name just a few examples that appeared within the time frame of this thesis).

Dedicated AR programming frameworks can make programming AR applications less cumbersome, as they abstract many commonly used functions. This allows one to gain independence of the used hardware or operating system. Additionally, one can encapsulate certain functions by only exposing a subset of the implementation, and consequently reducing the complexity. Some existing frameworks only focus on specific aspects such as tracking. An example is ARToolKit [53], which can be used to determine camera pose relative to physical markers, but still requires a lot of handwritten code for adding content, renderings and interactions. Consequently, other frameworks try to provide a nearly complete set of functions to create AR applications including scenegraph, rendering, support of input devices and window management. Typical examples of these frameworks include the MARS system [47] or the Studierstube framework [124].

However, most of these frameworks were not optimized for running on mobile phones, which results in poor performance. This was usually caused by a heavy CPU load (e.g., simulation of floating point numbers) and a huge memory footprint. The first frameworks that were available for the handheld devices were the Symbian port of the ARToolKit tracking library [43] or the ARToolKitPlus library [155] for pose tracking on smartphones. In the following we give a brief overview of the framework used, and further extended within the scope of this thesis.

In this thesis, we use Daniel Wagner's *Studierstube ES* framework [149] for implementing AR applications and realizing prototypes following the concept of AR 2.0. Unlike ARToolKit and ARToolKitPlus, Studierstube ES is a framework providing most functions for creating AR applications, such as basic vision and tracking, math, scenegraph and rendering and window handling. While sharing the name of the Studierstube framework, Studierstube ES was completely written from scratch and does not share any code with the Studierstube framework. Studierstubes ES was especially designed and optimized for handheld devices such as smartphones (ES = Embedded Systems).

Studierstube ES is mostly written in C/C++ and initially targeted Windows CE and Windows Mobile devices. It was also possible to create AR applications for desktop versions of Windows. However, over the duration and partially within the scope of this thesis the Studierstubes ES framework was extended to run on other platforms such as Symbian, Linux, Mac OS X, iOS and Android.

The initial release of Studierstube ES consisted of four modules [149]: ARToolkitPlus for detecting and tracking from 2D markers, Muddleware as a multi-user middleware, a

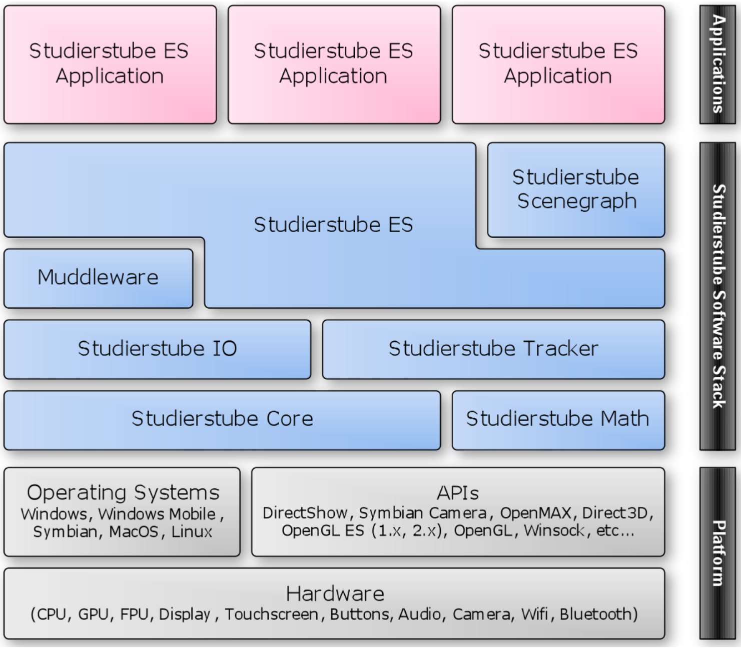

Studierstube ES Application
Studierstube ES Application
Studierstube ES Application
Applications
Studierstube ES
Studierstube Scenegraph
Muddleware
Studierstube IO
Studierstube Tracker
Studierstube Core
Studierstube Math
Studierstube Software Stack
Operating Systems
Windows, Windows Mobile , Symbian, MacOS, Linux
APIs
DirectShow, Symbian Camera, OpenMAX, Direct3D, OpenGL ES (1.x, 2.x), OpenGL, Winsock, etc...
Hardware
(CPU, GPU, FPU, Display , Touchscreen, Buttons, Audio, Camera, Wifi, Bluetooth)
Platform

Figure 4.1: High-level overview of Studierstube ES.

rendering abstraction layer (could be either OpenGL ES or Direct3D Mobile) and the core Studierstubes ES module. This Studierstube Core provides scenegraph, GUI, and networking, but also provided a mechanism to connect all other components, including a hardware abstraction layer.

Along with support for additional platforms, later releases of Studierstubes ES included a redesign of the framework, to which the author of this thesis contributed. The idea behind this redesign was not to change existing algorithms, but to extend the functionality and prepare the framework so that it can more easily ported to other platforms.

The new Studierstube ES consists of 7 modules (see Figure 4.1): Studierstube Core, Studierstube Math, Studierstube Tracker, Studierstube IO, Muddleware, Studierstube Scenegraph and finally Studierstube ES. An application developer creates her/his own Studierstube ES application by using classes, methods and functions provided by Studierstube ES and also accesses the other modules through Studierstube ES API calls.

In the following, we briefly review the core components of Studierstube ES, as it was used to create the prototypes described later in this thesis. We excluded Muddleware from this chapter, as it was not used within the scope of this thesis.

Studierstube Core The diversity of mobile platforms and devices usually requires many changes to optimize software for use on multiple devices. While high-level programming languages such as Java could solve the problem, they are usually to slow to run even basic AR programs at interactive frame-rates on mobile phones. When creating the Studierstube Core module, we wanted to create a module that contained all platform-dependent code, making it easier to port Studierstube ES to other platforms by just adding the new

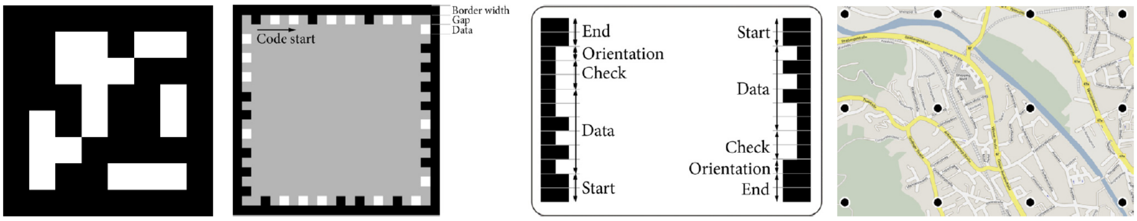


Figure 4.2: Marker designs trackable by Studierstube Tracker: regular black/white marker, layout of frame markers, split markers and dot markers (from left to right).

platform-specific code to Studierstube Core.

Studierstube Core abstracts basic types, especially various types for strings, numbers, dates and container data types. It further provides file access for various platforms, basic window handling and hardware abstraction layers. The latter is especially useful for accessing the camera, integrated sensors (e.g., GPS, Compass, Accelerometer) and audio devices for audio playback.

Studierstube Math The essential math functions are provided by Studierstube Math. Until recently, floating point operations on mobile phones were slow as they had no dedicated FPU. Therefore, two of the key functionalities provided by Studierstube Math were optimized floating point operations, as well as a fixed point implementation that could be used for time critical operations. It further provided methods for solving linear equations, which are often required for pose computations.

Studierstube Tracker Tracker Tracking is a fundamental element of each AR framework. In Studierstube ES, the Studierstube Tracker module is mainly responsible for providing this functionality. Studierstube Tracker is a tracker capable of tracking different markers including regular black/white markers called *simple ID markers*, markers storing the id in the frame called *frame markers*, markers called *split markers* that are optimized for holding them in the hand, and finally, *dot markers* that use a set of dots printed on an image whereby the image is used as a template and used together with the dots for tracking (see Figure 4.2). Besides markers, Studierstube Tracker can also track from DataMatrix barcodes and markers used in ARToolkit.

The main steps in the tracking pipeline of Studierstube Tracker are independent of the used markers. These steps are thresholding of the image, the detection of the fiducial within the thresholded image, marker detection and extraction of the encoded ID, corner filtering, pose estimation using the filtered corners, and pose filtering. For more details on how to track markers using Studierstube Tracker, we refer to the original publications to which the author of this thesis contributed [150] [151].

Studierstube Tracker is dependent on Studierstube Core and Studierstube Math, as it relies on functionalities provided by these modules. However, since no other external modules are needed, Studierstube Tracker can be built as a separate library and used without Studierstube ES in case one only wants to use the tracker. Furthermore, it is possible to use other trackers in addition to Studierstube Tracker, such as the tracker for natural feature-based tracking [153] or the panorama-based tracker presented in the next section.

Studierstube IO Using mobile phones as a platform for AR allows many new application scenarios. For instance, AR browsers, but also multi-user applications, use the mobile phone as a client to access the content stored on remote severs, for displaying it later in AR. Studierstube IO provides the functionality to communicate over a network with remote computers. This includes TCP/IP-based communication as well as UDP. Studierstube IO can provide streaming access to these data by letting remote data appear as local data so that the application does not need to differentiate between local and remote data.

In addition, Studierstube IO provides the functionality to open, read and write several commonly used file formats such as those for storing data in XML, images (JPEG, PNG, BMP, TIFF, etc.), or video data, just to name a few. Studierstube IO can be further used to mount zip files for reading and writing compressed or bundled data.

Studierstube Scenegraph The Studierstube Scenegraph provides a lightweight scenegraph to the programmer. The scenegraph is able to load, parse, and render graphic objects stored in a XML format. Studierstube Scenegraph encapsulates the graph structure in a way that it is easy to maintain on a mobile phone by avoiding a big memory footprint. The renderer of the scenegraph is written in a way that several different rendering APIs such as OpenGL and OpenGL ES can be used.

Studierstube ES Studierstube ES can be seen as the kernel for Studierstube ES applications, providing access to the other modules as well as functionality to start an application.

Besides basic kernel functionality, Studierstube ES contains a renderer that can be used with graphics APIs such as OpenGL, OpenGL ES, OpenGLES 2. The renderer's main job is to traverse the scenegraph. Studierstube ES offers dedicated render paths for software and hardware 3D accelerated devices. Furthermore, Studierstube ES implements functions for graphics user interfaces to draw and interact with buttons, text and other graphical elements.

If necessary, Studierstube ES can be configured at compile-time through XML configuration files to exclude certain modules from the build in case they are not needed (e.g., Muddleware can be excluded if a single-user application is created while Studierstube Scenegraph could be excluded in case another, or no scenegraph, should be used or 3D rendering is not needed). Furthermore, it is possible to strip down and configure certain modules to only contain needed functions. This helps to reduce the memory footprint of the final application.

Applications It is possible to build basic Studierstube ES applications without programming, by setting up a set of configuration files describing basic functionalities, e.g., linking certain 3D models to markers. However, accessing the full functionality of Studierstube

ES in order to run more sophisticated applications requires working with the APIs provided by Studierstube ES. Therefore, a programmer inherits his application from a set of classes provided by Studierstube ES. These classes abstract the main application logic and provide the rendering loop and certain events triggered by Studierstube ES or from external inputs. Starting from these classes, the developer can access further functionality of Studierstube ES using the C/C++ interface.

Content Database Many AR applications, but especially AR browsers, rely on remote data-sources hosting the content displayed on the client side. We consequently also needed a server hosting the content for our research prototypes. For several of the prototypes presented in this thesis, we used existing server software created within the IPCity project [1]. This server software implements a database accessible via an HTTP interface hosting geo-referenced content. Besides geo-referencing using GPS position, the server also supports referencing against various types of markers, natural feature targets and numerical IDs.

The database can host various types of content. Basic textual content can be encapsulated in KML or KMZ (a zipped version of KML) or encapsulated in other XML-based formats together with the position or type of reference. The XML-based meta data is analysed on the server side, if the schema of the XML is known, and the information is extracted. Furthermore, the XML can be validated if the schema of the XML is known to guarantee a valid XML file.

Other content formats can be easily integrated by registering new MIME formats to the server. In our case, this was for instance used for adding support for audio files, images, 3D content (stored as a serialized scenegraph), or video files. The front-end of the server is based on an HTML webpage, but as part of this thesis we extended Studierstube IO by classes which allowed for the convenient implementation of a client-server communication for accessing the database to upload and download content.

The server front-end provided several queries for filtering and accessing the content. Other supported queries can include a GPS position and a radius for accessing all content in the proximity as well as queries specifying the content type, username, marker or anchoring type, time stamp or a combination of all of these.

4.2 Panorama-based Tracking for Handheld Augmented Reality

One of most important requirements for any AR system is the tracking module used. The tracking (or anchoring) used determines the precision of the whole system, and consequently, the interface and possible interactions are also affected. As outlined in Chapter 3, our requirements for an AR 2.0 environment are even more specific because we require stable and precise tracking on a global scale that needs to run on mobile handheld devices at interactive framerates.

Tracking in outdoor environments is an especially demanding task with many challenges: Firstly, tracking in outdoor environments is expected to cope with the large environment. Furthermore, the bigger the environment (city-scale, country-scale, global-scale) in which the device should be tracked, the more likely it is that no scene knowledge (e.g.,

[1] www.ipcity.eu

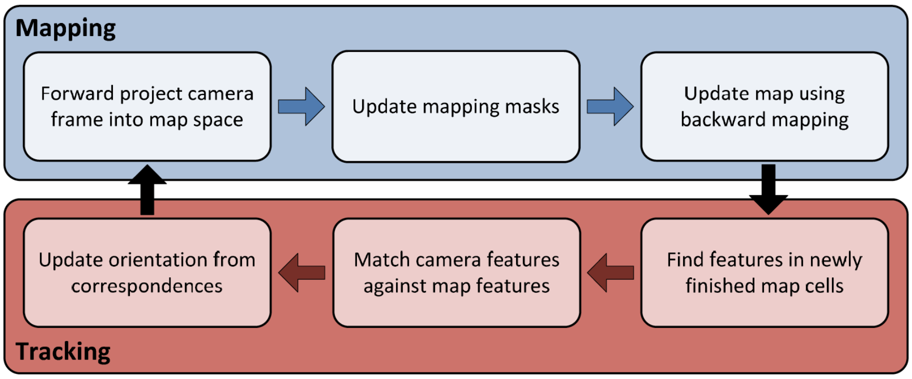

Mapping
Forward project camera frame into map space
Update mapping masks
Update map using backward mapping
Find features in newly finished map cells
Match camera features against map features
Update orientation from correspondences
Tracking

Figure 4.3: High-level overview of the mapping and tracking pipeline.

3D model of the environment such as present by Reitmayr et al. [108] or markers attached to the environments) exists. Thus, in an optimal case, the tracking works without scene knowledge or creates scene-knowledge on the fly (e.g., as a model or map of the environment such as in SLAM-based systems [56]). However, these approaches usually need to maintain the map while extending it, which is becoming increasingly complicated with bigger maps. Consequently, SLAM-based approaches on handheld devices are so far only feasible in small environments [58].

Another challenge is precision and pose stability (jitter). While vision-based approaches such as SLAM or marker-based tracking using 2D markers or Natural Features are known to be relatively precise with regards to accuracy and pose jitter, sensor-based approaches that rely on GPS, compass and accelerometers are prone to jitter and incorrect pose information due to an inaccurate signal (noise) and magnetic interferences. Still, most outdoor AR systems and especially AR browsers for handheld devices rely on sensor-based tracking only. Currently, the compass frequently can be off by tens of degrees, resulting in AR applications indicating directions rather than offering precisely placed overlays. Overall, when starting this thesis there was, and continues to be, a lack of precise tracking solutions for outdoor AR applications. This does not only limit the current generation AR applications but poses a major gap when realizing the concept of AR 2.0.

To close this gap we present in this chapter a system for tracking in outdoor environments that targets some of the shortcomings of existing tracking solutions for outdoor environments. It reduces the complexity of SLAM trackers by only working with 3DoF (orientation) instead of 6DoF and uses a panoramic map of the environment, which is easier to create and to maintain. We therefore assume the user of the system will remain in position while only performing rotational movements. While this might sound limiting, the analysis of the users' movements within our AR browser survey indicates that this is a typical movement pattern while browsing AR content in the environment [34].

We show that using *panoramic mapping and tracking*, we can achieve reliable and precise tracking in unprepared outdoor environments that can be extended by combining it with pose information from sensors. In the following, we present the key ideas for panoramic mapping and tracking, but refer to the original publication for an extended

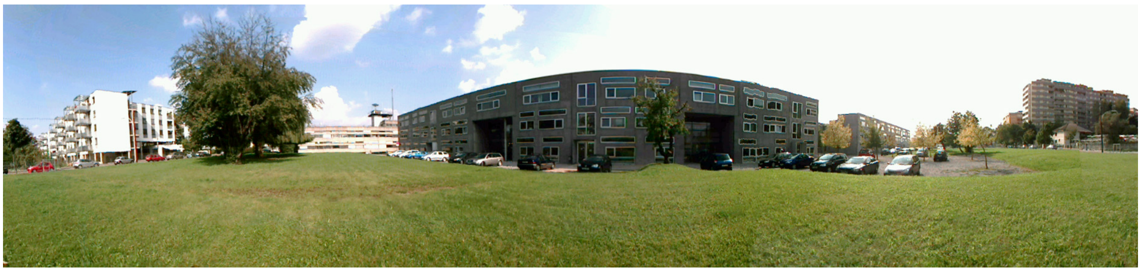


Figure 4.4: Panorama created on the fly using panoramic mapping and tracking.

discussion [152][66].

The core idea of this approach is to create a panoramic map of the environment using the incoming camera feed. We build and extend the panoramic map based on visual features and correspondences between the incoming camera image and the panoramic map. While extending the panoramic map, we also compute the current orientation of the camera using the panoramic map. Similar to a SLAM-based tracker, our approach can be split into a mapping step and a tracking step (see Figure 4.3).

4.2.1 Panoramic Mapping

As with most SLAM-inspired trackers we lack pose information for the initial frame. Therefore, we initialize the tracker with the pose information from the phone's accelerometer and compass. Based on the initial orientation data, we start the mapping process by projecting the first camera image onto the environment map representing the panorama. We decided on a cylindrical representation of the panorama to avoid problems with discontinuities (cube maps) or distortions (spherical maps). The cylindrical map covers 360° horizontally, while the range covered vertically ranges from $-38.15°$ to $38.15°$, which we determined to be sufficient for most outdoor environments. The remaining area would mostly contain the mapped ground and sky, both of which are unlikely to contain good features for tracking. The coloured panoramic map that is created has a size of 2048x512 pixels (see Figure 4.4).

Furthermore, the map is split into a regular grid of 32x8 cells, which simplifies working with the unfinished map (see Figure 4.5). Each cell is marked as either finished (when completely filled with pixels) or unfinished. Once a cell is marked as finished, we extract keypoints for tracking.

4.2.1.1 Filling the Map with Pixels

To fill the panoramic map with the image data, first we estimate the position of the current camera frame within the panoramic map by projecting the coarse outline of the camera frame (essentially five sample points per edge of the camera frame) onto the panorama. For the projection, we assume the camera to be at the centre of the cylindrical representation of the panorama (see Figure 4.6). This part of the algorithm is called *forward mapping.*

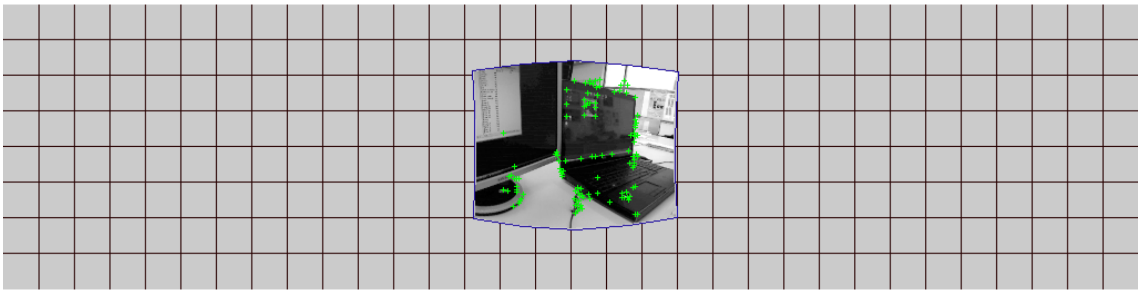


Figure 4.5: Grid of cells composing the map, after the first frame has been projected. The green dots mark keypoints used for tracking.

The mathematical details of the projection can be found in the original paper [152]. We essentially use ray-casting from the camera into the cylinder.

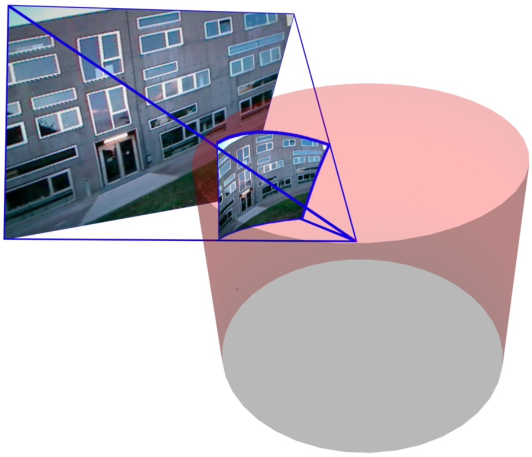


Figure 4.6: Projection of camera pixels into the cylindrical mapped panorama. The camera is assumed to be in the center of the cylinder

The forward-mapped camera frame gives an almost pixel-accurate mask for those pixels that the current video image contributes. However, if we rely on forward mapping to fill the panoramic map, we might create holes or over draw pixels. To fill the map with pixels, we therefore apply *backward mapping*. For backward mapping, we apply the inverse function to forward mapping. We iterate over the pixels in the panoramic map that are also supposed to be visible in the current camera image (determined via the forward mapping). We fill these pixels by computing the corresponding camera pixel (backward mapping) and write the camera pixels colour value in the panoramic map. As the coordinates determined via backward mapping are likely to be between the pixels we interpolate linearly to determine the final colour with sub-pixel accuracy.

We apply a mask to identify pixels that are already mapped and do not need to be mapped again. This helps to reduce the number of pixels that need to be mapped for each camera frame and consequently helps to improve the speed of the mapping process,

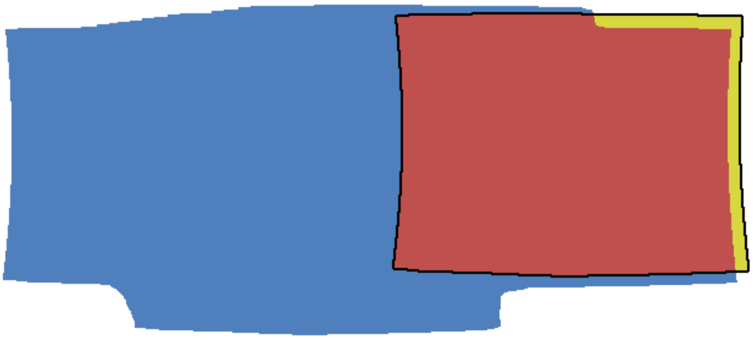


Figure 4.7: Masks created during a rotation of the camera to the right. Blue: Mask M of already mapped pixels; Black border: Mask T(P) for the current camera pose P; Red: Intersection of M and T(P); Yellow: Mask N representing the pixels that still need to be mapped.

as only a small fraction of the total camera pixels need to be mapped. We decided to not use a mask storing a binary value for each pixel (indicating if the pixel is mapped or not) as this requires checking each pixel. Instead, we compute the spans per rows that are already mapped and store only the left and right end of the spans. This allows a quicker check because we only need to compare the left and right coordinates of two spans.

The mask M (see Figure 4.7) is given by the pixels that are already mapped onto the panorama. Initially this mask is empty. For every incoming frame, we project the camera frame using the current pose P onto the panoramic map creating the mask *T(P)*. The resulting mask N is determined by row-wise boolean operation, and only masks pixels that are in the camera mask *T(P)*, but not in the map mask M. Consequently, every pixel of the panoramic map is only written once and not updated in later iterations.

4.2.2 Loop Closing

While the map is extended towards 360°, we face the problem of discontinuities at both ends of the panorama. These discontinuities are caused by accumulative tracking and mapping error, eventually leading to the problem that both ends of the panoramic map do not align well. We implemented a loop closing algorithm that overcomes this problem by creating an extended map (covering 405°) and matching both ends of the panoramic map. As soon as the map extends towards 360°, we extract FAST keypoints [116] at both ends of the extended panorama. Later we match these keypoints from one end of the panorama with ones at the other end. As soon as a good match is determined, we resample the map onto the final 360° (see Figure 4.8). For details of the loop closing process, we refer to our paper [152].

4.2.3 Panoramic Tracking

The tracking step of our panoramic mapping and tracking system computes the current camera orientation by matching features between the current camera image and the partially created panorama. Consequently, both steps, *mapping* and *tracking*, strongly rely

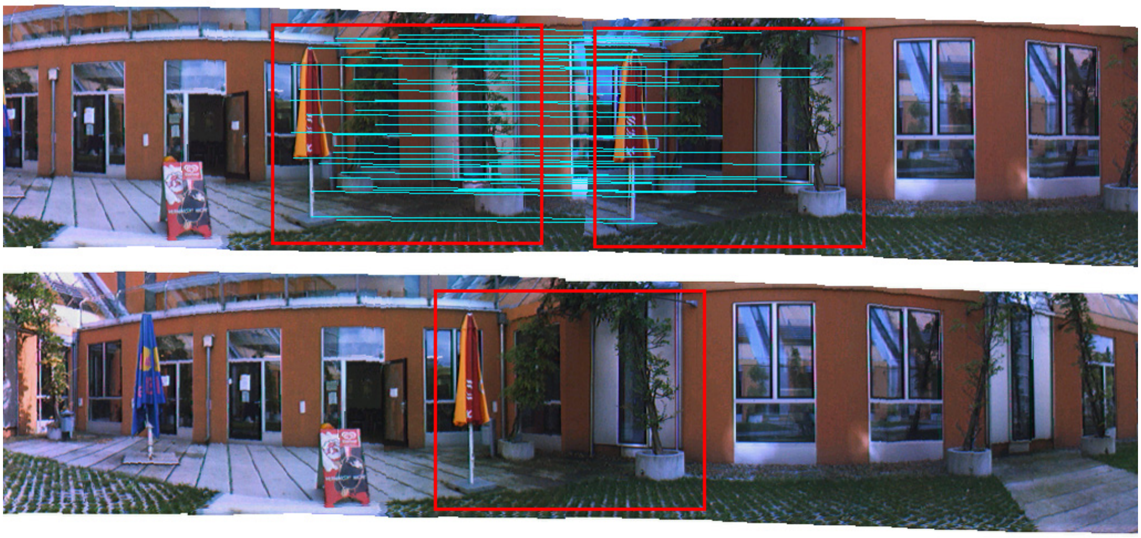


Figure 4.8: Loop closing applied on an example panorama; Top: Part of a 405° loop with overlapping areas. Bottom: Part of the same panorama closed to 360°

on each other as shown in Figure 4.3. We must know the camera orientation in order to project onto the panoramic map and the panoramic map to compute the camera orientation. In the following, we discuss how to compute the orientation of the camera by using the partially finished panoramic map that is still extended in parallel.

4.2.3.1 Keypoint Extraction

In a first step, we extract keypoints within the partially finished panoramic map using the FAST keypoint detector [58]. We avoid creating keypoints at the intersection of mapped pixels and pixels that are not mapped by only considering cells of the panorama that are fully mapped (see Figure 4.5). We compute the FAST keypoints not only on the highest resolution of the map cells, but also on scaled down versions that later compensate for faster movements. We further weight the FAST corners based on their score (strength of the corner) given by the FAST algorithm. After sorting the FAST corners for their strength, we we only keep a fixed number of the best FAST corners for each scale: For 64x64 pixel resolution cells (highest resolution), we keep 40 keypoints, for 32x32 we keep 20, and for 16x16 pixel resolution we keep 15.

4.2.3.2 Keypoint Tracking

We track the keypoints from one frame to the next by applying an active-search using a motion model. To reduce the area of the active search, we apply a motion model with constant velocity to predict the camera orientation for the current camera frame. For the initialization, we use the initial pose as estimate, but for later frames the current velocity is computed based on the movement between the previous frames.

This initial camera orientation is refined by forward projecting the current camera frame onto the map space using the orientation gained from the motion model. We identify the finished map cells that are in the panorama, but should also be visible in the current camera image. We back-project all those keypoints from these finished cells into the camera image by filtering out keypoints that are outside the camera image. We also create affinely warped patches of 8x8 pixels around the keypoints using the predicted camera orientation to predict the current appearance of the keypoints in the camera image. We refine the initially predicted position of the keypoints projected onto the camera image by applying Normalized Cross Correlation (NCC) template matching over a search area. We further speed up the template matching by applying it first on the scaled down version and later refining it using sub-pixel accuracy by fitting a 2D quadratic term to the matching scores of the 3x3 neighbourhood.

This tracking approach is drift free with respect to the panoramic map, but small errors in the mapping process might accumulate. However, these errors can usually be reduced once the panoramic map is extended to 360° and the loop-closing algorithm is started, which eventually re-projects the map to 360° and consequently reduces the error in the map.

4.2.3.3 Orientation Update

During the keypoint tracking, we determined a set of correspondences between the panoramic map (3D cylinder coordinates) and the camera image (2D coordinates in the camera image). We use these correspondences in a non-linear refinement process using Gauss-Newton iteration with the initial orientation estimate as a starting condition. We optimize in a similar way as for a 6DoF pose estimation (see [152]), but only consider the rotational parameters.

Similar to the keypoint-tracking algorithm, we determine the orientation estimate in lower resolutions first and refine this estimate using half and full resolution, but with smaller search areas. For the mathematical details of this orientation refinement we refer to our paper [152].

4.2.3.4 Relocalization

We have presented how the tracker uses a motion model in combination with active search to track the orientation from one frame to the next. Unfortunately, this approach does not permit a re-localization in case the motion estimate is wrong or could not be computed because not enough keypoints were visible in previous frames.

We therefore added a dedicated re-localization mechanism to handle these cases. The re-localization uses a history of stored keyframes with their corresponding camera orientation. These keyframes are determined while tracking from the panorama, but only a low-resolution version is kept to minimize the memory footprint of the final system. In case we cannot compute an orientation estimate using our frame-to-frame tracking, we match a scaled down version of our current camera frame against this history of keyframes. For increased robustness, we also blur these image samples - keyframes and camera images - before applying NCC-based template matching as proposed by Klein [57]. Once a good match is identified we refine the associated orientation.

In addition to using scaled down keyframes, we further reduce the memory footprint by keeping only one keyframe for certain orientations. We quantize the orientation into 12 bins for yaw (±180°), 4 bins for pitch (±30°) and 6 bins for roll (±90°) and keep only one keyframe for each bin. We update these stored keyframes if we revisit these orientations and notice that the keyframes are older than 20s. In total, the re-localization adds less than 1.5MByte of memory to the memory footprint in the case that all bins are used.

4.2.4 Initialization from Existing Maps

Sometimes it might be useful to start tracking from a previously created, partially finished panoramic map instead of creating a new one (e.g., to finish the panorama created in a previous step). In that case, the re-localization presented previously cannot be applied, as there is not a saved history of keypoints available.

For these cases, we apply a different method for initialisation of the panoramic tracking and mapping approach. We start by loading the partially finished panoramic map. Once loaded we start to compute FAST keypoints and create Phony SIFT descriptors [152] for the keypoints. We also compute keypoints and create the descriptor for the current camera image. After successfully establishing a match between these two sets of descriptors, we apply a RANSAC [29] -based approach to determine the orientation from the list of matched keypoints. Given that the overlap between the map and the current camera image is not very small, an orientation is usually found within a few frames and used to restart the mapping and tracking process.

4.2.5 Evaluation

We evaluated our prototype for panoramic mapping and tracking by creating 30 panoramas in different everyday environments that we evaluated in terms of robustness, accuracy and performance. All the panoramas were recorded on an Asus P565 smartphone with an XScale ARM CPU running at 800MHz.

4.2.5.1 Robustness

When recording the panoramas, we were aiming for a coverage of 360° to be mapped onto the panoramas. However, we tolerated cases where the full vertical area was not filled. While we were able to record 25 panoramas covering the 360° environment we could not finish five panoramas. This means that the tracking was working in some areas, but the mapping and tracking was unable to work properly with some visual characteristics visible in the camera view (see Figure 4.9). These characteristics were usually a lack of visible features or texture, moving objects causing wrong orientation estimates that could not be handled by RANSAC, or repetitive features (e.g., pebbles, grass). However, all these cases are general problems for visual tracking and not unique to our system.

4.2.5.2 Accuracy

For measuring the accuracy of the tracking, we analysed 25 panoramas that covered the full 360°. All the panoramas were created with loop-closing enabled and we made use

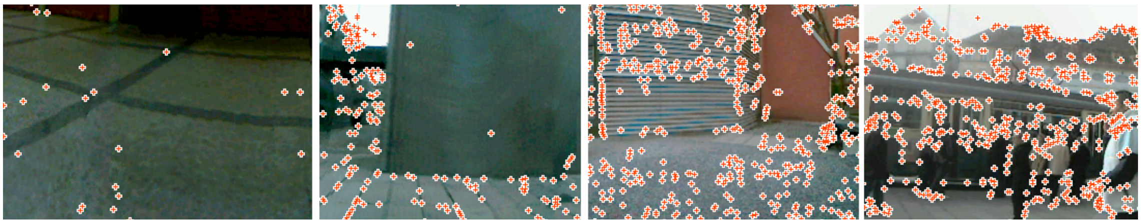


Figure 4.9: Problematic scenes: a) Floor that is lacking texture. b) Wall that is lacking on texture, only line details on the floor. c) Line details on the wall (keypoints are due to sampling artifacts), pebbles on the floor. d) Moving objects (tram, people) covering most of the image.

of this by measuring the offset that needed to be corrected when closing the loop. This measured offset indicates the accumulative error that was created within the mapping process. Assuming a properly calibrated camera, we measured an average error of 1° that corresponds to a 5-6 pixels offset.

We further analysed the pose jitter using a camera on a static mount. The measured jitter was about 0.05° for head, pitch and roll respectively.

Our presented method tolerates a rotation around an axis, which is not centred in the phone (the user often rotates around her centre of gravity rather than around the mobile phone). Fortunately, in most outdoor scenarios, the distance between the camera and objects in the environment is large compared to the involuntary translational motion that occurs when rotating a handheld device (usually the length of the arm). As shown by DiVerdi et. al [23], errors are therefore negligible.

4.2.5.3 Performance

We evaluated the performance of the panoramic mapping and tracking on an off-the-shelf smartphone from Asus. The Asus P565 smartphone comes with a single core XScale ARM CPU running at 800MHz. We achieved real-time performance of $\sim$ 15 milliseconds for the mapping and tracking approach with about 10.6 milliseconds for the tracking step and 4.5 milliseconds for the mapping step. Once the map is filled or the camera is pointed towards areas that are already in the map, the total costs are lower as the mapping costs are negated. On more recent phones with CPUs running above 1GHz, the mapping and tracking is usually 30fps and only limited by the incoming camera stream, leaving enough CPU cycles for other tasks such as rendering.

However, the presented approach for loop closing is more expensive and can take up to 10 seconds if high quality re-sampling is enabled (Lanczos filter). If a less demanding re-sampling, such as nearest neighbour filtering, is used the costs can be reduced to 2s. Using nearest neighbour filtering will cause visual artifacts in the panorama, but will not affect the tracking noticeably. However, in practice, loop closing is a rare situation since the creation of 360° maps takes between 1-2 minutes and is required only once for a map.

If the system is to be reinitialized from an existing map, a large number of features need to be extracted from the loaded panoramic map. Fortunately, this is only done once. Reinitialization of each camera frame by matching against the panoramic map takes $\sim 120ms$ on the phone given that the descriptors for the map are created.

4.2.5.4 Summary

The presented panorama tracker exhibits a novel approach to improving the robustness and accuracy of tracking the orientation of mobile phones, especially in large outdoor environments. It thereby runs at interactive frame-rates, leaving enough resources for other computations such as rendering. Unfortunately, the described implementation of the panorama-based tracker only tracks a movement relative to the initial position, which can be seen as a drawback. However, later work such as that done by Schall et al. [122] and our work by Langlotz et al. [66] present approaches for how to incorporate position estimates from other sensors, such as compasses and accelerometers, yielding relative measurement and further improve stability. The panorama-based tracker only allows tracking the orientation and still requires GPS sensor data for the position estimate. This introduces a mean positional error of 5-8m [163]. However, later work by Arth et al. [2] presents an approach that uses our presented panorama-based tracker combined with image-based localization which gives accuracy within cm range for the initial positional estimate, and therefore, overcomes the limitations of a GPS position estimate with limited accuracy.

In addition to tracking information, we also create a panoramic map of the environment, which can be useful for interface purposes, but also for anchoring information to the environment, as we show later in this thesis.

Overall, we presented a first approach for precise and robust tracking in outdoor environments which fulfils our stated requirements of AR 2.0 by closing the existing gap in tracking technologies. This approach therefore forms the core for our prototypes to be presented later in this thesis.

4.3 Conclusion

We introduced two core components for implementing the concept of AR 2.0 that are fundamental because they are technical enablers. These fundamental technologies enable us to implement prototypes presented later in this thesis in order to demonstrate the conceptual model of AR 2.0. Firstly, we briefly presented Studierstube ES, a framework for Augmented Reality on mobile phones. We use and extend this framework throughout this thesis by adding functionality to handle different media formats, as well as by adding interfaces necessary to share and distribute the content created using the backend that was presented. Secondly, we introduced a novel tracking approach suitable for tracking on mobile phones in outdoor environments. This approach of panoramic mapping can be seen as a plugin for extending the existing tracking functionality within Studierstube ES. Thereby, the contribution of the panorama-based tracking is essential to building the prototypes presented later in this thesis which go beyond current generation AR browsers in terms of tracking robustness and accuracy. This affects the quality of the interface and allows completely new ways of interacting with the device and the displayed content.

In the following chapters we will introduce the prototypes developed, which demonstrate the concept of AR 2.0 by showing how to create and integrate various forms of social media within AR applications.

Chapter 5

Editing and Using 2D Annotations in Augmented Reality

Contents

Two-dimensional annotations, text or 2D graphics, are the most common media type displayed when using outdoor Augmented Reality systems. This chapter presents our research targeting the in-situ creation and usage of textual annotations and 2D graphics for mobile Augmented Reality. We report on several evolutions of the system we developed, which allows for a precise and robust placement of annotations and graphics in an unprepared outdoor environment. We also present the interface that is used for creating, placing and browsing the textual or graphical content. Finally, we provide results from a user study and technical evaluations of our approach, investigating the usability of the interface and the robustness of the algorithms used for tracking and detection.

5.1 2D Annotations Within Outdoor Augmented Reality

One traditional use scenario of Augmented Reality is to provide additional information about the current environment using 2D annotations. These 2D annotations can be either textual annotations (labels or text-blocks) or graphical annotations (images).

Although 2D annotations have already been presented in the Touring Machine Prototype [28] and are available in all current AR browsers, these systems continue to have limitations regarding the placement and function of 2D annotations. Firstly, existing systems fully rely on sensors for tracking device position and orientation. The sensors used are typically GPS for the position, and a combination of compass, accelerometers, and gyroscopes for the orientation estimate. But given the nature of these sensors, the estimate is not very accurate and can differ by more than 30m (GPS) or tens of degrees degrees (compass) from the original position as desribed by Zandberg et al. [163] and Schall et al. [121]. This makes it hard to overlay precisely registered information and results in giving directional hints rather than precise augmentations. Furthermore, most of these sensors suffer from noise in their estimate, which can be perceived by the users of an AR system as the displayed information jitters around the estimated position as described by Schall et al. [121]. The sensor drawbacks mentioned are even worse on smartphones, as demonstrated by Schall et al. [122], as the quality of the sensors used is usually below the quality of dedicated sensors. This is mostly due to constraints in manufacturing costs and size of the device.

Secondly, all presented approaches developed for AR browsers lack an interface that allows users to create and place 2D annotations in their current environment. Typically, the annotation content is created beforehand in a desktop environment and placed using tools such as Google Earth, or dedicated applications that make use of Google Maps to link information to the environment. Consequently, many of the annotations created are just coarsely referenced (usually also lacking height as they are referenced at the current ground level) and cannot be used to tag or describe small objects in the environment, such as windows, doors or other objects of similar or smaller size. Furthermore, this approach for creating annotations does not permit a spontaneous creation of information while in place.

In this chapter, we present an approach that tackles several of the presented limitations of current AR browser systems. We discuss how laypersons can create and precisely link 2D annotations to the environment using a mobile phone. Additionally, we show how the created 2D annotations can be shared with other users. Finally, we demonstrate shared annotations that are precisely augmented in other users' current view and present how to accurately track the user's orientation to guarantee a convincing overlay of the textual annotation that is less affected by jitter.

The first iteration is based on our research presented by Wagner et al. [152] and describes how an augmentation can be statically anchored within a panorama. The second iteration proposed extending that system by targeting scenarios, where users have a different position than the one where the annotation was created at. This requires more advanced algorithms to re-detect the annotation's anchor point within a newly created panorama and is based on our work presented by Langlotz et al. [68]. Finally, we demonstrate how to integrate further constraints to guarantee increased robustness even in the case of changing environmental conditions. These additional constraints result from additional use of sensors, creation of the panorama with an extended dynamic range, and determination of a global transformation that estimates the position of the annotations as presented in Langlotz et al. [66].

Despite the fact that these approaches for robustly overlaying textual annotations use additional features for estimating the position on the user's environment, we show that it is possible to combine these algorithms with an easy-to-handle interface, which allows a layperson to create, place, and share textual annotations with other users.

5.2 Panorama-Referenced 2D Annotations

In the context of mobile AR applications, having a panoramic image of the environment has many advantages. As we showed previously in Chapter 4 as well as described by Wagner et al. [152] and DiVerdi et al. [23], a panoramic image can be used to precisely track a user's orientation. Furthermore, it allows for applications that make direct use of annotated panoramas instead of the augmented live camera image such as presented by Wither et al. [160]. In that example, the phone is tracked using a gyroscope and a prerecorded panorama is displayed based on the current tracking information. That prerecorded panorama can be altered by adding textual or graphical annotations. The advantage is that the visual representation of the environment always accurately aligns with the displayed information. However, the displayed environment in form of the prerecorded panorama might not match the current environment (e.g., different lighting conditions, seasonal changes, wrong position estimate).

We therefore decided not to replace the live camera image, but instead, to match the live camera image against a prerecorded panorama. Assuming the current user is in the same position as the user who created the prerecorded panorama, it should be possible to align their views and consequently transfer annotations that are registered within the first panorama into the view of the second user. We call this approach *panorama-referenced 2D annotations*.

We implemented our prototype of panorama-referenced 2D annotations using the previous research on panorama-based tracking in outdoor environments and extended it for our purposes. As explained in Chapter 4, our proposed method can create high-resolution panoramas on the fly.

We extended this system by adding an interface that allows users of the application to add 2D annotations on such panoramas directly on a mobile phone. The user creates an annotation by tapping on the screen where the annotation should be created, causing the application to prompt for the text of the annotation. Alternatively, one can also shoot a picture with the camera or load an image from the photo library. Once the text is typed-in or the picture is provided, the application stores the 2D annotation together with the corresponding coordinate, the *anchor point*, in the panorama. This coordinate can be easily determined by projecting the selected coordinate via the camera image onto the panoramic image, using the orientation information obtained from the panorama-based tracker.

Once the users quits the application, the system saves all annotations and their map coordinates in an XML file. We write and compress the XML file together with the final panoramic image into a zip container. We add the GPS position to the created data and send the zip file to the server (see Chapter 4 for details on the server implementation) hosting the GPS-tagged content. When another user is exploring the same location, the user's phone will send its current GPS position to the server and retrieve the zip file again. Our system is able to initialize the tracking using the panoramic image contained in the zip file.

We realized this initialization from a prerecorded panorama using Phony SIFT descriptorscreated by Wagner et al. [153], which allow for a fast, robust and rotation invariant matching (see Figure 5.1). We can directly mount the downloaded zip file and load the pre-

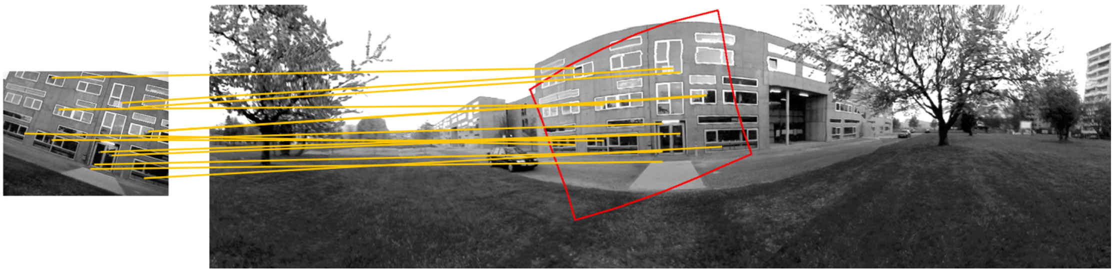


Figure 5.1: Initialisation from a prerecorded panorama using SIFT descriptors. (Left) Grey-scale version of the current camera image used for matching. (Right)Grey-scale version of the prerecorded panorama used for matching.

recorded panorama from it. After that, we apply Phony SIFT to compute keypoints and their descriptors in the panorama image. We also compute the Phony SIFT descriptors on the current camera image. We apply a brute-force algorithm to match the keypoints of the two images, as it turns out that creating a spill-forest as described in [153] for a big dataset such as a panoramic image is too time consuming on a mobile phone.

Once the correspondences for the two datasets are computed, we apply a *Random Sample Consensus* (RANSAC)-like approach [29] to compute the orientation of the camera image. We use a windows with a horizontal size of 7 cells (corresponds to $78,75°$) in the panoramic image, which contains the largest number of matched correspondences. We only use the correspondences within this window, as this is likely to describe the position of the camera image within the panorama. To start the RANSAC refinement, we take two matched keypoints to compute the orientation. This orientation forms the hypothesis for the RANSAC and is therefore checked against the remaining matched keypoints. If a large number of the keypoints support the hypothesis, it is refined using all matched keypoints. Otherwise, a new hypothesis is computed and checked. If no good match can be found between the current camera image and the panoramic image, the entire matching process is repeated with the next incoming camera image. A good match can usually be found within a few frames, while the matching takes between 190ms for the first frame and 120ms for all following frames (all timings are computed on an Asus P565 smartphone with an XScale ARM CPU running at 800MHz). The difference of 70ms is mostly due to the fact that for matching the first frame, we need to compute the keypoints and the descriptors for the prerecorded panoramic image, but can reuse this information for later matching steps.

Once the current view is matched against the prerecorded panorama, it is possible to transfer all 2D annotations referenced against the prerecorded panorama into the current view. The transformed coordinates can be used to precisely augment the annotations into the view of the current user (see Figure 5.2).

Overall, the proposed workflow of the system is as follows: The first user walks to a place of choice and starts the application. By rotating the phone, a panoramic image of the environment is created. The user places some 2D annotations that are referenced in the

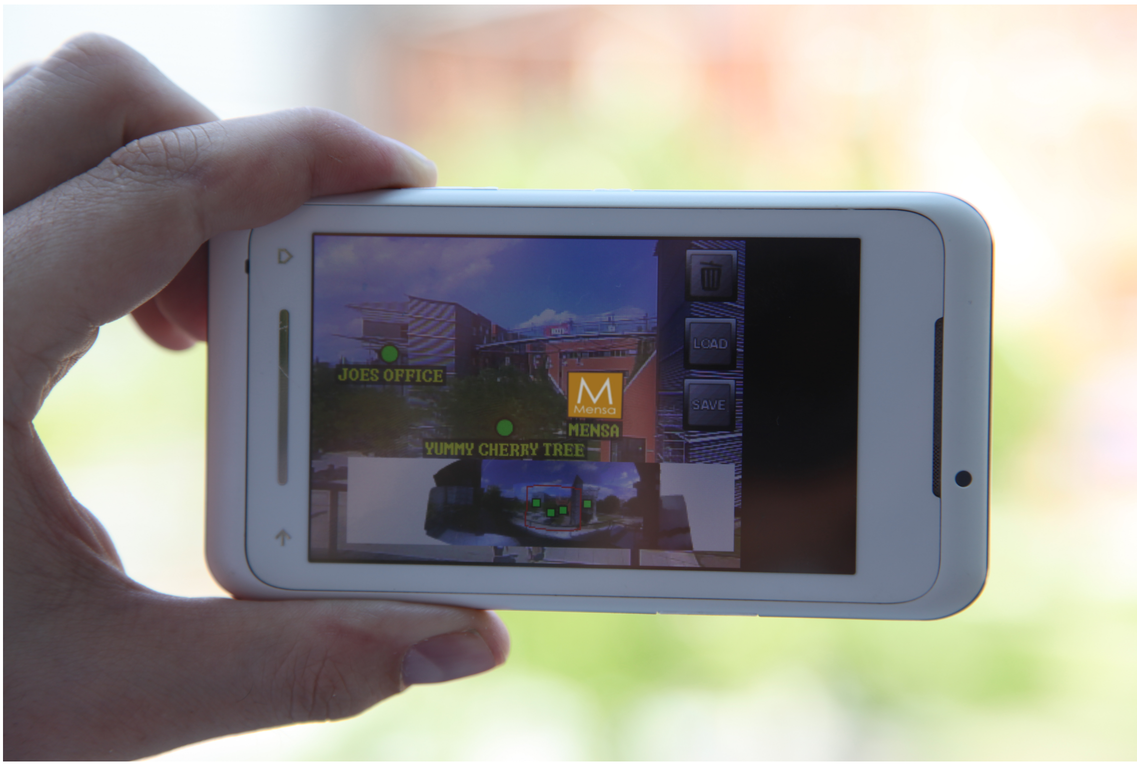


Figure 5.2: Implemented prototype showing panorama-referenced 2D annotations and a partially recorded panorama that is used for tracking and matching the annotations.

panorama coordinate system and stored together with the panoramic image in a zip file. This zip file is sent to the server and tagged with the GPS coordinate. A second user can retrieve these 2D annotations by walking to the same position. The server is automatically queried for 2D annotations using the current GPS address and delivers the dataset created by the first user in this location. The application loads the contained panoramic image and reinitializes the system within regard to the original panorama. Thus, the second users is able to browse and see the 2D annotations of the first user.

5.3 Template-Based Detection and Tracking of Annotations in Outdoor Environments

The previous approach allowed transferring the anchor points of 2D annotations that are given in panorama coordinates to be transferred into a new panorama by matching the current camera frame against the prerecorded panorama. Of course, this only works if both users, the one creating the application and the one browsing the annotations, stay in the same position. However, given the nature of current generation built-in GPS sensors, estimating the same position is only possible with a given error. This error is, even if in

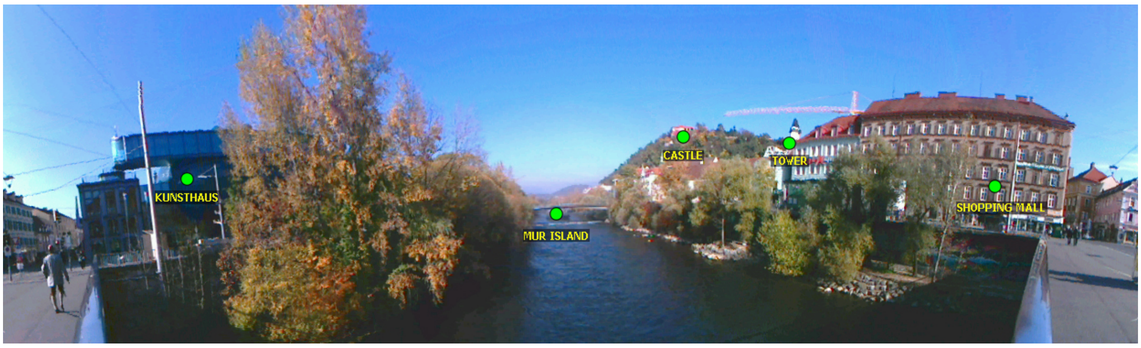


Figure 5.3: Our vision-based system presents an improvement over regular compass-based annotation systems. By creating and storing panoramas, we are able to locate and visualize annotations with pixel accuracy even if seen from slightly different positions.

good conditions (wide and open area, clear sky), in the range of several meters. Thus, this approach is hard to realize under realistic conditions. Furthermore, it always requires uploading the whole panoramic image, which can be quite huge.

To solve these two issues, we present an approach that allows users to create and place textual annotations on mobile phones that are then stored in a self-descriptive way. We termed this approach *template-based detection and tracking of annotation*. Similar to the previous approach, annotations can be shared with other users via a server that hosts the GPS-tagged content. However, we only uploaded a small image patch that describes the annotation's anchor point within the panorama instead of uploading the complete panoramic image. For retrieving the annotations, we used the current GPS position for efficient indexing, but identified the annotation's position using template matching against the panoramic map. This approach yields accurate and robust registration of annotations within the environment, even when seen from a slightly different position compared to where the annotations were created (see Figure 5.3). At the same time, we reduced the amount of shared data necessary because only image patches have to be uploaded to the server.

5.3.1 Annotation Detection and Tracking

In the following, we present a new approach, which does not rely on the prerecorded panoramic image and the position of the annotations within the panorama, as it stores annotations in a self-descriptive way. To describe the annotations we selected template matching using *Normalized Cross Correlation* (NCC) [13] instead of using SIFT descriptors because we discovered the SIFT has many disadvantages if applied to our problem:

Applying SIFT to describe an annotation's anchor point would require matching the annotation's anchor point within a panorama. To accomplish that, we would need to build descriptors and a search structure for our whole panoramic image which has a size of about 2048x512pixels. Even using our fast and efficient implementation of Phony SIFT would be very slow on mobile phones. This is particularly problematic as this would require

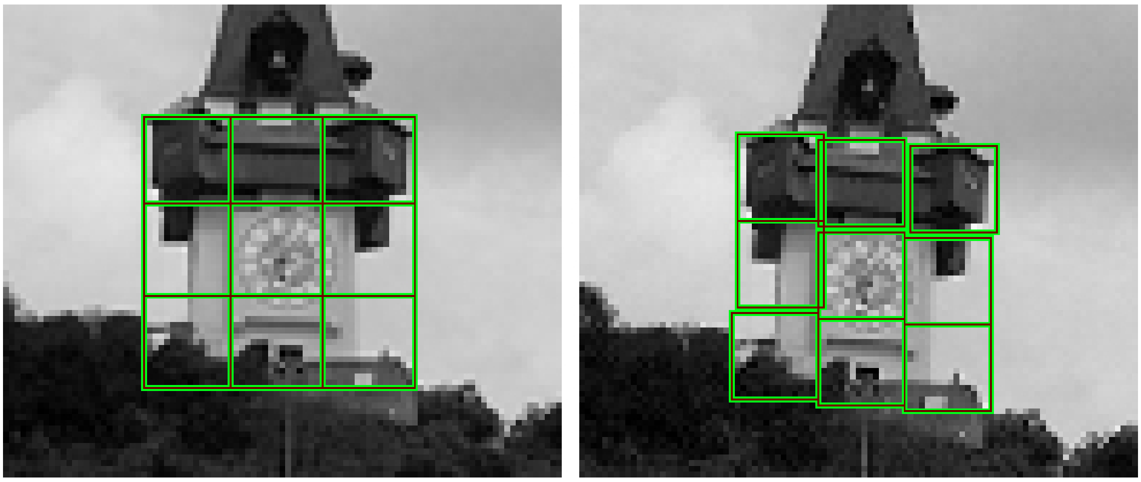


Figure 5.4: The support area of an annotation is described using a 3x3 grid of templates encoded using a Walsh transform. (Right) Matching the templates from a slightly different camera perspective.

rebuilding the descriptors and the search structure every time the panorama is extended. Additionally, we want to keep the support area describing an annotation's anchor point as small as possible, and SIFT is known to deliver poor performance when used with small image patches. This is usually due to the fact that the supporting area doesn't provide enough unique features to robustly match that image patch.

Furthermore, many of the features of SIFT-based matching are not needed. Since we can compare against the panoramic map and we can assume that this map is recorded in an upright position, we do not need SIFT's rotation invariance. As we can also assume that we are at a very similar position, we do not need SIFT's scale invariance.

Instead, we propose to use NCC for matching the image patches. It shares the brightness and contrast invariance with SIFT, but is more effective for smaller image patches. Matching image patches against the panoramic image instead of the current camera image further allowed us to decouple the problem from the current view. Thus we can match annotations' anchor points that are not in the current view, but are already in the recorded panorama. This further allows us to address the problems of showing the live view and creating the panorama while also matching the image patches against the panorama separately and implementing them as different threads.

In the following, we describe an annotations anchor point as an image patch with a size of 48x48pixels. To make the description of the image patch more robust against small rotational or perspective changes we divided it into 9 smaller templates forming a 3x3 layout as it can be seen in Figure 5.4. Each of these 9 templates with size 16x16 pixels is matched independently, but together they should roughly (with 5 pixels tolerance in each direction) assemble the given 3x3 layout.

We empirically determined that this configuration works, gives the best result, and also accommodates small uniform changes such as a different scale or slightly different perspective caused by a different viewing angle. Furthermore, each image patch describing an annotation's anchor point requires only about ~ 2 kilobytes in storage instead of the > 1 megabyte that was required to save the panorama. Finally, saving the annotations independently of each other instead of describing them via a panorama makes them easier to merge in case several users contribute annotations to one position.

5.3.1.1 Walsh Transforms for Faster Template Matching

One problem with using template matching occurs if many image patches have to be checked against a large image, as every pixel position within the image has to be checked. In our case, an image patch has a size of 16x16, but it needs to be checked against a panorama of size 2048x512 pixels. This check is very slow, especially on smartphones with minimal, slow memory and a different caching structure compared to desktop computers. To overcome this disadvantage of template matching, we applied Walsh Transforms [96] as a pre-check as they have several distinctive advantages: They (1) approximate NCC-based template matching, making it only necessary to check the best matches resulting from Walsh Transforms-based matching with NCC. Walsh Transforms (2) are fast to execute and use integral images, which makes the execution speed independent of the template's size. Finally, Walsh Transforms (3) scale well with the number of image patches to be checked because they only have to be computed once and can be used for matching against an arbitrary number of transformed templates.

However, Walsh Transforms that rely on integral images are quite memory intensive, and even worse, they are hard to create for incomplete images such as our panoramic map, which is constantly updated while the user rotates the phone. As soon as new pixels are projected into the panoramic map, the integral image has to be updated as well.

To overcome these limitations, we apply a divide and conquer strategy. We divide the panoramic map into smaller tiles and start the template matching against a tile as soon as it is regarded as finished. Thus all pixels within that tile are filled. Once a tile is finished, we start to create an integral image that it is large enough so that every pixel within the tiles can be checked as part of a dense search. Thus we have to create the integral image with an overlap at the right and the bottom. For each pixel location within this integral image, we compute eight Walsh Transforms and compare them against the Walsh Transform of the image patch describing the annotation's anchor point.

As previously stated, Walsh Transforms only approximate NCC-based template matching by giving a lower bound of the matching error. Consequently, we apply NCC on the best matches resulting from the Walsh Transforms to get a precise matching score. We store the position of the ten best matches together with the NCC scores. If four of the nine templates forming the 3x3 arrangements can be matched, we also check if they form roughly the correct layout (relative position with respect to each other) (see Figure 5.4). We empirically determined that at least four of the nine templates should match to guarantee a good ratio between false positives and failed detections, as we wanted to reduce the risk of having false positives. All matches that succeed in this final test are regarded as detected and the annotation's anchor point is put at the corresponding position.

Due to the fact that the panorama might not be finished, later, based on the number of matched templates, a better position may be detected. In that case, because we store the ten best matches with the corresponding NCC scores, we can update the position of the annotation's anchor point.

5.3.1.2 Real-Time Scheduling of Annotation Detection

As previously mentioned, we use the template matching of the image patches with the panoramic image instead of the camera image. Besides the mentioned advantage of allowing an implementation using parallel programming, this also has a further distinctive advantage: it allows a guaranteed frame rate.

We put each tile that is considered to be finished in a queue that holds all the tiles that need to be checked for the occurrences of image patches describing an annotation's anchor point. In each rendering loop iteration, we check as many tiles as possible while still guaranteeing an interactive framerate. Because the operations that are needed to check the image tiles are simple and the timings are predictable, we can easily schedule the number of tiles that can be processed each frame. We evaluated the timings on an ASUS smartphone (ASUS P565). Matching one cell against 12 annotations takes $\sim$ 54ms. Targeting a frame rate of 20Hz (50ms per frame) allows scheduling 10ms for detection in every frame. On slower phones, finding the annotations requires more time because it takes more time for a cell to be checked, however interactive framerates are still guaranteed.

5.3.2 Prototype

Based on the algorithm presented, we implemented a prototype application to test performance in a real scenario, but also to run a first field trial with possible end users. The prototype application implemented a campus-based information system that, similar to commercial AR browsers, displayed 2D information linked to objects in the environment.

In addition to the AR view, in which the user sees the augmented annotations, we also integrated a map view. This was needed because our prototype only supports augmentation while in a static position, and furthermore, requires users to be close to the spot where the annotations were created in order to retrieve them. The integration of the aerial map (see Figure 5.5 left) therefore helps users navigate the environment while not using the AR interface. The aerial map view always places the user's current position (determined using GPS) in the center of the map, while spots that already contain annotations from other users show up at the positions where the annotations were created. Once the user moves close to a position where annotations were created, the application starts to download the annotation data in a background thread. To compensate for an inaccurate GPS position, we download all annotation data within a certain proximity of the user.

Once the user is close to an annotated spot, she can switch the interface using an animated transition [90]], which allows her to transition from the aerial view into the first-person AR view (see Figure 5.5 right).

Transitioning from the aerial view to the first-person AR view automatically triggers the system to start creating a panoramic image that is used for tracking as described in Chapter 4 and is further used to match the downloaded annotations. Once the system determines a good match using the template matching, the annotation is displayed using the matched position as the annotation's anchor point.

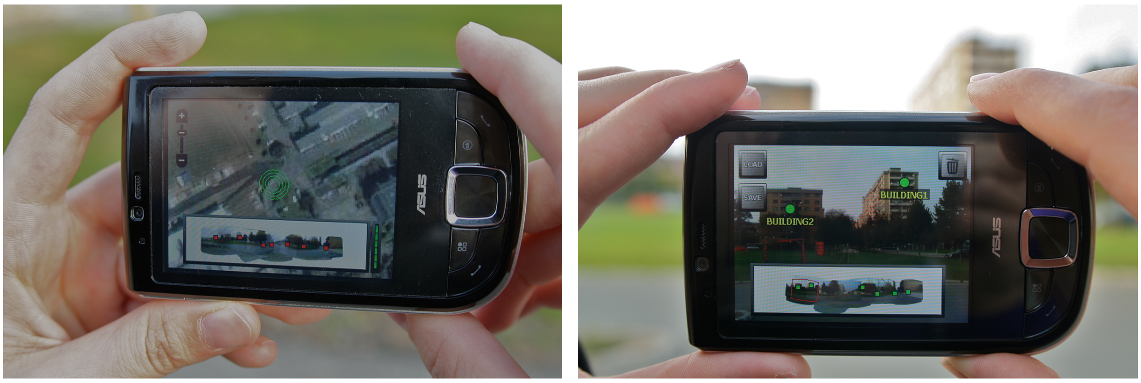


Figure 5.5: (Left) 2D map overview showing nearby annotations. (Right) First-person view of the annotated panorama.

If the phone that was used to create the annotation was equipped with a compass, it is also possible to indicate the rough direction of the annotations in the aerial map view as well as in the panorama of the environment, which is shown as a small image below the AR view. This can assist the user in finding the annotations. If the users decide to insert an annotation they tap the screen and type in the textual annotation. Similarly, they can load an image from the phone's memory to place it at the selected position. Because our approach requires the annotation to be at a position that results in a well structured image patch, we analysed the image patch using the approach presented by Shi and Tomasi [129]. While the user is tapping, the system provides feedback using a coloured dot that is at the position the user tapped the screen (red indicates that no annotation can be made at that position as there are no significant image features, while green indicates that a annotation can be created). After placing the annotation, the 48x48 image patch is extracted, which is later used for template matching. This image patch together with the 2D information of the annotation is compressed, GPS tagged, and uploaded to a content server as presented in Cxhapter 3. This content server allows us to query for content in a specified proximity, as well as to combine this query with other tokens such as users or user groups. The final workflow of the implemented system can be seen in Figure 5.6.

5.3.3 Results

Based on our prototype, we conducted a first field trial to gather feedback from users about the interface and tracking performance. At the same time, we were logging the application data on the phone to yield data for a technical analysis covering the redetection performance of the prototype.

The test was composed of three parts. Firstly, we asked the users to browse two different datasets that were presented in a random order. Both datasets were created beforehand. But while one dataset was created one day earlier under different environment conditions (weather, position of the sun), the other one was created no more than 30 minutes before the actual test. Therefore, the second dataset closely reflected the conditions within the actual test. Each of the created datasets contained six annotations that were

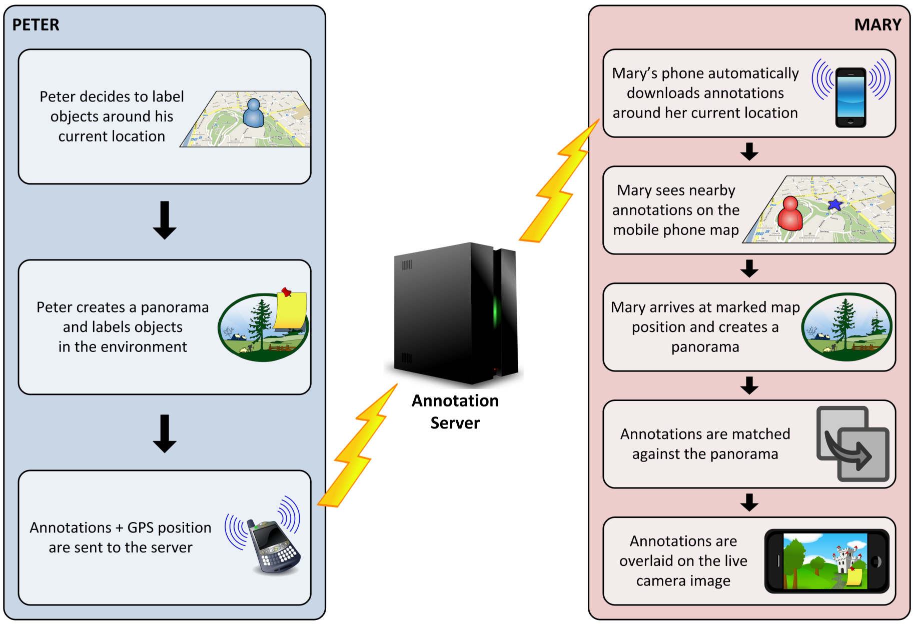

PETER
Peter decides to label objects around his current location
Peter creates a panorama and labels objects in the environment
Annotations + GPS position are sent to the server
Annotation Server
MARY
Mary's phone automatically downloads annotations around her current location
Mary sees nearby annotations on the mobile phone map
Mary arrives at marked map position and creates a panorama
Annotations are matched against the panorama
Annotations are overlaid on the live camera image

Figure 5.6: The workflow of the panoramic AR annotation system involves two users - Peter creates annotations and at a later time Mary browses through these annotations.

distributed over a large part of the current environment (panorama). The annotations labelled objects that were within 10-200m of the users. Two out of the six annotations were created at a spot 5m away from the current position of the user, so they had a slightly different parallax (one object was roughly 15m away the others approximately 70m). The objects were annotated with numbers from one through six, and we asked to users to point to the annotated object and the number as soon as they identified it. In the second part of the trial we asked the participants to create some annotations on their own. We asked them to create annotations for several objects that were indicated by the experimenter. These objects were ranged from big to very small, so that they were hard too see on the screen. Finally, we asked the participants for feedback using a semi-structured interview.

5.3.3.1 Preliminary User Feedback on Usability

For the trial we recruited eight users (three female/five male, age 22-34yr) with no previous experience with Augmented Reality or location-based systems. The first task for participants was to identify annotated objects in the environment. All participants succeeded in that task as long as the system was able to re-detect the annotations. This showed that the visualization used is good enough to establish a clear link between the annotated object and the annotation itself. Three participants even commented that they were sure that the annotated object was the correct, one as the annotation stayed exactly at the same position. This feedback supports of our hypothesis that the panorama-based tracking is fast and very accurate. Furthermore, none of the participants noticed any drifting or jumping in the labels once detected. However, some of them experienced an occasional loss of tracking, which was indicated by a question mark on the screen; our users were consistently able to recover quickly by pointing the camera towards a previously visited region. Six out of the eight users stated that as they became more familiar with the application, they could better avoid tracking problems. This was also noticeable as users progressed from a very stiff to a more relaxed posture over time. Users reported that the primary causes for disruption in the panorama-based orientation tracking were moving too fast or pointing the phone to the sky.

Six out of the eight users commented that by exploring the proximity of an annotation helped with the redetection in case that annotation did not show up immediately. This strategy was possible because we were showing the participants the estimated position based on the compass data. The remaining users said that the 2D annotation showed up right after exploring the relevant part of the environment and that there was no need to further explore the area. None of the users noticed any cases where the annotation changed position while using the application (which can result from finding a better match in a recently added part of the panorama).

The interface was well received by all the participants, and the preview panorama especially received many positive comments. It was used by all participants except for one for orientation within the environment and to identify unexplored regions that might contain annotations. Furthermore, several participants in the trials used the preview panorama to identify visited areas they could use to reinitialize the tracking. They noticed that pointing the phone at already identified areas helped the panorama-based tracker to "snap-in" and restart tracking.

The only complaints focused on the small screen space, which is determined by the device used, but some users said it makes it hard to identify the annotated objects on the screen. That fact caused many users' gaze to often switch from the display to the real environment for verification. That was especially noticeable in the cases where we asked the users to create an annotation referring to a very small object. Two user suggested integrating a virtual magnifying glass or a zoom function, while some others suggested dynamically adjusting the dots that indicate the 2D annotation's anchor point.

5.3.3.2 Matching Results

As part of the evaluation we also logged the application data, while the users tested our application. The logged data includes the newly created panoramic images, which are used for matching the image patches describing the annotation anchor point. We were also storing the three best matches of each annotation within the panorama regardless of whether the matching threshold was high enough to be considered as a valid match. Beside this, we were also recording cases where the tracking was lost, storing by how long the tracking was lost, together with the last position where the tracking was working.

The data obtained within the user tests showed significant differences in redetection performance. The set of annotations which were recorded right before the field trial showed a high rate of redetected annotations (in total 43 out of 48 annotations could be detected, redetection rate = 89.53%). While the annotations recorded one day before the trials showed a lower redetection rate (in total 27 out of 48, redetection rate = 56.25%). Both tests showed no false positives, which confirmed our choice of the chosen threshold

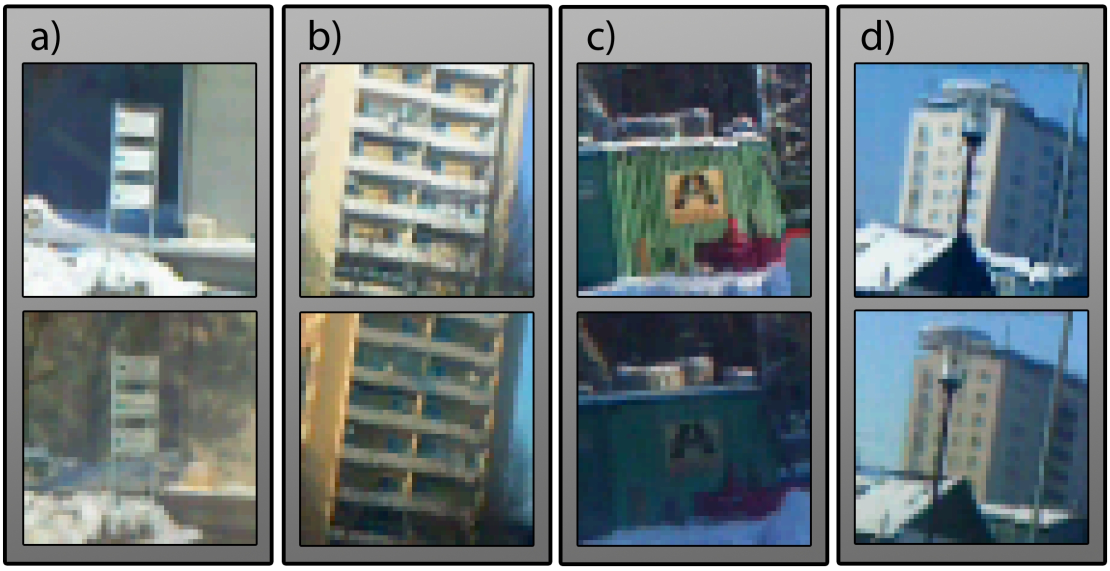


Figure 5.7: Appearances of annotations' anchor points in maps taken at different times of day or different locations. The upper row is the original appearance as recorded, while the lower row is the appearance of the same feature in the target panorama.

for the NCC-based template matching. The results indicate that our approach shows acceptable results in the case where annotations were recorded under similar environmental conditions. However, the detection rate was low for annotations created under a different environmental conditions.

These results were backed up by a deeper analysis of the logged data, which showed that the varying lighting conditions caused most of the problems in terms of redetection. Many problems were caused by shadows that changed intensity and directions. One dataset was recorded not only one day earlier, but also at a different time with a different position of the sun (and consequently different shadows), which all strongly affected the image patches used for matching (see Figure 5.7a). The changing position of the sun also caused self-shadows to vanish, leading to less visible object details (see Figure 5.7c). Another problem that showed up in some logged datasets is caused by the parallax effect and related to the different user positions (see Figure 5.7d). While the building in Figure 5.7d would have matched regardless of the different position and with a different light condition, the streetlight caused the template matching to fail. However, many of the image patches could be matched even under varying conditions because the primary structures remained visible, see for example the patch in Figure 5.7b.

5.4 Robust Detection and Tracking of 2D Annotations

In the previous section, we showed a system that matches small image patches in order to define anchor points of 2D annotations against a panoramic image that is created in a background thread. Once detected, the image patches are precisely augmented as the user's orientation is tracked using the approach presented in Chapter 3.

However, the results from the user study indicated that there are problems that are primarily caused by changes in lighting due to weather conditions or the changing position of the sun. This in large part due to the fact that the proposed solution relies entirely on vision-based matching, and is therefore susceptible to temporal variations such as shadows or vegetation changes. Furthermore, all matches are treated independently, as no prior knowledge is used to optimize the template matching, or to place annotations based on previous matching results.

Another problem that was not evident in the field trials, but in several tests afterwards, is related to the fact that the camera in current generation mobiles phones continuously adjusts the exposure settings. This leads to visible artifacts in the final panoramic image, which is not only used for tracking purposes, but also to extract the image patches while creating an annotation and for matching the image patches while browsing previously created annotations. Consequently, these artifacts can affect the results achieved with the vision-based matching of image templates.

Altogether, the results achieved are not sufficient for using this approach in an application requiring that annotations should be reliably reproduced for extended periods of time.

While the participants of the field study commented that they were always able to reinitialize the vision-based orientation tracking, they reported an occasional loss of tracking. This loss of tracking results from the absence of features to track in the current camera image or a fast movement which causes keypoint detection to fail due to motion-blur in the image. Again, this was cause by the fact that we relied only on vision for orientation tracking.

In the following, we present an approach that tries to overcome some limitations of the previous approach by improved algorithms tackling the low re-detection results under conditions where the lighting changes between creation of the annotations and retrieval of the annotations for browsing. Additionally, we present algorithms that improve the tracking stability in case of fast motions or in case of camera images that contain a low number of features.

We will start by presenting an approach that allows for the creation of panoramic images with an extended dynamic range, which can better represent the wide variety of illumination conditions found outdoors and overcomes the problem of automatic exposure adjustment of current generation mobile phones. We further research the use of internal orientation sensors to prime an active search scheme for the anchor points, which improves the matching results by suppressing incorrect assignments.

Finally, we show how to statistically estimate a global transformation that minimizes the overall position error of anchor points when transitioning them from the source panorama in which they were created to the current view represented by a new panorama. This step considers multiple hypotheses for association of anchor points to known candidates and as a result, further suppresses wrong associations. Once the global transformation is computed, it also helps to transform those annotations into the user's view that could not be matched reliably.

Once the anchor points are re-detected, we track the user's movement using a novel 3DoF orientation tracking approach that combines vision tracking with the absolute orientation from inertial and magnetic sensors. This fusion improves tracking performance even under fast motion and provides important input for initialization of the visual tracking component.

5.4.1 Extended Dynamic Range Panoramic Maps

The existing approach for matching the annotations' anchor points used image templates that define an annotation's anchor point, which are then matched against a panoramic image created in a background thread. Thus, the whole approach relied heavily on good image quality because even small artifacts could affect the template matching. Unfortunately, the automatic exposure adjustment in most current generation smartphones affects the quality of the panoramic images created with our approach. Automatic exposure causes the mobile phone camera to constantly adjust the exposure to compensate for local changes in brightness. On most platforms, this feature cannot be adjusted or set to a constant value, causing the resulting panoramic image to contain jumps in brightness at the boundaries where contributions of several images are stitched together (see Figure 5.8 top).

These discontinuities create artificial gradients that affect the template-based matching of the annotations' anchor points. One solution to suppress discontinuities within the panorama would be to set the camera to fixed exposure, or to set that exposure rate dynamically from the application, which would also allow for creating images with true dynamic range. However, to our best knowledge, there is only one API that allows that adjustment: The Frankencam API [1], which allows adjusting the camera on Nokia N900 phones and some selected NVidia Tegra devices.

Consequently, we had to create a different approach that allows us to improve the image quality that does not require specific camera drivers and thus can be used on all common smartphones. Such an approach should also be fast enough to perform the required improvements in real-time so as not to affect the performance of the panorama tracker.

The idea of our approach is to align the luminance of each frame that is about to be mapped onto the panoramic image so that it matches the surrounding area in the panoramic image. We therefore use the pixel estimates of the first frame that is mapped onto the panoramic image as a baseline for all further adjustments. We used FAST keypoints that are computed for each frame that is about to be mapped onto the panoramic image, and the FAST keypoints in the panoramic map to estimate the differences in intensities.

Because these FAST keypoints are computed as part of the tracking pipeline (see Chapter 4), they do not require any additional steps. We compute the differences in intensities by averaging the difference of intensities of all matched keypoints found in the camera image and the panoramic map. The resulting average difference is then applied

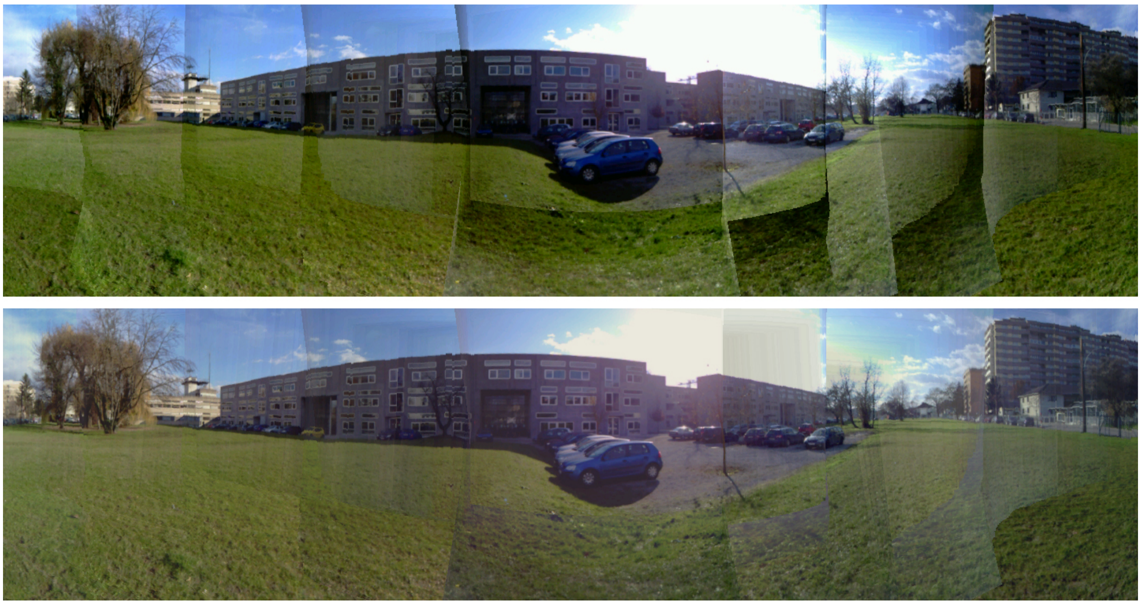


Figure 5.8: (Top) A panorama image containing visual artifacts, which are caused by the automatic and continuous exposure adjustment of current mobile phone cameras. (Bottom) A panorama image that was created by extending the dynamic range during mapping onto the panorama and applying a tone mapping afterwards.

to each pixel of the camera frame before it is mapped onto the panoramic image. As a consequence of this algorithm, we needed to extend the dynamic range that each color channel could hold. We empirically found that 16 bits per channel are enough to avoid clipping errors due to the limited dynamic range that could have resulted from adjusting the pixel values. The final panoramic image is computed by applying linear tone mapping that reduces the depth in each channel back to 8 bits per channel.

The resulting panoramic map can be seen in Figure 5.8, which shows that by using simple assumptions, we are able to remove the most severe artifacts caused by dynamic exposure adjustment. At the same time, the approach created only a minimal overhead, so we are fast enough to maintain the overall speed of the panoramic tracker creating the panoramic image.

5.4.2 Sensor Fusion for Improved Re-Detection

Current generation smartphones regularly include GPS, compass, accelerometer and recently even miniature gyroscopes. The accuracy of these sensors is usually inferior to a well-tuned visual tracking technique, but non-visual sensors are complementary because of their robust operation. We therefore integrated the compass and the accelerometers to create a better re-detection of annotations.

The improved re-detection is achieved by narrowing the search area for the vision-based template matching using the information obtained from the internal sensors. The

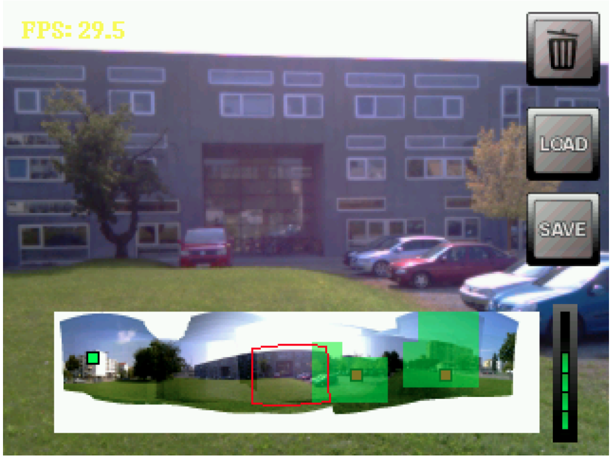


Figure 5.9: The green patches indicate regions that are likely to contain annotations' anchor point based on the sensors prior checking. We therefore adapt the NCC score to be less restrictive compared to other regions

region in the panorama where the annotation is likely to be located-based is determined based on a direction estimate from the internal sensors. The panoramic map is created at a resolution of 2048 x 512 pixels from 320 x 240 pixel sized camera images. A typical camera has a field of view of $\sim 60°$, so the camera resolution is close to the map resolution: 320 pixel / 60° times 360° = 1920 pixels. The theoretical angular resolution of the map is therefore 360°/2048 pixels = 0.176 degrees per pixel. Assuming a maximum error of the compass of $\sim 10°$, we can expect to find the annotation in a window of 57pixels around the estimated position. We consider an area three times larger than this window, but weight the NCC score with a penalty function proportional to the distance from the active search window (see Figure 5.9). Thus we only consider matches outside the primary search area if they have a very good matching score.

5.4.3 Matching Annotations Using a Global Transformation

In the previous approaches, the annotations were considered independently of each other during the re-detection. Thus, the detected position of an annotation was not used to optimize the re-detection of other annotations. Moreover, empirical analysis revealed that the main reason for wrong results from the NCC template matching came from situations with more than one good match for one annotation. This led to the problem that single annotations could not be detected reliably or were detected at the wrong location, whereas other annotations were robustly detected at the correct spot. This situation calls for additional geometric verification.

We approach the problem by considering the annotations in the source panorama (the panorama which was used to create the annotations) as a set for which a consistent geometric estimate must be achieved. Therefore, the detection is extended by the requirement to find a global transformation T, which maps the set of annotations from the source panorama onto the target panorama (representing the current environment) with a mini-

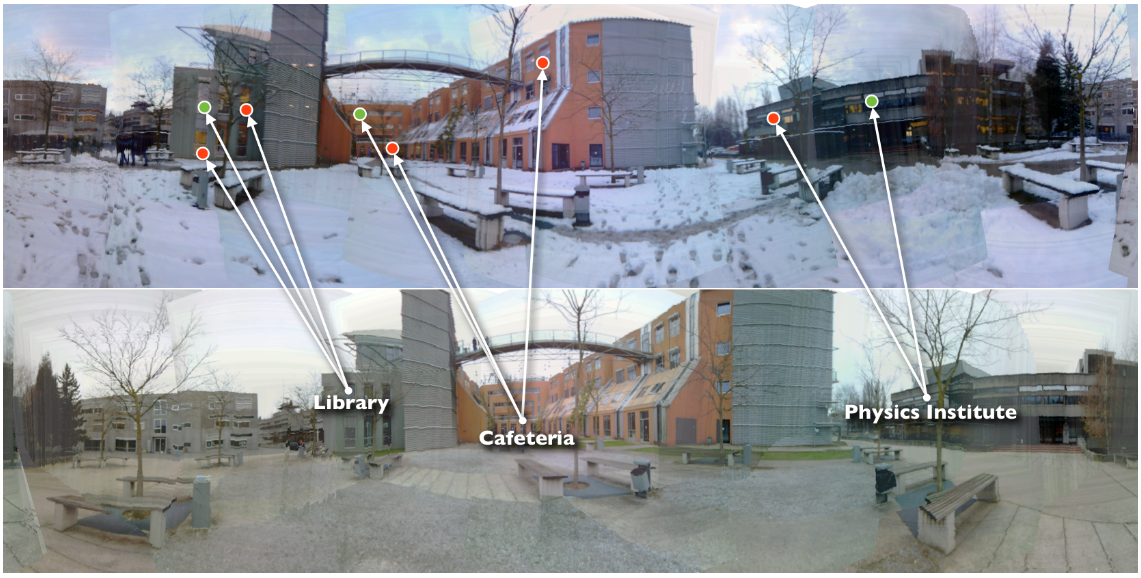


Figure 5.10: A source panorama that was used to create the annotations. (Top) A newly created panorama with the best candidates for placing the annotation resulting from the template-based matching. For every annotation anchor point, we store a maximum of three best matches. The green dots in the upper image have the best matching scores and are therefore used for label placement. The red ones are the second and third best matches of an annotation, which makes them a candidate for a possible correct match.

mized average error. As we assume the panoramas to be taken at the same position, the transformation is a pure rotation, aligning source and target panorama with three degrees of freedom.

To compute rotation T, we describe the position of an anchor point in the source panorama by representing anchor coordinates as a 3D vector from the camera position to a point on the cylindrical panorama (see Figure 5.10). We extended the workflow as presented in Section 3.2 to also store this 3D vector together with the image patch for each annotation. This dataset describing the annotation is uploaded to a remote server and tagged with the GPS address of the current position. We do not upload any panoramic images, as only this dataset is required to redetect the annotations. The size of the dataset is in the range a few kilobytes ($\sim$ 2 kilobytes for the image patch plus the 2D information), so it can be easily handled via a 3G connection.

Once a user approaches a place where annotations were created, the mobile phone accesses the closest datasets based on the GPS position. We take into account that GPS can be inaccurate and therefore we download all datasets that were created within proximity of 50m. After downloading the datasets, the anchor points are redetected using the template-based matching, and annotations are initially placed using the best match. But instead of using only the best match, we also keep the best three candidate matches

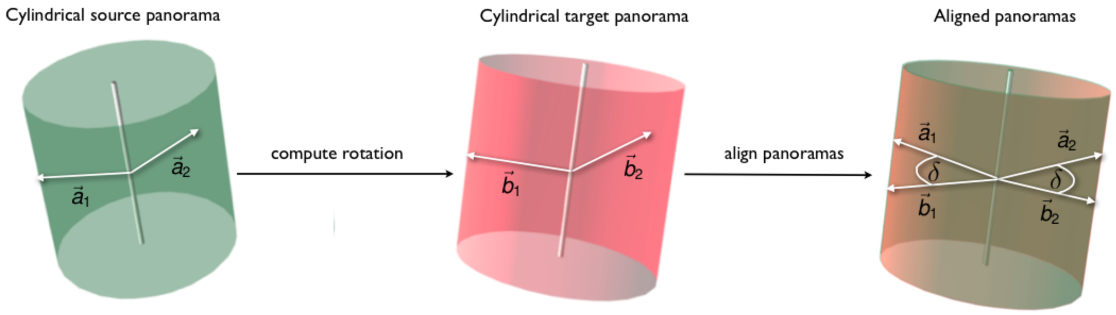


Figure 5.11: Illustration describing the alignment of two cylindrical mapped panoramas based on the position of the annotations anchor points. The two vectors $\vec{a}_1$ and $\vec{a}_2$ are pointing to two annotation positions in the cylindrical source panorama. The middle cylinder describes a panorama, which is created on the fly on the smartphone. The vectors $\vec{b}_1$ and $\vec{b}_2$ are pointing to two possible annotation positions in this new panorama. Rotating one cylinder into the other in order to align both vectors of each cylinder using absolute orientation with an error δ, results in a rotation, which can be used in a RANSAC calculation to determine a model with a sufficiently small error.

based on NCC score for later use. For all found candidate matches, we compute the vector-based position in the target panorama as we did for the original annotations in the source panorama.

While online tracking and mapping continues, a RANSAC based approach running in a background thread determines and updates a global rotation T. This rotation aims to optimally map the set of all annotations from the source panorama to the target panorama by aligning the panoramas.

We randomly select two annotations and one of their three best candidate positions in the target panorama as input for finding the best rotation using RANSAC. To find the best match, the rotation T between the two coordinate systems is calculated so that two vector pairs $\vec{a}_1$, $\vec{a}_2$ and $\vec{b}_1$, $\vec{b}_2$ can be aligned while minimizing an L^2 norm of remaining angular differences. We use the absolute orientation between two sets of vectors [49] to compute this rotation. The resulting rotation is the hypothesis for the RANSAC algorithm. All annotations are mapped to the target panorama using the current estimate for T, and the difference between the resulting 2D position in target map space and the annotation position found through template matching is determined. If the distance is below a threshold, the annotation is counted as inlier and its error is also counted as inlier. Its error is then added to an error score.

For a hypothesis with more than 50% inliers, a normalized error score is determined by dividing the raw error score by the number of inliers. The normalized score determines if the new T replaces the previous best hypothesis. This process is repeated, until a T with an error score below a certain threshold is found. Such a T is then used to transform all annotations from the source to the target panorama. Annotations for which no successful match could be found can now also be displayed at an appropriate position, although with less accuracy because their placement is only determined indirectly.

Obviously, the source and target panorama are never taken from the exact same position, and the resulting systematic error can affect the performance of the robust estimation. We empirically determined that a 50% threshold for inliers and a 10 pixel threshold for the normalized error score in 2D map coordinates yields a good compromise between minimizing overall error and reliable performance of the RANSAC approach. Finding the best rotation to align the two panoramas requires about $\sim$ 30ms for eight annotations. However the panoramas are not aligned each frame because it is only necessary to update the model once new candidates for annotations' anchor points are detected based on the vision-based template matching.

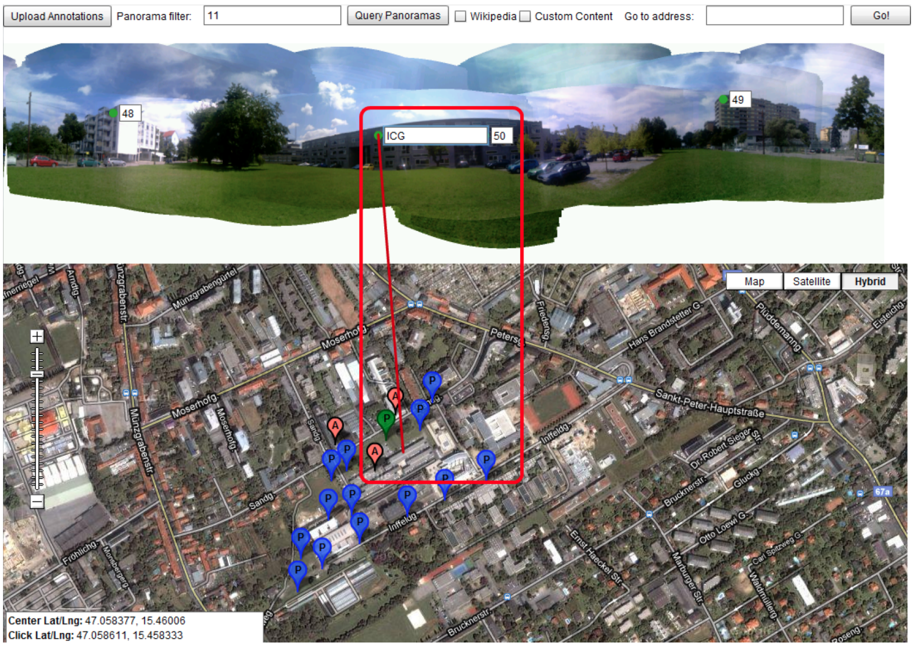
5.4.4 Application

We implemented a prototype as part of a campus-wide information system, where users can create and share information about buildings and other objects in an outdoor environment. The prototype's main difference was a revised redetection algorithm compared to the system presented in Section 5.3.2. We therefore left the interface, consisting of a map-based navigation interface and an AR-based browsing interface that are both handled via touch screen, unchanged as it had already proved its good usability in our previous user study.

However, we revised the web-based server interface to give the user additional functionalities. The previous approaches used the web-based server infrastructure presented in Chapter 4 mainly for storing the content on the Internet and accessing it using content filters based on GPS position, user name, content type or size.

For our new final application, we extended the server by creating an interface that allows the user to manage existing textual annotations, add new textual annotations, and also to create new textual annotations based on existing sources such as Wikipedia or other sources. The main purpose was mainly to have a visual debug tool, which gives full access to the stored content on the server. However, it could also be used in practice as a complementary desktop interface used in parallel with the already existing mobile interface. In that context, it could be used to view 2D annotations created with our in-situ approach while not in place (similar to a Google Streetview interface), or to add new or modify existing data when not in place. The interface uses Adobe Flash and is accessible in a normal web browser.

The web interface is composed of two parts: The lower part shows a Google Map view indicating positions where users created a panoramic image to place 2D annotations. The upper part shows the created panoramic image in case one of the points in the map view was clicked (see Figure 5.12). The general workflow for using the system follows: The user navigates in the Google Maps view to the position where she wants to add or modify a 2D annotation. She can only add or modify annotations at positions where a user has already created a panorama using our smartphone application. These positions are highlights with a blue "P" on the Google Maps view. The user can select the position she wants to work with by clicking the blue "P" in the view. Once selected, the web interface will show the selected panorama in the upper part of the screen. If there are any annotations linked at that position they will show up in the panoramic image (see Figure 5.12).



Panorama filter: 11

Center Lat/Lng: 47.058377, 15.46006
Click Lat/Lng: 47.058611, 15.458333

Figure 5.12: Screenshot of the web-interface of the server-located database. Beside displaying existing 2D annotations it allows users to add new, and modifying existing, content. We used it primarily to verify and check the 2D annotations created on the mobile phone.

The user can now add new annotations by clicking on the panoramic image at the position where she wants to create a new annotation. Similar to the smartphone-based application, the system extracts an image patch, in the background, around the annotation's anchor point, which is later used for redetecting the annotation's anchor point. The user can type in textual information, or add textual information to existing annotations. It is further possible to link the annotations not only into the panorama but also to a GPS position. To link to a GPS position, users drag the annotation's anchor point from the panoramic image to the position on the map where y she wants to link the annotation (see Figure 5.12). The GPS position of the annotation within the current panorama is displayed with a red "A" in the Google Maps view.

Besides creating new annotations, it is also possible to link existing data to the panoramic image. We demonstrated this with Wikipedia articles. If this feature is enabled, GPS-tagged Wikipedia articles appear on the Google Maps view. The GPS-tagged Wikipedia articles are provided by the Geonames geographical database[1] and are depicted with a white "W" in the Google Maps view. The user of the web interface can then drag these Wikipedia articles onto the panoramic view. This causes the application to create a new textual annotation with the headline of the Wikipedia article as text. This is only a limited example, but it shows how existing GPS-tagged content can be integrated in our system using the web interface.

[1] http://www.geonames.org

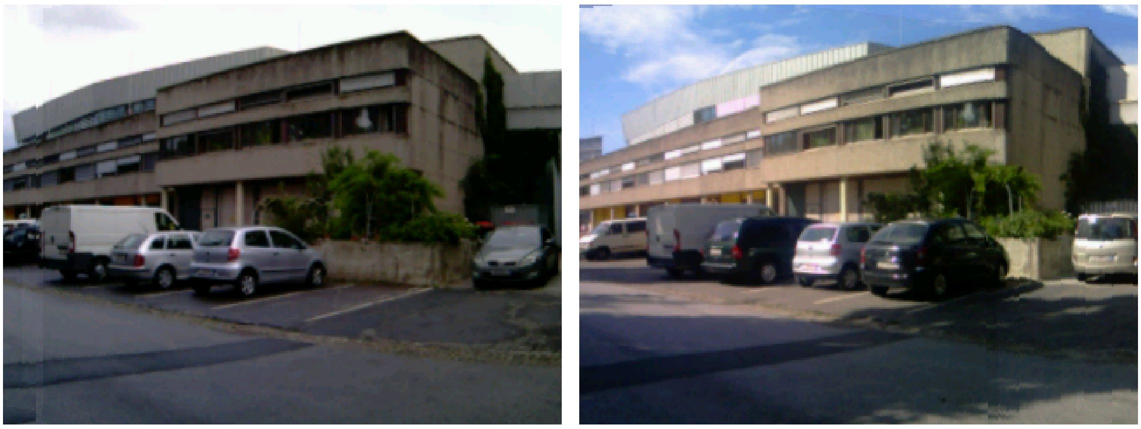


Figure 5.13: Fragments of two panorama images showing the different environment conditions during the evaluation.

5.4.5 Experiments and Results

We implemented and evaluated our approach on a common smartphone (HTC HD2) as part of a campus information system. During the evaluation, we focused the re-detection rate used for detecting the annotation anchor points.

To test the re-detection performance, we created 12 panoramas at different positions on our campus, with the goal of obtaining a diverse set of images and environmental conditions (see Figure 5.13). The average distance between these panoramas was 50m. For each panorama we created four to six annotations, leading to 58 annotations in total. For better comparison, we created panorama images using both the extended dynamic range approach and using the standard 8-bit dynamic range. We proceeded by attempting to match the collected annotations against new panoramas created from the incoming video stream.

As our approach requires the user to be at the same position from where the annotations and the source panorama were created, we evaluated the matching performance within a 2m radius to the original position. As GPS was sometimes inaccurate, we had a case where at one position two annotated spots were assumed to be within 50m, which resulted in the application downloading the datasets of both annotated spots and choosing the one that achieved the highest scores in the NCC-based template matching for further processing.

The evaluation procedure was set up so that all combinations of re-detection enhancements were systematically tested. The baseline system without any enhancements resulted in a re-detection rate of about 40%, which is less than reported previously [12] because of the more difficult environmental conditions used to create the data sets. The results are summarized in Figure 5.14. The sensor fusion improves re-detection by about +15%, to a point where the RANSAC approach for determining the global transformation finds enough inliers, so that the combined sensor fusion and global transformation technique delivers an 86% re-detection rate. The extended dynamic range representation seems only effective in

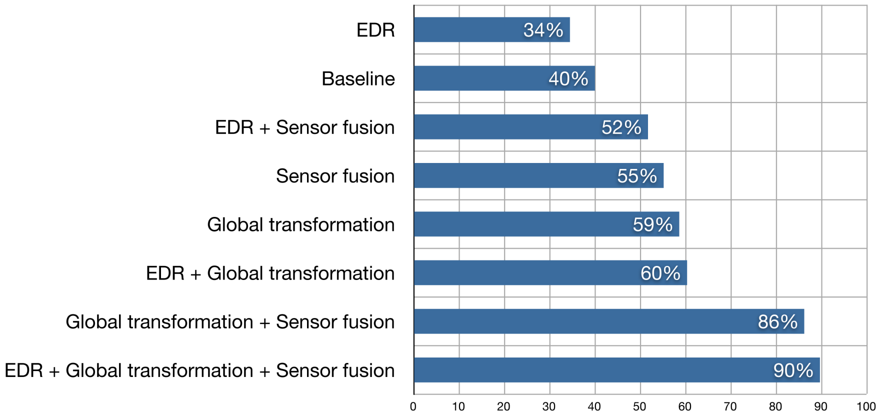


Figure 5.14: Evaluation results when analysing the re-detection performance.

improving results that are already very good a bit further, while extended dynamic range applied alone to difficult situations can even slightly reduce matching performance. However, the combination of all three enhancements leads to an overall re-detection of 90%, which is more than twice the original performance and probably satisfactory for everyday operation.

5.5 Conclusion

In this chapter, we presented a series of incremental approaches for detection and tracking of 2D annotations in mobile AR applications that all rely on a panoramic image of the environment created in real-time. The approaches presented allow users visiting the same location to share textual annotations augmented in the live camera view. The annotations created by the first user are detected in the view of the second user with one of the following approaches:

- Loading a prerecorded panorama and matching it against the current camera view to transfer the annotation's anchor point defined in the prerecorded panorama.
- Matching image patches to define an annotation's anchor point against a newly created panorama of the environment.
- Using improved template matching that uses a smaller search area, and additionally applies geometric constraints to verify positioning.

Once the annotations are detected, we track the user's orientation using a reliable hybrid tracking approach, which allows us to correctly augment the annotations in the live camera view. We showed that the presented approaches outperform previous approaches in terms of robustness and accuracy, yielding a 90% re-detection rate even under strong temporal variations in the environment.

All presented approaches rely on a similar interface for creating and sharing the textual annotations on a mobile phone. In addition to the fact that the approaches presented deliver better tracking and re-detection performance than most existing approaches, the interface implemented also proved in user tests to be easy to handle. Furthermore, the interface allows users to create textual annotations in-situ and in a spontaneous manner, two things that were not possible with the existing approaches.

We further demonstrated a web interface that extended the client software to allow people to create or modify content when they are not in-place. It further allows for the integration of existing GPS-tagged content into the system so that the users can tap into a bigger source of existing content.

The approaches presented here are generally applicable to outdoor AR, but more specifically improve smartphones, which have rather low quality sensors and limited computation power for computer vision.

Chapter 6

Situated Multimedia for Mobile AR Applications

Contents

In this part of the thesis, we focus on multimedia content for mobile AR applications. In particular, we present prototypes that demonstrate the use of audio, video and 3D content and investigate how such content can be created and used as an additional source of information within mobile AR applications.

On the one hand, audio and video content are commonly used multimedia formats, and today's Internet is hard to imagine without services like YouTube[1], Vimeo[2] or Soundcloud[3]. However, AR browsers and most other existing mobile AR applications do not make use of this existing content, or these media types in general. On the other hand, 3D content is, though common in many desktop AR applications or AR games, not often used in mobile AR applications such as AR browsers. After all, the current generation of mobile AR applications does not fulfil one of the key requirements for Augmented Reality 2.0, namely integration of rich and interactive content (see Chapter 3).

In this chapter we present solutions investigating how audio, video and 3D content can be utilized in a mobile AR system and show how audio, video and 3D content for AR can be created on a mobile device. This chapter is based on, and summarizes, our work presented in detail by Langlotz et al. [65] [69] [67].

We start by introducing the concept of Audio Stickies that, similar to physical sticky notes, can be precisely put on objects or places in the environment to add information. However, in this case the added information uses auditory modalities rather than written

[1] www.youtube.com

[2] www.vimeo.com

[3] www.soundcloud.com

text. We discuss our implementation and use-cases and give information on the user study we conducted on Audio Stickies.

We continue this by introducing an approach for using video material within mobile AR applications. We report on the recording and creation of these AR videos and give details on our implementation of an application to allow users to replay them in place. Results of a user study with domain experts demonstrate the usability and usefulness of our approach. Finally, we present an approach for the creation of 3D content within mobile Augmented Reality along with a preliminary study on the feasibility of our approach. We conclude this chapter by summarizing our core findings and give an outlook on other application scenarios beyond those investigated.

6.1 Visually-Guided Spatial Audio Annotations

Smartphones are a ubiquitous element in today's lifestyle. Increasing numbers of people use them to communicate with each other, browse the Internet, or listen to music while walking in the city. Mobile technologies are also used passively for location-based services (e.g., tourist information systems) and actively for leaving tags or geo-cached items. While using this wide range of applications, users come into contact with various ways of representing information including text and graphics, but also audio and video data. While these forms of information presentation play an important role in today's internet and in social platforms, they are not as common in Augmented Reality applications. The majority of research and commercial applications still focus on overlaying textual annotations or three-dimensional content.

In the following, we present Audio Stickies as a novel way of implementing augmented spatial audio in an outdoor environment. Similar to written sticky notes, the goal was for users to be able to place Audio Stickies precisely in their environment as a means of asynchronous communication. Users can leave Audio Stickies at certain, precise positions, and other users can browse them by pointing their mobile phone towards visual hints representing the Audio Stickies. The visual hints are augmented in the user's view, indicating that an audio annotation was placed there (see Figure 6.1). To experience a seamless augmentation of the environment, precise and stable registration is crucial. This applies to visual augmentation as well as to augmented spatial sound and, therefore, also to our Audio Stickies.

6.1.1 Spatial Sound with Audio Stickies

Similar to the previously demonstrated systems for displaying 2D annotations (see Chapter 5), we rely on the panorama-based tracker introduced in Chapter 4 to achieve for a precise tracking of users' orientation in outdoor environments. Using the panorama tracker, we built an application that includes the capability for recording and playing back Audio Stickies.

While using the application, the user "looks through" the mobile device and experiences an Augmented Reality view of the current environment. To create and place an audio annotation, the user must specify the point where she wants to place the audio annotation by touching the mobile phone's touch screen. As with textual annotations (see

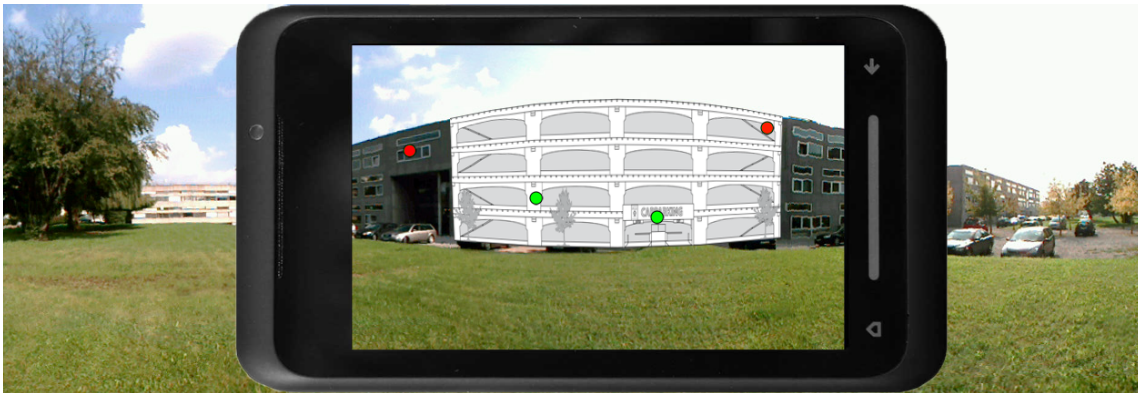


Figure 6.1: Concept image of the Audio Stickies browser: an architectural design alternative is spatially laid over an existing building. Colored dots indicate the position of user-created Audio Sticky comments that can be heard by pointing the center of the phone's screen towards them. The green dots indicate that the current Audio Sticky comments are played, while the red dots indicate that they are currently out of focus.

Chapter 4), Audio Stickies are stored in relation to a panorama coordinate system. Therefore, we had to transform the currently selected screen coordinate via the current tracking information into the corresponding coordinate in the panorama coordinate system. This can be achieved by casting a ray onto the cylinder that represents the cylindrical mapped panorama (see Figure 6.2).

Once the coordinate of the selected point is determined in the panorama coordinate system, a small widget is shown on the screen. The widget allows the user to record an audio comment. The recorded comment is then stored and referenced to the selected position. We limit the maximum length of each audio annotation to 10 seconds and interactively show the remaining time with a progress bar. Once created, the Audio Sticky can be shared and browsed by other people visiting the same spot, in the same way one can browse textual annotations (see Chapter 5).

To activate and perceive the Audio Stickies, the user browses the AR view by moving the mobile phone. In each frame, we cast a ray r from the center of the screen via the panorama cylinder into the panorama to compute the focus-point R (the center of the currently visible camera image) of the user in panorama coordinates (see Figure 6.2). Once R is determined, we compute the direction vector d_n and the distance from all Audio Stickies A_n to the focus-point R. We play only those sounds for which the distance to the focus point is below a certain threshold (see Figure 6.2). Depending on the threshold, it is possible that several Audio Stickies can be played simultaneously.

We use the distance and direction to the focus point from the Audio Stickies to adjust the volume and position in the stereo channels. Consequently, Audio Stickies closer to the focus point (the screen center) play louder. Moreover, the position in the stereo channels corresponds to the position on the screen. Audio Stickies placed to the right of the focus

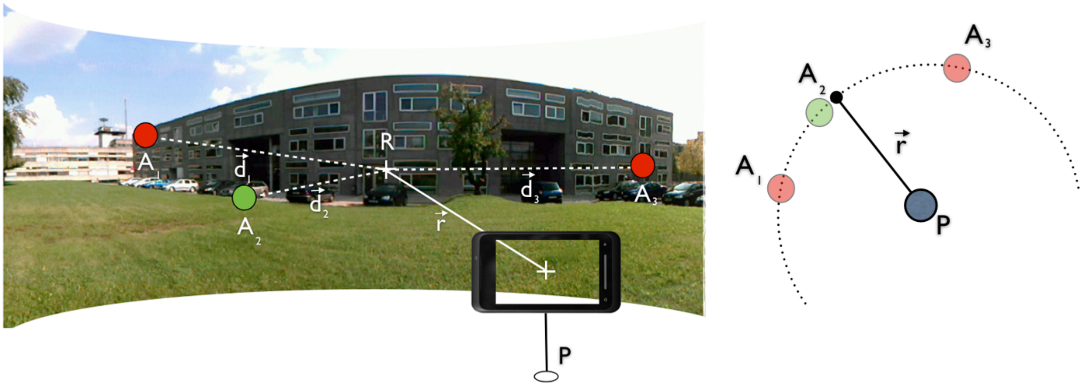


Figure 6.2: Illustration of the panorama-mapped Audio Stickies: (Left) User at position P browses Audio Stickies (A_1, A_2, A_3) in the environment. The current focus point R is determined by casting a ray r from screen center onto the panorama of the environment. The volume and the position in the stereo channel of each Audio Sticky are determined by analysing the vectors (d_1, d_2, d_3) pointing from the focus point to each Audio Sticky. (Right) Top down view illustration.

point appear louder in the right stereo channel. Based on the recommendations in [145], we adjust the threshold so that only up to two sound sources are played at maximum volume in order to suppress for audio clutter.

We also make use of additional visual cues to guide the user's view to Audio Stickies in her current environment. We augment the user's view with visual dots at the position of Audio Stickies. Along with providing visual guidance, the dots also support the control of the Audio Stickies' playback. Audio Stickies that are currently playing are shown with a green dot, while inactive Audio Stickies (determined based on their distance from the user's focal point) appear as red dots in the user's view. Once the user looks towards a red dot, the dot turns green and the Audio Sticky starts to play in a loop.

6.1.2 Implementation

Although implementing sound on a stand-alone, non-AR PC system is a rather trivial task, implementation on a smartphone required a number of special accommodations. First, a sound engine that works with the limited capabilities of a smartphone is necessary. This requirement includes the need for 3D support, or at the very least controllable stereo sound. The sound engine should be able to play several sounds simultaneously, because multiple users may wish to place multiple Audio Stickies in the environment. Playback delay should not affect interactive real-time performance and the engine should have a relatively small memory footprint, which allows program and multiple sound datasets to be held in memory at the same time. Finally, it should support certain sound file formats to achieve a balance between size and quality.

Several sound engines, audio storage formats/audio codecs and audio qualities/bitrates were considered and tested (native implementation on Windows Mobile, iAuxSoftSFX[4] and Hekkus Sound System[5]). Ultimately, we used the native API for recording sound and the Hekkus Sound System for playing sound, because the other options failed to meet our criteria (e.g. iAuxSoft has a big memory footprint). While the Hekkus Sound System is generally suitable, it does not fully support 3D sound like other systems such iAuxSoft do. We simulate 3D sound by using the sound panning between the left and the right channels and adjusting the volume settings. This technique is known as amplitude-panned sound sources [147]. Even though the technique does not simulate physical 3D sound, it is accurate enough for our purpose.

We determined empirically that a sampling frequency of 27kHz was sufficient, as the main purpose of the Audio Stickies was to record human voices. We stored the recorded audio files as WAV files, since playing several compressed (mp3, ogg vorbis) files simultaneously caused a noticeable delay due to the limited computing performance of smartphones. The final application performed with an average 25 frames per second on a Toshiba TG01 and on an HTC HD2.

6.1.3 User Study

We tested the feasibility and usability of our prototype system and approach with a user case scenario in a controlled field study, which is described in the following.

6.1.3.1 Scenario and Setting

There is a plethora of possible application scenarios. Almost all active, mobile location-based services, i.e., AR browsers and other applications where users leave text or graphics in-situ can benefit from Audio Stickies. Combined with social networking services, this facility could lead to versatile ways of synchronous and asynchronous communication and collaboration.

To evaluate our system, we chose a case scenario involving public participation in urban planning. Instead of filing formal written proposals, our system allows the capture of immediate spoken feedback, hence improving public participation in the planning process. We suggest that this participation is best achieved in-situ, where what is to be built can be viewed in its proposed context. Virtual architecture is visually overlaid over real urban scenes and citizens are asked to leave audio feedback on design alternatives.

In our scenario, we chose a location which is to be redesigned as part of a bigger urban redevelopment, where existing buildings are to be demolished and replaced by new buildings. Public consultation regarding this type of project usually involves only textual descriptions illustrated with design sketches (if at all). Sometimes, if a building project is of great interest large models made of wood and paper are provided for the people to comment on. Virtual flythroughs maybe provide in addition.

Our approach is to display new building designs in their actual context using Augmented Reality. We visually augment planned buildings onto the environment, overlaying

[4] www.iauxsoft.com

[5] www.shlzero.com

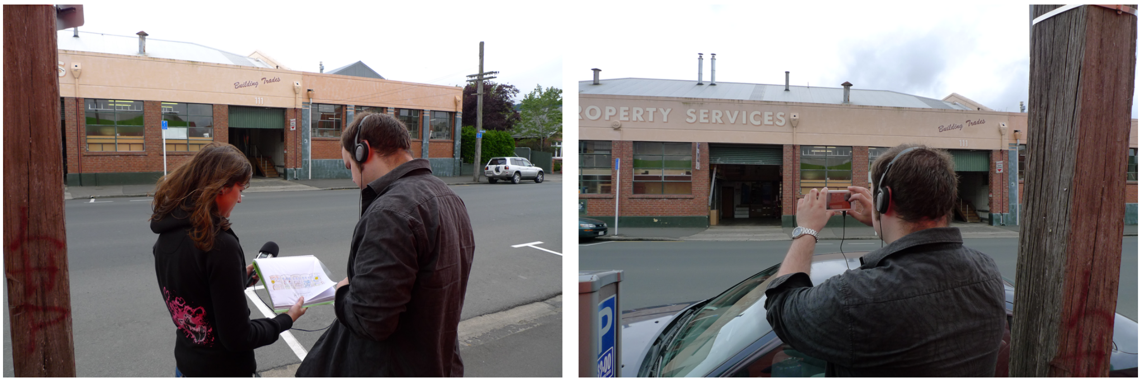


Figure 6.3: Participant conducting the pilot study: (Left) Participant looking over the paper-based drawings and listening to the recorded audio comments provided by the instructor. (Right) Participant using a mobile phone to actively browsing the environment for with augmented buildings and Audio Stickies from previous users using the mobile phone.

existing buildings in order to give a more realistic idea of how new buildings will blend with their environment.

Interested parties are given the immediate opportunity to provide feedback using Audio Stickies. In this way, people using the system can comment on buildings, while they are still in the planning phase. The system also allows users to place their Audio Stickies precisely on the objects on which they want to comment (e.g., elements of the façade). The whole system is implemented using off-the-shelf smartphones, which will allow for wide dissemination in the future, whether by experts or the general public.

For our usability study, we asked participants to browse planned buildings that are visually augmented onto the real environment. Simultaneously, they were invited to use Audio Stickies to comment on planned buildings and to share what they liked or disliked (see Figure 6.3). Participants wore headsets (headphones with an integrated microphone) connected to a mobile phone, in the same way they might when listening to music on the go. In this way, they were able to retrieve Audio Stickies created by previous users. Consequently, the number of collected Audio Stickies accumulated over time.

An initial pilot study with nine participants was conducted in a busy street next to the main campus of the University of Otago in Dunedin, New Zealand. For the pilot study, 3D models of the planned buildings resembled fictional sketches. As part of the pilot study, we also aimed to compare our prototypical Audio Stickies with a paper-based approach. Therefore, we showed participants printed designs of the buildings, the same designs that were visually augmented within our prototype (see Figure 6.3), while at the same time playing comments using an audio recorder

This situation proved realistic and challenging due to the noise of cars and pedestrians engaged in conversation nearby. Noise affects the perceived quality of the audio annotations placed by participants. However, even in a relatively noisy location, the ambient noise was reported to be acceptable, and we were able to use our prototype. Our pilot participants helped us to find flaws and positively commented on the general usability of the prototype. Overall, our system did well, and was found to be comparable to the paper-based approach, but achieved higher ratings in terms of usability and also in terms of precise localization of the audio comments.

6.1.3.2 Experimental Design

The user study was undertaken in two different environments: on the aforementioned busy street in Dunedin and in a contrasting quiet area on a university campus in Graz, Austria. This allowed us to estimate the influence of environmental noise and distractions on the usability of our system. Fifteen participants were recruited for each site (30 total). None of those who took part were experts in Augmented Reality or in Augmented Audio. Twenty-two participants were male (73.3%) and eight female (26.7%); the age range was 21 - 50 years (M=28.44, SD=6.7).

Each participant was given a demonstration that explained how the prototype worked. They were then allowed to try the system and encouraged to ask questions.

Participants were asked to browse four virtual 3D-models of the planned buildings for the site and allowed to listen to the Audio Stickies recorded by previous users. The models represented very different use cases of the buildings, including a car park, a food court, a teachers' college and a student accommodation house. An architectural designer created all the virtual building models that were used. We decided to create discussion-provoking designs to stimulate comments. The experimenter told the participants that the virtual buildings were possible candidates to extend the current environment as part of an ongoing Master Plan.

Participants who wanted to record any of their comments using Audio Stickies were able to do so by touching the model (screen) at the appropriate position (also see Figure 6.4). The number of Audio Stickies increased with each subsequent participant. To initialize the system for the first participants, the experimenter created two comments for each building design and used these as a starting point for discussions.

While the participants were using the application, the experimenter noted any observations. The session ended when the participants had browsed all of the augmented buildings and did not want to place any more Audio Stickies. They were allowed to browse through the augmented building prototypes and audio comments back and forth for as long as they wanted.

After the participants finished commenting on the different parts and aspects of the building alternatives, they were asked to answer questions from a questionnaire using 7-point Likert-like scales ranging from 1 = "strongly disagree" to 7 = "strongly agree". The first part of the questionnaire contained demographic questions. The second part contained questions specific to the usability and usefulness of the prototype with items from a questionnaire developed by Lewis [71] together with some questions specific to our scenario. The questions were followed by a short interview to try and elicit information regarding any problems or difficulties the participants experienced.

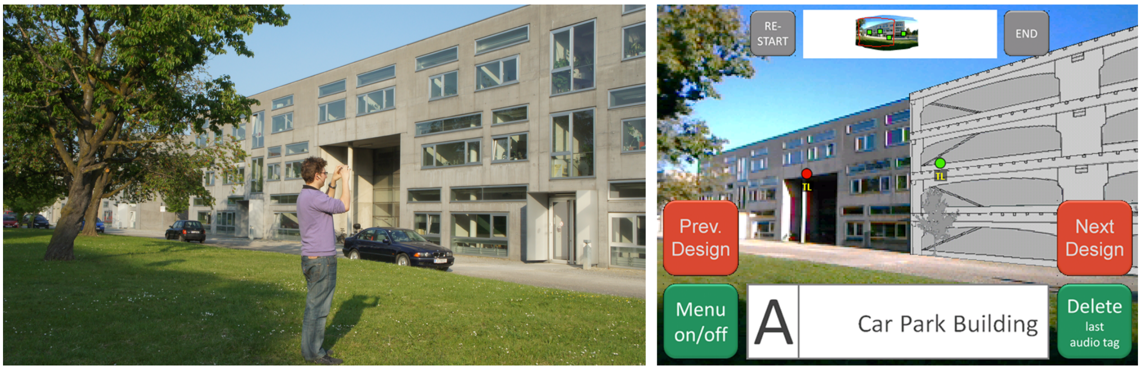


Figure 6.4: (Left) Participant of the user study while creating an Audio Sticky. (Right) Screenshot of the user interface showing existing Audio Stickies and an augmented building they can comment on. Tapping the screen at the designated position creates a new Audio Sticky.

6.1.3.3 Results

All participants successfully finished the experiment and browsed the four proposed building designs together with the existing Audio Stickies. While they were not required to generate a specific number of Stickies, almost all participants created one Audio Sticky for each building.

Generally, the Audio Stickies were used to express users' opinions on certain aspects of the architectural design (e.g. "The planned facades of the building are mainly from concrete, which does not integrate well with the mainly green environment. Therefore, I wish that the architects rethink the use of more natural materials like wood or natural stone"). Other comments included the use of the buildings or the desire to see specific features (e.g. "I like the idea of adding a parking garage to the university campus, but I hope that the university also remembers to reserve some space in the building that can be used to drop bikes"). Others also used the Audio Stickies to comment on previous messages ("I agree to the other comments that the architects should use more natural materials and that the university should take care of existing and new green area").

The data gathered from the user study showed that audio annotations are seen as a useful source of information (M = 5.72, SD = 1.25, Figure 6.6(f)). While in Graz the average answer to the question "The audio tag environment was acoustically very cluttered" was 2.14 (SD = 1.03, Figure 6.6(h)), the participants in Dunedin scored 4.07 (SD = 1.58). Scores for audio and visual clutter changed with the increasing number of sessions (and increasing number of Audio Stickies) for both test sites with a different rate (also see Figure 6.5).

Participants in Graz answered the question "The ambient noise was very distracting" with 1.79 (SD = 0.80, Figure 6.6(m)), whereas participants in Dunedin answered this question significantly differently with 3.87 (SD= 1.64) (t-test, p = 0.0002).

The participants answered that while using the system, they could "easily identify the links between the audio tags and parts of the buildings" (M = 5.21, SD = 1.66, Figure

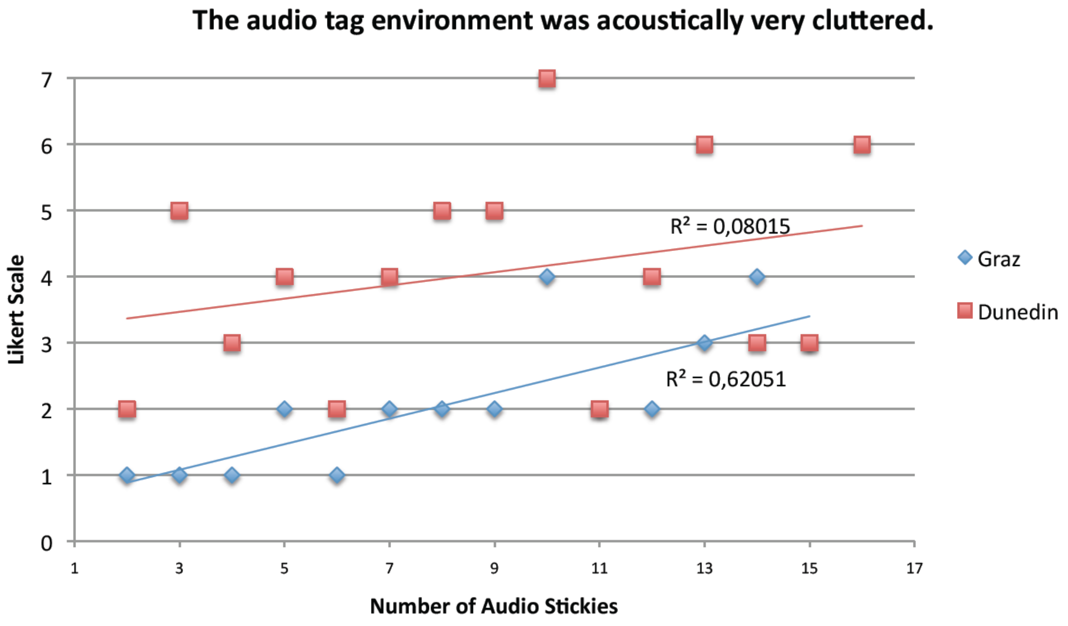


Figure 6.5: Feedback regarding the perceived audio clutter dependent on the number of Audio Objects with 1 =”Not cluttered at al”, 7 = ”Very cluttered”.

6.6(j)), but there was a difference in how easy it was to control (Graz M = 6.21, Dunedin M = 4.93, Figure 6.6(k)) and discriminate between Audio Stickies (Graz M = 5.57, Dunedin M = 4.60, Figure 6.6(l)).

Besides the differences between the two locations, the participants agreed that the system was easy to learn (M = 6.21, SD = 0.94, Figure 6.6(d)) and easy to handle (M = 5.79, SD = 0.82, Figure 6.6(a)). This led to the conclusion that the participants solved the task efficiently (M = 5.66, SD = 0.86) (see also Figure 6.6(b)).

Audio tagging as implemented in this scenario was regarded as useful (M = 5.90, SD = 1.18, Figure 6.6(g)). Similarly, seeing the planned buildings in their current context was noted to be very useful (M = 6.45, SD = 0.78, Figure 6.6(n)). Users found such utility was supported by stable and precise tracking expressed by their degree of agreement with the statement ”the models were accurately augmented in the environment” (M = 5.62, SD = 0.94, Figure 6.6(e)).

Several participants reported that they felt uncomfortable hearing their own voices, a feeling that many of us know from video or audio recordings of ourselves, and which is known to be a result of bone-conduction.

The increasing number of audio annotations did not seem to significantly affect the perceived visual or audio clutter, the ability to localize the sounds, or the perceived controllability of the system.

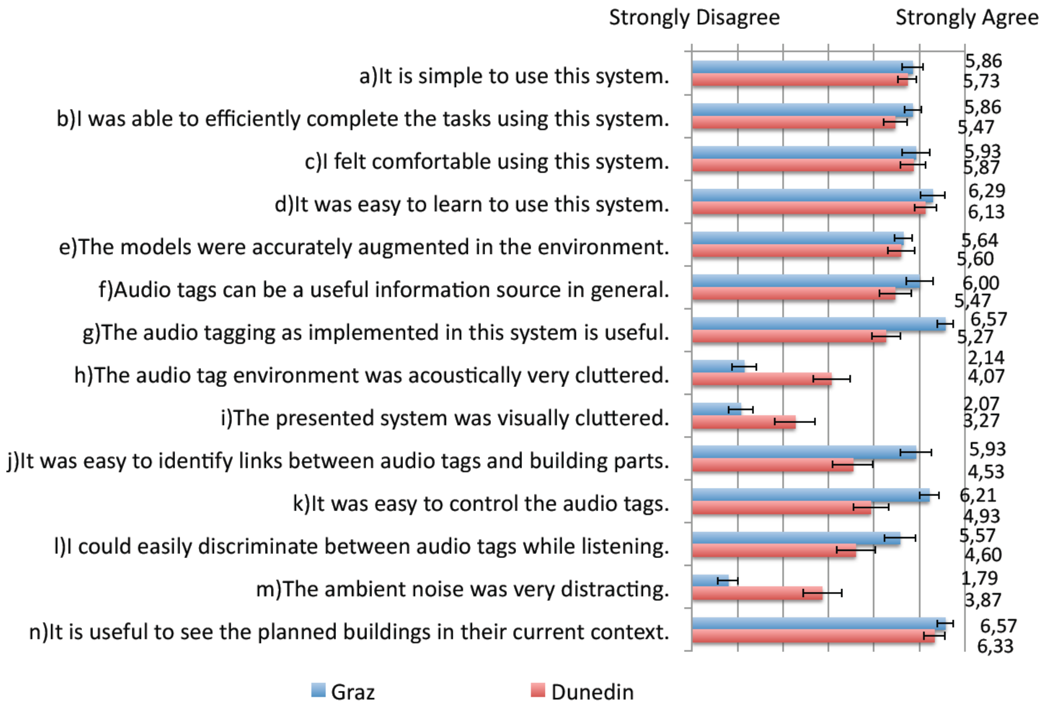


Figure 6.6: Questionnaire results of user study on 7-point Likert-like scales (1 = "Strongly disagree" to 7 = "Strongly agree").

6.1.3.4 Discussion

The gathered data showed that audio annotations are perceived as a useful source of information. Participants from both sites agreed that the system was easy to learn and easy to handle.

They also reported that the tracking was perceived as precise, stable and fast, which allowed for a seamless integration of the augmented buildings and Audio Stickies into the real world.

The participants from both locations reported that even if many Audio Stickies were present, it was still easy to control the annotations by looking towards them, and to discriminate between different Audio Stickies. Regardless of the amount of Audio Stickies and ambient noise, all participants were able to identify the links between the Audio Stickies and their related objects. There was a significant difference between the two locations in terms of perceived audio distractions. In Graz, participants did not report disturbing ambient noise. In Dunedin, however, all participants commented on how noisy the environment was.

While the results support our approach and the implemented prototype, it shows that audio clutter can still be a problem, especially if many Audio Stickies are present. For both locations the average results were acceptable, but it was noticeable that users performed slightly worse with an increasing amount of Audio Stickies. The difference for both locations is likely caused by the different amount of environmental noise that adds to the audio clutter. Audio clutter seems to be a general problem in the domain of Augmented Audio and deserves more research.

Overall, participants reported positively on our approach of using precisely placed Audio Stickies as a general information source and as a natural way of interacting with the environment. We demonstrated that Audio Stickies also work in rather noisy environments. Furthermore, the user interface was suitable even for novice users. The vision-based tracking system worked seamlessly and did not cause any problems for the participants.

6.2 Situated Compositing of Video Content in Mobile Augmented Reality

The availability of inexpensive mobile video recorders and the integration of high quality video recording capabilities into smartphones have tremendously increased the amount of videos being created and shared online. With more than 50 hours of video uploaded every minute on YouTube and billions of videos viewed each day, new ways to search, browse and experience video content are highly relevant.

However, current user interfaces for online video tools mostly replicate existing photo inter-faces. Features such as geo-tagging, or browsing geo-referenced content in a virtual globe application such as Google Earth (or other map-based applications) have primarily been directly reproduced for video content.

In this work, we investigate how we can offer a new user experience to a mobile user through compositing the user's view of the real world with prerecorded geo-referenced video content. Similar to MacIntyre et al. [79] [**?**], we are interested in extracting the salient information from the video (e.g., moving people or objects) and offering the possibility to spatially navigate the video (by rotating the phone) mixed with the view of the real world. In contrast to their work, we focused on mobile platforms in outdoor environments and also looked at offering simple ways to record and capture this type of video content with only a minimal input. We also have fewer restrictions during the recording, as we support rotational camera movements and do not rely on green screen-type technology for recording the video augmentations.

In the following we present our interactive AR technique, which offers accurate spatial registration between recorded video content (e.g., people, motorized vehicles) and the real world with a seamless visual integration (e.g., a recorded break dancer overlaid on live camera video). Our system allows users to replay geo-referenced video sequences, in order to re-enact a past captured event for a broad range of applications covering sports, history, cultural heritage, or education. We support a variety of tools for the user to control video playback and apply video effects, hence delivering the first prototype of a real-time AR video montage tool for mobile platforms (see Figure 6.7).

Our system thereby operates in three steps. The first step (”record”) is the shooting of the video, including geo-tagging and uploading to a remote server for further processing (or social access).

In the second step, we extract the object of interest in the video frames, which we later augment in place. This pre-processing task can be performed remotely (server hosting

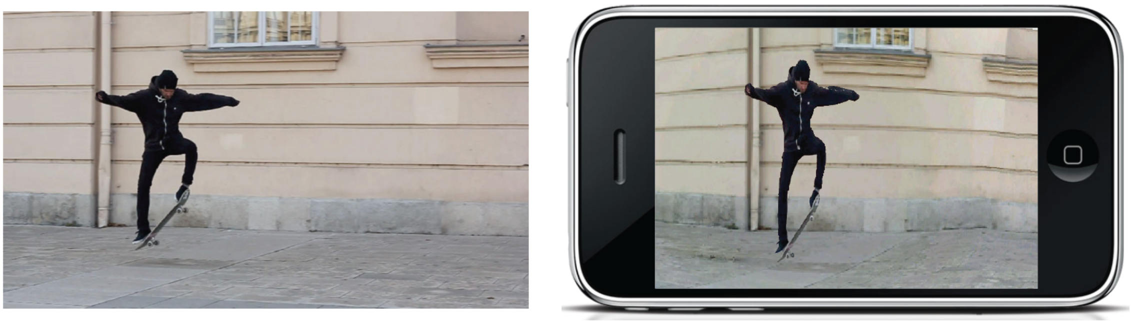


Figure 6.7: Illustration of situated video augmentations. (Left) Original video footage recorded using a mobile phone. (Right) Illustration of the augmented video application. The foreground video object - in this case the skateboarder - is augmented in the users view.

the video) or can be done locally (desktop PC). We apply a segmentation that only requires that the user outlines the object of interest in the first frame. We also extract the background information of the video and assemble it into a panoramic representation of the background, which we later use for the precise registration of the video content into the environment.

The final step of the system is the "replay" mode. This mode is enabled once a mobile user moves close to the position where a video sequence was shot. The system then downloads the previously created information. While the user explores the environment, the video is registered using computer vision and augmented into the user's view. The proposed system contributes to the field of Augmented Reality by demonstrating how to seamlessly incorporate video content into outdoor AR applications and by allowing end users to participate in the content creation processes.

In the following, we give a detailed description of our approach and describe the algorithms used for our AR video compositing technique.

6.2.1 Video Shooting ("Record")

The video capture is performed using standard video devices such as a smartphone or a digital camera (compressed, high definition). Our approach requires the camera location to be fixed while recording, however rotational movements of the camera are possible. The recorded video is then geo-tagged with the user's current GPS location for later use, and uploaded to a cloud-based server or transferred to a personal computer.

6.2.1.1 Offline Video Processing

In this step, we process the video to extract relevant information. The main challenge here is to separate the object of interest in the video (foreground) from the remaining information such as the background or other moving objects that are not of interest. We later use the object of interest as an overlay, but we also want to keep the background information as it is needed to register the video overlay onto the new scene. This preprocessing can be

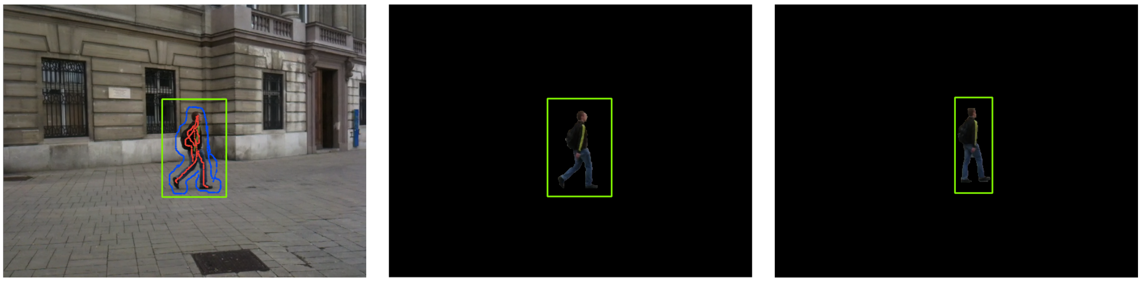


Figure 6.8: User-initialized video segmentation. (Left) Manual initialization of the segmentation step. User sketches the foreground object (red) and outlines the background (blue). (Middle) Result of applying GrabCut segmentation to subsequent video frames: Segmented foreground object and size-optimized texture (green outline). (Right) Tracking the segment using Lukas-Kanade Tracker allows segmentation of later frames even in cases when the appearance changes.

done on the mobile device, on a personal PC, or on a cloud server hosting the uploaded video (e.g., on YouTube as part of the video conversion and analysing step).

Foreground Segmentation We start segmenting the video by applying a variation of the GraphCut algorithm, namely GrabCut, presented by Rother et al. [118]. To initiate the algorithm, the user has to roughly sketch the object of interest (the foreground object) and mark some of the background pixels on the video image (in practice close to the foreground object, see left Figure 6.8). Then GrabCut delivers a segmentation of the video image, which separates the foreground object from the background.

This approach was developed for static images, but in this case needs to be applied to each frame as we are using temporal content. To avoid the cumbersome task of marking every individual video frame manually, we extended the method in a similar way to the approach presented by Mooser et al. [89]. The idea is to use the segmentation output of the GrabCut algorithm of the previous video frame to initialize the segmentation computation for the current frame.

Because there is likely movement of the object of interest between the two frames, we cannot naively apply the result of the segmentation from the previous frame to the current one. We address this issue by estimating the position of the segmented foreground object in the current frame by computing the optical flow of pixels between the previous and the current frame using the Lucas-Kanade algorithm [75]. This gives us an approximation of the foreground object's position in the current frame. We also dilate the estimated foreground object's footprint to compensate for tracking inaccuracies.

We compute the boundary of the estimated foreground object, and select pixels within (pixels of the foreground object) and outside (background pixels) that boundary to use as input for GrabCut. Applying this approach for each frame yields foreground objects for all consecutive frames of the video. We apply a dilate and erosion operation on the segmented foreground objects to remove noisy border pixels and only keep the largest connected component as foreground object in case the segmentation computed more than one segment. Contrary to the approach of Mooser et al. [89], we do not consider the edge saliency for tracking the object within the camera frames and also do not apply a banded graph cut-based segmentation. However, we also kept an option to manually initialize the GrabCut for specific frames in case the object of interest is not segmented properly.

The segmented foreground object is often only a fraction of the size of the full video frame (see Figure 6.8). To reduce the data we store the foreground object by only saving the bounding rectangle around it and its offset within the video frame.

Background Information Once the foreground object is extracted, we also need the background information, which we use for registering the object into the view of the user. We take the segmented frames and focus in the following only on the background pixels.

Due to the possibility that a user will rotate the camera while recording the video, the recorded frames will potentially hold different portions of the scene's background. Furthermore, the foreground object also occludes parts of the background, reducing the amount of visual features that are later available for vision-based registration. As we want to reconstruct as much background information as possible, we do not only take into account the background information from one video frame, but from all frames, and integrate them into one panoramic image.

We create this panoramic image keeping the background pixels by using a modified version of the panoramic mapping and tracking approach presented in Chapter 4. In the original implementation, we used features in the incoming video frames to register the frames and stitch them into a panoramic image. We also demonstrated how to track the camera motion R_S of the recording camera while constructing the panoramic image. Additionally, we assumed the camera movements were only of rotational nature.

We adapted the technique of panoramic mapping and tracking to handle alpha channels and to only map pixels onto the panoramic image that are considered to be background pixels. The resulting holes in the panorama caused by the occluding foreground object, can be closed by later frames of the video as the foreground object moves within the camera frames, revealing occluded background information in later frames (see Figure 6.9). We store both the resulting panoramic image that contains the background information of the video, and the camera rotation R_S for each video frame.

All this information, including the segmented video, the panoramic image holding the back-ground information and the camera rotation for each frame, and GPS geo-location, is packaged into a specific data structure and saved in a compressed file. This packaged dataset can be easily shared online and made available via a cloud repository.

6.2.2 Online Video Processing ("Replay")

For this step, we assume that the user's phone is also equipped with GPS similar to the recording device. We can query video augmentations located in the vicinity of the current user's location. The augmentations can be retrieved from local storage or online from a cloud repository. Once the data is decompressed, we start registering the video into the current user's view.

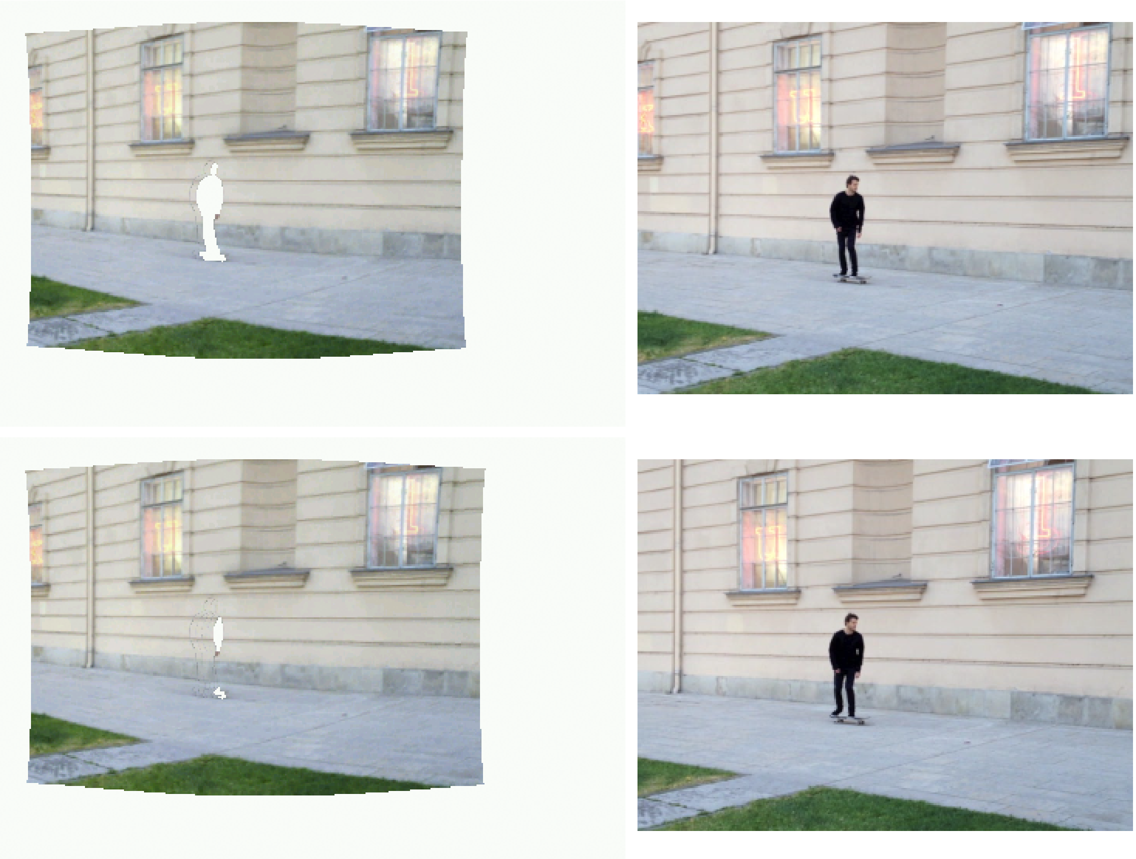


Figure 6.9: Creating a panoramic image containing only background information. (Top) Holes that are caused by occlusion are closed by adding the background pixels from later video frames (Bottom). The right side shows the latest video frame that is used as input for the panorama computation.

For this purpose, we also use the panoramic mapping technique from Chapter 4: We build a new panoramic image from the current camera feed and also track the current camera rotation R_T.

The use of the panorama-based tracking allows for greater precision in the registration and the tracking, as we do not rely on noisy sensor values. As mentioned earlier this comes with the drawback of supporting only rotational movements. However, most users only perform rotational movements while using outdoor AR applications (see Chapter 3), making this constraint acceptable in most scenarios.

While building the new panorama of the environment, we try to match the loaded panorama holding the background pixels against this newly built panorama. The matching is performed using a point feature technique (in our case with Phony SIFT [153]). As soon as the overlapping area, the area holding image information that is in both panoramas, is large enough, the matching using Phony SIFT should succeed and provide the

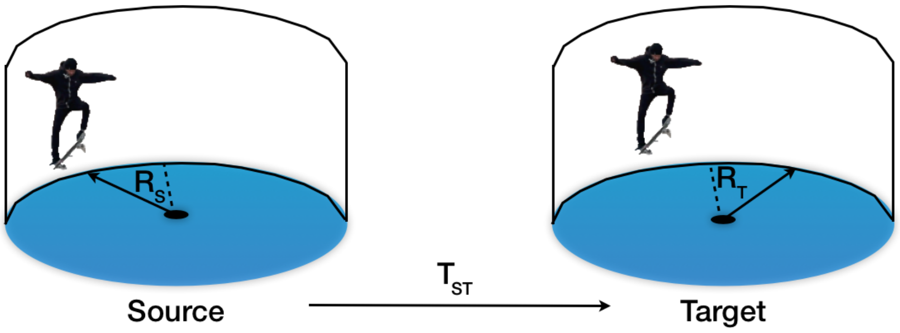


Figure 6.10: Illustration of the applied transformation between the source camera and the target camera used for replaying the augmented video.

transformation T_{ST} describing the relative motion between the camera used to record the video (the source camera S) and the camera where the video information should be registered (the target camera T).

By assuming that the user of the system is roughly at the same position where the video was recorded (identified via GPS), we can constrain the transformation T_{ST} to be purely rotational (see Figure 6.10).

Using this transformation T_{ST}, we can transform each pixel from the source panorama into the target panorama and vice versa. This allows us to play the video information by overlaying the current environment with the object of interest from the video frame. We therefore load the video frames and apply for each video frame the combination of the transformation R_S (the orientation of the source camera computed in the offline video processing step), the transformation T_{ST} (the transformation between the source and the target camera gained from the registration) and the transformation R_T (the orientation of the target camera computed using the panorama-based tracking), which allows us to precisely augment the video content into the user's view (see Figure 6.10).

Please note that by using the panorama-based tracker, we obtain an update of the transformation R_T at each frame. This allows us to rotate the target camera completely independently from the orientation of the source camera and to maintain the precise registration of the video in the current view.

6.2.3 Prototype

Video augmentations can be used for a wide range of applications covering areas such as entertainment or education if integrated into outdoor AR applications such as AR browser systems.

We implemented our technique in a prototype of a mobile video editing AR application. Inspired by the current tools proposed in desktop video editing applications, we focused on some of their major features: video layers, video playback controls and video effects. In the following we present an overview of the user interface and the post-effects implemented in our system. We also describe a case study of our technique.

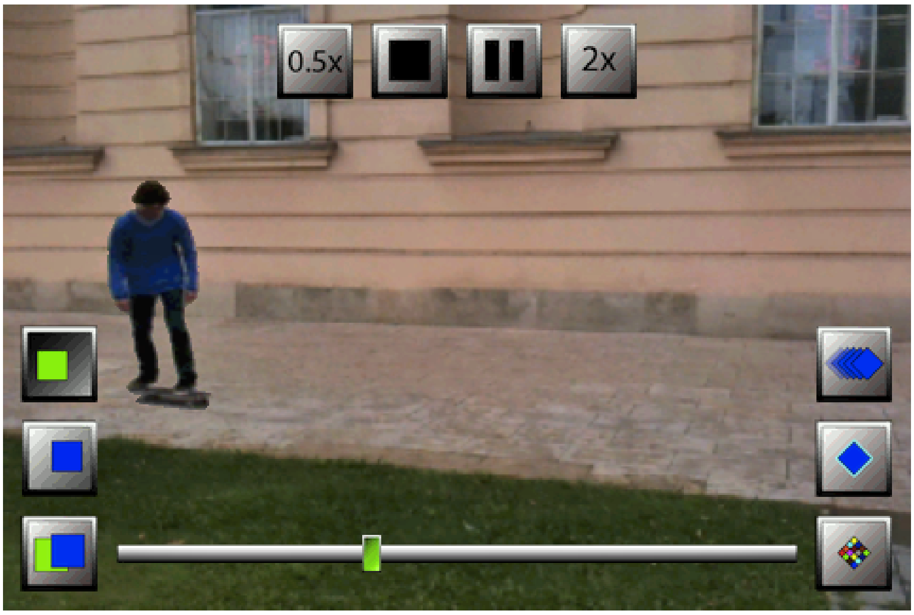


Figure 6.11: Screenshot of our prototype showing a video augmentation together with the control groups for video control (top and bottom), video layers (left side) and video effects (right side).

6.2.3.1 User Interface

The interface of our prototype is inspired by the design of graphical user interfaces generally found in video editing tools. We created three different groups of functions distributed around the screen that can be operated with a touch screen. The control groups of our user interface are (illustrated in Figure 6.11) the video control group, the video layer group, and the video effects group. When the menu options in our interface are not used for a certain time (in our case 5 seconds), the menu disappears to give more space to the video, but will be redisplayed as soon as the user touches the screen.

The functions in the video control group consist of the playback control found in most video players: play control buttons, time slider and speed buttons. In addition, we added the possibility to slow down or increase the speed of the currently played video. The video layer group provides access to the different video sequences accessible at this GPS location, organized in different depth layers. The end-user can control and activate/deactivate these different layers, which can be played independently of each other or simultaneously.

The last group of items, the video effects, trigger real-time video effects that can be applied to the different video sequences. Each effect can be switched on or off and the effects can be combined.

6.2.3.2 Post Effects and Layers

Applying visual effects is an important part of video post-production. Effects are used to highlight actions, and create views that are impossible in the real world, such as slow motion. Normally these effects are applied to the video material in a rendering step that

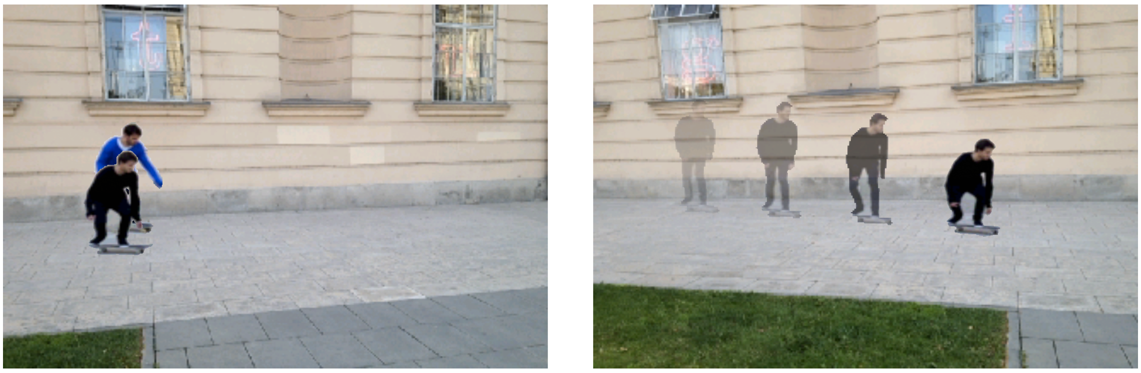


Figure 6.12: Examples of layers and an example of realized post effects as used in our skateboard tutor application as captured from an iPhone 4. (Left) Playing back two video augmentations layers allows the comparison of the riders' performance. (Right) Flash-trail effects visualize the path and the motion within the video.

is carried out offline [73].

Because of the nature of our approach, we are able to perform a wide variety of these video effects in real-time on a mobile device without the need to pre-render the video. We explored space-time visual effects such as multi-exposure effects, open flash, and flash-trail effects. Multi-exposure effects simulate the behaviour of a multi exposure film where several images are visible at the same time. We can easily simulate this behaviour for cameras with a fixed viewpoint by augmenting several frames of our video at the same time. This results in the subject appearing several times within the current view, such as in a multiple exposure image.

An extension of this effect is the flash-trail effect. This effect also displays multiple instances of the same subject, but the visibility of each instance depends on the amount of time that has passed (see right Figure 6.12). This effect supports a better understanding of the motion in the recorded video. We implemented the flash-trail effect by blending in past frames of the augmented video with increasing amounts of transparency. Thereby, the strength of the transparency and the time between the frames can be freely adjusted

Our approach allows us to play back more than one video at the same time (see left Figure 6.12). We can compare actions that were performed at the same place, but at different times, by integrating them into one view, thus bridging time constraints. Each video corresponds to a video layer, and the user can switch between these layers or play them simultaneously.

Other visual effects that can be enabled are glow or drop-shadow variations that can be used to highlight the video object. In the case of several video layers playing at the same time, the glow effect can be used to highlight a certain video layer. All these effects do not require any preprocessing and are carried out on the device while playing back the video. Therefore, they can be combined or switched off on demand.

6.2.3.3 Skateboard Tutor Application

We demonstrated our system to end-users as part of a skateboard tutoring application. Skateboard videos are a good representative of dynamic real-world content naturally evolving in our real-world environment. The different ranges of manoeuvres performed with a skateboard (tricks) are largely bound to the environment and location through natural or artificial obstacles and ramps. Skateboarding videos also make use of a variety of camera shooting techniques (perspectives, movement, optics) necessary to accommodate the dynamic of the skateboarder moving through the real environment. Tutorial/how-to videos represent a large portion of YouTube videos [128] today, which confirms the potential of this format for teaching skills or competences. Skateboard tutorials (> 30.000 hits on YouTube) serve therefore as a good application of our technique.

Our skateboard tutoring application allows recording skateboard tricks that can be shared with other users for demonstration and learning purposes. The application can be used to overlay the pre-recorded video content (extracted skateboarders) in place, allowing users to replay and experience the tricks and actions performed by another user (or from a previous day) in the correct context (see Figure 6.13). This application can support the learning process because online skateboard videos, generally recorded with a fish-eye lens, frequently give a distorted perception of the skateboarder in the real environment.

Our test skateboard videos were recorded with normal digital cameras or smartphones and are processed using our approach of situated video compositing for AR. We also make use of the proposed post effects and layers. The layer approach allows recording skateboard manoeuvres on the fly, which can later be played back in parallel with other stored manoeuvres for comparison (e.g., speed, height of jumps). The flash-trail effect can be used to highlight the motion and the path of the rider.

We implemented the offline video processing using OpenCV[6]. The mobile skateboard tutoring application has been implemented on the iOS platform using the Studierstube ES framework (see Chapter 4). We tested the application successfully on an Apple iPhone 3GS, iPhone 4S and an iPad2. Across these devices, the application runs in real-time between 17fps (Apple iPhone 3GS) and 28 fps (Apple iPhone 4S/iPad2).

6.2.4 Evaluation

We conducted a preliminary user study for gathering first user feedback on our technique as well as for identifying flaws, areas for improvement, and additional ideas for the applicability of our technique.

6.2.4.1 Scenario and Setting

Producers of skateboard videos are usually also consumers, leveraging the possibility to collect feedback for both the creation of video augmentations and the consumption of video augmentations. We therefore invited skilled skateboarders (domain experts), who had experience in creating skateboard videos or tutorials and who published their videos online via popular sharing platforms.

[6] www.opencv.org

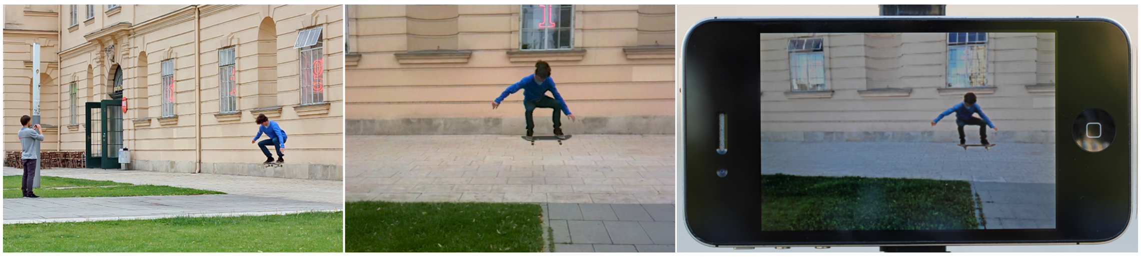


Figure 6.13: Scenario as used during the user study. (Left) Skateboarder was recorded with a mobile phone while performing his actions. (Middle) Frame of the recorded video sequence. (Right) The same action as augmented within our skateboard tutoring application as captured from an iPhone 4.

Our main objective with the user evaluation was to identify the usefulness and applicability of our approach as well as the usability of our created prototype. In total we had 5 expert users with $>$ 7 years of skateboarding experience (all male, 25-28 years), all of them were involved in producing skateboard videos, some produced videos for marketing. All considered themselves as not tech-savvy. Two have minimal knowledge of AR, though none had any experience with any kind of AR application. One participant stated he was not very familiar with using mobile devices as he restricted his usage of mobile phones to placing calls or write messages.

6.2.4.2 Procedure

We gave all participants the chance to get hands-on experience with our prototype, which we demonstrated on both an iPhone 3GS and an iPad2 (see Figure 6.14). After introducing the project, we gave them a short demonstration of the application, showing the different features. They were able to test the integrated effects, and to try the video layers by playing two video layers that were augmented at the same time. We selected two participants to create their own skateboard videos that were later augmented, while all other users only had the chance to experience the augmented videos. After the participants finished trying out the prototype, we asked them a series of questions as part of a semi-structured interview.

In the interview all participants confirmed that our prototype was easy to use. Only one user didn't feel really comfortable while using the device and interacting with the application; this was the participant not experienced with smartphones. We also asked the participants about the social aspect of using the application outdoor in a busy area, and participants replied that they were really comfortable in this regard. All users commented that our system was easy to learn, and the current interface was well received.

When questioned about usefulness, the participants all scored our application as really useful. They confirmed our hypotheses regarding the difficulties of perception and understanding when using traditional online videos, and the viability of our in-context video augmentation. Three of the five participants said that they enjoyed the freedom of having control of the camera orientation during playback, as it was not relying on the

Figure 6.14: Evaluation of our prototype with domain experts using an Apple iPad2.

orientation of the recording camera (fixed in traditional video recording techniques). They commented positively on the ability to play several videos/layers at the same time that are overlaid in parallel. They described it as really useful for comparing their own runs with the tutorial video to detect differences. They also liked the flash-trail effect, and stated that this effect seems to be useful for studying ”the line” a rider skates.

When asked about the general applicability and the usefulness of experiencing video augmentations in place, the participants generally responded very positively on potential usage in other application areas. However, two of them pointed out during the interview that in order to use the application successfully users must visit the site of the original action, which makes more sense in specific cases. Therefore, both of them stated that in general they view the application more as a gadget, and could not extrapolate other convincing use cases. As we presented them other possible use cases at the end of the interview (city guides, parades/events within the city) they answered that they could also see potential in this kind of application, but needed to experience it before providing a reliable assessment.

The last part of the interview focused on the visual quality of the technique, in terms of spatial and visual integration. Three participants had the feeling that the scene and the rider were three-dimensional, which gave a sense of ”authenticity”. One participant perceived the rider to be two-dimensional, but the scene to be 3D. The remaining two participants stated that it was all overlaid in 2D. They all commented that the movement of the augmented skateboarders within the scene was very realistic. Even after being explicitly asked, they could not remember seeing any drifting between the augmentation and the background. However, when asking about the seamless visual integration, we received more mixed answers. They stated that sometimes the skateboarder did not retain the same appearance because the background was too dark or incorrectly lit. Two participants also noticed small segmentation errors (e.g., a wheel of the skateboard disappeared in a couple of video frames).

The two participants that shoot their own video reported the application was simple to use, and the additional step required was acceptable for the generated outcome. When asked about constraints in the camera motion during shooting (limited to rotational movements of the camera) they said that it is likely acceptable in most cases. They explained that the vast majority of people contributing to YouTube make short videos with smartphone devices from a single point of view. One of the participant said: ”The given constraints fit the medium, as I think the majority of the short online videos were shot in this [constrained] way”. Thus, our system fits all criteria generally used by laypersons recording skateboarding videos. Finally, during the open questions, one participant proposed the possible use of our application as a mobile blue screen, which would allow users to capture objects and scenes and assemble them together using the layer view.

6.2.5 Discussion

Overall the evaluation using our skateboard experts showed that our approach has advantages over existing mobile video applications (shooting, video effects, playback). However, the final outcome and degree of usefulness strongly depends on the use case. Even though all of our participants were not tech-savvy, they quickly mastered the use and handling of our prototype application.

A major limitation of our prototype pointed out by the participants was the visual quality of the overlay. Even though feedback for visual quality was above average, the users complained about a lack of visual coherence, stating that the video augmentation looked different from the current environment. In our case, this was mostly caused by cloudy weather at the time of recording resulting in low contrast actors, as opposed to the mostly sunny conditions during the playback of the video augmentations. This can be addressed in future versions of our prototype by implementing adaptive visual coherence. The basic idea is to compare the background panorama of the video with the current environment in order to adjust the video augmentation in terms of contrast and color.

Another problem was that the segmentation sometimes was not accurate enough, especially if applied to a well-structured background as required for vision-based registration. Although more sophisticated segmentation algorithms and better algorithms for tracking the segmented objects exist, they require more expensive computation or GPU implementations, and need to be investigated in the context of this work. Particularly because the segmentation as used in our system has requirements that conflict with the vision-based registration used in our approach. Namely, a less structured background achieves better results in foreground-background segmentation, while it causes difficulties when used to register the augmentation based on the background information.

Despite these drawbacks, our application showed that augmented video could be an interesting addition to AR especially because video content is often easier to create than 3D content

Professional applications can benefit from video augmentations as realized in our approach. Augmented reality-based tourist guides could display more interactive content e.g., by capturing the tour or guide for later replay. Furthermore, authoring such content is less demanding than creating dynamic 3D content. This allows users to easily create in-situ narratives similar to the concept of situated documentaries presented by Höllerer et al. [46].

Many augmented reality applications can benefit from the simplicity of creating video augmentations using our approach, which allows laypersons to create content and share it with friends. Our approach enables the creation of videos of certain events (e.g., parades, street artists, etc.) and playback in place at a different time.

Greater separation between the constraints used during shooting and those used for replaying can be explored further. Cinematography components such as camera type, camera movement, visual style of the image, location and its content are some examples of elements that can be altered, modified or "warped" between the record and replay. One can imagine recording a cyclist in the Tour de France with a rolling technique and replaying it fixed in another location. Furthermore, real-time montage with live video, online content, and collaborative editing can leverage the full potential of mobile AR.

6.3 Creation of 3D Content within Mobile AR Applications

Three-dimensional content is a widely used media in Augmented Reality applications. Furthermore, location-based systems or virtual worlds such as Google Maps or Bing Maps increasingly make use of 3D content, in particular virtual 3D models of real buildings or objects. However, despite the fact that this content is increasingly used, it is usually still created by artists in an offline process and in a desktop environment. In the following, we present our work on creating 3D content for mobile AR, which enables the potential for 3D content created in-situ. Our work particularly targets an inexperienced audience. Our goals is to enable users to actively participate in the content creation process by creating content in an AR environment, which can be useful for:

- Drawing 2D and 3D annotations, which can be used to add or highlight information displayed in AR.
- Creation of geo-referenced virtual duplicates of real objects, e.g., buildings, which can later be used in different visualization systems, such as a 3D data viewer. Furthermore, virtual duplicates can be placed in different environments at new locations or can be stored as 3D models for inventory.
- Creation of 3D models that can be used in AR-based games or for phantom rendering and occlusion handling.

The requirements of our system are that it must work in-situ, does not need a prepared environment, e.g., printed marker or existing features databases, and allows spontaneous editing 3D content while in place. In the following, we present our prototype by outlining the tracking component employed, describing the interface created, and giving a summary of a qualitative user evaluation we conducted.

6.3.1 Tracking

There are different scales in which mobile AR applications can be employed. In this work we differentiate between small working environments such as desktop environments and larger environments such as outdoor scenes. Small workspaces describe scenarios where the working volume is mostly bound to one specific object, which can nearly always be seen in the phone's camera view.

Larger working environments describe scenarios where the environment around the user contains many objects spread in a wider area that cannot always be seen. Both scenarios require different tracking techniques, which is why we treat them separately in this work.

6.3.1.1 Tracking Infrastructure for Small Working Environments

For small working environments, we decided to use a natural feature-based approach for tracking the smartphone's position relative to a dedicated target. Therefore, we employed the natural feature-based tracking system presented by Wagner et al. [153], because it works in real-time on smartphones, while also being robust enough to handle many common problems such as partial occlusions. This approach tracks the position of the phone using a database containing features of an object in the current camera view. Thus, it requires a feature database of the currently tracked target to be available on the phone.

As we assume an unprepared environment, we extended the tracking approach by Wagner et al. [153] in a way so that allows the users to create this feature database using the smartphone. The user has to point the phone running our prototype towards an object that should be used as a tracking target. As given by the approach of Wagner et al., this object should be planar or at least have a larger planar surface that can be used for tracking. The user indicates the planar region by sketching its outline and the system starts to analyse the amount of features and their distribution in the selected region [129]. Furthermore, the system analyses if the target was captured from the front by checking if the target has a rectangular shape (see Figure 6.15). We allow small errors, as they can be corrected by re-projecting the area into a rectangle, but larger errors produce problems when computing the feature description. A last check calculates the size of the object in the camera frame, and warns the user if the object is too small to create a feasible tracking database.

Once all requirements with regards to the target are fulfilled, the system sends the picture of the target to a server and the server creates the natural feature database. Additionally, the client running on the smartphone can send the GPS address of the current position (if available) to geo-reference the newly created feature database.

Once created, the natural feature database is then transmitted back to the client. Furthermore, it is stored on the server by using the transmitted GPS position as a filter in case another client later asks for trackable targets close to that position.

This means that the user running her AR application on the phone can create a new natural feature database for tracking using her smartphone or query the server for already existing ones. For both tasks the phone is tracked in 6DoF in a previously unprepared environment with minimal effort and minimal delay. The creation of the natural feature database takes 2s in average and the transmission takes between 5-10s assuming a typical database size of less than 200kb and a 3G internet connection.

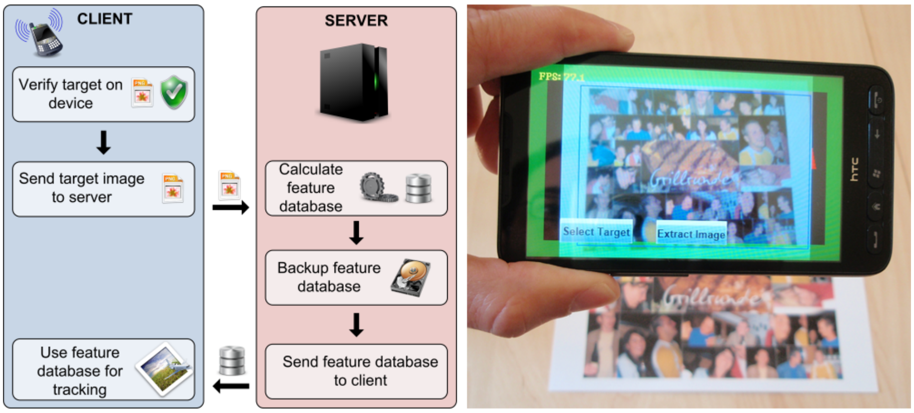

CLIENT
Verify target on device
Send target image to server
Use feature database for tracking
SERVER
Calculate feature database
Backup feature database
Send feature database to client
Select Target
Extract Image

Figure 6.15: Illustration of the target creator. (Left) Workflow of the target creator. (Right) Picture of the device while establishing a new target.

6.3.1.2 Tracking infrastructure for large working environments

As mentioned previously, we define a large working environment as an environment with many objects, which are not always in the current camera view. The typical scenarios for using such a large working environment are outdoor environments. In such cases, tracking planar surfaces such as in small indoor environments is not feasible. We therefore replaced the tracking for small working environments, which required a static tracking object, with an approach that assumes a static position and allows only rotational movements. In outdoor environments, these assumptions are appropriate, since typically the user will stay at one position and explore the environment by rotating the phone as we already discussed in Chapter 3.

Tracking in outdoor environments has very challenging requirements, such as an accurate registration to a given coordinate system, robustness and real-time processing. To satisfy these requirements, we used a sensor-fusion based approach that combines our panoramic-mapping and tracking algorithm (see Chapter 4) with accelerometer and compass data as presented by Langlotz et al. [66]. Thereby, the magnetic compass and the accelerometers provide absolute orientation measurements in the earth coordinate system, while the panorama-based vision tracking provides relative orientation tracking. To combine the accuracy and robustness of vision tracking with the absolute orientation of the inertial and magnetic sensors, the results of the panorama-based tracking and the sensors are fused in a Kalman filter-based framework. As this approach only tracks rotational movements, we use the GPS sensor of current generation smartphones to estimate the current user position.

6.3.2 In-Situ Editing

Existing authoring solutions address a specific user group. Most of them target professional content creators and media artists. The few targeted at an inexperienced audience are desktop-only applications like BuildAR[7]. In our system, we target an inexperienced

[7] www.buildar.co.nz

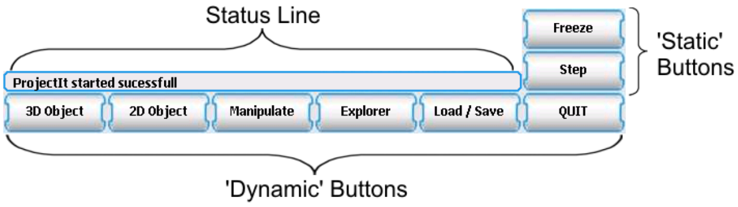


Figure 6.16: Overview of the menu.

audience to create content for mobile Augmented Reality. The minimal requirement is that they are comfortable using smartphone technology (e.g., taking pictures, sending short messages). Therefore, we limited the functionality to basic, yet powerful functions to create 3D content. The functions comprise the generation of 3D primitives, such as cubes and cylinders, but also polygonal geometry, and the geometrical modification of the generated primitives, such as scaling moving, and changing the texture of these objects. Additionally, 2D annotations and textual content can be created.

6.3.2.1 User Interface

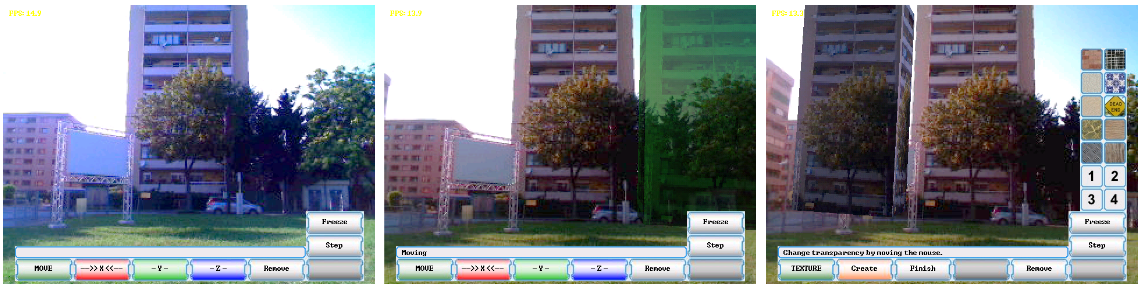
Since more and more mobile phones have no physical keyboard, all manipulations and authoring functions are controlled using a touch screen. To interact and model the geometric primitives, we developed an easy-to-use interface, which relies on touch events on the screen (see Figure 6.16). The main menu categories are (1) ”3D objects” for creating and placing 3D primitives, (2), ”2D objects” for drawing 2D elements or text, (3) ”Manipulate” for manipulating objects already created, (4) ”Explorer” for accessing the inventory of objects, and (5) ”Load / Save” for storing a scene or scene elements.

Furthermore, we added a status bar indicating the current state of the application. Another feature that we added is the freeze functionality. Early feedback within the design iteration process highlighted that accuracy was problematic when using the touch screen and keeping the current view on the scene. This raised the problem of how to accurately operate the device while pointing the camera towards the designated spot. We decided to handle this problem by adding a dedicated freeze mode. Once users are in the position to start adding digital information to the scene, they can fix the view by pressing the freeze button on the touch screen. This simulates a fixed position by freezing the current camera frame and allows the user to move the device without affecting the tracking. After a successful completion of the selected task, the user can unfreeze the view.

6.3.2.2 Generating Registered 3D Primitives

To generate 3D primitives, the user starts drawing a 2D shape of the primitive by touching the screen. The 2D footprint is generated on the ground plane.

In the case of a small tracked environment, the ground plane is the marker plane defined by the planar target used for natural feature tracking. To determine the 3D coordinates of


Moving

Change transparency by moving the mouse.

Figure 6.17: Example showing the created augmented 3D content using the system configured for large working environments. (Left) Real scene as displayed in the camera view of the smartphone, (Middle) Augmented scene showing the created virtual object (highlighted in green) during placement operation. (Right) Final scene showing a created duplicate of an existing building augmented to the left of the real building.

the footprint, we project the screen coordinate given by the touch event into the marker plane using the pose information from the tracker.

When used in a large working environment, the ground plane is the physical ground. In that case, we compute the intersection between the ray defined by the screen coordinate with the current viewing direction given by the tracker and the physical ground. This can be determined by projecting the 2D coordinate in screen space to a 3D point on the cylindrical map representing the panorama. Using the coordinate of the camera and the computed ray, we can determine the intersection with the physical ground plane.

The calculated intersections are used as input to generate the footprint of the objects. After defining the 2D footprint on the ground plane, the user can extrude the footprint to form a 3D object (see Figure 6.17). Hence, the supported 3D models range from cubes, tubes and spheres to objects with arbitrary polygonal ground planes.

6.3.2.3 Generating Registered 2D Objects

In addition to the creation of 3D objects, the system also supports a variety of 2D tools and objects, which can be used to draw or annotate the environment. They comprise lines, rectangles, circles, and freehand drawings. Similar to basic paint programs, the user can choose between pencils, brushes, or a graffiti spray tool to define the shape or characteristics of 2D primitives. In order to create registered 2D objects, the objects are not created in the camera space, but in the world space. In small working environments, where objects are registered in relation to a natural feature tracking target, the created 2D objects are inserted in the plane of the tracking target. Therefore, the intersection point between the selected point in the camera image and the ground plane is determined to be similar to the creation of 3D objects and is used as a coordinate of the newly created 2D object.

In large working environments, the usage of the intersection point between the ray of the selected point on the camera view and the ground plane as input for 2D objects only allows the creation of objects that are attached to the earth's surface. But typically users

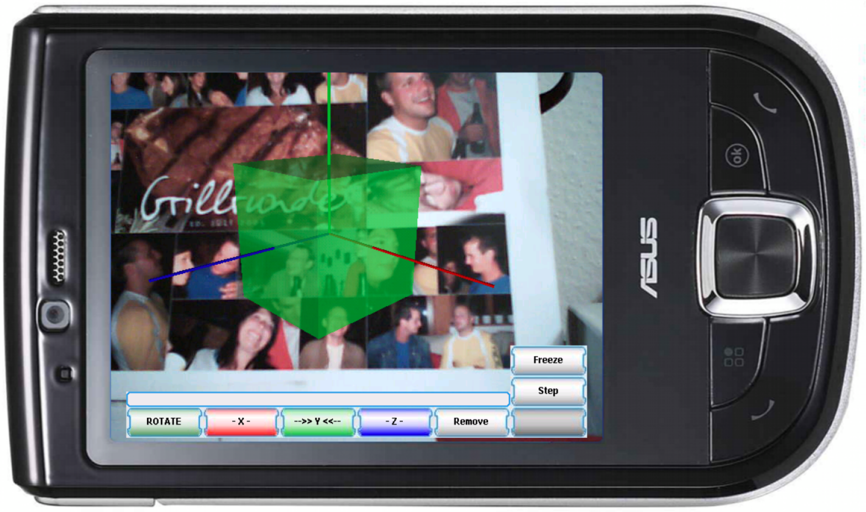


Figure 6.18: Translating the graphical content ,selecting the axis, and using the touchscreen.

want to attach 2D annotations in relation to objects in the scene. Therefore, we create the 2D objects in relation to the panoramic map, similar to the approach presented in Chapter 5. With this approach, the user is able to create 2D objects attached to each visible object in the scene. The disadvantage of this approach is that the 2D objects can only be viewed from positions close to the position where the objects were created, since they are registered to the panoramic map of this location.

6.3.2.4 Modifying Registered 3D and 2D Primitives

After creating different 3D objects, the user may want to manipulate the object. Therefore we integrated different manipulation tools. The object can be translated, scaled, and rotated by selecting the axis of transformation in the GUI and sliding in the corresponding direction over the touch screen (see Figure 6.18).

6.3.2.5 Applying Textures to the Registered 3D Objects

To enable the creation of more realistic models, we provide the functionality to assign different colors to the objects as well as to texture the object (see Figure 6.19). To assign a new color to the object, the user can select from a color palette as shown in Figure 6.19 (right). To assign a new texture, the user can select from a set of predefined textures, which can be mapped to the object (Figure 6.19 middle). Furthermore, it is possible to create new textures by selecting a region of the current camera image, which can later be used as texture and assigned to the objects (Figure 6.19 left). This is especially useful if the goal is to create virtual duplicates of real objects (see Figure 6.17). In that case it can be also convenient to immediately use the corresponding physical texture of the object. This can be achieved by back projecting the created 3D object into the camera image and using the projected area of the camera image as texture. This is ideal if objects have to be created quickly, because no texture cutout is required.

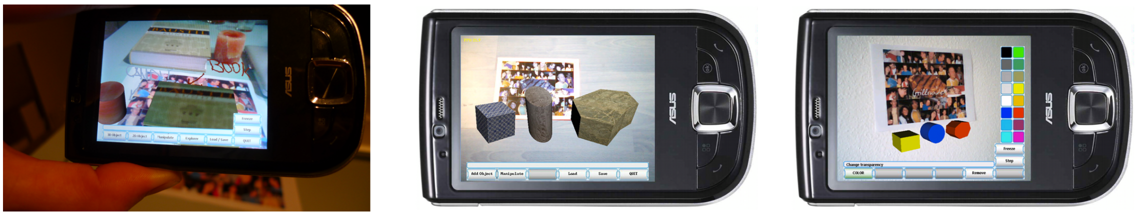


Figure 6.19: Example showing augmented 3D content created using the system. (Left) Augmented scene with augmented duplicates of real objects (book and candle). (Middle) Augmented scene showing different object shapes with different applied textures. (Right) Augmented scene showing different objects with different colors and interface for selecting colors.

6.3.3 Media Sharing

We used our server-client infrastructure described in Chapter 4 to host and share the content, which was created with our prototype. Within the prototype, AR content and all related files are submitted to the server and stored in a zip container. This is done because the content is more complex and can consist of many files such as 3D models, textures, and text information together with an XML markup file to describe the content and add meta information. We developed our own XML markup language schema to express all relevant information and named it Augmented Reality Markup Language (ARML). The resulting XML markup file is also placed in the container and later used for indexing the content and inferring the relationships concerning, among others, place, time, and user. This system can easily be adapted to only download content of known friends (e.g., by using Facebook's friends list) or updating friends (e.g., via Twitter or Facebook) if new content is published, as we support user accounts and sharing (see Chapter 4).

Beside the possibility of sharing the generated content with other users utilizing the remote media server, each user has a private inventory. This allows users to place objects into the local inventory on the device for later use. Thus the user is able to create an object (e.g., annotated with a texture of the current environment), pick it up and insert it into a scene at a different location.

6.3.4 Preliminary User Feedback on Usability

To obtain feedback on our created prototype we conducted a small study focusing on qualitative feedback from the participants. We contacted 7 people (3 female and 4 male, age 22-56yr) with no, or very little, previous knowledge in Augmented Reality. Two participants were architects, one of whom had previous experience using CAD programs such as AutoCAD or Google Sketchup.

After an introduction to the prototype's functionality demonstrated by the study's coordinator, the participants of the study had the chance to freely try out the prototype with no task given. This phase, never excelled four minutes and was followed with a small indoor scenario and an outdoor scenario. Both scenarios were previously created by the coordinator and needed to be duplicated by the study participants. Due to the qualitative nature of our study we were not measuring the difference between the original model created by the instructor and those created by the participants, nor did we time how long it took to create the models. The participants stopped when they thought that their 3D content was sufficiently close to the original. After completing those scenarios, they had another time slot when they were able to try out the prototype as they pleased. Most of the participants took advantage of this opportunity as well. While using the prototype, the participants were requested to openly express their thoughts and comments. These were noted by the coordinator of the study and used as a starting point for the semi-structured interview following the trials.

Overall, the participants showed huge interest in the prototype and all offered additional ideas for how to employ it for practical use. The first aspect that participants addressed was the tracking component. The panorama-based tracking used in the outdoor scenarios was particularly well received by all participants and they did not recall any problems when asked about it. Three participants noticed that the tracking stopped for a short time, but similar to previous tests with the panorama-based tracker 5), they all confirmed that it immediately re-started working, and therefore, did not pose any problem. For the indoor scenario, the participants successfully created and used planar targets for tracking the relative position of the device. However, in two cases the created planar target could not be tracked reliably. A later analysis showed that in the first case the target was too dark when captured, leading to problems in the feature description, despite the fact that the prototype initially accepted the target given the amount of features. The second problem with the natural feature-based tracking arose as one of the architects tried to track from a thin line drawing (a footprint of a planned building). Interestingly, both architects in our study tried this independently of each other. However, in one case the drawing was too thin and the features too repetitive to be successfully tracked, while in the second case the structure and texture were beneficial for the tracking.

The participants had a rather mixed opinion regarding the interface. Some participants reported that too many clicks and actions were required to create the objects, while others agreed that it was still acceptable and within reasonable limits. Regardless of the actions required, all participants agreed that the freeze mode was a very useful addition to the system and consequently, reported there were no major problems in precision caused by jitter, tracking problems, or complex interactions. However, three of the participants stated that they experienced problems when trying to precisely register objects in the outdoor environment. However, according to the participants this was not caused by jitter or similar issues, but rather by the limited screen size and the distance to the objects (e.g., buildings), which reduced lengths of 1m or less to a few pixels. When asked how this affected their results, two participants stated that the suboptimal registration caused objects to appear to fly or hover above the ground. The remaining participant stated that the registration issues did not affect the results, but rather the time it took to achieve the results. The participants also disagreed on the usefulness of reducing the dimensions for translating or rotating an object. Although half of the participants approved of the current process, the remaining participants stated that they would prefer to be able to translate and rotate the objects in all dimensions simultaneously. However, they could not propose suggestions for how to realize such an approach. Despite these problems, all users succeeded in duplicating the presented scene, which was confirmed both by their opinion and that of the coordinator.

During the interview, we asked all participants to name use-cases they might think of for this kind of application. For both architects use-cases from their professional field immediately came to mind. These included building indoor and outdoor digital mock-ups while in-situ (”Architects usually sketch in 2D, while this tool would give us the ability to sketch things in 3D [...] and share it with customers or colleagues”). Several answers were given by two other participants who said that they could envision creating a model of their house and putting it in Google Earth. Other frequently named use-cases included game-making scenarios such as treasure hunts or scavenger hunts, where one could leave augmented hints or signs that are created with our tool.

Overall, the participants showed interest in the possibilities of our prototype. Further, they did not face major problems with the tracking approaches used. However, while all of them solved the given tasks and seemed to enjoy using the application, some participants expressed concerns with the interface and asked for further improvements to reduce the number of touch events or increase flexibility when placing objects.

6.3.5 Summary

We presented a prototype, which can be used for user-driven creation of three-dimensional content. We showed how users can create content in place by utilizing two different approaches for tracking the device, which depend on the size of the working environment. Furthermore, neither approach requires any preparations.

We combined this tracking with simple, but powerful authoring functions, which allow the user to create new 2D and 3D content as well as to replicate existing 3D objects. This approach can be used for various AR applications such as phantom rendering, games, art installations, and social AR applications like AR browsers. We allow the user to sketch in AR using augmented graffiti, which allows users to color the environment or place annotations. Embedding a client-server approach to support content sharing between multiple users completes the system, which fulfils all aspect of an AR 2.0 environment.

During preliminary user tests, we identified the need for an improved interface, which we will try to realize in a future version. While most of the users were able to handle the content creation process, they stated that too many steps were required to create the content.

6.4 Conclusion

Within this chapter we presented several prototypes for creating and using different forms of social media in mobile Augmented Reality. We started by presenting a new approach for creating an Augmented Reality system based on user-generated and precisely linked augmented audio, which we named Audio Stickies. Accurate tracking allows users to place audio information at a finer granularity, resulting in higher accuracy and a higher density of audio annotations than with traditional (GPS) techniques. We showed how the system was combined with visual overlays to guide the users, highlight the position of audio comments, and display additional information.

We evaluated the implementation of our approach with a user study, which allowed participants to express their opinion on proposed new building designs by using Audio Stickies. These Audio Stickies can be placed in the environment and are linked to real objects or augmented objects such as planned buildings.

The user study demonstrated that user-created audio annotations are seen as a valuable source of information. And users positively acknowledged the way they were implemented in our system. We were able to show that even inexperienced users were able to create, browse, and share audio annotations, and that all users understood the link between the audio annotations and the objects to which they were referring. The concept of Audio Stickies can be combined with existing approaches that display visual information without additional hardware

We followed our work on Audio Stickies by presenting an approach for in-situ compositing of user-generated video content in mobile Augmented Reality. We showed how to create and process video files for use in mobile AR, as well as how to register them precisely in the user's environment using the panorama-based tracking approach. Even though the approach is limited to rotational movements of the cameras (due the usage of a panoramic representation of the environment), we demonstrated it can still be applied to many existing outdoor AR applications, as this motion pattern is common when using AR browsers or shooting short videos.

We demonstrated the application with a skateboard tutoring prototype. Our prototype allows users to experience skateboard tricks and actions recorded by others, which are augmented in-place and displayed at interactive frame rates on mobile phones.

Lastly, we presented a prototype that allows users to create 3D content within unprepared indoor and outdoor environments. Similar to the other prototypes, this work demonstrated the ability to create information in-place and on the device, while using a remote server only for the sharing of the created digital information. We evaluated this approach with inexperienced users to gain feedback regarding use cases and flaws of the interface. All participants showed huge interest in the possibilities of our prototype, successfully solved the given tasks, and navigated the application on their own without guidance or a given purpose.

In addition to the feedback from the individual studies, we received general positive feedback and results regarding the panorama-based tracking algorithm used. This proves the initial results gained from the technical evaluation presented in Chapter 4 as well as the movement patterns used in AR browsers identified through our study presented in Chapter 3.

Overall, in this chapter we presented several prototypes, which demonstrate important aspects of Augmented Reality 2.0. These aspects include the demonstration of rich and interactive media accessible through AR. Our prototypes blend media into the environment, rather than only displaying it in a window as is done in most existing AR browsers. Further, we demonstrated authoring tools and interfaces that allow inexperienced users to create content that is widely accepted and considered to be useful by other users. This was investigated through user studies and proved our initial hypothesis that user-generated content, in the form of social media, can be consumed and produced on mobile AR systems. We further showed within our studies that participants are open to, and very accepting of, user-generated AR content beyond textual annotations and tagging, namely audio, video, and graphical content.

Chapter 7

Conclusion

Contents

7.1 Summary of the Results

The aim of this thesis was to investigate the creation and usage of situated media content, specifically targeting AR browser applications. In particular, we aimed to show that the current gap in content quantity and quality is a major bottleneck in mobile AR systems. Our goal was to demonstrate that the problem of content quantity could be overcome because the creation of user-generated AR content is technically feasible. In order to show the applicability of our approach, we created prototypes that enable the use of social media in AR. Further, we extended the types of augmentations used beyond textual annotations and tagging, to include audio, video, and 3D models as well.

Within this thesis, we organized the projects completed into four distinct parts. We started in Chapter 3 by outlining the current state of AR browsers as the most common type of mobile AR application. We used the outcome of our survey, in particular the identified lack of useful content, to frame requirements for next generation mobile AR environments, which we coined Augmented Reality 2.0. Within the concept of Augmented Reality 2.0 we presented social situated media as a fundamental component for next generation mobile AR applications. We also identified challenges and prerequisites that would need to be addressed in order to work towards a successful AR 2.0 environment.

We continued our research by presenting in Chapter 4 technical foundations that address the challenges we identified in Chapter 3 and are needed to implement prototypes for an Augmented Reality 2.0 environment. We specifically highlighted the necessary core components of a mobile AR framework using Studierstubes ES as an example, and followed by presenting our approach of panoramic mapping and tracking used to realize precise tracking in unknown outdoor environments. We finished the chapter by presenting a technical evaluation of the panoramic tracking approach.

Based on the technical foundations, in Chapter 5 we showed our prototypes for editing and using 2D media, text, and images in mobile AR applications. Through these prototypes we also investigated different algorithms for re-detecting media created by one user in the view of a different user. We studied the usability of the interface as well as giving a technical evaluation of its re-detection performance.

In Chapter 6 we expanded the research presented in Chapter 5 to other media forms. In particular, we demonstrated the technical feasibility of prototypes that utilized user-generated audio, video, and 3D content. We further investigated the applicability and usefulness of each prototype through user studies.

In the following, we reiterate the results achieved and outline lessons that can be learned from this thesis. We conclude by giving an outlook to future research directions that might evolve from the results of this thesis.

7.2 Lessons Learned

Mobile AR applications, and in particular, AR browsers have been tried out by millions of users. However, we hypothesized that most users are early adopters who have an interest, but do not regularly use this technology due to the low quality and quantity of the displayed information. This hypothesis was supported by our initial study on AR browsers.

- A significant number of people tried AR browsers on their personal mobile devices and mostly responded positively on the technology. As shown in Chapter 3, users also pointed out their interest in this type of application. Furthermore, end-users confirmed the future potential of mobile AR technology in areas such as content browsing and navigation, not only in the survey 3, but also in the open discussions following the user studies presented in Chapters 5 and 6.

- The current scarcity of content available for mobile AR applications, together with the poor quality of the displayed media, poses according to the users the main bottleneck for mobile AR applications, especially with regards to browsing information in the user's environment. This was confirmed by the users participating in our survey presented in Chapter 3. Further, they also stated complaints about the poor quality of the user interface, user experience, and platform-related problems such as issues with battery life.

These observations motivated the development of requirements for successful end-user mobile AR systems that were created and documented within the course of this thesis. From the design of our prototypes and the outcomes of the studies conducted we can draw the following conclusion:

- We showed that by taking into account the following key requirements, a mobile AR system can be built for scalability in terms of users and content by integrating social media. These key requirements for end user mobile AR applications include:
  - A low-cost platform for mobile AR

  - Mobility to realize AR in a global space
  - Large-scale real-time AR tracking
  - Standardized content formats
  - Backend infrastructure for distribution of AR content and applications
  - End-user focused tools for creating AR content in-situ
  - A rich and interactive media environment accessible through AR
  - Mobile and intuitive interaction with the content

Tracking is necessary for nearly any kind of AR system. The constraints and characteristics of the deployed tracking component need to be considered when designing the other components of a mobile AR system (e.g., interface and anchoring of situated media). The general conclusions that can be drawn include:

- Carefully analysing the movement patterns that are dictated by the device's characteristics and the required tasks helps to expose the requirements necessary for the tracking technology. When using phone-based mobile AR applications in outdoor environments to browse for information in the user's proximity, the user typically performs only rotational movements. This fact enables specific limitations in the tracking system. This was observed within our initial study on current mobile AR applications (see Chapter 3), as well as verified throughout our later studies in Chapters 5 and 6 where not a single participant in the studies showed the intention to change position while using the AR applications.
- In phone-based mobile AR systems, the panoramic tracking approach was demonstrated to be fast and robust, while also increasing the precision of AR applications. As verified in all the studies performed in Chapters 5 and 6, the panorama-based tracker did not pose any problems when handled by the users. Even though the functionality and constraints of the approach were not explained to users, they were rarely confronted with interruptions caused by tracking issues. In the rare cases these interruptions occurred, the users were able to immediately restart the tracker within seconds without assistance. The tracking was fast enough to have enough processing time to render interface and content (Chapter 4). Therefore, it poses, despite the limitations of orientation tracking only, an appropriate solution when tracking the camera in outdoor mobile AR applications.

The core finding, which was explored in the main chapters of this work, is that consuming and producing user-generated content on a mobile AR system can be technically realized and proved usable in many scenarios. In particular we showed:

- A carefully analysed combination of tracking and interface technology allows even untrained users to create, use, and share situated media within a mobile AR environment. Because we were targeting untrained users, the interfaces implemented do not reassemble ideas from professional applications such as timelines (popular in many professional video and audio editing applications) or complex CAD-systems, but instead focus on simple but powerful user input and tools. This design decision was widely supported by the feedback resulting from the studies in Chapters 5 and 6.

- In general, our studies showed that there is wide acceptance for user-generated AR content beyond textual annotations and 2D tagging, namely for audio, video, and 3D content. Additionally, the studies revealed many possible use cases are unexplored.
- Working with 3D content requires more research in the area of intuitive interfaces. The results of the qualitative study suggest that once created, precisely positioning 3D content is still a time-consuming and cumbersome task.

Finally, it is also worth mentioning that the user feedback from the studies suggests that situated media in mobile AR applications is especially useful in cases where the AR aspect is fully exploited. This means, for example, that the overlaid information is precisely integrated and blended into the reality while also enough real-world information (captured with the camera) is visible to fully use the contextual information.

7.3 Future Directions

We see several research directions which would allow for further evolution of the key aspects of Augmented Reality 2.0. One is to investigate new device form factors as low cost AR platforms. Currently smartphones are the best choice when looking for an off-the-shelf device for mobile AR applications. However, several companies have started to investigate the potential of future generation mobile devices such as Head Mounted Displays (e.g., Google Project Glass) or wrist-worn devices such as smartphones with the form factor similar to wrist-watches. Especially HMD manufacturers consider AR as a key technology. However, the movement and usage patterns and ergonomic considerations are different when using Augmented Reality on those devices.

For example, while mobile AR on smartphones is, according to our research, mostly used while only performing rotational movements (see Chapter 3), this will be likely different when HMDs are used. Such a change in movement and usage would affect the type of tracking technology required, as well as the interaction metaphors that would need to be used in mobile AR applications, thereby necessitating further investigation.

Similarly, smartwatches or wrist-worn devices for AR pose new challenges due to their size. Visual AR would likely not be a major interface for these devices due to their limited screen size. However, audio AR could compensate for the lack of screen space by playing audio augmentations using wireless headphones. Similar to the presented approach in this thesis, audio AR on smart-watches would require a careful analysis of the movement pattern used, and probably even more importantly, an analysis of the pointing or selection metaphor employed when using wrist-worn devices.

However, even future mobile AR technology requires precise tracking of the device. While already an active search area, real-time tracking especially in large outdoor environments (e.g., city scale) is still hard to achieve and works only within given constraints (e.g., full environment information available). While we believe that the presented panorama-based tracking is already a significant step forward in orientation tracking, we also think that it can be further improved, for example by adding the image-based localization techniques presented by Arth et al. [2] [3]. Also, SLAM-like tracking techniques might become an interesting alternative when problems such as memory consumption and the required movement pattern are solved.

The characteristics and tracking technology in future devices will also affect how situated media is anchored to objects and places. New technologies may be needed if the panorama-based tracking is removed from the system and replaced by a tracking technology that works in a substantially different manner.

One of the most wide-open research fields is still the area of interaction with situated media. Within this thesis, we applied a horizontal approach by presenting several techniques for interacting with various forms of situated media, ranging from textual content to video and 3D content. However, depending on the scenario, other interaction forms might be useful. For example, when tracking in 6DoF the precise and intuitive placement of media in 3D is still a challenging task that gets even more ambitious when complex media is used. We think that techniques from computer graphics and modelling, such as sketching, can be applied and combined with computer vision methods to refine given interactions. Combined techniques should thereby aim for precision and intuitiveness as a result of the combination of the techniques (e.g., roughly placed objects are aligned with real-world geometry captures using computer vision).

Within this thesis we also identified possible use cases to evaluate our prototypes. However, we strongly believe that other use cases of situated media are possible, but need further investigation, for example in large field user studies investigating the sharing aspect, or researching the use of situated media for communication and outdoor games. We also think that the research presented in this thesis can be seen as a tool for applying AR. By giving inexperienced AR users this tool for placing and sharing media information in AR, we enable users to create new applications and uses cases that the research community has not initially foreseen. This can consequently help to identify additional useful application scenarios for mobile AR.

Within the scope of this thesis we have already demonstrated the possibilities of various media types in AR. However, we think that future research should also aim to use and further investigate the possibilities of the various media types in AR instead of treating visual and auditory media as widely separated communities.

Future versions of mobile AR applications, and especially AR browsers, should also rely more on standardized content formats and a unified media distribution architecture. Within this thesis, we relied on open formats such as XML or existing web technology as much as possible, but agreeing on open standards for Augmented Reality not only requires effort from the research community, but from industry as well. Organizations such as the Khronos Group[1] or the World Wide Web Consortium (W3C)[2] will likely need to be involved in future decision on open standards. We also expect future research and industry efforts to integrate situated media into social platforms (e.g., Facebook or Wikipedia) as well as media databases (e.g., YouTube).

Finally, we think that, driven by the spirit of social networks and Web 2.0, but also

[1] www.khronos.org

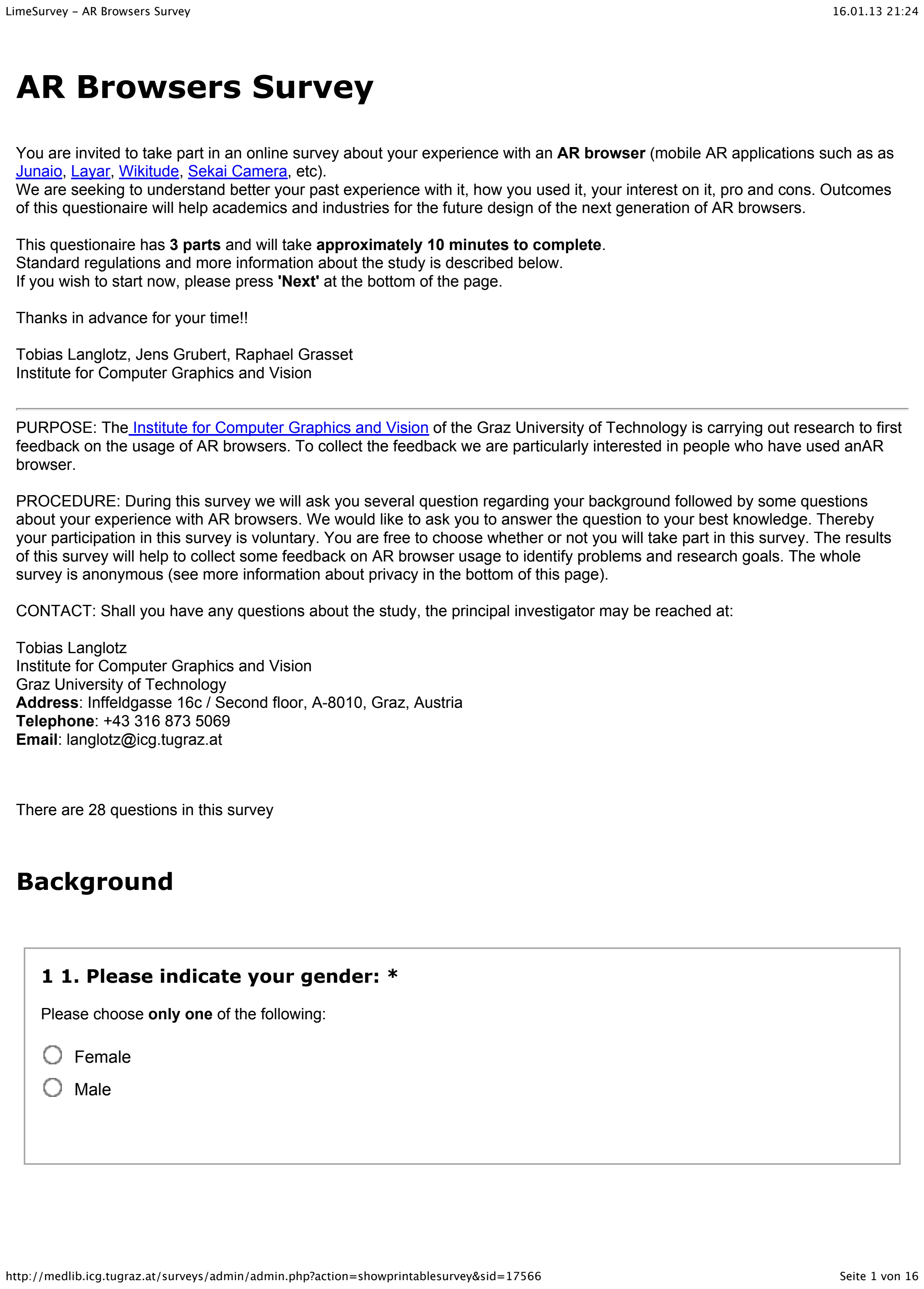
[2] www.w3.org

by the recent advances made in the field of mobile AR technology in terms of hardware and software, Augmented Reality 2.0 is on its way to being widely used, and we hope that this thesis contributed a small portion to its development.

Appendix A

Questionnaires

A.1 This section presents the questionnaire used in the user studies



LimeSurvey - AR Browsers Survey 16.01.13 21:24

AR Browsers Survey

You are invited to take part in an online survey about your experience with an **AR browser** (mobile AR applications such as as Junaio, Layar, Wikitude, Sekai Camera, etc).
We are seeking to understand better your past experience with it, how you used it, your interest on it, pro and cons. Outcomes of this questionaire will help academics and industries for the future design of the next generation of AR browsers.

This questionaire has **3 parts** and will take **approximately 10 minutes to complete**.
Standard regulations and more information about the study is described below.
If you wish to start now, please press **'Next'** at the bottom of the page.

Thanks in advance for your time!!

Tobias Langlotz, Jens Grubert, Raphael Grasset
Institute for Computer Graphics and Vision

PURPOSE: The Institute for Computer Graphics and Vision of the Graz University of Technology is carrying out research to first feedback on the usage of AR browsers. To collect the feedback we are particularly interested in people who have used anAR browser.

PROCEDURE: During this survey we will ask you several question regarding your background followed by some questions about your experience with AR browsers. We would like to ask you to answer the question to your best knowledge. Thereby your participation in this survey is voluntary. You are free to choose whether or not you will take part in this survey. The results of this survey will help to collect some feedback on AR browser usage to identify problems and research goals. The whole survey is anonymous (see more information about privacy in the bottom of this page).

CONTACT: Shall you have any questions about the study, the principal investigator may be reached at:

Tobias Langlotz
Institute for Computer Graphics and Vision
Graz University of Technology
Address: Inffeldgasse 16c / Second floor, A-8010, Graz, Austria
Telephone: +43 316 873 5069
Email: langlotz@icg.tugraz.at

There are 28 questions in this survey

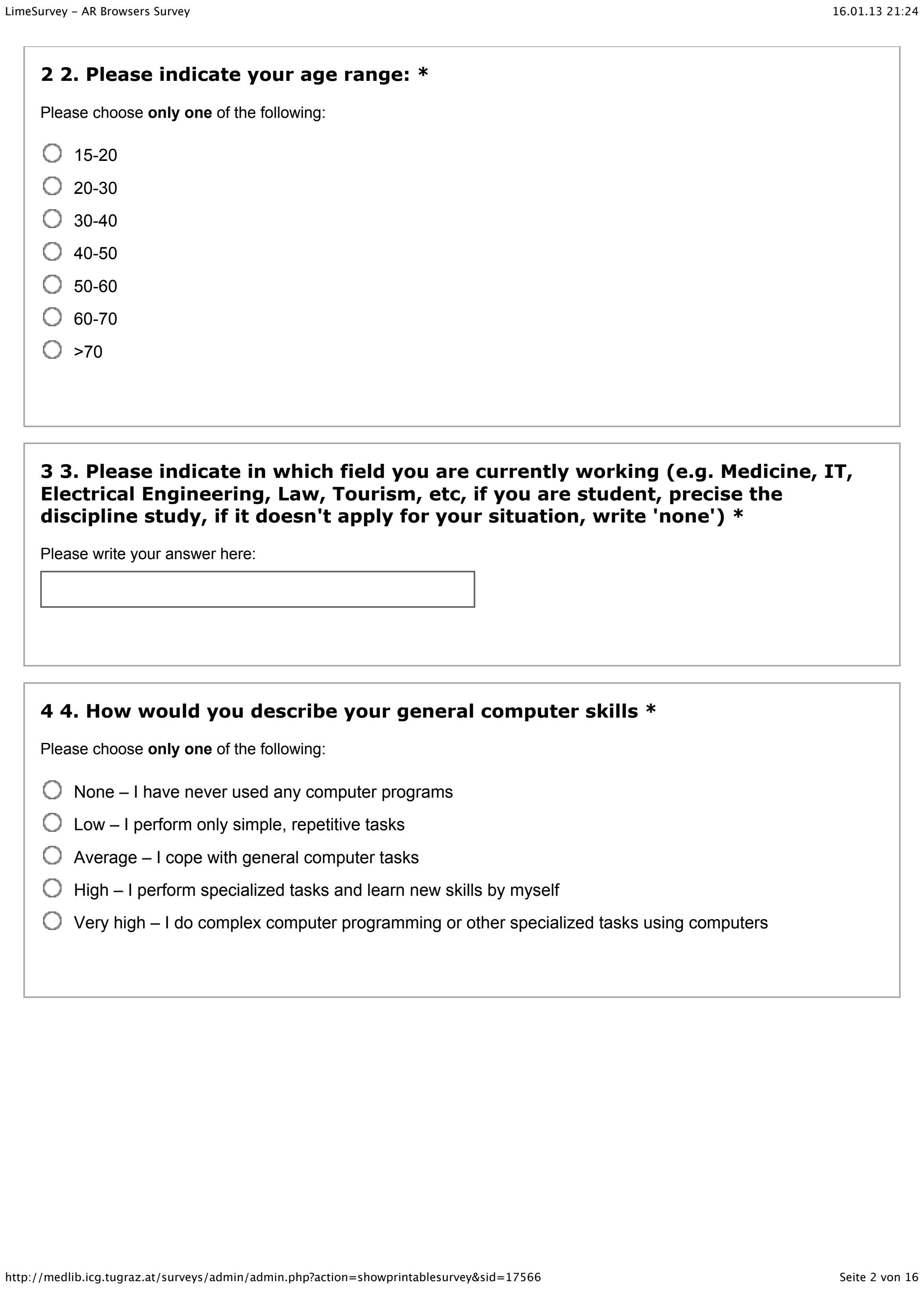
Background

1 1. Please indicate your gender: *

Please choose **only one** of the following:

○ Female
○ Male

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 1 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

2 2. Please indicate your age range: *

Please choose **only one** of the following:

○ 15-20
○ 20-30
○ 30-40
○ 40-50
○ 50-60
○ 60-70
○ >70

3 3. Please indicate in which field you are currently working (e.g. Medicine, IT, Electrical Engineering, Law, Tourism, etc, if you are student, precise the discipline study, if it doesn't apply for your situation, write 'none') *

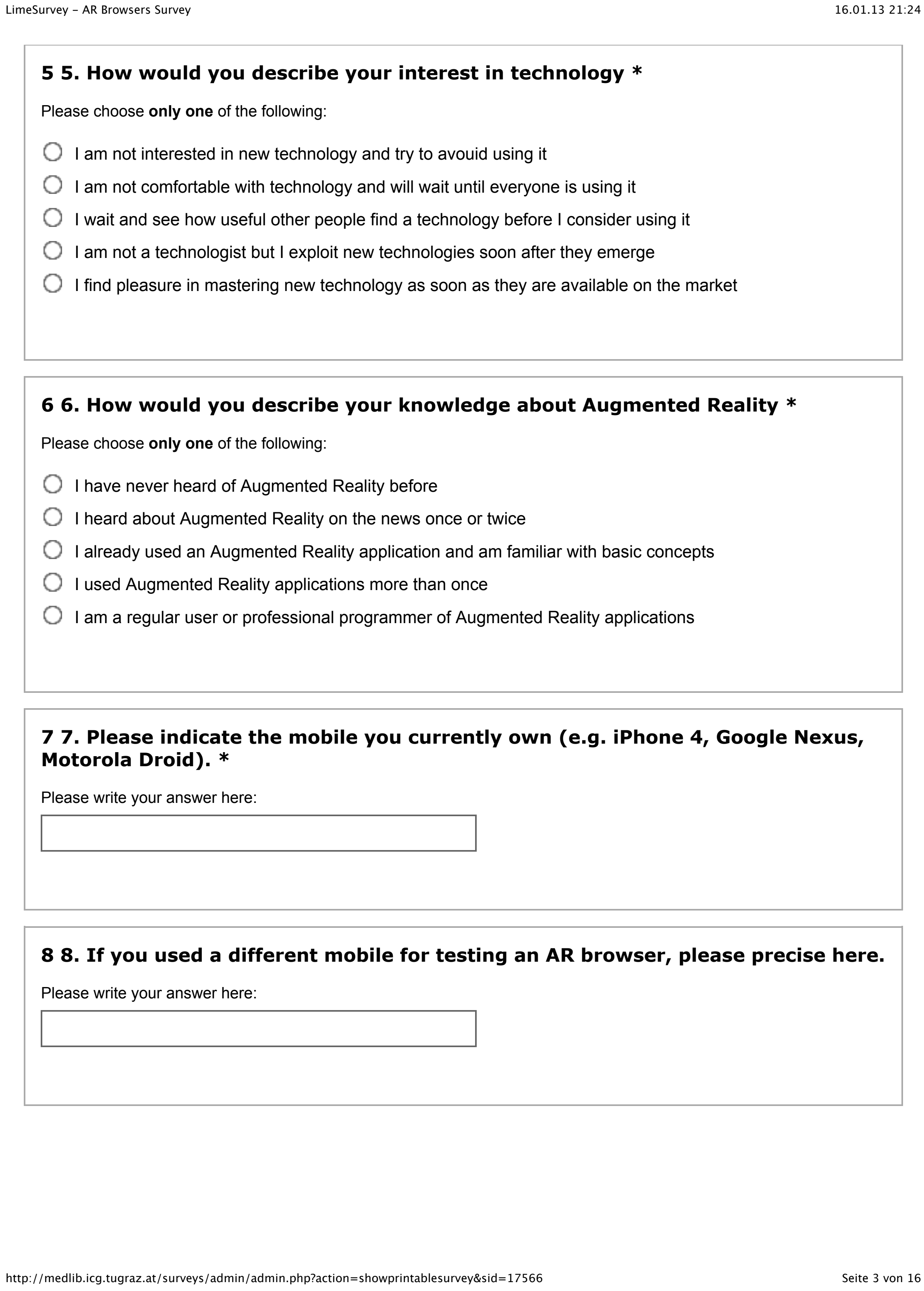
Please write your answer here:

4 4. How would you describe your general computer skills *

Please choose **only one** of the following:

○ None – I have never used any computer programs
○ Low – I perform only simple, repetitive tasks
○ Average – I cope with general computer tasks
○ High – I perform specialized tasks and learn new skills by myself
○ Very high – I do complex computer programming or other specialized tasks using computers

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 2 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

5 5. How would you describe your interest in technology *

Please choose **only one** of the following:

○ I am not interested in new technology and try to avouid using it

○ I am not comfortable with technology and will wait until everyone is using it

○ I wait and see how useful other people find a technology before I consider using it

○ I am not a technologist but I exploit new technologies soon after they emerge

○ I find pleasure in mastering new technology as soon as they are available on the market

6 6. How would you describe your knowledge about Augmented Reality *

Please choose **only one** of the following:

○ I have never heard of Augmented Reality before

○ I heard about Augmented Reality on the news once or twice

○ I already used an Augmented Reality application and am familiar with basic concepts

○ I used Augmented Reality applications more than once

○ I am a regular user or professional programmer of Augmented Reality applications

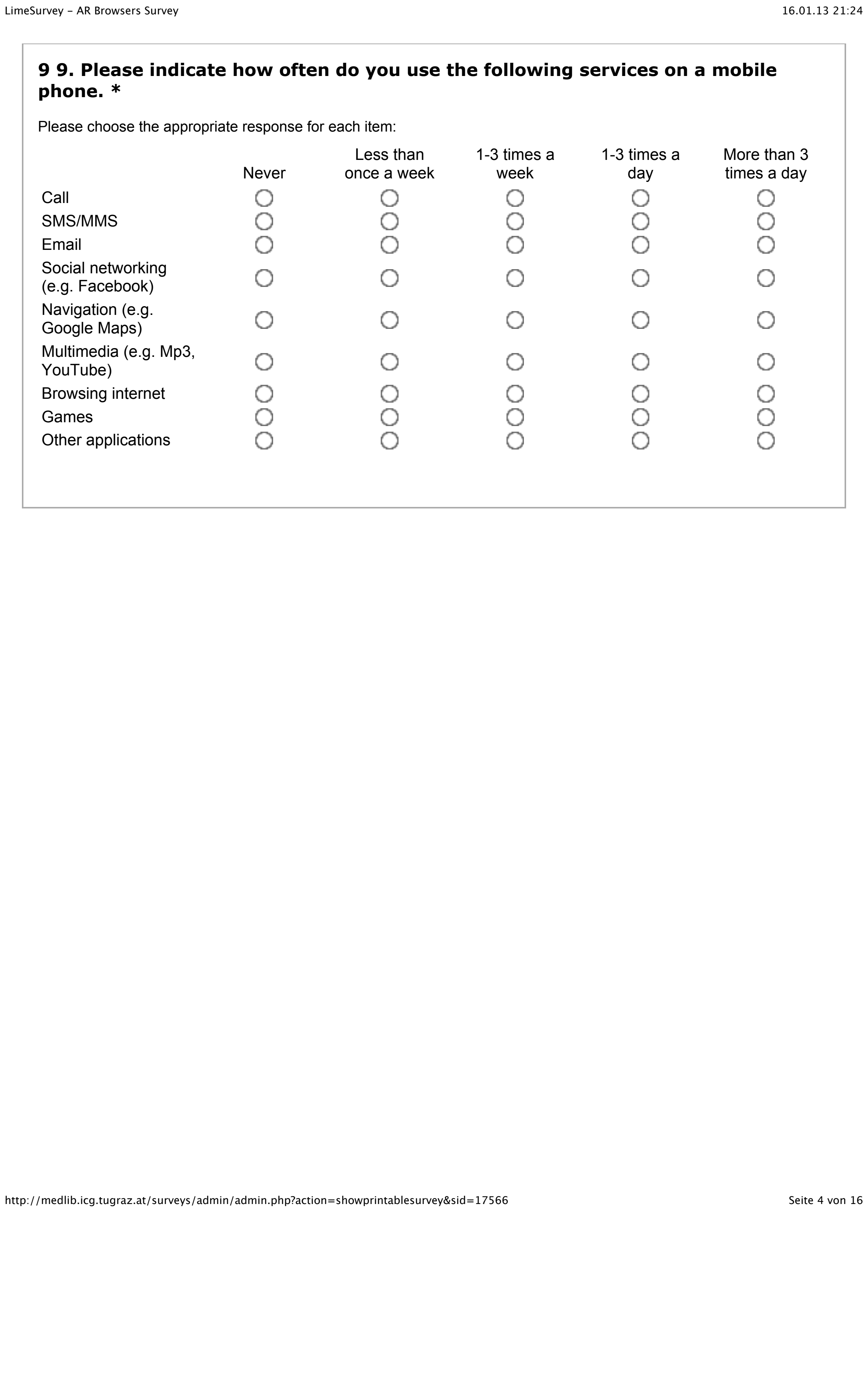
7 7. Please indicate the mobile you currently own (e.g. iPhone 4, Google Nexus, Motorola Droid). *

Please write your answer here:

8 8. If you used a different mobile for testing an AR browser, please precise here.

Please write your answer here:

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 3 von 16



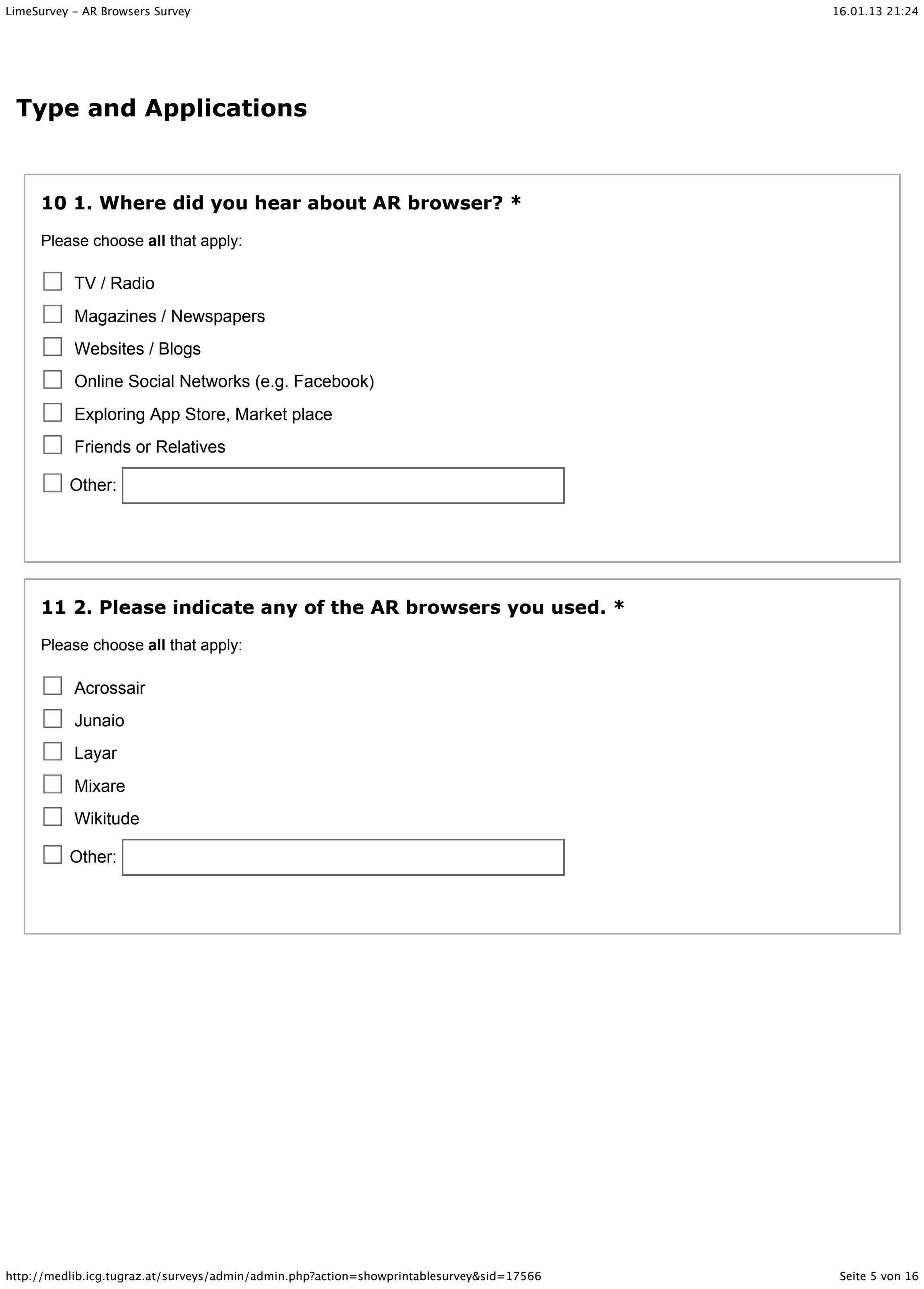
LimeSurvey - AR Browsers Survey 16.01.13 21:24

9 9. Please indicate how often do you use the following services on a mobile phone. *

Please choose the appropriate response for each item:

	Never	Less than once a week	1-3 times a week	1-3 times a day	More than 3 times a day
Call	○	○	○	○	○
SMS/MMS	○	○	○	○	○
Email	○	○	○	○	○
Social networking (e.g. Facebook)	○	○	○	○	○
Navigation (e.g. Google Maps)	○	○	○	○	○
Multimedia (e.g. Mp3, YouTube)	○	○	○	○	○
Browsing internet	○	○	○	○	○
Games	○	○	○	○	○
Other applications	○	○	○	○	○

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 4 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

Type and Applications

10 1. Where did you hear about AR browser? *

Please choose **all** that apply:

□ TV / Radio

□ Magazines / Newspapers

□ Websites / Blogs

□ Online Social Networks (e.g. Facebook)

□ Exploring App Store, Market place

□ Friends or Relatives

□ Other:

11 2. Please indicate any of the AR browsers you used. *

Please choose **all** that apply:

□ Acrossair

□ Junaio

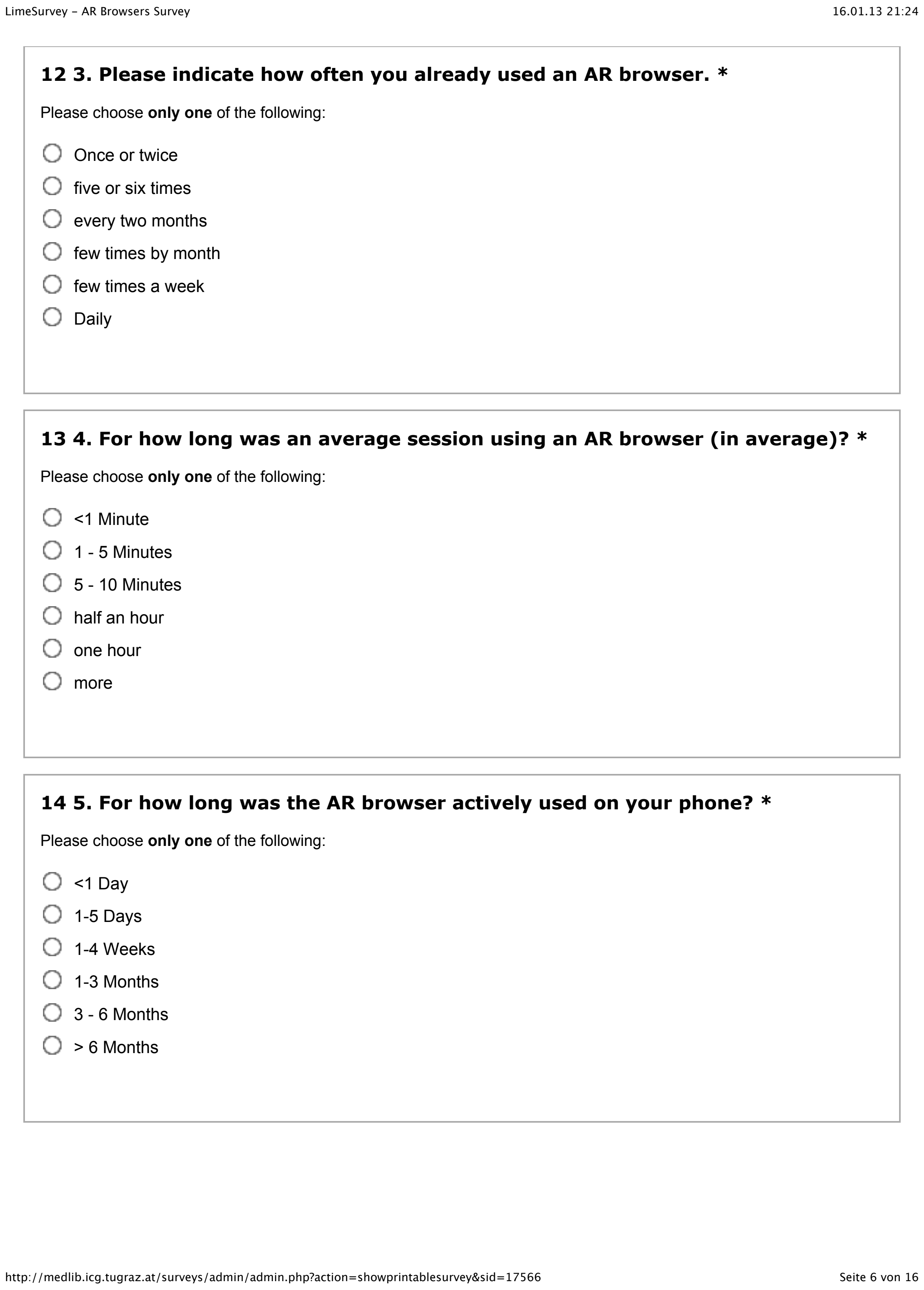
□ Layar

□ Mixare

□ Wikitude

□ Other:

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 5 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

12 3. Please indicate how often you already used an AR browser. *

Please choose **only one** of the following:

○ Once or twice

○ five or six times

○ every two months

○ few times by month

○ few times a week

○ Daily

13 4. For how long was an average session using an AR browser (in average)? *

Please choose **only one** of the following:

○ <1 Minute

○ 1 - 5 Minutes

○ 5 - 10 Minutes

○ half an hour

○ one hour

○ more

14 5. For how long was the AR browser actively used on your phone? *

Please choose **only one** of the following:

○ <1 Day

○ 1-5 Days

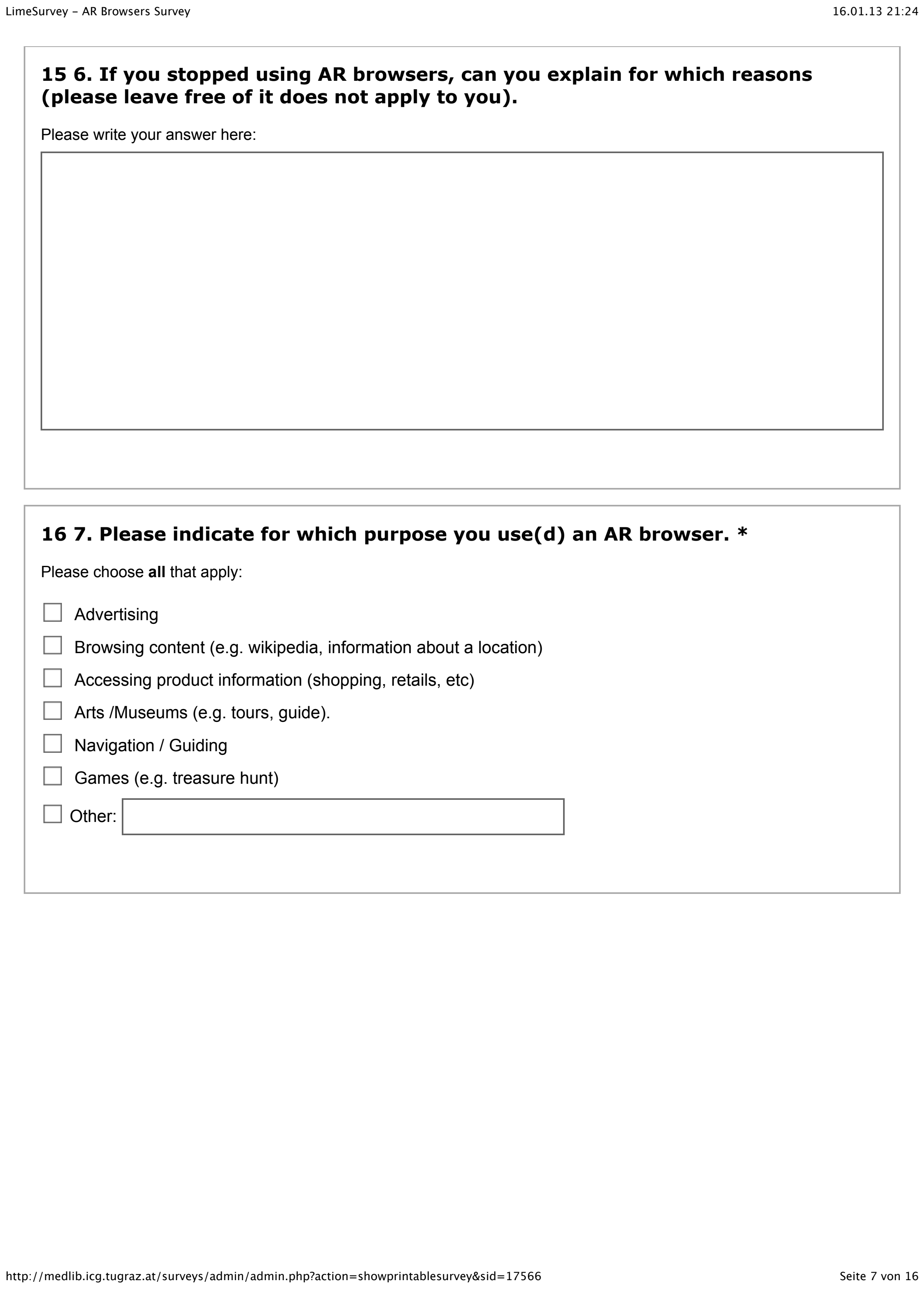
○ 1-4 Weeks

○ 1-3 Months

○ 3 - 6 Months

○ > 6 Months

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 6 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

15 6. If you stopped using AR browsers, can you explain for which reasons (please leave free of it does not apply to you).

Please write your answer here:

16 7. Please indicate for which purpose you use(d) an AR browser. *

Please choose **all** that apply:

□ Advertising

□ Browsing content (e.g. wikipedia, information about a location)

□ Accessing product information (shopping, retails, etc)

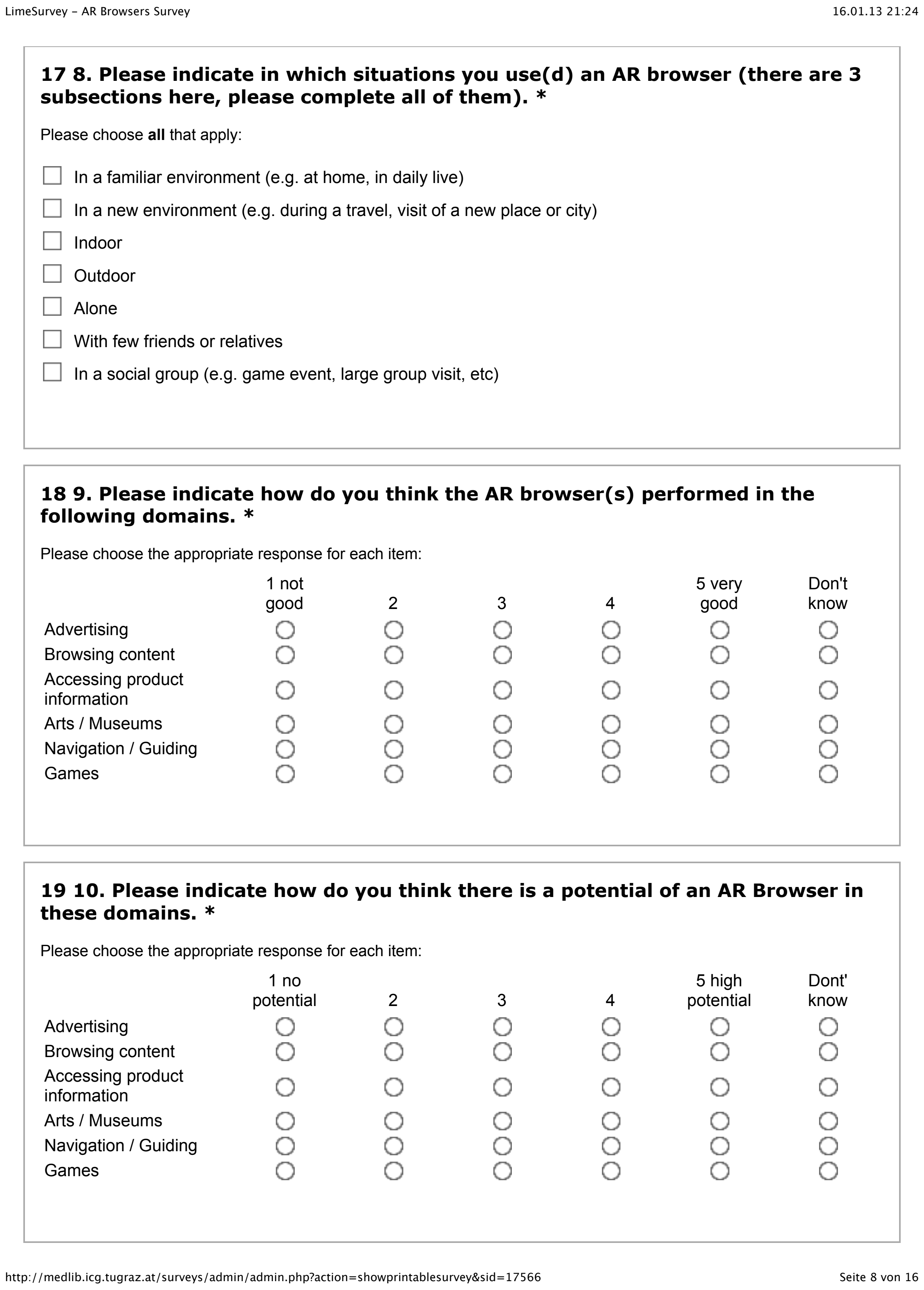
□ Arts /Museums (e.g. tours, guide).

□ Navigation / Guiding

□ Games (e.g. treasure hunt)

□ Other:

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 7 von 16



17 8. Please indicate in which situations you use(d) an AR browser (there are 3 subsections here, please complete all of them). *

Please choose **all** that apply:

- □ In a familiar environment (e.g. at home, in daily live)
- □ In a new environment (e.g. during a travel, visit of a new place or city)
- □ Indoor
- □ Outdoor
- □ Alone
- □ With few friends or relatives
- □ In a social group (e.g. game event, large group visit, etc)

18 9. Please indicate how do you think the AR browser(s) performed in the following domains. *

Please choose the appropriate response for each item:

	1 not good	2	3	4	5 very good	Don't know
Advertising	○	○	○	○	○	○
Browsing content	○	○	○	○	○	○
Accessing product information	○	○	○	○	○	○
Arts / Museums	○	○	○	○	○	○
Navigation / Guiding	○	○	○	○	○	○
Games	○	○	○	○	○	○

19 10. Please indicate how do you think there is a potential of an AR Browser in these domains. *

Please choose the appropriate response for each item:

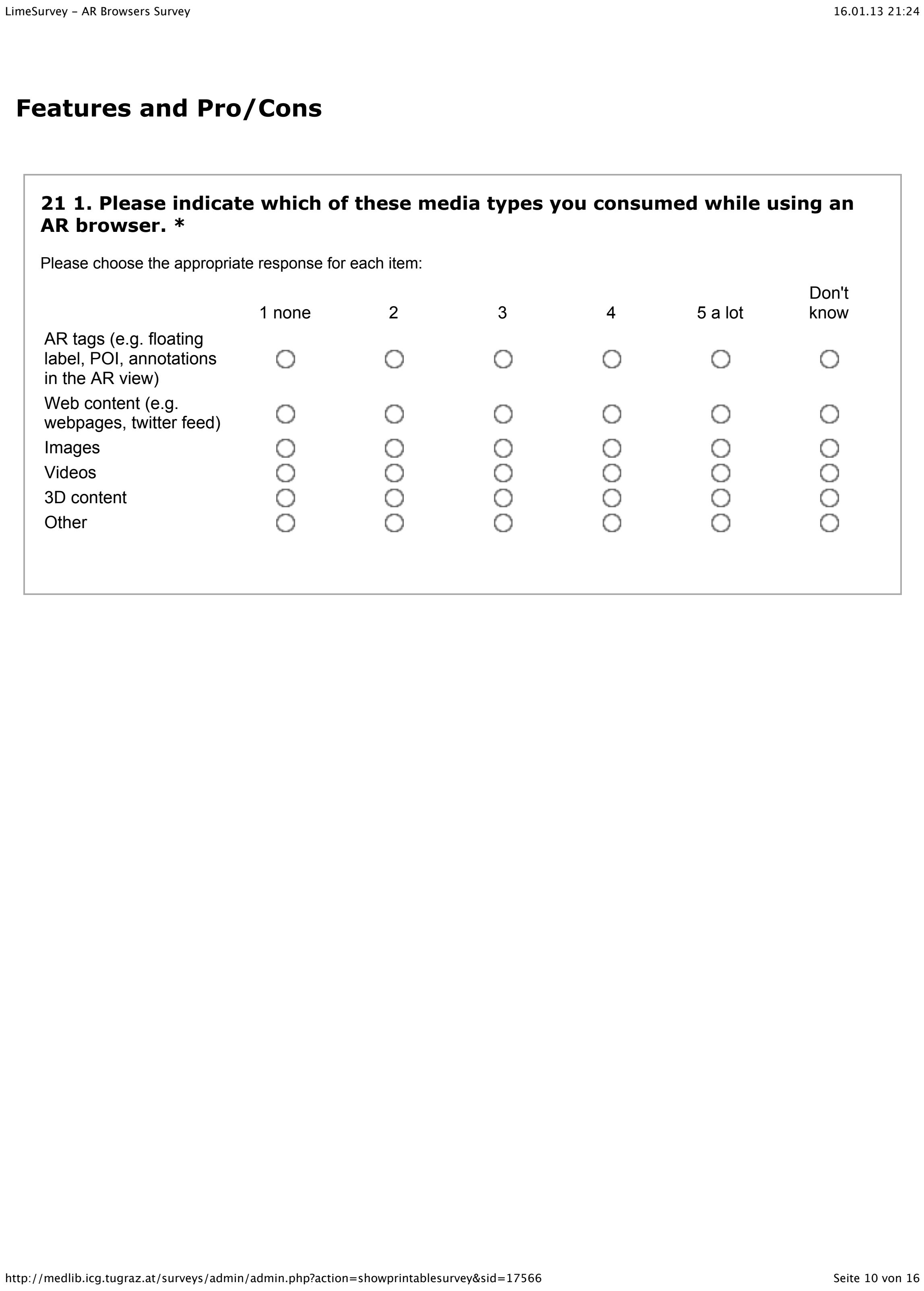
	1 no potential	2	3	4	5 high potential	Dont' know
Advertising	○	○	○	○	○	○
Browsing content	○	○	○	○	○	○
Accessing product information	○	○	○	○	○	○
Arts / Museums	○	○	○	○	○	○
Navigation / Guiding	○	○	○	○	○	○
Games	○	○	○	○	○	○

LimeSurvey - AR Browsers Survey 16.01.13 21:24

20 11. Please indicate if you see any other application areas of an AR browser.

Please write your answer here:

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 9 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

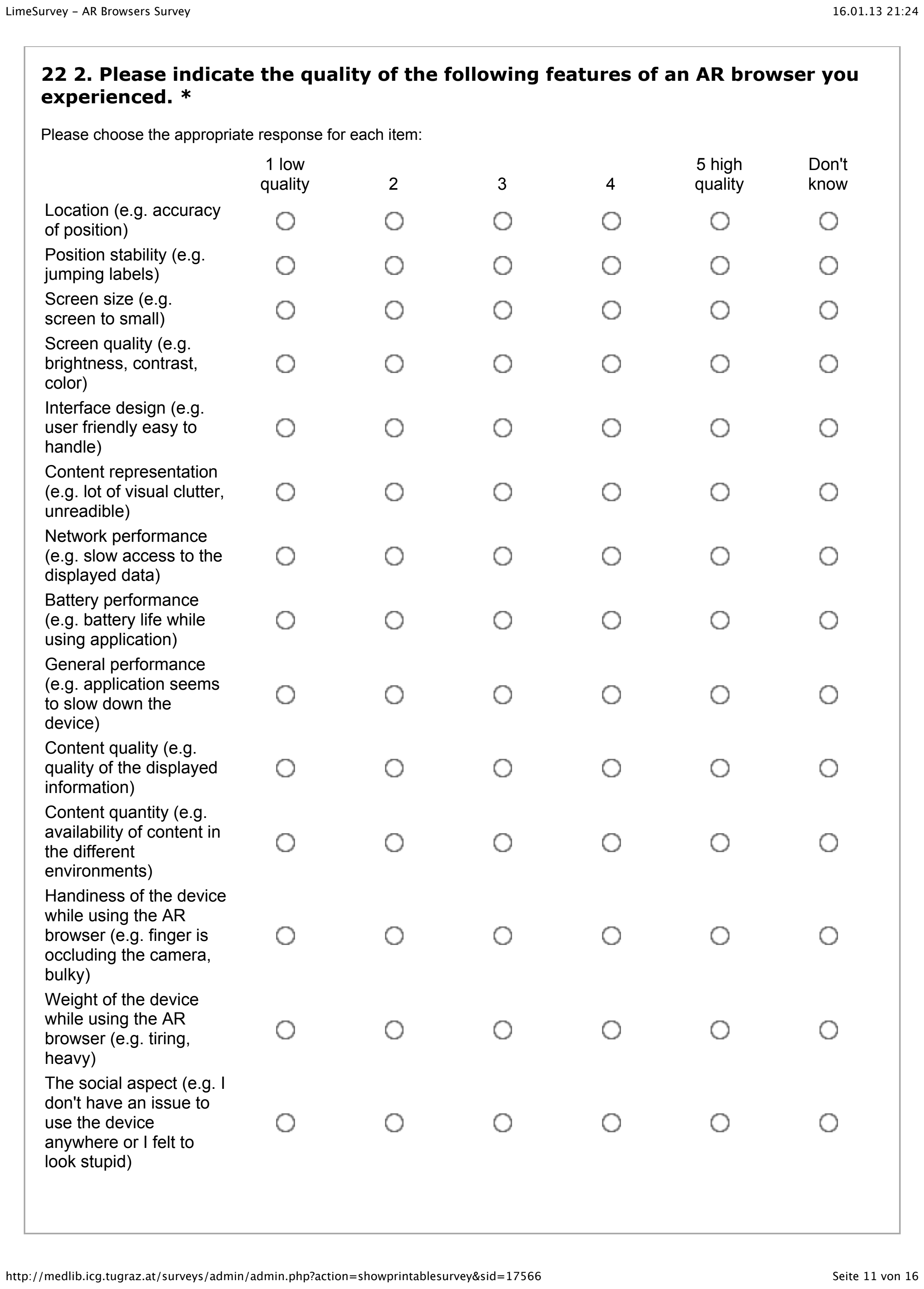
Features and Pro/Cons

21 1. Please indicate which of these media types you consumed while using an AR browser. *

Please choose the appropriate response for each item:

	1 none	2	3	4	5 a lot	Don't know
AR tags (e.g. floating label, POI, annotations in the AR view)	○	○	○	○	○	○
Web content (e.g. webpages, twitter feed)	○	○	○	○	○	○
Images	○	○	○	○	○	○
Videos	○	○	○	○	○	○
3D content	○	○	○	○	○	○
Other	○	○	○	○	○	○

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 10 von 16



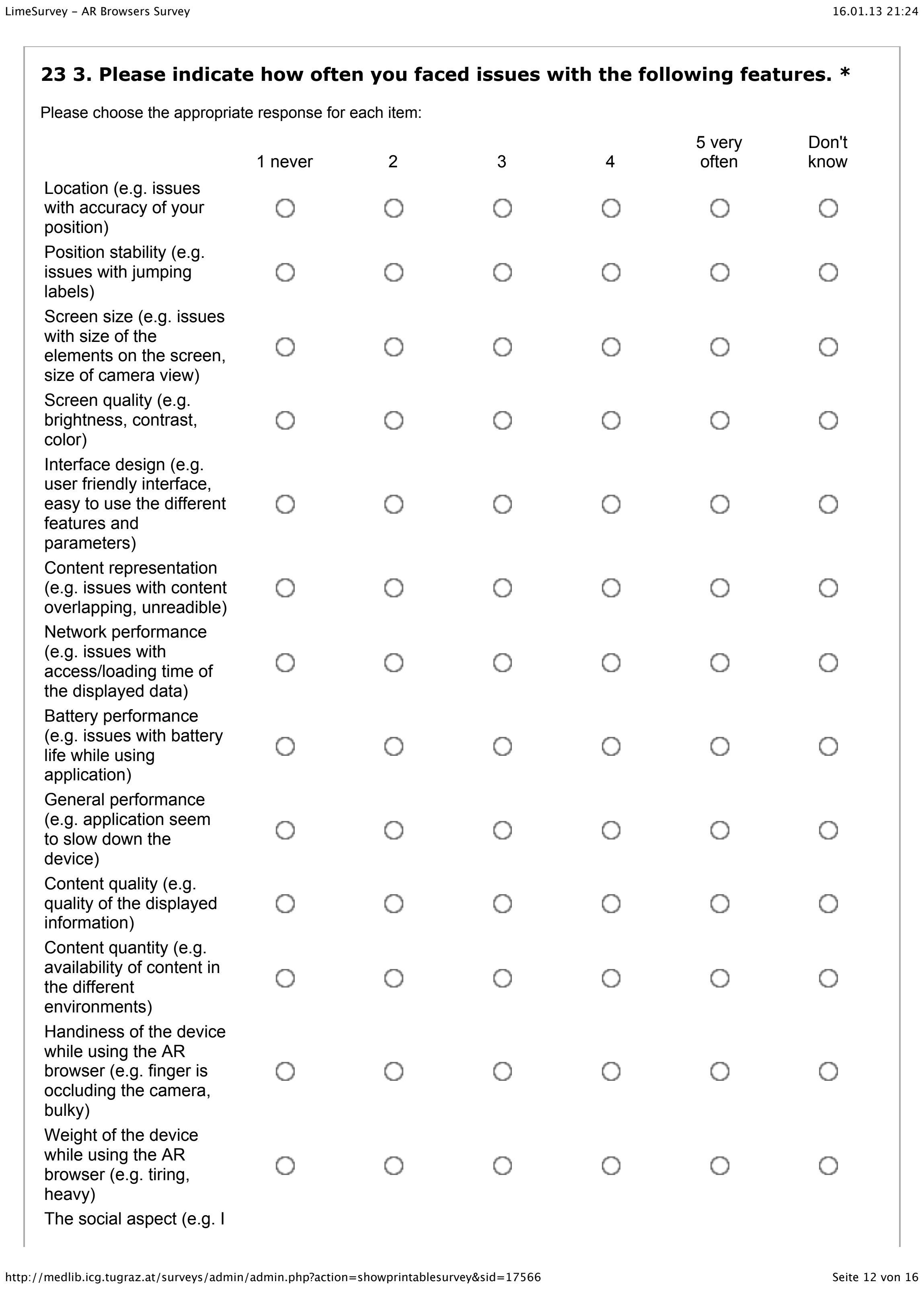
LimeSurvey - AR Browsers Survey 16.01.13 21:24

22 2. Please indicate the quality of the following features of an AR browser you experienced. *

Please choose the appropriate response for each item:

	1 low quality	2	3	4	5 high quality	Don't know
Location (e.g. accuracy of position)	○	○	○	○	○	○
Position stability (e.g. jumping labels)	○	○	○	○	○	○
Screen size (e.g. screen to small)	○	○	○	○	○	○
Screen quality (e.g. brightness, contrast, color)	○	○	○	○	○	○
Interface design (e.g. user friendly easy to handle)	○	○	○	○	○	○
Content representation (e.g. lot of visual clutter, unreadible)	○	○	○	○	○	○
Network performance (e.g. slow access to the displayed data)	○	○	○	○	○	○
Battery performance (e.g. battery life while using application)	○	○	○	○	○	○
General performance (e.g. application seems to slow down the device)	○	○	○	○	○	○
Content quality (e.g. quality of the displayed information)	○	○	○	○	○	○
Content quantity (e.g. availability of content in the different environments)	○	○	○	○	○	○
Handiness of the device while using the AR browser (e.g. finger is occluding the camera, bulky)	○	○	○	○	○	○
Weight of the device while using the AR browser (e.g. tiring, heavy)	○	○	○	○	○	○
The social aspect (e.g. I don't have an issue to use the device anywhere or I felt to look stupid)	○	○	○	○	○	○

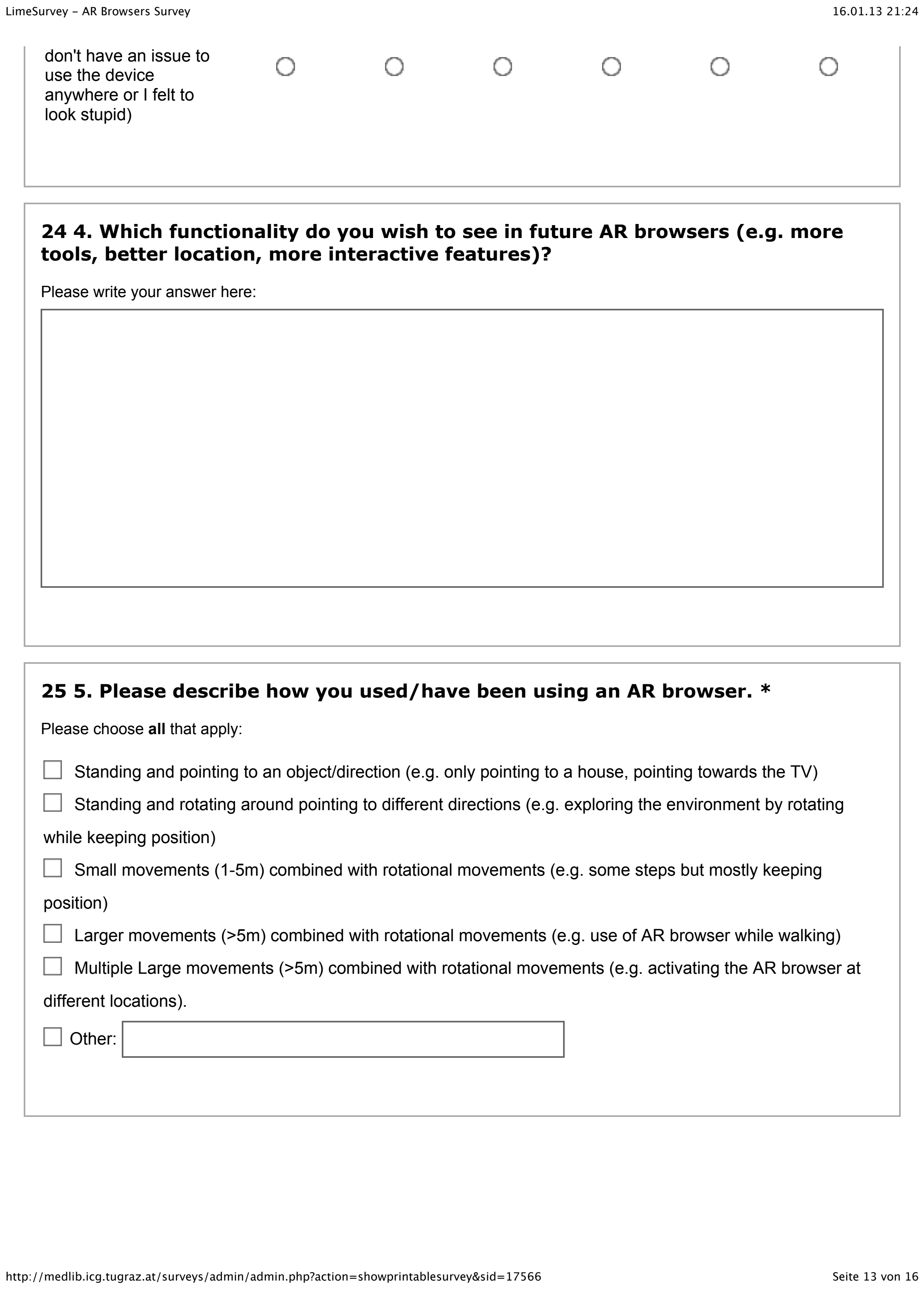
http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 11 von 16



23 3. Please indicate how often you faced issues with the following features. *

Please choose the appropriate response for each item:

	1 never	2	3	4	5 very often	Don't know
Location (e.g. issues with accuracy of your position)	○	○	○	○	○	○
Position stability (e.g. issues with jumping labels)	○	○	○	○	○	○
Screen size (e.g. issues with size of the elements on the screen, size of camera view)	○	○	○	○	○	○
Screen quality (e.g. brightness, contrast, color)	○	○	○	○	○	○
Interface design (e.g. user friendly interface, easy to use the different features and parameters)	○	○	○	○	○	○
Content representation (e.g. issues with content overlapping, unreadible)	○	○	○	○	○	○
Network performance (e.g. issues with access/loading time of the displayed data)	○	○	○	○	○	○
Battery performance (e.g. issues with battery life while using application)	○	○	○	○	○	○
General performance (e.g. application seem to slow down the device)	○	○	○	○	○	○
Content quality (e.g. quality of the displayed information)	○	○	○	○	○	○
Content quantity (e.g. availability of content in the different environments)	○	○	○	○	○	○
Handiness of the device while using the AR browser (e.g. finger is occluding the camera, bulky)	○	○	○	○	○	○
Weight of the device while using the AR browser (e.g. tiring, heavy)	○	○	○	○	○	○
The social aspect (e.g. I						



LimeSurvey - AR Browsers Survey 16.01.13 21:24

don't have an issue to use the device anywhere or I felt to look stupid)	○	○	○	○	○	○

24 4. Which functionality do you wish to see in future AR browsers (e.g. more tools, better location, more interactive features)?

Please write your answer here:

25 5. Please describe how you used/have been using an AR browser. *

Please choose **all** that apply:

□ Standing and pointing to an object/direction (e.g. only pointing to a house, pointing towards the TV)

□ Standing and rotating around pointing to different directions (e.g. exploring the environment by rotating while keeping position)

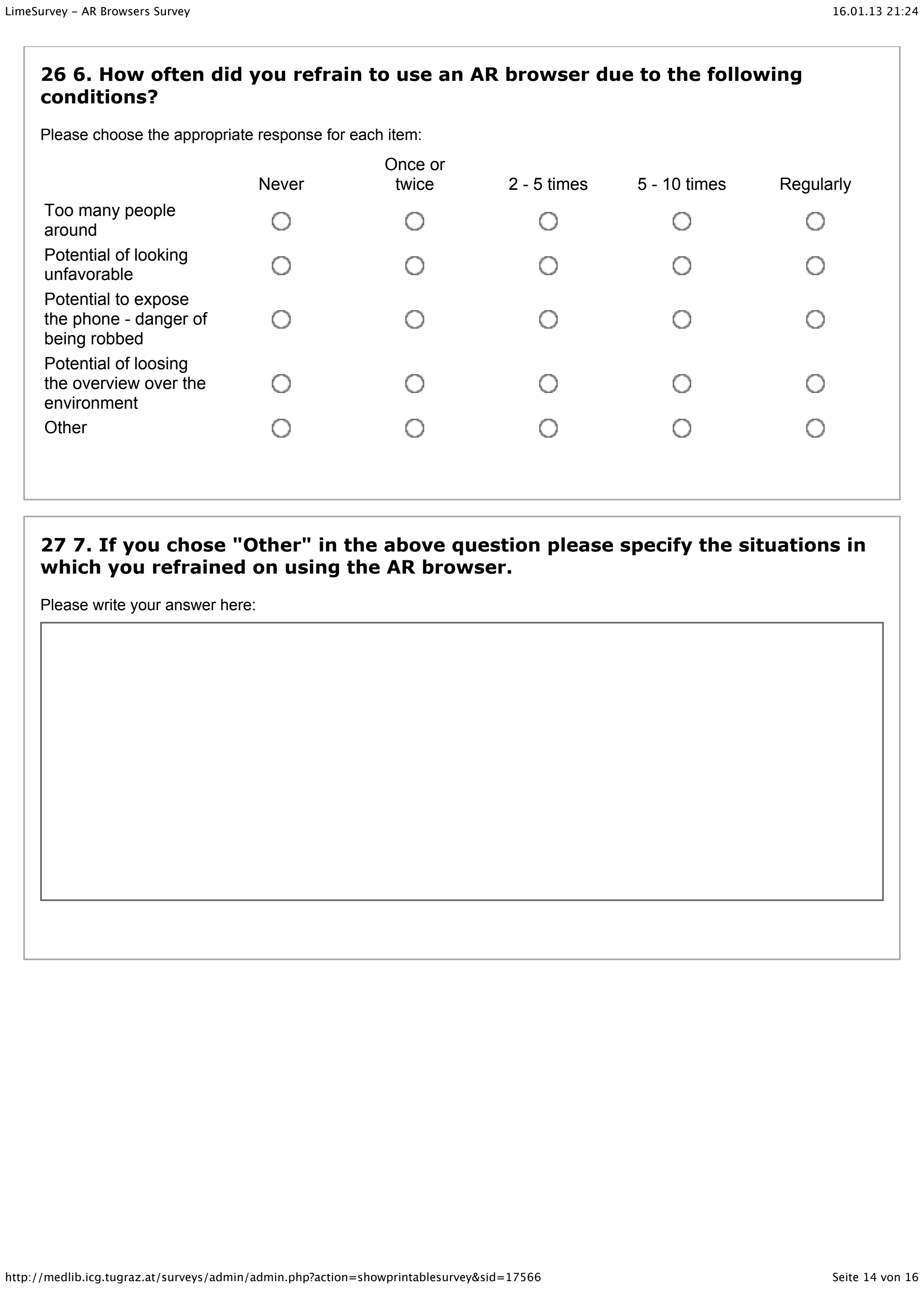
□ Small movements (1-5m) combined with rotational movements (e.g. some steps but mostly keeping position)

□ Larger movements (>5m) combined with rotational movements (e.g. use of AR browser while walking)

□ Multiple Large movements (>5m) combined with rotational movements (e.g. activating the AR browser at different locations).

□ Other:

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 13 von 16



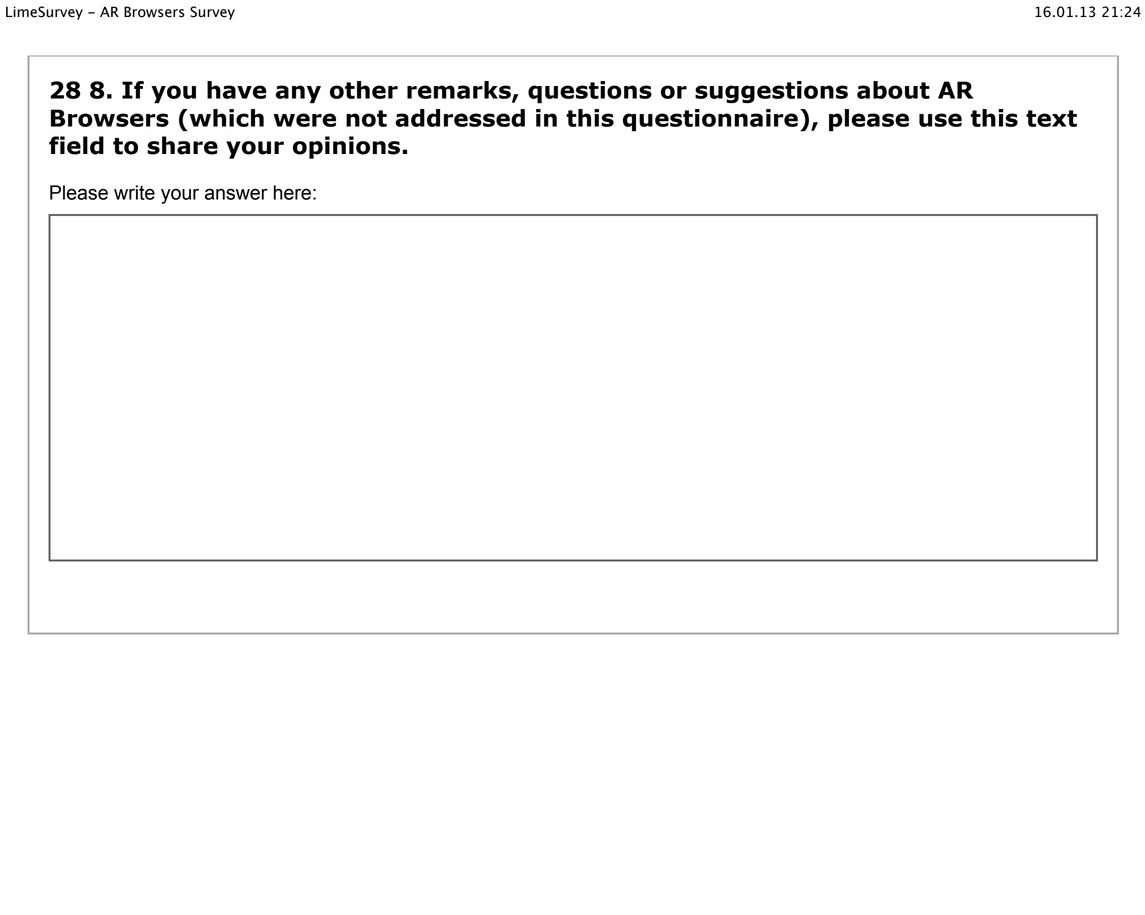
26 6. How often did you refrain to use an AR browser due to the following conditions?

Please choose the appropriate response for each item:

	Never	Once or twice	2 - 5 times	5 - 10 times	Regularly
Too many people around	○	○	○	○	○
Potential of looking unfavorable	○	○	○	○	○
Potential to expose the phone - danger of being robbed	○	○	○	○	○
Potential of loosing the overview over the environment	○	○	○	○	○
Other	○	○	○	○	○

27 7. If you chose "Other" in the above question please specify the situations in which you refrained on using the AR browser.

Please write your answer here:

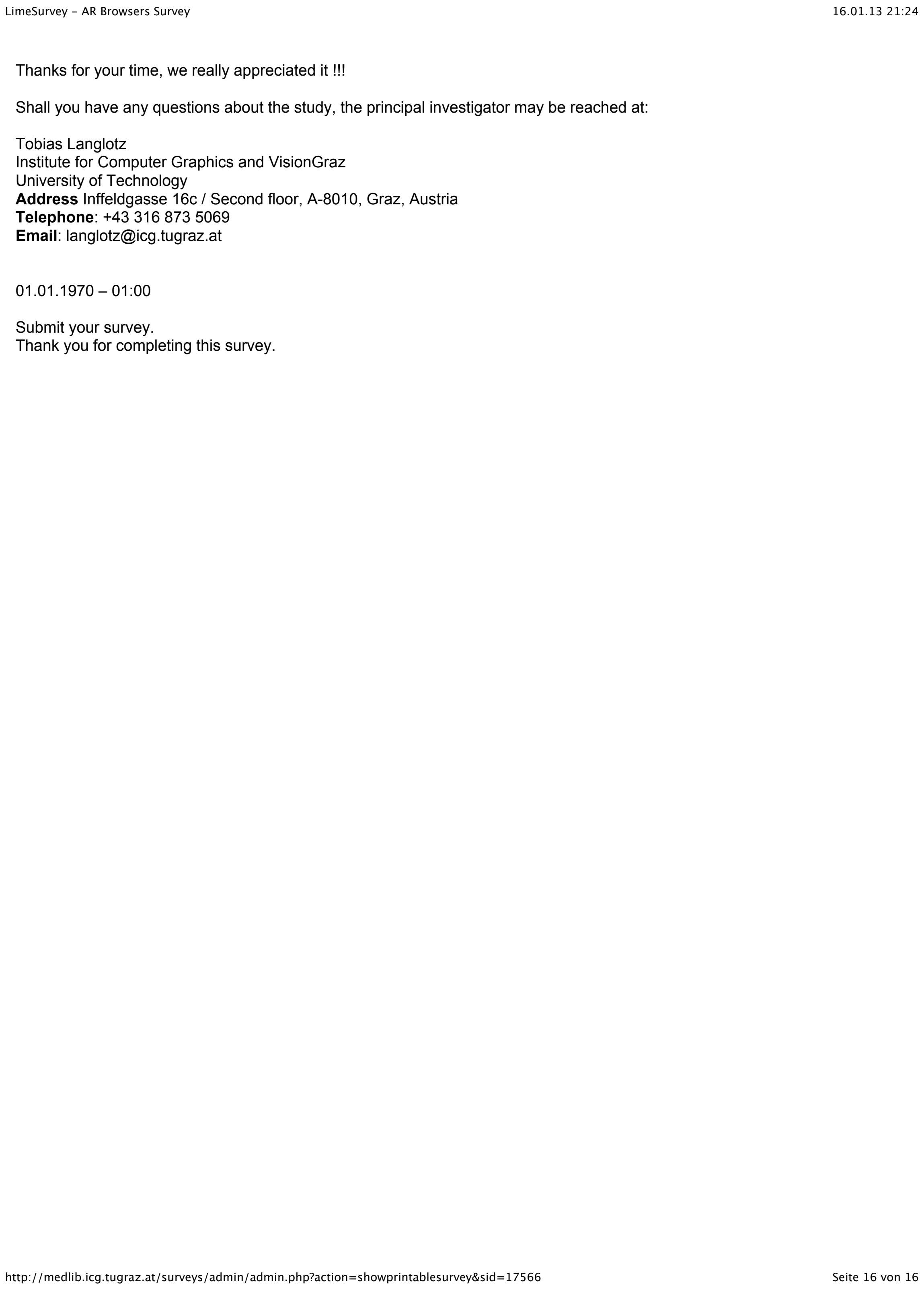


LimeSurvey - AR Browsers Survey 16.01.13 21:24

28 8. If you have any other remarks, questions or suggestions about AR Browsers (which were not addressed in this questionnaire), please use this text field to share your opinions.

Please write your answer here:

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 15 von 16



LimeSurvey - AR Browsers Survey 16.01.13 21:24

Thanks for your time, we really appreciated it !!!

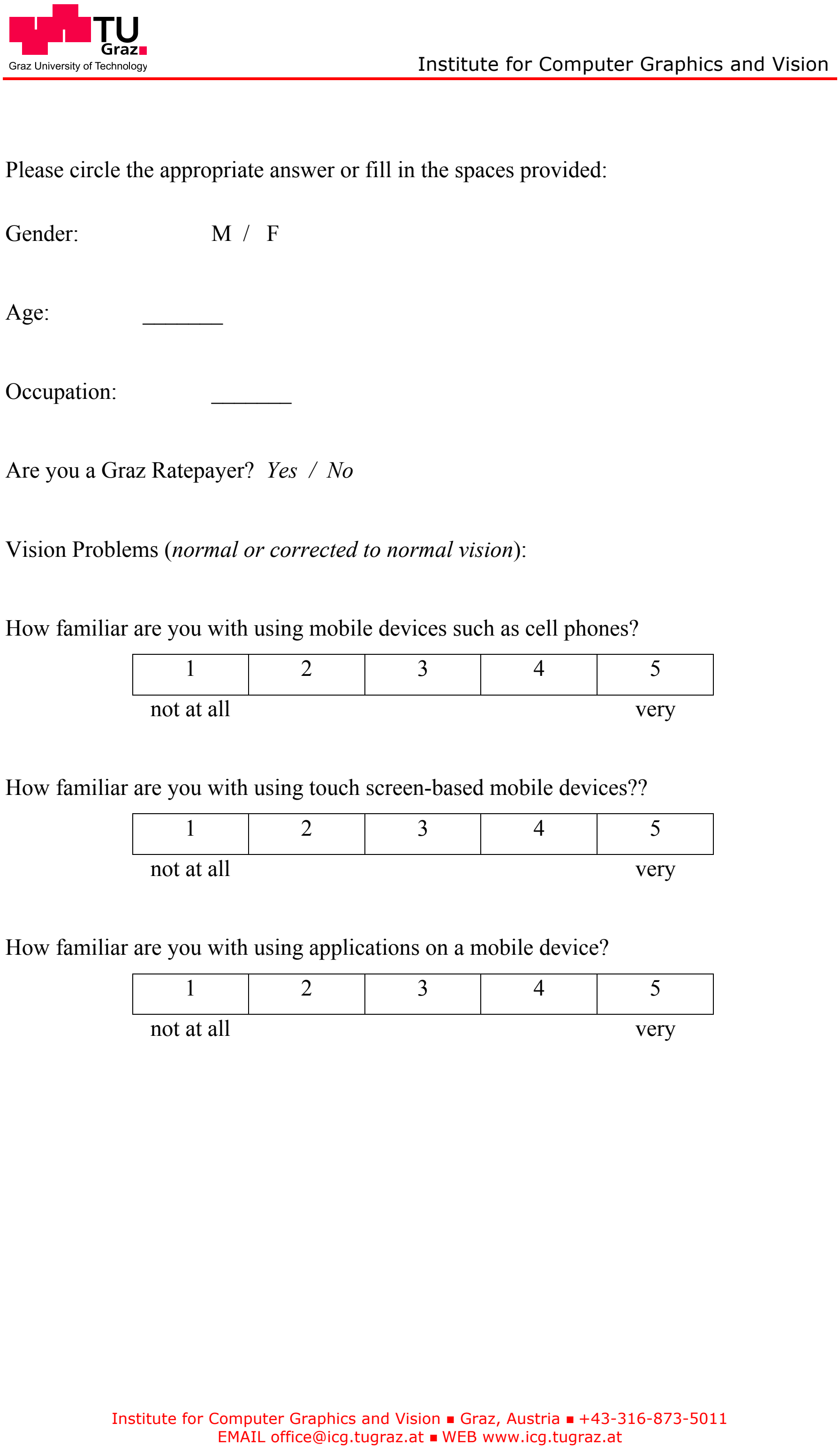
Shall you have any questions about the study, the principal investigator may be reached at:

Tobias Langlotz
Institute for Computer Graphics and VisionGraz
University of Technology
Address Inffeldgasse 16c / Second floor, A-8010, Graz, Austria
Telephone: +43 316 873 5069
Email: langlotz@icg.tugraz.at

01.01.1970 – 01:00

Submit your survey.
Thank you for completing this survey.

http://medlib.icg.tugraz.at/surveys/admin/admin.php?action=showprintablesurvey&sid=17566 Seite 16 von 16



Graz University of Technology

Institute for Computer Graphics and Vision

Please circle the appropriate answer or fill in the spaces provided:

Gender: M / F

Age: _______

Occupation: _______

Are you a Graz Ratepayer? *Yes* / *No*

Vision Problems (*normal or corrected to normal vision*):

How familiar are you with using mobile devices such as cell phones?

1	2	3	4	5

not at all very

How familiar are you with using touch screen-based mobile devices??

1	2	3	4	5

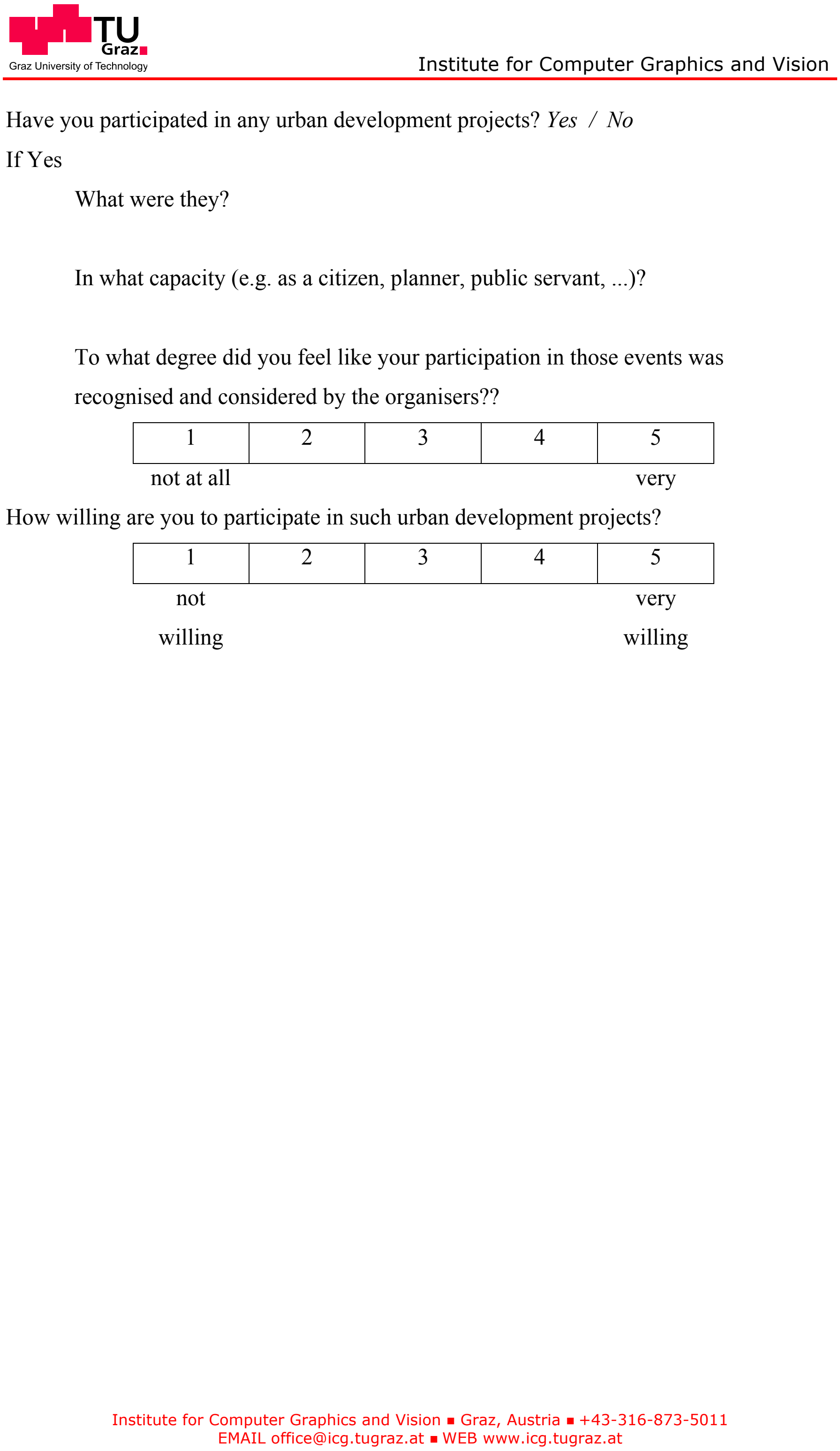
not at all very

How familiar are you with using applications on a mobile device?

1	2	3	4	5

not at all very

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



Graz University of Technology

Institute for Computer Graphics and Vision

Have you participated in any urban development projects? *Yes / No*

If Yes

What were they?

In what capacity (e.g. as a citizen, planner, public servant, ...)?

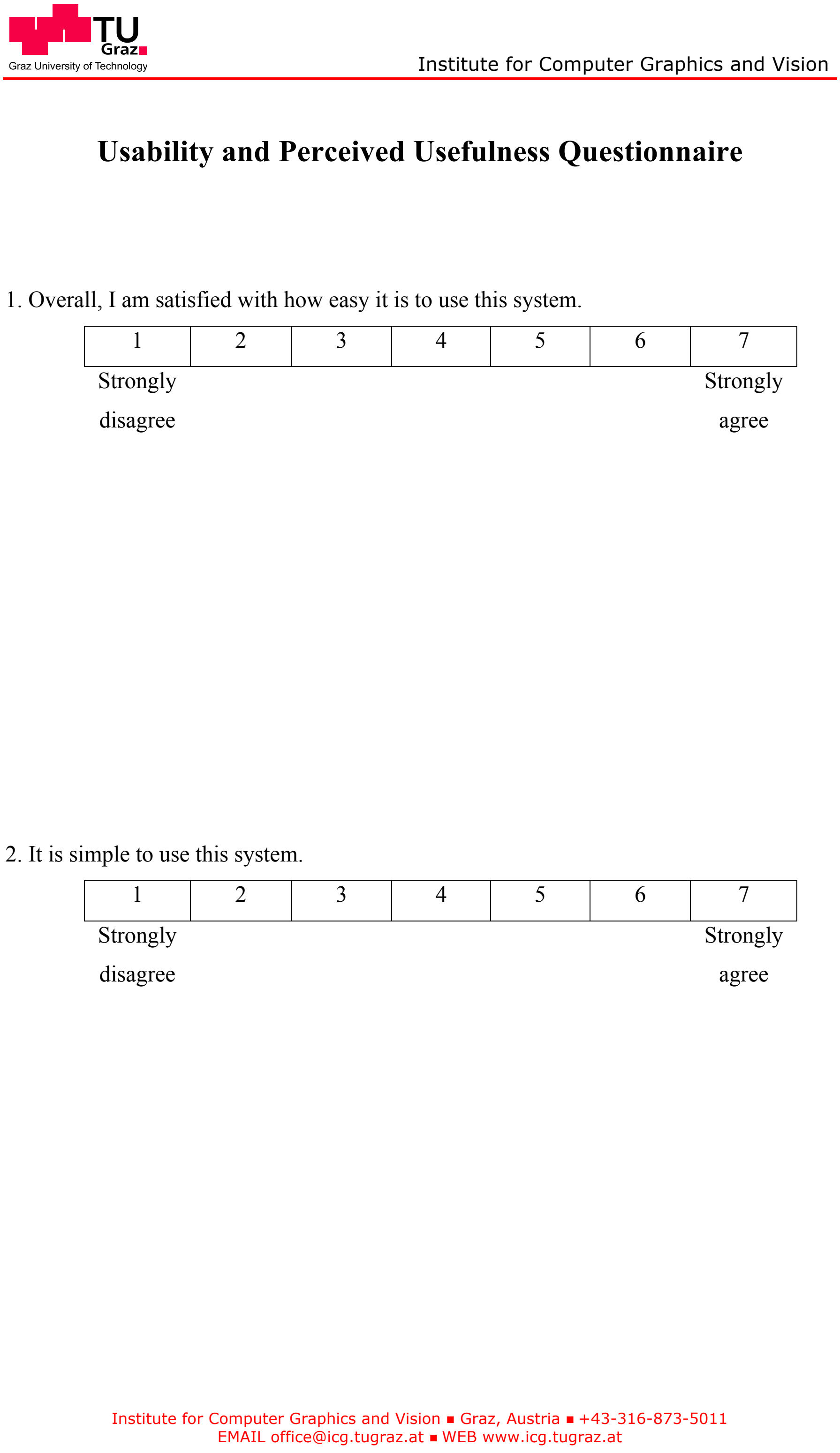
To what degree did you feel like your participation in those events was recognised and considered by the organisers??

1	2	3	4	5
not at all				very

How willing are you to participate in such urban development projects?

1	2	3	4	5
not willing				very willing

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



Graz University of Technology

Institute for Computer Graphics and Vision

Usability and Perceived Usefulness Questionnaire

1. Overall, I am satisfied with how easy it is to use this system.

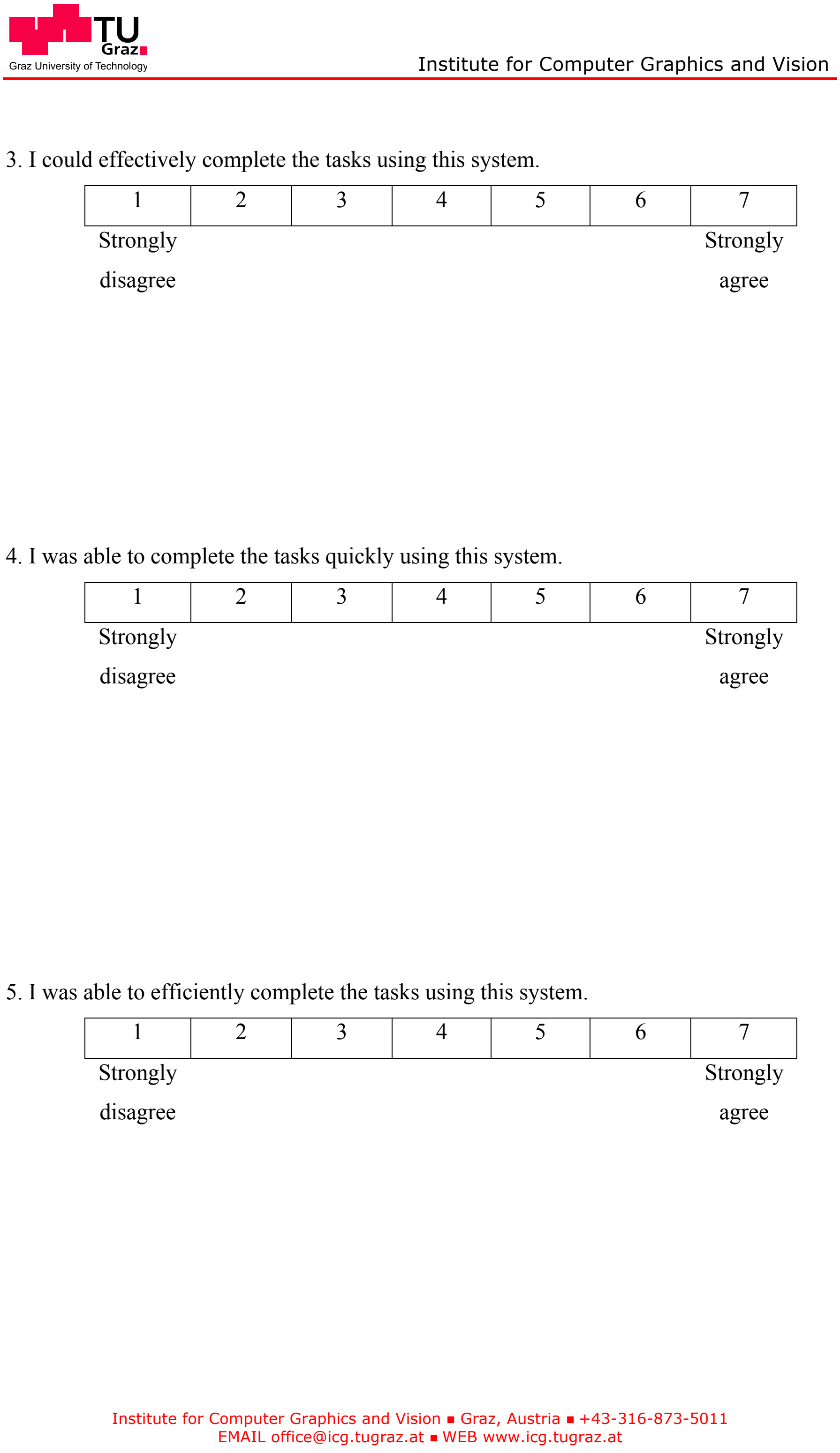
1	2	3	4	5	6	7

Strongly disagree — Strongly agree

2. It is simple to use this system.

1	2	3	4	5	6	7

Strongly disagree — Strongly agree



Graz University of Technology

Institute for Computer Graphics and Vision

3. I could effectively complete the tasks using this system.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

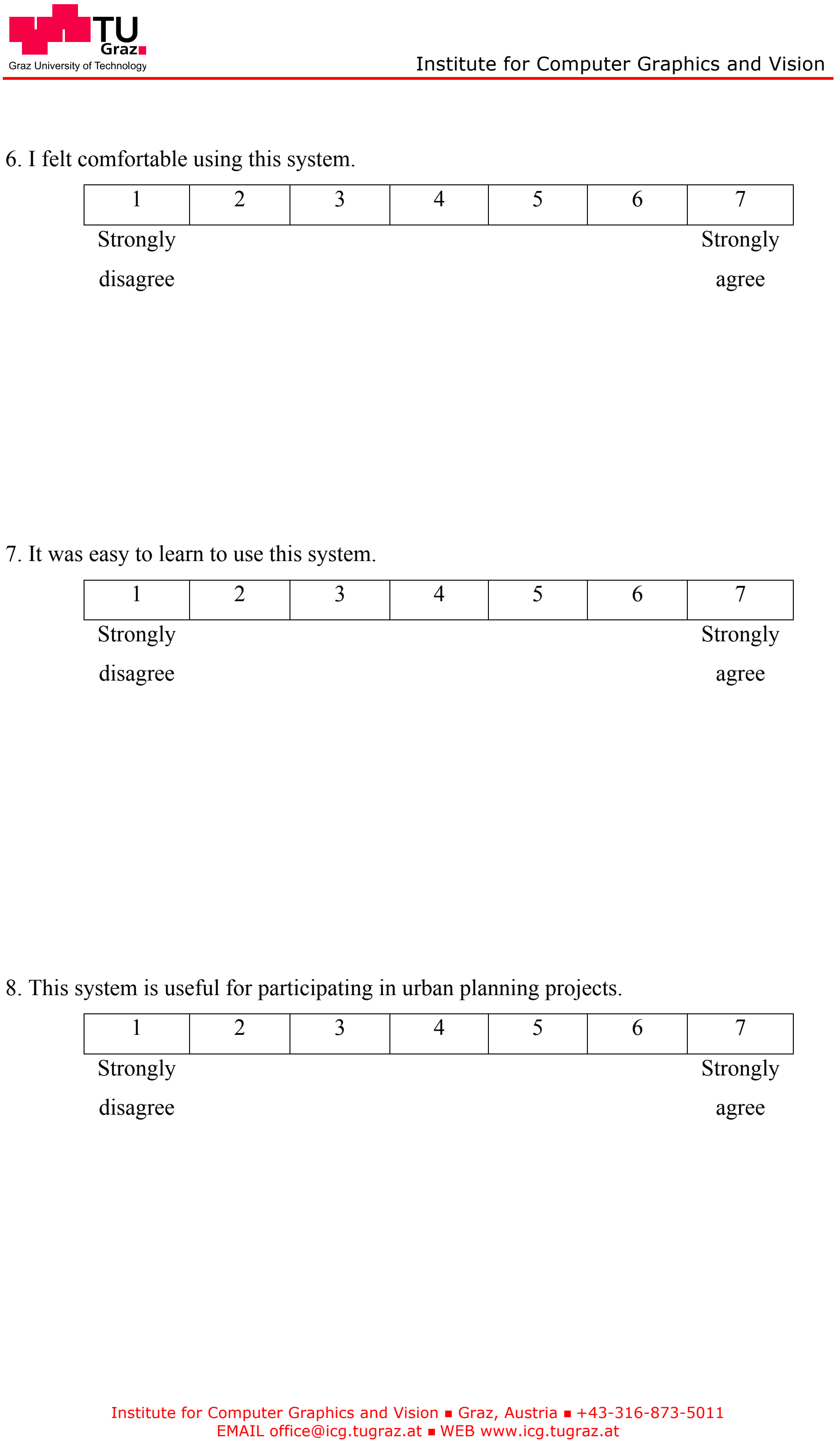
4. I was able to complete the tasks quickly using this system.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

5. I was able to efficiently complete the tasks using this system.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



TU Graz
Graz University of Technology

Institute for Computer Graphics and Vision

6. I felt comfortable using this system.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

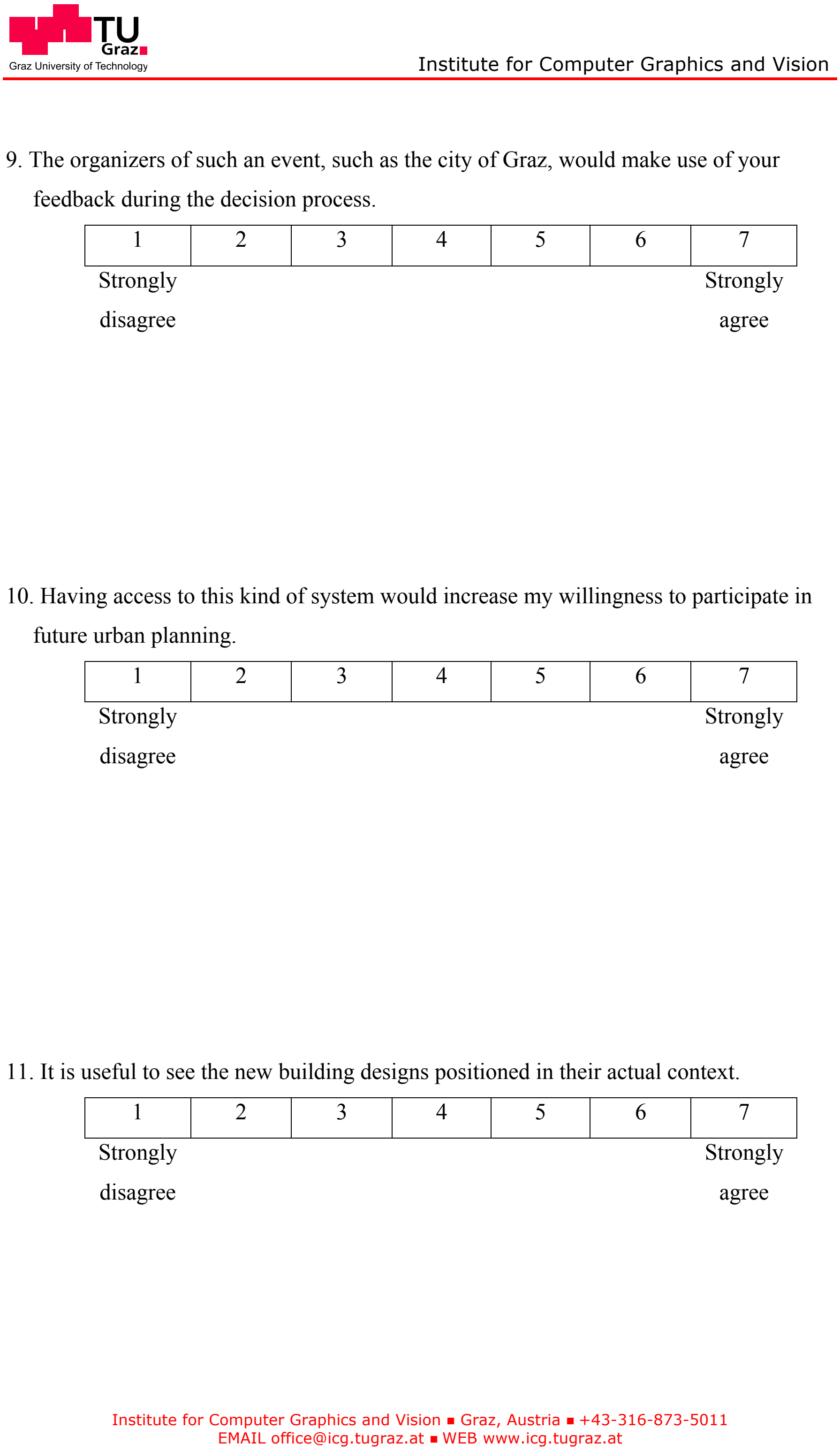
7. It was easy to learn to use this system.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

8. This system is useful for participating in urban planning projects.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



Graz University of Technology

Institute for Computer Graphics and Vision

9. The organizers of such an event, such as the city of Graz, would make use of your feedback during the decision process.

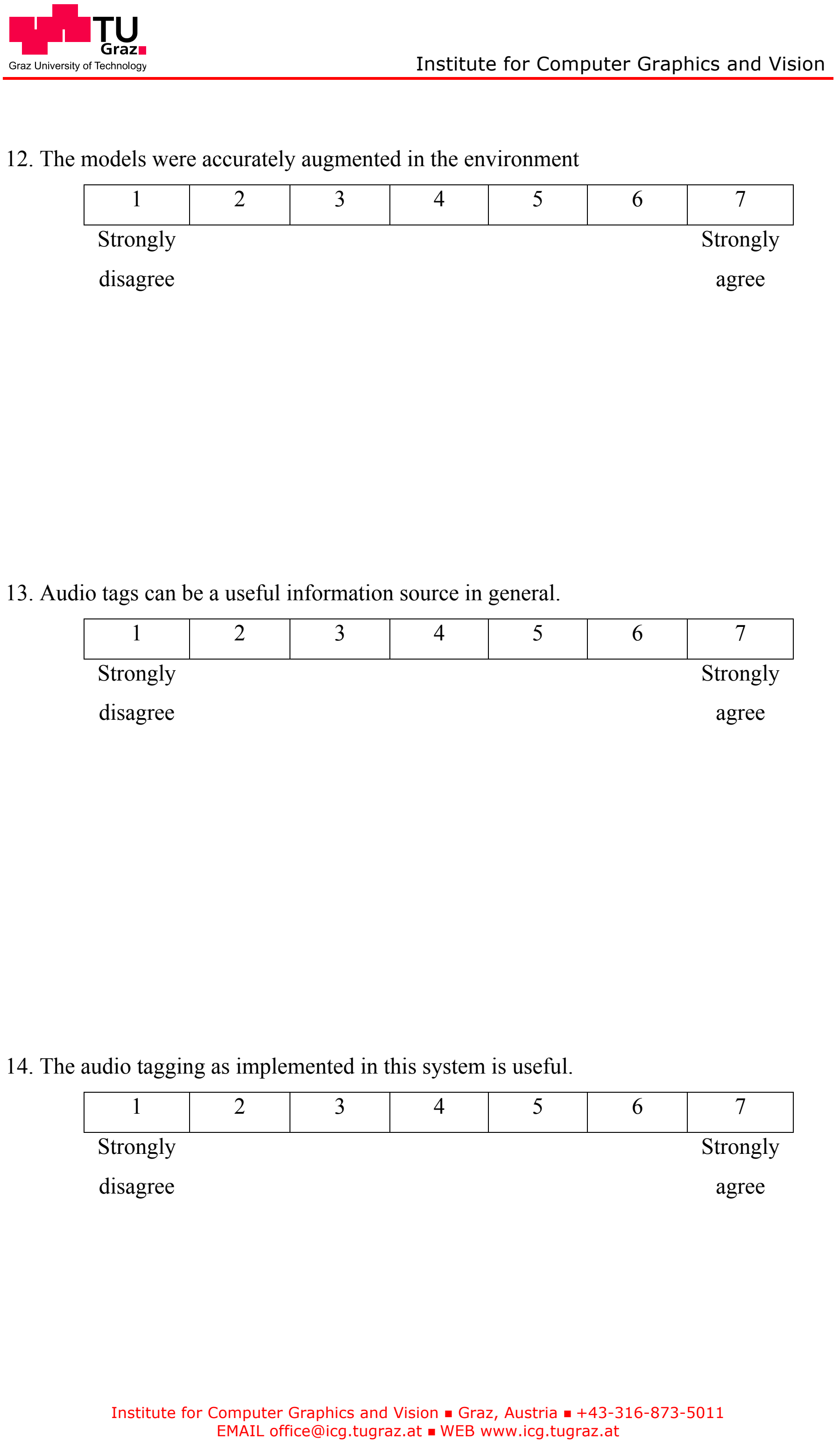
1	2	3	4	5	6	7
Strongly disagree						Strongly agree

10. Having access to this kind of system would increase my willingness to participate in future urban planning.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

11. It is useful to see the new building designs positioned in their actual context.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree



12. The models were accurately augmented in the environment

1	2	3	4	5	6	7

Strongly disagree — Strongly agree

13. Audio tags can be a useful information source in general.

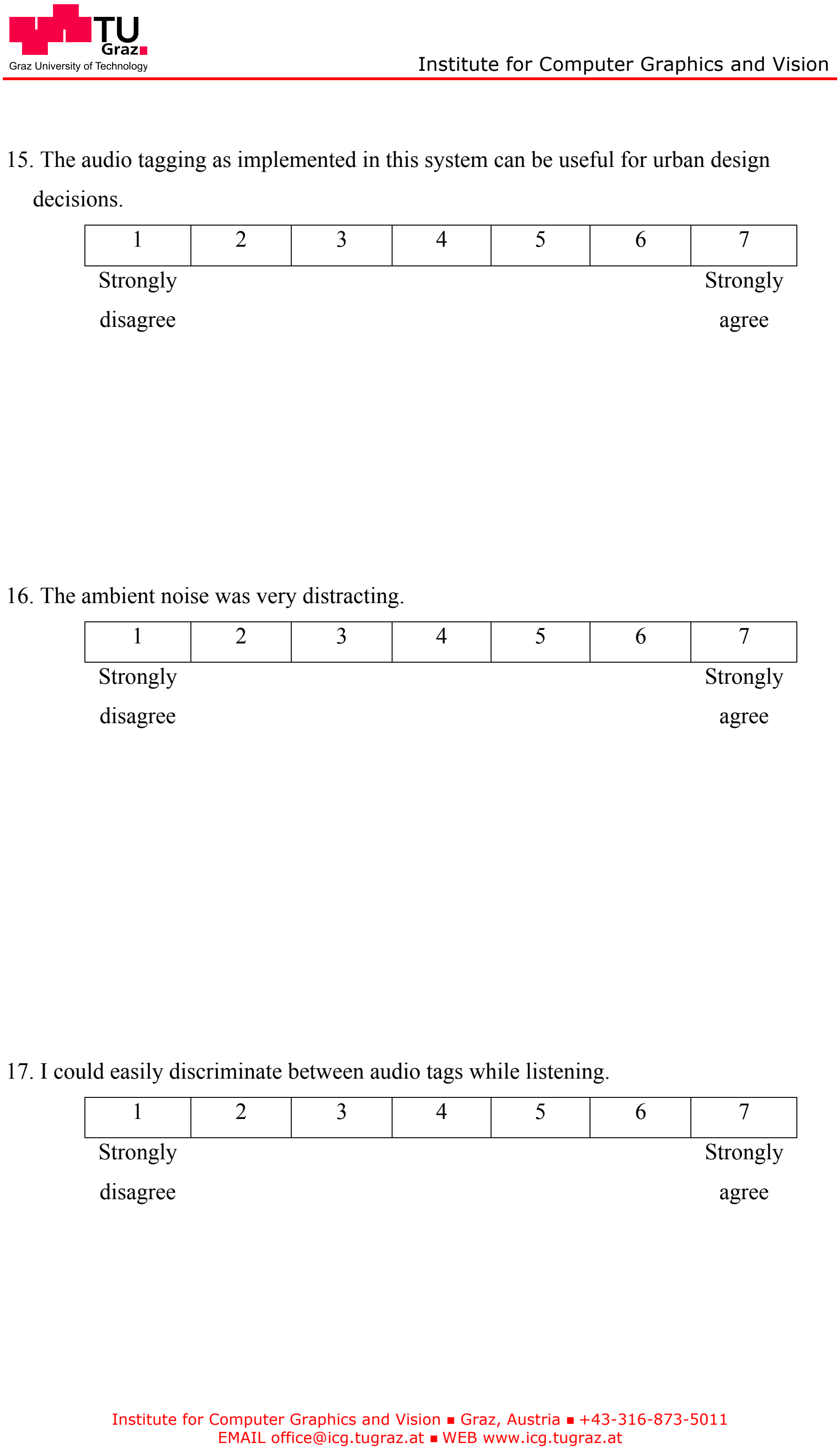
1	2	3	4	5	6	7

Strongly disagree — Strongly agree

14. The audio tagging as implemented in this system is useful.

1	2	3	4	5	6	7

Strongly disagree — Strongly agree



TU Graz
Graz University of Technology

Institute for Computer Graphics and Vision

15. The audio tagging as implemented in this system can be useful for urban design decisions.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

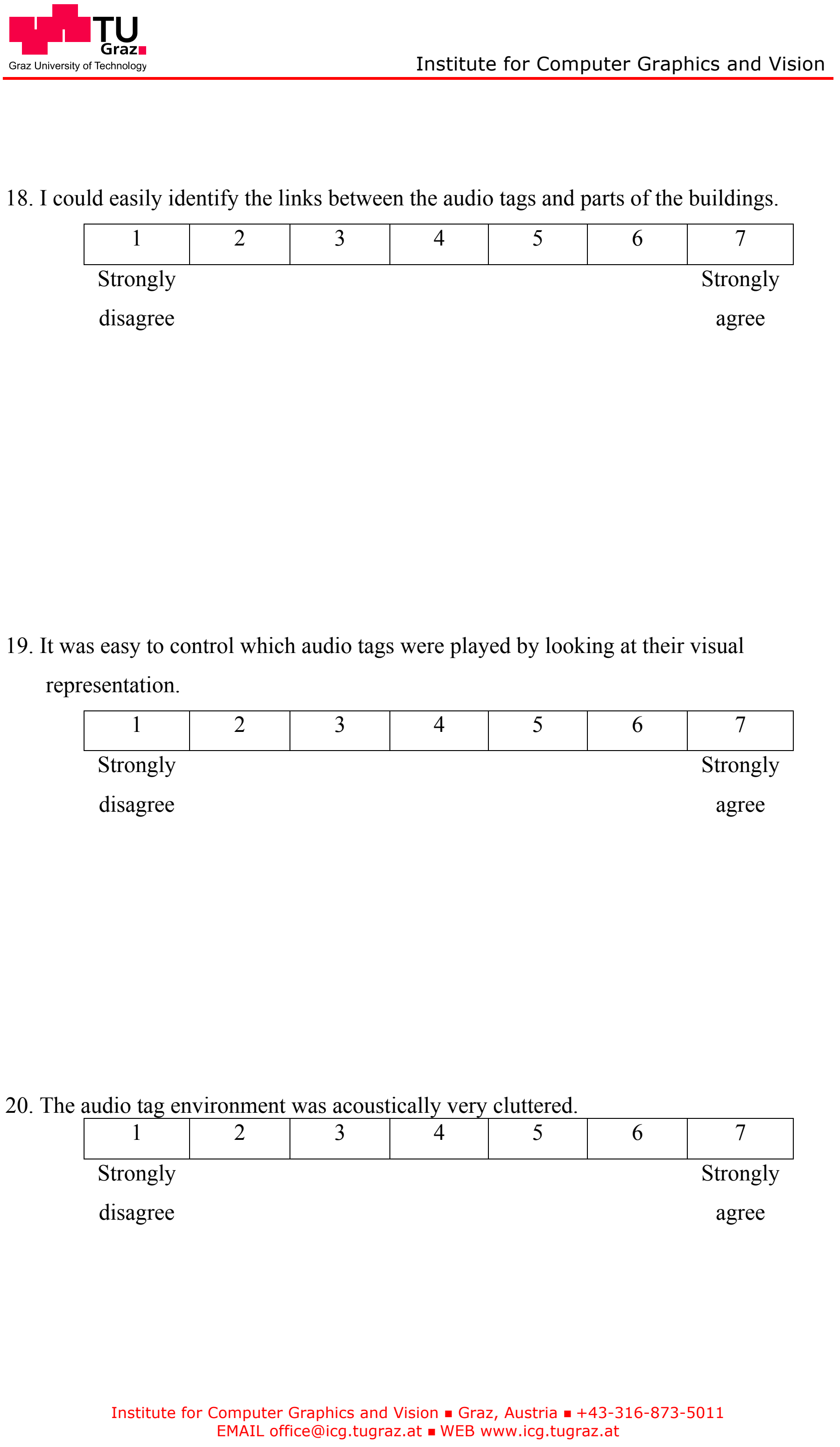
16. The ambient noise was very distracting.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

17. I could easily discriminate between audio tags while listening.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



TU Graz
Graz University of Technology

Institute for Computer Graphics and Vision

18. I could easily identify the links between the audio tags and parts of the buildings.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

19. It was easy to control which audio tags were played by looking at their visual representation.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

20. The audio tag environment was acoustically very cluttered.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

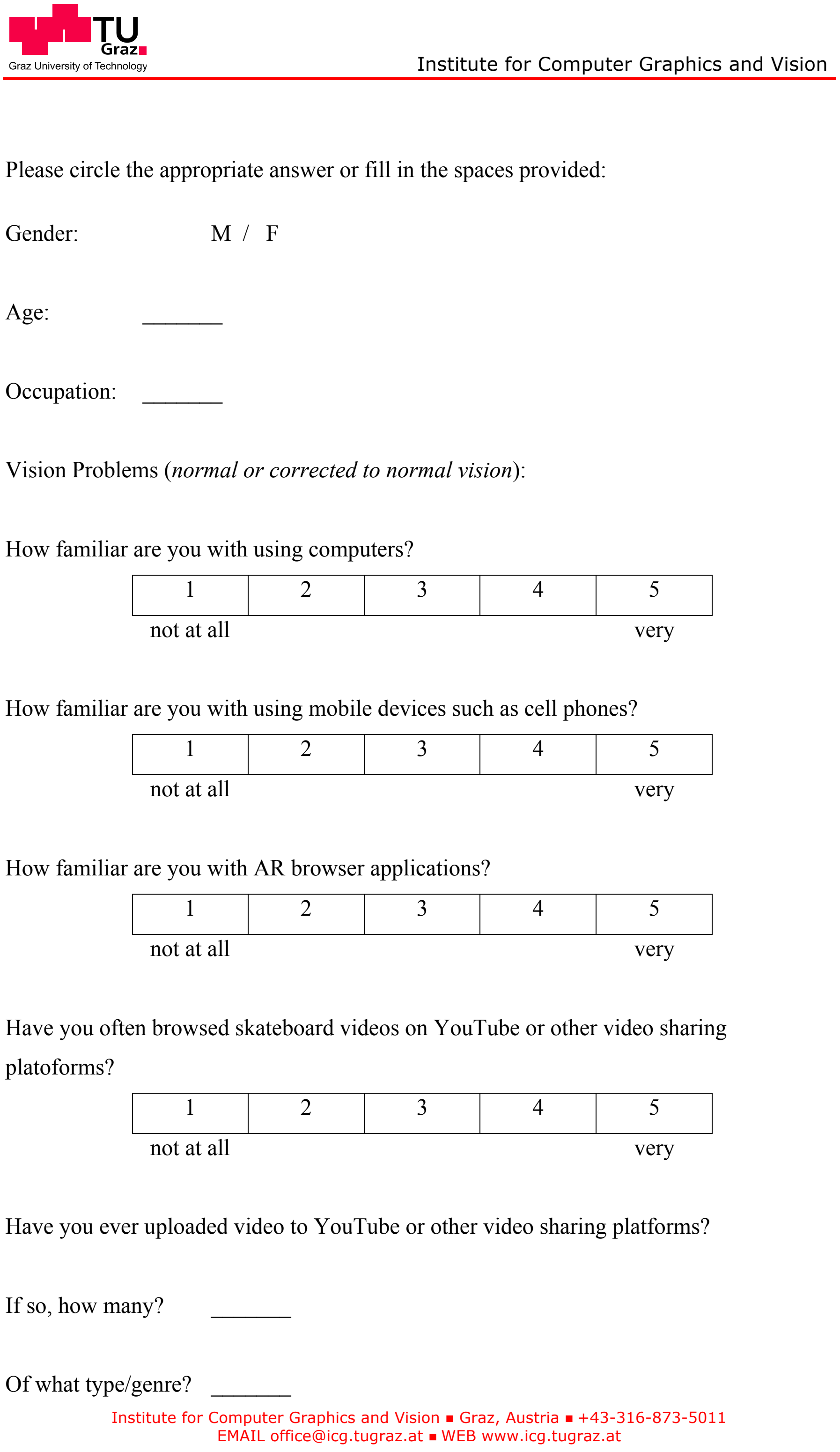
Institute for Computer Graphics and Vision ■ Graz, Austria ■ +43-316-873-5011
EMAIL office@icg.tugraz.at ■ WEB www.icg.tugraz.at

21. The presented system was visually cluttered.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

22. In terms of this urban design scenario, what information do you think was missing from the smart phone application?

23. What general suggestions would you make to help improve this system for public participation in urban planning projects?



TU Graz
Graz University of Technology

Institute for Computer Graphics and Vision

Please circle the appropriate answer or fill in the spaces provided:

Gender: M / F

Age: _______

Occupation: _______

Vision Problems (*normal or corrected to normal vision*):

How familiar are you with using computers?

1	2	3	4	5
not at all				very

How familiar are you with using mobile devices such as cell phones?

1	2	3	4	5
not at all				very

How familiar are you with AR browser applications?

1	2	3	4	5
not at all				very

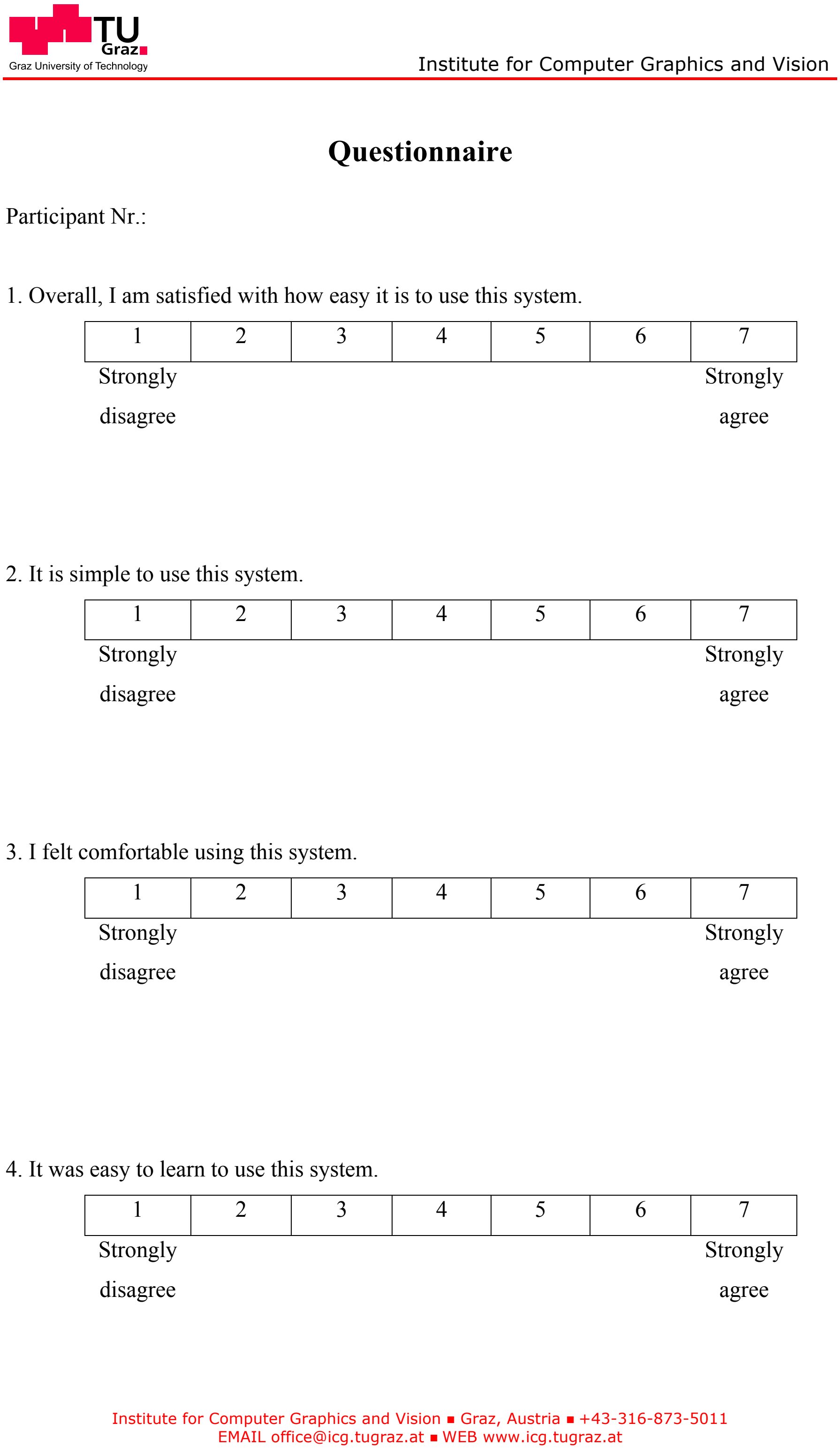
Have you often browsed skateboard videos on YouTube or other video sharing platoforms?

1	2	3	4	5
not at all				very

Have you ever uploaded video to YouTube or other video sharing platforms?

If so, how many? _______

Of what type/genre? _______



Graz University of Technology

Institute for Computer Graphics and Vision

Questionnaire

Participant Nr.:

1. Overall, I am satisfied with how easy it is to use this system.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

2. It is simple to use this system.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

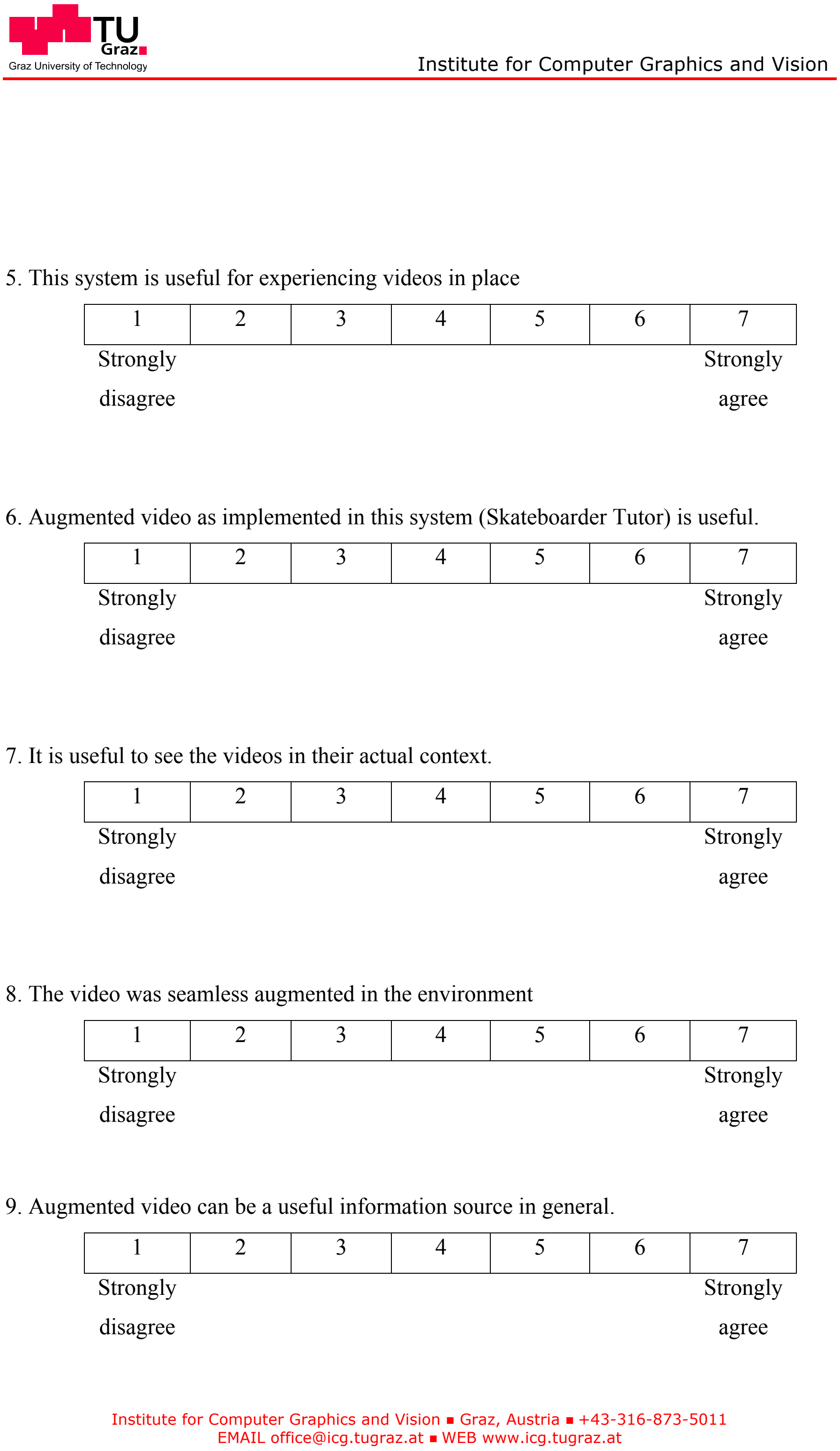
3. I felt comfortable using this system.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

4. It was easy to learn to use this system.

1	2	3	4	5	6	7
Strongly						Strongly
disagree						agree

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



TU Graz
Graz University of Technology

Institute for Computer Graphics and Vision

5. This system is useful for experiencing videos in place

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

6. Augmented video as implemented in this system (Skateboarder Tutor) is useful.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

7. It is useful to see the videos in their actual context.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

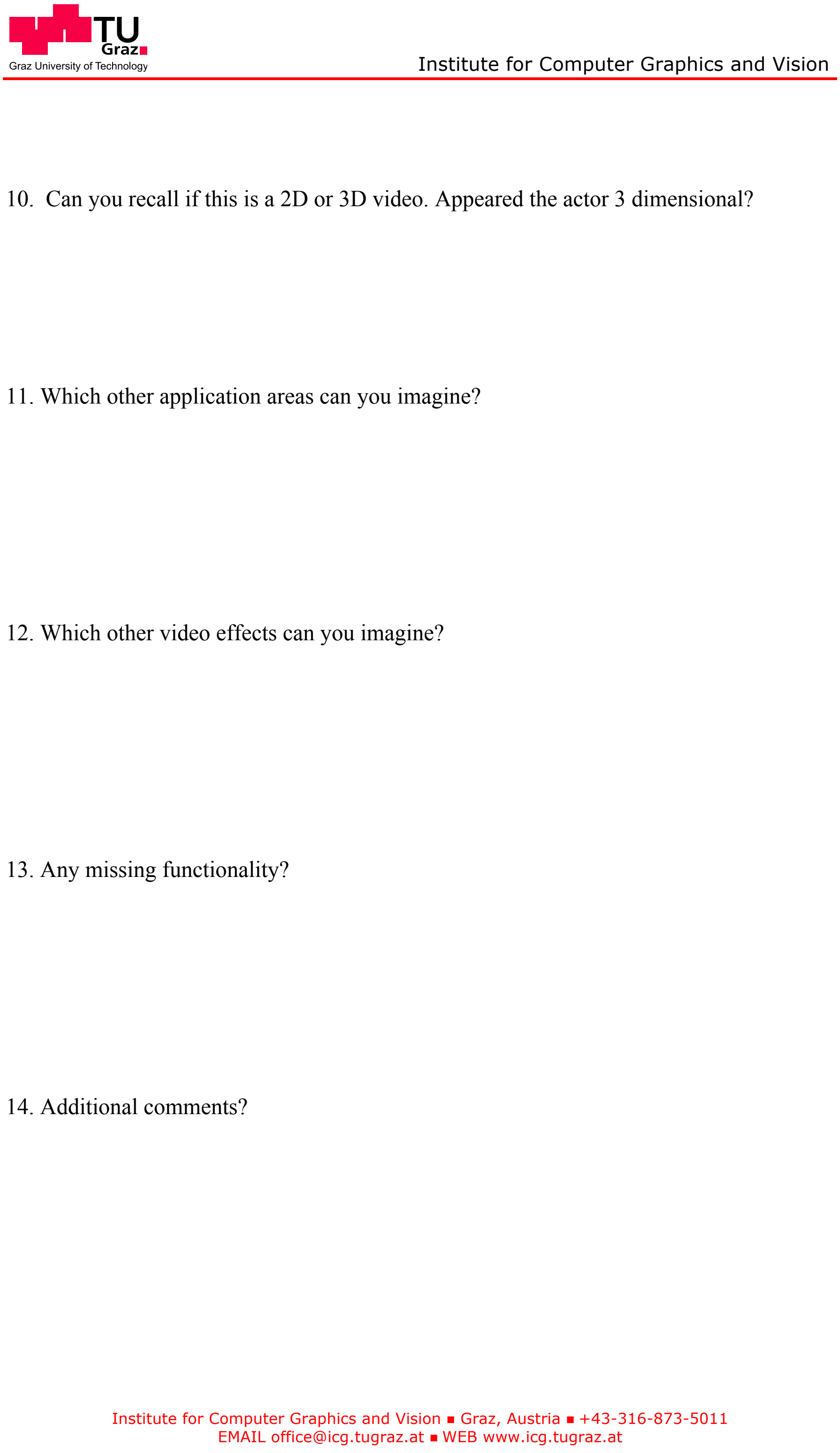
8. The video was seamless augmented in the environment

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

9. Augmented video can be a useful information source in general.

1	2	3	4	5	6	7
Strongly disagree						Strongly agree

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at



Institute for Computer Graphics and Vision

10. Can you recall if this is a 2D or 3D video. Appeared the actor 3 dimensional?

11. Which other application areas can you imagine?

12. Which other video effects can you imagine?

13. Any missing functionality?

14. Additional comments?

Institute for Computer Graphics and Vision ▪ Graz, Austria ▪ +43-316-873-5011
EMAIL office@icg.tugraz.at ▪ WEB www.icg.tugraz.at

Bibliography

[1] Adams, A., Horowitz, M., Park, S. H., Gelfand, N., Baek, J., Matusik, W., Levoy, M., Jacobs, D. E., Dolson, J., Tico, M., Pulli, K., Talvala, E.-V., Ajdin, B., Vaquero, D., and Lensch, H. P. A. (2010). The Frankencamera. *ACM Transactions on Graphics*, 29(4):1–12.

[2] Arth, C., Klopschitz, M., Reitmayr, G., and Schmalstieg, D. (2011). Real-time self-localization from panoramic images on mobile devices. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 37–46. IEEE.

[3] Arth, C., Mulloni, A., and Schmalstieg, D. (2012). Exploiting sensors on mobile phones to improve wide-area localization. In *ICPR*, pages 2152–2156.

[4] Azuma, R. (1993). Tracking requirements for augmented reality. *Communications of the ACM*, 36(7):50–51.

[5] Azuma, R. (1995). A Survey of Augmented Reality. 6:355–385.

[6] Baillot, Y., Brown, D., and Julier, S. (2001). Authoring of Physical Models Using Mobile Computers. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, page 39.

[7] Ballan, L., Brostow, G. J., Puwein, J., and Pollefeys, M. (2010). Unstructured video-based rendering: Interactive Exploration of Casually Captured Videos. In *ACM SIGGRAPH 2010 papers on - SIGGRAPH '10*, volume 29, page 1, New York, New York, USA. ACM Press.

[8] Bastian, J., Ward, B., Hill, R., van den Hengel, A., and Dick, A. (2010). Interactive modelling for AR applications. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 199–205. IEEE.

[9] Bederson, B. B. (1995). Audio augmented reality. In *Conference companion on Human factors in computing systems - CHI '95*, pages 210–211, New York, New York, USA. ACM Press.

[10] Behringer, R., Chen, S., Sundareswaran, V., Wang, K., and Vassiliou, M. (1999). *A novel interface for device diagnostics using speech recognition, augmented reality visualization, and 3D audio auralization.* IEEE Comput. Soc.

[11] Bell, B., Feiner, S., and Hollerer, T. (2002). Information at a glance [augmented reality user interfaces]. *IEEE Computer Graphics and Applications*, 22(4):6–9.

[12] Böhmer, M., Hecht, B., Schöning, J., Krüger, A., and Bauer, G. (2011). Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 47–56, New York, NY, USA. ACM.

[13] Briechle, K. and Hanebeck, U. D. (2001). Template matching using fast normalized cross correlation. 4387:95–102.

[14] Bunnun, P., Damen, D., Calway, A., and Mayol-Cuevas, W. (2012a). Integrating 3D object detection, modelling and tracking on a mobile phone. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 273–274. IEEE.

[15] Bunnun, P. and Mayol-Cuevas, W. W. (2008). OutlinAR: an assisted interactive model building system with reduced computational effort. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 61–64. IEEE.

[16] Bunnun, P., Subramanian, S., and Mayol-Cuevas, W. W. (2012b). In-Situ interactive image-based model building for Augmented Reality from a handheld device. *Virtual Reality*.

[17] Caudell, T. and Mizell, D. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, pages 659–669 vol.2. IEEE.

[18] Chekhlov, D., Gee, A., Calway, A., and Mayol-Cuevas, W. (2007). Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *International Symposium on Mixed and Augmented Reality (ISMAR)*.

[19] Chen, D. M., Baatz, G., Koser, K., Tsai, S. S., Vedantham, R., Pylvanainen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., Girod, B., and Grzeszczuk, R. (2011). City-scale landmark identification on mobile devices. In *CVPR 2011*, pages 737–744. IEEE.

[20] Cheok, A. D., Goh, K. H., Liu, W., Farbiz, F., Fong, S. W., Teo, S. L., Li, Y., and Yang, X. (2004). Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal and Ubiquitous Computing*, 8(2):71–81.

[21] Corbin, J. and Strauss, A. (2008). *Basics of Qualitative Research*. Sage, Thousand Oaks, Cal. 3rd edition.

[22] Criminisi, a., Reid, I., and Zisserman, A. (1999). Single view metrology. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 434–441 vol.1. Ieee.

[23] DiVerdi, S., Wither, J., and Hollerer, T. (2008). Envisor: Online Environment Map Construction for Mixed Reality. In *Proceedings of the IEEE Virtual Reality Conference 2008*, pages 19–26. IEEE.

[24] Doan, A., Ramakrishnan, R., and Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, 54(4):86.

[25] Dow, S., Lee, J., Oezbek, C., Maclntyre, B., Bolter, J. D., and Gandy, M. (2005). Exploring spatial narratives and mixed reality experiences in Oakland Cemetery. In

Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology - ACE '05, pages 51–60, New York, New York, USA. ACM Press.

[26] Dünser, A., Grasset, R., and Billinghurst, M. (2008). *A survey of evaluation techniques used in augmented reality studies*. Human Interface Technology Laboratory New Zealand.

[27] Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore, E., and Bylund, M. (2001). GeoNotes: Social and Navigational Aspects of Location-Based Information Systems. pages 2–17.

[28] Feiner, S., MacIntyre, B., Höllerer, T., and Webster, A. (1997). A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal Technologies*, 1(4):208–217.

[29] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

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

[31] Freeman, R. and Steed, A. (2006). Interactive modelling and tracking for mixed and augmented reality. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '06*, page 61, New York, New York, USA. ACM Press.

[32] Fründ, J., Geiger, C., Grafe, M., and Kleinjohann, B. (2001). The augmented reality personal digital assistant. In *The 2nd International Symposium on Mixed Reality ISMR2001, The Virtual Reality Society of Japan*, Japan.

[33] Grasset, R., Langlotz, T., Kalkofen, D., Tatzgern, M., and Schmalstieg, D. (2012). Image-driven view management for augmented reality browsers. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 177–186. IEEE.

[34] Grubert, J., Langlotz, T., and Grasset, R. (2012). Augmented Reality Browser Survey. Technical report, Graz University of Technology.

[35] Guven, S. (2006). *Authoring and presenting situated media in augmented and virtual reality*. PhD thesis, New York, NY, USA. AAI3213515.

[36] Guven, S. and Feiner, S. (2004). Authoring 3D hypermedia for wearable augmented and virtual reality. In *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, pages 118–126. IEEE.

[37] Hallaway, D., Feiner, S., and Höllerer, T. (2004). Bridging the gaps: Hybrid tracking for adaptive mobile augmented reality. *Applied Artificial Intelligence*, 18(6):477–500.

[38] Haller, M., Dobler, D., and Stampfl, P. (2002). Augmenting the reality with 3D sound sources. In *ACM SIGGRAPH 2002 conference abstracts and applications on - SIGGRAPH '02*, page 65, New York, New York, USA. ACM Press.

[39] Haller, M., Stauder, E., and Zauner, J. (2005). Amire-es: Authoring mixed reality once, run it anywhere. In *11th International Conference on Human-Computer Interaction (HCII-2005)*, pages 22–27.

[40] Hansen, F. A. (2006). Ubiquitous annotation systems. In *Proceedings of the seventeenth conference on Hypertext and hypermedia - HYPERTEXT '06*, page 121, New York, New York, USA. ACM Press.

[41] Haringer, M. and Regenbrecht, H. T. (2002). A Pragmatic Approach to Augmented Reality Authoring. page 237.

[42] Henrysson, A. and Billinghurst, M. (2007). Using a mobile phone for 6 DOF mesh editing. In *CHINZ; Vol. 254*, pages 9–16.

[43] Henrysson, A., Billinghurst, M., and Ollila, M. (2005a). Face to face collaborative AR on mobile phones. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, pages 80–89. IEEE.

[44] Henrysson, A., Ollila, M., and Billinghurst, M. (2005b). Mobile phone based AR scene assembly. In *ACM International Conference Proceeding Series; Vol. 154*, page 95.

[45] Höllerer, T. and Feiner, S. (2004). Mobile Augmented Reality. *Telegeoinformatics: Location-based Computing and Services*, pages 1–39.

[46] Höllerer, T., Feiner, S., and Pavlik, J. (1999a). Situated Documentaries: Embedding Multimedia Presentations in the Real World. In *In Proceedings of the 3rd IEEE International Symposium on Wearable Computers (ISWC '99)*, pages 79–86.

[47] Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., and Hallaway, D. (1999b). Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6):779–785.

[48] Höllerer, T., Wither, J., and DiVerdi, S. (2007). Anywhere augmentation: Towards mobile augmented reality in unprepared environments. In Gartner, G., Cartwright, W., and Peterson, M., editors, *Location Based Services and TeleCartography*, Lecture Notes in Geoinformation and Cartography, pages 393–416. Springer Berlin Heidelberg.

[49] Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America. A*, 4(4):629–642.

[50] Jiang, B. and Neumann, U. (2004). A robust hybrid tracking system for outdoor augmented reality. In *IEEE Virtual Reality 2004*, pages 3–275. IEEE.

[51] Kang, H. and Shneiderman, B. (2000). Visualization methods for personal photo collections: browsing and searching in the PhotoFinder. In *International Conference on Multimedia Computing and Systems/International Conference on Multimedia and Expo - ICME(ICMCS)*, volume 3, pages 1539–1542 vol.3.

[52] Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59 – 68.

[53] Kato, H. and Billinghurst, M. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94. IEEE Comput. Soc.

[54] Kim, K., Lepetit, V., and Woo, W. (2010). Keyframe-based modeling and tracking of multiple 3D objects. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 193–198. IEEE.

[55] Kim, S., DiVerdi, S., Chang, J. S., Kang, T., Iltis, R., and Höllerer, T. (2007). Implicit 3D modeling and tracking for anywhere augmentation. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology - VRST '07*, page 19, New York, New York, USA. ACM Press.

[56] Klein, G. and Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. In *Symposium on Mixed and Augmented Reality*, pages 1–10.

[57] Klein, G. and Murray, D. (2008). Improving the agility of keyframe-based SLAM. In *Proc. 10th European Conference on Computer Vision (ECCV'08)*, pages 802–815, Marseille.

[58] Klein, G. and Murray, D. (2009). Parallel Tracking and Mapping on a camera phone. *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, 41(1):83–86.

[59] Knöpfle, C., Weidenhausen, J., and Chauvigné, L. (2005). Template Based Authoring for AR based Service Scenarios. *Proceedings of the IEEE*, 2005:237–240.

[60] Kooper, R. and MacIntyre, B. (2003). Browsing the Real-World Wide Web: Maintaining awareness of virtual information in an AR information space. *International Journal of Human-Computer Interaction*, 16(3):425–446.

[61] Kretschmer, U., Coors, V., Spierling, U., Grasbon, D., Schneider, K., Rojas, I., and Malaka, R. (2001). Meeting the spirit of history. In *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage - VAST '01*, page 141, New York, New York, USA. ACM Press.

[62] Kurz, D. and Ben Himane, S. (2011). Inertial sensor-aligned visual feature descriptors. In *CVPR 2011*, pages 161–166. IEEE.

[63] Kurz, D. and Benhimane, S. (2011). Gravity-aware handheld Augmented Reality. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 111–120. IEEE.

[64] Langlotz, Tobias, G. J. and Grasset, R. (2013). Augmented Reality in The Real World: AR Browsers - Essential products or only gadgets? *Communications of the ACM*.

[65] Langlotz, Tobias, R. H. Z. S. and Schmalstieg, D. (2013). Audio Stickies: Visually-guided Spatial Audio Annotations on a Mobile Augmented Reality Platform. *Submitted to ACM OzCHI.*

[66] Langlotz, T., Degendorfer, C., Mulloni, A., Schall, G., Reitmayr, G., and Schmalstieg, D. (2011). Robust detection and tracking of annotations for outdoor augmented reality browsing. *Computers & Graphics*, 35(4):831–840.

[67] Langlotz, T., Mooslechner, S., Zollmann, S., Degendorfer, C., Reitmayr, G., and Schmalstieg, D. (2012a). Sketching up the world: in situ authoring for mobile Augmented Reality. *Personal and Ubiquitous Computing*, 16(6):623–630.

[68] Langlotz, T., Wagner, D., Mulloni, A., and Schmalstieg, D. (2010). Online Creation of Panoramic Augmented Reality Annotations on Mobile Phones. *IEEE Pervasive Computing.*

[69] Langlotz, T., Zingerle, M., Grasset, R., Kaufmann, H., and Reitmayr, G. (2012b). AR Record&Replay. In *Proceedings of the 24th Australian Computer-Human Interaction Conference on - OzCHI '12*, pages 318–326, New York, New York, USA. ACM Press.

[70] Lee, G. A., Kim, G. J., and Billinghurst, M. (2005). Immersive authoring. *Communications of the ACM*, 48(7):76–81.

[71] Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78.

[72] Leykin, A. and Tuceryan, M. (2004). Automatic determination of text readability over textured backgrounds for augmented reality systems. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, pages 224–230, Washington, DC, USA. IEEE Computer Society.

[73] Linz, C., Lipski, C., Rogge, L., Theobalt, C., and Magnor, M. (2010). Space-time visual effects as a post-production process. In *Proceedings of the 1st international workshop on 3D video processing - 3DVP '10*, page 1, New York, New York, USA. ACM Press.

[74] Loomis, J. M., Golledge, R. G., and Klatzky, R. L. (1993). Personal guidance system for the visually impaired using GPS, GIS, and VR technologies. In *Virtual Reality*, pages 71–74.

[75] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI'81 Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679.

[76] MacIntyre, B., Bolter, J. D., Vaughn, J., Hannigan, B., Gandy, M., Moreno, E., Haas, M., Kang, S.-H., Krum, D., and Voida, S. (2003). Three Angry Men: An Augmented-Reality Experiment In Point-Of-View Drama. *IN PROCEEDINGS OF TIDSE 2003*, pages 24 – 26.

[77] MacIntyre, B., Gandy, M., Dow, S., and Bolter, J. D. (2004). DART: a toolkit for rapid design exploration of augmented reality experiences. *Symposium on User Interface Software and Technology*.

[78] MacIntyre, B., Hill, A., Rouzati, H., Gandy, M., and Davidson, B. (2011). The Argon AR Web Browser and standards-based AR application environment. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 65–74. IEEE.

[79] Macintyre, B., Lohse, M., Bolter, J. D., and Moreno, E. (2001). Ghosts in the Machine : Integrating 2D Video Actors into a 3D AR System Georgia Institute of Technology. In *2nd International Symposium on Mixed Reality*.

[80] MacIntyre, B., Lohse, M., Bolter, J. D., and Moreno, E. (2002). Integrating 2-D video actors into 3-D augmented-reality systems. *Presence Teleoperators*, pages 189–202.

[81] Magnusson, C., Molina, M., Rassmus-Gröhn, K., and Szymczak, D. (2010). Pointing for non-visual orientation and navigation. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10*, page 735, New York, New York, USA. ACM Press.

[82] Makri, A., Arsenijevic, D., Weidenhausen, J., Eschler, P., Stricker, D., Machui, O., Fernandes, C., Maria, S., Voss, G., and Ioannidis, N. (2005). ULTRA: An Augmented Reality System for Handheld Platforms, Targeting Industrial Maintenance Applications. In *Proceedings of 11 th International Conference on Virtual Systems and Multimedia (VSMM'05)*.

[83] McGookin, D. and Brewster, S. (2011). PULSE. In *Proceedings of Interacting with Sound Workshop on Exploring Context-Aware, Local and Social Audio Applications - IwS '11*, pages 12–15, New York, New York, USA. ACM Press.

[84] Mcgookin, D., Brewster, S., and Priego, P. (2009). Audio Bubbles: Employing Non-speech Audio to Support Tourist Wayfinding. In Altinsoy, M. E., Jekosch, U., and Brewster, S., editors, *Proceedings of the 4th International Conference on Haptic and Audio Interaction Design*, volume 5763 of *Lecture Notes in Computer Science*, pages 41–50, Berlin, Heidelberg. Springer Berlin Heidelberg.

[85] Michael, R. and Beat, G. (265). Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects. In *Advances in Pervasive Computing*, page 2004.

[86] Microsoft Corporation (2011). Location Based Services Usage and Perceptions Survey Presentation URL: http://www.microsoft.com/download/en/details.aspx?id=3250.

[87] Milgram, P. and Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays. *IEICE Transations on Information and Systems*, E77-D(12):1321–1329.

[88] Mohring, M., Lessig, C., and Bimber, O. (2004). Video See-Through AR on Consumer Cell-Phones. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 252–253. IEEE.

[89] Mooser, J., You, S., and Neumann, U. (2007). Real-Time Object Tracking for Augmented Reality Combining Graph Cuts and Optical Flow. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–8. IEEE.

[90] Mulloni, A., Dünser, A., and Schmalstieg, D. (2010). Zooming interfaces for augmented reality browsers. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*, page 161, New York, New York, USA. ACM Press.

[91] Mustonen, T., Olkkonen, M., and Hakkinen, J. (2004). Examining mobile phone text legibility while walking. In *CHI '04 extended abstracts on Human factors in computing systems*, CHI EA '04, pages 1243–1246, New York, NY, USA. ACM.

[92] Mynatt, E. D., Back, M., Want, R., and Frederick, R. (1997). Audio aura. In *Proceedings of the 10th annual ACM symposium on User interface software and technology - UIST '97*, pages 211–212, New York, New York, USA. ACM Press.

[93] Naaman, M., Paepcke, A., and Garcia-Molina, H. (2003). From where to what: Metadata sharing for digital photographs with geographic coordinates. In *11th International Conference on Cooperative Information Systems (COOPIS 2003)*.

[94] Newcombe, R. A. and Davison, A. J. (2010). Live dense reconstruction with a single moving camera. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1505. IEEE Comput. Soc.

[95] Newman, J., Ingram, D., and Hopper, A. (2001). Augmented reality in a wide area sentient environment. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 77–86. IEEE Comput. Soc.

[96] Nillius, P. and Eklundh, J.-O. (2002). Fast block matching with normalized cross-correlation using walsh transforms. In *Computational Vision and Active Perception Laboratory (CVAP), KTH, Tech. Rep.*

[97] Nojima, T., Inami, M., Kawabuchi, Y., Maeda, T., Mabuchi, K., and Tachi, S. (2001). An Interface for Touching the Interface. In *ACM SIGGRAPH 2000 Conference Abstracts and Applications*, page 125.

[98] Olsson, T. and Salo, M. (2011). Online user survey on current mobile augmented reality applications. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 75 –84.

[99] O'Reilly, T. (2005). What is Web 2.0? Design patterns and business models for the next generation of software. pages 1 –49.

[100] Pan, Q., Reitmayr, G., Rosten, E., and Drummond, T. (2010). Rapid 3D modelling from live video. In *Proceedings of the 33rd International Convention MIPRO*, pages 252 –257.

[101] Paterson, N., Naliuka, K., Jensen, S. K., Carrigy, T., Haahr, M., and Conway, F. (2010). Design, implementation and evaluation of audio for a location aware augmented reality game. In *Proceedings of the 3rd International Conference on Fun and Games - Fun and Games '10*, pages 149–156, New York, New York, USA. ACM Press.

[102] Piekarski, W. and Thomas, B. (2001). Tinmith-Metro: new outdoor techniques for creating city models with an augmented reality wearable computer. In *Proceedings Fifth International Symposium on Wearable Computers*, pages 31–38. IEEE Comput. Soc.

[103] Piekarski, W. and Thomas, B. H. (2002). The Tinmith system: demonstrating new techniques for mobile augmented reality modelling. *Australian Computer Science Communications*, 24(4):61–61–70–70.

[104] Piekarski, W. and Thomas, B. H. (2003). Augmented reality user interfaces and techniques for outdoor modelling. In *Proceedings of the 2003 symposium on Interactive 3D graphics - SI3D '03*, page 225, New York, New York, USA. ACM Press.

[105] Prince, S., Cheok, A. D., Farbiz, F., Williamson, T., Johnson, N., Billinghurst, M., and Kato, H. (2002). 3D Live: Real Time Captured Content for Mixed Reality. In *ISMAR '02 Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, page 7.

[106] Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C. (2005). iLamps: Geometrically Aware and Self-Configuring Projectors. In *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*, page 5, New York, New York, USA. ACM Press.

[107] Regenbrecht, H. and Specht, R. (2000). A mobile Passive Augmented Reality Device - mPARD. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, pages 81–84. IEEE.

[108] Reitmayr, G. and Drummond, T. (2006). Going out: robust model-based tracking for outdoor augmented reality. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 109–118. IEEE.

[109] Reitmayr, G., Eade, E., and Drummond, T. W. (2007). Semi-automatic Annotations in Unknown Environments. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4. IEEE.

[110] Reitmayr, G. and Schmalstieg, D. (2001). Mobile collaborative augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 114–123. IEEE Comput. Soc.

[111] Rekimoto, J. (2008). Matrix: a realtime object identification and registration method for augmented reality. In *Proceedings. 3rd Asia Pacific Computer Human Interaction (Cat. No.98EX110)*, pages 63–68. IEEE Comput. Soc.

[112] Rekimoto, J., Ayatsuka, Y., and Hayashi, K. (1998). Augment-able Reality: Situated Communication through Physical and Digital Spaces. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computer*, page 68.

[113] Rekimoto, J. and Nagao, K. (1995). The World through the Computer: Computer Augmented Interaction with Real World Environments. In *Proceedings of the 8th annual ACM symposium on User interface and software technology - UIST '95*, pages 29–36, New York, New York, USA. ACM Press.

[114] Ribo, M., Lang, P., Ganster, H., Brandner, M., Stock, C., and Pinz, A. (2002). Hybrid tracking for outdoor augmented reality applications. *IEEE Computer Graphics and Applications*, 22(6):54–63.

[115] Ritzer, G. and Jurgenson, N. (2010). Production, consumption, prosumption: The nature of capitalism in the age of the digital 'prosumer'. *Journal of Consumer Culture*, 10(1):13–36.

[116] Rosten, E. and Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. In *In Proceedings of ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443, Berlin/Heidelberg. Springer-Verlag.

[117] Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-time video annotations for augmented reality. In Bebis, G., Boyle, R., Koracin, D., and Parvin, B., editors, *Advances in Visual Computing*, volume 3804 of *Lecture Notes in Computer Science*, pages 294–302. Springer Berlin Heidelberg.

[118] Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309.

[119] Rozier, J., Karahalios, K., and Donath, J. (2000). HearThere: An Augmented Reality System of Linked Audio. pages 63–67. in ICAD '00.

[120] Satoh, K., Anabuki, M., Yamamoto, H., and Tamura, H. (2001). A hybrid registration method for outdoor augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 67–76. IEEE Comput. Soc.

[121] Schall, G., Mulloni, A., and Reitmayr, G. (2010). North-centred orientation tracking on mobile phones. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 267–268. IEEE.

[122] Schall, G., Wagner, D., Reitmayr, G., Taichmann, E., Wieser, M., Schmalstieg, D., and Hofmann-Wellenhof, B. (2009). Global pose estimation using multi-sensor fusion for outdoor Augmented Reality. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 153–162. IEEE.

[123] Schildbach, B. and Rukzio, E. (2010). Investigating selection and reading performance on a mobile phone while walking. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, MobileHCI '10, pages 93–102, New York, NY, USA. ACM.

[124] Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavári, Z., Encarnação, L. M., Gervautz, M., and Purgathofer, W. (2002). The Studierstube Augmented Reality Project. *Presence: Teleoperators and Virtual Environments*, 11(1):33–54.

[125] Schmalstieg, D., Langlotz, T., and Billinghurst, M. (2010). Augmented Reality 2.0. In Sabine, C., Brunnett, G., and Welch, G., editors, *Virtual Realities, Dagstuhl seminar series*, pages 13–38.

[126] Schmalstieg, D. and Wagner, D. (2008). Mobile Phones as a Platform for Augmented Reality. In *Proceedings of the IEEE VR 2008 Workshop on Software Engineering and Architectures for Realtime Interactive Systems*, pages 43–44.

[127] Settel, Z., Wozniewski, M., Bouillot, N., and Cooperstock, J. R. (2009). Audio graffiti: A location based audio-tagging and remixing environment. In *International Computer Music Conference (ICMC)*, Montreal. 4 pages.

[128] Sharma, A. and Elidrisi, M. (2008). Classification of multimedia content (videos on YouTube) using tags and focal points. Technical report.

[129] Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, pages 593–600. IEEE Comput. Soc. Press.

[130] Simon, G., Henri, U., and Inria, P. (2010). In-Situ 3D Sketching Using a Video Camera as an Interaction and Tracking Device. In *Proceeding of Eurograhics 2010*.

[131] Simon Julier Yohan, S. J. (2000). BARS: Battlefield Augmented Reality System. In *NATO Symposium on Information Processing Techniques for Military Systems*, pages 9–11.

[132] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3):835.

[133] Spohrer, J. C. (1999). Information in places. *IBM SYSTEMS JOURNAL*, 38(4):602–628.

[134] Stahl, C. (2007). The roaring navigator. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services - MobileHCI '07*, pages 383–386, New York, New York, USA. ACM Press.

[135] Starner, T., Mann, S., Rhodes, B., Levine, J., Healey, J., Kirsch, D., Picard, R. W., and Pentland, A. (1997). Augmented reality through wearable computing. *Presence Teleoperators and Virtual Environments*, 6(4):386–398.

[136] Sundareswaran, V., Wang, K., Chen, S., Behringer, R., McGee, J., Tam, C., and Zahorik, P. (2003). 3D Audio Augmented Reality: Implementation and Experiments. page 296.

[137] Sutherland, I. E. (1968). A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I)*, page 757, New York, New York, USA. ACM Press.

[138] Tano, S., Takayama, T., Iwata, M., and Hashiyama, T. (2006). Multimedia Informal Communication by Wearable Computer based on Real-World Context and Graffiti. In *2006 IEEE International Conference on Multimedia and Expo*, pages 649–652. IEEE.

[139] Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W. (2000). ARQuake: an outdoor/indoor augmented reality first person application. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 139–146. IEEE Comput. Soc.

[140] Thomas, B., Demczuk, V., Piekarski, W., Hepworth, D., and Gunther, B. (1998). A wearable computer system with augmented reality to support terrestrial navigation. In *Proceedings of the Second International Symposium on Wearable Computers*, pages 168–171. IEEE Comput. Soc.

[141] Toffler, A. (1980). *The Third Wave*. Bantam Books.

[142] van den Hengel, A., Dick, A., Thormählen, T., Ward, B., and Torr, P. H. S. (2007). VideoTrace: rapid interactive scene modelling from video. *International Conference on Computer Graphics and Interactive Techniques*, 26(3).

[143] van den Hengel, A., Hill, R., Ward, B., and Dick, A. (2009). In situ image-based modeling. In *Symposium on Mixed and Augmented Reality*, pages 107–110.

[144] Vazquez-alvarez, A., Oakley, I., and Brewster, S. (2010). Urban sound gardens: Supporting overlapping audio landmarks in exploratory environments.

[145] Vazquez-Alvarez, Y. (2010). Designing spatial audio interfaces for mobile devices. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*, page 481, New York, New York, USA. ACM Press.

[146] Vazquez-Alvarez, Y., Oakley, I., and Brewster, S. A. (2011). Auditory display design for exploration in mobile audio-augmented reality. *Personal and Ubiquitous Computing*, 16(8):987–999.

[147] Ville, P. and Karjalainen, M. (2001). Localization of Amplitude-Panned Virtual Sources I: Stereophonic Panning. *Journal of the Audio Engineering Society*, 49:739–752.

[148] Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Almeida, L., Stricker, D., Gleue, T., Christou, I. T., and Carlucci, R. (2001). Archeoguide. In *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage - VAST '01*, page 131, New York, New York, USA. ACM Press.

[149] Wagner, D. (2007). *Handheld Augmented Reality*. PhD thesis, Graz University of Technology.

[150] Wagner, D., Langlotz, T., and Schmalstieg, D. (2008a). Robust and unobtrusive marker tracking on mobile phones. In *Proceedings of the 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality 2008*, pages 121–124. IEEE Computer Society.

[151] Wagner, D., Langlotz, T., and Schmalstieg, D. (2008b). Technical Report on Robust and Unobtrusive Marker Tracking on Mobile Phones. Technical report, Graz University of Technology.

[152] Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. (2010). Real-time panoramic mapping and tracking on mobile phones. In *2010 IEEE Virtual Reality Conference (VR)*, pages 211–218. Ieee.

[153] Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2008c). Pose tracking from natural features on mobile phones. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 125–134.

[154] Wagner, D. and Schmalstieg, D. (2003). First Steps Towards Handheld Augmented Reality. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 127.

[155] Wagner, D. and Schmalstieg, D. (2007). ARToolKitPlus for Pose Tracking on Mobile Devices. In *Proceedings of 12th Computer Vision Winter Workshop*.

[156] Wither, J., Coffin, C., Ventura, J., and Hollerer, T. (2008). Fast annotation and modeling with a single-point laser range finder. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 65–68. IEEE.

[157] Wither, J., DiVerdi, S., and Hollerer, T. (2006). Using aerial photographs for improved mobile AR annotation. In *IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006*, pages 159–162.

[158] Wither, J., DiVerdi, S., and Höllerer, T. (2009). Annotation in outdoor augmented reality. *Computers & Graphics*, 33(6):679–689.

[159] Wither, J. and Hollerer, T. (2005). Pictorial Depth Cues for Outdoor Augmented Realityn. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 92–99.

[160] Wither, J., Tsai, Y.-T., and Azuma, R. (2011). Indirect augmented reality. *Computers & Graphics*, 35(4):810–822.

[161] Woo, D., Mariette, N., Salter, J., Rizos, C., and Helyer, N. (2006). Audio Nomad. In *Proceedings ION GNSS 2006*.

[162] You, S., Neumann, U., and Azuma, R. (1999). Orientation tracking for outdoor augmented reality registration. *IEEE Computer Graphics and Applications*, 19(6):36–42.

[163] Zandbergen, P. A. and Barbeau, S. J. (2011). Positional Accuracy of Assisted GPS Data from High-Sensitivity GPS-enabled Mobile Phones. *Journal of Navigation*, 64(03):381–399.