

Robust Wide-Area Tracking and Intuitive 3D Interaction for Mixed Reality Environments

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor/in der technischen Wissenschaften

by

Annette Mossel

Registration Number 0727827

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Priv.-Doz. Dr. techn. Hannes Kaufmann

The dissertation has been reviewed by:

(Priv.-Doz. Hannes Kaufmann)

(Prof. Mark Billinghurst)

Wien, September 4, 2014

(Annette Mossel)

Erklärung zur Verfassung der Arbeit

Annette Mossel
Lindengasse 41/9, 1070 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasserin)

Acknowledgements

This PhD thesis would not have been possible without the support of my family, my beloved friends near and far, my advisors and my colleagues from the IMS.

First of all, I would like to thank my advisor Priv.-Doz. Dr. Hannes Kaufmann for introducing me to the research field, for providing me with scientific insights into mixed reality, for all the valuable discussions about ideas and projects and for all his help, feedback and support from the beginning of my time at the IMS. Furthermore, I thank Prof. Mark Billinghurst for being my second reviewer and for his valuable feedback for this thesis. I also would like to thank Prof. Christian Breiteneder for giving me the opportunity to work at the IMS, for valuable discussions and for supporting my plans and ideas.

I would like to thank my colleagues who were involved in my research for their feedback and support. In particular, I thank Georg Gerstweiler and Emanuel Vonach for their strong contributions to the results of Part II, my graduate student Benjamin Venditti whose work significantly contributed to the results presented in Part III, Christian Schönauer, Georg Gerstweiler, Michael Bressler, David Zeller and Mathis Csisinko for their contributions to Part IV, Dr. Matthias Zeppelzauer for valuable research discussions, test assistance as well as proof reading and Dr. Dalibor Mitrovic for providing me with scientific and non-scientific (sweets and cookies) support. I especially thank Ingrid Lissa for taking a lot of administrative work out of my hands and all of my colleagues and tutors, who helped me in holding my lectures. Furthermore, I would like to thank Dr. Klaus Chmelina from Geodata Ziviltechniker GesmbH for the good cooperation during our joint research projects. Financial support for this work was obtained by FFG - Austrian Research Promotion Agency with project no: 822680 (B1) as well as by VrVis Research GmbH (Vienna, Austria) within the EU 7th Framework project *I² Mine* (no: NMP2-LA-2011-280855).

Besides my colleagues, I wish to express my most heartfelt thanks to my wonderful flatmates for providing me a cozy and great place to live, especially Saskia Kuhlmann for making Vienna a new home for me and Julia Stockenreiter for all her advise and encouragement at any time. I am deeply grateful to my parents, my sister, my brother in law and my complete family for their love, endorsement and support of my ideas and plans throughout my entire life. Finally, I would like to thank Matthias for being such a wonderful partner, friend and colleague in one person. I really appreciate sharing my life with you.

Abstract

Mixed reality has been a focus of research for many years and has recently gained particular importance with the emergence of powerful, low-cost input- and output devices as well as processing platforms that foster the applicability of virtual simulations for everyday usage. However, this leads to significant challenges since the creation of compelling mixed reality environments requires knowledge and robust techniques in the areas tracking, visualization, interaction and in the non-obligatory areas distribution and authoring.

This thesis focuses on the development of novel techniques and algorithms to contribute to the solution of fundamental problems in the areas of tracking, interaction, and application development of mixed reality systems. Firstly, a novel system for wide-area optical tracking in unconstrained indoor environments is presented that is capable of stereo camera calibration and model-based tracking of rigid-body targets in environments with poor illumination, static and moving ambient light sources, occlusions and harsh conditions, such as fog. The experimental results demonstrate the system's capabilities to track targets up to $90m$ and its applicability to act as a mixed reality tracking system as well as a general purpose measurement tool for future (underground) surveying tasks, such as autonomous machine guidance. Secondly, we investigated concepts for intuitive 3D interaction in virtual environments, specifically in one-handed handheld mixed reality. To address the shortcomings of state-of-the-art 3D selection and manipulation techniques, the novel algorithms *DrillSample* for selection, and *3DTouch* and *HOMER-S* for manipulation are proposed. All three approaches aim at reducing the necessary input through the user's fingers to provide easy to understand and straightforward interaction. Therefore, they incorporate the 6-degree-of-freedom pose that is obtained through optical tracking, resulting in a one-finger interaction for precise selection of partly or fully occluded objects with high visual similarity. Thirdly, the novel software framework *ARTIFICe* is presented that facilitates the development of compelling mixed reality environments. It aims at minimizing the initial hurdles of application development as it is inexpensive and provides a powerful graphical interface to easily access and author tracking, interaction, visualization and distribution.

With the presented contribution, we aim at leveraging the applicability of mixed reality into unconstrained everyday environments that are used by non-experts.

Kurzfassung

Mischrealitäten, als durch den Computer simulierte dreidimensionale Umgebungen, sind seit vielen Jahren Gegenstand der Forschung. In jüngster Zeit hat das Aufkommen von leistungsfähigen und kostengünstigen Recheneinheiten sowie Ein- und Ausgabegeräten zu gesteigerten Bemühungen geführt, Mischrealitäten verstärkt in Alltagssituationen einzusetzen. Dies jedoch führt zu einer Vielzahl von Herausforderungen, da für deren Entwicklung robuste Techniken und Kenntnisse in Lokalisation, Visualisierung, Interaktion und optional Verteilung erforderlich sind.

Der wissenschaftliche Beitrag dieser Dissertation umfasst neue Techniken und Algorithmen für Lokalisation, Interaktion und Anwendungsentwicklung von Mischrealitäten. Im ersten Teil dieser Arbeit wird ein neues Lokalisierungssystem vorgestellt, das in Innenräumen auf Distanzen von bis zu $90m$ die 3D Position von mit visuellen Markierungspunkten ausgestatteten Objekten bestimmen kann. Das System ist hierbei sowohl während Kamerakalibrierung wie auch Lokalisierung robust gegenüber visuellen Störeinflüssen der Umgebung, wie beispielsweise statischen und bewegten Lichtquellen sowie Verdeckungen. Der zweite Teil dieser Arbeit beschäftigt sich mit Mensch-Maschine-Interaktion in dreidimensionalen Systemen, speziell in mobilen Mischrealitäten mit berührungssensitiven Bildschirmen. Um die Schwächen bestehender 3D Interaktionstechniken zu beheben, werden die neuen Algorithmen *DrillSample* für Objektselektion, sowie *3DTouch* und *Homer-S* für Objektmanipulation vorgestellt. Die drei Techniken zielen alle auf leicht verständlich und einfach zu bedienende Interaktionen ab, indem sie notwendige BenutzerInneneingaben reduzieren und die Position sowie Orientierung des mobilen Endgeräts miteinbeziehen. Dadurch können mit lediglich einem Finger präzise teilweise oder gänzlich verdeckte Objekte ausgewählt werden, und entweder ohne Finger oder mit einem bzw. zwei Fingern Objekte verschoben, gedreht und skaliert werden. Im dritten Teil dieser Arbeit wird das neue Softwareframework *ARTIFICe* vorgestellt, das die Erkenntnisse der ersten beiden Teile einbezieht und der einfachen Erstellung von hochwertigen Mischrealitäten dient. Es stellt der BenutzerIn hierfür eine übersichtliche Benutzeroberfläche zur Verfügung, über die man auf Techniken und Hardwareschnittstellen für Lokalisation, Interaktion, Visualisierung und Verteilung zugreifen kann.

Der vorgestellte wissenschaftliche Beitrag zielt darauf ab, die Erstellung und Bedienung von Mischrealitäten im Alltag zu vereinfachen und zu fördern.

Contents

I	Introduction	1
1	Introduction to Mixed Reality	3
2	Motivation & Contribution	5
2.1	Resulting Publications	8
2.1.1	Peer Reviewed	8
2.1.2	Technical Reports	9
3	Thesis Organization	11
II	Wide-Area Optical Tracking	13
1	Introduction	15
1.1	Motivation & Problem Statement	16
1.2	Research Objective	16
1.3	Organization	16
2	Theoretical Foundations	19
2.1	Principles of Optical Tracking	19
2.1.1	Accuracy & Performance	19
2.1.1.1	Performance Measures	19
2.1.1.2	Sources of Error	20
2.2	Tracking Pipeline	21
2.2.1	Feature Segmentation	22
2.2.1.1	Natural Features	22
2.2.1.2	Artificial Features	23
2.2.2	Model Fitting	24
2.2.2.1	2D Domain	25
2.2.2.2	3D Domain	28
2.2.3	Pose Estimation	29
2.3	Projective Geometry	30
2.3.1	The Pinhole Camera Model	31
2.3.2	Camera Model Extensions	32

CONTENTS

2.3.2.1	Principal Point Offset	32
2.3.2.2	Skew Parameter	33
2.3.2.3	Camera Lens Distortions	33
2.3.2.4	Camera Rotation & Translation	35
2.3.2.5	Intrinsic & Extrinsic Camera Parameters	36
2.3.3	Multiple-View Geometry	36
2.3.3.1	Epipolar Geometry	36
2.3.3.2	Stereo Correspondence Problem	38
2.3.3.3	Computing the Camera Projection Matrix	38
2.3.3.4	3D Point Reconstruction	40
2.3.4	Camera Calibration	40
2.4	Summary	42
3	Related Work	43
3.1	Radio Frequency & Ultra Sound	44
3.2	Optical Tracking	45
3.3	Laser Measurement Systems	47
4	Methodology	51
4.1	System Requirements	52
4.2	Evaluation of Target Visibility	52
4.2.1	Test Setup	53
4.2.2	Test Results	53
4.3	Methodological Approach	54
4.3.1	Vision System	54
4.3.2	Target Design Guidelines	55
4.3.3	Calibration	57
4.3.3.1	Intrinsic Calibration	57
4.3.3.2	Extrinsic Calibration	58
4.3.4	Interference Filtering	61
4.3.4.1	Hardware-based Target Identification	62
4.3.4.2	Software-based Target Identification	65
4.3.5	3 Degree-Of-Freedom Tracking	67
4.3.6	Occlusion Recovery	69
4.4	System Development	69
4.4.1	Hardware	69
4.4.2	Software	70
4.4.3	System Costs	74
5	Experimental Results	75
5.1	Test Platform	75
5.2	Test Cases & Performance Measures	75
5.2.1	Calibration Performance	76
5.2.2	Tracking Performance	76

5.3	Tracking for Mixed Reality	77
5.3.1	Target Design	77
5.3.1.1	Prototype	78
5.3.2	Test Environment	79
5.3.3	Model Training	79
5.3.4	Camera Calibration	80
5.3.5	3D Position Accuracy	81
5.3.6	3D Position Stability	82
5.3.7	Tracking Performance	82
5.4	Hand-held Target Tracking for Tunneling	82
5.4.1	Target Design	83
5.4.1.1	Tracking Scenarios	84
5.4.2	System Prototype	84
5.4.3	Test Environment	86
5.4.4	Model Training	86
5.4.5	Camera Calibration	86
5.4.6	Accuracy & Stability of 3D Position Estimation	88
5.4.7	Tracking Performance	90
5.5	Machine Tracking for Underground Guidance	91
5.5.1	Shortcoming of Existing Technology	92
5.5.2	Test Environment	92
5.5.3	Target Design	93
5.5.3.1	Evaluation of LED Range	93
5.5.3.2	Target Prototype	94
5.5.4	Model Training	95
5.5.5	Camera Calibration	95
5.5.6	Accuracy & Stability of 3D Position Estimation	96
5.5.6.1	Influence of Vibrations	97
5.5.7	Tracking Performance for Machine Guidance	98
5.5.7.1	Tracking under normal Visibility	98
5.5.7.2	Tracking with Occlusions and Poor Visibility	99
5.6	Conclusion	101
6	Summary	105
 III User Interfaces for 3D Interaction		107
1	Introduction	109
1.1	Motivation & Problem Statement	110
1.2	Research Objective	111
1.3	Organization	111
2	Theoretical Foundations & Related Work	113

CONTENTS

2.1	User Interfaces in Mixed Reality	113
2.1.1	3D Interaction	114
2.1.1.1	3D Selection and Manipulation Tasks	114
2.1.1.2	3D Selection & Manipulation Metaphors	115
2.1.2	3D Selection & Manipulation in Handheld Mixed Reality	115
2.2	3D Object Selection	116
2.2.1	Virtual Hand Metaphors	117
2.2.2	Virtual Pointing Techniques	117
2.2.2.1	One-Step Selection Techniques	118
2.2.2.2	Two-Step Selection Techniques	119
2.3	3D Object Manipulation	121
2.3.1	For Immersive Environments	121
2.3.2	For 2D Multi-Touch Devices	122
2.4	Summary	124
3	3D Selection in Handheld Mixed Reality	125
3.1	Requirements	126
3.2	Design Guidelines	126
3.3	The DrillSample Technique	127
3.3.1	Selection Design	128
3.3.2	Mobile Raycasting	129
3.3.3	Algorithm	130
3.3.4	Crucial Aspects of the Algorithm	130
3.3.4.1	Length of the DrillSample Ray	132
3.3.4.2	Z-Position of the DrillSample	132
3.4	Performance Studies	133
3.4.1	Baseline Techniques	134
3.4.2	Adaptions for Handheld Mixed Reality	134
3.4.3	Objectives	135
3.4.4	Experimental Design and Procedure	135
3.4.5	Implementation	137
3.4.6	Test Scenarios	137
3.5	Experimental Results	138
3.5.1	Quantitative Evaluation	139
3.5.1.1	Performance Evaluation	139
3.5.2	Subjective Evaluation	141
3.5.3	Qualitative Evaluation	142
3.6	Discussion	143
3.6.1	Variations of the Algorithm	145
4	3D Manipulation in Handheld Mixed Reality	147
4.1	Methodological Approach	148
4.1.1	Requirements & Prerequisites	148
4.1.2	Design Guidelines	148

4.1.3	The 3D Touch Technique	149
4.1.3.1	Translation	149
4.1.3.2	Rotation	150
4.1.3.3	Scaling	151
4.1.4	The HOMER-S Technique	151
4.1.4.1	6DOF Manipulations	151
4.1.4.2	Scaling	152
4.1.5	Assistance Design	153
4.1.5.1	Mode Switches	153
4.1.5.2	Supporting Visualization	154
4.1.6	Crucial Aspects	155
4.2	Performance Studies	155
4.2.1	Prerequisites	155
4.2.2	Objectives	156
4.2.3	Experimental Design and Procedure	157
4.2.4	Subjects & Apparatus	158
4.2.5	Test Scenarios	158
4.2.5.1	Positioning on a Plane	158
4.2.5.2	Positioning in 3D Space	159
4.2.5.3	Positioning & Rotation in 3D Space	159
4.2.5.4	Non-Uniform Scaling & Positioning in 3D Space	159
4.3	Experimental Results	160
4.3.1	Quantitative Evaluation	160
4.3.1.1	Performance Evaluation	160
4.3.2	Subjective Evaluation	162
4.4	Discussion	164
5	Summary	167
IV	Creating Mixed Reality Environments	169
1	Introduction	171
1.1	Motivation	172
1.2	Organization	172
2	Background & Related Work	173
2.1	Key Elements of a Mixed Reality Framework	173
2.2	Application Development & Scene Management	174
3	Framework Architecture	177
3.1	Base Infrastructure	178
3.1.1	Functionalities of Unity	178
3.1.2	Core Concepts of Unity	178

CONTENTS

3.2	Middleware	179
3.2.1	OpenTracker	179
3.2.2	Vuforia	180
3.2.3	Supported Setups & Hardware	181
3.2.3.1	Desktop Mixed Reality	181
3.2.4	(Semi) Immersive Mixed Reality	182
3.2.4.1	Handheld Mixed Reality	182
3.3	Application Layer	183
3.3.1	The ARTiFICe Manager	183
3.3.1.1	Tracking Module	184
3.3.1.2	Interaction Module	184
3.3.1.3	Collaboration & Distribution	185
3.4	Workflow for Application Development	187
4	Developed Mixed Reality Environments	189
4.1	Test Setups & Environment	189
4.2	Non-Immersive Mixed Reality	190
4.2.1	Single & Multi-User Desktop Mixed Reality	190
4.2.2	Multi-User Handheld Mixed Reality	191
4.3	Combined Non- & Semi-Immersive Mixed Reality	192
4.4	Combined Semi- & Full Immersive Mixed Reality	192
5	Summary	195
V	Conclusion	197
1	Findings & Outlook	199
1.1	Wide-Area Optical Tracking	200
1.1.1	Open Topics	201
1.2	3D Interaction	202
1.2.1	Open Topics	203
1.3	Creating Mixed Reality Environments	204
1.3.1	Open Topics	204
VI	Appendix	207
	Bibliography	209
	List of Figures	223
	List of Tables	227
A	User Studies	229

PART I

Introduction

1	Introduction to Mixed Reality	3
2	Motivation & Contribution	5
3	Thesis Organization	11

Chapter 1

Introduction to Mixed Reality

The generation of computer simulated environments that combine virtual and real content has been a focus of research for many years. It is now gaining particular importance with the emerge of powerful, low-cost input- and output devices as well as processing platforms that foster the applicability of virtual simulations for everyday usage. Typical application domains for such systems are training, therapy, education and entertainment [98]. The combination of real and virtual content is referred to as *Mixed Reality* and can be defined as a computer generated 3D simulation with different levels of blending of real and virtual scene objects. These levels are described by the *Milgram Continuum* [19] that encompasses all possible variations and compositions of real and virtual objects, as depicted in Figure 1.1.



Figure 1.1: The Milgram continuum describing the variations of mixed reality.

While *Reality* shows a real environment where no augmentation with virtual objects occurs, the observed environment in *Augmented Reality* mostly consists of real objects that are augmented with a few virtual objects. *Augmented Virtuality* consists of mostly virtual objects that are augmented with few real objects while *Virtual Reality* completely locks out the real world and only displays virtual objects in the observed environment. Each state of the Milgram continuum can be further categorized depending on the provided amount of immersion that correlates with the involved input and output devices. A *Non-Immersive* system mostly consists of a non-stereoscopic screen and 2D discrete user interfaces, such as mouse and keyboard. The user views the virtual scene through the output device, that acts as a window into the virtual world. Thereby, the user is fully aware of the reality that surrounds him. Examples are desktop setups that provide a stationary view into the virtual scene, and handheld mixed reality that allows the user to change the viewpoint by moving the mobile device. *Semi-Immersive* systems provide

1. INTRODUCTION TO MIXED REALITY

an increased amount of immersion by enabling stereoscopic viewing and 3D interaction. This is usually achieved by employing stereo projection walls that are viewed through tracked shutter glasses. The user typically can freely walk in front of the wall and the involved interaction devices allow for interaction in 3D. Although the user cannot fully immerse into the mixed reality environment, as it does not entirely surround him, the amount of immersion is increased by stereoscopic viewing, natural walking and 3D object interaction. *Fully Immersive* setups incorporate head mounted displays as well as portable 3D interaction devices, enabling the user to freely move throughout the entire tracking space. For a in depth review of the different flavors of mixed reality, the reader is kindly referred to [83].

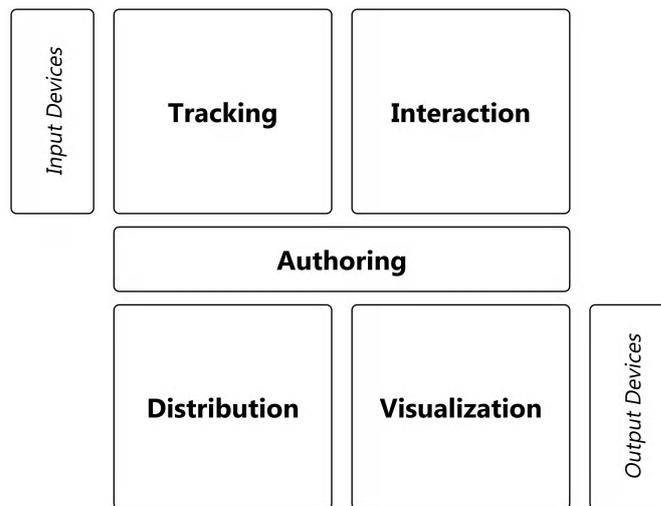


Figure 1.2: Components of a mixed reality system.

The creation of compelling mixed reality environments is built upon the mandatory key components tracking, visualization, interaction and the non-obligatory module distribution. *Tracking* of users as well as of interaction devices is necessary to allow egocentric scene view and to enable *Interaction* between the user and the virtual environment; *Visualization* is required to render the entire 3D scene on an output device, such as a screen, a projection wall or a head mounted display. In addition, *Distribution* of the scene objects and of the user’s interactions allows for a remote mixed reality setup engaging one or more users to view and interact collaboratively with the virtual simulation. To create, maintain and deploy the mixed reality application, an *Authoring* module that interfaces with the four mentioned components is a valuable asset, especially for non-experts. It provide means to manage the 3D scene and to set up the entire system before deployment. The components of a mixed reality system are illustrated in Figure 1.2.

This thesis focuses on the development of novel techniques to contribute to the solution of fundamental problems in the areas of tracking, interaction, and mixed reality application development.

Chapter 2

Motivation & Contribution

The major objective of this thesis is the development of novel techniques and systems to leverage the applicability of mixed reality into unconstrained everyday environments that are used by non-experts.

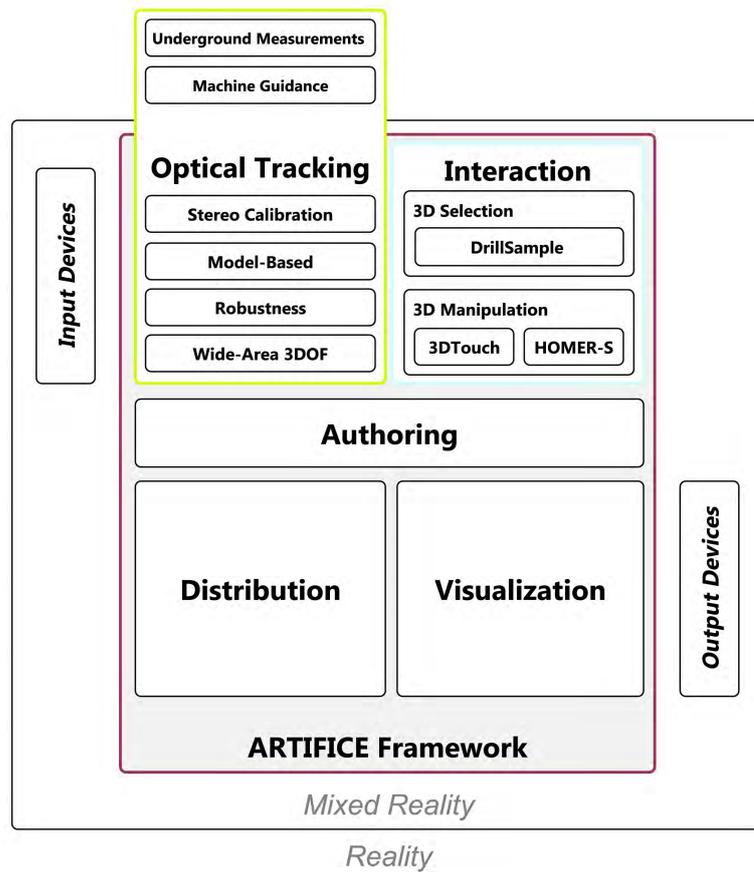


Figure 2.1: Investigated concepts, their relationship and the presented contribution.

2. MOTIVATION & CONTRIBUTION

For this purpose, we investigated concepts in the area of tracking, interaction and mixed reality application development, as depicted in Figure 2.1, that resulted in the following contributions.

1. A novel optical tracking system with enhanced robustness against environmental interferences and extended volume coverage that requires a minimal amount of vision hardware.
2. Novel techniques for 3D selection and manipulation that employ 3D position and orientation of the input device as well as incorporate real world metaphors to highly simplify necessary user input.
3. A novel software framework to develop collaborative and distributed mixed reality applications. It features a powerful graphical user interface for authoring and supports a large number of off-the-shelf input as well as output devices.

Tracking systems determine the position and orientation of an object in space, such as the user's head mounted display or an interaction device. A large number of different tracking technologies exist and each method has its advantages and disadvantages regarding volume coverage, tracking accuracy, sensitivity to interferences as well as scalability. Thus, there is no general tracking technology that perfectly suits all variations of tracking scenarios. Infrared optical tracking detects targets within camera images in the near infrared spectrum. This technology has been found to be fast, accurate as well as scalable to a certain extent, and is widely used to provide tracking in mixed reality applications. However, state-of-the-art systems suffer from sensitivity to ambient interfering lights during calibration and tracking, furthermore they only cover standard room sized environments with a small amount of vision hardware. This yields lack of tracking support for wide unconstrained indoor environments and results in high hardware costs and complex setup as well as maintenance routines when extending the tracking volume. Thereby, the system is impractical for everyday usage, especially for non-experts. To overcome these limitations, a system for model-based optical 3D position tracking of rigid-body targets is presented. The proposed system is capable to cover wide, unconstrained indoor volumes and provides robust calibration and tracking while requiring a minimal hardware setup of two cameras. The experimental results demonstrate the system's capabilities to act as a mixed reality tracking system as well as a general purpose measurement tool for future (underground) surveying tasks, such as autonomous machine guidance. It was successfully applied in three different unconstrained wide area indoor environments, providing relative millimeter point accuracy up to $30m$ and centimeter deviation up to $90m$. These results clearly improve state-of-the-art systems and reveal the system's applicability to use cases that go beyond mixed reality scenarios.

As described, tracking is a fundamental building block of a mixed reality system and is the technological foundation to enable interaction with a virtual 3D scene through the involved interaction devices. Therefore, it is applied to investigate novel techniques

to provide intuitive interaction between the user and the 3D simulation. Intuitive interaction can be defined as a mean that enables users to interact with a scene object using their real-world knowledge for selection and object manipulation. In a handheld mixed reality system, a user typically holds a portable device in one hand to view the scene onto the display that shows a live camera image that is augmented with virtual scene objects. Throughout this thesis, the handheld device is referred to a smartphone with a touch sensitive display to simultaneously detect multiple finger inputs. The user’s second hand interacts with the scene objects using the multi-touch input. However, two problems arise in such a situation: the imprecise finger touch input for selection yields the high probability of inaccurate extraction of small objects, especially when they are partly or fully occluded or surrounded by highly similar virtual scene objects. For object manipulation, such as translating, rotating and scaling, existing methods use complex multi-finger gestures to provide full 3D manipulations. However, most of these gestures are difficult or impossible to apply in a one-handed setup and their usage additionally requires prior knowledge. To address the shortcomings of state-of-the-art 3D selection and manipulation techniques, three novel methods are proposed and evaluated throughout user studies. Firstly, the 3D selection technique *DrillSample* is described that only requires single touch inputs. Upon selection of multiple objects, the user can indicate the desired object in a refinement step that presents the objects in their original spatial context. Thereby, it allows the user to precisely disambiguate between objects with high similarity in visual appearance and enables the selection of strongly or entirely occluded objects. For a comprehensive evaluation of the *DrillSample* selection technique, a summative evaluation was conducted by comparing *DrillSample* with two baseline techniques across three different selection scenarios based on variations of object density and visibility. As demonstrated by the study results, *DrillSample* overall outperforms the state-of-the-art baseline methods and was found the best general purpose selection method for visible as well as partly and fully occluded objects, independent of their visual appearance. To overcome shortcoming of state-of-the-art 3D manipulation techniques using 2D multi-touch input, two novel methods *3DTouch* and *HOMER-S* are proposed. Both support the spatial rigid manipulations translation, rotation and the non-rigid manipulation scaling. *3DTouch* provides 3D translation and rotation as well as non-uniform scaling by fusing one- or two-finger touch input with the handheld’s 6 degree-of-freedom (DOF) pose that is obtained using optical tracking. The integral 6DOF manipulation is decomposed into two separate tasks, enabling a single touch input to be sufficient to access all three 3DOF during translation and rotation. A two-finger pinch gesture allows for non-uniform scaling in 3D. *HOMER-S* provides interaction beyond the (limited) screen dimensions by decoupling the manipulation process from any touch input. It aims at DOF-integration and maps the 6DOF device pose onto the object upon selection. Thereby, full 6DOF manipulation as well as non-uniform scaling is performed by employing real-world metaphors. In a comprehensive user study, the performance, accuracy and ease-of-use for both techniques are assessed across four different test scenarios with varying manipulation tasks. The results reveal both techniques to be intuitive to translate and rotate objects. *HOMER-S* lacks accuracy compared to *3DTouch* but achieves a

2. MOTIVATION & CONTRIBUTION

significant performance increase in terms of speed for full 6DOF manipulations.

While tracking and interaction are two key components to develop a mixed reality simulation, a crucial factor to leverage mixed reality for everyday usage is quick application prototyping and development. Since creating a mixed reality application requires knowledge in all of the building blocks as depicted in Figure 1.2, a high entry threshold for development is the result. At the moment of investigating mixed reality frameworks, there were no inexpensive toolkits available that provided interfaces to extend the framework with novel techniques for tracking and interaction as well as that featured a powerful graphical authoring component. This technological gap fostered the development of a cost-efficient software framework *ARTIFICE* that enables quick prototyping of collaborative and distributed mixed reality environments. It features a loosely-coupled, modular software architecture that overcomes limitations of state-of-the-art frameworks regarding costs, usability and extensibility. *ARTIFICE* provides tracking data by several input devices and offers a number of built-in interaction methods, including the novel techniques of this thesis. It enables multi user collaboration in distributed virtual scenes and incorporates recently emerged, popular off-the-shelf input devices, such as Microsoft Kinect, Razer Hydra and mobile phones, running Android and iOS. The framework was employed for proof-of-concept application development to evaluate the investigated concepts of this thesis. Furthermore, *ARTIFICE* was used by more than 100 students during their university graduate program who were not familiar with mixed reality technology before. It allowed them to develop distributed applications within just a couple of weeks that incorporated different tracking devices and as well as interaction techniques. These results indicate that *ARTIFICE* can act as a foundation to further leverage the simplification of application development and thereby the pervasiveness of mixed reality.

2.1 Resulting Publications

The work presented in this thesis has appeared in the following publications:

2.1.1 Peer Reviewed

- [1] Annette Mossel, Christian Schönauer, Georg Gerstweiler, and Hannes Kaufmann. “ARTiFICE-Augmented Reality Framework for Distributed Collaboration”. In: *Presented at Workshop on Off-The-Shelf Virtual Reality, IEEE VR, USA, 2012, published in International Journal of Virtual Reality* 11.3 (2012), pp. 1–7.
- [2] Annette Mossel, Georg Gerstweiler, Emanuel Vonach, Klaus Chmelina, and Hannes Kaufmann. “Robust Long-Range Optical Tracking for Tunneling Measurement Tasks”. In: *European Geosciences Union - General Assembly 2013*. Vol. 15. Vienna, Austria: Geophysical Research Abstracts, 2013, p. 1.

- [3] Annette Mossel and Hannes Kaufmann. “Wide Area Optical User Tracking in Unconstrained Indoor Environments”. In: *Proceedings of the The 23rd International Conference on Artificial Reality and Telexistence (ICAT)*. Tokyo, Japan: IEEE, 2013, pp. 108–115.
- [4] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. “3DTouch & HOMER-S: Intuitive Manipulation for One-Handed Handheld AR”. In: *Proceedings of the Virtual Reality International Conference on Laval Virtual (VRIC '13)*. Laval, France: ACM Press, 2013, pp. 1–10. ISBN: 9781450318754.
- [5] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. “DrillSample: Precise Selection in Dense Handheld Augmented Reality Environments”. In: *Proceeding of 15th Int. Conf. of Virtual Technologies (VRIC'13)*. Vol. 00. Laval, France: ACM Press, 2013, p. 10. ISBN: 9781450318754.
- [6] Annette Mossel, Georg Gerstweiler, Emanuel Vonach, Klaus Chmelina, and Hannes Kaufmann. “Vision-based Long-Range 3D Tracking, applied for Underground Surveying Tasks”. In: *Journal of Applied Geodesy* 8.1 (2014), pp. 43–64.

2.1.2 Technical Reports

- [1] Annette Mossel, Thomas Pintaric, and Hannes Kaufmann. *Analyse der Machbarkeit und des Innovationspotentials der Anwendung der Technologie des Optical Real-Time Trackings für Aufgaben der Tunnelvortriebsvermessung*. Tech. rep. Austria: Institute of Software Technology and Interactive Systems, Vienna University of Technology, 2008.
- [2] Klaus Chmelina, Egmont Lammer, Annette Mossel, and Hannes Kaufmann. “Real-Time Machine Guidance with Tracking Cameras”. In: *Proceedings of Aachen International Mining Symposia (AIMS)*. Aachen, Germany, 2014.
- [3] Klaus Chmelina, Annette Mossel, and Hannes Kaufmann. “Echtzeitvermessung mit Infrarottrackingkameras - Untersuchung einer neuen Messtechnik für untertage”. In: *Proceedings of 17. Internationaler Ingenieurvermessungskurs*. Zürich, Switzerland: Herbert Wichmann-Verlag, Offenbach/Berlin, 2014.

Chapter 3

Thesis Organization

The organization of this thesis follows the identified key components of a mixed reality system, as shown in Figure 2.1. It presents the performed research in three parts.

Part II focuses on wide area optical tracking in unconstrained indoor environments. After reviewing the principles of optical tracking and multi-view imaging, competing state-of-the-art tracking systems are discussed and compared. The background chapters are followed by the methodological approach that describes the theoretical principles that were investigated and developed to build the proposed robust wide-area tracking system. The system's prototype is then evaluated in depth by testing it in three different use cases: 1) user tracking in a mixed reality setup, 2) handheld target tracking for tunneling application and 3) tracking for machine guidance in underground environments. Finally, a summary presents findings and concludes this part.

The work in optical tracking is followed by Part III that presents the investigated concepts and developed algorithms for 3D object selection and manipulation in a one-handed handheld mixed reality environment. In the first chapter of this part, theoretical foundations of 3D selection and manipulation are given and state-of-the-art techniques are reviewed and discussed. Next, the methodological approach of the novel selection technique is described and the results of the conducted user study are presented. After the study on object selection, two novel approaches for object manipulation are described. Both techniques are examined by a comparative user study and the results are statistically evaluated and discussed. Finally, conclusions of the novel techniques are given and the investigated concepts of this part are summarized.

Part IV presents a software framework that enables the development of collaborative multi-use distributed mixed reality applications by integrating different hardware devices, tracking technologies and interaction metaphors. After giving an overview of related work, the design approach of the proposed framework is described. Next, the capabilities of the framework were evaluated by developing example applications that support various input devices as well as encompass different setups of mixed reality, ranging from desktop, handheld, semi- to full immersive compositions of real and virtual objects.

Finally, in Part V the author summarizes the thesis and the presented contributions, and discusses open topics in the context of the investigated topics in mixed reality.

Wide-Area Optical Tracking

1	Introduction	15
1.1	Motivation & Problem Statement	16
1.2	Research Objective	16
1.3	Organization	16
2	Theoretical Foundations	19
2.1	Principles of Optical Tracking	19
2.2	Tracking Pipeline	21
2.3	Projective Geometry	30
2.4	Summary	42
3	Related Work	43
3.1	Radio Frequency & Ultra Sound	44
3.2	Optical Tracking	45
3.3	Laser Measurement Systems	47
4	Methodology	51
4.1	System Requirements	52
4.2	Evaluation of Target Visibility	52
4.3	Methodological Approach	54
4.4	System Development	69
5	Experimental Results	75
5.1	Test Platform	75
5.2	Test Cases & Performance Measures	75
5.3	Tracking for Mixed Reality	77
5.4	Hand-held Target Tracking for Tunneling	82
5.5	Machine Tracking for Underground Guidance	91
5.6	Conclusion	101
6	Summary	105

Chapter 1

Introduction

In mixed reality environments, accurate and fast tracking of arbitrary points, such as the user's head and hand, is crucial for creating a compelling virtual environment that provides seamless interaction. A number of tracking technologies and approaches exist, as depicted in Figure 1.1.

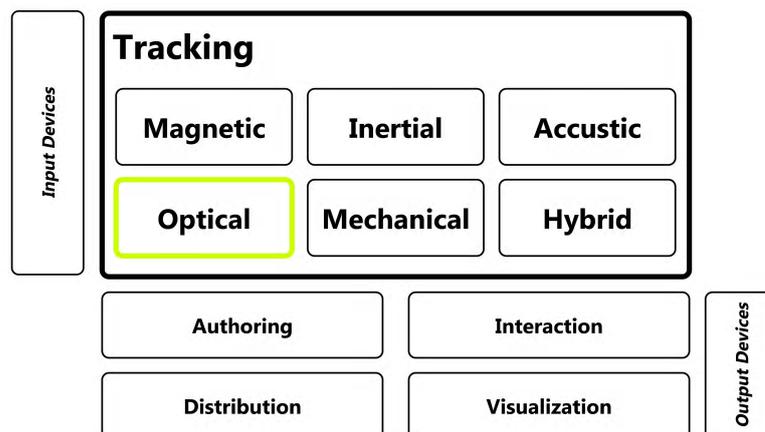


Figure 1.1: Tracking approaches, with the field of contribution marked bold.

All of them have their advantages and disadvantages regarding volume coverage, tracking accuracy, sensitivity to interferences as well as scalability, thus there is no general tracking technology that suits perfectly every tracking scenario. The focus of this thesis is optical tracking, therefore this technology will be discussed and the contribution in this field will be presented within this part of the thesis. For an in depth discussion of the other tracking technologies, the reader is kindly referred to [83].

1. INTRODUCTION

1.1 Motivation & Problem Statement

Optical tracking has been proven to be a reliable alternative to competing tracking technologies since it is less susceptible to noise, it allows multiple objects to be tracked simultaneously, trackable optical markers can be individually designed, they are lightweight, re-configurable and wireless and an optical tracking system can cover large areas. However, state-of-the-art optical tracking systems are mostly designed for standard room sized environments or require a large number of vision sensors (cameras) to cover larger volumes to keep the precision high. This yields significant, high hardware costs as well as complex setup and maintenance routines, making it impractical for general use, especially for non experts. Thus, low-cost wide area tracking with high precision remains a challenge but is indispensable to lower the costs to build compelling immersive virtual environments. The increasing demand for such systems is indicated by the success of recently emerged low-cost hardware, such as the head mounted display Oculus Rift, the Razer Hydra for 3D interaction as well as the Microsoft Kinect for full body motion capture. They massively lowered the initial costs to build a fully immersive VE, but only for small tracking volumes. Furthermore, state-of-the-art optical tracking systems are sensitive to environmental interferences such as lights and reflexions, especially during target training and camera calibration. This yields limited usability in every day tracking scenarios as well as error prone tracking results. Hence, the further employment of virtual reality scenarios for applications that are located in unconstrained environments such as rooms with wall illumination, entertainment stages, manufacturing workshops or even construction sites are impeded by the following three limitations 1) tracking coverage, 2) system sensitivity as well as 3) system scalability & costs.

1.2 Research Objective

To overcome limitations of state-of-the-art optical tracking technology, the following research objectives have been defined. Firstly, a throughout evaluation of existing methods, algorithms and hardware systems is conducted to analyze the requirements to a wide area tracking system for unconstrained environments. Next, existing methods have to be tested, extended and then integrated in a novel system to allow for camera calibration and tracking under heavy interferences. Finally, the system is required to be evaluated in real-life scenarios to draw a robust conclusion on its capabilities, limitations and possible application scenarios.

1.3 Organization

This part is organized as follows. In Chapter II.2, the optical tracking problem is defined, the theory of multi-view imaging to solve the optical tracking problem is discussed and the most common recognition methods are reviewed. In Chapter II.3, competing state-of-the-art tracking approaches for 3D position estimation in indoor environments are reviewed and compared. In Chapter II.4, a description of the developed robust wide-area tracking

system is given and its capabilities and accuracy are evaluated in Chapter II.5 within three different test scenarios; 1) user tracking in a mixed reality setup, 2) handheld target tracking for tunneling application and 3) tracking for machine guidance in underground environments. Finally, Chapter II.6 gives conclusions.

Chapter 2

Theoretical Foundations

In this chapter, we describe the fundamental theoretical concepts of optical tracking.

2.1 Principles of Optical Tracking

The term *tracking* relies to the technology to first detect and then track arbitrary features in space over time to be able to determine the position as well as orientation of the *tracker*, which is the object that observes these features. In optical tracking, the tracker object is an imaging device, such as a mono, color or depth sensing camera.

Pose Tracking In a three-dimensional tracking space, 3D-position and -orientation can be estimated, constituting a *6 degrees of freedom* (DOF) pose of the tracker [62, 83]. 6DOF pose determination is fundamental for view-dependent visualization as well as 3D interaction, thus it is the crucial underlying technology for a mixed reality system.

Tracking Scenarios In an *Outside-Looking-In* tracking scenario, the tracker is fixed and observes a scene to track features (see Section 2.2.1.2) that are attached to an arbitrary object, such as a user. On the contrary, in an *Inside-Looking-Out* scenario, the tracker is attached to the tracking object and observes and tracks fixed features [62].

2.1.1 Accuracy & Performance

The overall capabilities of a tracking system can be expressed by the performance measures, which are described in the following Section 2.1.1.1. The system's performance is thereby influenced by various internal and external sources of error, as specified in Section 2.1.1.2).

2.1.1.1 Performance Measures

Latency describes the time delay between the change in tracker pose and the time, the system has estimated and outputs the new tracker pose [62, 83]. It involves

2. THEORETICAL FOUNDATIONS

data capturing, model recognition and pose estimation. It correlates with the *Update Rate* that is the number of measurements that the tracking system outputs per second. The tracker update rate is usually higher than the overall (system's) update rate. In optical tracking, *Update Rate* and thus *Latency* depend on the imaging device's frequency and the speed of the processing unit to estimate the tracker's pose.

Accuracy expresses the difference of estimated and real tracker pose. It is influenced by internal and external sources of error, *Tracker Jitter* and *Tracker Drift*. *Tracker Jitter* represents the change in tracker output when the tracker object is stationary. *Tracker Drift* is the steady increase in tracker error over time. To avoid error prone tracking results, it must be periodically zeroed by using a secondary tracker of a type that does not have a drift [62, 83]. In case of an optical tracking system, *Tracker Jitter* decreases with increasing the imaging sensor resolution as well as decreasing the distance between tracker and observed feature. *Tracker Drift* can be decreased to zero if position and orientation are estimated with every new incoming image frame.

Robustness expresses the capabilities of the tracking system to uniquely identify the tracker object and to correctly estimate its pose [85]. Robustness relies on the system's ability to deal with the various sources of error, on a proper hardware setup for the intended tracking volume and on a properly designed tracker target model.

2.1.1.2 Sources of Error

Optical tracking systems are very sensitive to the reliability of their inputs. According to [57], overall lighting conditions and estimated camera model (see Section 2.3.4) are two sources of errors. The findings of [57] can be extended and furthermore split into *internal* and *external* sources of errors. Internal sources of error encompassed errors that are implicitly given in optical tracking due to the underlying sensor hardware and data processing. External sources of errors are caused by external circumstances that are present in the tracking volume.

An optical tracking system has to cope with the following internal sources of error.

Optical Aberrations & Camera Model Optical tracking systems require a precise estimation of the camera model's parameter to allow for accurate 3D point computation. The intrinsic camera parameters are required to provide a correct perspective transformation between points in 3D space and points in the 2D camera plane. Since every object lens has (at least minimal) optical aberration that results in distorted camera images, these distortions can be minimized by applying the intrinsic image distortion (radial and tangential) coefficients. The extrinsic camera parameters describe the spatial relationship of the tracking system's cameras that encompasses position and orientation; the parameters highly influences the accuracy of the 6DOF *Pose Estimation* (see Section 2.2).

Image Processing Aberration The target model points must be robustly and precisely segmented within all camera images. Since an image sensor consists of discrete pixels, rasterization causes inaccuracies during *Feature Segmentation* (see Section 2.2). The magnitude of rasterization artifacts depends on imaging sensor resolution, sensor noise as well as tracking distance. Thus, depending on the intended tracking coverage, imaging hardware must be properly selected.

Sensor Noise Thermal deviation influences the amount of noise on the image sensor and causes jitter on the image. Depending on pixel size and density, the sensor temperature and thus jitter can increase. High sensor noise decreases the quality of feature segmentation.

In addition to the internal sources of error, the following external factors can reduce the performance of the tracking system.

Interfering Lights Various light sources, such as sun light, wall illumination and moving light sources can exist in an everyday optical tracking scenario. They can massively interfere with the estimation of the camera model as well as the unique identification of the target model during tracking, resulting in inaccurate pose estimates.

Occlusion Partially occluded target models can result in a complete loss of tracking, or can lead to inaccurate and false positive *Feature Segmentation* and hence *Pose Estimation*.

Target Model The applied target model must be properly designed depending on the intended tracking system coverage to allow for accurate *Model Fitting* and *Pose Estimation* (see Section 2.2). Inaccurate target models result in systematic pose estimation errors.

2.2 Tracking Pipeline

Figure 2.1 shows the optical tracking pipeline that processes the incoming images (frames) to provide the target's 6DOF pose to the system. As it is illustrated in Figure 2.1, the pipeline consist of the following four main sub-tasks:

Feature Segmentation To detect and segment the observed optical feature in a camera image, image processing techniques are applied. They depend on the used optical feature (see Section 2.2.1). To account for image aberrations, the underlying *Camera Model* (see Section 2.3.4) is incorporated into the segmentation process.

Model Fitting To determine the correspondence between the segmented 2D features and the underlying *Target Model*, a fitting routine is performed based on the tracking model properties. Depending on the model, the fitting is performed in 2D or in 3D; for 3D model fitting, the camera model is integrated to transform 2D feature points back into 3D space.

2. THEORETICAL FOUNDATIONS

Pose Estimation Based on the model fitting, the 3D position and/or orientation of the applied target model can be calculated. This process requires both the camera and target model properties.

Predictive Filtering is applied to minimize the influence of tracker jitter, tracking drift and to reduce the effect of latency. This is an optional step in the pipeline.

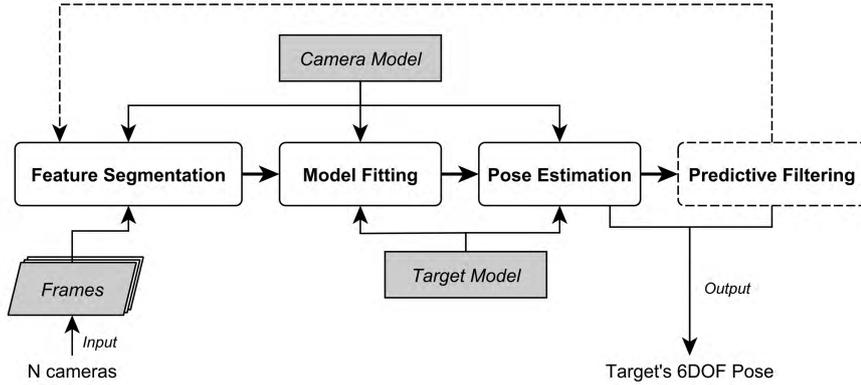


Figure 2.1: The optical tracking pipeline.

Except *Predictive Filtering*, as it is an optional task and does not form the fundamental base for optical tracking, the outlined subtasks are explained in detail in the following sections.

2.2.1 Feature Segmentation

To detect and segment the observed optical features in a camera image, they must be automatically extracted from the incoming frames. Optical tracking approaches can be divided into tracking of *Natural Features* or *Artificial Features*.

2.2.1.1 Natural Features

Natural feature tracking refers to the process of detecting and describing prominent and distinctive structures in the observed environment, such as edges, corners and gradients. Thus, features neither need to be attached to objects that are subject of tracking nor need to be artificially inserted into the tracking environment. With the introduction of robust local descriptors that are invariant to scale as well as rotation and that are robust - up to a certain extent - against illumination changes and viewing direction [58], employing natural features became popular for a broad number of computer vision applications and for optical tracking. Natural feature processing usually consists of three stages *Detection*, *Description* and *Matching*. For detection, distinctive local features are computed in each processed frame. The feature descriptor represents the neighborhood of detected features in a rotation and scale invariant way (i.e. by the computation of a gradient

histogram). To find matches between the corresponding feature points across multiple frames, the similarity between their descriptors is assessed. To facilitate this matching, the descriptors should be distinctive and insensitive to local image deformations.

A wide variety of algorithms exists for feature detection, description, and matching. Prominent examples of feature detectors are *Harris Corner* [12] and *FAST* (Features from Accelerated Segment Test) [74, 110]. Popular methods that comprise feature detection, description, and matching are *SIFT* (Scale-invariant Feature Transform) [36] and *SURF* (Speeded Up Robust Features) [88] that outperforms SIFT in terms of speed and robustness against different image transformation as claimed by its authors. Another well known feature descriptor is *BRIEF* (Binary Robust Independent Elementary Features) [105] that targets real-time applications and allows running feature point matching at low computational costs and memory load. Although it has weaknesses for robust matching in case of large changes in rotation and scale, it performs faster for feature description calculation and matching compared to SIFT. Another fair alternative to SIFT and SURF in terms of computation costs and matching performance is *ORB* (Oriented FAST and Rotated BRIEF) [117] that is rotation invariant and resistant to noise. Internally, it uses FAST for feature detection and a modified BRIEF descriptor to enhance the performance. The recently presented descriptor *FREAK* (Fast Retina Keypoint) [118] is computationally more efficient, computes faster, has lower memory load and is also more robust than SIFT and SURF. Thereby, it is a competitive alternative in particular for embedded applications.

Choosing an adequate feature, and thus an appropriate detector, descriptor and matcher heavily depends on the given application scenario and the requirements. In general, feature descriptors perform too slow to be applied for applications that require high update rates, such as real-time tracking. Therefore, solely applying feature detectors such as FAST is a good choice to estimate the 6DOF pose in an Inside-Looking-Out tracking scenario, as it is computationally efficient and provides a high number of detected features. In application scenarios such as the estimation of external parameters for camera calibration (see Section 2.3.4), which do not necessarily require real-time performance but highly robust features, more computationally complex algorithms might be employed. However, the computation of any kind of natural features requires sufficient illumination and distinct geometrical structure within the observed environment; non-textured surfaces, repeating structures, glass as well as poor illumination yield little, no or unstable features. In case of tracking, this leads to error prone results or even loss of tracking. As our intended tracking environments might not necessarily serve constant illumination and distinct geometrical structures, we focus in this thesis on optical tracking using artificial features.

2.2.1.2 Artificial Features

Artificial feature tracking is based on the detection of predefined, prominent features that are inserted into the tracking volume. These features are then considered as *optical markers* that need to be detected for tracking. Due to the prior knowledge of their properties and their distinctive visual appearance, it is more likely that the tracking

2. THEORETICAL FOUNDATIONS

system is able to detect them with increased robustness, accuracy and speed. They can either be arranged on a planar surface (see Section 2.2.2.1) or consist of spherical optical markers so that their 2D representations in the camera image are defined by circles whose centroids are computed. If multiple markers are rigidly grouped together, they form a *Rigid Body Target* that can be used for *Model Fitting* (see Section 2.2.2).

The optical marker can contain known patterns, it can have a specific shape or color and can be retro-reflective as well as light emitting, as illustrated in Figure 2.2.

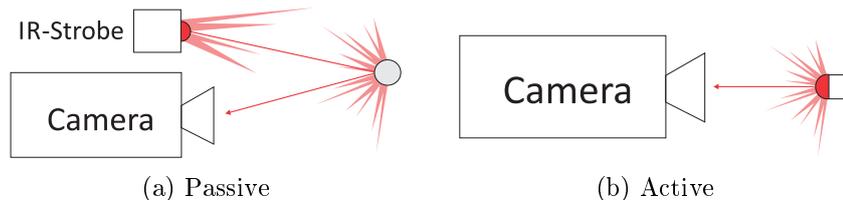


Figure 2.2: Types of optical markers.

Passive markers reflect infrared light that is strobed into the tracking volume back to the camera, while active markers directly emit light towards a camera. Passive markers require special retro-reflective surface coating as well as an additional light emitter to illuminate the whole tracking volume, while in case of active markers, multiple light emitting diodes must be individually powered. Spherical shaped optical markers result in circular pixel-blobs (*Blob*) in the camera image whose centroid is computed for model fitting and pose recognition.

2.2.2 Model Fitting

The process of *Model Fitting* describes the problem of determining the correspondences between the detected 2D image features and the optical features of the tracked object. It can be accomplished by matching and fitting to the underlying *Target Model*, that describes the structure of features on the tracked object.

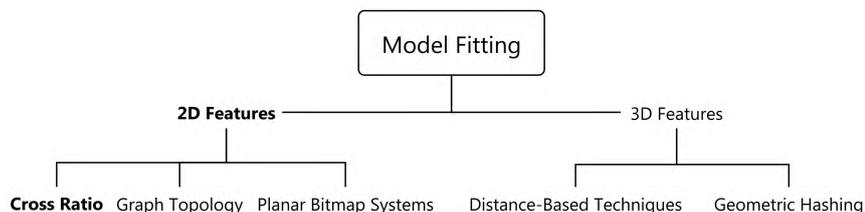


Figure 2.3: Taxonomy of model fitting depending on domain and property.

As depicted in Figure 2.3, methods for model fitting can be divided into techniques that are either applied in the 2D- or 3D-domain.

2.2.2.1 2D Domain

The target model of the tracked objects can be completely identified in 2D by processing the imaging data from a single camera. This is generally accomplished by exploiting properties that are invariant under perspective projection. There are a number of projective invariant properties such as *Cross Ratio* and *Graph Topology* or *Planar Bitmap Targets* that share the idea of projective invariant properties. The three approaches are briefly described in Table 2.1.

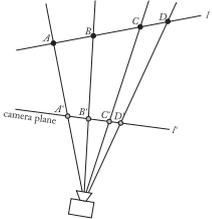
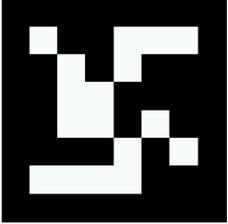
	<p>Cross Ratio When projecting 3D points onto a 2D camera plane, neither distances nor ratios of distances are preserved [56]. However, the <i>Cross-Ratio</i> as a ratio of distances as well as the collinearity of point sets is preserved [17].</p>
	<p>Graph Topology When projecting a 2D graph structure as depicted in the Figure (Source: [86]) onto a camera image plane, its topology remains constant, as long as the parts of the graph do not overlap. Then, model fitting and pose estimation can be performed, as proposed in [76, 86].</p>
	<p>Planar Bitmap Systems Planar bitmap systems, such as [35, 163], encode information into a bitmap that can be retrieved after perspective projection. Using a planar pattern, optical aberrations can be accurately removed for robust pattern recognition by using correlation techniques.</p>

Table 2.1: Projective invariant features in the 2D domain.

Graph topology and planar bitmap patterns are useful for many applications. While the binary patterns of ARToolkit and Vuforia [35, 163] must be fully visible and cannot cope with occlusions, ARTag [63] introduced an error correcting code as bitmap to reduce the occlusion problem. Graph topology [76] is more robust and can cope with partial occlusions. However, to be able to detect targets at larger distances that are based on graph topology or planar bitmap patterns, large targets would be required. This reduces usability and increases manufacturing effort. In contrary to planar targets, target models that exploit the cross ratio of its markers can be designed more flexibly since only a minimum of four points are required. The tracking system that is presented in Chapter II.4 accomplishes model fitting by evaluating the cross ratio. Therefore, the underlying approach is described in detail in the next paragraph.

2. THEORETICAL FOUNDATIONS

Cross Ratio As described in [54, 85], the *Cross Ratio*, as a ratio of ratios of distances, can be computed based on four collinear points, labeled as A, B, C, D .

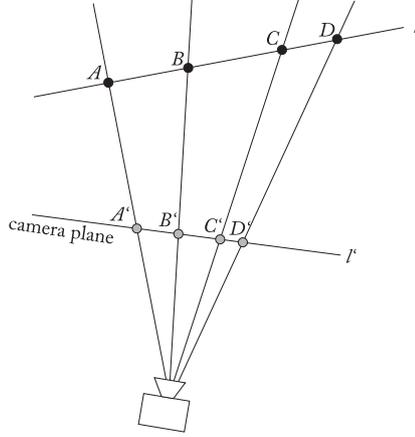


Figure 2.4: After perspective projection of the four points, the projective invariant properties of the cross ratio are expressed by $\lambda(A, B, C, D) \doteq \lambda(A', B', C', D')$. The points' collinearity is preserved as well, as $l \doteq l'$.

The cross ratio is defined as the real number λ by

$$\lambda = \frac{|AB|/|BD|}{|AC|/|CD|}, \quad (2.1)$$

where $|AB|$ denotes the length of the line segment between points A and B . Its projective invariant properties are illustrated in Figure 2.4. As it can be seen in Equation 2.1, the computation of the cross ratio depends on the order of the four points (*quadruple*), resulting in $4! = 24$ possible orderings. Instead of comparing the cross ratio of the detected features during model fitting with all possible permutations, p^2 -invariants according to [30] can be computed. These are representations of point sets that are insensitive to projective transformations and permutations of the labeling of the quadruple. These p^2 -invariants use λ as argument for the projective and permutation invariant function $J(\lambda)$ that is determined as follows:

$$J(\lambda) = J_2(\lambda)/J_1(\lambda), \quad (2.2)$$

where J_1, J_2 are computed as linear combinations of λ , as denoted in Equation 2.3.

$$\begin{aligned} J_1(\lambda) &= \frac{(\lambda^6 - 3\lambda^5 + 3\lambda^4 - \lambda^3 + 3\lambda^2 - 3\lambda + 1)}{\lambda^2(\lambda - 1)^2} \\ J_2(\lambda) &= \frac{(2\lambda^6 - 6\lambda^5 + 9\lambda^4 - 8\lambda^3 + 9\lambda^2 - 6\lambda + 2)}{\lambda^2(\lambda - 1)^2} \end{aligned} \quad (2.3)$$

Training Before tracking, the properties of the pattern i must be obtained once during a training phase. Therefore, the target's points are detected in the camera image and checked for collinearity. However, collinearity and cross ratio are sensitive to noise (see Section 2.1.1.2) that influence the accuracy of point segmentation. To account for noise when computing the points' collinearity, the following metric, as introduced in [14] and further described in [54], is used for determining the collinearity of three points (*triple*): for a triple of homogeneous points p_1, p_2, p_3 , where $p_j = (x, y, 1)$, $j = 1, \dots, 3$, define a moment matrix $M_{123} = \sum p_i p_i^t$ and calculate its smallest eigenvalue ev_{123} . For three "perfectly" collinear points, $ev_{123} = 0$, indicating their linear dependency. If the point coordinate computation is influenced by noise, $ev_{123} \neq 0$ but provides an approximation of the three points' collinearity. During training, the smallest eigenvalue of the moment matrix for all three triples of the quadruple is calculated and the maximum smallest eigenvalue ev_i^{max} of all triples is stored. To account as well for noise during the p^2 -invariant calculation, the minimum J_i^{min} and maximum J_i^{max} values of the pattern's p^2 -invariant are stored, denoted as

$$p_{range}^2 = [J_i^{min}, J_i^{max}]. \quad (2.4)$$

Summarizing, the pattern's p^2 -invariant properties encompasses ev_i^{max} and p_{range}^2 that are subsequently used to determine the target at the model recognition stage.

Model Recognition During tracking, model fitting is performed by employing the following two steps.

1) For each detected quadruple Q_j , compute the maximum smallest eigenvalue ev_j^{max} and perform a collinearity check to find all possible quadruple candidates $Q_{cand,1} \dots Q_{cand,n}$, by

$$Q_j = \begin{cases} Q_{cand,n} & \text{if } ev_j^{max} \leq ev_i^{max} \\ \emptyset & \text{otherwise} \end{cases}$$

2) For each candidate $Q_{cand,1} \dots Q_{cand,n}$, compute its $p_{cand,n}^2$ -invariant and perform p_{range}^2 -test to identify the quadruple of the target model Q_{model} , by

$$Q_{cand,n} = \begin{cases} Q_{model} & \text{if } J_i^{min} \leq p_{cand,n}^2 \leq J_i^{max} \\ \emptyset & \text{otherwise} \end{cases}$$

To summarize, by exploiting the projective invariant properties of a rigid body target that is equipped with four optical markers, a computationally lightweight 2D model fitting approach is provided. The main advantage of model fitting in 2D over the 3D domain is that no stereo correspondence is required, hence it can be performed without knowledge of the external camera parameters, as described in Section 2.3.2.5.

2. THEORETICAL FOUNDATIONS

2.2.2.2 3D Domain

To detect a target model that features a three dimensional geometric constellation, as illustrated in Figure 2.5, multiple views of the scene as well as the cameras' stereo geometry are required. Features across multiple views are first matched by applying stereo correspondence matching (see Section 2.3.3.2) and then transformed into 3D by applying projective triangulation (see Section 2.3.3.4). Within the resulting 3D point cloud, the target model can be fitted using geometric hashing, exploiting the euclidean distances between the 3D points or by combining both base techniques [84].

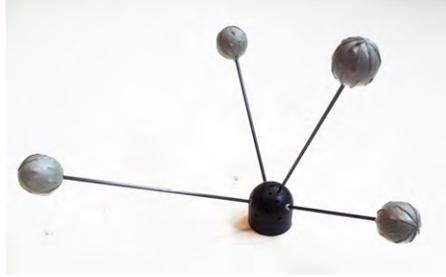


Figure 2.5: An example of a passive 3D rigid body target.

3D Distance Techniques 3D distance fitting methods basically exploit the Euclidean distances between unique points within the target model's geometric constellation. In [33, 48], trackable objects are equipped with three optical markers that form a non-regular triangle in which all inter-marker distances have to be unique. In a preprocessing stage, the model of this triangle is obtained and then applied to the detected 3D points during tracking by minimizing the sum of differences between the model markers distances and the measured distances. As a generalization of [33], model fitting based on the distance property is used by applying point patterns [57]. By measuring the 3D distance between each marker pair of the pattern, a pattern distance matrix P is constructed. During the recognition step, a distance matrix C of all detected 3D points is calculated. In C , all sub matrices C_i are determined that fit in P . A least squares fitting metric is applied to find the sub-matrix within C_i that best matches P .

Geometric Hashing The geometric hashing approach, introduced by [1], identifies the target model's features in a set of detected 3D features based on a lookup table. The model's features are represented in an affine invariant as well as redundant way (to account for occlusions) and are stored in a hash lookup table, which is generated in a preprocessing stage. Here, for each of three non-collinear model features, a coordinate system basis with respect to the three features is defined, then the features are parameterized with respect to the basis and stored in a 3D hash table. To detect the target model during tracking, three detected features are selected, parameterized with respect to their coordinate system base and then a hash table lookup is performed; matches in the hash table vote for this model. For each candidate model the affine transformation is recovered,

the candidate is transformed, and tested against the target model; if the match is not sufficient, another three detected features are picked and the process is repeated. Due to the required transformation of data points into a reference frame, the effectiveness of geometric hashing highly depends on the amount of candidates that need to be examined.

2.2.3 Pose Estimation

Pose Estimation describes the problem of determining the position and orientation of the tracked target model in 3D space. For that, the detected model points, the target model points and the multiple view geometry that is encapsulated in the camera model must be given. As shown in Figure 2.6, the pose can be estimated by either using techniques for computing the 3D rigid transformation between two sets of point-correspondences or, if a one-to-one correspondence between the detected points and the target model points could not be obtained, by applying optimization techniques.

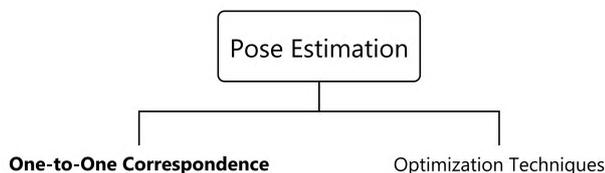


Figure 2.6: Taxonomy of pose estimation.

A general purpose, representation-independent optimization approach is the Iterative Closest Point (ICP) algorithm, introduced by [15] that matches a set of obtained points to the points of a model, either in 2D or 3D. In the tracking system that is presented in Chapter II.4, a pose estimation by a one-to-one point correspondence can be obtained. It is described in detail in the next paragraph.

One-to-One Correspondences Before estimating the pose, the 3D representation of detected points must be computed. Hence, for 2D model fitting, a transformation of the detected 2D model points must be first performed. This is accomplished by determining the 2D point correspondences across multiple views and by calculating their 3D positions using multiple view geometry (see Section 2.3.3). As soon as a one-to-one point correspondence between obtained 3D and model points is given, the pose estimation problem can be reduced to the absolute orientation problem. The 3D rigid transformation between these two points sets can be generally expressed by Equation 2.5.

$$p_i = Rm_i + T + V_i \tag{2.5}$$

Given the 3D data points set p_i and the corresponding model points m_i , $i = 1 \dots N$, the rotation R , the translation vector t and the noise vector V_i shall be obtained in order to optimally map $m_i \rightarrow p_i$. Solving for the optimal transformation $[\hat{R}, \hat{T}]$ typically requires

2. THEORETICAL FOUNDATIONS

to minimize a least squares error criterion ϵ^2 that is given by

$$\epsilon^2 = \sum_{i=1}^N \|p_i - \hat{R}m_i - \hat{T}\|^2. \quad (2.6)$$

There are a number of closed-form solutions to solve this problem, such as a quaternion-based approach [11] or by computing the singular-value decomposition of a derived matrix [10]. An overview of the four major techniques is given in [25], concluding that none of the four algorithms was found to be superior in all cases. The only truly distinguishing factor was determined in execution time that also depends on data set size and computer hardware and configuration. Thus, the choice of the algorithms depends mostly on data set size.

2.3 Projective Geometry

Projective geometry serves as a mathematical framework for 3D multi-view imaging and can be applied to model the mapping between 3D world points and 2D image points, known as the image-formation process as well as to reconstruct 3D objects from multiple images. Thus, projective geometry can serve as the underlying mathematical framework to solve for the following requirements of the thesis' optical tracking pipeline:

1. An abstract camera model, including a description to model the relationship between a 3D world point and its corresponding 2D image point, and vice versa.
2. A geometric foundation to search and describe point correspondences across multiple camera views as well as to reconstruct 3D geometry.

In 3D space, lines, planes and points are usually described using *Euclidean Geometry*. A point $X \in \mathbb{R}^3$ in Euclidean space is represented in so called *inhomogeneous coordinates* with a 3-element vector $(x, y, z)^T$. To avoid the disadvantage of Euclidean geometry when projecting a 3D point onto an image plane¹, projective geometry can be applied, representing X in *homogeneous coordinates* as a 4-element vector $(x_1, x_2, x_3, x_4)^T$, such that

$$x = \frac{x_1}{x_4}, y = \frac{x_2}{x_4}, z = \frac{x_3}{x_4}, \text{ where } x_4 \neq 0. \quad (2.7)$$

The general mapping between a point in an n -dimensional Euclidean space to a $(n + 1)$ -dimensional projective space employs the homogeneous scaling factor λ and can

¹This operation requires a perspective scaling operation (division) using a scale factor, resulting in a non-linear operation.

then be described as

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \lambda x_3 \\ \vdots \\ \lambda x_n \\ \lambda \end{pmatrix}, \text{ where } \lambda \neq 0. \quad (2.8)$$

Projective geometry is used in conjunction with the *Basic Pinhole Camera*, as described below. It is the most specialized and simplest camera model and acts as a mathematical foundation for the presented tracking approach of this thesis. For a comprehensive and in depth review of single and multiple view projective geometry, the reader is referred to [56].

2.3.1 The Pinhole Camera Model

Generally said, a camera model is represented by matrices and describes a mapping between a 3D world (object space) and a 2D image (image space). The pinhole camera model performs this $3D \rightarrow 2D$ mapping as a projective projection. The geometry of the pinhole camera is illustrated in Figure 2.7.

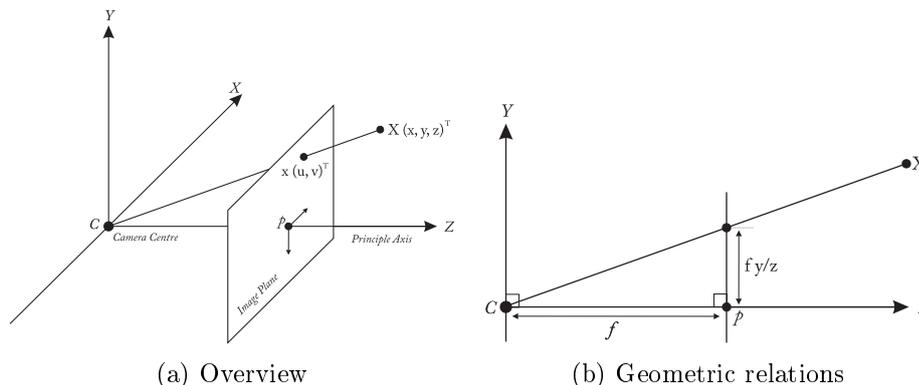


Figure 2.7: The pinhole camera geometry with camera center C coincides with the coordinate system's origin. The image plane is placed with distance f in front of C .

The center of the perspective projection C is the point in which all incoming rays intersect and is denoted as *camera center* (or *optical center*). With the pinhole camera model, a point $X = (x, y, z)^T \in \mathbb{R}^3$ is mapped to a point $x = (u, v)^T \in \mathbb{R}^2$ on the *image plane* (or *focal plane*) where a line from X to C meets the image plane. The *principle axis* (or *optical axis*) is the line perpendicular to the image plane passing through C . The principal point p is denoted as the point where the principle axis intersects with the image plane. The plane through C that is parallel to the image plane is called *principle plane*.

2. THEORETICAL FOUNDATIONS

Let C be the origin of the Euclidean coordinate system and the principal axis being collinear to the Z -axis. Consider the image plane placed at $Z = f$, where f denotes the *focal length*. As illustrated in Figure 2.7b, the point $(x, y, z)^T$ can be mapped to $(f x/z, f y/z, f)^T$ on the image plane. Ignoring the final coordinate, the above mapping can be expressed as a projective mapping by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} \frac{f x}{z} \\ \frac{f y}{z} \end{pmatrix}. \quad (2.9)$$

As mentioned in Section 2.3, such a non-linear division operation should be avoided. Using projective geometry and homogeneous coordinates, the relation from Expression 2.9 can be re-formulated in terms of matrix notation as

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} f x \\ f y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad (2.10)$$

where the homogeneous scaling factor $\lambda = z$ and where the homogeneous 3×4 matrix is called the *Camera Projection Matrix* P . Expression 2.10 can be written compactly as

$$x = PX. \quad (2.11)$$

Deriving from Expression 2.10, P is defined for the pinhole model as

$$P = K[I | 0], \quad (2.12)$$

with K denoting the *Camera Calibration Matrix* and expressed by

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.13)$$

2.3.2 Camera Model Extensions

The basic pinhole camera models the $3D \rightarrow 2D$ point mapping for a system that does not suffer from aberrations caused by the employed optic components and by the imaging sensor. However, in practice these aberrations occur and thus, the underlying camera model must describe these properties as well to allow for a precise projective mapping.

2.3.2.1 Principal Point Offset

The expression from Section 2.3.1 assumes that the origin of coordinates in the image plane coincides with the principle point p . In practice, the imaging systems often define

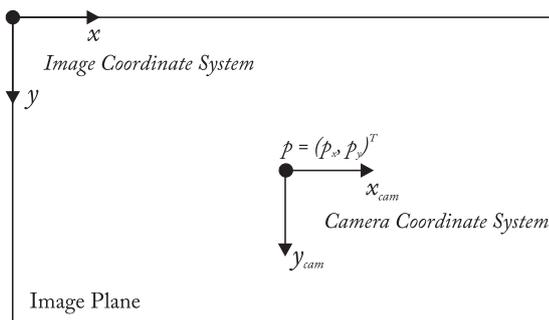


Figure 2.8: The principal point offset.

the origin of the pixel coordinate system at the top-left pixel of the image, as depicted in Figure 2.8. Thus, a conversion of coordinate systems is necessary.

Let be $(p_x, p_y)^T$ the coordinates of p ; then the Expression 2.10 can be extended by integrating the principal point position into K , resulting in

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} f x + z p_x \\ f y + z p_y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2.14)$$

2.3.2.2 Skew Parameter

In Equation 2.10, it was further implicitly assumed that the pixels of the image sensor have equal scales m_x, m_y in both axial directions with a square aspect ratio (i.e. 1 : 1) and are not skewed. However, in practice it might not be the case. To account for both imperfections of the imaging system, the parameters $m = (m_x, m_y)$ and s can be employed to model non-squared and skewed pixels. The projective mapping is then denoted as

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f m_x & s & p_x & 0 \\ 0 & f m_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2.15)$$

The imaging hardware that is employed throughout this thesis has squared and non-skewed pixels. Thus, we assign $m_x, m_y = 1$ and $s = 0$.

2.3.2.3 Camera Lens Distortions

In practice, distortion effects can be observed in most camera lenses. Incorporating these distortions adds non-linear components to the linear transformations, as defined by Equation 2.10.

Radial lens distortion maps straight lines as curves, with increasing magnitude towards the image edges. It is generally stronger in wide-angle lenses and the most present

2. THEORETICAL FOUNDATIONS

form of lens distortion. Two types of radial distortion can be distinguished and are depicted in Figure 2.9. The barrel radial distortion maps lines curved outwards from the image center while the pincushion radial distortion maps lines pinched towards the image center.

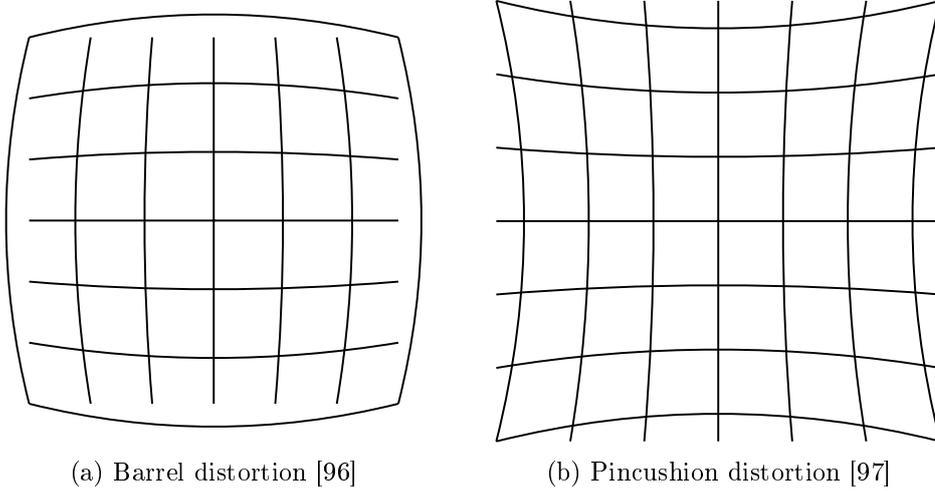


Figure 2.9: Two common types of radial distortion.

Tangential lens distortion is caused by imperfect centering of the lens components and by other manufacturing defects. It results in the lens not being exactly parallel to the imaging plane.

According to [27], the overall lens distortion can be accurately modeled by the sum of the radial and tangential distortion vectors to map the image coordinates $\langle u, v \rangle$ to their distorted counterparts $\langle \hat{u}, \hat{v} \rangle$. To describe the radial distortion, k_i denotes the radial distortion coefficients and $r = \sqrt{u^2 + v^2}$. The radial distortion vector is then defined as

$$\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} u(k_1 r^2 + k_2 r^4 + \dots) \\ v(k_1 r^2 + k_2 r^4 + \dots) \end{pmatrix}. \quad (2.16)$$

The tangential distortion vector with the coefficients p_1, p_2 is defined by

$$\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} 2p_1 uv + p_2(r^2 + 2u^2) \\ p_1(r^2 + 2v^2) + 2p_2 uv \end{pmatrix}. \quad (2.17)$$

For computing $\langle u, v \rangle$ based on $\langle \hat{u}, \hat{v} \rangle$, usually called *inverse mapping* or *normalization*, no general algebraic expression exists [27] because of the high degree distortion model. However, a number of approximative solutions exist, such as a numerical approach [133] or recovering the real pixel coordinates from the distorted ones by involving a non-linear search for implicit parameters [27].

2.3.2.4 Camera Rotation & Translation

In the above equations to model the basic pinhole camera and its extensions to describe the additional internal camera parameters, it was assumed that the origin of the *camera coordinate system* coincides with the origin of an Euclidean coordinate system that the principal axis is pointing straight down the camera's z-axis (see Figure 2.7), so that a 3D point X can simply be expressed in the camera coordinate system by Equation 2.12 that was

$$x = PX, \text{ with } P = K[I | 0].$$

To unbound from this constraint, points in 3D space are generally expressed in terms of a different Euclidean system, the *world coordinate system*. World and camera coordinate system are related to each other by a rotation R and a translation t , as depicted in Figure 2.10.

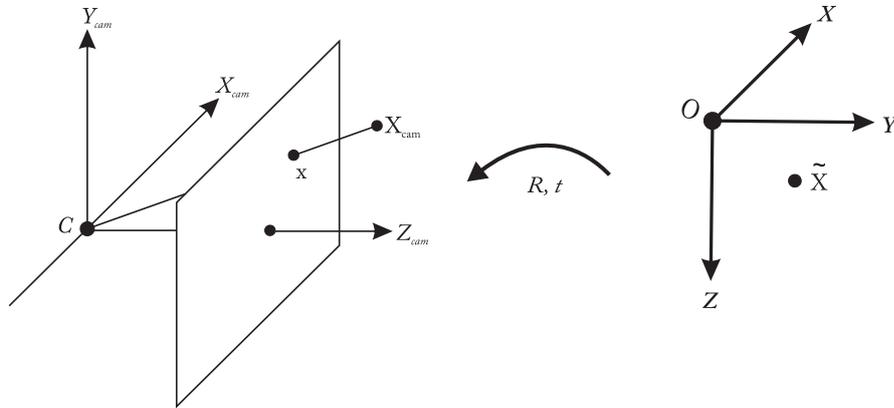


Figure 2.10: The Euclidean transformation between the world and the camera coordinate system.

Given a point \tilde{X} in the world coordinate system, the same point in the camera coordinate system X_{cam} is obtained using homogeneous coordinates by

$$X_{cam} = \begin{pmatrix} R & -R\tilde{C} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} R & -R\tilde{C} \\ 0 & 1 \end{pmatrix} \tilde{X}. \quad (2.18)$$

where \tilde{C} is the position of the camera center C in world coordinates and R describes the orientation of the camera coordinate system with respect to the world coordinate system. \tilde{C} is determined by translating C with t to the world coordinate origin O and then rotate it by R . To obtain the pixel coordinates of $x = (u, v, 1)^T$ of \tilde{X} , the updated spatial relations are fused into Equation 2.12 so that

$$x = P\tilde{X}, \quad (2.19)$$

with $P = K[R | t]$, $t = -R\tilde{C}$.

2. THEORETICAL FOUNDATIONS

2.3.2.5 Intrinsic & Extrinsic Camera Parameters

Summarizing Sections 2.3.1 and 2.3.2, the mathematical description of an abstract camera model with its extensions is given.

Intrinsic Camera Parameters comprises focal length, principal point offset, pixel scale as well as skew and are expressed by the *Camera Calibration Matrix* K . The intrinsic parameter lens distortion is defined by the radial and tangential coefficients k_i and p_1, p_2 , respectively. All intrinsic camera parameters remain constant unless the optical setup is modified.

Extrinsic Camera Parameters describe the external position and orientation of the camera in respect to the 3D world and are expressed by the homogeneous 4×4 matrix $[R|t]$. As soon as the camera is moved in the world space, the extrinsic parameters must be recomputed.

K and $[R|t]$ are encapsulated in the *Camera Projection Matrix* P , thus P relates 3D space measurements to image measurement. Consequently, P depends on both the world coordinate and image coordinate system.

Determining the intrinsic and extrinsic camera parameters yields the process of *Camera Calibration*. Estimating internal and external camera parameters in one process is known as *strong calibration*, determining only one parameter set at a time is called *weak calibration*. Camera calibration methods are reviewed in Section 2.3.4. Before optical tracking with multiple cameras can be performed, the camera parameters must be known.

2.3.3 Multiple-View Geometry

After reviewing the abstract model of a single camera to describe the relationship between a 3D world to corresponding 2D image points, the geometric foundation to reconstruct the 3D point's coordinates out of corresponding image points across multiple camera views is reviewed in this section. 3D reconstruction of a point is an indispensable task in the tracking pipeline.

2.3.3.1 Epipolar Geometry

The geometric model to search for point correspondences across multiple camera views and to model the spatial camera constellation to be able to estimate the 3D position of a corresponding point pair is known as *Epipolar Geometry*.

As depicted in Figure 2.11, it is constituted between the non-coincident optical centers C, C' of two pinhole cameras and a 3D point $\tilde{X} \in \mathbb{R}^3$. \tilde{X} is projected onto both image planes, resulting in the corresponding 2D point pair $x, x' \in \mathbb{R}^2$. The epipolar geometry is then expressed by

- The baseline, as the line going through C, C' .
- The *epipolar plane*, which is defined by a \tilde{X} and C, C' .

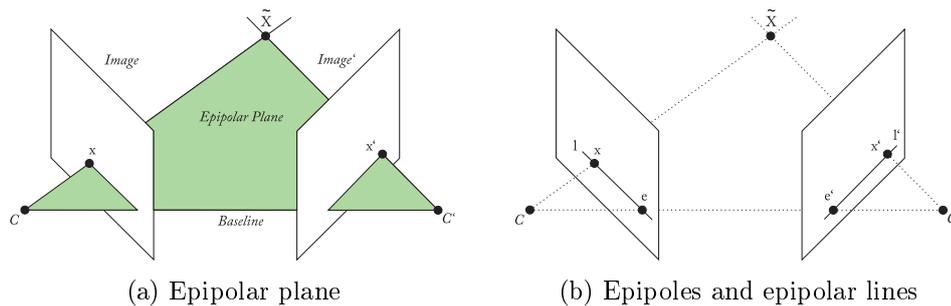


Figure 2.11: The epipolar geometry.

- The *epipolar line* l , which is determined by the intersection of the image plane with the epipolar plane. It passes through the projected point and the epipole e of the first image plane (i.e. x and e) and is the projection of the optical ray that runs through the optical center and the projected 2D point of the second image plane (i.e. C' and x').
- The *epipole* e as the 2D image point where the baseline intersects with the image plane. All epipolar lines in an image pass through the epipole, which also corresponds to the projection of the optical center of the other camera onto the image plane, i.e. e' is the projection of C .

The epipolar geometry is algebraically represented by the *Fundamental Matrix* F that is a homogeneous 3×3 matrix of *rank* 2 that satisfies

$$x'^T F x = 0 \quad (2.20)$$

for all corresponding points $x \leftrightarrow x' \in \mathbb{R}^2$. After estimating F , as described in Section 2.3.4, the geometric model of the epipolar geometry can be exploited to estimate the unknown coordinates of a 3D point \tilde{X} by performing the following steps:

Solve Correspondence Problem For x , its corresponding point x' is constrained to lie on the epipolar line l' . Using this *epipolar constraint*, x' can be determined by performing a search along the epipolar line l' . However, correspondence ambiguities can occur and must be robustly resolved before determining the corresponding 2D point pair, as described in Section 2.3.3.2.

Compute Camera Projection Matrices After solving the stereo correspondence, P and P' for both cameras must be derived, as described in Section 2.3.3.3. They encapsulate both internal and the external parameters of each camera. The external parameters describe the position and orientation of each individual camera in relation to each other, respectively in relation to the world coordinate system.

3D Point Reconstruction Based on the known camera projection matrices and the point correspondence x, x' , the unknown 3D coordinates \tilde{X} in the world coordinate

2. THEORETICAL FOUNDATIONS

system can be reconstructed by performing a *Projective Triangulation*, as described in Section 2.3.3.4.

2.3.3.2 Stereo Correspondence Problem

As it is defined by the epipolar geometry, for an image point x , its corresponding point x' lies on the epipolar line l' . This search along the line can be reduced to a one-dimensional search problem when all epipolar lines are parallel. This can be achieved by rectifying the image pair, as described in [56]. Image rectification is advisable for images taken from widely different viewpoints. However, even when reducing the dimension, the search along the epipolar line can be ambiguous, since multiple features in the right image may lie on the same epipolar line of a feature in the left image. To solve for this stereo correspondence problem, further matching constraints that exploit the features' properties, such as *Similarity*, *Uniqueness*, *Continuity* and *Ordering of Points*, can be applied [51]. As the tracking approach of this thesis is based on infrared optical tracking of spherical shaped markers, the resulting blobs in the camera images do not contain enough information to use the above mentioned characteristics. The blobs provide practically identical characteristics in both images, thus, model fitting and recognition methods as described and discussed in Section 2.2.2 must be applied in conjunction with stereo correspondence search to solve correspondence ambiguities.

2.3.3.3 Computing the Camera Projection Matrix

Based on F , the camera projection matrices P, P' of both cameras can be derived. However, since this results in a projective ambiguity, it is more advisable to use the Essential Matrix E to extract P, P' up to scale. E is a specialization of F using normalized image coordinates, thus the camera calibration matrices (K, K') of both cameras must be known. E can then be obtained by

$$E = K'^T F K. \quad (2.21)$$

A pair of corresponding 2D image points x, x' are normalized by computing $\hat{x} = K^{-1}x$, respectively $\hat{x}' = K'^{-1}x'$, re-expressing the defining equation of F from Equation 2.20 as

$$\hat{x}'^T E \hat{x} = 0. \quad (2.22)$$

Using normalized image coordinates, Equation 2.19 can then be reformulated as

$$\hat{x} = [R | t] \tilde{X}, \text{ where } P = [R | t]. \quad (2.23)$$

This can be thought of as projection of \tilde{X} onto the image plane with respect to a camera $[R | t]$ that has an identity matrix I as K . Since K, K' are given, only rotation and translation from one camera to the other needs to be determined. As it is given for F , a pair (P, P') uniquely determines E , however the inverse is not true. Thus, it is common to define (P, P') as

$$P = [I | 0], \quad P' = [R | t] \quad (2.24)$$

and compute P' by factorizing E into the product SR , where S is a skew symmetric matrix and R is a rotation matrix, using *Single Value Decomposition* (SVD). As reviewed in detail in [56], this results in a four-fold ambiguity, meaning that there are four possible geometrical combinations of translations and rotations giving four possibilities for P' , as illustrated in Figure 2.12.

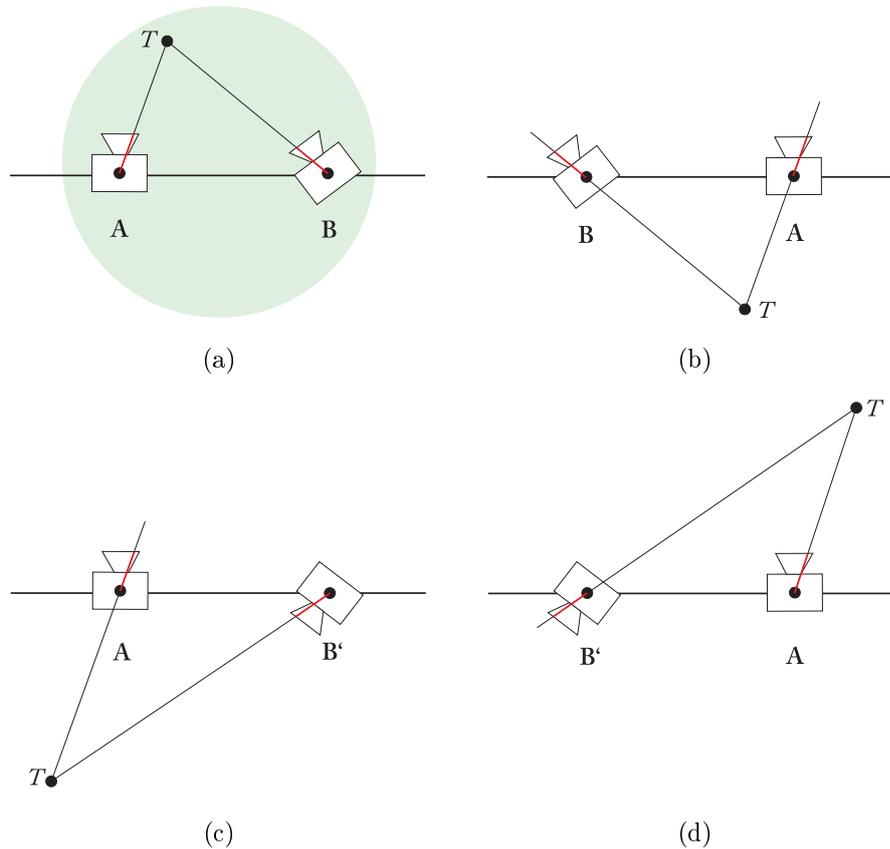


Figure 2.12: The four possible solutions for P' , as combinations of rotations and translations.

Between 2.12a and 2.12b, respective 2.12c and 2.12d, the translation vector is reversed (baseline reversal). Between 2.12a and 2.12b, respective 2.12c and 2.12d, camera B rotates 180° about the baseline. As it is shown, only in one of the four solutions the point T is in front of both cameras. Thus, it is sufficient to test with a single point to determine if it is in front of both cameras to solve the four-fold ambiguity. Therefore, a test point from the data is taken, its 3D coordinates are reconstructed with each combination of (P, P') , then the 3D point's depth in both cameras is determined and finally, the pair (P, P') is chosen that has a positive depth for both cameras.

2. THEORETICAL FOUNDATIONS

2.3.3.4 3D Point Reconstruction

The process of reconstructing the unknown coordinates of a 3D point \tilde{X} from two corresponding 2D image points x, x' is known as *Back-Projection* or *Triangulation*. The triangulation problem is defined as determining the intersection of the two rays in space that correspond to x, x' ; this intersection is then \tilde{X} . These two rays will meet in space if and only if x, x' satisfy the epipolar constraint from Equation 2.20, respective 2.22. In the absence of 2D point measurement inaccuracies, the triangulation problem can then be easily solved and there is a point \tilde{X} that projects to $x \leftrightarrow x'$ and thus exactly satisfies $x = P\tilde{X}$ and $x' = P'\tilde{X}$. However, digitalization errors such as sensor noise result in erroneous measured points x, x' that do not in general satisfy the epipolar constraint.

In this case, a pair of optimized image points $\hat{x} \leftrightarrow \hat{x}'$ must be determined that reproduces the erroneous measured points $x \leftrightarrow x'$ as closely as possible by minimizing the residual errors between the reprojected and measured image points [82] and satisfying the epipolar constraint $\hat{x}'^T F \hat{x} = 0$. Once $\hat{x} \leftrightarrow \hat{x}'$ are found, their corresponding rays will meet precisely in space and \tilde{X} can be obtained by any triangulation method, such as the *Linear Triangulation*, *Linear Least Squares Triangulation* or *Bundle Adjustment* [26, 82].

2.3.4 Camera Calibration

Determining both the intrinsic and extrinsic camera parameters yield the process of *Camera Calibration*. For each camera that is involved in a multiple view setup, both parameter sets are described by the *Camera Projection Matrix* P . A number of calibration approaches exist, and all share the common principle of determining the cameras' parameter by initially obtaining a specific number of 3D world \rightarrow 2D image point relations, to later use these relationships in an optimization procedure. The existing approaches for multiple view camera calibration can be described based on the applied calibration object and its dimensionality, as illustrated in the taxonomy of Figure 2.13.

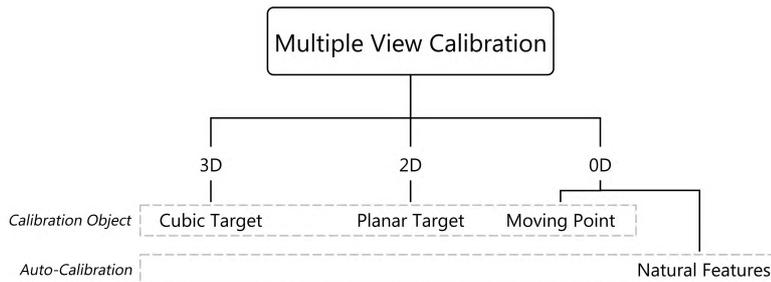


Figure 2.13: A calibration taxonomy by dimension of the applied apparatus.

Calibration based on a 2D or 3D reference target usually observes the object that is only shown at a few different orientations [38, 40] undergoing an unknown translation. The object's 2D, respective 3D geometry, is known with high precision. In 2D, this is typically a planar pattern (see Figure 2.14a), and in 3D two or three planar pattern in

an orthogonal geometric arrangement to each other. With such a reference object, each cameras' internal (focal length, principal point offset, aspect ratio, radial and tangential distortion coefficients) and external parameters (position and orientation) can be computed efficiently [17]. The required calibration setup can be easily constructed, however this approach suffers from declining ease of use due to increasing necessary pattern size as calibration distance to the camera and baseline increases.

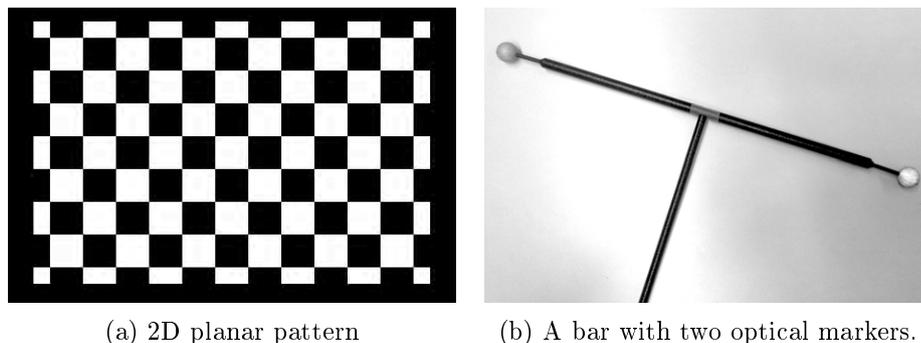


Figure 2.14: Reference targets for intrinsic and extrinsic camera calibration.

Multiple view camera calibration can also be performed with a 0D object, such as corresponding points across the views. Either these points are manually generated by waving a single point [45, 69], such as a light emitting diode, retro-reflective sphere, through the volume or by extracting natural features [102, 116, 99, 125] from the observed scene, which is referred to as *Auto-Calibration*. Since these single point methods cannot account for the estimation of distortion coefficients, they are mostly intended to recover only the extrinsic parameters. To overcome this limitation, a 2D planar pattern calibration can be applied before recovering all internal parameters. Using a moving point or extracting natural features from the image, a sufficient number of corresponding image pairs (a minimum of seven is required) can be computed to estimate the Fundamental Matrix F , i.e. by performing the *Normalized 8-Point Algorithm* [9, 56]. It seeks for estimating F by constructing a set of linear equations, or in presence of noise, by solving a linear least square minimization problem. As described in the computation of the camera projection matrix (Section 2.3.3.3), the extrinsic parameters can be derived through this estimated epipolar geometry, up to a scale factor. To overcome the limitations of the calibration based on multiple single points, in [48] an extension is presented; a bar with optical markers at both ends is used as calibration target where the physical distance between the spheres is known (see Figure 2.14b). Thereby, internal and external camera parameters can be determined linearly in an initialization step, and then refined with a nonlinear least squares optimization method. Furthermore, the scale factor can be determined from the real and known distance between both spheres.

As stated in [60], there is no calibration technique that suits best for all use cases. However, the following recommendations are given that influenced the design of the calibration approach of this thesis:

2. THEORETICAL FOUNDATIONS

- Whenever possible, calibrate the cameras in a single or multi-view setup with a 2D or 3D reference object. Calibration with single point correspondences cannot usually achieve an accuracy comparable to a calibration using a higher dimensional reference object.
- Whenever possible, calibrate as many parameters with a calibration object as possible. Thereby, the number of parameters to be estimated can be reduced for any subsequent self-calibration.

2.4 Summary

This chapter has introduced the fundamental concepts of optical pose tracking and projective geometry. Projective geometry uses homogeneous coordinates to represent the position of 2D image and 3D world points and is able to describe the projection of a 3D point onto a 2D image plane with a linear camera projection matrix P that comprises the intrinsic and the extrinsic camera parameters. In a multiple view setting, the projection matrices of both cameras can be computed from the Fundamental Matrix F that constitutes the epipolar geometry that describes the geometric relationship between multiple camera images. The epipolar geometry is required to reconstruct a 3D point's coordinates out of corresponding images points, which is an indispensable task in the optical tracking pipeline.

Chapter 3

Related Work

The main objective of this part of the thesis is to develop a novel approach to be able to track objects in large, unconstrained indoor environments. Therefore, the tracking system must be capable to cope with ambient interfering lights, infrared radiation, temporary occlusions and even harsh environmental conditions, such as fog and dust. We aim at tracking at large distances with a small amount of hardware to minimize the necessary preconditioning of the tracking environment.

To track objects in space and especially in large volumes, different techniques exist from commercially available products to on-going research prototypes. Extensive research has been performed to develop indoor location systems (ILS) for enabling context-aware applications, user tracking and surveillance [44]. Since this work focuses on positioning in indoor environments, we do not discuss related work based on global navigation satellite systems (GNSS) or tracking solely based on inertial sensors, as inertial measurements suffer from significant drift over time, especially for position estimation. Moreover, we do not incorporate magnetic tracking into the discussion of related technologies, as it is subject to interference from ferromagnetic materials in the tracking volume and magnetic fields generated by other electronic devices, and it is sensitive against conductive materials that are placed near to emitters or sensors. These factors tremendously limit potential tracking environments and making it impractical for our intended test setups. Regarding optical tracking, techniques based on natural features are not reviewed as well, since they require prominent and distinctive structures for pose estimation, as described in Section 2.2.1.1. These distinct features must either be found on the tracked object in an Outside-In scenario, or have to be distributed throughout the volume in an Inside-Out tracking setup. For both scenarios, a reliable feature distribution and an adequate illumination cannot be guaranteed in the intended tracking environments that have been investigated within this thesis, as described in Chapter II.5.

To summarize, the most relevant tracking technologies for the intended wide area indoor environments are radio frequency (RF), ultra-sonic and model-based optical systems. Since they all have advantages and disadvantages regarding accuracy, latency, reliability, scalability and cost, no de-facto standard has been established yet. Thus, we outline state-of-the-art ILS techniques and discuss their advantages and disadvantages.

3. RELATED WORK

3.1 Radio Frequency & Ultra Sound

Radio frequency systems based on Wi-Fi infrastructure or radio-frequency identification (RFID) [34] require a number of readers within the measurement volume to enable object tracking with low latency in large volumes [73]. However, WiFi signals tend to be extremely noisy and signal strength highly depends on surrounding building structures and materials. Thus, precise position estimation cannot be guaranteed even with multiple readers in the volume. In addition, the extensive pre-conditioning of the tracking volume is cost-intensive due to the amount of necessary hardware.

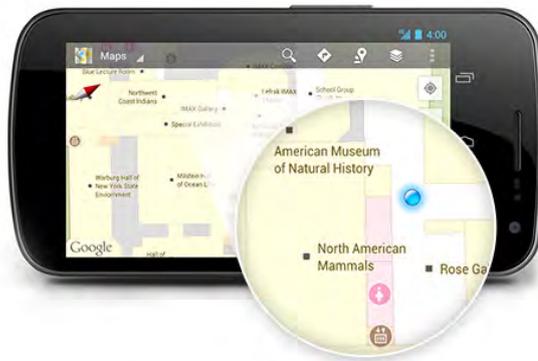


Figure 3.1: Tracking of a smartphone using Google Indoor Maps [156].

Recently, a number of commercially available ILS applications such as Google Indoor Maps [156], SensionLab [164] as well as Indoo.Rs [157] emerged to localize a smartphone (and thus its user) by fusing mobile cellular data, WiFi and inertial measurements to minimize position jitter from WiFi data. Google Indoor Maps that is depicted in Figure 3.1 optimizes the position accuracy by pre-measuring and mapping the signal strength of the WiFi spot within the volume. However, this process takes time before the actual tracking can start. Furthermore, all systems require pre-built indoor floor plans for position visualization and only provide – in best case – several meter accuracy.

Ultra-sonic location systems such as [67, 50] rely on time-of-flight measurement of ultra-sonic signals, calculated using the velocity of sound. Such systems are scalable and can track multiple moving objects. However, current systems offer in the very best case meter-level accuracy under optimal conditions for 3D position estimation [136]. Furthermore, precision and range are not reliable since velocity of sound in the air is highly dependent on environmental conditions, especially humidity and temperature. Especially at long ranges, ultra-sonic systems are often extremely noisy and for that reason not a proper solution for our system’s objectives.

Compared to ultrasound, the RF-based Ultra Wide Band (UWB) technology enables distance measurements without line-of-sight requirements. An example for such a system

is Ubisense [166] that employs TDoA¹ and AoA² measurements between mobile tags and a minimum of four fixed base stations, as shown in Figure 3.2.

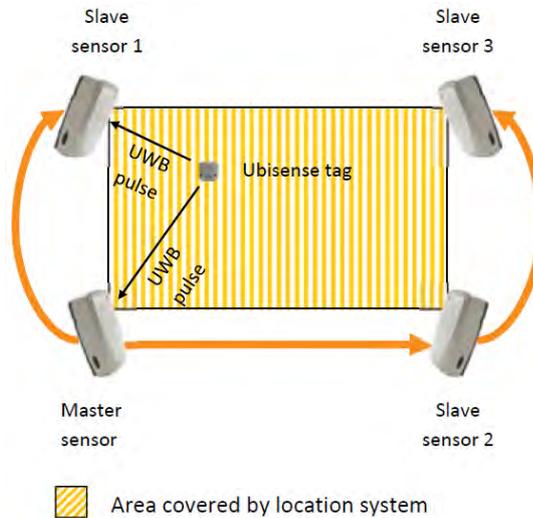


Figure 3.2: A simple four sensor Ubisense system [166].

It offers fast signal speed and hence high sample rates (approximately 135 Hz) and provides an accuracy of down to $0.2m$. The LPM system by Abatec [61] offers a sample rate of 1 kHz with an accuracy down to $0.15m$. It measures the distance between fixed base stations and mobile tags based on the frequency modulated continuous wave principle [4]. Although large distances can be covered, the ultrasound and RF-based systems are expensive and the resulting accuracy is not sufficient for precise user tracking for virtual reality applications.

3.2 Optical Tracking

Model-based optical tracking systems require the target to be within the line-of-sight of one or more cameras to estimate its 3D coordinates from the 2D image-projections, as described in Chapter II.2. It is robust against magnetic, electric and acoustic interference and works with light-emitting (active) or retro-reflective (passive) targets.

One camera is sufficient for tracking in an *Inside-Out* scenario that is intended for the InterSense IS 1200 system [151]. It offers a scalable, cost-effective solution for wide area tracking as it fuses optical tracking of planar bitmap patterns (see Section 2.2.2.1) with inertial measurement data. Therefore, an inertial measurement unit is combined with a single camera and attached to the trackable object to observe passive markers that have to be distributed throughout the volume. While this setup offers high updates rates with very low latency (max. $8ms$) it requires sufficient illumination and a large

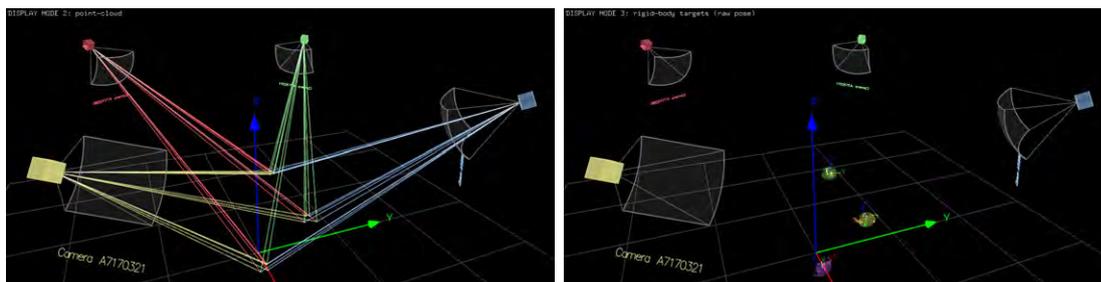
¹Time-Difference-of-Arrival

²Angle-of-Arrival

3. RELATED WORK

number of targets that have to additionally be in close range to the camera to ensure robust tracking. These prerequisites make this system impractical and even impossible to apply for our intended environments. As the implicit nature of Inside-Out tracking requires well-distributed visual features throughout the volume, it can be concluded that using active targets would also not be a sufficient approach for our research objectives since it would violate the goals of omitting pre-conditioning of the environment and of minimizing the necessary amount of hardware components.

Outside-In optical tracking systems require the target to be within the line-of-sight of two or multiple cameras. In the following, a number of state-of-the-art Outside-In model-based tracking systems are presented.



(a) Calculating the 3D position of each point (b) Estimating the pose of each target

Figure 3.3: Multiple target tracking using iotracker with 4 cameras, [84].

The near infrared (NIR) spectrum based systems, such as Vicon [145], A.R.T [132] or iotracker [84, 141] offer (sub)-millimeter accuracy in standard room sized environments ($4 \times 4 \times 3m$) and provide tracking of multiple targets with very low latency, as depicted in Figure 3.3. To enlarge the tracking volume, those systems increase the number of employed cameras (up to 50 in A.R.T). However, this causes a tremendous growth of costs and setup complexity. The PPT-E system [146] is able to cover areas up to $20 \times 20m$ with a minimum of four cameras but sub-millimeter tracking accuracy is guaranteed only for volumes up to $3 \times 3 \times 3m$. No accuracies are provided for larger volumes.

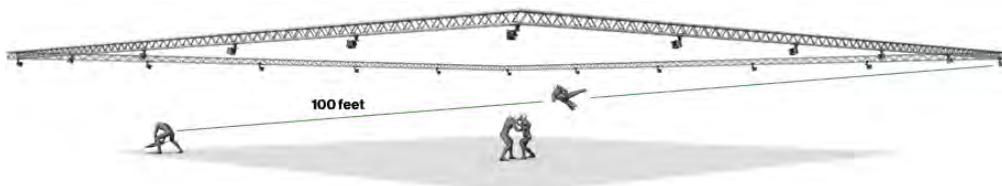


Figure 3.4: A tracking setup using the Prime41 system, [138].

The Prime41 system [138] offers multiple user tracking by detecting passive targets up to $30m$, using a perimeter setup with multiple cameras, as shown in Figure 3.4. However, no further details on accuracy nor the number of cameras are given to cover this volume. Furthermore, as the most cost efficient systems of the above mentioned, one Prime41

3.3 Laser Measurement Systems

camera still costs about €5000. A minimal 4-camera perimeter setup results in pure camera costs of €20.000 (without software), which is a multiple of our complete system costs.

For tracking in larger, unconstrained indoor environments, such as tunnels and mines, examples of application of optical tracking systems are rare and only exist for highly special measurement purposes. As depicted in Figure 3.5, one example is the application of a hand-held digital camera in combination with fixed installed visual markers for monitoring tunnel wall displacements by close-range photogrammetry [29, 129]. The system requires huge installation effort and therefore is not practical for daily application. A further example is the use of a tracking camera and retro-reflecting targets to track the relative position between two shields of a double shield tunnel boring machine as part of a guidance system. The system is in use in several tunnel projects and reported to function properly [135]. However, both optical tracking systems are not designed to simultaneously track several targets over longer distances in real-time.

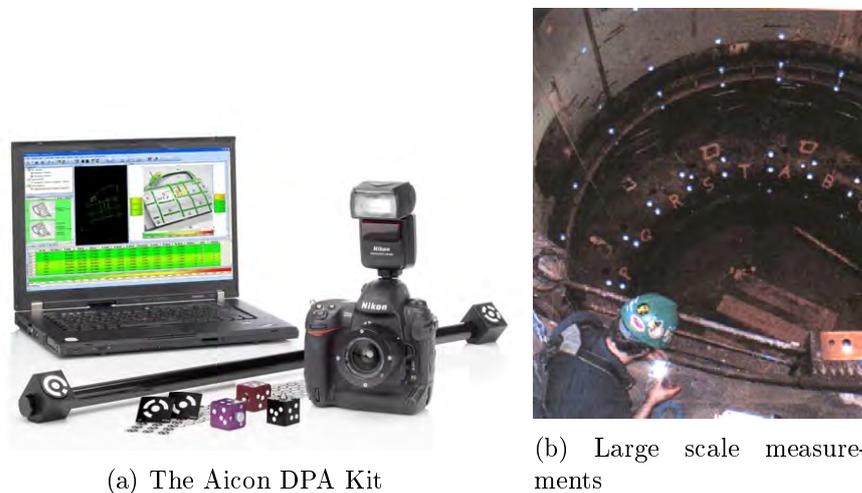


Figure 3.5: The AICON DPA-Pro System, [129].

Summarizing, existing Outside-In optical systems rely on artificial features for model-based tracking and are thus robust against environments with non-distinctive geometric structure and poor illumination. However, for wide area tracking they require a complex system setup and thus are cost intensive. Furthermore, existing NIR tracking technology remains to be highly sensitive to ambient interfering lights and infrared radiation, especially during camera calibration, making those systems incapable of being deployed in unconstrained indoor environments.

3.3 Laser Measurement Systems

For determining the 3D position of objects with very high accuracy, classical surveying methodology such as laser measurement systems are widely applied in research and

3. RELATED WORK

industry. The employed instruments (total stations, terrestrial laser scanners and laser trackers) simultaneously measure the horizontal and vertical angle to the target-point together with the slope distance by using laser distance measurement. Based on these polar observations, the 3D coordinates of the target-point are then processed. Depending on the specific surveying task, the target-point is either a geodetic prism or a non-signalized point, directly located on the object surface (reflector-less measurement). The most frequently used instrument type is the total station [91, 100]. In the application field, it can be found manually operated as well as integrated in automatic measurement and mobile multi-sensor systems. Advanced total stations have the capability to automatically search for, recognize, measure and even lock a prism, thus, are able to follow a slowly moving object. These options are primarily used to facilitate manual operation, increase speed of work and are indispensable when kinematic surveying is to be performed. Total stations are highly accurate for large distances of $100m$ and more. They are used for setting out, network measurements, tunnel heading control, machine guidance and displacement monitoring. However, specialized personnel are required for instrument control and several (kinematic) visual objects cannot be simultaneously sighted and measured.

The technology of laser scanning by use of Terrestrial Laser Scanners (TLS) [13] is also broadly common in underground construction. It is routinely applied for a variety of purposes, such as tunnel profile control, volume determination and check of tunnel surface quality [77, 89, 90]. Recent research work [120] aims to use the technology for monitoring of tunnel wall displacements. As with total stations, laser scanners are operated either manually or automatically when integrated in tunnel laser scanning systems. They can perform static and kinematic scanning. However, the technology requires extensive post-processing of 3D point clouds and does not allow for efficient measurement of defined points or objects with low latency. So far, the technology does not provide real-time capability.



Figure 3.6: Leica Absolute Tracker AT901 with T-Probe, [148].

Recently, Leica Geosystems introduced an approach that integrates an optical tracking system with a laser tracker [93, 158]. It offers automatic lock-on and tracking of the

3D position (by the laser tracker) and 3D orientation (by the optical tracking system) of a hand-held target [159] with high precision and low latency up to $18m$. The system is shown in Figure 3.6. As a portable system, it is designed for industrial applications (e.g. prototyping and reverse engineering, tooling inspection and part mating, positioning and aligning of machines). By using a special corner cube reflector, the range can be extended up to $160m$ but only for the laser tracker, not for the optical tracking system. However, up to now, this system is only used for very particular measurement tasks in tunnel construction. The only example of regular use is the check of tunnel segment geometry, a daily task performed in the segment factory. Up to now, laser trackers cannot be found underground as they are expensive and not considered robust enough to operate in harsh environments. Besides, they cannot simultaneously track multiple targets.

Chapter 4

Methodology

To overcome the limitation of existing optical tracking systems, as described in Section 1.1, a robust wide-area *Outside-Looking-In* optical tracking system for position tracking is described that requires only two cameras to track targets up to distances of 30 – 100m, depending on the tracking task. It provides high tracking accuracy while being robust against interfering lights during calibration and tracking.

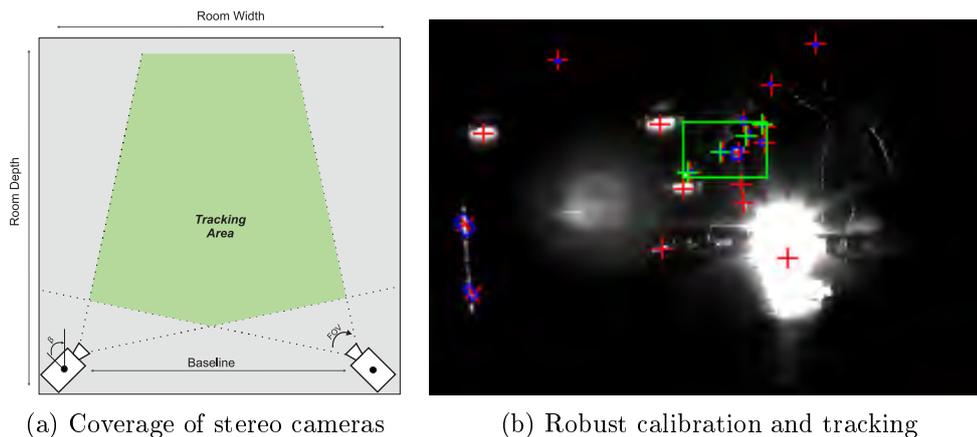


Figure 4.1: Key properties of the proposed optical tracking system.

In Figure 4.1, the properties and capabilities of the proposed system are shown. Figure 4.1a illustrates the system’s hardware setup and the resulting tracking coverage. Figure 4.1b depicts a successfully detected target of our system in the camera image that can be subsequently employed for calibration and tracking. The tracking is achieved despite heavy interfering lights as they might occur in an unconstrained indoor environment. By heavily minimizing the amount of necessary vision hardware, the system is highly cost effective and easy to set-up. Although current infrared optical tracking systems lack the capabilities of robust wide-area 3D position estimation, the underlying technology is very promising since it offers high precision with very low latency. Therefore, we heavily extend this technology to overcome limitations in terms of distance coverage, sensitivity

4. METHODOLOGY

in harsh environments and the amount of simultaneous trackable targets.

4.1 System Requirements

To achieve the research objective from Section 1.2, the following requirements were specified to be fulfilled by the tracking system:

Cover Wide Tracking Volume: Target(s) shall be tracked with two cameras up to distances of $100m$. To account for varying real-life tracking scenarios, the distance between both cameras (baseline) may vary. Both cameras are connected to one processing unit, thus data exchange interfaces are required that support long distance cable transmission.

Accurate Camera Calibration: To optimally compensate optical aberrations, the intrinsic and extrinsic calibration must be able to be performed with the complete camera encasement. The extrinsic calibration has to be capable to be performed during on-going activities in the tracking volume and thus must be able to cope with heavy interferences.

Unique Target Identification: Interfering light sources must be filtered to allow for a robust target detection during calibration and tracking, as illustrated in Figure 4.1b.

Continuous & Accurate 3D Position: The hardware and software algorithms have to ensure precise target detection at large distances and in environments with poor visibility due to particles (dust, dirt) in the air. Continuous 3D position estimation must be provided within the whole tracking volume.

Robust Hardware Casing: To ensure system reliability in real-life environments, hardware components (cameras, lenses, target, processing unit) have to be encased to be dust- and dampness proof. Nevertheless, the system must be easy and quick to setup and the target should be usable even with thick gloves. Furthermore, side effects on the camera's field-of-view (FOV) as well as optical aberrations must be considered when encasing the vision parts of the system.

4.2 Evaluation of Target Visibility

Based on the system's requirements, a preliminary study was conducted [94] to define an appropriate hardware setup to perform wide area 3D position measurements in varying, unconstrained and even harsh indoor environments using infrared optical markers. Therefore, we firstly took into account our previously outlined factors that influence accuracy of 3D position estimation as well as tracking performance (see Section 2.1.1.2) to derive a test hardware setup. Next, we practically evaluated combinations of target types and camera setups to determine obtainable tracking distances. To be able to

perform tracking in even harsh indoor environments, we performed measurements in a tunnel during on-going construction to evaluate the best operating distances when using (1) passive or (2) active markers as targets (see Figure 2.2). Furthermore, we tested different object lens configurations to determine an optimal balanced optical setup for the intended tracking system. Therefore, we tested different focal lengths during the distance measurements. An optimal configuration has minimal optical aberrations while providing high light throughput and a sufficient field-of-view (FOV) to cover the intended tracking volume. Lenses with short focal length have stronger optical aberrations characteristics but provide larger FOVs than lenses with longer focal lengths. Furthermore, the optics system must provide sufficient depth-of-field (DOF) to ensure that the target appears sharp in an image taken within the intended tracking volume. DOF increases as focal length and aperture decreases. However, since accurate 3D position estimation relies on robust blob centroid computation, pixels of the target blobs ideally are bright and clearly distinguishable from the surrounding pixels. For that reason, large aperture must be employed to provide maximal light throughput emitted from distant targets.

4.2.1 Test Setup

A high resolution machine vision camera (1/1.8" Mono CCD, $1624 \times 1224 px$) with a vari-focal lens (focal lengths $f = 12 - 36mm$, aperture = $F2.8 - 16$) and a long-wave pass filter, placed in front of the camera, was selected. As passive target, retro reflective foil targets in combination with a $850nm$ illuminator were employed. The active target comprised an infrared light diode with a peak wavelength at $850nm$ and a viewing half-angle of 23° .

4.2.2 Test Results

Test images have been captured with $8bit$ pixel depth at distances of $30m$, $50m$ and $70m$, employing open aperture ($f/2.8$). We defined a blob to be robustly detectable if it features 80%-100% of the maximal luminance [84].

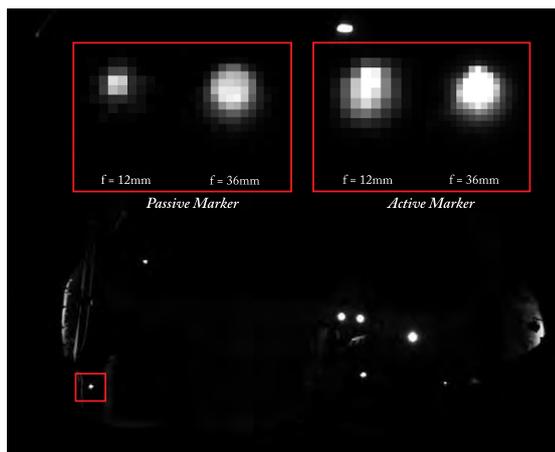


Figure 4.2: Blobs at $50m$ distance with minimal/maximal focal length of $f = 12 / 36mm$.

4. METHODOLOGY

As illustrated in Figure 4.2, passive as well as active targets were robustly segmented in the camera image up to a distance of $50m$. The diminished blob’s brightness of the passive target is well illustrated in Figure 4.2. For testing, we manually increased shutter speed and gain of the camera. The brightness of the passive target increased but likewise did image noise. This should be avoided to provide accurate feature segmentation. Furthermore, brightness of other light sources or reflective material (i.e. construction vests) has increased as well. This can result in blooming (and hence tracking loss) of the target when getting in close range to these interfering areas. At a distance of $70m$, blobs of passive targets could not be robustly detected while active targets were still visible and could be accurately segmented despite dust and dirt in the air.

Consequently, active targets are suitable to fulfill the proposed system’s objectives. A focal length of $25mm$ proved to allow the best balance between optical aberrations, sufficient blob brightness (and thus accurate feature segmentation and 3D estimation) and adequate FOV as well as DOF to cover the entire intended tracking volume with objects being in focus.

4.3 Methodological Approach

The overall system’s work-flow is depicted in Figure 4.3. The projective invariant properties of the target model are trained and subsequently employed for 2D model recognition during calibration and tracking. Hence, the same target model can be used to perform extrinsic calibration and an additional calibration apparatus can be avoided.

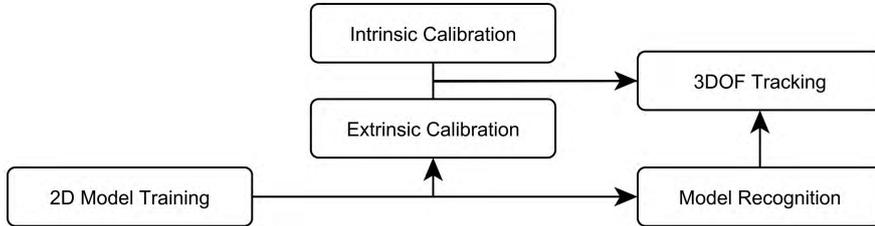


Figure 4.3: Overview over the system’s workflow.

4.3.1 Vision System

The vision component of the proposed tracking system comprises two cameras, lenses and filters. Following our preliminary study, we derived an optimal balanced optical setup (sensor size, focal lengths, aperture) for the intended tracking volume that minimizes optical aberration and rasterization effects while providing a sufficient field-of-view (FOV) as well as depth-of-field to cover the intended tracking volume with objects in focus. The coverage depends on focal length f , the distance between the cameras (baseline) as well as the amount of yaw-rotation β of each camera, as depicted in Figure 4.4.

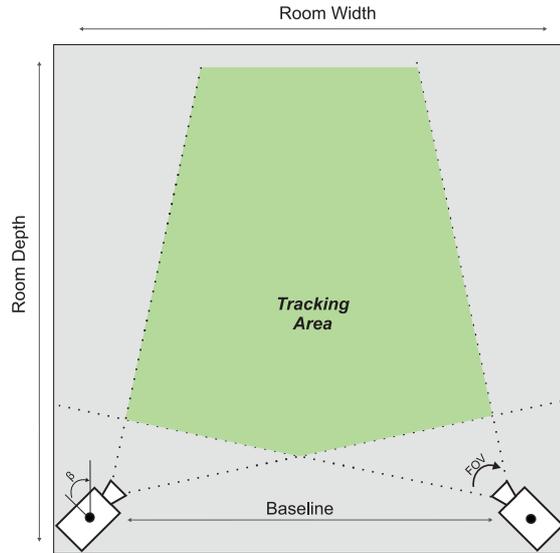


Figure 4.4: Coverage of stereo cameras

Our system uses high-resolution machine vision cameras in combination with low-distortion lenses that feature large aperture and minimal optical aberrations, as described in Section 4.4.1. The high quality cameras provide low heat evolution and large image sensors yielding little sensor noise, so jitter in the camera image can be minimized. Together with high resolution image sensors, precise segmentation can be provided even at long distances. The cameras offer high global shutter speed to allow for low-latency tracking and to minimize motion blur when the target is moving fast. Both cameras form a *Stereo Camera Rig* and are shutter-synchronized by an external trigger signal to guarantee temporal synchronous image pairs. To enhance robust target identification, a long-wave pass filter is inserted into the optical path to ensure light transmission only in the NIR spectrum. To provide wide area tracking in width and depth, the baselines can heavily vary in the intended tracking environment. Thus, we propose to use the GigE Vision standard [130] to guarantee lossless image transmission while providing long cable lengths. Both cameras are connected to one workstation for image processing and tracking.

4.3.2 Target Design Guidelines

The geometric constellation of our target design constitutes a line approach, as illustrated in Figure 4.5.

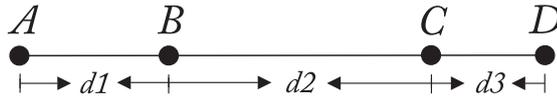


Figure 4.5: The 2D model design features projective invariant properties.

4. METHODOLOGY

It consists of four collinear optical markers which are attached in fixed distances d_1 , d_2 , d_3 to each other; thereby, cross-ratios and their projective invariant properties can be exploited for robust target identification, as described in Section 4.3.4, as well as occlusion recovery can be performed, as explained in Section 4.3.6.

Within the whole intended tracking volume, the target must be reliably visible in the cameras' images to ensure robust feature segmentation. As described in Section 2.2.1.2 and illustrated in Figure 2.2, two common types of artificial features exist that can be applied to targets for infrared optical tracking. Since precise feature segmentation in scenarios with interferences as well as at large distances can only be assured using active markers, we use infrared light emitting diodes (IR-LED) as optical markers. To protect the IR-LEDs and to prevent optical aberrations (flare artifacts on the blob edges in the camera images), each IR-LED is covered with a translucent diffuse plastic sphere, as shown in Figure 4.6.

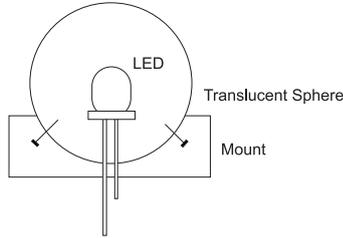


Figure 4.6: LED is coated with a translucent diffuse plastic sphere.

With this simple design, multiple unique constellations can be easily designed to simultaneously track multiple targets in the same tracking volume. Existing 3D rigid body targets (e.g [95]) also offer permutation invariant geometric constellations to track multiple targets. However, our line approach has three advantages over 3D targets that are crucial for our intended research goals.

1. We can re-purpose the tracking target as calibration apparatus by detecting the two outermost IR-LEDs during extrinsic camera calibration. Thereby, the amount of necessary hardware for setup and maintenance can be reduced.
2. Even during calibration, the target can robustly be tracked despite interfering lights, since the 2D characteristics of the target allows for *Model Fitting* already in the image domain instead of in 3D space, as it is common in competing approaches [84, 145, 132].
3. Fixing the IR-LEDs in a 2D manner increases the physical robustness of the target against accidental breaking off when touching the target during usage; this is especially an issue for tracking at larger distances since the target requires enlarged dimensions as well. Accidental breaking off is a common problem with the sensitive 3D rigid targets (see Figure 2.5) that need frequent replacement or repair by experts.

4.3.3 Calibration

As described in Section 2.3, the camera’s intrinsic and extrinsic parameter must be known to perform precise feature segmentation and to provide 3D point reconstruction of the target model’s IR-LEDs. Due to the large baselines and the intended range of our proposed tracking system, a 2D or 3D calibration apparatus is not applicable for both intrinsic and extrinsic parameter estimation. Following the calibration guidelines from Section 2.3.4, we split internal and external calibration into two separate steps, estimating the intrinsics with a planar calibration target ($2D$ feature) and the extrinsics based on points ($0D$ feature).

4.3.3.1 Intrinsic Calibration

The Camera Calibration Toolbox [133] was used for intrinsic parameter estimation; it requires a 2D planar chessboard pattern for determination of the *Camera Calibration Matrix* K (see Section 2.3.2.5). To enhance the estimation of the parameters, all optical components (camera with lens and filter) of the final tracking setup should be included in the calibration procedure. However, with such a setup, a normal black/white chessboard pattern would not be visible in the camera image. Therefore, we extended the standard intrinsic calibration setup by developing a chessboard plane made of retro-reflective foil that is illuminated with an infrared light source to provide chessboard images in the NIR¹ spectrum. The complete intrinsic setup is illustrated in Figure 4.7.

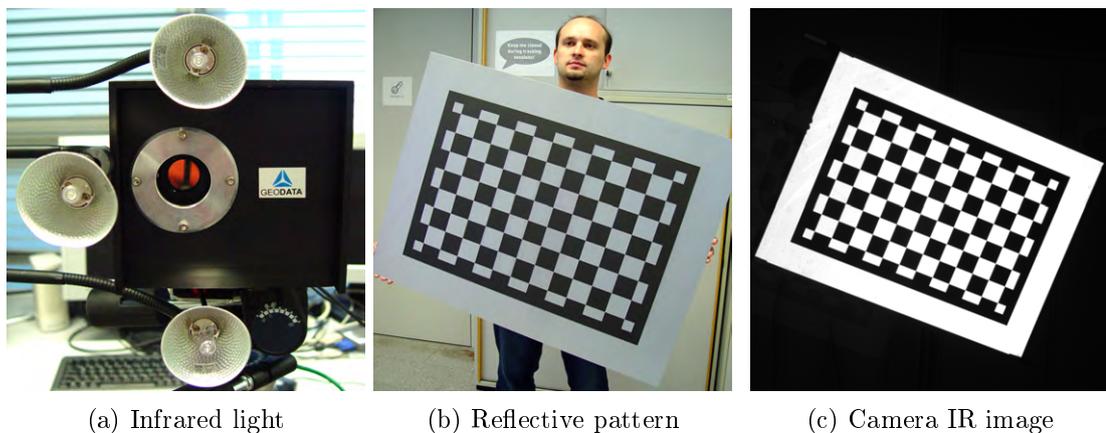


Figure 4.7: Intrinsic camera calibration with a retro-reflective pattern.

Since the lens configuration must not change after intrinsic calibration and the tracking will be at large distances, the focus settings are set to unlimited that results in a blurred pattern at close ranges. With this setting, the images of the tracking system’s cameras and lenses (see Section 4.4.1) are in focus from $4m$ onwards. Thus, the pattern must have a sufficient size to cover the entire camera image at a distance of $4m$. Further-

¹Near Infrared

4. METHODOLOGY

more, the sharpness of calibration images was enhanced by closing the aperture ($f/8$) for increased depth of field.

4.3.3.2 Extrinsic Calibration

After the tracking system with its two cameras is physically set up, the geometric relation between the cameras is estimated by the extrinsic calibration process, yielding the definition of the two *Camera Projection Matrices* P, P' (see Section 2.3.3.3). As described in Section 2.3.4, calibration apertures of varying dimensionality can be used for extrinsic parameter estimation. Toolboxes such as [133, 140] estimate (P, P') by using a 2D pattern. For our calibration scenario, such a pattern would have to be extremely large to be visible at distances of $10 - 70m$ while being planar to provide precise corner extraction. Furthermore, its surface would have to be composed of retro-reflective foil, which is sensitive and requires additional hardware for pattern illumination. Such a target would neither be transportable nor suitable. Therefore, we exploit methods that use $0D$ features (points) for Fundamental Matrix F estimation as extrinsic calibration. Auto-calibration approaches that are based purely on natural features [102, 116, 99, 125] are not applicable since they require well-distributed features throughout the entire tracking volume to function robustly. This can be easily true in cluttered and well-illuminated environments but is hard to achieve in rather dark environments or scenarios with little geometric structures. Re-using existing light sources or reflectors require manual selection of correspondences in each image and a fair distribution cannot be guaranteed as well. Hence, this approach is omitted as well.

Using Artificial Points for P-Matrix Computation The calibration approach of this thesis thus follows the idea of using artificial points that are created by manually waving the calibration target through the volume to achieve a high amount of detectable features. To allow for calibration in unconstrained environments with interfering lights, methods using a single point [45, 69, 84, 141] are not sufficient. Those approaches, as depicted in Figure 4.8, require the background to be trained and to manually mask interferences in the camera images to avoid false positive feature correspondences; obviously, those techniques cannot cope with moving interfering lights.



Figure 4.8: Trained background (left) and manual masking (right), [141].

The stereo camera calibration approach [48] tries to overcome this limitation by evaluating the screen-space coordinates of two blobs – that corresponding physical markers have a known distance – over a sequence of camera images (see Figure 2.14b). To find the image correspondences, the algorithm seeks for the two longest paths of possible marker motion in each camera image and assumes that no other reflections or markers are moved through the entire working volume in a similar manner as the calibration apparatus. Using the corresponding image points, the approach estimates the Essential Matrix E by performing the *Nominalized 8-Point Algorithm* (see Sections 2.3.3.3 and 2.3.4). While being more robust against interferences than the single point approaches, this method has another advantage. The affine transformation to obtain real-world distance units [mm] is not only computed once (as in existing approaches such as [69, 84]) and which can result in inaccurate tracking at larger distances, but takes into account the measured distance between both optical markers of each processed camera frame. The scale is then obtained by

$$scale = \frac{d_{real}}{d_{mean}}, \quad (4.1)$$

where d_{real} is the real known distance between the two markers and d_{mean} is the mean distance calculated based on all measured distances between the two markers over all observed image frames. This scale is then applied to the Equation 2.24, re-formulating t by

$$t_{metric} = t \cdot scale, \quad P' = [R|t_{metric}]. \quad (4.2)$$

However, as described above, certain criteria must be fulfilled for correct functioning of [48]. To expunge any assumptions of marker movement, to allow short tracks or even point pair correspondences without any spatial connections to each other, we developed a pipeline that extends the approach of [48], as illustrated in Figure 4.9.

A line target, as described in Section 4.3.2, is used as calibration apparatus. Since its pattern can be recognized in a 2D camera image, no epipolar geometry is necessary to provide correct point correspondences for the estimation of E . During calibration, interferences are filtered and the target is identified (*Model Identification*) using the developed pipeline from Section 4.3.4. This pipeline returns a set of four ordered points p for each camera L and R of a frame at time t .

$$S_L^t = \{p_{L,1}^t, p_{L,2}^t, p_{L,3}^t, p_{L,4}^t\}, \quad S_R^t = \{p_{R,1}^t, p_{R,2}^t, p_{R,3}^t, p_{R,4}^t\} \quad (4.3)$$

where $p_{L,i}^t, p_{R,i}^t \in \mathbb{R}^2, i = 1 \dots 4$.

Although the model fitting is reliable, the matching in each image is still independent from each other. Thereby, errors can occur such as a false positive identification in camera 1 and a hit in camera 2, or a hit in camera 1 and no detection in camera 2 (due to occlusions). Such erroneous input data would decrease the stability of the estimation of E and thus should be avoided. Therefore, a *Similarity Check* between both sets S_L^t, S_R^t is performed. It is based on the idea, that the detected target has a similar

4. METHODOLOGY

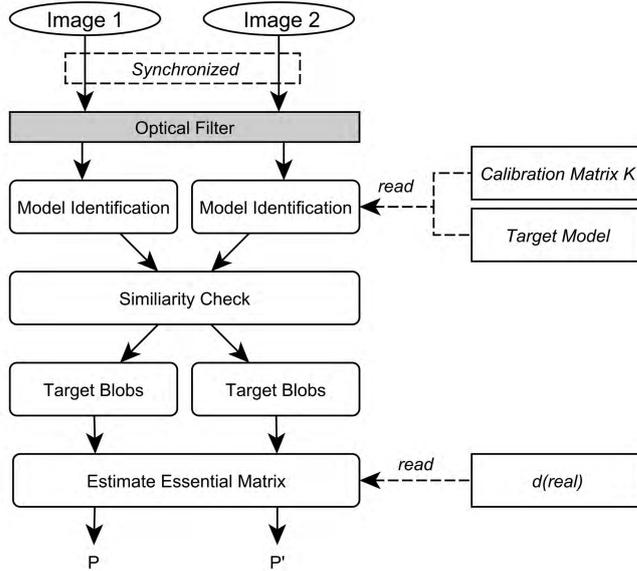


Figure 4.9: Extrinsic calibration pipeline.

orientation in both images at time t up to a threshold, depending on the camera setup. For the similarity evaluation, the target in the left image is considered as a vector $\vec{v}_L = \overline{p_{L,1}, p_{L,4}}$, respectively \vec{v}_R in the right image. The angles (ϕ_x, ϕ_y) between \vec{v} and the x-axis, respectively the y-axis, are determined for the left and the right image. Outliers are detected if the angles differ by more than a given threshold λ , as in Equation 4.4. The same is done for the y-axis. Thereby, the algorithm can be used on images taken from both horizontally and vertically aligned cameras.

$$outlier = \begin{cases} \vec{v}_L, \vec{v}_R & \text{if } |\phi_{x,L} - \phi_{x,R}| > \lambda \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

If outliers have been detected, the point sets (S_L^t, S_R^t) are rejected, if not, the sets are considered as correct target blobs and are fed into the calibration routine of [48]. Since K is known from Section 4.3.3.1, the *Normalized 8-Point Algorithm* is applied for computation of the Essential Matrix E to enhance the stability of the epipolar geometry estimation [56]. To obtain a metric scale for Equation 4.2, the distance between the two outermost IR-LEDs of the calibration target are measured to sub-millimeter accuracy with a high precision total station, yielding d_{real} . The resulting world coordinate system is illustrated in Figure 4.10.

With our described pipeline, we achieve a robust calibration procedure that can be performed in the presence of static and moving light sources. No pre-conditioning of the volume is necessary and background training as well as manual masking can be omitted, which increases the system's ease of use during setup and maintenance. Furthermore,

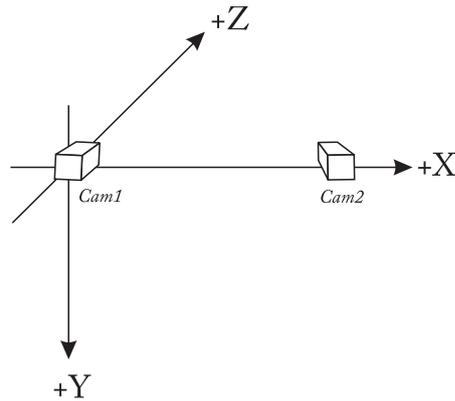


Figure 4.10: Resulting camera coordinate system for tracking.

by re-using a tracking target for extrinsic calibration and scale estimation, additional equipment can be minimized.

4.3.4 Interference Filtering

To provide robust target identification at each stage of a optical tracking system workflow (extrinsic calibration, target tracking), static and moving interfering lights must be robustly filtered out. In unconstrained tracking environments, as described in Section 1.1, a varying number of ambient light sources (wall illumination, spot lights, reflections, vehicle lights, ...) might exist.

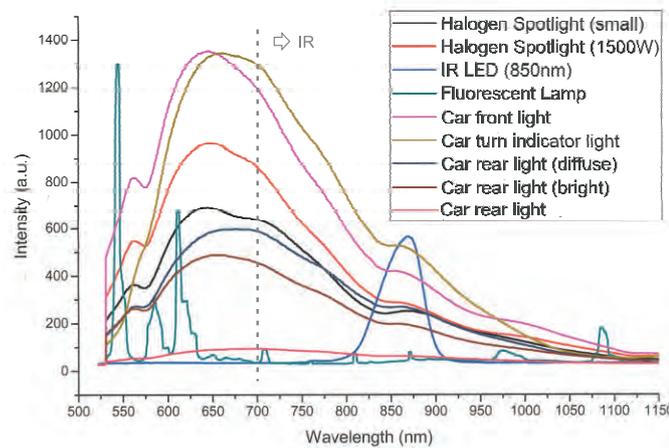


Figure 4.11: Wavelengths of various light sources.

To evaluate the wavelength emission, we measured frequently occurring standard illumination sources with a spectrograph. Their emission curves are illustrated in Figure 4.11. As depicted, almost all ambient light sources show infrared radiation. A portion

4. METHODOLOGY

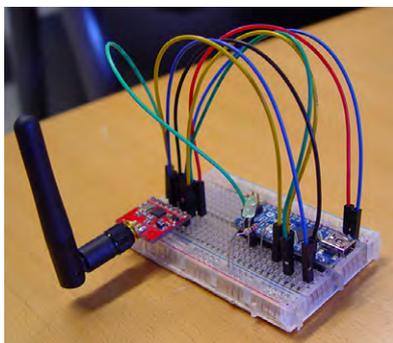
of the interferences can be filtered by inserting a longwave pass filter with a cut-on value of $780nm$ into the optical path. However, most of the interfering lights are still visible in the camera images and result in bright circular blobs, similar to the IR-LEDs from the target model.

To robustly detect the target amongst static and moving interfering lights, we investigated different concepts based on hardware and software filtering that are presented in the following.

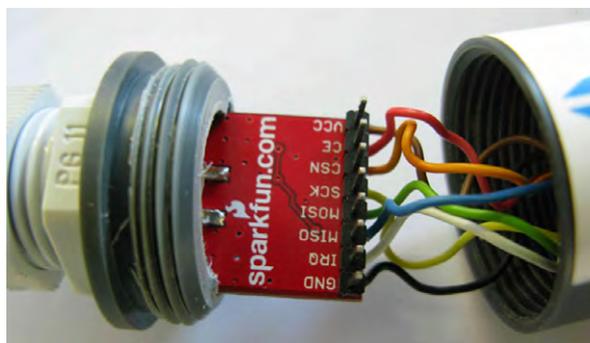
4.3.4.1 Hardware-based Target Identification

The main idea of the presented hardware-based filtering approaches is to detect the blobs of the target without requiring the knowledge of the target's geometric structure. Thereby, also point-based targets, consisting of a single LED, can be robustly segmented and tracked.

The first concept aims at changing the target's LED state (on/off) in two subsequent frames. This can be accomplished by remotely controlling the LEDs via a wireless communication. The difference in luminance in both frames can then be evaluated (*Luminance Filtering*) to robustly detect the LED's position. To change the LED's state, we first evaluated a number of wireless communication technologies, such as RFID [70], ZigBee [155] and radio chips in the GHz band. Due to its low price, high data throughput and small form factor, the 2,4GHz Nordic nRF24L01+² chip was finally chosen for wireless data transmission. The target control unit consists of the open-source platform Arduino Nano 3.0 [131], that features a ATmega328 micro-controller, and the circuit to interface with the Nordic nRF24L01+ radio chip.



(a) First prototype of receiver



(b) Receiver integrated in tracking target

Figure 4.12: Radio module for target communication for luminance-based filtering.

To extend the radio frequency reception range, the radio chips for both base station and target are equipped with 2.4GHz dipol-antennas with a power gain of 5 dBi³ and 2.2

²NordicSemiconductor: <http://www.nordicsemi.com>

³dBi: decibels-isotropic

dBi, respectively. Thereby, a communication range of $120m$ with an estimated round-trip-time of $5ms$ can be provided. In Figure 4.12, the development process of the radio module is shown.

Due to the implicit nature of the radio connection, LED state changes and image capturing cannot be precisely synchronized in time by a hardware trigger. For that reason, we further investigated a concept of filtering interfering lights by applying wavelength filtering using a motorized filter unit. As luminance filtering, it aims at detecting the target's blobs without requiring a predefined and well-known target geometric structure by evaluating two subsequently captured frames. Since a single infrared longwave pass filter inserted in the optical path of the camera removes only those parts of the ambient illumination that solely emit in the visible light spectrum, we used a motorized filter-wheel to be able to change the applied filters at run-time.



(a) A motorized filter wheel [153]

(b) The wheel fitted into the casing

Figure 4.13: Using a motorized filter wheel for wavelength-based filtering.

The employed filter wheel⁴ is therefore equipped with two filters, a shortwave pass filter (VIS) to transmit all wavelengths shorter than the cut-off length of $780nm$, and a longwave-pass filter (IR) to transmit all wavelengths longer than the cut-on length of $780nm$. A stepper motor⁵ is used to control the filter wheel and is connected to the workstation over USB 2.0. With this setup, the change time between two adjacent filters is $200ms$. To robustly couple the filter wheel with the camera, a solid casing was designed that fixates the wheel with a customized apparatus in front of the camera. In Figure 4.13, the filter wheel and the developed camera encasement are depicted.

To access both radio and filter wheel to control the LED state and to change between filters during run-time, a software module was developed, as depicted in Figure 4.14. As illustrated, the software processing accesses either the radio- or the wheel control to change the LED's state or the employed filter between two subsequently captured

⁴Thorlabs Motorized Fast-Change Filter Wheel FW103S/M

⁵Thorlabs T-Cube TST001

4. METHODOLOGY

frames I_k, I_{k+1} . For luminance filtering, I_k is captured and recorded while the LEDs are switched on, I_{k+1} with LEDs are switched off, respectively. For wavelength filtering, I_k is recorded with IR filter, I_{k+1} with VIS filter, respectively.

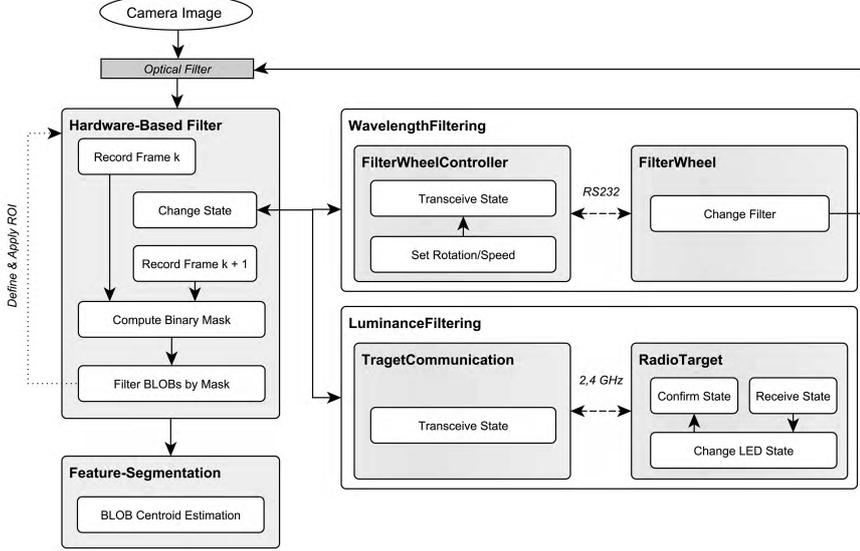


Figure 4.14: Pipeline to detect target features using hardware-based filtering.

Thereby, the LEDs are only visible in I_k for both filtering approaches. In the next step, a binary filter mask $M \in \{0, 1\}$ is computed. Therefore, I_k, I_{k+1} are converted to binary images with a given threshold α to filter noise and areas with low luminance, resulting in $m \times n$ matrices $B(k), B(k+1) \in \{0, 1\}$. To identify the LEDs in $I(k)$, a negated pairwise *Logical Implication*⁶ is applied to each element i, j of $B(k), B(k+1)$, as denoted in Equation 4.5.

$$M_{ij} = \neg(B(k) \rightarrow B(k+1)) := \neg(\neg b_{ij}(k) \vee b_{ij}(k+1))_{i=1, \dots, m; j=1, \dots, n} \quad (4.5)$$

Thereby, only the areas that show the LEDs are marked in M with a *logical true*. The mask is then applied to I_k to segment the LED's blobs and to define the region-of-interest (ROI) that is subsequently used for binary mask processing. Finally, the blob's centroids are computed using the feature segmentation algorithms from Section 4.3.4.2.

Our initial tests indicated promising results for both approaches to detect static LEDs in the presence of ambient interfering lights. However, as soon as the target was quickly moved a robust LED detection could not be provided due to the latency introduced by the round trip of the radio connection and by the time the wheel requires to change between two adjacent filters. To reduce the latency for wavelength filtering, a high current stepper motor or even a multi-spectral camera would be an interesting option to provide robust

⁶Logical implication is also known as *Material conditional* or *Logical conditional*.

target tracking of a single LED or of a multi-LED target without requiring to evaluate the target’s geometric properties. These ideas are subject of future research.

Based on the initial evaluation, we decided to perform tracking solely using a software-based target identification pipeline to be able to provide fast, reliable and robust tracking of static and moving targets. This pipeline is presented in the next section and is employed throughout the following chapters of this part.

4.3.4.2 Software-based Target Identification

To robustly detect the target amongst static and moving ambient interfering lights, we developed a software-based identification pipeline, as depicted in Figure 4.16. It is built around a 2D model fitting approach that exploits the permutation and perspective invariant properties (see Section 2.2.2.1) of the target design (see Section 4.3.2). The target model must be thus trained once before it can be recognized during calibration and tracking.

Model Training To obtain the unique properties of a target pattern, it is trained once in an off-line process to determine its *Model*, as illustrated in Figure 4.15.

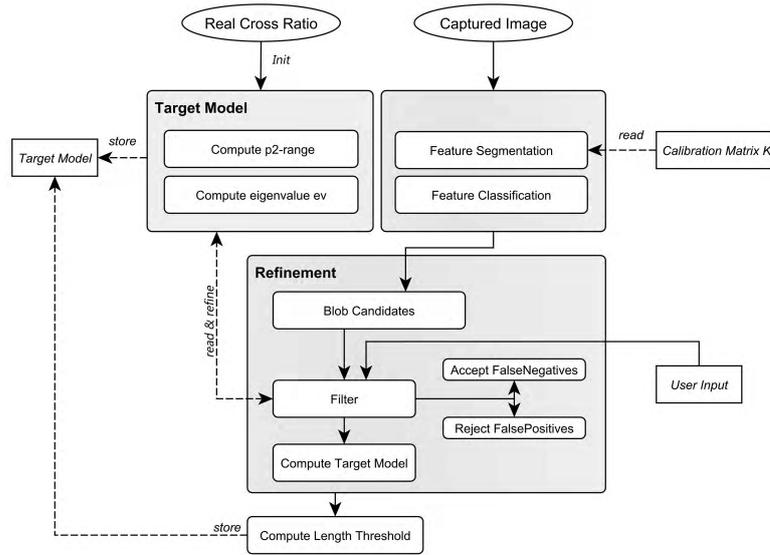


Figure 4.15: Pipeline to obtain the target’s model.

The following steps are performed:

1. The distances $d1, d2, d3$ (see Figure 4.5) between the target’s LEDs are precisely measured using a total station.
2. Based on $d1, d2, d3$, the cross ratio λ is computed and used as input argument for the function J to obtain an initial estimate for p_{range}^2 . The eigenvalue of the

4. METHODOLOGY

moment matrix M as a measure for collinearity is set to an initial value such as $ev \neq 0, ev < 0$.

3. The target is captured at all intended tracking distances to obtain a sufficient number of samples (images) for the complete tracking volume.
4. Each of the captured images is then processed and blob candidates are obtained by performing feature segmentation and classification (see Figure 4.16). p_{range}^2 and ev are applied to the blob candidates and subsequently refined to account for noise of cross ratio and collinearity.
5. After the refinement phase, the minimum and maximum length of the target in the 2D images over all images are measured to obtain a threshold th_{range} .
6. Finally, the obtained model is stored, containing p_{range}^2 as the minimum and maximum values of the pattern's p^2 -invariants, ev , as the collinearity error model and th_{range} .

Model Identification After a new image (*frame*) is captured from the camera with the attached long-wave pass filter, all blobs are segmented (*Feature Segmentation*) as proposed in [84]. First, the camera image is transformed to a binary image using a dynamic threshold. Blobs are created by applying a connected component analysis as well as a circular Hough transform [49].

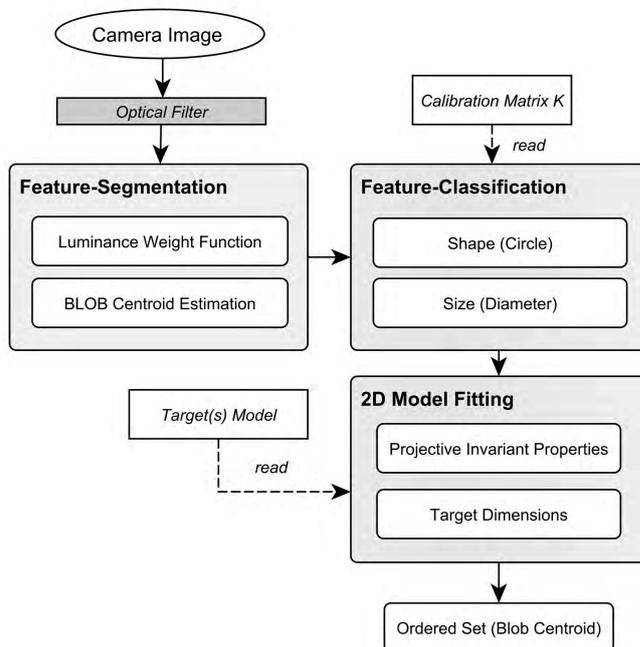


Figure 4.16: Pipeline for model identification.

Next, the center of each blob (centroid) is determined using a luminance-weighted average of the connected pixels, which describe the blob’s 2D position with sub-pixel accuracy. For further processing, the centroids are undistorted based on the *Camera Calibration Matrix* K (see Section 2.3.2.3). In the next step, each resulting blob is classified by performing shape- and size-based classification (*Feature Classification*). The minimum and maximum values for the size-filter can be manually defined to provide quick configuration for different tracking ranges. The classification results in circular-shaped blobs (*Blob Candidates*) that diameters lie within the specified range. In practice however further filtering must be performed since interfering lights can have a similar size as the target’s IR-LED blobs. Based on approaches [54, 76, 75, 81], a 2D *Model Fitting* within the set of remaining blob candidates is performed. As described in Section 2.2.2.1, the p^2 -Invariants of the blob candidates as well as their collinear properties are computed and compared to the pre-calculated target model. Thereby, false positive blob candidates are rejected and the target’s blobs are determined. Due to the permutation invariant properties of the computed p^2 -invariants, an ordered set of blobs $S^t = \{p_i^t\}, i = 1 \dots N, p \in \mathbb{R}^2$ for each image at time t is output to be further used for calibration or tracking.

4.3.5 3 Degree-Of-Freedom Tracking

To track optical markers in 3D space, the following two problems have to be solved: 1) the 2D blobs have to be identified throughout all camera views and then transformed to 3D marker locations, and 2) the 3D markers need to be tracked through time. The online image-processing pipeline for tracking is depicted in Figure 4.17.

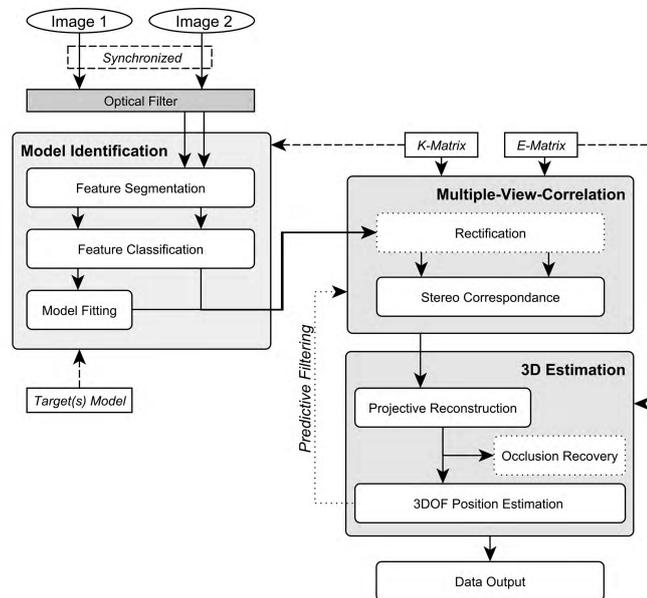


Figure 4.17: Tracking pipeline.

4. METHODOLOGY

Given an intrinsically and extrinsically calibrated, shutter-synchronized stereo camera rig, the tracking is performed as follows. After a new frame is received from each camera, blob candidates are segmented and classified in both frames (see Section 4.3.4.2). Our approach uses the projective invariant properties that were obtained during model training (see Section 4.3.4.2) to search for a pattern within an image. The 3D position of the pattern’s optical marker are only computed if and after the pattern was found in the image. To minimize computational load, the model identification is only performed in *Image 1* by applying model fitting within the set of all blob candidates. After the target blobs have been determined in *Image 1*, their correspondences have to be identified in *Image 2* amongst all blob candidates that result from the feature classification by exploiting the epipolar geometry, which is encapsulated in E (see Section 2.3.3.1). For each target blob in *Image 1*, a search for its corresponding blob is performed along its epipolar line (*Stereo Correspondence*) in *Image 2* (see Section 2.3.3.2). Thereby, corresponding features over multiple camera views can be identified (*Multiple-View-Correlation*). Depending on the camera setup of the tracking system, the baseline might be large and image pairs thus have been taken from widely differing viewpoint. Following [26, 56], it is advisable in those cases to perform image rectification to produce a pair of matched epipolar projections before stereo correspondence analysis.

By applying model fitting within the 2D projections of the target’s IR-LED not only a drastically reduced set of correspondence candidates and ambiguities is obtained but the combinatorial complexity of the multiple-view correlation problem can be considerably decreased as well. By performing a projective triangulation between each correlated 2D blob-tuple (*Projective Reconstruction*), the 3D-coordinate of each optical marker can be reconstructed. Following [84], we apply the standard *Singular Value Decomposition* (SVD) to obtain the initial 3D estimate for each blob-tuple, followed by bundle adjustment [39] with a Levenberg-Marquardt non-linear least squares algorithm for refinement. This results in a 3D point cloud of the reconstructed model points $T = \{P_1, P_2, P_3, P_4\}, P \in \mathbb{R}^3$. To further increase the algorithm’s robustness against outliers of the model fitting, the model points T are validated with a threshold to account for noise against the target’s geometric constraints $d1, d2, d3$ (see Section 4.3.4.2) and volume. Based on T and a given distance d_{epi} as the real distance between the outermost IR-LED and the epicenter of the target, the target’s epicenter $C \in \mathbb{R}^3$ can be calculated (*Position Estimation*) as follows.

$$C = P_4 - (d_{epi} * \hat{m}) \quad (4.6)$$

Therefore, we normalize the vectors $\vec{a} = \overline{P_2P_1}$, $\vec{b} = \overline{P_3P_2}$, $\vec{c} = \overline{P_4P_3}$, resulting in $\hat{a}, \hat{b}, \hat{c}$. By calculating the arithmetic mean of $\hat{a}, \hat{b}, \hat{c}$, we determine the mean direction \hat{m} which is applied according to Equation 4.6. Thereby, an arbitrary point along the line can be determined, resulting in the 3D pose of the target.

In order to enhance the robustness when tracking the target through time, the resulting target pose can be fed into a recursive filter (*Predictive Filtering*). Thereby, jitter can be reduced and the system’s intrinsic latency can be compensated. Since we currently aim for position tracking, the non-extended Kalman Filter [3, 20] is therefore employed.

4.3.6 Occlusion Recovery

If a target's IR-LED and an interfering light source lie on the same line of sight of the camera, their corresponding blobs can overlap in the images. Furthermore, parts of the target can be occluded, i.e. when the target gets partly hidden behind an object in the scene. Our model fitting approach requires four optical markers. Currently, the proposed target identification pipeline can compensate one occluded marker while retaining the capability of detecting the target within the set of blob candidates. After projective reconstruction, the 3D positions of occluded markers can be reconstructed based on the target's geometric model and the resulting 3D point cloud. The recovery of occluded IR-LEDs optimizes the accuracy of the 3D position estimate of the target's epicenter. With this recovery functionality, loss of tracking can be reduced in cases of occlusions or over-blooming by (stronger) interfering light sources.

4.4 System Development

Based on the methodological approach, we developed a hardware- as well as software system to test our tracking system in large, unconstrained indoor environments.

4.4.1 Hardware

Our hardware prototype comprises targets, the vision system and a notebook as main processing unit. The schematics of the hardware components as well as cabling and power supply are illustrated in Figure 4.18.

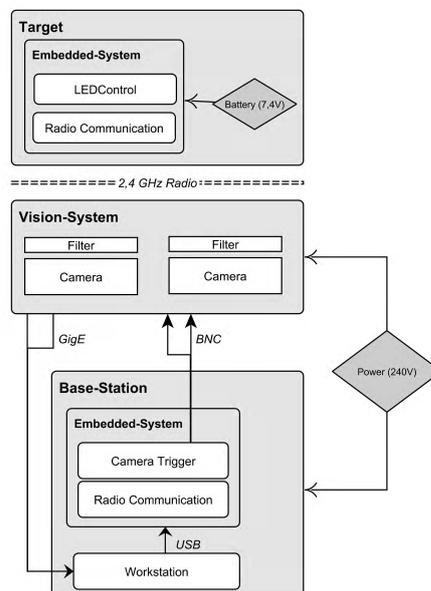


Figure 4.18: The cabling of the hardware prototype.

4. METHODOLOGY

Each target consists of a minimum of four IR-LEDs that can be remotely controlled via 2,4GHz radio hardware module. It has to be noted that the wireless LED control is not used for target identification during tracking, as it was proposed in Section 4.3.4.1 but rather for convenience during the evaluation that is presented in Chapter II.5. To be able to remotely switch the target on and off, the hardware setup from Section 4.3.4.1 is used. Since further target specifications depend on the given wide area tracking task, additional target design details are given in Chapter II.5.

The system's vision system consists of two Dalsa Genie HM1400/XDR cameras which feature low heat evolution and a global-shutter 1" mono CMOS-sensor with high NIR⁷ spectral sensitivity. Low heat evolution and large image sensors yield little sensor noise to minimize jitter in the camera image. Together with the high resolution image sensors, precise segmentation can be provided even at longer distances. The cameras offer high global shutter speed to minimize motion blur when the target is moving fast. It is capable of delivering 60 frames per second (fps) with a resolution of 1400x1024 pixels. It provides external trigger functionality and uses the GigE Vision [130] standard. Thereby, lossless image transmission while providing long cable lengths can be guaranteed. Following the results from Section 4.2, both cameras are equipped with a EdmundOptics NT63-246 high-resolution and fast (f/1.4-f/16) fixed focal lens ($f = 25mm$). To filter light from the visible wavelength spectrum, we attached a Heliopan RG-780 long wave pass filter allowing only wavelength above 780nm to transmit. Both cameras are powered by an external 12 VDC (1,5A) supply.

Both cameras are shutter-synchronized from a square-wave current loop signal that is generated by the trigger unit with a built-in programmable oscillator. The trigger unit comprises two BNC connectors⁸ and the trigger signal, generated by an Arduino Uno board [131]. Similar to the target, the Arduino Uno interfaces with the 2.4GHz radio module, consisting of a Nordic nRF24L01+ chip and a 5dBi dipol-antenna. Via USB 2.0, the Arduino board connects to the mobile workstation for communication with the tracking software as well as for power supply.

The workstation runs the software prototype and features two Gigabyte Ethernet host adapters (1x built-in, 1x ExpressCard) to interface via ISO/IEC 11801 (Category 6) cable with the cameras. The components of the base station are centrally powered by one external 240 ACV supply.

4.4.2 Software

The developed software framework follows a three-tier-architecture comprising hardware abstraction, a processing layer and data visualization on a graphical user interface, as shown in Figure 4.19. The processing core consists of loosely-coupled modules for the offline processes intrinsic calibration and model training, as well as for the online processes target identification, extrinsic calibration and tracking. The modules and their

⁷Near Infrared

⁸BNC: Bayonet Neill Concelman connector

functionalities are centrally accessed by the controller component that delivers data from the processing layer to the GUI.

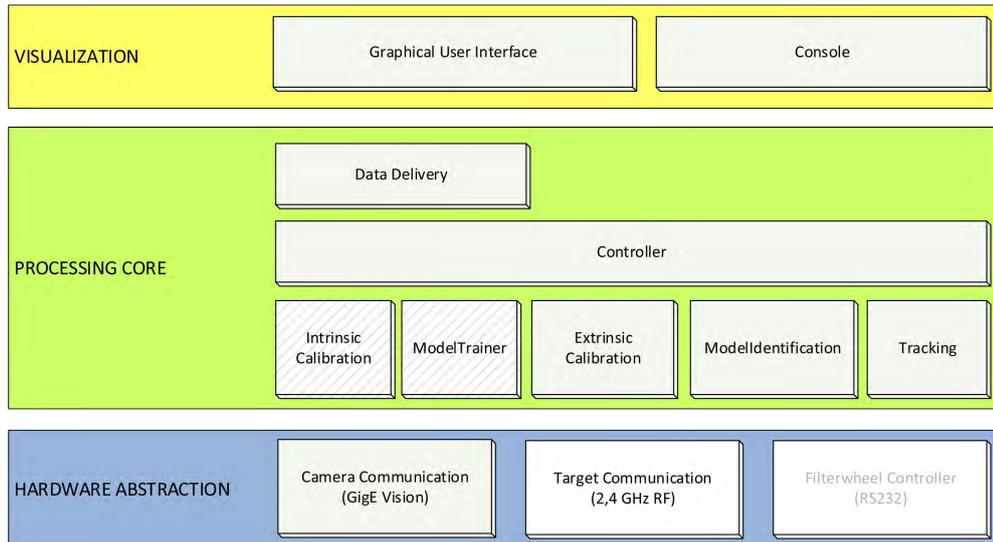


Figure 4.19: Software architecture and modules.

Our software framework prototype is implemented in C/C++ and MATLAB. For the intrinsic camera calibration, the open-source MATLAB Camera Calibration Toolbox [133] was integrated. With the open-source Arduino IDE [131], we developed the embedded component for camera synchronization and radio communication.

Training and intrinsic calibration are performed in an offline process and are implemented as stand-alone software packages. The graphical user interface of the model training component is shown in Figure 4.20.

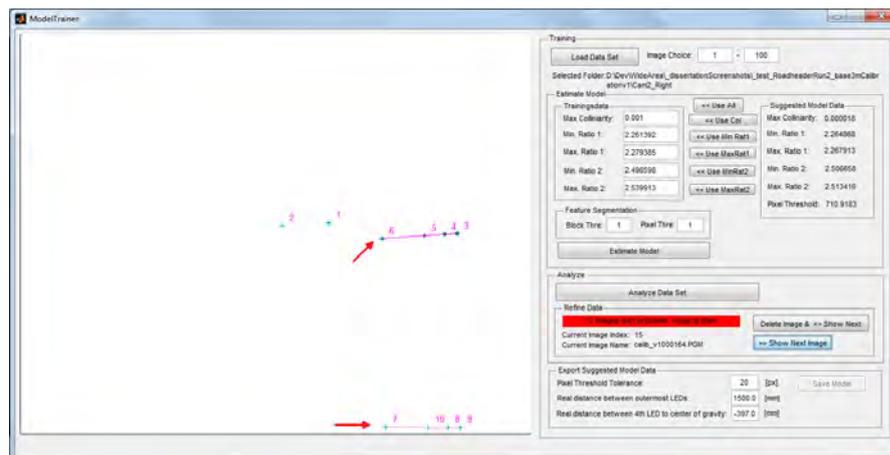


Figure 4.20: User interface of semi-autonomous Model Trainer.

4. METHODOLOGY

Based on a selected model training set, the model properties are automatically extracted and the user is informed about problems during autonomous model identification. In case of a detection of a problematic image, the user can manually adjust collinearity and p^2 -invariant range or can discard the image from the training set. If no problematic training image was found, the estimated model properties are stored in a XML model file.

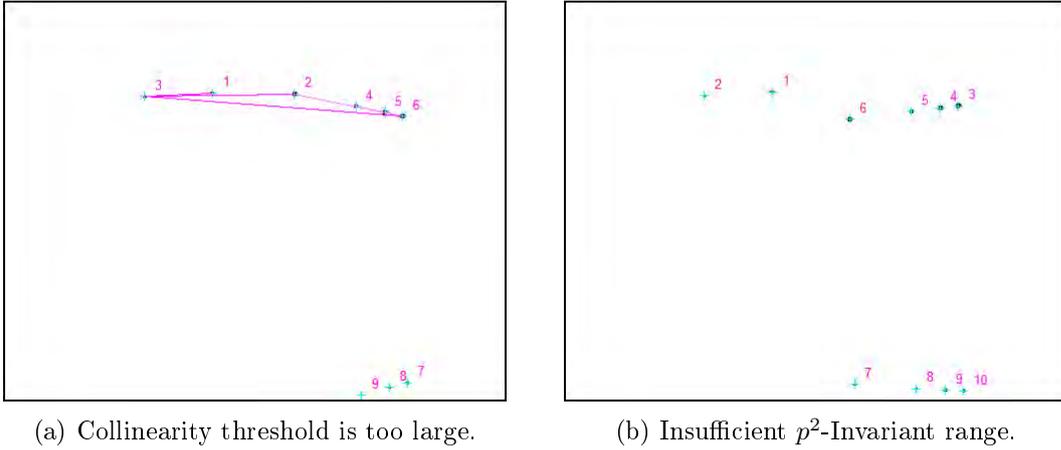


Figure 4.21: Examples of incorrect model recognition during training.

The presented Figures 4.20, 4.21a and 4.21b show interesting examples of the model detection in training images that have been captured in unconstrained settings. For visualization purpose, the camera images in all figures have been inverted. The image in Figure 4.20 has been captured in an outdoor test environment during night (see Section 5.5). In this example, target reflections in a water puddle causes the model training to detect the target twice in the image based on the given projective invariant settings, as indicated by the red arrows. Since the model detection correctly performs with the provided collinearity and p^2 -invariant range, no manual adjustment of the values is desired and the training image can be discarded from the set. In Figure 4.21, two examples are given for incorrect model recognition because of insufficient model properties. In Figure 4.21a, the collinearity threshold is too large, resulting in an incorrect identification of non-collinear blobs. In Figure 4.21b, the p^2 -invariant range is incorrect for the applied model, thus no model was identified in the depicted image. In both cases, the system proposes enhanced value for collinearity and p^2 -invariant range that the user can either apply or manually adjust the values to increase the accuracy for model detection.

The graphical user interface of the *Controller* module for analyzing the input data during calibration and tracking is depicted in Figure 4.22. In this example, the same situation as in Figure 4.20 is shown. However, due to filtering and correspondence analysis, the blobs that are reflected in the water (indicated by the red arrow) are not considered for model fitting and subsequent tracking, demonstrating the robustness of the model identification pipeline.

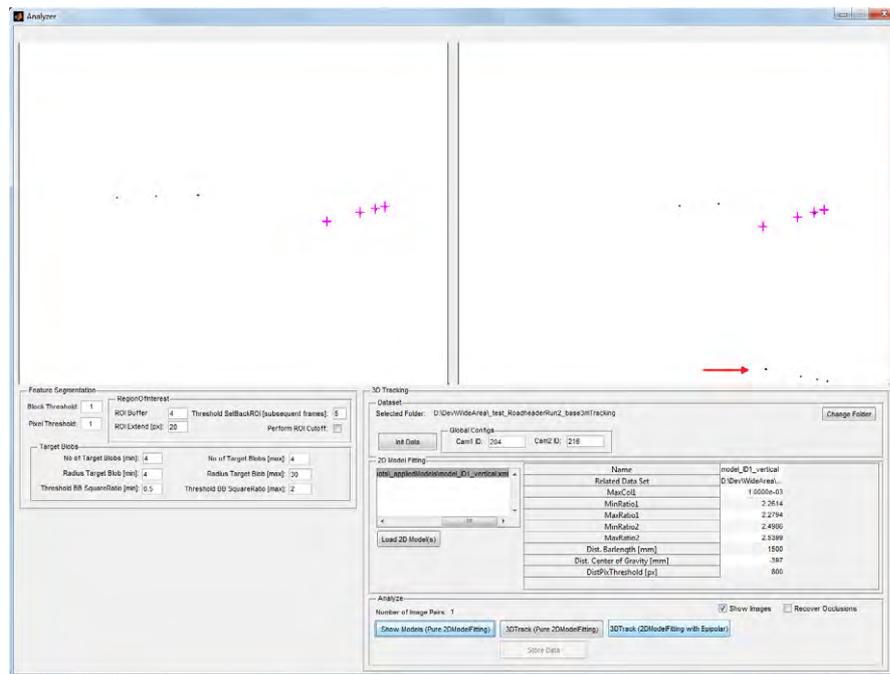


Figure 4.22: User interface of Controller to analyze data during calibration and tracking.

All parameters for feature segmentation and model fitting and tracking are centrally stored in one XML configuration file, that can be edited and is read during system start-up. The parameters for feature segmentation and model fitting are shown in Listing 4.1

Listing 4.1: Configuration for image processing and model fitting.

```
<image_processing>
  <frame>
    <dimensions>
      <width>1392</width>
      <height>1024</height>
    </dimensions>
  </frame>
  <calibration>
    <intr_cam1>..\intrinsic_calibration_204.xml</intr_cam1>
    <intr_cam2>..\intrinsic_calibration_216.xml</intr_cam2>
    <extr>..\calibration\wheelLoader\baseline_3m</extr>
    <extr_ext>TXT</extr_ext>
  </calibration>
  <segmentation>
    <!-- ranges for luminance weighting -->
    <blobfinder_block_threshold>1</blobfinder_block_threshold>
    <blobfinder_pixel_threshold>1</blobfinder_pixel_threshold>
    <!-- target properties -->
    <min_no_blobs>4</min_no_blobs>
    <max_no_blobs>4</max_no_blobs>
    <!-- ranges for shape classification-->
    <min_rad_blobs>4</min_rad_blobs>
    <max_rad_blobs>30</max_rad_blobs>
    <min_bb_threshold>0.5</min_bb_threshold>
    <max_bb_threshold>2</max_bb_threshold>
  </segmentation>
</model_fitting>
```

4. METHODOLOGY

```
<!-- access to individual model files -->
<data_folder>..\appliedModels\</data_folder>
</model_fitting>
</image_processing>
```

Information about the hardware abstraction and access are stored as well in the configuration file, as illustrated in Listing 4.2.

Listing 4.2: Configuration for hardware access.

```
<hardware>
  <image_acquisition>
    <camera rigPosition = 'left'>
      <model>DalsaXDR1400HM</model>
      <name>S4405216</name>
      <ip>192.168.1.2</ip>
      <gigEInterface>0</gigEInterface>
    </camera>
    <camera rigPosition = 'right'>
      <model>DalsaXDR1400HM</model>
      <name></name>
      <ip>192.168.2.2</ip>
      <gigEInterface>1</gigEInterface>
    </camera>
  </image_acquisition>
  <serial_bus>
    <com_port>COM8</com_port>
    <baud_rate>57600</baud_rate>
    <input_buffer_size>8</input_buffer_size>
    <output_buffer_size>1</output_buffer_size>
    <bytesAvailableFcnMode>byte</bytesAvailableFcnMode>
    <bytesAvailableFcnCount>1</bytesAvailableFcnCount>
    <bus_timeout>300</bus_timeout>
  </serial_bus>
  <trigger>
    <fps>30</fps>
  </trigger>
  <radio>
    <handshake_counter>20</handshake_counter>
  </radio>
</hardware>
```

4.4.3 System Costs

As stated in Section 1.1, cost efficiency is one of the objectives of the presented racking system. Therefore, we minimized the amount of necessary hardware and focused on off-the-shelf components as well as open source hardware and software. The current hardware prototype costs in total \sim €7300, excluding camera- and target casing. The price includes both cameras (each €2000 with IR filter), lenses (each €600), notebook (€2000), the synchronization unit (€30 for Arduino, BNC adapters and cabling) and technical parts for the target (€60 for Arduino, radio chip, battery, wires, IR-LEDs and target material).

Chapter 5

Experimental Results

Based on the methodological approach from Chapter II.4 and the implemented prototype, the system's capabilities were experimentally evaluated within three different application scenarios that share the requirements of wide area tracking in an unconstrained and even harsh indoor environment:

1. User tracking for mixed reality applications
2. Handheld target tracking for tunneling
3. Machine guidance for mining

In each scenario, the robustness of target identification and the accuracy of the relative 3D position estimation was tested with the platform from Section 5.1 and evaluated using the performance measures as described in Section 5.2.

5.1 Test Platform

We tested our system on a Lenovo W520 notebook, featuring an Intel Quadcore i7 2820QM at 2,3GHz, 8 GB memory and Windows7 (64bit). The notebook acts as processing core unit that runs the software prototype. It features two Gigabyte Ethernet host adapters (1x built-in, 1x ExpressCard) to interface via Category 6 cable with the cameras.

5.2 Test Cases & Performance Measures

As described in Section 2.1.1.2, the sources of error for an optical tracking system originate from a combination of optical aberrations, image processing inaccuracies as well as varying lighting situations. Since these factors potentially influence both the estimation of the external camera parameters as well as the position tracking, we separated them into two test cases in each of the three scenarios.

5. EXPERIMENTAL RESULTS

5.2.1 Calibration Performance

Calibration performance was measured by evaluating the target identification robustness and the subsequent accuracy of the estimated relative 3D positions. Therefore, the detected blob centroids $p \in \mathbb{R}^2$ in both cameras images are plotted as a function of 2D measurements over time, as defined in Equation 5.1.

$$f(x, y) = p_{x,y}(t_k), k = 1, \dots, n. \quad (5.1)$$

Thereby, false positive and loss of calibration target identification, target occlusions and the feature distribution across the image are visualized and can be evaluated. The calibration performance is further examined by evaluating the relative accuracy of the estimated 3D positions. Their implicit dependency on the determined camera parameters allow for conclusions to be drawn about the quality of the calibration.

5.2.2 Tracking Performance

Following the performance measures from Section 2.1.1.1 that are applied to measure the capabilities of a tracking system, the following measures are evaluated during testing the tracking performance of the three different tracking scenarios.

Relative Position Accuracy To obtain a valid ground truth for evaluating the relative position accuracy of the estimated 3D target position, the geometric distance between the two outermost target's IR-LEDs is firstly measured to millimeter precision using the Leica TPS700. Thereby, ground truth d_{bar} is determined. During tracking, the position of target's IR-LEDs $L_1..L_4 \in \mathbb{R}^3$ are calculated for each frame i and used for obtaining $\hat{d}_{bar,i} = \|L_4, L_1\|$, where $\|$ denotes the Euclidean norm. To avoid distortion of the 3D position reconstruction, no predictive filtering is applied for testing. The estimated bar length \hat{d}_{bar} is then applied to obtain the arithmetic mean $\hat{\mu}_{bar}$ with standard deviation $\hat{\sigma}_{bar}$ over all processed frames $i = 1..n$, its absolute arithmetic mean deviation $|\hat{\epsilon}_{bar}|$ and root mean square are denoted as follows.

$$\hat{d}_{bar}(RMS) = \sqrt{\frac{1}{n}(\hat{d}_{bar_1}^2 + \hat{d}_{bar_2}^2 + \dots + \hat{d}_{bar_i}^2)} \quad (5.2)$$

$\hat{d}_{bar}(RMS)$ is subsequently employed to obtain the deviation $x_{RMS}(bar)$, as an accuracy measure of the distance between the two outermost LEDs, and $x_{RMS}(P)$, as a measure of the relative accuracy of a single LED. Both measures are obtained as follows:

$$x_{RMS}(bar) = d_{bar} - \hat{d}_{bar}(RMS) \quad (5.3)$$

$$x_{RMS}(P) = \frac{x_{RMS}(bar)}{2} \quad (5.4)$$

Thereby, the relative 3D position accuracy of a single target point can be evaluated against a ground truth throughout the tracking volume.

Position Stability Based on the estimated target’s IR-LED $L_1..L_4$, the target’s epicenter $C = C_{x,y,z} \in \mathbb{R}^3$ is determined during tracking, as described in Section 4.3.5. To evaluate static jitter of the system and thus the stability (inner accuracy) of the 3D point estimation, the standard deviation $\hat{\sigma}$ of C_x, C_y, C_z as well as C over the sequence of consecutive frames is calculated and used to evaluate the system’s intrinsic tracking performance.

Tracking Latency To obtain a measure for time-dependent tracking performance, the systems latency is measured as the time delay between the change in tracker pose and the time, the system has estimated and outputs the new tracker pose.

5.3 Tracking for Mixed Reality

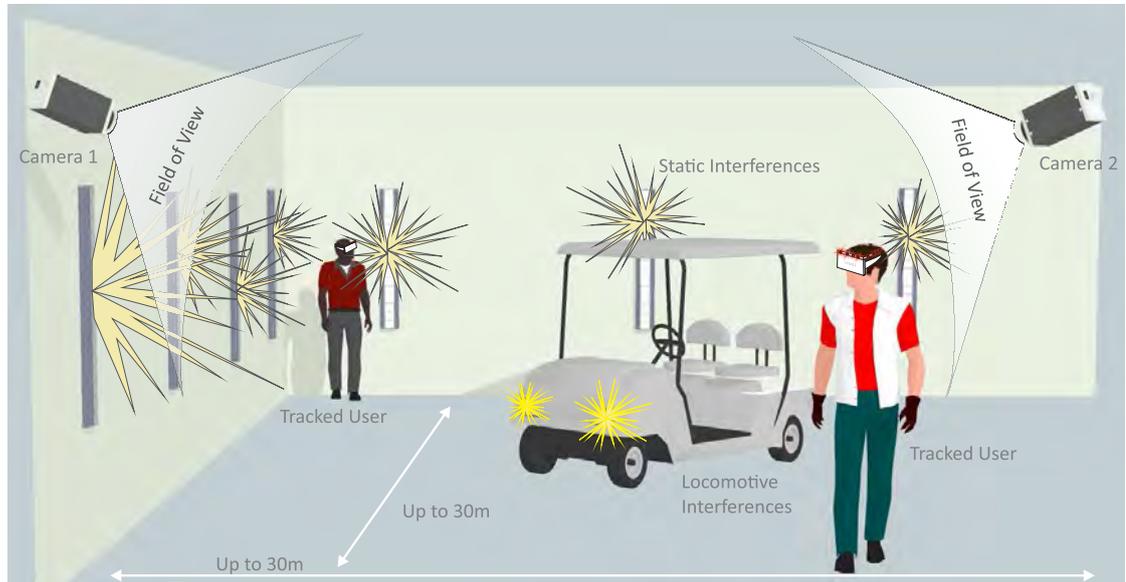


Figure 5.1: Wide area user tracking in a mixed reality setup.

Wide area user tracking can be applied to a number of application scenarios, such as user tracking in mixed reality in environments using redirected walking approaches [144, 154], tracking of artists on stages or personnel in workshops and factories. In Figure 5.1, an example scenario for user tracking in mixed reality environments is depicted, that is characterized by static and moving light sources and distances up to 30m.

5.3.1 Target Design

Following the design guidelines from Section 4.3.2 to allow for robust target identification (Section 4.3.4) and occlusion recovery (Section 4.3.6), we developed a line target. It offers continuously adjustable positioning of the IR-LEDs by fixing each LED separately with

5. EXPERIMENTAL RESULTS

nuts on a rigid bar. This ensures a rapid arrangement of the required IR-LEDs in a permutation invariant geometric constellation.

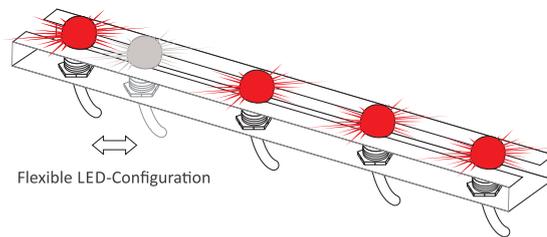


Figure 5.2: Wide area user tracking in a mixed reality setup.

Applying the proposed target design to a semi-immersive VR scenario in which the user is tracked in front of a projector wall, a single line target is sufficient to determine the user's (head) 3D position.

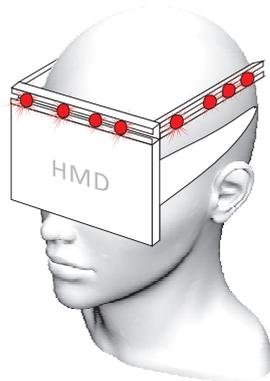


Figure 5.3: Target design for head tracking.

In a fully immersive VR environment, the user freely moves in space and wears a head mounted display for visualization. In such a scenario, using a single line target for tracking in combination with two cameras results in occlusions as soon as the user turns around. Since we want to minimize the amount of (costly) vision hardware, the occlusion problem can be compensated by applying a redundant target setup with unique targets for user head tracking, as depicted in Figure 5.3.

5.3.1.1 Prototype

The target prototype has a total length of 687mm and is equipped with four IR-LEDs *OSRAM 4850 E7800* in a permutation invariant constellation. Each IR-LED emits a peak wavelength of 850nm with a radiant intensity of $40\text{mW}/\text{sr}^1$ and features a viewing half angle of $\pm 23^\circ$. Thereby, robust feature segmentation up to a distance of 30m

¹mW/sr: milli watts per steradian

can be performed. With the employed vision hardware setup from 4.4.1, that features a 1" CMOS sensor with a resolution of 1400×1024 pixels, a minimum distance of 130mm between two neighboring LEDs is advisable with a shutter speed of $100\mu\text{s}$ to avoid blob overlaps in the camera image during rotations and at large distances. With this prototype, tracking in the intended volume can be provided.



Figure 5.4: Target prototype attached on a HMD.

Tracking in a smaller volume automatically leads to a decreased target size with the above mentioned setup. To further reduce the physical target size for volumes up to 30m , LEDs with different radiant intensity properties are applicable.

5.3.2 Test Environment

Since we were lacking access to an indoor environment that features the intended tracking ranges, we deployed the prototype in an outdoor environment during twilight and night. We added light sources (neon lights, halogen spots up to 1500W) to simulate wall illuminations, reflections and locomotive interfering lights. Thereby, we established a controllable realistic simulation of the intended tracking scenario. Both calibration and tracking were performed in an environment with static as well as moving interfering lights. We employed a baseline $d_{base} \approx 10\text{m}$ and tracking distances between the vision system and target d_{track} of $7,5 - 30,0\text{m}$.

5.3.3 Model Training

As the target's prototype from Section 5.3.1.1 is used for calibration and tracking, its model was obtained in an offline process, as described in Section 4.3.4.2. First, the real distances d_1, d_2, d_3 between the target's LEDs were precisely measured with millimeter precision using a Total Station (Leica TPS700). Afterwards, the target's projective invariant properties were calculated by evaluating 110 captured camera images across the entire tracking volume from 5 to 30m .

5. EXPERIMENTAL RESULTS

5.3.4 Camera Calibration

Before setup, both cameras were intrinsically calibrated in an offline process, as described in Section 4.3.3.1, using 34 images that captured the retro-reflective chessboard pattern from different angles and distances. For extrinsic calibration and subsequent tracking, the stereo camera system was setup with the following parameters to account for tracking distance and poor lighting situation:

- Real baseline $d_{base} \approx 10m$
- Yaw-rotation $\beta_{cam1} = 30^\circ$, $\beta_{cam2} = -30^\circ$
- Lens focus ∞
- Aperture $1.4/f$
- Shutter speed $1000\mu s$

Using the tracking target from Section 5.3.1.1, we performed the calibration at a distance around $15.0m$ from the cameras.

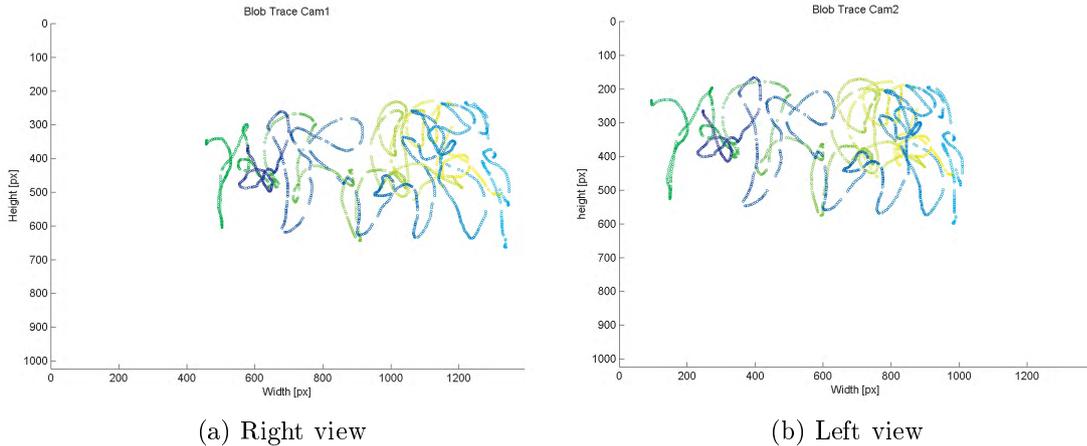


Figure 5.5: Corresponding blob traces used for extrinsic calibration.

We ran three different calibration tests with ~ 1200 frames each to evaluate the robustness of the calibration procedure. As depicted in Figure 5.5, our system robustly identifies the target despite static and locomotive interfering lights, resulting in continuous blob traces of the two outermost IR-LEDs. As illustrated, the blob trace was interrupted at some points due to complete occlusion of the target because of obstacles in the environment. Despite the unconstrained test calibration environment, our system robustly estimated the *Essential Matrix* E at each run. In average, E was determined with a duration of $\sim 110s$.

The second factor for evaluating the calibration are the tracked 3D points. We found the calibration yielding consistent 3D point estimates for all tracking distances, as presented in detail in Section 5.3.5.

5.3.5 3D Position Accuracy

To evaluate the accuracy of relative 3D position estimation, we performed measurements at six different distances between camera and target, denoted as d_{track} for each calibration procedure. At each accuracy run, the 3D coordinate of each target's IR-LED $L_1..L_4$ as well as of the target's epicenter $C = C_{x,y,z}$ was estimated based on 300 consecutive frames. Thereby, accuracy and stability were evaluated for the entire tracking volume. The obtained $x_{RMS}(P)$ values for each calibration run and each tracking distance d_{track} are listed in detail in Table 5.1.

	Calibration 1	Calibration 2	Calibration 3
d_{track}	$x_{RMS}(P)$	$x_{RMS}(P)$	$x_{RMS}(P)$
5m	3.39 [mm]	2.99 [mm]	1.78 [mm]
10m	4.12 [mm]	3.91 [mm]	2.63 [mm]
15m	4.76 [mm]	4.54 [mm]	4.58 [mm]
20m	6.08 [mm]	6.23 [mm]	7.47 [mm]
25m	6.64 [mm]	6.97 [mm]	8.92 [mm]
30m	7.44 [mm]	7.96 [mm]	9.22 [mm]

Table 5.1: Relative accuracy $x_{RMS}(P)$ of three independent calibrations.

In Figure 5.6, the arithmetic mean of $x_{RMS}(P)$ over all three calibration runs with respect to the tracking distance is depicted.

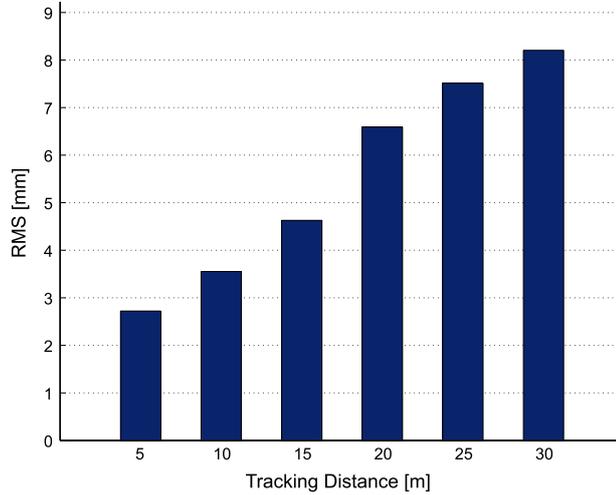


Figure 5.6: Mean of relative accuracy $x_{RMS}(P)$ over all three calibrations.

5. EXPERIMENTAL RESULTS

5.3.6 3D Position Stability

To evaluate static jitter of the system and thus the stability (inner accuracy) of the system, we fixated the target and tracked it over a sequence of 200 consecutive frames. In each frame, C_{xyz} was calculated to determine the empirical standard deviation $\hat{\sigma}_x$, $\hat{\sigma}_y$ and $\hat{\sigma}_z$ of the target's center of gravity. Throughout the entire tracking volume and across the three calibration runs, we found sub-millimeter deviation for 3D position estimation with $\hat{\sigma}_x = 0.05mm$, $\hat{\sigma}_y = 0.03mm$, $\hat{\sigma}_z = 0.11mm$, resulting in an overall mean standard deviation of $\hat{\sigma} = 0,06mm$ for C .

5.3.7 Tracking Performance

To determine the system's capability to continuously track a target throughout the entire tracking space, we moved it through the whole volume. The resulting 3D position reconstruction of each target's IR-LED is illustrated in Figure 5.7.

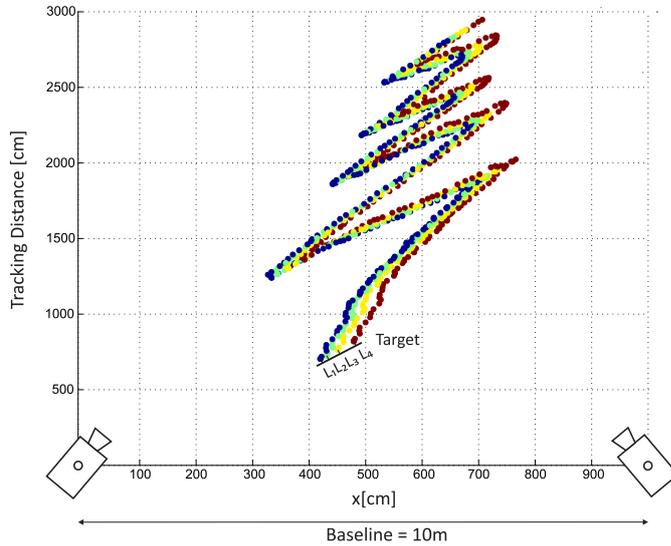


Figure 5.7: 3D position tracking from 5 – 30m.

Depending on the number of interfering lights, our system identifies and tracks a target with a latency of $\sim 69ms$ within the unconstrained test environment.

5.4 Hand-held Target Tracking for Tunneling

To further exploit the capabilities of the developed tracking system beyond application scenarios for mixed reality, it was tested in an underground scenario, using a hand-held target to track the 3D position of arbitrary static points or the moving target over time. As described in Section 3.3, existing technology lacks the ability of tracking a fast moving target, tracking of multiple targets as well as tracking without manual sighting.

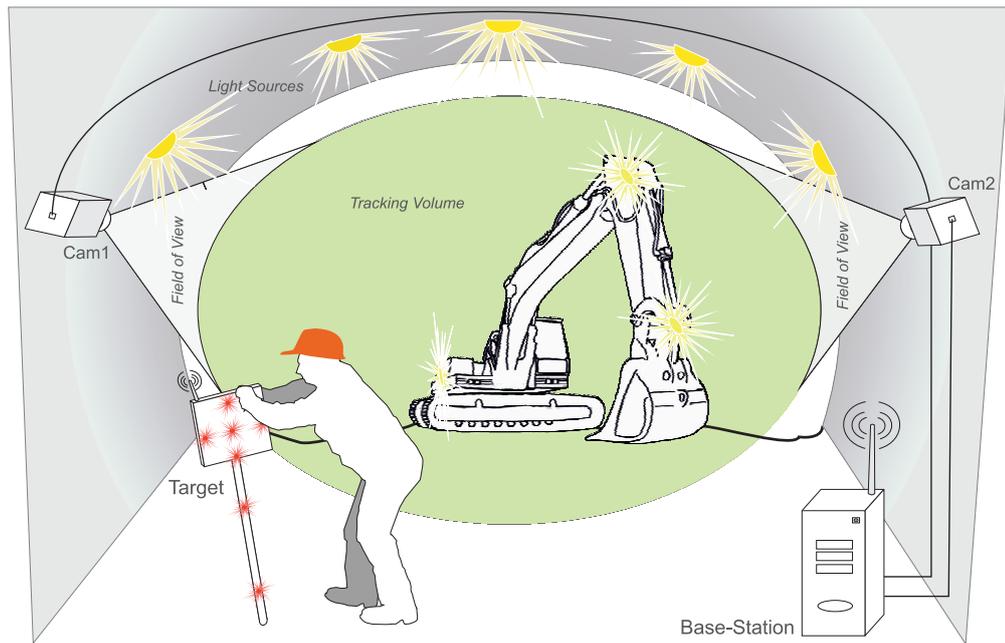


Figure 5.8: Tracking situation in an underground environment.

The intended underground tracking scenario, such as a tunnel or a mine, is illustrated in Figure 5.8. Two cameras are directed towards the tracking volume and connected to one processing unit. As soon as the hand-held target comes into sight of the cameras, tracking of the target’s 3D position automatically starts.

Compared to the previous scenario from Section 5.3, the tracking system does not only need to be able to cope with static and moving interferences, such as wall illumination and (strong) vehicles lights, but also with larger distances and harsh environmental conditions, such as dust or dirt. Dust, as a large number of small particles in the air, can influence the visibility of the target, especially at long distances, and hence decrease the quality of feature segmentation during calibration and tracking. To account for these additional challenges, a specialized hand-held target was developed and all vision components were carefully encased to enable tracking from 30 – 70m.

5.4.1 Target Design

Following the design guidelines from Section 4.3.2, the core geometric constellation of our target design constitutes a line approach.

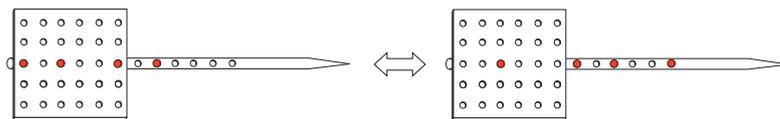


Figure 5.9: Multiple unique target constellations.

5. EXPERIMENTAL RESULTS

As depicted in Figure 5.9, our target design provides an array of holes at fixed distances, in which one or multiple IR-LEDs can be mounted. This allows for the rapid arrangement of multiple IR-LEDs in a permutation invariant geometric constellation. Furthermore, multiple unique constellations can be easily designed to simultaneously track one or more targets in the same tracking volume. To be able to test the setup with planar patterns in the future as well, it provides a rectangular area at one end.

5.4.1.1 Tracking Scenarios

With the proposed design for the tracking target, the 3D position of a static point can be measured. This is a common tunneling task. Since the target features a 20cm long tip without any optical markers attached, also points that are not visible to the cameras can be tracked, as shown in Figure 5.10. Thereby, the disadvantage of vision-based tracking systems that require a line-of-sight between cameras and measured point can be compensated to a certain extent.

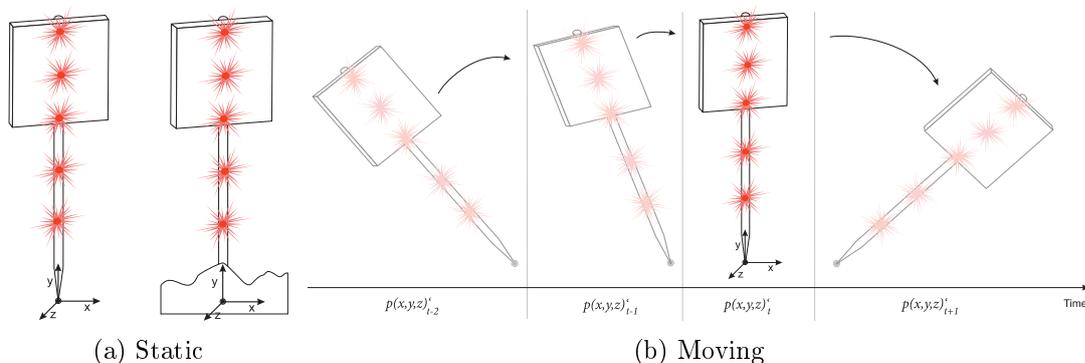


Figure 5.10: 3D position estimation of visible or invisible static and moving target's tips.

As the target is freely moved in space, as depicted in Figure 5.10b, the 3D position of the target's tip is continuously tracked.

5.4.2 System Prototype

The target prototype was developed in cooperation with Geodata Ziviltechniker GesmbH, Austria and is depicted in Figure 5.11.



Figure 5.11: Developed target prototype.

5.4 Hand-held Target Tracking for Tunneling

The maximal distance between the two outermost IR-LEDs is 820mm , while the targets total length is $120,0\text{cm}$. The target is equipped with six IR-LEDs *OSRAM 4850 E7800* to be able to construct a planar pattern in future as well. However, all experimental results are based on four collinear LEDs. As in the previous test scenario, each IR-LED emits at a peak wavelength of 850nm with a radiant intensity of $40\text{mW}/\text{sr}$ and features a viewing half angle of $\pm 23^\circ$. A minimum distance of 175mm between two neighboring LEDs with a shutter speed of $1000\mu\text{s}$ is required to ensure robust feature segmentation up to a distance of 70m . This distance was empirically determined with the given hardware setup, as described in Section 4.4.1, that features a 1" CMOS sensor with a resolution of 1400×1024 pixels. In Figure 5.12, details of the prototype are shown, including the coating of the IR-LED as well as dampness-proof cabling.

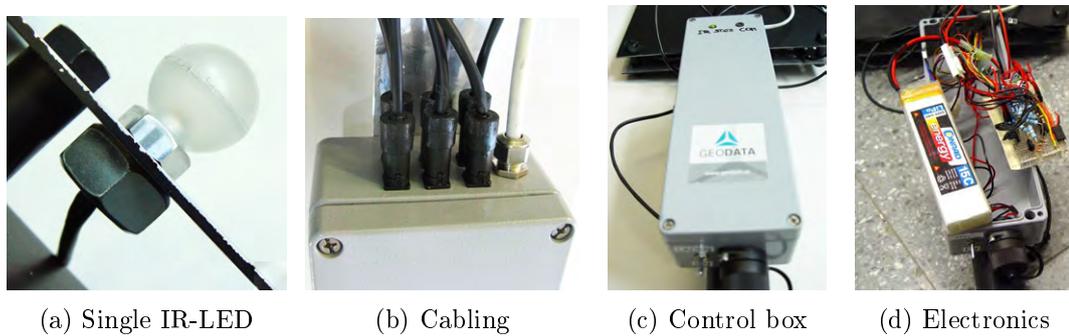


Figure 5.12: Details of the developed target prototype.

All electronic components for LED control, radio and power supply are robustly encased in the control box that features feedback LEDs to inform the user about the current tracking state. Furthermore, each camera was encased separately to be protected against dampness and dust. The components of the base station, comprising notebook with power supply, camera trigger and radio were encased as well for protection and to be transportable.

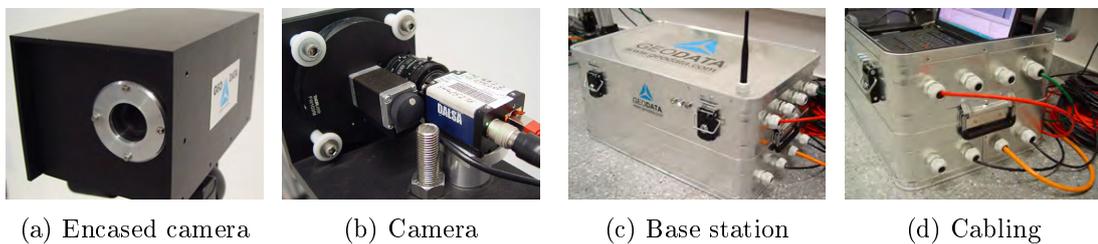


Figure 5.13: Robust and dampness proof encasement of cameras and base station.

5. EXPERIMENTAL RESULTS

5.4.3 Test Environment

We deployed the prototype in an underground metro station that offers a long-range tracking volume with static illumination characteristics, similar to a tunnel construction site. Furthermore, we dynamically applied moving light sources by hand, i.e. halogen light up to an intensity of $1500W$, to establish a controllable and realistic simulation of the application scenario, as shown in Figure 5.14. Again, both calibration and tracking was performed in an environment with static as well as moving interfering lights.



(a) Cameras facing into test environment.

(b) Light situation during calibration

Figure 5.14: Test environment in a metro underground station.

With respect to underground measurement scenarios, we performed calibration and tracking tests with baselines d_{base} from $6 - 12m$ and distances between the vision system and target d_{track} from $30 - 70m$. Therefore, we prepared our test volume by measuring and marking fixed spatial points on the ground within the tracking volume in distances of $d_{track} = 30, 40, 50, 60, 70m$, using a Leica TPS700.

5.4.4 Model Training

As the target’s prototype from Section 5.4.2 is used for calibration and tracking, its model again was obtained in an offline process, as described in Section 4.3.4.2. First, the real distances d_1, d_2, d_3 between the target’s LEDs were precisely measured with millimeter precision using a Total Station (Leica TPS700) and $d_{bar} = 820mm$ was obtained. Afterwards, the target’s properties were calculated by evaluating 205 captured camera images across the entire tracking volume from $30 - 70m$. To enhance robustness of the obtained model, we rotated the model during training as well.

5.4.5 Camera Calibration

Before setup, both cameras were intrinsically calibrated in an offline process, as described in Section 4.3.3.1, using 44 images captured from different angles and distances. For extrinsic calibration and subsequent tracking, the stereo camera system was setup with the following parameters to account for tracking distance and poor lighting situation:

- Real baselines $d_{base} \approx 6 - 12m$
- Lens focus ∞

5.4 Hand-held Target Tracking for Tunneling

- Aperture $1.4/f$
- Shutter speed $1000\mu s$

Upon each physical re-configuration of the camera stereo system, we performed extrinsic calibrations in various distances d_{calib} between camera and target with a total number of ~ 1400 frames at each run.

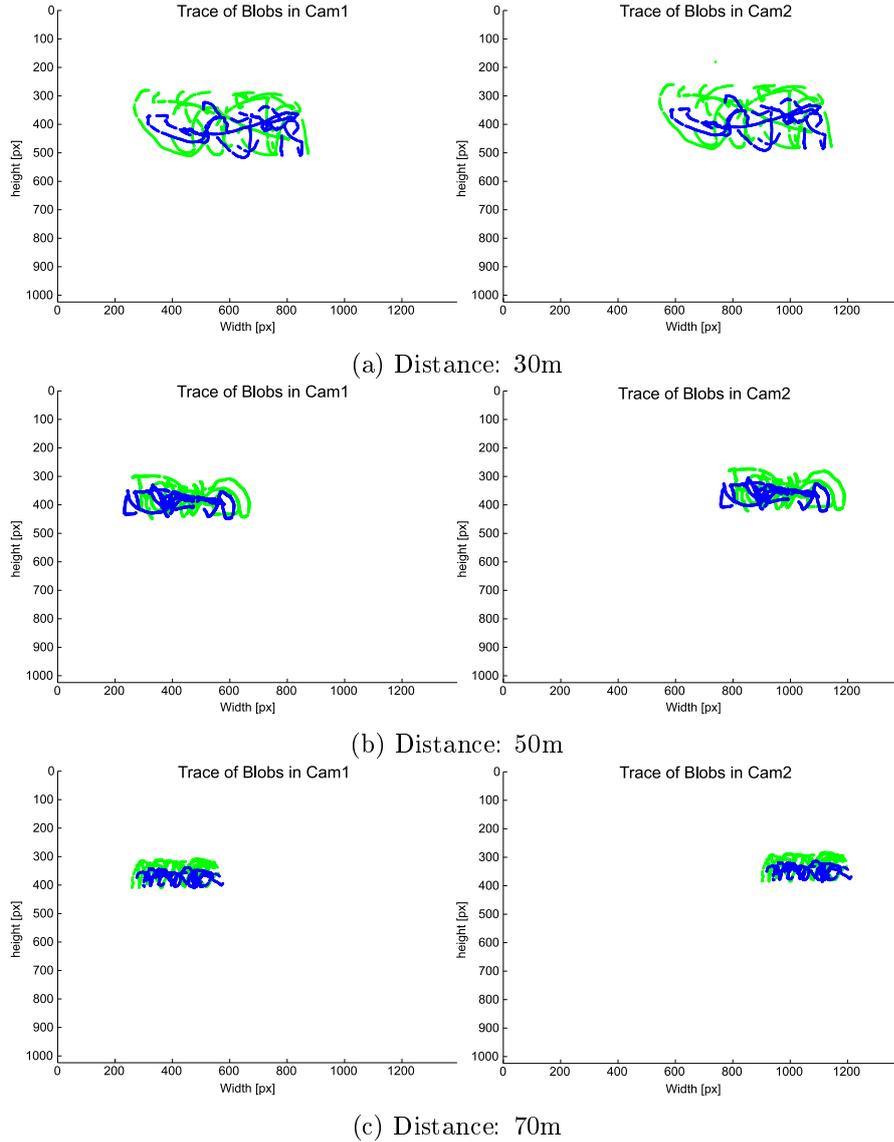


Figure 5.15: Calibration with $d_{base} \approx 6m$.

Again, our system had to continuously identify the target despite the interfering lights in the tracking volume. Figure 5.15 depicts the continuous feature segmentation

5. EXPERIMENTAL RESULTS

and resulting blob traces of the two outermost IR-LEDs for a baseline $d_{base} \approx 6m$.

As shown in Figure 5.15, for all d_{calib} our system robustly detects the target and can provide continuous blob traces. It is furthermore shown, how the coverage of blob traces in the camera images decreases as distance between cameras and target increases. With decreasing blob coverage, a decrease in accuracy of the estimated extrinsic parameters could be observed.

The calibration tests indicate the importance of well distributed blob coverage on the image to obtain an accurate extrinsic calibration result.

5.4.6 Accuracy & Stability of 3D Position Estimation

To evaluate the accuracy and stability of the relative 3D position estimation, we fixated the target’s tip to the previously measured spatial markers on the ground. Next, we performed $yaw(\alpha)$, $pitch(\beta)$ and $roll(\gamma)$ rotations around the fixated tip over a sequence of consecutive frames. Applying these movements, we received data for an extensive and robust evaluation of the entire tracking pipeline. In Figure 5.16, the reconstructed 3D positions off all IR-LEDs $L_1..L_4$ and the target’s tip (epi center) C are visualized. The sub-figures show the collected data from different perspectives.

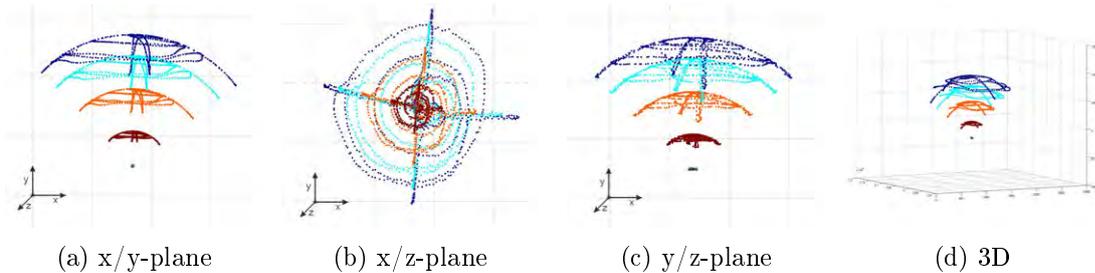


Figure 5.16: Target movement during accuracy and stability measurements.

We performed six runs in varying distances, $d_{track} = 30 - 70m$, with two different baselines, $d_{base6} = 6m$ (distance approximation = $5.95m$), $d_{base12} = 12m$ (distance approximation = $12.29m$) and $d_{calib} = 30m$. Each test was running 300 consecutive frames with α, β, γ ranging from $0 - 45^\circ$. For each run, the 3D coordinates $L_1..L_4$ as well as C were estimated to be able to evaluate relative position accuracy by analyzing $\hat{\mu}_{bar}$, $\hat{\sigma}_{bar}$ and $|\hat{\epsilon}_{bar}|$, and the stability (inner accuracy) of the 3D point, using $\hat{\sigma}(C)$.

Relative Position Accuracy To evaluate the accuracy of the relative 3D position estimation, we performed measurements at three different distances between camera and target, denoted as d_{track} for each baseline. At each run, the 3D coordinate of each target’s IR-LED $L_1..L_4$ as well as of the target’s epicenter $C = C_{x,y,z}$ were estimated based on 300 consecutive frames. ϵ_{bar} with respect to both baselines d_{base} and all tracking distances d_{track} is depicted in Figure 5.17. As it can be seen for both baselines, $|\hat{\epsilon}_{bar}|$ increases as d_{track} increases. This is due to a more inaccurate feature segmentation at larger distances

5.4 Hand-held Target Tracking for Tunneling

since blob size and luminance diminish. This causes bigger rasterization artifacts than in close range that reduces the accuracy of blob centroid computation.

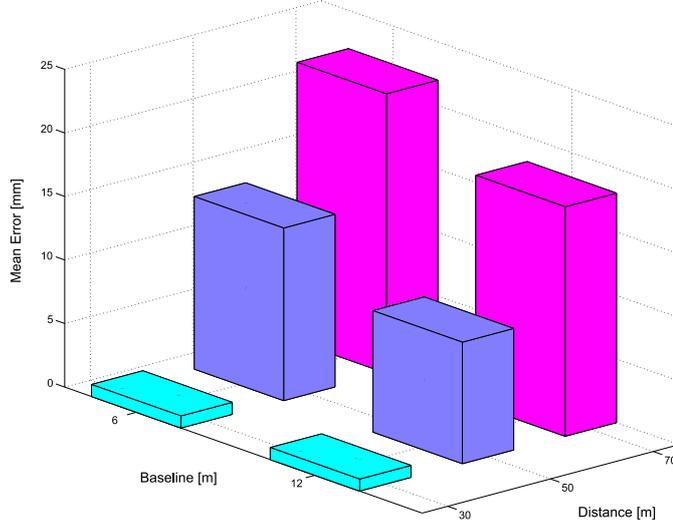


Figure 5.17: $|\hat{\epsilon}_{bar}|$ for all d_{base} and d_{track} .

Furthermore, the distances between the blobs in the camera images decrease, especially when large rotations of $\alpha, \beta = 45^\circ$ are applied. With d_{base12} , more accurate results at larger distances can be achieved compared to d_{base6} . Triangulation, as described in Section 2.3.3.4, can be more robustly performed as the baseline d_{base} increases since the glancing intersection between both rays decreases. All results of $\hat{\mu}, \hat{\sigma}$ and $x_{RMS}(bar)$ are listed in detail in Table 5.2.

d_{track} [m]	$d_{base} \approx 6m$		$d_{base} \approx 12m$	
	$ \hat{\epsilon}_{bar} $ [mm]	$\hat{\sigma}_{bar}$ [mm]	$ \hat{\epsilon}_{bar} $ [mm]	$\hat{\sigma}_{bar}$ [mm]
30	0.95	5.29	0.94	1.54
50	13.58	14.24	9.56	3.46
70	21.98	11.04	18.06	10.09

Table 5.2: Deviations and error of d_{bar} .

Up to $30m$ with $d_{base} \approx 6 - 12m$, the system is able to provide relative 3D accuracy with sub-millimeter deviation of $0.95mm$ for d_{base6} , and $0.94mm$ for d_{base12} . At $70m$, the system achieves 3D accuracy with a maximal deviation of $21.98mm$ for d_{base6} , and $18.06mm$ for d_{base12} . Hence, accuracy decreases as distance increases, and larger baselines results in better accuracy, especially at large distances. However, our evaluation for d_{base6} also reveals 3D position outliers in the result set for $30m$ and $50m$ since as a consequence, $\hat{\sigma}_{bar}$ is larger. This does not indicate an overall lack of 3D position robustness since $\hat{\sigma}_{bar}$ is low at $30m$ with d_{base6} and at all distances with d_{base12} . Since no filtering was applied

5. EXPERIMENTAL RESULTS

to avoid distortion of the 3D position estimation results, such outliers and its influence can be minimized application tracking using predictive filtering.

Overall, our proposed system provides a relative 3D measurement accuracy with an absolute maximal error $|\hat{\epsilon}_{bar}| = 21.98mm$ ($\hat{\sigma}_{bar} = 11.04mm$) for baselines $d_{base} \approx 6-12m$ throughout the entire volume. This accuracy has been achieved under constant movement and changes in rotation of α, β, γ up to 45° .

Stability After evaluating the accuracy of the relative position estimation, we evaluated the stability of the relative position estimation over 300 consecutive frames. Again, we continuously rotated the target by $\alpha, \beta, \gamma = 0 - 45^\circ$. The results are shown in detail in Table 5.3 with respect to d_{base} and d_{track} .

d_{track} [m]	$d_{base} \approx 6m$			$d_{base} \approx 12m$		
	$\hat{\sigma}_x$ [mm]	$\hat{\sigma}_y$ [mm]	$\hat{\sigma}_z$ [mm]	$\hat{\sigma}_x$ [mm]	$\hat{\sigma}_y$ [mm]	$\hat{\sigma}_z$ [mm]
30	4,07	3,61	12,80	4,92	4,04	5,57
50	4,62	4,49	24,32	6,09	3,09	11,94
70	4,18	6,98	44,92	7,50	5,29	29,61

Table 5.3: Standard deviations $\hat{\sigma}(C)$ at different tracking distances d_{track} .

The deviation of C correlates with the results and findings of Section 5.4.6. Above all, $\hat{\sigma}_z$ increases most as d_{track} increases while $\hat{\sigma}_x, \hat{\sigma}_y$ remain rather constant and are $\leq 7,5mm$ for the entire tracking volume. Thus, tracking of the head's 3D position is very stable for the x/y -axes with both baselines d_{base6}, d_{base12} . Our optical setup as well as the software processing results in millimeter deviation for $C_{x,y}$ with both baselines up to $70m$. These results can be improved by using image sensors with higher resolution. $\hat{\sigma}_z$ varies most at $70m$ with d_{base6} with a maximal deviation of $44,92mm$. With larger baselines, the 3D position estimation of C_z gets more stable ($\hat{\sigma}_z$ is decreasing for $d_{base} \approx 12m$).

5.4.7 Tracking Performance

Besides the accuracy and stability evaluation, we performed tests to determine the system's capability to continuously track the target in the intended tracking space. Therefore, we moved and rotated the target through the whole volume for $d_{track} = 30 - 70$ and inserted static and moving interfering lights into the tracking volume.

Currently the system provides ten 3D position estimates per second ($10fps$). Those rates allow for interactive tracking of static and moving objects. Figure 5.18 illustrates the target tracking and depicts the 3D position of $L_1..L_4$ as well as C . As illustrated, the target is robustly and continuously tracked with various rotations through the entire tracking volume.

5.5 Machine Tracking for Underground Guidance

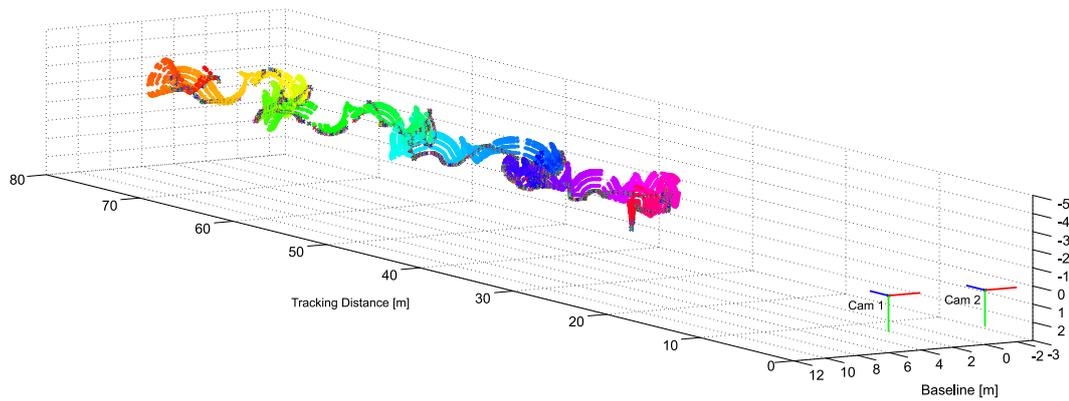
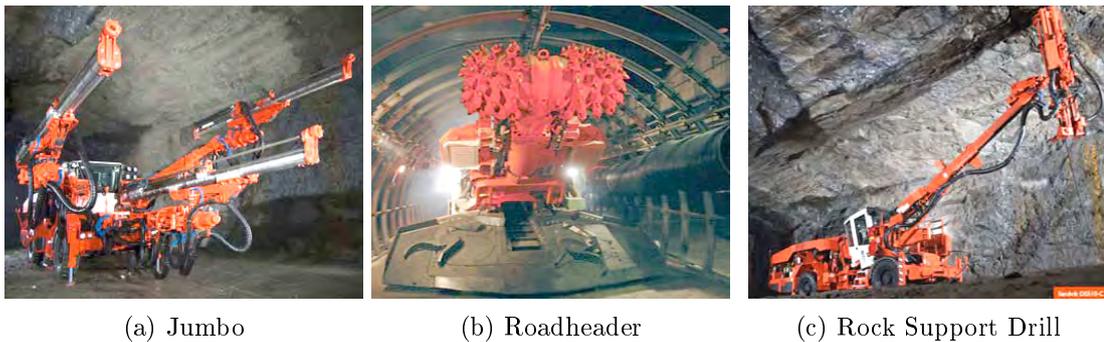


Figure 5.18: 3D position tracking of a moving target through the entire volume.

5.5 Machine Tracking for Underground Guidance



(a) Jumbo

(b) Roadheader

(c) Rock Support Drill

Figure 5.19: Examples of modern underground machinery.

Besides the ability to measure static or moving points using a handheld target, there is a huge demand in underground construction to track machines to enable remote control. Machines, as shown in Figure 5.19 such as roadheaders, jumbos, dredgers, contribute to significant cost reductions and the increase of safety and efficiency of underground works. For an efficient control of these machines the continuous and precise determination of their 3D position and orientation in the underground space is mandatory. The productivity of such machines depends on their efficient control. Therefore, an on-board machine control system is required to be able to measure, process and provide quickly, accurately and reliably all data that is needed for an optimal machine operation. One important subsystem of any such control system is the machine guidance system (navigation system) that is responsible for the determination of the absolute 3D position and orientation of a given machine and (more importantly) its different tools (e.g. booms, cutting heads) in the underground space.

5. EXPERIMENTAL RESULTS

5.5.1 Shortcoming of Existing Technology

As described in Section 3.3, classical surveying methodology such as laser measurement systems are widely applied to determine the 3D position of objects with very high accuracy. Existing automatic systems use conventional tunnel lasers in combination with active laser targets/laser receivers that are installed on the machine (e.g. for jumbos). Other approaches apply classical surveying methods such as tachymetry where computer-controlled, robotic totalstations automatically and periodically measure to shutter prisms mounted on the machine (e.g. as used for roadheaders). However, the existing technologies suffer from the following shortcomings:

- They are highly specialized and designed for particular types of machines only; therefore, they lack the universal application to other machine types.
- They can only measure and thus control one machine at a time and lack the capability of tracking multiple machines as well as machine parts that simultaneously operate.
- They can only be used for the purpose of machine guidance but not also for other measuring and surveying tasks such as setting out, profile control or deformation monitoring.
- They lack real-time tracking capability, especially when using totalstations.
- They are expensive, in particular their sensor hardware.

5.5.2 Test Environment

As a first approach to overcome the shortcomings of existing underground machine guidance systems, the developed tracking system from Chapter II.4 was tested by tracking two line targets that are rigidly attached to a wheel loader.



(a) Environment with uncased camera



(b) Wheel loader with two line targets

Figure 5.20: Details of the test environment.

The tests were conducted in cooperation with Geodata Ziviltechniker GesmbH and Sandvik Mining and Construction Central Europe GmbH, Austria. The loader was tracked open air at twilight and night during standstill and in motion, as well as under the influence of moving interfering lights as well as artificial smoke. The described test environment is illustrated in Figure 5.20, the images are manually brightened by 20% for enhanced visualization. In this environment, the tracking system has to cope with additional challenges compared to Section 5.4, such as heavy vibrations of the wheel loader during movement and standstill with engine at rest, as well as with an increased tracking volume ranging from 20 – 110m. With respect to underground measurement scenarios in tunnels and mines, we performed calibration and tracking tests with baselines d_{base} from 3 – 9m and distances between the vision system and target d_{track} from 20 – 110m.

5.5.3 Target Design

To account for the additional environmental challenges from Section 5.5.2, the robust encasement from Section 5.4.2 was reused and re-configurable machine targets were developed. Following the design guidelines from Section 4.3.2, the geometric constellation of our target design constitutes a line approach.

5.5.3.1 Evaluation of LED Range

To enable reliable tracking throughout the extended tracking range, robust feature segmentation and blob centroid determination must be ensured. Therefore, different LED types from various suppliers have been evaluated at distances from 30 – 110m featuring radiant intensities from 40 – 230mW/sr. The aim was to find the IR-LED with the best balance between appropriate intensity for long distance feature segmentation and minimal distance between two neighboring LEDs. For all tests, the vision setup from Section 4.4.1 was employed. We ran the LEDs with $V_F = 1,5V$, $I_F = 100mA$ and an operating voltage of 5V and used the vision system from Section 4.4.1 for comparison. Images were captured with 8bit, a shutter speed of 1000 μs , unlimited focus and open aperture ($f/1.4$). Over all tests, the IR-LED *Vishay TSHG6210* with 230mW/sr and a half angle of $\pm 10^\circ$ achieved the best blob quality at large distances.

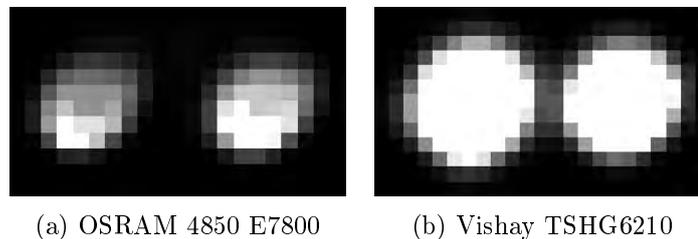


Figure 5.21: Comparison of blob quality at 110m with an inter LED distance of 34cm.

5. EXPERIMENTAL RESULTS

In Figure, the blobs of *Vishay TSHG6210* and *OSRAM 4850 E7800* (used for the target prototypes from Sections 5.3 and 5.4) are illustrated. The difference in luminance quality and even distribution is clearly visible.

5.5.3.2 Target Prototype

For the first machine tracking prototype, a target has been constructed in cooperation with Geodata Ziviltechniker GesmbH that consist of multiple *Vishay TSHG6210* IR-LEDs. Each LED is encased in a plastic hemisphere which acts as a light diffuser (see Figure 5.22a) and is installed in the center of a retro-reflecting tape target (see Figure 5.22b).

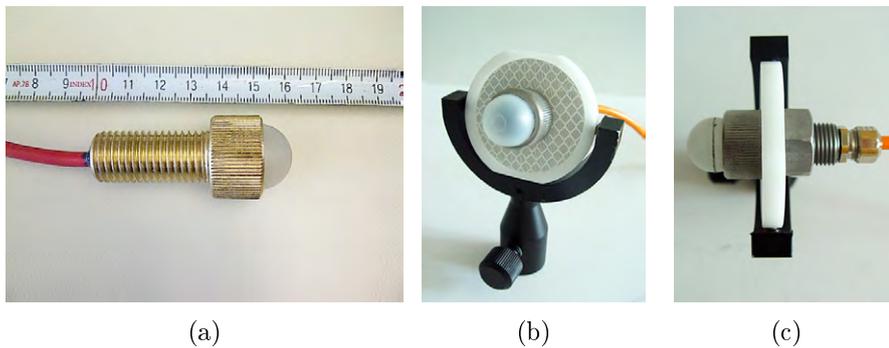


Figure 5.22: A single optical target comprising the encased IR-LED attached to a reflective geodesic foiled target.

The diffuser serves for an optimal light diffusion and feature segmentation as well as protects the IR-LED. The target design enables simultaneous geodetic measurement and optical tracking; thereby, the camera system's world coordinate system can be transformed into a geodetic reference system for comparison as well as real-life use. Since the coordinate system transformation is future work, this part is not covered and discussed within the thesis.

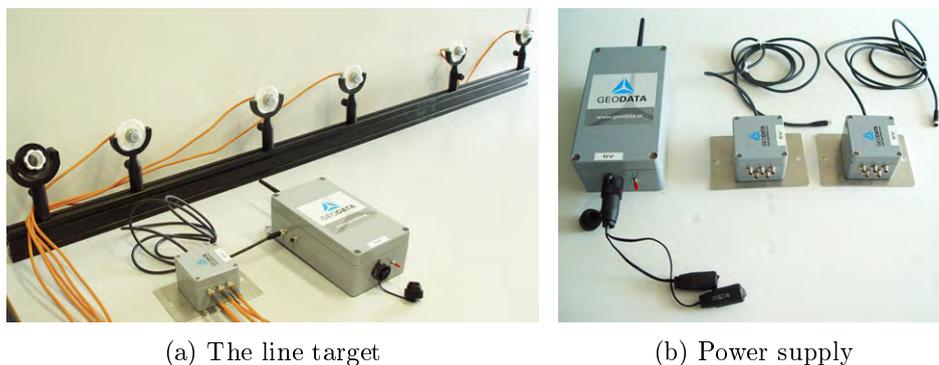


Figure 5.23: The IR-LED line target prototype for machine tracking.

To follow the overall target design guidelines from Section 4.3.2, four to five of the single IR-LEDs are combined to form a line target, as shown in Figure 5.23. Each single target is mounted to a 160.0cm square bar steel and its position can be freely adjusted along the bar. The minimal LED distance is 22cm to be able to distinguish between two neighboring LEDs at a distance of 120m, given the vision hardware from Section 4.4.1. A geodesic prism can be attached as well, as shown in Figure 5.23a to measure the target with a theodolite as well. We developed two of these line targets to test multiple constellations as well as simultaneous tracking. All IR-LEDs of both targets are centrally powered by one main unit (see Figure 5.23b), featuring battery as well as 240Hz power supply.

5.5.4 Model Training

As the target’s prototype from Section 5.4.2 is used for calibration and tracking, its model again was obtained in an offline process, as described in Section 4.3.4.2. First, the single LEDs of each target were set to a unique geometric constellation; then the real distances $d1, d2, d3$ between the target’s LEDs were measured using a total station (Leica TPS700), resulting in the following distances for *Target 1*: $d1 = 25.0cm, d2 = 40.0cm, d3 = 85.0cm$; and $d1 = 25.0cm, d2 = 55.0cm, d3 = 70.0cm$ for *Target 2*. Hence, for both targets, the distance between the two outermost IR-LEDs $d_{bar} = 150.0cm$. Afterwards, the properties for each target were calculated by evaluating 255 captures camera images across the entire tracking volume from 20 – 110m. This results in the following p^2 -Invariant ranges, defined by $[J_i^{min}, J_i^{max}]$:

$$\begin{aligned} p_{range}^2(Target1) &= [2.2270, 2.5200] \\ p_{range}^2(Target2) &= [2.1108, 2.1696] \end{aligned}$$

As it can be seen, the chosen geometric constellation of both targets results in different, non-overlapping p^2 -Invariant ranges. This is important for robust model identification.

5.5.5 Camera Calibration

Before setup, both cameras were intrinsically calibrated in an offline process, as described in Section 4.3.3.1, using 42 images captured from different angles and distances. The stereo camera system was setup with the following parameters to account for constrained baselines of a later application environment, the intended tracking distance as well as poor lighting situation. At each run, the system was calibrated with ~ 1100 images.

- Real baselines $d_{base} \approx 3m, 9m$
- Lens focus ∞
- Aperture $1.4/f$
- Shutter speed $1000\mu s$

5. EXPERIMENTAL RESULTS

5.5.6 Accuracy & Stability of 3D Position Estimation

To evaluate the accuracy and the stability of the relative 3D position estimation, we performed measurements at different distances d_{track} of both targets during standstill of the wheel loader with engines shut off. At each accuracy run, the 3D coordinate of each target’s IR-LED $L_1..L_4$ as well as of the target’s epicenter $C = C_{x,y,z}$ was estimated based on 180 consecutive frames at 10 fps. Thereby, accuracy and stability were evaluated for the entire tracking volume. The obtained $x_{RMS}(P)$ values as well as the empirical standard deviations of $\sigma(C)$ of the horizontal target with a baseline $d_{base} \approx 9m$ are listed in Table 5.4.

d_{track} [m]	$x_{RMS}(P)$ [mm]	$\hat{\sigma}_x$ [mm]	$\hat{\sigma}_y$ [mm]	$\hat{\sigma}_z$ [mm]
20	7,28	0.19	0.12	0.73
30	17,19	0.18	0.09	0.59
40	29,04	1.57	1.28	5.86
50	42,62	0.94	0.27	3.51
60	49,04	0.56	0.23	3.16
70	49,60	0.65	0.37	4.33
80	60,72	1.05	0.59	4.71
90	78,88	0.90	0.50	5.91
100	89,31	3.70	1.28	23.53

Table 5.4: Relative point accuracy and standard deviation $\hat{\sigma}_C$ for $d_{base} \approx 9m$.

The results of the relative point accuracy show deviations in the low *cm*-range throughout the volume, and up to 80*m* a very high distance-invariant stability (a good repeatability of measurement results) in the low *mm*-range as well as even below 1*mm* in the *X/Y*-plane (vertical cross section). As to be expected and explicable by theory (see Section 2.3.3.4) reconstruction accuracy and stability decreases with the distance of the target to the cameras as the intersection angle for 3D point reconstruction becomes smaller. For measuring distances higher than approx. 100*m* the low *cm*-level is exceeded in the stability results, leading to unreliable point measurements as well as increased system jitter. For $d_{base} \approx 9m$, measurements above 100*m* could not be performed due to immobile objects that were in the line of sight of *Camera 1*. As shown in Table 5.5, similar stability results were found for $d_{base} \approx 3m$ throughout the volume. Since no objects were in the line of sight, target identification and tracking could be obtained until 120*m* distance. However, we observed instabilities in the calibration process leading to unreliable point measurements for $d_{base} \approx 3m$ and higher point accuracy deviations compared to the previous experiments for $d_{base} \approx 9m$. This was found due to a insufficient blob coverage of only about 50% in both camera images in the specific test environment. Since we could not repeat the field test, we further investigated this issue, as described in Section 5.6.

To summarize our findings of the overall tracking performance, the target prototype has a maximum measuring range of approx. 120*m* under good conditions (clear

5.5 Machine Tracking for Underground Guidance

d_{track} [m]	$\hat{\sigma}_x$ [mm]	$\hat{\sigma}_y$ [mm]	$\hat{\sigma}_z$ [mm]
20	0.03	0.03	0.15
30	0.13	0.10	1.74
40	0.14	0.08	2.22
50	0.32	0.17	4.57
60	0.39	0.14	4.08
70	0.72	0.15	5.90
80	2.79	0.41	7.47
90	2.97	0.54	10.72
100	2.71	0.43	18.21
110	6.31	0.85	26.77
120	4.24	0.82	31.60

Table 5.5: Empirical standard deviation $\hat{\sigma}_C$ for $d_{base} \approx 3m$.

atmosphere, good visibility, rectangular viewing direction of the IR-LEDs towards the camera). At greater distances, the IR-LEDs cannot be reliably segmented by the *Model Identification* pipeline anymore. Up to 80m, the points' stability is reliable, resulting in robust point measurements and small system jitter. Increasing the distance between the LEDs associated with a higher radiant intensity of each LED would provide an improved target visibility and tracking stability at larger distances.

5.5.6.1 Influence of Vibrations

To evaluate the influence of heavy vibrations, such as the wheel loader engine, to relative point accuracy and stability, the targets were measured in 20, 40, 60m distances during standstill with engine shutoff (*Test 1*) and at standstill while the machine's motor was running (*Test 2*). For each run at each distance, about 200 frames were evaluated with 10fps. The comparison of the tracking results of the horizontal wheel loader target is described in Table 5.6.

$d_{track}[m]$	$\hat{x}_{RMS}(P)[mm]$	Test 1			Test 2			
		$\hat{\sigma}_x[mm]$	$\hat{\sigma}_y[mm]$	$\hat{\sigma}_z[mm]$	$x_{RMS}(P)[mm]$	$\hat{\sigma}_x[mm]$	$\hat{\sigma}_y[mm]$	$\hat{\sigma}_z[mm]$
20	7.36	0.10	0.09	0.33	7.37	0.68	0.42	2.30
40	32.63	0.17	0.16	0.68	32.51	0.54	0.39	3.40
60	53.40	0.81	0.41	3.52	53.23	1.07	0.47	4.61

Table 5.6: Comparison of relative point accuracy $x_{RMS}(P)$ and standard deviation $\hat{\sigma}(C)$ without (motor shut off) and under heavy vibrations (motor running).

Since no predictive filtering was applied during evaluation, the table shows the unaltered results of the influence of external vibrations. The tests reveal that the system's jitter increase from sub-millimeter to low millimeter deviation when the wheel loader's

5. EXPERIMENTAL RESULTS

engine is running. However, the increased jitter was not found to be strong enough to have a significant influence on relative point accuracy. This is due to the fast shutter speed of $1000\mu s$ which should be further decreased to account for this very fast movements of the target; thereby, standard deviation could be reduced as well.

5.5.7 Tracking Performance for Machine Guidance

As described in Section 5.5.2, a wheel loader was equipped with the two line targets (Figure 5.20b) and tracked during operation to gain practical experience in the performance and capability of the system prototype for machine guidance applications. Currently the system provides ten 3D position estimates per second ($10fps$). For mining and tunneling applications such as machine guidance, this update rate is already sufficient.

5.5.7.1 Tracking under normal Visibility

First, we tracked the wheel loader under normal visibility conditions during driving operation from $20 - 110m$, as depicted in Figure 5.24 where only the horizontal target is shown for better visualization.

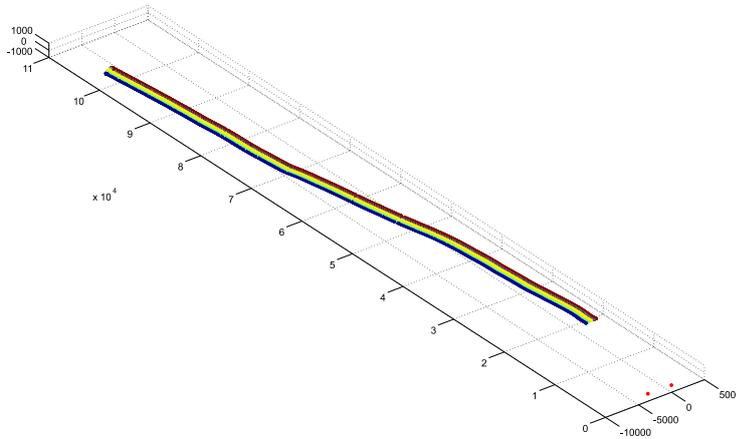


Figure 5.24: Kinematic tracking of the horizontal target from $20 - 110m$ with $d_{base} \approx 3m$.

The wheel loader was tracked over a sequence of 2560 frames; in only two frames of this data set tracking was not successful. All tracking results are directly plotted, as no filtering to remove outliers is applied. Thereby, the robustness and accuracy of the entire tracking pipeline could be objectively evaluated, resulting in a robust and continuous tracking.

5.5.7.2 Tracking with Occlusions and Poor Visibility

Next, disturbing infrared light sources were held in the line of sight and fog has been produced artificially by a machine to simulate difficult environmental conditions and other disturbing influences. In the following, three example images are given to test both environmental interferences. In each example, both targets are simultaneously tracked and the data output of the tracking pipeline indicates a successful target tracking of the horizontal target with blue crosses, and green for the vertical target. In case of occlusions, a successful target model identification is marked by yellow crosses.

To test the system’s robustness to filter interfering lights, both targets were tracked over a sequence of 499 subsequent frames while the wheel loader was positioned at a distance $d_{track} = 23m$ in front of the cameras with $d_{base} \approx 9m$. The horizontal target could be tracked in each frame of the sequence as it was not heavily affected by the interfering lights. The vertical model was subject to heavy occlusions and interferences. It’s model could be fully identified in more than 50% of the frames. In 206 frames, occlusions of one or more IR-LEDs occurred. In case of one occluded LED, the target model could still be identified and its 3D epicenter was estimated after occlusion recovery was performed. In the accuracy and stability data, we found comparable results to the measurements obtained during accuracy and stability evaluation in Section 5.5.6 (see Table 5.4) with $x_{RMS}(P) = 8.90 mm$ and $\hat{\sigma}_{x/y/z} = 0.09/0.11/0.32 mm$. This demonstrates the robustness of model identification and recovery of the tracking pipeline.

As it can be seen in Figures 5.25 and 5.26, the manually inserted disturbing lights only affect the model identification in case that the interfering lights are directly in front of or very close to the target’s IR-LEDs. This case is given in Figure 5.25 where the heavy interference leads to the occlusion of one LED of the vertical target. However, the tracking pipeline is still able to correctly identify the target model, as indicated by the yellow crosses in Figure 5.25b. Thereby, the system is able to subsequently recover the missing LED in 3D for epicenter estimation.



(a) View on the scene in visible light spectrum (b) IR scene view with tracking state output

Figure 5.25: The vertical target is partly occluded by an interfering light but can still be successfully identified, as indicated by the yellow crosses.

5. EXPERIMENTAL RESULTS

The heavy light interference that is illustrated in Figure 5.26 did not lead to occlusions of the target due to the LED's properties. Both targets' models are fully identified and tracked by the tracking pipeline despite the interference.



(a) View on the scene in visible light spectrum (b) IR scene view with tracking state output

Figure 5.26: Both targets' models are fully identified and tracked despite heavy interfering light.

Poor visibility due to fog or dust clearly reduces the measuring range and the tracking update rate. This is a common disadvantage of all optical tracking systems as well as geodetic total stations. However, as it is depicted in Figure 5.27, our system is able to cope even with dense fog in front of the IR-LEDs since their radiant intensity is strong enough. Tracking loss was only temporary for a few frames and system readiness was immediately and automatically reestablished as soon as visibility improves. This is a huge advantage compared to i.e. geodetic total stations, where tracking loss requires additional sighting of the target.



(a) View on the scene in visible light spectrum (b) IR scene view with tracking state output

Figure 5.27: Both targets' models are fully identified and tracked during fog tests.

5.6 Conclusion

We have experimentally evaluated the tracking system’s performance properties in three different wide area tracking environments, all featuring unconstrained lightening conditions and two having additionally harsh characteristics. The proposed system provides quick setup since it needs a minimal hardware setup consisting of two high quality machine vision cameras and a standard (portable) workstation for data processing. Besides stereo camera setup, pre-conditioning of the tracking volume is not required since interfering lights during camera calibration and tracking are filtered out and partly occluded targets can be recovered. Targets are designed to be re-configurable and are equipped with standard infrared light emitting diodes. We demonstrated the system’s capabilities to extrinsically calibrate the stereo camera system as well as target tracking despite heavy interferences (lights, fog). Thus, the tracking system can operate during on-going activities in the volume, featuring it to be highly unobtrusive. The system offers tracking with interactive frame rates providing centimeter precision of the relative 3D position estimates up to $100m$.

We proposed a wide area tracking prototype that can be used for user tracking in mixed reality applications. Our results demonstrate relative 3D point accuracy $x_{RMS}(P) < 9.22mm$ with sub-millimeter static position jitter $\hat{\sigma} = 0.0675mm$ throughout the entire tracking volume, ranging from $5 - 30m$. We tested our system with several different target constellations, which can be detected within both camera views with rotations yaw and pitch from $0 - 45^\circ$ as well as roll from $0 - 360^\circ$. To our best knowledge, no competing approach provides comparable accuracy for this range, especially not with the minimal amount of only two cameras. Therefore, the presented system goes clearly beyond state-of-the-art.

We demonstrated the capabilities of optical tracking to be applicable to measurement scenarios beyond mixed reality environments. By providing robust hardware encasement and a simple but flexible target design, it can be used in underground scenarios such as tunnels and mines. It can be simultaneously used for a large variety of independent underground surveying tasks, such as setting out, profile control, deformation monitoring, personnel tracking for safety and machine tracking. It provides relative 3D point accuracy with a deviation of $\leq 21.98mm$ throughout the tracking volume of $12 \times 8 \times 30 - 70m$. Up to $80m$, we demonstrated relative point accuracy of $x_{RMS}(P) < 60,72mm$ with a very high distance-invariant stability, indicated by the (sub)-millimeter static position jitter ($\hat{\sigma}_x = 1.05mm$, $\hat{\sigma}_y = 0.59mm$, $\hat{\sigma}_z = 4.71mm$). Compared to state-of-the-art underground measurement systems, our approach has the capabilities of 1) automatically starting to track one or multiple targets as soon as the target is within the view of the vision system, thus manual sighting can be omitted, 2) tracking moving as well as partly occluded targets, 3) provides a flexible target design that allows general usage of various tracking and measuring tasks and 4) addresses the need for highly automated positioning systems [68, 77].

5. EXPERIMENTAL RESULTS

During our experimental tests and extensive evaluation, the following aspects have been identified to further optimize the system. 1) The software prototype of the proposed tracking pipeline offers interactive frame rates. However, the MATLAB image processing components [137] should be replaced by C/C++ modules and parallelization should be exploited to decrease tracking latency. This reduces this shortcoming to a pure software development task. 2) As every optical technology, the proposed system requires good visibility. In presence of strong fog and dust, the achievable measuring range is reduced, however, this effect can be partly mitigated by using LEDs with higher radiant intensity as well as LED arrays. 3) Furthermore, a free line of sight must be provided for both (all) cameras. For mixed reality tracking, this shortcoming can be reduced by mounting the cameras high up the wall to avoid occlusions by users. In case of underground tracking scenarios, this can be problematic in limited space and in crowded situations, especially close to tunnel walls.

Compared to indoor tracking technologies, such as RFID that support multiple targets in a large volume, our proposed system supersedes pre-conditioning of the tracking volume to provide cost- and time-efficiency. Comparing the presented system to state-of-the-art infrared optical tracking systems in terms of range coverage and accuracy, it significantly extends the available tracking range up to $100m$ while requiring only two cameras and providing a relative 3D point accuracy with sub-centimeter deviation up to $30m$ and low-centimeter deviation up to $100m$, as shown in Tables 5.1, 5.2, and 5.4. To our best knowledge, none of the existing systems, as described in Section 3.2 gives accuracy specifications for distances greater than $10m$. Due to the implicit line characteristic of the target design, orientation can only be provided up to two DOFs. However, as depicted in Figure 5.3 for user head tracking, this can be compensated by combining several line targets into one composite target. Tracking accuracy in terms of orientation has not been part of this thesis and will be evaluated in the future. For underground surveying tasks, the achieved relative 3D point accuracy is adequate for machine guidance but was found not accurate enough for tasks such as setting out. However, the following aspects were identified to increase the accuracy. Extending the baseline results in better depth accuracy, while using an image sensor with higher resolution minimizes segmentation inaccuracies that leads as well to enhanced precision. The main aspect of optimization was found in the extrinsic calibration approach.

The evaluation of our proposed calibration method indicates promising results. Despite interfering lights, the target's LEDs are robustly segmented to ensure sufficient and reliable camera parameter estimation. However, tests revealed some limitations of the current approach. The manual movement of the target through the volume keeps the tracking system independent from additional (fixed-installed) visual features. However, not all areas of the camera image can be covered and most blobs are found in the camera images' center which results in an unbalanced blob distribution, as depicted in Figures 5.5 and 5.15. Especially in the vertical direction, distribution is limited by

human size and the length of the calibration target as well as by the natural boundaries of the physical environment, such as the ceiling and the ground. The distribution can be improved by using a longer calibration apparatus but as stated only to a certain extend. Therefore, a future aspect of the research is to use additional visual features that are extracted from the environment and fuse them with the blob features to increase the feature distribution along the edges and in the corners of the images. In a well illuminated environment, i.e. for mixed reality tracking, natural features can be extracted from the environment. In an underground environment, where illumination is poor and geometric structures are mostly found around the front face, natural feature extraction would not significantly enhance the feature distribution in the camera images. Here, the installation of additional single IR-LED markers would serve as an adequate solution. They could be equally distributed within the tracking volume and autonomously detected and subsequently extracted using the hardware interference filtering approaches from Section 4.3.4.1. Thereby, the system's unique features to function in an unconstrained environment while requiring a small amount of hardware and little user interaction would be retained.

Chapter 6

Summary

In this part, a robust wide area optical tracking approach was presented that estimates the 3D position of model-based targets. The approach extends state-of-the-art optical tracking systems by proposing a robust extrinsic stereo camera calibration, by presenting a highly re-configurable target design, and by providing a software-based processing pipeline that enables the system to cope with large tracking distances, static and moving interfering lights, partly occluded targets as well as disturbances such as fog and dust during calibration and tracking. We employ projective invariant property matching to robustly identify the model-based optical apparatus (target) that is used for extrinsic calibration and tracking. For estimating the external camera parameters, the apparatus is used to artificially generate $0D$ image features that are crucial in poorly illuminated environments with little geometric structure. Furthermore, the target's properties support reliable correspondence matching without requiring the epipolar geometry for correspondence analysis. During tracking, the approach allows model fitting already in the 2D image domain that results in a drastically reduced set of correspondence candidates. This in turn considerably decreases the combinatorial complexity of the multiple-view correlation problem.

We perform experiments with the developed software and hardware prototype in three different tracking scenarios that all feature large distances and unconstrained indoor environments. From the experiments we observe that model identification is robust against strong interfering lights, partly occlusions as well as fog that is often present in harsh environments. Furthermore, the experiments showed minimal system jitter and millimeter deviation of relative 3D point accuracy up to $30m$ and centimeter deviation up to $110m$. This outperforms competing optical tracking systems in terms of volume coverage, point accuracy and robustness. Furthermore, only a minimum of two cameras are required that significantly reduces the system's cost and complexity. This eases the necessary efforts for setup and maintenance of a mixed reality system and thereby makes it more suitable for non-experts.

In addition, we demonstrated the system's abilities to act as a wide area tracking system

6. SUMMARY

for underground surveying tasks that significantly pushes the borders of state-of-the-art optical tracking approaches that are exclusively designed and thus only applicable for mixed reality applications. Compared to competing measurement technology for underground environments, our system is re-configurable to track handheld targets as well as any kind of machines, it omits manual target sighting and allows tracking of fast movements as well as multiple targets at a time. This clearly extends state-of-the-art optical tracking technology. In terms of accuracy, it can not compete with existing laser measurement technologies but can be a first foundation for automated guidance for underground machine control.

It was found by the experimental data that the generated blob features for extrinsic camera calibration can be insufficient in terms of image coverage, caused by physical limitations of the environments through which the calibration apparatus is moved. This had lead to further research for the future to provide more reliable calibration parameters for stereo camera setups with large baseline in poorly and non-cluttered environments.

Summarizing, the demonstrated system's properties allows for robust and cost efficient wide area tracking in mixed reality and beyond. By overcoming limitations of existing optical systems, it can foster the further emerging of mixed reality into the mainstream. A broad range of wide area tracking scenarios can be envisioned, such as user tracking in virtual environments, at entertainment stages, in manufacturing workshops as well as automated control of machines in underground environments.

User Interfaces for 3D Interaction

1	Introduction	109
1.1	Motivation & Problem Statement	110
1.2	Research Objective	111
1.3	Organization	111
2	Theoretical Foundations & Related Work	113
2.1	User Interfaces in Mixed Reality	113
2.2	3D Object Selection	116
2.3	3D Object Manipulation	121
2.4	Summary	124
3	3D Selection in Handheld Mixed Reality	125
3.1	Requirements	126
3.2	Design Guidelines	126
3.3	The DrillSample Technique	127
3.4	Performance Studies	133
3.5	Experimental Results	138
3.6	Discussion	143
4	3D Manipulation in Handheld Mixed Reality	147
4.1	Methodological Approach	148
4.2	Performance Studies	155
4.3	Experimental Results	160
4.4	Discussion	164
5	Summary	167

Chapter 1

Introduction

As outlined in Chapter I.1, tracking is one of the mandatory key components to create a mixed reality environment. Furthermore, tracking is the crucial foundation for interaction in a mixed reality environment that enables a user to explore and interact with the virtual simulation. As it is depicted in Figure 1.1, interaction can be grouped in the categories *3D Selection*, *3D Manipulation*, *Navigation* with the subtasks *Travel* and *Wayfinding*, *System Control*, *Symbolic Input* and *Modeling* [62].

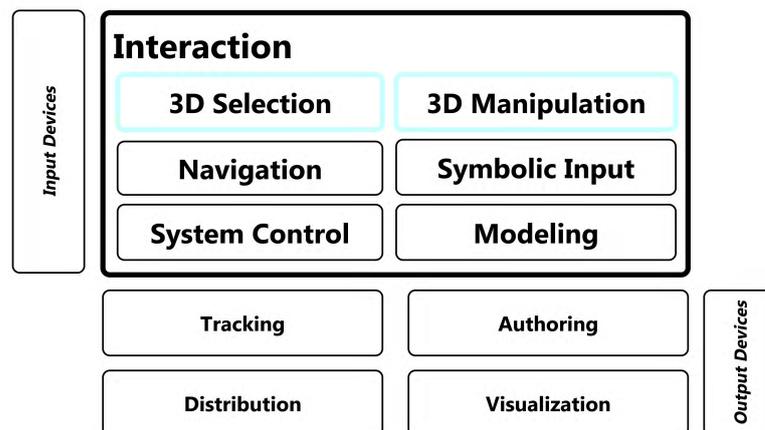


Figure 1.1: Interaction categories, with the fields of contribution marked bold.

By employing techniques for 3D selection and manipulation, the user is provided with means to select virtual objects and subsequently position, rotate (*spatial rigid object manipulation*) and scale (*spatial non-rigid object manipulation*) them. Navigation allows for moving in and around a virtual environment and incorporates travel and wayfinding. While travel enables the user to explore a mixed reality environment by employing techniques and devices for locomotion, wayfinding describes the cognitive process of defining a path through the environment aided by natural or artificial cues. System control tasks aim at changing the state of the system, usually through a graphical user interface or

1. INTRODUCTION

a command. Symbolic input addresses the tasks to process symbolic input data, such as text and numbers. Finally, modeling aims at creating 3D objects and modify their properties, including their spatial and visual appearance.

Within this thesis, contribution to *3D Selection* and *3D Manipulation* is presented by demonstrating novel techniques to select and manipulate objects in one-handed handheld mixed reality scenarios. In contrast to previously published work about portable handheld devices for mixed reality [64, 78], in this thesis, the handheld device is referred to a smartphone with a touch sensitive display to simultaneously detect multiple finger inputs. As described in Section 2.1 in Part II, pose tracking is the crucial foundation to enable 3D selection and manipulation through the involved interaction devices. While a novel system for wide area *Outside-Looking-In* tracking was presented in Part II, existing methods for *Inside-Looking-Out* 6DOF pose tracking are used as technological prerequisite in this part.

1.1 Motivation & Problem Statement

Recently emerged mobile hardware devices enable real-time rendering of a large number of 3D models. To interact with such a dense virtual scene, precise object selection and manipulation (translate, rotate, scale) are required. Existing interaction techniques for handheld mixed reality usually use the multi touch capabilities of the device for interacting with the virtual scene. Since the user has usually only one hand available for interaction while the other is holding the device several problems arise for object selection and manipulation.

3D Selection Using the imprecise finger touch input for selection yields the high probability of inaccurate extraction of small objects, especially when they are partly or fully occluded or surrounded by highly similar virtual scene objects. To increase the accuracy of the selection process, state-of-the-art approaches usually propose two-handed techniques which can not be applied to a selection task for which only one hand is available. Furthermore, in case of multi-object selection, existing approaches do not provide context information about the original spatial layout of the selected objects, making it impossible to precisely select a desired object amongst visual similar ones.

3D Manipulation As there is just one hand available for object manipulation, only simple touch gestures of one hand are suitable. In addition, since the implicit characteristics of a mobile touch screen provide only 2D data, all three coordinate axes can never be simultaneously addressed for object manipulation. To cope with this input limitations, state-of-the-art methods use complex multi-finger gestures to provide an integral way for 3D manipulations. However, their usage not only requires prior knowledge that reduces overall intuitiveness, but also multi-finger gestures that are either impossible to apply in a one-handed setup or difficult to perform on a mobile device. Thus, interaction space is limited to the physical screen size and usability can suffer because users occlude the object with their fingers [28].

1.2 Research Objective

To overcome the limitations of state-of-the-art 3D selection and manipulation techniques in one-handed handheld mixed reality, the following research objectives were formulated:

- To address the requirements of selection, novel methods have to be developed to enable precise object selection with spatial context preservation by only requiring one-finger touch input. Thereby, disambiguating and selecting strongly occluded objects or objects with high similarity in visual appearance is possible.
- To reduce the amount of finger touch input for full 6DOF object manipulation, algorithms have to be developed to provide an intuitive user interface. The focus lies on exploiting the possibilities of available tracking pose data as well as degree-of-freedom decomposition.
- All novel interface methods are to be evaluated in comprehensive user studies to explore their performance, usability and accuracy and to be able to draw a reliable conclusion on their benefits over state-of-the-art techniques.

1.3 Organization

In Chapter III.2, an overview over the theoretical foundations of 3D selection and manipulation in mixed reality environments is given and competing state-of-the-art approaches are discussed and compared. In Chapter III.3, the methodological approach of the developed selection technique and its evaluation by a thorough user study is presented. In Chapter III.4, two novel 3D manipulation techniques are described and evaluated in a comparative study. Finally, Chapter III.5 gives conclusions to the developed 3D interaction techniques.

Chapter 2

Theoretical Foundations & Related Work

This chapter covers an overview of the theoretical foundations of 3D selection and manipulation in mixed reality environments and presents related work that is relevant for the performed research.

2.1 User Interfaces in Mixed Reality

Over the last decades, computer users have become familiar with a specific set of 2D user interface components, comprising input hardware such as mouse and keyboard and output as the monitor. Furthermore, they got used to interaction techniques such as selecting a file by double-clicking, drag and drop as well as interaction metaphors as the desktop metaphor¹. However, these interface components are inappropriate for non-traditional computer environments [62], such as the various kinds of mixed reality. They represent a virtual 3D environment where traditional 2D interaction techniques and metaphors lack the capabilities to appropriately function in space.



Figure 2.1: An excerpt of 3D interaction devices.

¹The screen space on the monitor is treated as a conventional desktop where folders and documents can be placed.

2. THEORETICAL FOUNDATIONS & RELATED WORK

To enable a user to interact with virtual 3D objects, the interaction device (input) needs to be tracked and the virtual scene must be visualized (output). For the various graduations of mixed reality, a large number of possible input devices exist, ranging from multi-touch pads, 3D mice, joysticks to data gloves, 3D interaction pens as well as full body motion capturing. The input devices provide different degrees-of-freedom, thus their suitability depends on the required interaction task. In Figure 2.1, an excerpt of devices for 3D object selection and manipulation is depicted. Depending on the flavor of mixed reality, the virtual scene can be visualized to the user on a standard monitor, on a stereo projection wall, within a stereoscopic head mounted display or on a mobile screen. For a comprehensive overview and discussion of existing in- and output technology, the reader is kindly referred to [62, 83].

2.1.1 3D Interaction

As depicted in Figure 1.1, 3D interaction for virtual environments can be divided into the categories *Selection*, *Manipulation*, *Navigation*, *System Control*, *Symbolic Input* and *Modeling*. In the following sections, theoretical foundations and related work in the field of 3D selection and manipulation are described.

2.1.1.1 3D Selection and Manipulation Tasks

According to [62], 3D manipulation describes an interaction task that can be decomposed into the three basic canonical tasks *Selection*, *Translation* and *Rotation*. They act as building blocks and can be used to compose more complex scenarios [6]. These canonical tasks are used to define a taxonomy for this thesis that extends [62] and defines the following two major tasks:

3D Selection is the compound process of *Indication*, *Confirmation* and *Feedback* to select a desired virtual object in space. In the presence of multiple objects that have been indicated for selection, the confirmation process comprises a refinement task (two-step selection process). In case of a single selection, the indicated object is usually automatically confirmed by the system and subsequently used for manipulation.

3D Manipulation comprises translation (positioning), rotation and scaling of a previously selected object. All three manipulations are also referred as *RST* manipulations. Rotation is described by the three angles *yaw*, *pitch* and *roll* around the axes x , y , z . All three manipulation tasks can be performed independently, resulting in three separate tasks with a maximum of 3 degree-of-freedom each. When integrating 3D translation and rotation to one compound task, it comprises a full spatial rigid 6DOF manipulation [62].

As stated, 3D manipulation has been extended by *Scaling*, as it is a common and thus important manipulation task in real-world applications. Therefore, it was included into the formal definition of 3D manipulation for this thesis. In contrast to the spatial rigid

object manipulations translating and rotating, scaling does not preserve the shape of the object. It either scales the 3D object uniformly, meaning changing its size equally in each dimension, or non-uniformly, changing its size for each axis separately according to the user input.

2.1.1.2 3D Selection & Manipulation Metaphors

Many existing 3D selection and manipulation techniques are related to an interaction metaphor. Such a metaphor can comprise an action, an object or a combination of both and exploits the users' familiar knowledge to fulfill a specific interaction task. According to [31, 37], selection and manipulation techniques for immerse virtual environments can be classified as follows:

Exocentric Metaphors that are also known as the God's eye viewpoint. Here, users interact with the virtual environment from outside of it.

Egocentric Metaphors that allow users to interact from inside the environment. Thus, it embeds the user and is most common in mixed reality applications.

For egocentric interaction, the two major metaphors *Virtual Hand* and *Virtual Pointer* exist. The classical *Virtual Hand* is the virtual avatar of a physical interaction device and visualizes the device's real position and orientation in the virtual space. With techniques using the *Virtual Hand Metaphor*, users can reach and grab objects by "touching" and "grasping" them with their virtual hand. This metaphor can be used for object selection as well as manipulation, as described in Sections 2.3 and 2.3. Techniques based on the *Virtual Pointer Metaphor* allow the user to point at an object to indicate it for further interaction. To determine the pointing direction, usually the user's head orientation and the virtual hand's position are incorporated. Hence, tracking of head and interaction device is required. Many state-of-the-art selection techniques, as described in Section 2.2, are based on this metaphor and are characterized by virtual pointer direction, its shape and the method to disambiguate the target object.

2.1.2 3D Selection & Manipulation in Handheld Mixed Reality

In case of a handheld mixed reality environment, input and output comprises a single device. The user can interact with the virtual scene using the screen's multi touch capabilities and the 6DOF pose (position and orientation) of the device can be estimated using the device's built-in camera. This can be characterized as an optical *Inside-Looking-Out* tracking system with the mobile device as single tracker object, as described in Section 2.1 in Part II, state-of-the-art mobile devices provide real-time 3D rendering of dense virtual scenes and act as a "window into the virtual world", as illustrated in Figure 2.2. Regarding the applicability of interaction techniques classified by metaphor, both exocentric and egocentric approaches are applicable in a handheld mixed reality scenario. However, since the user can freely move the mobile device in space and thus gains a egocentric view into the virtual world through this "window", egocentric metaphors are more suitable.

2. THEORETICAL FOUNDATIONS & RELATED WORK

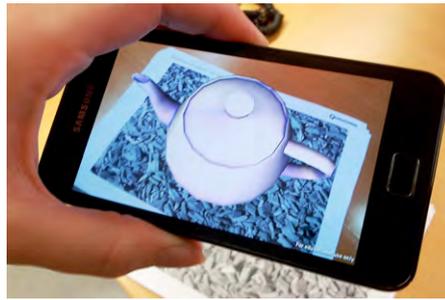


Figure 2.2: A mobile phone acting as a window into the virtual world.

Furthermore, the mobile device can be understood not only as a window into the virtual world but also as the virtual hand to interact with 3D scene objects. By the linear and direct mapping between the physical and virtual world in terms of perspective and interaction device, intuitive user interface components can be provided for 3D selection and manipulation. Therefore, we focus on state-of-the-art techniques based on egocentric interaction metaphors that are discussed in the following sections.

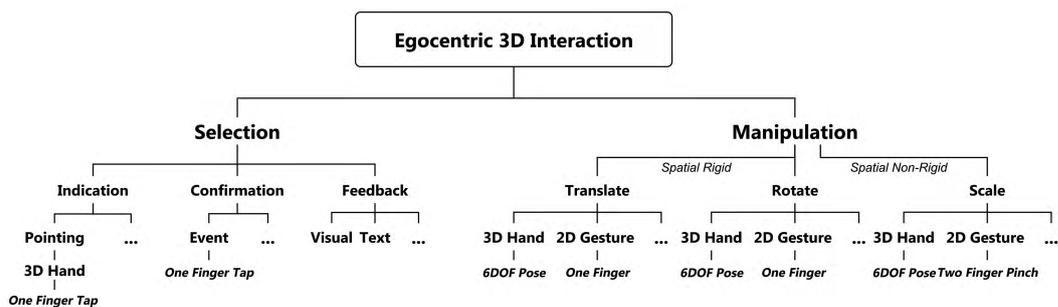


Figure 2.3: Taxonomy for egocentric object interaction in handheld mixed reality.

Following the theoretical concepts from Sections 2.1.1.1 and 2.1.1.2, a taxonomy for 3D object selection and manipulation for a one-handed handheld mixed reality setup was derived, as illustrated in Figure 2.3. It depicts only those concepts that are relevant for the performed research of this part. It classifies selection and manipulation techniques by egocentric metaphors and subsequently by task decomposition and acts as a theoretical foundation for the proposed techniques from Chapters III.3 and III.4.

After this overview on user interfaces in mixed reality, the foundations of 3D selection and manipulation are outlined within the following sections by presenting background and related work in both fields.

2.2 3D Object Selection

Selection is one of the universal interaction tasks in 2D as well as 3D and has been extensively studied [62]. As shown in literature, performance and usability of a selection

technique varies greatly, depending on specific task requirements (i.e. object size and distance) and the environment’s layout such as scene density and object occlusions. To indicate the desired object, the user can occlude the target, touch it or point at it.

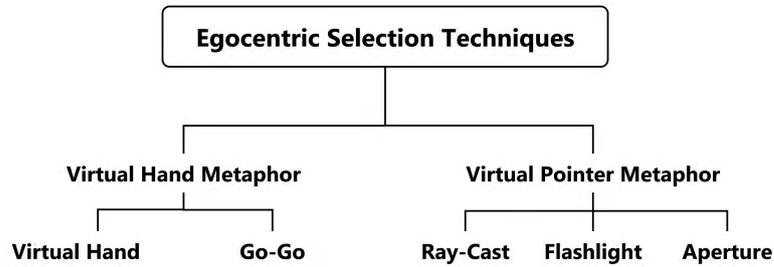


Figure 2.4: Taxonomy of immersive selection techniques classified by metaphor.

In Figure 2.4, an excerpt of immersive ego-centric selection techniques are shown that can be divided into *Virtual Hand* and *Virtual Pointing* metaphors, as described in Section 2.1.1.2.

2.2.1 Virtual Hand Metaphors

With *Virtual Hand* metaphors, such as VirtualHand [62] and Go-Go [22], the user selects objects in space by touching them; thereby, the desired object can also be fully occluded. To visualize the position and orientation of the virtual hand in the mixed reality environment, Virtual Hand techniques use a 3D pointer. To select an object, the 3D pointer is required to intersect with the desired object and the confirmation of the selection can be triggered with a designated command, e.g. a button press [62]. Using traditional Virtual Hand selection, the physical location $p \in \mathbb{R}^3$ of the interaction device is directly used as the 3D pointer’s position. However, this direct mapping introduces a spacial limitation since the physical length of the user’s arm limits the range within a desired object can be selected. Go-Go is similar to traditional Virtual Hand but extends the selection radius and thus the virtual arm by applying a non-linear mapping function to p . However, the Virtual Hand selection metaphors are not suitable in a handheld mixed reality setup because the same physical device is used as interaction input and visualization output.

2.2.2 Virtual Pointing Techniques

Virtual Pointing Techniques, such as Ray-Casting, Flashlight and Aperture [8, 62, 18, 21] are generally considered to be more precise than virtual hand techniques and provide a natural way to indicate an object [62]. They either employ one single step for object indication and confirmation or two steps and use the second step for refining the multiple objects that have been indicated for selection in the first step.

2. THEORETICAL FOUNDATIONS & RELATED WORK

2.2.2.1 One-Step Selection Techniques

Ray-Casting is a simple yet powerful selection technique for objects with a descent size on the image plane and also copes well with partly occluded objects in close distance [62]. To interact via *Raycasting* the user indicates an object to interact with by simply pointing at it. Therefore, a virtual ray along the users arm performing an pointing gesture is casted into the virtual environment and the closest object to the user that is intersected by the line is selected and used for subsequent interaction [62]. In an immersive environment, the direction \vec{p} of the virtual ray can be defined by either 1) the vector going through the user’s head position and its interaction device or 2) by the users gazing direction. The direction of the virtual ray is then attached to the position of the user’s virtual hand $h \in \mathbb{R}^3$, resulting in the definition of the virtual ray $p(\alpha)$, as defined in Equation 2.1.

$$p(\alpha) = h + \alpha\vec{p} \quad (2.1)$$

In a non-immersive desktop setup, Raycasting can also be used by casting a ray from the 2D screen point perpendicular to the display plane into the scene. To select small objects at larger distances, Raycasting can suffer from precise selection. This problem is mostly introduced by the high angular accuracy necessary for selecting small objects so that a small angular change induced, e.g. by tracker or hand jitter, causes a spatial digression at far distance [62].

Handheld Raycasting Adaptions state-of-the-art techniques for selecting 3D objects in a handheld setup usually use a simple pointing metaphor, triggered by a single tap event on the mobile screen [101, 114, 124]. However, in a cluttered virtual environment, these approaches lack precision due to users’ fingertip size. To enhance the accuracy of object selection, a set of 3D interaction techniques on mobile devices with a touch-screen is presented in [143]. The major objectives were to precisely select partly occluded objects as well as enhance the limited precision caused by the area that a finger tip covers on a comparable small screen of the mobile device. To overcome these problems for object selection the authors of [143] propose two techniques that use multi-touch input and are based on Raycasting.

The *Dual-Finger Midpoint Ray-Casting* technique is performed with three fingers. Between the two simultaneous touch points $f_1, f_2 \in \mathbb{R}^2$, the midpoint $C_{mid} \in \mathbb{R}^2$ is calculated from which a ray is casted perpendicularly to the device screen towards the virtual environment. During the selection, a cross hair is displayed at C_{mid} for user assistance. The first object that is casted by the ray is highlighted to indicate a selection candidate. The selection is confirmed with an arbitrary third touch on the screen. To increase the precision for highly occluded objects, the view can be zoomed at the midpoint by increasing or respectively decreasing the distance between both fingers. The *Dual-Finger Offset Raycasting* technique is performed with only two fingers. One finger is used to indicate the position f of the cross hair on the screen from which the ray is casted into the scene, as described above. The cross hair position C_{off} is calculated with a

pre-defined offset $o \in \mathbb{R}^2$ as follows.

$$C_{off} = (f_x + o_x, f_y + o_y) \quad (2.2)$$

The second finger is used to either change the zoom level of the view, modify the offset o or confirm the selection. Both techniques tackle and overcome the problems of partly occluded targets as well as precise selection of small objects that can occur when performing Raycasting triggered by an imprecise finger tap on a mobile handheld device. However, both Raycasting adaptations are hardly suitable for one-handed handheld scenarios due to the following reasons. Close to the screen's corners and borders, both methods are impractical or even impossible to apply. Furthermore, large portions of the touch screen are occluded and important information of the desired object's surrounding is not visible if multiple fingers are required for interaction. The limited interaction space on the handheld's touchscreen is further reduced as all fingers have to fit on the screen.

Volumetric Object Casting The *Flashlight* technique (also often called *Spotlight* or *Conecasting*) extends the idea of Raycasting by replacing the ray with a cone-shaped selection volume [18]. All objects that fall completely or partly within this selection volume can be selected, thereby the technique enables easy selection of small and distant objects without requiring the pointing precision of Raycasting. To solve for ambiguities that can occur if more than one objects fall within the conic volume, the following two rules are applied [62]. The object that is closer to the center line of the selection cone is selected. And second, if the angle between the center line of the selection cone is the same for two or more objects, than the object closer to the interaction device is selected. However, this approach has its weakness if small and tightly coupled objects in a dense scene shall be uniquely selected since the angle of the cone cannot be adjusted. As an extension of Flashlight, *Aperture* [21] allows the user to interactively control the angle of the selection cone by using a second interaction device. Although this is an intuitive extension and allows for more precise selection, this method is not applicable in a handheld scenario where only one interaction device is present.

2.2.2.2 Two-Step Selection Techniques

As described, Ray-Casting, Flashlight and Aperture can select partly occluded objects but cannot cope with fully occluded objects in a single selection step. To select entirely occluded objects, all objects that lie within the conic selection volume are considered as selection candidates (Identification). A second, additional refinement step is required to let the user manually resolve all ambiguities and to confirm the selection of the desired object (Confirmation). Using this volumetric casting, several two-step selection techniques exist and two promising ones are discussed in the following paragraphs.

SQUAD The *Sphere-Casting Refined by QUAD-Menu* selection technique [115] (SQUAD) was designed as a rapid and accurate method using a Sphere-Cast for identifying candidate objects, followed by a multi-step progressive refinement. *Sphere-Casting* extends

2. THEORETICAL FOUNDATIONS & RELATED WORK

the idea of simple Raycasting and performs object identification and confirmation in a two-step process. Firstly, simple Raycasting is performed and the first intersection with a scene object determines the position at which additionally a sphere is cast. Its size is calculated based on the distance between interaction object and the intersected object. All objects intersecting the sphere are subject of the refinement in the next step. For refinement, the image plane is split into four equally sized areas, the quad-menu, in which all candidate objects are evenly distributed neglecting their original spatial position. Afterwards, the user can progressively narrow the selected candidate objects by manually choosing a quadrant from the menu. Each time a quadrant is selected, all objects of this quadrant are rearranged amongst all four quadrants. Therefore, a minimum of $\log_4(n)$ selection steps is necessary to select the desired object out of n candidates. Although SQUAD overcomes the difficulty of precise selection of small objects by employing a volumetric cast, the progressive refinement has shown to be cumbersome to use, especially in dense virtual scenes. Furthermore, SQUAD does not preserve the original spatial context during refinement, resulting in false selections if the desired object is not uniquely distinguishable from its surrounding objects by its visual appearance.

Expand *Expand* [119] was proposed motivated by the problems that SQUAD induces when it removes the objects from its original context during refinement. It is designed to work for dense conditions when multiple objects may be subject of an indicated selection. It provides two dimensional spatial context preservation and precise selection of objects that are partly or completely occluded. The major difference between SQUAD and Expand is the usage of a dynamically sized grid instead of a fixed QUAD-menu. This ensures a spatially correct relocation of the selected objects, resembling their original spatial arrangement. Therefore, more information is given to the user to identify the desired object for selection. Furthermore, the SQUAD’s progressive refinement is omitted and an animation is introduced that visualizes the original spatial context. The object identification is performed using Flashlight selection, as described above. For the refinement step, all objects intersecting the conic volume are cloned and moved from their original to their designated position on the virtual grid; this process is visualized through an animation. The arrangement of the clones in the grid reflects the spatial context of their original counterparts, as depicted in Figure 2.5.

For confirmation, the user can manually select the desired object by pointing at it. The possibly cumbersome progressive refinement step of SQUAD is drastically simplified by Expand displaying all candidate objects at once on a dynamically sized grid. For selecting an object from a set of well arranged and previously visible objects, Expand should work well, as it was designed to work in conditions where many objects are within the cursor position. Unfortunately, no further details of the mapping process $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ of the selected 3D objects onto the grid are given in [119]. It is uncertain if and how the mapping f resembles the original 3D arrangement so that false selections for objects with a similar or identical visual appearance can be avoided, especially when the objects are partially or fully occluded.

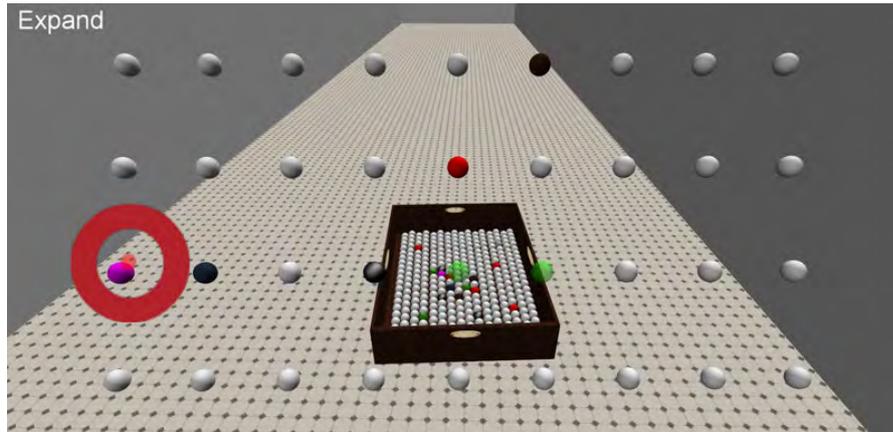


Figure 2.5: The Expand refinement view, courtesy of [119].

2.3 3D Object Manipulation

As soon as an object is selected, it can be used for subsequent manipulation. According to [62], positioning and rotating an object are universal manipulation tasks. A plurality of techniques exists, ranging from 3D immersive methods to 2D multi-touch techniques that all aim at transforming the selected object in space. 3D manipulation using large multi-touch 2D displays in tabletop environments has gained interest, i.e. by Hancock [79]. However, methods for these large-scale environments are limited to the tabletop metaphor and hence not suitable for 3D manipulation using general-purpose multi-touch displays. On handheld devices, recent approaches [114, 124, 80, 109] explore the capabilities of an user-facing camera for gesture-based manipulation using markerless finger- and hand tracking. The position and orientation of the finger or hand are mapped to the virtual object for manipulation. However, tracking the hand lacks accuracy compared to estimating the pose of a handheld device by employing natural feature tracking or by using the handheld’s built-in inertial unit. Thus, the related work of this part is focused on techniques designed for manipulation in mixed reality environments that can be adapted to handheld scenarios, as well as techniques using multi-touch finger input.

2.3.1 For Immersive Environments

The simple virtual hand metaphor that was described for selecting objects in Section 2.2.1 can also be directly used as well as extended for 3D object manipulation.

Virtual Hand When using the simple *Virtual Hand* technique, a user can manipulate a virtual object by directly mapping the movement and rotation of the interaction device - thus its 6DOF pose - onto the virtual hand object [62]. The relationship between a state of the interaction device S_r and the virtual hand object S_v is described by the

2. THEORETICAL FOUNDATIONS & RELATED WORK

zero-order transfer functions from Equation 2.3.

$$P_v = \alpha P_r, R_v = R_r \quad (2.3)$$

The position of the virtual hand $P_v \in \mathbb{R}^3$ is directly derived from the position of the interaction device $P_r \in \mathbb{R}^3$ multiplied with a scaling factor α to match possibly different scales of the real and virtual coordinate systems. The rotation of the interaction device R_r is applied with a 1 : 1 ratio to the virtual hand's rotation R_v . Both zero-order transfer functions are also called linear mappings, resulting in an intuitive manipulation because they directly simulate our interaction with everyday objects. In the following, this is referred to as *Real World Metaphors*. Due to the linear mapping, *Virtual Hand* methods are classified as isomorphic interaction techniques.

HOMER To overcome limitations of selecting objects using Virtual Hand (see Section 2.2.1), the hybrid technique *HOMER* (hand-centered object manipulation extending ray-casting) [24] uses simple Raycasting for selection and Virtual Hand for manipulation. Upon selection, the virtual hand travels to the object and is attached to it. Until de-selection, the interaction device pose is mapped onto the selected objects. After the user triggered the de-selection, the virtual hand object travels back to its original position that is again identical with the interaction device position. For manipulation, a scaling constant α is calculated according to Equation 2.4.

$$\alpha_h = \frac{D_o}{D_h} \quad (2.4)$$

It is defined as the ratio of the distance D_h between the user and the real interaction object (*real hand*) and the distance D_o between the user and the virtual object upon selection. During selection, the position of the virtual hand r_v is linearly scaled using α_h , as defined in Equation 2.5.

$$r_v = \alpha_h r_r \quad (2.5)$$

Thereby, a user is allowed to position virtual objects within a large range during the manipulation.

2.3.2 For 2D Multi-Touch Devices

Touch input by multiple fingers for 2D object manipulation has become a de-facto standard on smartphones to transform objects in 2D [72]. The direct mapping between finger touches and 2D object manipulation is straightforward and thus easy to understand for users. Various manipulation techniques have recently been designed for multi-touch displays to rotate, translate and scale objects in a three-dimensional manner. However, the implicit characteristics of a two-dimensional input device leads to several drawbacks regarding 3D manipulation. In contrast to i.e. Virtual Hand, a full 6DOF manipulation as an integral process of positioning and rotation in one step is a tedious and not straightforward task to solve with 2D multi touch input. As stated in [126], a multidimensional

object can be characterized by its attributes and classified in the two categories: integral structure and separable structure. According to the theory of perceptual structure of visual information [5], visual object attributes are separable if their dimensions are perceptually distinct and identifiable. It yields an integral structure if they can be perceptually combined to form a unitary whole. For example, the position and rotation of a 3D object are two integral attributes, thus full 6DOF manipulation can be defined as an integral task.

According to this theory, relevant state-of-the-art methods for 3D object manipulation are coarsely classified by their characteristic of separately performing the RST tasks (*Manipulation Separability*) or integrating translation and rotation to one compound task (*Manipulation Integrability*) and treat only scaling separately.

In [79], a technique for one-, two- and three-touch input interaction techniques is presented to manipulate 3D objects on any kind of multi-touch display. By using three touch interactions, simultaneous translation and rotation can be performed. This approach is limited to 5DOF and requires a large number of simultaneous touch inputs, which is not applicable to one-handed interaction on a mobile device. The Z-Technique [108] uses multi-touch input of two fingers and adjusts the depth position of the object by moving both fingers on the screen. This method requires prior knowledge of the specific two-finger gesture and does not provide 3D orientation manipulation. To handle full 6DOF manipulation, in [103] all DOFs are integrated and the technique allows the user to directly manipulate 3D objects with three or more touch points. This approach takes perspective into account and is promising, but requires at least three points and mostly two hands for interaction input to access all 6DOF. Instead of integrating all 6DOF, in [126] it is proposed to separate the 3D manipulation into translation and rotation, resulting in a 3DOF problem using 2D touch input. The approach combines the Z-Technique [108] to control 3D position with [103] for orientation control. In [143], approaches are presented to separately translate, rotate and scale virtual objects with two fingers. Each technique decomposes the 3DOF tasks into subtasks with reduced degree-of-freedom. Although only two fingers are required to provide 3D RST manipulation, a larger set of gestures needs to be known which does not make the technique intuitive to use.

In addition to these multi-touch techniques, manipulation metaphors that have been particularly designed for handheld MR are introduced by [101, 71]. Both approaches freeze the current real-world view for touch manipulation and aim on reducing faulty user input due to a shaky handheld environment. In [113], multi-modal input for 6DOF object manipulation is used. Translation is performed via touch sliders and the handheld's inertial sensor data is directly mapped to the object to change its orientation. Scaling of the object is done through a pinch gesture using two fingers. The mobile device's inertial unit is also used in [124] to provide object translation in space. [23] investigates the use of the device's tilt as input for small screen interfaces to control menus, scroll bars and view point. This early work is promising and can be extended to work as a 3D object viewer, but does not offer full 6DOF manipulation control of an object.

In [65, 66] natural feature tracking is used to estimate the 6DOF device pose. The authors compare the usage of keypad buttons with one-handed physical movement of

2. THEORETICAL FOUNDATIONS & RELATED WORK

a phone in order to move and rotate a selected object. The rotation of the selected object is chosen based on the orientation of the phone in space after the selected object has been released. Intuitive 3D rotation of an object was the main motivation in [107]. This approach extends the virtual trackball metaphor by using a second phone as rear input device. This allows for accessing the full sphere to control 3D rotation using simple touch gestures. This work is very interesting, but does not offer translation and scale operations and requires a special hardware setup.

Travel techniques for mobile virtual environments using touch input for viewpoint translation and the built-in sensors to control the viewpoint's orientation are explored in [112] while in [121], an approach for sensor-based interaction with 3D data on a mobile device is proposed. It provides interaction techniques for gaming environments for translation and rotation using simultaneously touch input and the device orientation for object manipulation. The proposed rotation requires touching an object with a finger and then rotating the device. Thereby, the object is fixed and the scene is rotated around it. This does not allow for intuitive manipulation. Since no detailed information about the rotation and translation algorithms and their limitations are given, this approach cannot be further evaluated in comparison to the proposed approach in Chapter III.4 .

2.4 Summary

This chapter presents an introduction and overview of the theoretical foundations of 3D selection and manipulation, classified by tasks and metaphors. Following the theory, a taxonomy for 3D object selection and manipulation in a handheld mixed reality scenario is introduced that is further applied throughout this part of the thesis. Subsequently, related state-of-the-art techniques for 3D object selection and manipulation that are suitable for one-handed handheld mixed reality environments are presented.

Chapter 3

3D Selection in Handheld Mixed Reality

To address the limitations of existing selection techniques for handheld mixed reality scenes, as described in Section 1.1, *DrillSample* is presented as a novel technique for 3D object selection in dense virtual scenes. *DrillSample* is a two-step technique providing precise selection and disambiguation of visible, partly occluded or invisible objects, which can also be highly similar in appearance. To cope with the imprecise finger input, it allows the user to confirm its object indication in an optional second refinement step, if more than one object has been selected in the initial step. At the refinement step, all indicated objects are presented to the user as 3D virtual clones for confirmation that is again achieved using a single tap input. The original 3D spatial context of the selected objects is preserved in this detailed visualization view. Compared to competing approaches, *DrillSample* only requires single tap input for object indication and confirmation while it is fully 3D context preserving.

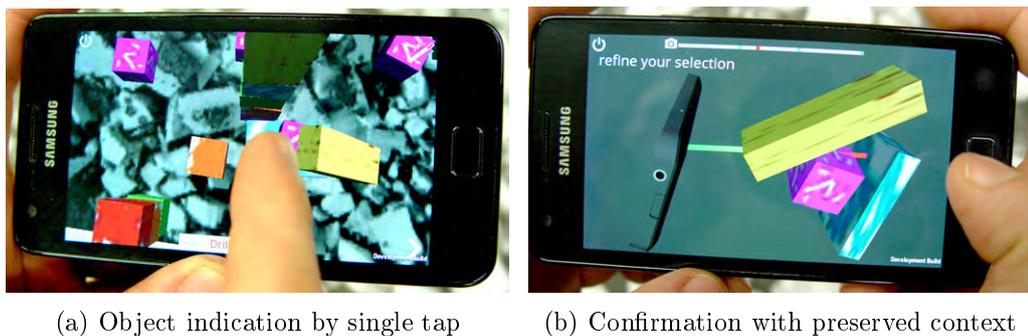


Figure 3.1: The two-step *DrillSample* technique.

In Figure 3.1, an example of selecting an object in a highly occluded scene using *DrillSample*'s indication and refinement capabilities is illustrated. Within this chapter, the methodological approach to develop the novel technique is firstly presented, followed

3. 3D SELECTION IN HANDHELD MIXED REALITY

by a throughout user study to be able to perform an in depth evaluation and comparison with competing approaches.

3.1 Requirements

To achieve the research objective from Section 1.2, requirements were specified to be fulfilled by the 3D selection technique. When designing a selection technique for handheld mixed reality, there are important factors that influence performance and ease-of-use. Since precise selection in dense one-handed handheld mixed environments should be achieved, the application scenario's specific characteristics must be taken into account during selection design as well as for the choice of a baseline technique to guarantee a fair evaluation. The requirements can be summarized as follows:

Single I/O Device Input and output comprise a single device. Thus, independent tracking of user's interaction and output device is not available compared to other mixed reality scenarios. The handheld's device pose needs to be tracked by appropriate techniques, such as *Inside-Looking-Out* optical tracking.

Limited Gesture Complexity Touch input by fingers can be imprecise due to the large area the user's fingertip covers on the screen. Since there is only one hand available for interaction, complex multi hand- and finger gestures cannot be applied to improve selection precision.

3.2 Design Guidelines

Based on our motivation and the outlined requirements, we developed the following design guidelines to enable precise selection in a one-handed dense handheld AR environment.

Keep Direct Touch Abilities One of the most appealing aspects of touch displays is the ability to directly "touch" an object in order to select it. We aim to support this direct manner and do not introduce an offset to the cursor due to the disadvantages that are mentioned in Section 2.2.2.1.

Keep Interaction Simple Since multi finger interaction is not a straight forward metaphor and requires prior knowledge of specific gestures, we aim to reduce user touch input complexity for object selection. Only one-finger input should be applied to allow precise object selection. Two-finger input using a single hand should only be applied for optional interaction such as detailed inspection of selected objects.

Enable Disambiguation and Unique Selection Since objects can be partly occluded or even invisible in dense virtual scenes, it is important to provide a technique that supports selection of these objects. Furthermore, objects can be highly similar in visual appearance. Thus, it is important to present multiple selected objects in the correct spatial context to assist object disambiguation while taking the limited screen size into account.

3.3 The DrillSample Technique

Inspired by Raycasting and Expand (see Section 2.2.2), the novel selection technique DrillSample was designed in an iterative fashion according to the outlined guidelines while meeting the specified requirements. The workflow of DrillSample is illustrated in Figure 3.2.

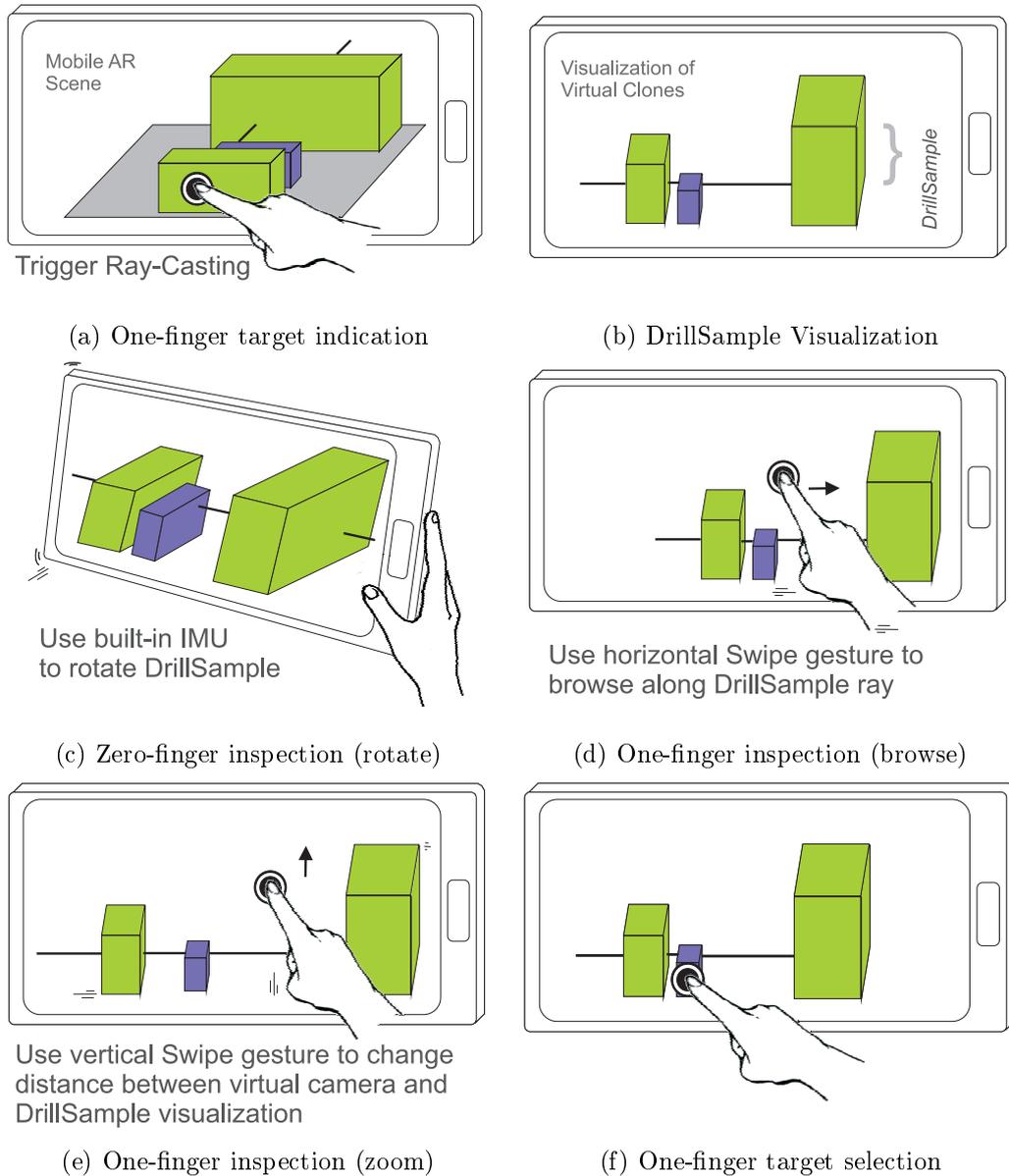


Figure 3.2: DrillSample’s two-step selection process.

It requires single device tracking and only one finger input to select an object in a two-

3. 3D SELECTION IN HANDHELD MIXED REALITY

step interaction process. The selection method provides one initial indication step (see Figure 3.2a) and an optional refinement step (see Figure 3.2b) for selection confirmation in case of multiple object indication. By visualizing the indicated object in a spatial context preserving manner during the refinement step, disambiguation of partly or fully occluded objects as well as objects with very high similarity is provided. This type of visualization and thus the technique's name was motivated from taking a drill (soil) sample for geological measures to visualize and analyze the different segments. In case only one object is indicated, it is immediately confirmed by the *DrillSample* technique, as it was proposed in the original Raycasting method. Thereby, simple selection tasks do not suffer from increased interaction steps.

3.3.1 Selection Design

DrillSample starts with a single tap on the screen which triggers *Mobile Raycasting*, as described in Section 3.3.2. Instead of selecting only the first (and closest) scene object, it returns all objects that have been intersected by the virtual ray. In the second refinement step, this set of objects is presented to the user as 3D virtual clones by visualizing them as if they were "pulled" out of the virtual scene, thus constituting the drill sample. All clones are rendered on a solid gray background (see Figure 3.1b) with the live tracking is turned off. The drill sample is aligned parallel to the horizontal axis of the image plane and the clones are arranged on a horizontal line. This is referred to as the *DrillSample Visualization*. The x- and y-position of the clone's centers corresponds to the hit point of the ray with the original objects, while the depth information is represented by the clone's position on the horizontal line of the *DrillSample* visualization. The spatial context of the indicated objects from the original scene layout is preserved upon casting by the virtual ray that extends the idea of *Expand* in the depth domain. Thereby, simple disambiguation of selected objects that are occluded or of similar visual appearance is provided. The drill sample visualization allows for a detailed inspection of the indicated objects by the following interactive options:

- By using the handheld's built-in Inertial Measurement Unit (IMU)¹, the user can rotate the drill sample with the pivot point at screen center to inspect objects from different angles (see Figure 3.2c).
- By applying a horizontal one finger swipe gesture, the user can browse through the clones by traveling along the horizontal line (see Figure 3.2d).
- With a vertical one finger swipe gesture (or an undirected two finger pinch gesture) the virtual camera can be zoomed in and out to provide a detailed view or an overview onto the *DrillSample* visualization (see Figure 3.2e). This interaction is especially helpful on small displays to gain a quick overview if many objects have been selected.

¹Depending on the hardware, the Inertial Measurement Unit consists of accelerometer, gyroscope and magnetometer.

The refinement step is finished with confirming the desired objects from the drill sample using a single finger tap gesture (see Figure 3.2f)). Upon confirmation, the user is informed about the selection, the 3D clones are removed from the scene and the live tracking is rendered in the display again. A formal description of the DrillSample state flow is given in Figure 3.3 where the specific gestures throughout the different phases are depicted.

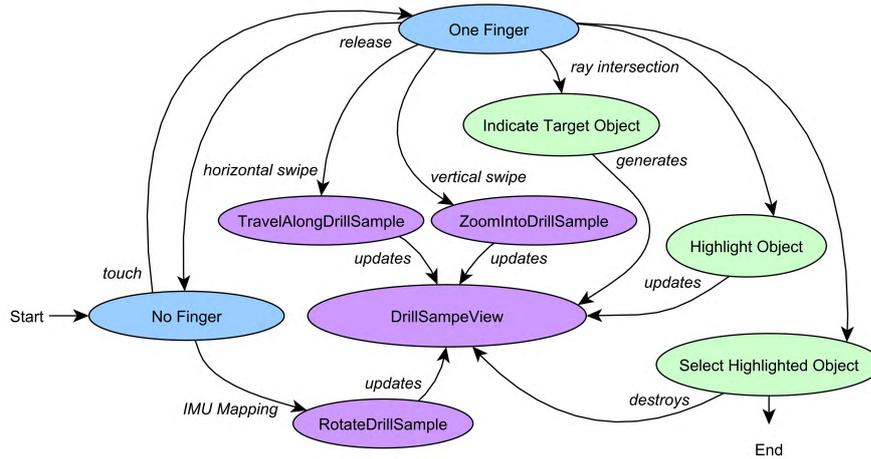


Figure 3.3: State diagram for DrillSample selection.

DrillSample is especially useful in dense environments but also works well in sparse scenes when only single objects are selected. While the selection process requires additional time in case of multiple object indication, an object is immediately confirmed and selected if only one object was casted by the virtual ray in the first step.

3.3.2 Mobile Raycasting

Original Raycasting (Section 2.2.2.1) requires the user’s head and its interaction device to be tracked to calculate the pointing direction.

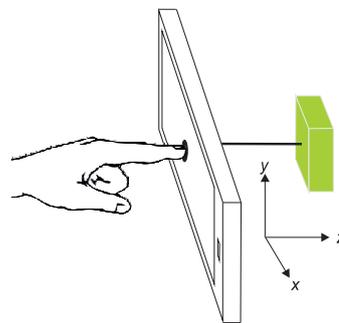


Figure 3.4: Ray-Casting adapted to use it in a handheld mixed reality.

3. 3D SELECTION IN HANDHELD MIXED REALITY

This is not applicable to handheld mixed reality as there is only one tracker object (the mobile device) available. However, the approach for desktop mixed reality scenarios can be easily extended for a handheld scenario by applying a tap gesture, as depicted in Figure 3.4.

In the absence of two points in space to calculate the virtual ray’s direction, *Mobile Raycasting* can be seen as the transformation of the 2D tap point $p_t = (x, y) \in \mathbb{R}^2$ from screen space to world space so that the ray is shot into the virtual scene in a perpendicular way with the direction \vec{d}_{ray} . As the virtual camera’s parameter (image plane dimensions, field of view) are known and its 6DOF pose is given by the optical tracking, obtaining the 3D point $P_T \in \mathbb{R}^3$ of p_t can be solved by applying a standard screen to world space re-projection. Mobile raycasting can then be performed along $\vec{r}(P_c, \vec{d}_{ray}) \in \mathbb{R}^3$, with $P_c \in \mathbb{R}^3$ being the virtual camera’s position and \vec{d}_{ray} the direction from P_c to the back-projected point P_T . After the ray is shot, all objects are tested for intersection and one of the following options is applied:

1. All casted objects are returned as a set of clones $S(O_i)$, $i = 1 \dots N$. This is applied during DrillSample selection.
2. The first casted object $S(O_1)$ is selected, as in the original Raycasting approach.

Each object O_i contains the object’s orientation upon selection $o_{sel} \in \mathbb{R}^4$, the ray hitpoint’s position $p_{hit} \in \mathbb{R}^3$ as well as the object’s geometry G .

3.3.3 Algorithm

To formalize the illustrated selection process, the proposed DrillSample is described in pseudo code in Algorithm 3.1.

3.3.4 Crucial Aspects of the Algorithm

Initial tests of the algorithm revealed that certain rotations at refinement (see Figure 3.2b) should be restricted since they were found not to be beneficial for the users’ perception of the spatial context or were even confusing. Most important, all rotations around the roll axis and rotations around the pitch axis in $[180^\circ, -180^\circ]$ should be discarded. Thereby, the DrillSample is always aligned to the horizontal screen with the first object that was hit positioned on the left side (see Figure 3.1b). Furthermore, the 1:1 mapping between device and DrillSample orientation proved to be too cumbersome to inspect the objects from their back sides. Therefore, it was found to be useful to speed up the rotation around the yaw axis 3-times and around the pitch axis 1.5-times.

To provide a reasonably refined visualization of the virtual clones, there are two critical aspects that are discussed in the following sections:

1. The length of the DrillSample line needs to be optimized while preventing intersection of the clones and preserving their relative distances.
2. The optimal Z-Position of the DrillSample to the virtual camera must be obtained.

Algorithm 3.1: DrillSample selection technique in pseudo code.

```

Set: selectedObject  $\leftarrow$  NULL;
Set: objectConfirmed  $\leftarrow$  false;
Set: list of hit objects  $S(O) \leftarrow \emptyset$ ;

Step 1: Target Indication;
Detect tap point  $p_t \in \mathbb{R}^2$  and perform raycast along  $\vec{r}(P_c, \vec{d}_{ray})$ ;
Obtain  $S(O_i)$ ,  $i = 1 \dots N$ ;
if  $S(O) > 1$  then
    Step 2: DrillSample Construction;
    Calculate and optimize DrillSample length  $l$  (Section 3.3.4.1);
    Calculate pivot point  $p_{piv}$  at the center of all hit-points  $p_{hit,i}$ ;
    Rotate objects in  $S(O_i)$  around  $p_{piv}$  so that  $l \parallel$  image plane's horizontal axis;
    Perform Z-Positioning of the DrillSample (Section 3.3.4.2);

    Step 2a: DrillSample Inspections;
    while objectConfirmed = false do
        if Rotation of Mobile Device then
            | Map  $orientation_{device}$  to DrillSample around  $p_{piv}$ ;
        end
        if Horizontal Swipe Gesture then
            | Obtain tap point  $p_s \in \mathbb{R}^2$  and swipe direction  $\vec{d}_s$ ;
            | Use  $\vec{d}_s$  to travel along the DrillSample if it spans multiple screens;
        end
        if Vertical Swipe Gesture then
            | Obtain tap point  $p_s \in \mathbb{R}^2$  and swipe direction  $\vec{d}_s$ ;
            | Use  $\vec{d}_s$  to zoom in/out (Section 3.3.4.2);
        end

        Step 3: Object Confirmation;
        Detect tap point  $p_t \in \mathbb{R}^2$  and perform raycast along  $\vec{r}(P_c, \vec{d}_{ray})$ ;
        Set: objectConfirmed  $\leftarrow$  true;
        Set: selectedObject =  $S(O_{sel})$ ;
    end
    Set:  $S(O) \leftarrow \emptyset$ ;
    Set: DrillSample = NULL;
else
    | Set: selectedObject =  $S(O_1)$ ;
end

```

3. 3D SELECTION IN HANDHELD MIXED REALITY

3.3.4.1 Length of the DrillSample Ray

Since the relative distance of objects to each other is sufficient to preserve the spatial context, the real length of the ray should be scaled for visualization to provide an optimal overview. If the objects are far away from each other, the ray might be shortened, or stretched to reveal objects that are inside of another (e.g. a ball in a bucket). The optimal amount by that the ray should be scaled depends on the shortest distance between the convex hulls of the two neighboring objects along the direction of the ray. For objects with overlapping hulls, the distance is specified as a negative value and positive otherwise. Assuming n objects on the DrillSample and the shortest distance between $(n - 1)$ neighbors is denoted by d_i , the length of the ray x is then computed by

$$x = -d_i * (n - 1) \quad (3.1)$$

The precise calculation of these distances can be computationally costly, especially in dense environments with complex shapes. To minimize the computational load, we chose an approximation with linear complexity by treating all objects as spheres (see Figure 3.5) with the maximum extent of the objects' bounding box used as its radius and the hit point as its center.

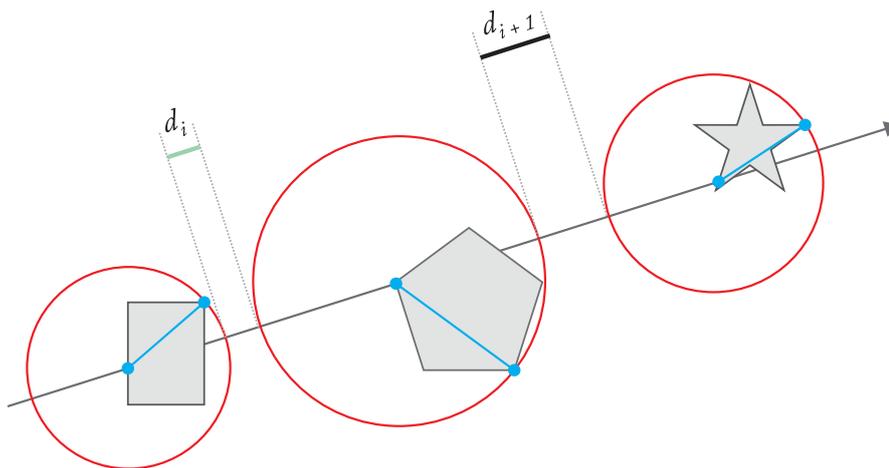


Figure 3.5: Sphere approximation of clones' size to calculate the optimal ray length.

For objects whose center point is not close to the ray or have a complex concave shape, this may not be visually pleasing as it overestimates the real distance between neighboring objects. More elaborate algorithms can be employed in the future to enable an optimal adjustment of the length.

3.3.4.2 Z-Position of the DrillSample

The proposed algorithm visualizes an overview of all ambiguously indicated objects. Depending on the spatial properties of the objects and their relation to each other, this can result in the following challenges.

1. **The larger the distance between clones varies**, the lesser the DrillSample ray can be compressed. To provide an overview of all clones on one screen, the ray must be positioned at greater distance to the virtual camera. This could result in small clones being barely visible.
2. **The more the size of the clones varies**, the less likely there is a distance to the virtual camera at which all clones are nicely visible. Small objects may appear too small or big objects might be clipped at the near image plane.
3. **The more objects are selected**, the less likely the overview provides a meaningful starting point for refinement, as the clones in the overview appear too small, as in 1.

Thus, the distance between the virtual camera and the DrillSample depends on the size of the clones and their relative distances to each other. The distance D_{ov} of the virtual camera to obtain an adequate overview can be calculated as denoted in Equation 3.2.

$$D_{ov}(B_{DSS}) = \frac{exp}{\tan(fov * 0.5)} + B_{DSS}(z), \quad (3.2)$$

where

$$exp = \begin{cases} B_{DSS}(y) & \text{if } R_B < R_{fov} \\ B_{DSS}(x) & \text{if } R_B \geq R_{fov} \end{cases}$$

$$fov = \begin{cases} fov(y) & \text{if } R_B < R_{fov} \\ fov(x) & \text{if } R_B \geq R_{fov} \end{cases}$$

While $B_{DSS} \in \mathbb{R}^3$ is the DrillSample's axis-aligned bounding box represented as an expansion vector, R_{fov} is the aspect ratio of the virtual camera's field of view, R_B the aspect ratio of the bounding box's side facing the camera and fov is the field of view of the virtual camera. Additionally it has to be ensured, that neither the near nor the far clipping plane of the virtual camera are violated. The interval, in which users may modify the distance of the camera to the DrillSample with a vertical swipe gesture (see Figure 3.2e), is then limited to $[D_{ov}(B_C), D_{ov}(B_{DSS})]$ by the bounding box of the biggest clone B_C on the DrillSample. It can be noted, that depending on a specific application, other schemes for the computation of D_{ov} might be suitable.

3.4 Performance Studies

For a comprehensive evaluation of the proposed selection technique, a summative evaluation was conducted by comparing *DrillSample* with the two baseline techniques *Mobile Raycasting* and *Expand*, as described in Section 3.4.1, across three different selection scenarios with different variations of object density and visibility.

3. 3D SELECTION IN HANDHELD MIXED REALITY

3.4.1 Baseline Techniques

Most of the virtual pointing techniques that are discussed in Section 2.2.2 are originally not designed for handheld mixed reality environments, while popular multi-touch selection techniques aim at selecting 2D objects. Thus, a direct comparison of these techniques is hard to obtain. Related work [143] introduces a qualitative evaluation of 3D selection techniques in handheld 3D environments. For performance analysis, they propose an adaption of Go-Go using swipe gestures to adjust the virtual arm length and multi-touch input to select an object. However, this adaption changes the direct mapping between virtual hand, arm length and target object of the original algorithm and does not apply for a clean and fair performance evaluation of selection techniques in handheld mixed reality. For the summative evaluation of DrillSample, Raycasting [62] and Expand [119] were chosen as baseline techniques since they are both applicable in dense environments, they can be adapted to function in one-handed handheld MR without changing the original mapping characteristics during interaction and both fulfill the requirements from Section 3.1. Furthermore, Expand is a two-step technique as well and thus acts as a valid baseline for performance measurements regarding selection speed and number of interaction steps.

3.4.2 Adaptions for Handheld Mixed Reality

For the study, the adapted Mobile Raycasting from Section 3.3.2 was employed using it in its single selection mode. As described in Section 2.2.2.2, Expand is a two-step technique in which virtual scene objects are selected using Cone-Casting [18] and are presented in a second refinement step aligned on a virtual grid for object confirmation. To use it for one-handed mixed reality, one-finger tap gestures are employed within the three phases of our *Mobile Expand* adaption that can be described as follows. For object indication, a cone cast is performed, similar to the Mobile Raycasting approach, using a single tap on the device's screen to indicate the cone's direction. All objects intersecting the cone are subject of a second refinement step. This second step is preceded by a non-interactive animation showing the objects moving from their original positions upon cone-casting to their designated positions on the virtual grid. During the refinement step, all casted objects are presented aligned on a grid in front of a solid gray background. The selection is confirmed by a tap gesture above the desired object. Since the original publication [119] does not provide detailed information about the grid alignment, the following positioning onto the grid was performed for *Mobile Expand*:

1. For i objects intersecting the cone, a dynamically sized grid is created with $m \geq 4i$ cells to provide a sufficient number of positions to resemble the original context of the casted objects. For visualization, each cell is represented by its center point $c_i \in \mathbb{R}^2$ on the screen.
2. For each object i , its projected 2D screen position $p_i \in \mathbb{R}^2$ is calculated.

3. For each screen position p_i , its closest c_i is determined by evaluating the euclidean distance $\|p_i - c_i\|$ and the object i is placed into its calculated cell with center point c_i .

For purposes of simplification, the mobile adaptations hereinafter are referred as *Raycasting* and *Expand*.

3.4.3 Objectives

The main goal of the experiment is to evaluate the performance and ease of use of Drill-Sample compared to competing techniques. In this study, we focus on selection of objects in closer range in dense environments. A second objective is to examine the performance of the spatial context preservation of our proposed algorithm in environments with objects of high visual similarity. In designing the experiment, we formulate the following hypotheses:

- H1** Raycasting will be best suited for non-occluded objects.
- H2** Expand and DrillSample will perform considerably better than Raycasting in environments with overlapping, partly occluded or invisible objects, which differentiate significantly in appearance, in terms of speed and precision.
- H3** Expand will suffer in environments with objects of high visual similarity. Likewise, DrillSample will perform considerably better than Expand in terms of speed and precision.

3.4.4 Experimental Design and Procedure

We conducted the study using a 3x3 within-subjects factorial design where the independent variables are selection technique and task scenario.

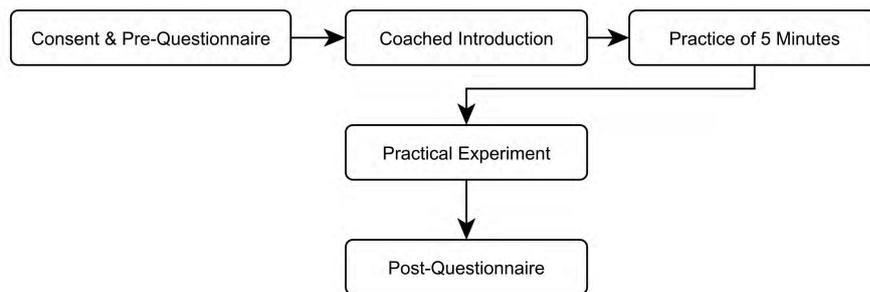


Figure 3.6: User study procedure.

The selection techniques are Raycasting, Expand and DrillSample, while the scenarios included three different experimental tasks with varying selection conditions in close range. The dependent variables are *Task Completion Time* and *Number of Selection*

3. 3D SELECTION IN HANDHELD MIXED REALITY

Steps. Task completion time represents the time it takes to successfully finish a specific scenario from the time the user started it, while number of selection steps comprises the amount of necessary object selections to successfully finish a selection task. This measure indicates precision of the applied technique. Furthermore, we measured user preferences for both technique in terms of speed, accuracy, and ease of use. The user study procedure is depicted in Figure 3.6.

The user study consists of a pre-questionnaire followed by a practical test and a post-questionnaire. The material of the study is presented in Appendix VI.A. It took approximately 25 minutes for each participant to finish the user study.

No.	Question
Q1	What is your gender?
Q2	How old are you?
Q3	About how often do you play video games?
Q4	What percentage of your gaming is playing mobile 3D games?
Q5	Do you have a multi-touch Smartphone?
Q6	Do you have any flexibility or pain issues with your primary hand, fingers or arm?

Table 3.1: Pre-Questionnaire

No.	Question
Q1	How adequate do you feel the time allotted for practice was?
Q2	How comfortable were you with using a smartphone for task completion?
Q3	How would you rate the RAYCAST selection technique in terms of usability? Speed? Accuracy?
Q4	How would you rate the EXPAND selection technique in terms of usability? Speed? Accuracy?
Q5	How would you rate the DRILLSAMPLE selection technique in terms of usability? Speed? Accuracy?
Q6	Rank the three selection techniques in order of desired use (with 1 being the most desired).
Q7	When determining how much you like using a selection technique, how important in influence on your decision was usability? Speed? Accuracy?
Q8	Regarding the visualization during the refinement process of the DRILLSAMPLE technique, how helpful and useful was the linear arrangement for spatial visualization?

Table 3.2: Post-Questionnaire

At the beginning of the study, each participant was asked to read and sign a stan-

dard consent form and to complete a pre-questionnaire, as described in Table 3.1. Upon completion, the participant was given a detailed description of the practical part about "Selection in Handheld Mixed Reality". A tutor coached them on how to use the handheld device and how to perform selection in the testing environment. Afterwards, each participant had five minutes time to practice the three selection techniques. Once they started the study, they were not interrupted or given any help. Upon completion of the practical part, they were asked to fill out a post-questionnaire (see Table 3.2).

Of the 28 participants ranging from 23 – 38 years, 12 were female and 16 male. 12 users had no experience of playing mobile 3D games and 7 had no experience with smartphones. One person reported to have occasionally severe pain issues in her/his primary hand's wrist. All 28 participants yielded successful simulation trials from which all data was used for analysis.

3.4.5 Implementation

All computations – tracking, rendering, selection and manipulation of virtual objects – are performed on a smartphone using Android OS. For developing and testing the proposed interaction techniques, the Virtual and Augmented Reality Framework ARTIFICe is used that is described more in detail in Part IV of this thesis. To access touch inputs on the mobile device screen for triggering interaction, ARTIFICe uses Unity's [167] built-in Android interface to access the hardware layer. 6DOF pose data from Vuforia [163] is processed by the specific interaction technique (IT) and handed to the ARTIFICe interaction framework. Using ARTIFICe's interaction interface, all required selection techniques (DrillSample, Mobile Raycasting Mobile Expand) as well as manipulation techniques (3D Touch, HOMER-S) for the performance study in Section 4.2 have been implemented.

The practical test ran on a Samsung Galaxy S II I9100, featuring an Arm Cortex A9 Dual Core-Processor, a 4.27" WVGA multi-touch display and an 8 megapixel camera. Galaxy S II weighs 116g and has the physical dimensions of 125.3 x 66.1 x 8.49 mm. The phone was protected with a market available hard cover to minimize the problem of canceling the simulations by mistake by pushing the buttons on the side.

3.4.6 Test Scenarios

We built three different scenarios to cover different selection situations in dense 3D environments. They ranged from unique and un-occluded to non-distinguishable and fully occluded object selection tasks. Thus, we used occlusion and visual similarity as variables for task design. As the underlying building block [6] for interaction design, we applied the canonical task *Selection*, which refers to the task of acquiring a particular object from the entire set of objects available (see Section 2.1.1.1).

All scenarios are based on the same virtual working ground (black & white textured plane) that was printed to paper at 56x40cm and acted as a visual planar marker for the natural feature tracking toolkit [163]. The marker was placed on a table that was positioned at the center of a room so that users had around 150cm of obstacle free space to

3. 3D SELECTION IN HANDHELD MIXED REALITY

work within. All 28 users completed the three scenarios in random order. Each scenario featured a simple description of the upcoming task. The participants could inspect the scenario, without being able to manipulate it, in order to understand the task according to its description before starting with the actual test.

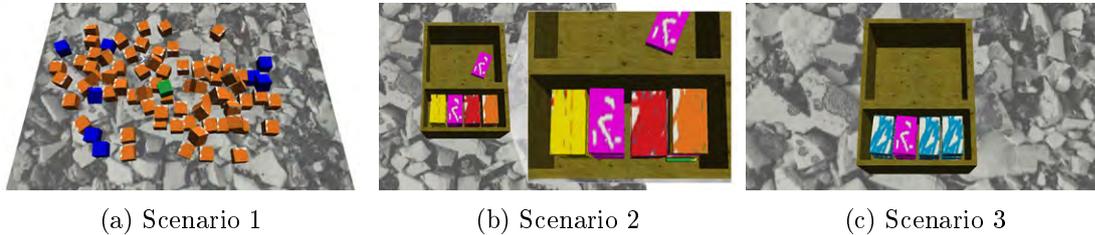


Figure 3.7: The three test scenarios of the performance user study.

The three scenarios are depicted in Figure 3.7 and are defined as follows:

Scenario 1: Unique Object & No Occlusion The user was challenged to select a green cube in the middle of the working ground which was cluttered with around 80 other cubes of the same size but of different color (see Figure 3.7a). The targeted object was easy to distinguish and not occluded by any of the objects in the scene. As soon the user selected or confirmed the selection of the green cube, the task finished automatically.

Scenario 2: Unique Object & Strong Occlusion The user had to select a green brick in the lower right corner of a wooden textured box (see Figure 3.7b). The box contained four stacks of different colored equally sized bricks. The targeted object was located on the very bottom of the last stack and it was the only brick that was colored in green. Although it was easy to distinguish, it was hardly visible due to the strong occlusion of the bricks stacked on top of it and the box’s walls. Again, on selection of the targeted object, the task finished automatically.

Scenario 3: Not-Unique & Strong Occlusion In this scenario the user had to select a brick from a wooden textured box again (see Figure 3.7c). The box contained four stacks of equally sized bricks. All bricks were colored in light blue except for the bricks of the second stack which had a magenta colored texture. The targeted object was located on the very bottom of the magenta colored stack. It was only distinguishable by its position in the stack and was hardly visible due to strong occlusions of the bricks stacked on top of it and the box’s walls. The number of bricks on top of the targeted object varied randomly for each participant from four to seven pieces.

3.5 Experimental Results

Based on the performance study, we conducted an evaluation on the quantitative data to examine performance of the three techniques and a subjective evaluation regarding

user’s preferences and feedback.

3.5.1 Quantitative Evaluation

The quantitative data gathered from the questionnaires and automatically collected data of the test application were analyzed with Friedman’s χ^2 test [123, 106] and repeated measures single factor ANOVA [55] accordingly on both *Task Completion Time* and *Number of Selection Steps* (see Section 3.4.4) as well as for each scenario (see Section 3.4.6). When suitable, we further employed post hoc analyses using pairwise *t-tests* or *Wilcoxon signed rank test* [123, 106] with the Holm’s sequential Bonferroni correction [7]. We focused on two different aspects during data analysis. Firstly, data of all participants regarding selection techniques was evaluated and secondly, we analyzed the techniques’ performance depending on tasks.

3.5.1.1 Performance Evaluation

The evaluation of the completion time shown in Figure 3.8 indicates significant differences for the three interaction techniques with ($F_{2,54} = 6.74, p < 0.00243$) for all tasks on average, but also with ($F_{2,54} = 9.27, p < 0.00035$), ($F_{2,54} = 21.84, p < 1, 1e - 7$) and ($F_{2,54} = 4.91, p < 0.011$) for the tasks 1-3 separately.

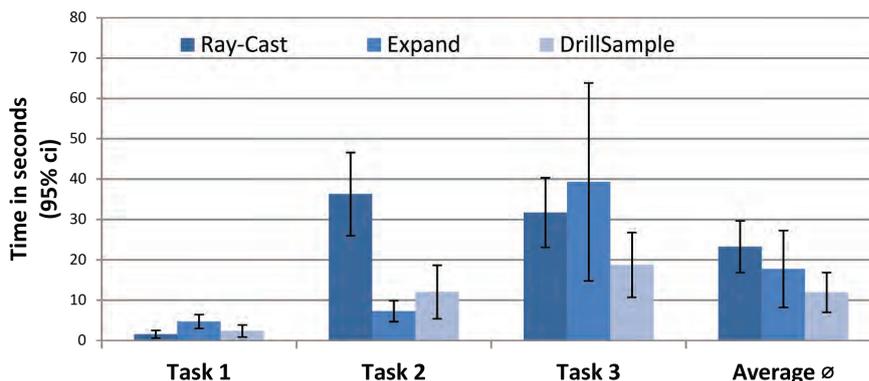


Figure 3.8: Mean completion time per task and on average.

The pairwise t-test shows, that only DrillSample is significantly faster than Raycasting with ($t_{27} = 4.33, p < 0.00018$) in the overall mean completion time. For task 1, the techniques Raycasting and DrillSample score significantly better than Expand with ($t_{27} = -3.82, p < 0.0007$) and ($t_{27} = 2.65, p < 0.0134$). Most likely because Expand uses a cone-cast to select objects, which results more often in a refinement-step compared to DrillSample that casts a ray. No significant difference was measured between Raycasting and Drill-Sample. For task 2, the techniques with an additional refinement step proved to be faster than Raycasting with Expand at ($t_{27} = 7.8545, p < 1.9e - 8$) and DrillSample at ($t_{27} = 3.73, p < 0.0009$), however no significant difference between DrillSample and Expand could be found. Here, Raycasting forces the user to successively

3. 3D SELECTION IN HANDHELD MIXED REALITY

select and put objects away until the desired object is easily accessible which results in a very time-consuming problem. In task 3, the users required significantly less time when using DrillSample, compared to Raycasting ($t_{27} = 3.24, p < 0.0031$) or Expand ($t_{27} = 2.6, p < 0.0148$). Raycasting fails as it did in task 2 because both problems force the user to move objects out of view step by step. Expand scores much worse than in task 2 because the targeted object cannot be distinguished out of its spatial context and because Expand is only aligning the objects to a two-dimensional grid. Between Raycasting and Expand no significant difference could be found. Significant differences can be seen in Figure 3.9 for the results of the number of selections for task 2 and task 3 as well as on average, each with ($F_{2,54} = 10.98, p < 0.0001$). Task 1 shows no significant differences at ($F_{2,54} = 0.491, p < 0.615$) and advises that all selection techniques perform well in the simplest case.

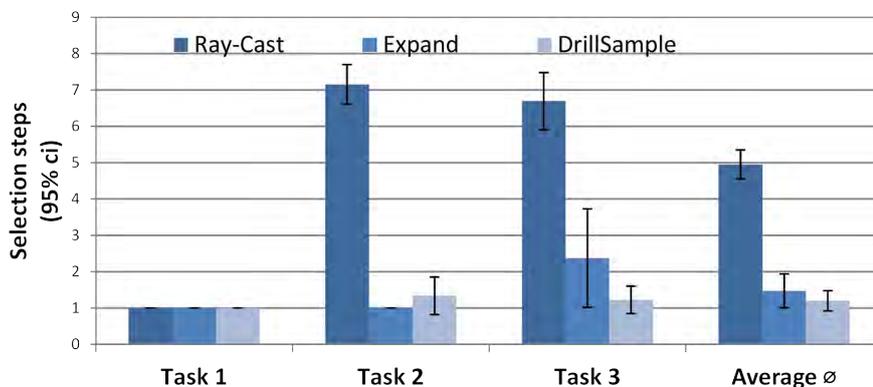


Figure 3.9: Mean selection steps per task and on average.

The pairwise comparison for selection steps on average found the techniques Expand ($t_{27} = 15.29, p < 8.04e - 15$) and DrillSample ($t_{27} = 18.83, p < 4.7e - 17$) to be significantly better than Raycasting, but no significance among another at ($t_{27} = 1.31, p < 0.2$). Similar to the task completion time, the number of selection steps in task 2 were significantly smaller for Expand at ($t_{27} = 18.4512, p < 7.78e - 17$) as well as for DrillSample with ($t_{27} = 13.93, p < 7.55e - 14$) compared to Raycasting, but also Expand ($t_{27} = -2.2, p < 0.036$) appears to be slightly less error-prone than DrillSample. Expand benefits at task 2 from the fact that the targeted object is easily distinguishable, but also from its coarse selection volume where techniques casting a ray may have a hard time to hit an object that is only slightly visible. In task 3, likewise for average completion time, we found DrillSample having less false selections than Raycasting ($t_{27} = 16.87, p < 7.29e - 16$) and Expand ($t_{27} = 2.61, p < 0.0146$). Additionally, Expand is significantly better than Raycasting at ($t_{27} = 8.34, p < 6.01e - 9$), too. A possible cause for Expand scoring worst in terms of completion time, but not on number of false selections could be that each refinement step costs extra time for the visualization, but also allows users to accidentally choose the targeted object each time.

In average, DrillSample outperforms Raycasting and Expand, both in completion

time as well as in number of selection steps.

3.5.2 Subjective Evaluation

Besides the performance measures based on quantitative data, we also examined the user's subjective evaluation on speed and accuracy of each technique. Furthermore, we also include the abstract performance value "ease-of-use" [32] to further evaluate the capabilities of the underlying technique. When answering the questions Q1-Q5, Q7 and Q8, users were able to choose from a 7-point Likert scale [2]. While all questions feature the highest rating at seven, and the lowest at one, Q1 states the best rating with four (appropriate).

The participants found the time allotted for practice appropriate with ($\mu = 3.93$ and $\sigma = 0.25$ at $\alpha = 0.05$). Using a smartphone to complete the different tasks was rated to be moderately comfortable with ($\mu = 5.72$ and $\sigma = 0.98$ at $\alpha = 0.05$). As depicted in Figure 3.10, all three techniques were rated at least above average but with significant differences regarding speed ($\chi_2^2 = 10.48$, $p < 0.0053$), ease of use ($\chi_2^2 = 9.53$, $p < 0.0085$) and accuracy ($\chi_2^2 = 15.27$, $p < 0.00048$).

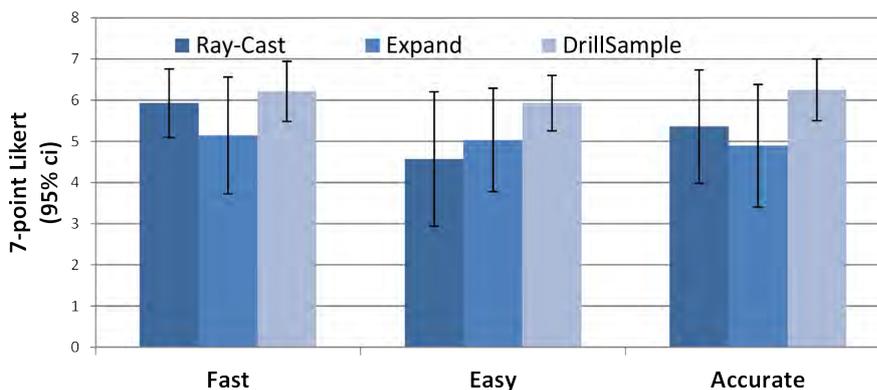


Figure 3.10: Users' average rating of Q3, Q4 and Q5.

Only DrillSample was found to be significantly faster than Expand in the pairwise comparison ($Z = -2.63$, $p = 0.0085$). Due to the Bonferroni adjustment, Raycasting failed to be significantly faster than Expand with ($Z = -2.088$, $p = 0.0368$). Raycasting was not found to be significantly different from DrillSample ($Z = -1.0558$, $p = 0.29108$). Expand was likely rated lower than the other techniques because it triggers refinement too often, while DrillSample only asks for refinement if objects overlap. Using Raycasting, users are not interrupted by a refinement step and might therefore consider it faster. Users' ratings on ease of use found DrillSample significantly better than Raycasting and Expand at ($Z = -2.84$, $p < 0.0045$) and ($Z = 2.91$, $p < 0.0036$). Raycasting was insignificantly different to Expand with ($Z = -0.89$, $p = 0.371$) even without the Bonferroni adjustment. Similarly, users found DrillSample significantly more accurate than Raycasting ($Z = -2.69$, $p < 0.007$) and Expand ($Z = -3.17$, $p < 0.0015$). Likewise

3. 3D SELECTION IN HANDHELD MIXED REALITY

Raycasting showed no significant difference to Expand at ($Z = -1.23, p = 0.218$). Both Raycasting and Expand are not easy to use or accurate, if objects are occluded or look very similar. Hence, both factors result in a tedious, and when using Expand even a confusing, sequence of interactions to select the desired object.

For question Q6, asking the participant to rank the selection techniques in order of desired use, significant rankings for 1st ($\chi_2^2 = 18.5, p < 9.6e - 005$), 2nd ($\chi_2^2 = 12.29, p < 0.0021$) and 3rd ($\chi_2^2 = 9.91, p < 0.007$) could be found as shown in Figure 3.11

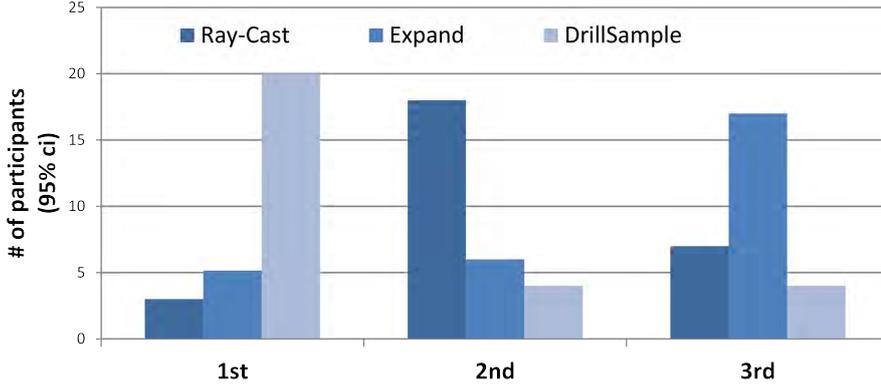


Figure 3.11: Users' rating of Q6.

Rank one was clearly given to DrillSample with ($Z = -3.54, p < 0.00039$) and ($Z = -3, p < 0.0027$) significantly outranking Raycasting and Expand. Rank two was given to Raycasting with ($Z = -2.98, p < 0.0028$) and ($Z = -2.45, p < 0.014$) significantly outranking DrillSample and Expand. Rank three seems to be given to Expand, however it only significantly outranks DrillSample with ($Z = -2.83, p < 0.004$) but not Raycasting at ($Z = -2.04, p = 0.041$) due to the Bonferroni adjustment. All other pair-wise interaction technique tests show no significant difference.

Users stated all aspects of Q7 evenly important with 6 (important) or higher when answering Q6. Addressing in Q8, how helpful the spatial visualization is, the participants found it useful to very useful with ($\mu = 6.5$ and $\sigma = 0.1$ at $\alpha = 0.05$).

3.5.3 Qualitative Evaluation

Based on the 3D formalization principles by [32], we outline a number of factors for the canonical interaction task *3D Selection* that influence performance in virtual environments. Since all three evaluated selection techniques are suited or explicitly designed for dense environments, we do not include "density" as a performance factor. The specified factors are:

1. Object Size: This object property is related to the geometric area, a 3D object covers on the output device screen. A selection technique must be capable to select objects of varying size.

2. Occlusion: In any virtual environment, but especially in a dense environment, objects can partially or fully occlude each other which may result in invisible objects. In such environments, selection must be precise and provide some assisting visualization to identify occluded objects.
3. Visual Appearance: The visual appearance of virtual objects can be of high similarity. Identifying the desired target object can result in problems in dense environments with occluded objects. In such environments, selection must provide an assisting visualization to disambiguate the desired object.

Based on the results from quantitative as well as subjective evaluation, we summarize our findings with respect to the proposed parameters in Table 3.3.

	Parameters		
	Object Size	Occlusion	Appearance
Raycasting	– [119, 62]	–	○
Expand	+ [119]	+	–
DrillSample	–	+	+

Table 3.3: Evaluation of selection techniques in handheld mixed reality.

Previous work [119, 62] report that Raycasting performs badly for objects covering only a small portion of the screen, while Expand performs well for the same case by casting a volume instead of a single ray. Beyond that, our findings indicate that Raycasting is well suited for selecting non-occluded objects which can be also similar in appearance. However, if the desired object is small and is located amongst similar looking objects, imprecise touch input can evoke wrong selection. Compared to Raycasting, Expand is well suited to select visible or fully occluded objects of varying size. But the grid representation during the refinement step does not provide full spatial correspondence to the original position of the selected objects; hence, precise selection of an object from a set of similar looking objects can be difficult and can result in wrong selections. DrillSample also lacks accuracy when selecting small objects due to the underlying use of Raycasting in combination with the imprecise single touch input. However, since DrillSample selects all objects which are cast by the ray, overlapping or occluded objects can be precisely selected due to DrillSample’s refinement step. Here, spatial context preservation provides a full overview that allows object disambiguation, which is especially of interest when selecting from a set of similar looking objects.

3.6 Discussion

We designed the experiment to compare three different techniques in terms of speed, precision and ease-of-use for performing 3D selection tasks with a multi-touch handheld device in a dense mixed reality scene. Many of the outcomes of our performance study

3. 3D SELECTION IN HANDHELD MIXED REALITY

were statistically significant which enable us to draw multiple meaningful conclusions. In H1 we proposed Raycasting to be best suited for selection of non-occluded objects. Results of completion time for task 1 support H1, since Raycasting significantly outperforms Expand. H1 can further be strengthened by taking the subjective evaluation into account where users considered Raycasting to be fast. DrillSample also performed significantly better than Expand for task 1. This indicates the strength of techniques casting a ray instead of casting a cone for visible object selection in close range, since a ray selects fewer objects. Thereby, just a few objects need to be presented at DrillSample's refinement step, while Cone-Casting is always coarser. There, more objects are presented during a refinement step, which takes more time for a user to get an overview before indicating the desired object. Therefore H1 can be supported to be true in terms of speed. Regarding precision, neither performance nor subjective evaluation revealed statistical significance to back up H1. Therefore, we must state H1 to be not true in terms of precision.

Results for evaluating speed and precision, when selecting almost fully occluded objects, clearly reveal Expand's and DrillSample's strengths. Both perform significantly faster and need less selection steps than Raycasting, which supports H2. Since no significant difference in completion time and interaction steps between Expand and DrillSample could be found, H2 can be backed up further. These results indicate that Expand and DrillSample are both equally suited for selecting an occluded object, which highly differs in appearance from the surrounding ones. Regarding precise selection of occluded objects with high visual similarity, DrillSample significantly outperforms both baseline techniques in terms of completion time and number of interaction steps. Based on these results, H3 can clearly be supported. It proves the advantage of our proposed spatial context preservation compared to the grid representation that Expand provides. The disadvantage of Expand's detailed visualization becomes even more apparent, since no significant difference in completion time could be found between Expand and Raycasting.

Regarding users' preference, the subjective evaluation clearly reveals users' being in favor of DrillSample. It significantly outranked both baseline techniques when users were asked for an overall ranking. This first rank can further be confirmed when looking at the details. Users ranked DrillSample highest in terms of speed, precision and ease-of-use. It significantly outperformed Expand in terms of speed, but not Raycasting. Since Raycasting does not provide a refinement step, it tends to be considered fast and "direct". The DrillSample's capability to precisely select the desired object over all three test scenarios was ranked significantly best in terms of precision. Finally, the users ranked DrillSample significantly best in ease-of-use. Based on these results and findings, we have developed a set of basis guidelines regarding object selection in closer range:

- Raycasting remains a good alternative selection technique for sparse selection tasks and as long as objects are fully visible.
- Expand remains a good alternative for visible or occluded objects of varying object size, as long as they differ in visual appearance.

- For visible or occluded objects, independent of their visual appearance, DrillSample is the best general purpose method.

3.6.1 Variations of the Algorithm

DrillSample is originally designed for multi-touch displays which allow for one-finger or two-finger processing. Tracking two independent contacts of the surface is only necessary for optional interactions in the DrillSample visualization view, thus the algorithm can be applied in various kinds of virtual environments with just one 2D or 3D interaction device. For example in a fully immersive environment, the user's interaction device can be used for Raycasting and object confirmation. Since the DrillSample visualization does not depend on display size but on the field of view (FOV) of the user's output device, such as a Head Mounted Display (HMD), the Image-Plane technique [62] can be applied to show the indicated objects in front of the user in space. Furthermore, the rotation of the interaction device can be mapped to rotate the DrillSample for inspection. As described, only a few minor changes of the original algorithms are necessary to apply the technique in another type of mixed reality environment without changing its original mapping characteristics.

Chapter 4

3D Manipulation in Handheld Mixed Reality

After a virtual object has been selected it can be subsequently transformed by translating, rotating and scaling it. However, using 2D touch input for 3D manipulation induces several problems, as described in Section 1.1. To address the limitations of existing 3D manipulation techniques for handheld mixed reality scenes, the two novel methods *3DTouch* and *HOMER-S* are presented which both support RST manipulations.

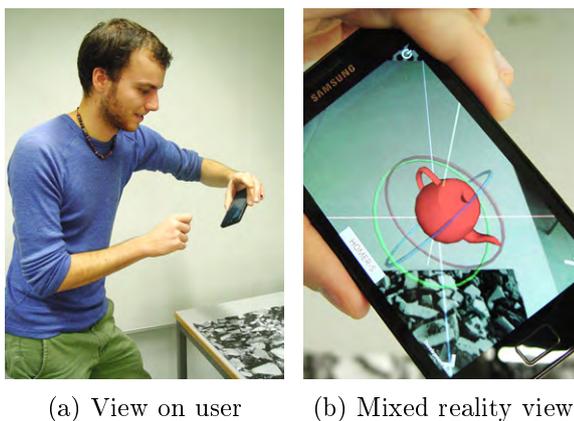


Figure 4.1: Touchless full 6DOF object manipulation using HOMER-S.

3DTouch provides 3D translation and rotation as well as non-uniform scaling by combining simple 2D touch gestures with the handheld's current 6DOF pose. The 6DOF manipulation is decomposed into two separate tasks where one-finger is sufficient to access all three 3DOF during translation and rotation. Scaling requires only a two-finger pinch gesture while providing non-uniform transformation in all three dimensions. *HOMER-S* pushes the idea of enabling intuitive 3D manipulation in handheld mixed reality further and aims on interaction beyond the (limited) screen dimensions by decoupling the manipulation process from any touch input. It is based on the immersive

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

VR technique HOMER (see Section 2.3.1) and maps the handheld's pose, regarded as the virtual hand, onto the object upon selection. The 6DOF access is exploited for full 6DOF manipulation, as illustrated in Figure 4.1, and 3D non-uniform scaling.

Compared to existing state-of-the-art methods, the novel techniques aim at improving intuitiveness and ease-of-use by reducing user touch input complexity and adapting real-world metaphors for object manipulation.

4.1 Methodological Approach

There are different findings in recent literature regarding DOF separation and integration to improve intuitiveness and ease-of-use for object manipulation. [104] states that DOF integration does not necessarily mean that the performance for orientation tasks is increased. This, however, contradicts the findings in [126]. The authors observe reduced interaction performance and user satisfaction for DOF integration for translation and rotation tasks. It is rather proposed to follow the structure of the input device than the task structure when designing the interaction technique. As the applied interaction device offers two different input structures, this proposition is the fundamental foundation of the two novel manipulation techniques. *3D Touch* follows DOF separation by employing the 2D multi touch structure of the input device while *HOMER-S* aims at DOF integration re-using the 6DOF information of the device pose.

4.1.1 Requirements & Prerequisites

To achieve the research objective from Section 1.2, the same requirements as for 3D selection have to be met by the 3D manipulation technique, as specified in Section 3.1. For both techniques, prior objects selection using i.e. *Mobile Raycasting* or *DrillSample* is assumed.

4.1.2 Design Guidelines

Based on the presented motivation and requirements, the following design guidelines were specified which were applied during algorithm development of both techniques:

Keep Direct Touch Abilities The probably most appealing aspect of touch displays is the ability to directly "touch" an object in order to interact with it. We aim on preserving this ability and do not introduce any offsets or non-direct gestures.

Simplify Touch Input Since multi finger interaction requires prior knowledge for correct usage of the touch gesture and can be hard to apply with only one hand, we aim to simplify touch gesture complexity for object manipulation. If necessary, we introduce degree-of-freedom separation to fulfill this guideline as well as mode switches to perform RST operations. Furthermore, we aim at adapting real world metaphors for touchless object manipulation.

4.1.3 The 3D Touch Technique

Following the design guidelines, the direct mapping between finger touches and virtual touch points is preserved in the proposed 3D Touch technique. According to [126], the separated structure of the input device is matched to the technique design by separating integral 3D manipulations into 3DOF entities for rotation, scaling and translations (RST). A mode switch is employed to change between the three manipulation entities at run-time, as described in Section 4.1.5.1. To comply with the requirement of limited gesture complexity when manipulating the remaining DOFs, simple 2D multi-touch manipulation gestures as in Hancock [72] are combined with the 6DOF device pose. Inspired by Reisman [103], the 2D screen coordinates of the touch input are transformed to 3D space. Thereby, 3D Touch is able to solely rely on one-finger (translate and rotate) or two-finger (scale) gestures to allow non-uniform scaling. In contrast to [143], one gesture for each 3DOF entity is sufficient to enable non-uniform manipulations without requiring a manual switch to address each dimension. Thereby, our proposed approach results in a minimal set of necessary gestures, each having a low complexity.

With the described methodology, our proposed approach features the seamless transition between the different DOF subtasks to fulfill each 3DOF manipulation task. To access all 3DOF of each RST task, neither an abstract switch, such as a button, nor applying a distinct gesture for each subtask is necessary. Since the user naturally changing his viewpoint in a handheld mixed reality setup, the provided handhand's pose and resulting perspective onto the virtual objects can be seamlessly exploited to obtain the accessible DOF at a moment in time. In the following paragraphs, algorithmic details of the described manipulation process are given. Upon selection, the 6DOF pose of the selected object $obj(R, T) \in \mathbb{R}^3$ is stored and the handheld's device pose $pose(R, T) \in \mathbb{R}^3$ is continuously updated.

4.1.3.1 Translation

3D translations are performed using single touch inputs that are combined with the current $pose(R, T)$. First, at two moments in time t , consecutive touch points $p(t_1), p(t_2) \in \mathbb{R}^2$ are projected from 2D screen into 3D world space, as described in Section 3.3.2, but with a specific distance d , resulting in the 3D points $P(t_1), P(t_2) \in \mathbb{R}^3$. The distance $d = ||pose(T) - obj(T)||$ is obtained upon selection, where $||...||$ denotes the Euclidian norm. Both points form the vector $\vec{v}(P(t_2), P(t_1))$ that is subsequently normalized, denoted as \hat{v} . To determine the current interaction dimension, the collinearity between \hat{v} and the normalized target coordinate system's basis vectors in world coordinates $\hat{e}_i \in \mathbb{R}^3, i = x, y, z$ is calculated by $c_i = \hat{e}_i \cdot \hat{v}$. The basis vector \hat{e}_i with the highest resulting scalar $|c_{max}| \in c_i$ indicates the dimension \hat{e}_{max} that is subsequently used for translation. The sign s of c_{max} determines the direction of the manipulation. Given the objects position $obj(T)$, the objects manipulated position $obj(T)'$ is obtained by

$$obj(T)' = obj(T) + (s \cdot ||\vec{v}|| \cdot \hat{e}_{max}). \quad (4.1)$$

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

Subsequently, the 3D position of the selected object is adjusted. In Figure 4.2, some example translations using the 3DTouch algorithm are illustrated.

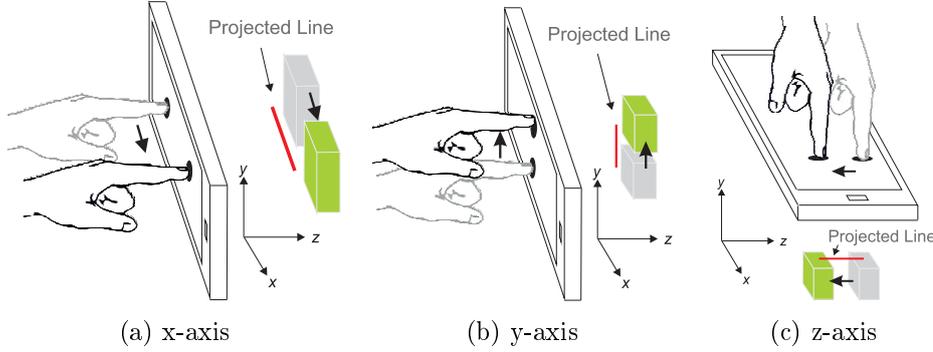


Figure 4.2: Examples of translations using 3DTouch.

Moving the finger right or left in Figure 4.2a causes a translation along the x-axis. Analogously, moving the finger up and down in Figure 4.2b, respective 4.2c, results in translations along the y- and z-axis.

4.1.3.2 Rotation

Similar to translations, 3D rotations are performed using single touch and the device pose.

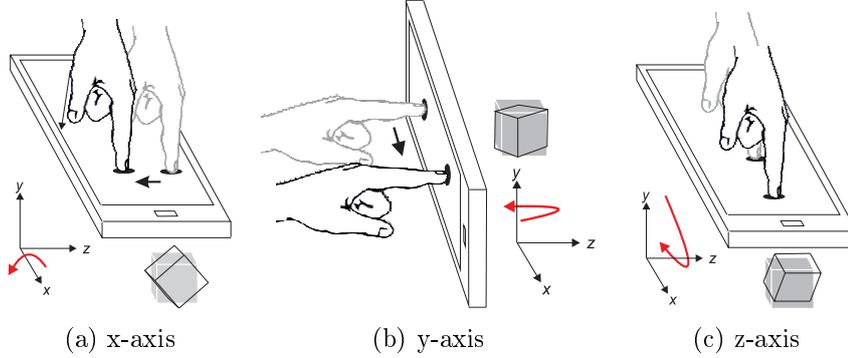


Figure 4.3: Examples of rotations using 3DTouch.

The algorithm is based on the proposed translation algorithm and extended by the following steps. Instead back-projecting the two touch point, a line perpendicular to $line(p(t_1), p(t_2)) \in \mathbb{R}^2$ is calculated and the two new points $line(p_{\perp}(t_1), p_{\perp}(t_2))$ are back-projected, resulting in $P(t_1), P(t_2) \in \mathbb{R}^3$. To calculate the angle of rotation, the factor f is determined, as described in Equation 4.2.

$$f_r = \frac{(360 \cdot s \cdot \|\vec{v}\| \cdot \hat{e}_{max})}{U} \quad (4.2)$$

s is the scalar taken from the translation algorithm, indicating a positive or negative rotation. $\|\vec{v}\|$ regulates the angle as a fraction of the circumference U of the bounding sphere of the manipulated object. The factor f_r is then applied to the current rotation in the object’s local coordinate system.

In Figure 4.3, examples of the resulting 3D rotations using the proposed 3DTouch algorithm are illustrated. Moving the finger up and down in Figure 4.3a causes a rotation around the x-axis. Analogously, moving the finger right or left in Figure 4.3b, respective 4.3c, results in a rotation around the y- or z-axis.

4.1.3.3 Scaling

The proposed algorithm supports non-uniform scaling. Therefore, a two-finger pinch-like gesture is used and applied with an adapted version of the proposed algorithm from Section 4.1.3.1. The touch points of both fingers $p_i \in \mathbb{R}^2$ at two moments in time $t(i)$ are back-projected into 3D, resulting in a set of points $P_i(t_i) \in \mathbb{R}^3 \mid i = 1, 2$. The sign of scaling and its amount depend on the direction and magnitude of the pinch gesture. Moving both fingers together results in negative scaling, moving apart determines a positive scaling. The scaling factor $f_s \in \mathbb{R}^3$ is then calculated as denoted in Equation 4.3.

$$f_s = (\|\vec{v}(t_2)\| - \|\vec{v}(t_1)\|) \cdot \hat{e}_{max}(t_1) \quad (4.3)$$

Finally, the sign for scaling is then determined by Equation 4.4 and f_s is added to the current scale in the object’s local coordinate system.

$$f_s = \begin{cases} f_s & \text{if } f_s > 0 \\ f_s \cdot (-1) & \text{else} \end{cases} \quad (4.4)$$

4.1.4 The HOMER-S Technique

The mapping between touch input and object manipulation of 3DTouch is straightforward and simple. However, the touch abstraction layer still exists and manipulation is limited to the screen size of the handheld’s device. Therefore, the novel HOMER-S technique is introduced, which integrates all 6DOF of a translation and rotation task by directly mapping the handheld’s pose onto the selected object. Scaling as a spatial non-rigid transformation is designed as a separate 3DOF task and re-uses the device’s position information for non-uniform object manipulation. Thereby, real-world metaphors for translation, rotation and scaling are imitated, touch input during manipulation is eliminated and the interaction space is extended to the user’s physical space.

4.1.4.1 6DOF Manipulations

Inspired by [66] and using the immersive 3D method HOMER [24] as foundation, the proposed technique HOMER-S was designed. The original HOMER algorithm uses the 6DOF pose of the user’s torso and that of the interaction device to manipulate an object.

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

Since a handheld setup features different characteristics, we adapted HOMER to be applicable in handheld mixed reality environments using a tablet or smartphone (therefore HOMER-S). The full 6DOF manipulation of HOMER-S is depicted in Figure 4.4.

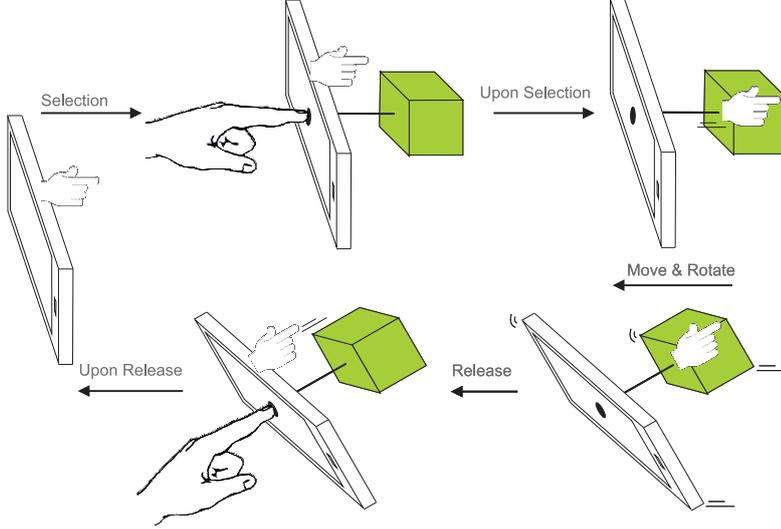


Figure 4.4: 6DOF translation and rotation using HOMER-S.

Rotations of the selected object around arbitrary axes are controlled independently. An isomorphic mapping between the handheld’s orientation and the virtual hand is applied to rotate an object around the hit point that describes the pivot point. Thereby, the physical movement and rotation of the mobile device directly influences the transformation of the selected object. By performing Mobile Raycasting, the object is released and the virtual hand moves back to the handheld’s position. The proposed HOMER-S algorithm is summarized in Algorithm 4.1.

4.1.4.2 Scaling

To scale an object, the virtual hand’s position $\vec{p}(vh) \in \mathbb{R}^3$ is used. At each frame at time t , $\Delta_p \in \mathbb{R}^3$ is obtained as described by:

$$\Delta_p = \vec{p}_t(vh) - \vec{p}_{t-1}(vh). \quad (4.5)$$

Δ_p is subsequently mapped onto the select object O to update its scale $\vec{s}(O) \in \mathbb{R}^3$ in a frame-wise manner, as described by:

$$s(O) = s \cdot (\Delta_p + \vec{s}(O)), \quad (4.6)$$

where the scalar s denotes a scaling factor that controls the amount of the frame-wise scaling and that can be adjusted to the specific application requirements. Thus, moving the virtual hand in positive direction of each axis will scale up; moving in negative will scale down the object. Thereby, a straightforward non-uniform scaling along all axes is achieved.

Algorithm 4.1: Algorithm of 6DOF manipulation with HOMER-S in pseudo code.

Data: Handheld's 6DOF pose $(h_p, h_o | \in \mathbb{R}^3)$
 Init *VirtualHand* $(vh_p, vh_o \leftarrow 0 | \in \mathbb{R}^3)$;
 Init *Object* $(O_p, O_o \leftarrow 0 | \in \mathbb{R}^3)$;
 Set *uponSelection* $\leftarrow false$;

while $(h_p \& h_o) = true$ **do**
 | $vh_p \leftarrow h_p$;
 | $vh_o \leftarrow h_o$;
 | **if** *O* is selected **then**
 | | **if** *uponSelection* = *false* **then**
 | | | $vh_{sel} \leftarrow h_p$;
 | | | $uponSelection \leftarrow true$;
 | | **end**
 | | *A rotation has performed*;
 | | $O_o \leftarrow h_o$;
 | | *A translation has performed*;
 | | $vh_{curr} \leftarrow h_p$;
 | | Calculate distance: $d(vh_{curr}, vh_{sel})$;
 | | Normalize vector: $\vec{v}_{norm} \leftarrow \vec{v}(vh_{sel}, vh_{cur})$;
 | | Set: $vh_p \leftarrow vh_{sel} + d \cdot \vec{v}_{norm}$;
 | **else**
 | | $uponSelection \leftarrow false$;
 | **end**
end

4.1.5 Assistance Design

To allow changing the manipulation tasks during run-time as well as to support the user with visual feedback about accessible axis for interaction, the following design modalities for assistance have been incorporated into both manipulation techniques.

4.1.5.1 Mode Switches

Since 3DTouch offers RST by decomposing each transformation into a separate 3DOF task, mode switches between the manipulation entities are required. This is realized through a simple button interface, as illustrated in Figure 4.5a. This mode switch introduces an additional extra input modality compared to previous work [103, 143]. However, as reported in literature [126, 104], DOF-separation of the manipulation task leads to better results than trying to use the separated DOF of a multi-touch display in an integral way, as demonstrated in [103]. Thereby, the additional input modality can be compensated by enhanced performance and ease of use.

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

When using HOMER-S, no DOF-separation is required for the integral task of translating and rotating an object. However, to provide the same structure for later evaluation, translation and rotation can also be performed as separated manipulation entities (see Figure 4.5b). HOMER-S takes advantage of exploiting real-world metaphors for translation, rotation and scaling an object in space. However, the metaphors for translating and scaling are akin in movement and hence are hard to distinguish if only the device pose is examined. Instead of introducing another, more complex metaphor for scaling, a mode switch between the 6DOF manipulation task and non-uniform scaling is proposed. Therefore, the simple button interface is applied, as described for 3D Touch.

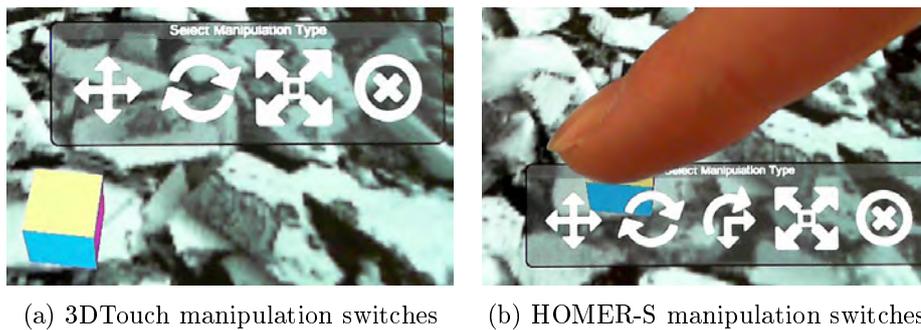


Figure 4.5: Floating GUIs of both techniques upon selection.

To summarize, HOMER-S provides the following manipulation entities: (1) translation, (2) rotation, (3) translation & rotation (6DOF), and (4) scaling.

4.1.5.2 Supporting Visualization

To increase the ease of use during object manipulation, supportive information is provided to users.

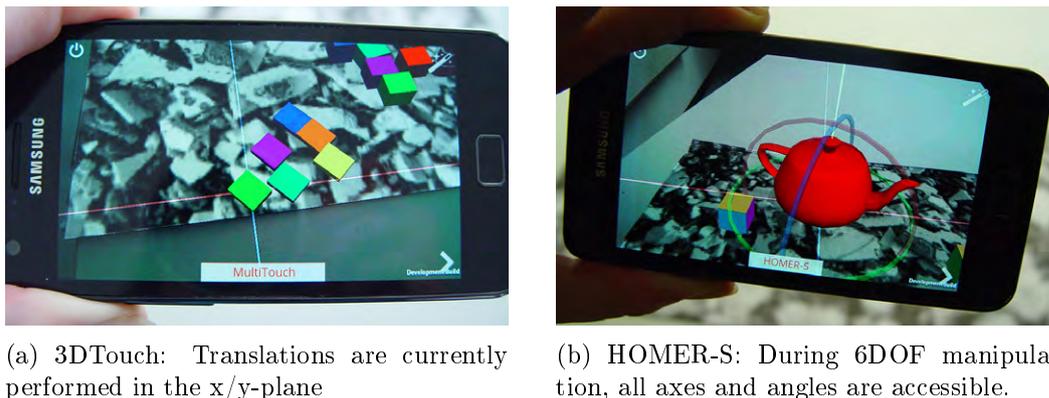


Figure 4.6: Supporting visualization depending on manipulation task and current accessible interaction axes.

Therefore, 3DTouch and HOMER-S draw axes during translation and scaling as well as gimbals during rotation to visualize accessible interaction axes and angles according to the current device pose, as illustrated in Figure 4.6.

4.1.6 Crucial Aspects

The nature of the proposed techniques offers intuitive handling of 3D manipulation tasks but introduces some crucial aspects as well.

Loss of Tracking Since both methods are designed for handheld mixed reality setups, a valid device pose is required. Loss of tracking thus results in malfunction of object manipulation. Currently optical tracking is proposed to estimate the device pose due to its accuracy, low latency and non-drift characteristics (see Section 2.1.1.1 in Part II). To increase the tracking robustness, fusing of optical tracking data with the measurements of the handheld device’s built-in inertial measurement unit can be applied. Thereby, a complete loss of tracking can be omitted in case of (temporary) occlusions or inconsistent light situations.

Rotation When performing rotational tasks with HOMER-S, a drawback is caused by the direct mapping of the device orientation onto the selected object. Given the implicit binding of input- and output device in handheld mixed reality, rotations around the pitch-axis are limited. This is especially true as soon as only one physical feature is used for optical tracking. 360° rotations around the yaw-axis can be applied by real world movements of the user, while rotations around the roll-axis are straightforward and employ the steering wheel metaphor.

4.2 Performance Studies

For a comprehensive evaluation of the two proposed manipulation techniques, a summative evaluation was conducted across four different manipulation scenarios based on variations of the employed interaction tasks.

4.2.1 Prerequisites

Object Selection The interaction task *Object Selection* was not reviewed within the following study. However, selection is required for subsequent manipulation. Therefore, Mobile Raycasting from Section 3.3.2 is employed across the four test scenarios.

Baseline Technique The immersive 3D manipulation techniques presented in Section 2.3.1 are not originally designed for handheld mixed reality setups and require separate tracking of the user’s head and the input device. This directly conflicts with the *Single I/O Device* requirement and can therefore not be applied without further adaption. However, any adaption needs to be carefully reviewed to ensure that the original

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

characteristics of the technique remain. As described in Section 3.4.1, the adaption to use Go-Go with a 3D multi-touch device clearly alters the original non-linear mapping approach and thus is not applicable as baseline technique. Multi-touch techniques such as [108, 143] use DOF-separation to obtain manipulation tasks with reduced degree-of-freedom. These subtasks are then performed with specific 2D multi-touch gestures. However, [108] only enables 3D translations. Although [143] provides gestures for RST manipulation, the necessary multi-touch gestures are inconsistent. For instance, a vertical movement of two fingers causes a translation along the y-axis but for scaling a change along the z-axis. These inconsistencies in combination with the required prior knowledge of the underlying multi-touch gestures do not allow for a valid comparison. In [103], RST operations are provided by employing two-handed three-finger gestures. However, using two handed multi-touch input violates the requirement of *Limited Gesture Complexity*, resulting in a difficult applicability for the one-handed handheld setup.

As the existing techniques do not apply for a clean and fair performance evaluation of manipulation techniques in one-handed handheld mixed reality, 3D Touch and HOMER-S are compared within the following study. Thereby, the characteristics of DOF-separation in contrast to DOF-integration according to the interaction task can be robustly evaluated.

4.2.2 Objectives

The main goal of the experiment was to evaluate the performance and usability of 3D Touch and HOMER-S. Since 3D Touch matches the separated structure of the multi-touch input device and HOMER-S adapts real-world metaphors by applying the integral structure of the given device pose, both techniques apply for straightforward manipulation. Thus, a second objective was to compare both techniques and to examine intuitive handling. In designing the experiment, the following hypotheses were formulated:

- H1** 3D Touch and HOMER-S are both designed to provide intuitive manipulation. Thus, both techniques will perform similar in terms of speed and ease-of-use for 3DOF tasks.
- H2** Since HOMER-S offers full 6DOF manipulation, it will perform considerably faster than 3D Touch for compound translation and rotation tasks.
- H3** Touch gestures enable a higher precision than free movements in 3D. Thus, 3D Touch performs better for fine manipulation tasks that require precise input.
- H4** Regarding prior knowledge, users with experience using multi-touch devices will perform equally or better with 3D Touch than with HOMER-S. Likewise, the design of HOMER-S enables better performance for users with no prior multi-touch knowledge.

4.2.3 Experimental Design and Procedure

We conducted the study using a 2x4 within-subjects factorial design where the independent variables are manipulation technique and task scenario. In a second order evaluation, user experience was the third independent variable. The manipulation techniques were 3DTouch and HOMER-S, while the scenarios included four different experimental tasks with varying types of canonical manipulation tasks and combinations. The dependent variables were *Task Completion Time* and *Number of Interaction Steps*. Task completion time represents the time it takes to successfully finish a specific scenario while number of interaction steps comprises the amount of necessary mode switches to successfully finish an (compound) manipulation task. Furthermore, we measured user preferences for both techniques in terms of speed, accuracy, and ease of use.

The user study was analogously designed to the *Selection Study* from Section 3.4. Thus, the same procedure as in Figure 3.6 was applied. The material of the study is presented in Appendix VI.A. At the beginning of the study, each participant was asked to read and sign a standard consent form as well as to complete the pre-questionnaire from Table 3.1.

No.	Question
Q1	How adequate do you feel the time allotted for practice was?
Q2	How comfortable were you with using a smartphone for task completion?
Q3	How would you rate the 3DTouch manipulation technique in usability? Speed? Accuracy?
Q4	How would you rate the HOMER-S manipulation technique in usability? Speed? Accuracy?
Q5	How would you rate intuitiveness of 3DTouch for 2D-translate, 3D-translate, rotate, move & rotate, scale an object?
Q6	How would you rate intuitiveness of HOMER-S for 2D-translate, 3D-translate, rotate, move & rotate, scale an object?
Q7	Which manipulation technique do you prefer to 2D-translate, 3D-translate, rotate, move & rotate, scale an object.
Q8	Rank the two manipulation techniques in order of desired use (with 1 being the most desired).
Q9	When determining how much you like using a manipulation technique, how important in influence on your decision was ease-of-use? Speed? Accuracy?

Table 4.1: Post-Questionnaire

Upon completion, the participant was given a detailed description of the practical part about "Manipulation in handheld Mixed Reality". A tutor coached them on how to use the handheld device and how to perform 3D manipulation in a test environment. Afterwards, each participant had five minutes time to practice both techniques. Once they started the study, they were not interrupted or given any help. Upon completion

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

of the practical part, they were asked to fill out a post-questionnaire (see Table 4.1). It took approximately 25 minutes for each participant to finish the user study. All 28 participants yielded successful simulation trials from which all data was used for analysis.

4.2.4 Subjects & Apparatus

Of the 28 participants ranging from 23 to 38 years, 12 were female and 16 male. 12 participants stated not to have any mobile 3D gaming experience at all, while 7 reported no experience with multi-touch smartphones. Table 4.2 gives an overview of users based on their prior experience.

Group	Inexperienced	Experienced
a) Mobile 3D Gaming	12	16
b) Smartphone	7	21

Table 4.2: Users grouped by prior experience

All computations – tracking, rendering, selection and manipulation of virtual objects – were performed on a smartphone using Android OS; more details are given in Section 3.4.5.

4.2.5 Test Scenarios

We built four different scenarios to simulate typical 3D manipulation situations. According to [62], the basic canonical tasks *position*, *rotation* and *scaling* were used to design the four test tasks of varying complexity. To manually identify the desired object for subsequent manipulation, another canonical task *selection* is required. Since the selection task is performed by all users in the same way and is equally designed over all four tasks, the necessary time does not influence the performance metrics.

All scenarios are based on the same virtual working ground (black & white textured plane) that was printed to paper at $56 \times 40 \text{cm}$ and acted as a visual planar marker for the natural feature tracking toolkit [163]. The 28 participants completed the four scenarios in a random order. Each scenario featured a simple description of the upcoming task. Before starting the actual tests, users could inspect the scenario without being able to interact with in order to understand the task according to its description. The four scenarios are depicted in Figure 4.7 and are defined in the following.

4.2.5.1 Positioning on a Plane

The first task comprises the canonical task *positioning*. The user was challenged to translate a pink cube in the lower left corner to the center of a green area in the upper right corner, as depicted in Figure 4.7a. The distance between the targeted object and its destination was 35cm on the horizontal plane. It was sufficient to complete the task with the cube partly overlapping the designated target.

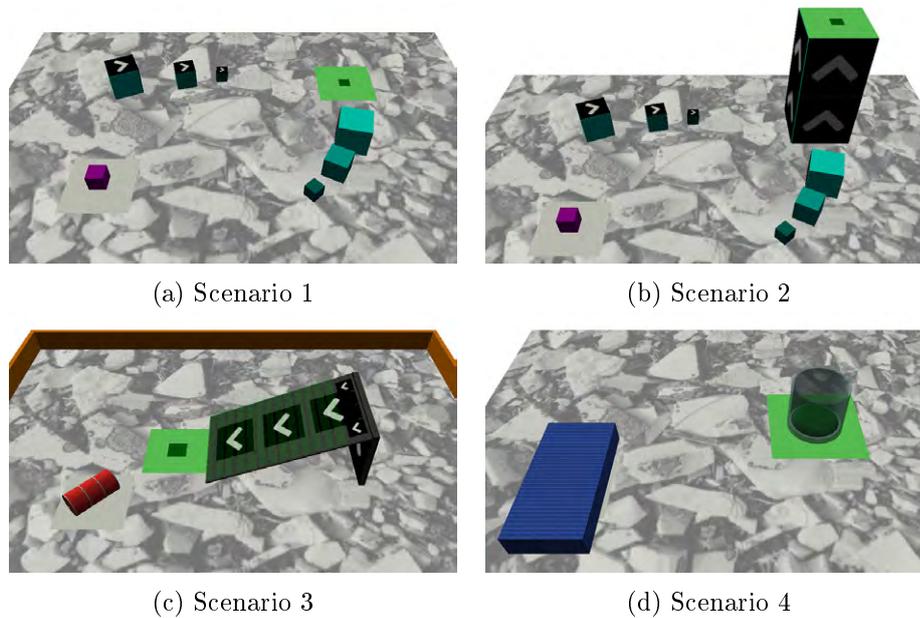


Figure 4.7: The three test scenarios of the performance user study.

4.2.5.2 Positioning in 3D Space

The second task extends the first scenario by requiring positioning in all three dimensions. The user was challenged to translate a pink cube in the lower left corner on top of a small tower in the upper right corner (see Figure 4.7b). The distance between the targeted object and its destination was 35cm on the horizontal plane and 20cm vertically. The destination area was again a square. If it was partly overlapped by the target object, the task was completed.

4.2.5.3 Positioning & Rotation in 3D Space

For better simulation of manipulation requirements in mixed reality applications, we applied an integral task design for the third scenario comprising a combination of *positioning* and *rotation*. The user was challenged to rotate a red barrel in the lower left corner by 45° around its vertical axis and translate it on top of an inclined plane (see Figure 4.7c). From there the barrel was supposed to roll down the plane and over a square at its bottom. The test was successfully completed if the barrel was let loose on the top of the inclined plane rolling down its full length and at least partly hitting the center of the destination area.

4.2.5.4 Non-Uniform Scaling & Positioning in 3D Space

A second integral task was designed for the fourth scenario. Here, the user was first challenged to scale a blue cube by a fifth in length and a third in width of its original

size and then move the cube into a glass positioned at the center of the scene (see Figure 4.7d). The distance between the targeted object and its destination was 38cm horizontally and 10cm vertically. The destination was the circular shaped bottom of the glass. Users needed to let the cube fall into the glass from above and as soon as it hit the bottom, the task was completed.

4.3 Experimental Results

Based on the performance study, we conducted an evaluation on the quantitative data to examine performance of the two techniques and a subjective evaluation regarding user's preferences and feedback.

4.3.1 Quantitative Evaluation

The quantitative data gathered from the questionnaires and automatically collected by the test application was analyzed with Friedman's χ^2 test¹ and repeated measures single factor ANOVA accordingly on both *Task Completion Time* and *Number of Interaction Steps* (see Section 4.2.3) as well as for each scenario (see Section 4.2.5). We focused on three different aspects during data analysis:

1. Data of all participants regarding the manipulation techniques is evaluated.
2. The techniques' performance was analyzed depending on tasks.
3. Data of selected participants - according to the user experience listed in Table 4.2 - was analyzed for each manipulation technique and task separately.

4.3.1.1 Performance Evaluation

Analyzing the overall mean completion time, no significant difference was found between HOMER-S and 3DTouch ($F_{1,27} = 0.00299$, $p = 0.957$), as illustrated in Figure 4.8. When inspecting the mean completion time for each task separately, again no significant differences could be found for both positioning tasks 1 (*Positioning on a Plane*) and 2 (*Positioning in 3D Space*) at ($F_{1,27} = 1.4$, $p = 0.2468$) and ($F_{1,27} = 0.814$, $p = 0.375$), respectively. However, task 3 (*Positioning & Rotation*) was performed significantly faster with HOMER-S ($F_{1,27} = 7.379$, $p < 0.0114$). In contrast to that, HOMER-S took significantly more time to complete task 4 (*Scaling & Positioning*) at ($F_{1,27} = 7.379$, $p < 0.0114$), as illustrated in Figure 4.9.

Analyzing the task completion time, grouped by users' knowledge according to Table 4.2 revealed no further significant differences other than the overall ones illustrated in Figure 4.9. No significant differences could be found for both positioning tasks when analyzing the users' experience. For task 3 (*Positioning & Rotation*), the significantly better performance of HOMER-S was never independent of the users' experience. The

¹Since the degree-of-freedom is $k = 2$ for this analysis, we denote $\chi_{k-1}^2 = \chi_1^2$ as χ^2 in the following.

inexperienced users of the mobile gamer group (a) as well as of the smartphone group (b) performed significantly faster with HOMER-S than with 3DTouch. The experienced users of both groups performed faster with HOMER-S as well, but not significantly. Furthermore, only the experienced groups of a) and b) had significant results for task 4 (*Scaling & Positioning*), since they were significantly faster using 3DTouch. No significant difference in performance between 3DTouch and HOMER-S could be found for the inexperienced users of both groups in task 4 (*Scaling & Positioning*).

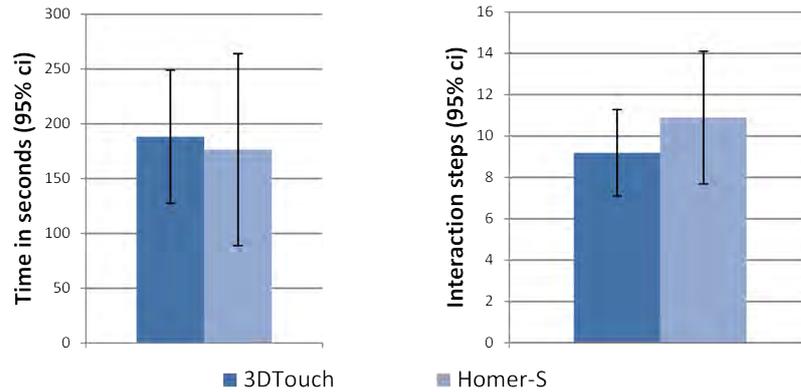


Figure 4.8: Mean completion time and mean number of interaction steps.

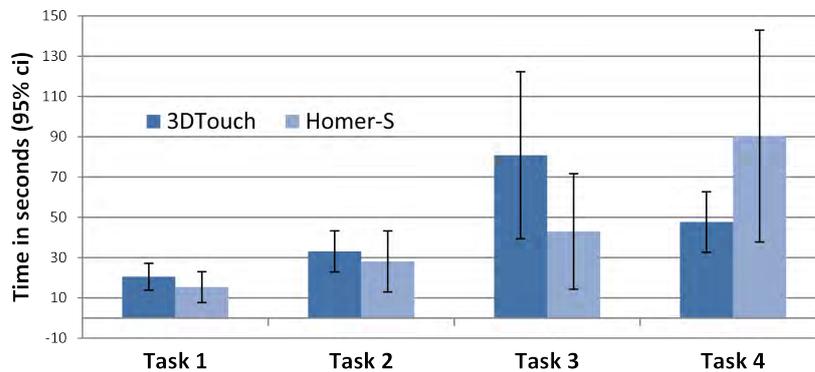


Figure 4.9: Mean completion time per task.

The results of the evaluation for the overall mean number of interaction steps exposed that 3DTouch enabled users to perform manipulations in significantly less steps than HOMER-S ($F_{1,27} = 4.552, p < 0.0421$), as illustrated in Figure 4.8. However, the evaluation of the number of interaction steps per tasks found only a significant difference in task 2 (*Positioning in 3D Space*) at ($F_{1,27} = 4.374, p < 0.046$) and in task 4 (*Scaling & Positioning*) at ($F_{1,27} = 12.81, p < 0.0013$), both in favor of 3DTouch. Figure 4.10 indicates no significant difference for both task 1 (*Positioning on a Plane*) or task 3 (*Positioning & Rotation*), both with ($F_{1,27} = 0.685, p < 0.415$).

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

The evaluation of the mean number of interaction steps, grouped by users' experience, revealed with one exception for task 3, no deviant results than those illustrated in Figure 4.10.

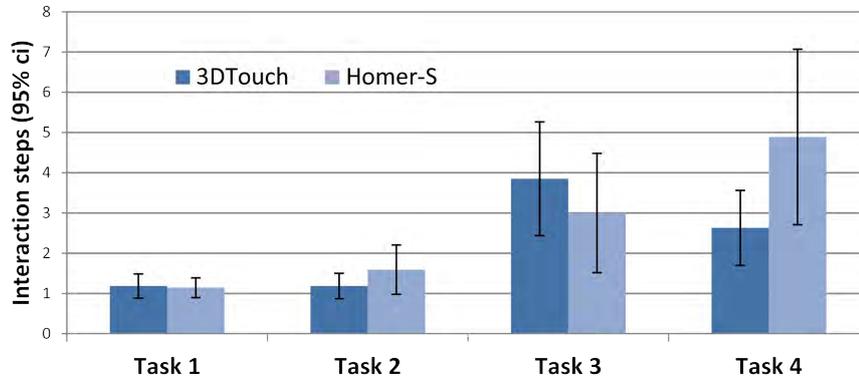


Figure 4.10: Mean number of interaction steps per task.

The significantly better performance of 3DTouch in task 2 (*Positioning in 3D Space*) could only be confirmed for the experienced users in a) and b). For task 3 (*Positioning & Rotation*), the inexperienced group of a) achieved significantly better results with HOMER-S than with 3DTouch. For all other groups no significance could be found for that task. For task 4 (*Scaling & Positioning*), only the experienced users of both groups had significantly better results with 3DTouch than with HOMER-S. No significant difference could be found for the inexperienced users of both groups.

4.3.2 Subjective Evaluation

When answering the questions Q1-Q6 and Q9, users were able to choose from a 7-point Likert scale [2].

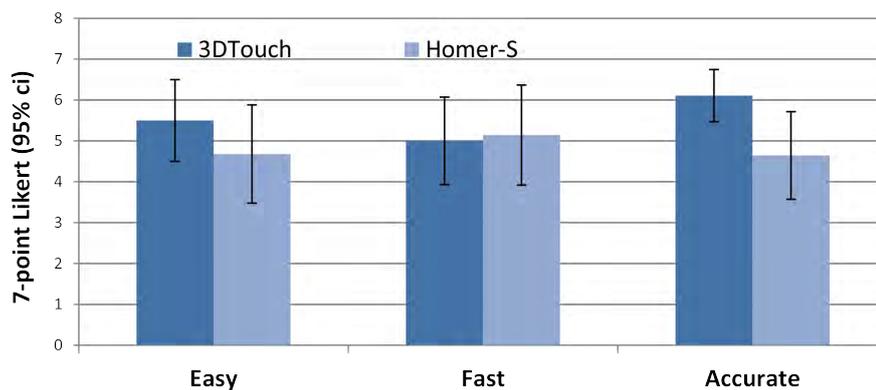


Figure 4.11: Users' average rating of Q3 & Q4.

While all questions feature the highest rating at seven, and the lowest at one, Q1 states the best rating with four (appropriate). Our participants found the time allotted for practice appropriate ($\mu = 4$ and $\sigma = 0.46$ at $\alpha = 0.05$). Using a smartphone to complete the different tasks was rated to be moderately comfortable ($\mu = 5.9$ and $\sigma = 1.14$ at $\alpha = 0.05$). As illustrated in Figure 4.11, the questions Q3 and Q4 revealed both to be average or good, but 3DTouch was rated significantly better for ease-of-use and accuracy with ($\chi^2 = 6.55$, $p < 0.0105$) and ($\chi^2 = 15.696$, $p < 0.0000744$) respectively. In terms of speed, no difference was confirmable.

Analyzing the subjective evaluation of ease-of-use, speed and accuracy, grouped by the user’s experience, revealed significantly better ratings of 3DTouch in ease-of-use only for experienced users of a) and b). 3DTouch’s better rating for accuracy was independent of the users experience in a) und b) except for inexperienced users in b) where no significant difference occurred. Users’ ranking of the two interaction techniques indicated no significant preference (Q8) ($\chi^2 = 0.57$, $p = 0.45$).

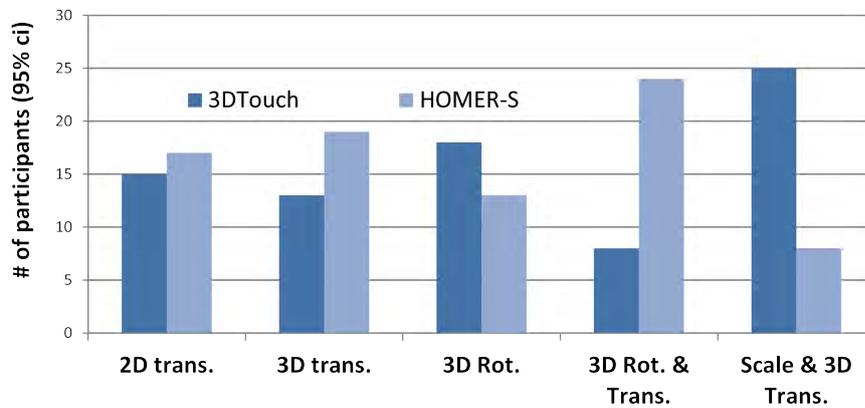


Figure 4.12: Users’ preferences given Q7.

A closer inspection of the users’ preferences grouped by individual manipulations revealed that for 2D- and 3D translation as well as rotation alone no significant difference in preferences could be found, as shown in Figure 4.12. For the integral 6DOF manipulation of task 3 (*Positioning & Rotation*), HOMER-S is significantly preferred with ($\chi^2 = 10.67$, $p < 0.0011$). For scaling, 3DTouch is significantly preferred with ($\chi^2 = 12.57$, $p < 0.00039$). This subjective evaluation reflects the results of the quantitative evaluation in terms on completion time.

No deviant results for 2D- and 3D-translation as well as rotation alone were revealed, when analyzing the ranking of each manipulation, grouped by the users’ experience. The users’ preference of both groups for the integral rotation and translation task 3 (*Positioning & Rotation*) revealed that HOMER-S was significantly preferred by the experienced users. Also the inexperienced users preferred HOMER-S, but not significantly. 3DTouch’s preference for scaling remains independent of user’s experience in both groups.

Question Q9 inquiring the users’ influence on their decision for questions Q3 and Q4

4. 3D MANIPULATION IN HANDHELD MIXED REALITY

yields with ($\chi^2 = 3.89, p < 0.143$) no significant difference for the three options ease-of-use, speed and accuracy. Users stated all aspects of Q9 similarly important, ranging from $\mu = 5.5$ (slightly important) to $\mu = 6.18$ (important).

4.4 Discussion

We designed the experiment to compare two different techniques for performing 3D manipulation tasks with a multi-touch handheld device. While 3DTouch separates the DOFs of the task to improve performance as shown in previous work [126], HOMER-S controls 6DOF in an integral way and takes advantage of simulating real-world metaphors.

Results show that for both techniques, no significant difference was found for overall mean task completion time, completion time for the positioning tasks, overall user preference or user preferences regarding the positioning tasks that support hypothesis H1. Inspecting performance and user's preference for compound canonical tasks, two findings can be stated. First, for 6DOF manipulation tasks, as simulated by task 3 (*Positioning & Rotation*), HOMER-S performed significantly faster than 3DTouch. This quantitative evaluation is supported by the user's subjective feedback. HOMER-S is significantly preferred for translation and rotation tasks by users as expressed in Q7. These findings support H2 and indicate the strength of the integral design of HOMER-S for compound canonical 6DOF tasks. This is also reflected by users' comments who described HOMER-S to be natural, of "more direct contact" and fun. Thus, these real world metaphors tend to be very intuitive and straightforward. The second finding when inspecting performance and user's preference for composite manipulation tasks reveals the strength of 3DTouch for scaling tasks. It took considerably less time to complete task 4 (*Scaling & Positioning*) using 3DTouch than with HOMER-S. Furthermore, users significantly preferred 3DTouch for scaling. Since no significant difference was found regarding the positioning tasks in completion time or user preferences, positioning can be neglected when evaluating task 4. This finding supports H3, since the scaling tasks required very fine manipulation in all three dimensions. H3 can further be backed up by the significant fewer number of interaction steps 3DTouch needed in task 2 (*Positioning in 3D Space*) and task 4 (*Scaling & Positioning*). Furthermore, the users' rating in Q3 & Q4 attested it a better accuracy.

Besides the assumption, that humans are able to control their fingers more precisely, the underlying metaphor can be another conceivable reason to further explain the underperformance of HOMER-S in scaling tasks. In the real world, usually two hands are involved to expand or shrink an object. Since HOMER-S only provides one virtual hand to simulate one real hand, this metaphor could not be adapted in a direct way. Thereby, a direct mapping could not be provided that limits HOMER-S straightforward usage for scaling. However, the pinch-like gesture to scale an object using 3DTouch is also not completely intuitive and straightforward. Since, more than half of our test group classified themselves as experienced mobile 3D gamers, they are familiar with using multi-touch for interaction; standard touch gestures such as the pinch-out and -in are known and well trained. This is also backed up by the results including user experience. There, the

results of 3DTouch for scaling are only significantly better for users who are experienced with smartphones or mobile 3D gaming.

Studying further details regarding user experience leads to H4. We proposed that prior touch knowledge would result in equal or better performance of 3DTouch compared to HOMER-S, while inexperienced users would perform better with HOMER-S due to its integral 6DOF design and adaption of real-world metaphors. For many results of the study, this is true. Regarding completion time, no significant differences between 3DTouch and HOMER-S could be found for positioning when analyzing experienced users. For 3D positioning, experienced users needed significantly less interaction steps when using 3DTouch. For integral positioning and rotation, experienced users of both groups performed faster with HOMER-S, but not significantly. Experienced users performed significantly faster for scaling in terms of completion time and number of interaction steps when using 3DTouch. They rated 3DTouch significantly better in terms of ease-of-use, but significantly preferred HOMER-S for 6DOF manipulation.

Regarding inexperienced users, H4 can be further backed up by the significant better performance in terms of completion time and number of interaction steps for task 3 (*Positioning & Rotation*) using HOMER-S. Users' comments reflect the quantitative results. Most users, especially the inexperienced, reported to have quickly familiarized with HOMER-S for any translations and rotations. However, exceptions when evaluating H4 could be found, too. The quantitative results do not indicate a better performance of inexperienced users using HOMER-S for positioning tasks. For scaling, HOMER-S did not result in better performance of the inexperienced users. However, despite of the good results of 3DTouch for scaling, inexperienced users did not significantly perform better using 3DTouch for scaling. The underlying two-fingers pinch gesture requires prior knowledge and thus, is not as straightforward and direct than the one-finger inputs for translate and rotate. But users' preference of 3DTouch's for scaling is independent of the users' experience. This is also reflected by users' comments. Some users experienced HOMER-S as being "too direct", since even small movements of the mobile device result in a transformation. Most users complained about HOMER-S being unintuitive to use for scaling. Based on these observations, we cannot draw a clear conclusion to support H4. Further research needs to be performed for a detailed evaluation of this hypothesis.

Based on these results and findings, we come to the following ultimate conclusions that can further act as basic design guidelines:

- Both methods provide intuitive manipulation with similar performance when the canonical tasks *Positioning* and *Rotation* are performed.
- HOMER-S outpaces 3DTouch in performance and ease-of-use when performing a compound, full 6DOF positioning and rotation tasks.
- 3DTouch is the better choice, if scaling is involved in the manipulation task.

Chapter 5

Summary

In this part, three novel 3D interaction techniques were introduced for selection and manipulation of 3D objects, all aiming on intuitive and straightforward 3D interaction in one-handed handheld mixed reality environments. With these results for object selection and manipulation, our research objectives from Section 1.2 are achieved.

Using the imprecise finger touch input for object selection yields the inaccurate extraction of small objects, especially when they are partly or fully occluded or surrounded by highly similar virtual scene objects. State-of-the-art approaches mostly propose two-handed techniques to increase selection accuracy, which is not applicable in the given interaction scenario. Furthermore, existing approaches do not provide sufficient contextual information upon object indication to precisely select a desired object amongst visually similar ones. To overcome the limitations, the novel technique *DrillSample* was developed with a major design focus on precise selection of objects in dense virtual scenes while reducing necessary 2D multi-touch input. DrillSample only requires one-finger tap gestures as input and splits the selection procedure into two steps. For object indication, Raycasting is employed that indicates all casted scene objects for later selection. In case of multi-object indication, their full 3D spatial context is preserved upon object indication allowing for disambiguation and precise selection of occluded objects or objects with high similarity in visual appearance. By employing a one-finger tap gesture, the desired object is selected within this refinement step. The possibly imprecise object indication is thereby compensated by the optional second refinement step. For a comprehensive evaluation of the DrillSample selection technique, a quantitative and qualitative evaluation was conducted by comparing *DrillSample* with the two baseline techniques *Mobile Raycasting* and *Expand* across three different selection scenarios based on variations of object density and visibility. The study clearly revealed the strengths of DrillSample in precise selection of objects within close range in dense virtual scenes. To select small and distant objects, Expand was found more sufficient as it applies a volumetric object casting. While Raycasting remains a good alternative for selecting visible objects in a sparse scene, DrillSample was found the best general purpose method for visible as well as partly and fully occluded objects, independent of their visual appearance.

5. SUMMARY

To provide 3D manipulations using 2D multi-touch, existing approaches usually use complex finger and hand gestures that are difficult or impossible to apply in a one-handed handheld interaction scenario. Furthermore, their application lowers intuitive handling since the complex gestures require prior knowledge. To address the limitations of existing 3D manipulation techniques for handheld mixed reality environments, the two novel methods *3DTouch* and *HOMER-S* are presented which both support translation, rotation and scaling as 3DOFs manipulation tasks. *3DTouch* provides 3D translation and rotation as well as non-uniform scaling by fusing simple one- or two-finger touch input with the handheld’s current 6DOF pose. The integral 6DOF manipulation is decomposed into two separate tasks, enabling one finger to be sufficient to access all three DOFs during translation and rotation. Scaling requires a two-finger pinch gesture while providing non-uniform transformation in all three dimensions. *HOMER-S* provides interaction beyond the (limited) screen dimensions by decoupling the manipulation process from any touch input. It aims at DOF-integration and maps the 6DOF device pose onto the object upon selection. Thereby, full 6DOF manipulation as well as non-uniform scaling is performed by employing real-world metaphors that are intuitive to use. In a comprehensive user study, performance, accuracy and ease of use for both techniques were assessed across four different test scenarios with varying manipulation tasks. The results reveal both techniques to be intuitive to translate and rotate objects. *HOMER-S* lacks accuracy compared to *3DTouch* but achieves a significant performance increase in terms of speed for full 6DOF manipulation.

Creating Mixed Reality Environments

1	Introduction	171
1.1	Motivation	172
1.2	Organization	172
2	Background & Related Work	173
2.1	Key Elements of a Mixed Reality Framework	173
2.2	Application Development & Scene Management	174
3	Framework Architecture	177
3.1	Base Infrastructure	178
3.2	Middleware	179
3.3	Application Layer	183
3.4	Workflow for Application Development	187
4	Developed Mixed Reality Environments	189
4.1	Test Setups & Environment	189
4.2	Non-Immersive Mixed Reality	190
4.3	Combined Non- & Semi-Immersive Mixed Reality	192
4.4	Combined Semi- & Full Immersive Mixed Reality	192
5	Summary	195

Chapter 1

Introduction

To create a compelling mixed reality environment, tracking and interaction are two key components, as it was extensively described and studied within the previous chapters of this thesis. A crucial factor to enable mixed reality for broad (everyday) usage is quick application prototyping and development.

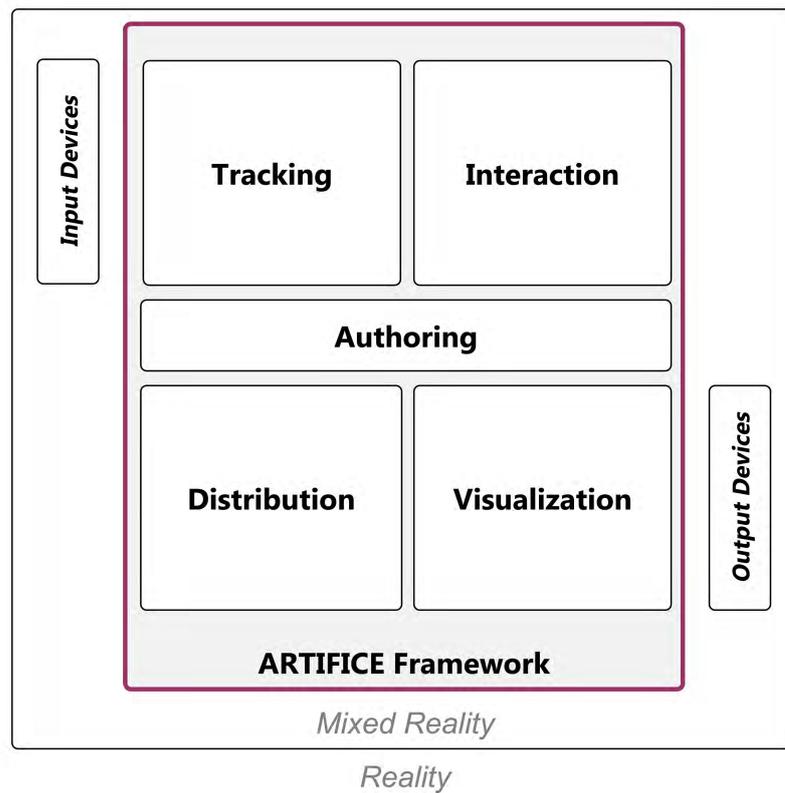


Figure 1.1: Key components of mixed reality, with the contributions marked bold.

1. INTRODUCTION

Application development, however, requires knowledge in all involved sub-domains, as depicted in Figure 1.1. This comprises tracking, interaction, scene authoring, 3D visualization and, optionally, network handling for distribution. For each component, a large variety of technologies, methods and algorithms exists, as for tracking and interaction described in the Chapters II.1 and III.1. This necessary knowledge results in a high entry threshold to create mixed reality applications, even for quick prototyping.

1.1 Motivation

To lower the entry threshold of application development and thereby, to leverage mixed reality technology for a broader everyday usage, an inexpensive toolkit is required that serves a powerful graphical interface to easy access and to author the modules visualization, tracking, interaction and distribution. Furthermore, to be able to employ such a framework for our performed research to develop test applications, it must provide interfaces to extend the framework with novel software techniques and it has to support state-of-the-art mobile devices running Android for handheld mixed reality application development. However, at the moment of investigating mixed reality frameworks, there were no inexpensive toolkits available that served the describes features and properties. This technological gap fostered the development of a cost-efficient software framework that enables quick prototyping of collaborative and distributed mixed reality environments. As existing toolkits and approaches have drawbacks regarding costs, usability, flexibility and extensibility, the implemented framework can act as foundation to further foster the simplification of application development and thereby the pervasiveness of mixed reality in general. Therefore, the proposed framework concludes the contributions of this thesis.

1.2 Organization

This part is organized as follows. After an overview over related frameworks is given in Chapter IV.2, the design approach of the proposed framework is described in Chapter IV.3. In Chapter IV.4, examples of applications that have been developed with the proposed framework are presented and a summary is given in Chapter IV.5.

Chapter 2

Background & Related Work

Developing and authoring mixed reality applications requires a lightweight and flexible but still powerful hard- and software framework, which is expendable to easily integrate new devices and technologies. Ideally, it supports diverse input and output devices, high quality real-time rendering, physics support, networking and scene management to build rich 3D applications.

2.1 Key Elements of a Mixed Reality Framework

A wide variety of hardware and software setups has been built in the past and all share a common general system architecture [53] that is illustrated by the modules depicted in Figure 2.1.

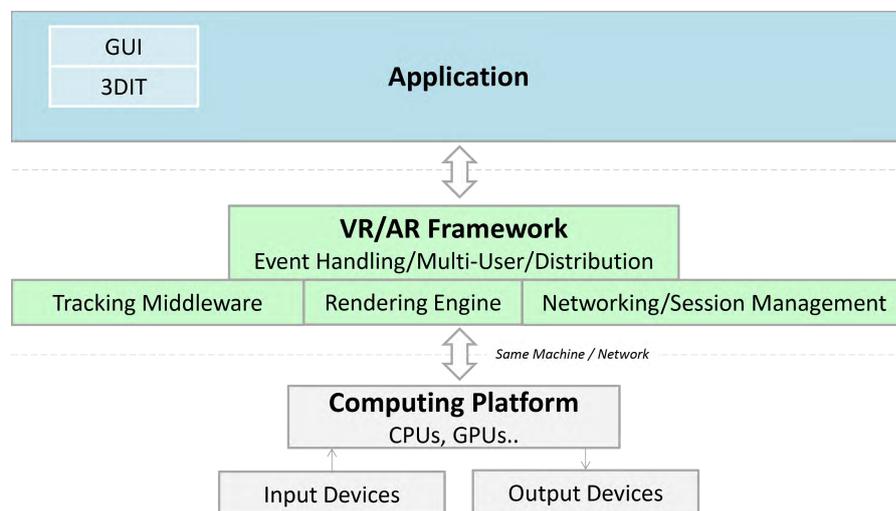


Figure 2.1: Mixed Reality system architecture.

The depicted general architecture can be applied to create non-immersive to fully

2. BACKGROUND & RELATED WORK

immersive mixed reality applications. Non immersive systems include 2D (multi)-screen setups, such as a desktop environment, where the user usually sits or stands in front of the screen interacting with a stationary input device, i.e. a joystick or 3D mouse. Semi-immersive scenarios employ a stereo projection with shutter glasses while the user's head and interaction device is tracked in space. Fully immersive setups are provided by using a multi-screen CAVE projection setup [16] with shutter glasses or by using head mounted displays for visualization. Again, the user's head and its interaction device (or the entire body) are tracked for visualization and interaction [83].

The hardware components (gray) of a mixed reality framework comprise input and output devices and a computing platform (e.g. workstation, mobile device) for device communication with a powerful graphics processor for 3D scene rendering. The software modules (green) of the middleware handle the tracking data, perform the 3D visualization and provide networking to allow a client-server based framework for single or multi-users. The middleware components communicate with the application layer that provides 2D and 3D graphical user interfaces (GUI), 3D interaction techniques (3DIT), 3D scene elements and layout as well as application specific behavior. The spatial position and orientation of the input and output devices might be tracked to apply 6DOF pose estimation. Tracking data of these devices is received by the computing platform and handed over to the framework's tracking middleware. The middleware processes, merges and transforms the input data to provide it in a consistent data format for subsequent usage within the application. Using this input data, 3D interaction techniques can be provided to the user by employing an event handling mechanism. Subsequently, the virtual scene is visualized to the user on its output device using the rendering engine. As visualization, tracking and interaction are fundamental components of a mixed reality application, multi-user support as well as 3D scene distribution are optional assets to allow for collaborative and distributed mixed reality setups. In such a case, the framework's networking and session module handles the connections of all users within the network and controls the communication amongst them to ensure correct event and scene synchronization.

2.2 Application Development & Scene Management

Since the mid-1990s, a number of mixed reality frameworks have been developed and a variety of systems supporting distributed applications emerged [43]. They mostly provide the integral components of a mixed reality application in an integrated development environment (IDE) to simplify application development and presume programming know-how. To further ease application prototyping and to provide a clear representation of the rendered virtual scene, 3D object management and scene authoring is advisable using a graphical user interface. Most of the high level programming toolkits are based on scene graph libraries, for example open source toolkits such as Studierstube [52], VR Juggler [42], Avango [92] or commercial ones like 3DVIA Virtools [168] and provide a complete framework for developing mixed reality applications. Studierstube is an application framework for collaborative augmented reality and incorporates all necessary

functionality such as scene graph rendering, networking, window management and support for input devices. It offers tracking of multiple input devices that are configured using XML files and allows multiple users that are embedded as nodes in the scene graph. While this C++ based framework is very powerful, it has several drawbacks regarding ease-of-use for application prototyping and cross-platform compatibility. While the open-source components allow deployment for Windows and Linux platforms, mobile devices are not supported. Furthermore, it lacks a state-of-the-art rendering engine that provides physics support and does not offer a graphical user interface for 3D scene management and authoring. Commercially available systems, i.e InstantReality [149] and MiddleVR [150], enable rapid application development with a comprehensive graphical user interface and support a wide variety of tracking and output devices. As drawback, only simple point and click metaphors [150] are provided as 3D user interface. 3DVIA Virtools [168] is a commercial development and deployment platform for interactive 3D content creation. It supports multiple users and physics behavior to create immersive and distributed applications using industry standard mixed reality peripherals. It offers a comprehensive graphical development environment and can deploy to a wide range of output devices. However, all three frameworks are cost intensive or just free to use in a private context.

Frameworks such as BuildAR [152] and DART [59] focus on enabling mixed reality application development by non-programmers. Using BuildAR, the programmer can associate virtual models with visually tracked planar markers. However, it does not provide more complex tracking behaviors, object interaction or a broader choice of tracking devices. One of the first AR frameworks using off-the-shelf software to design and develop mixed reality applications was the Designers Augmented Reality Toolkit (DART) [59]. DART is a plug-in for the popular Macromedia Director multimedia programming environment. It uses the familiar Director paradigms of a score, sprites and behaviors to allow a user to visually create complex mixed reality applications. DART also provides low-level support for the management of trackers, sensors, and cameras via a Director plug-in Xtra. However, DART is expensive due to licensing costs for Director. In addition, the time line based scene management is rather made for story telling environments than for non-linear mixed reality applications. Although there are several frameworks for building mixed reality systems on a stationary workstation, there is little support for handheld mixed reality [111]. Furthermore, none features straightforward integration of novel hardware devices and techniques while being cost efficient and providing an intuitive scene management to create collaborative distributed mixed reality applications.

Similar to Virtools, Unity3D [167] provides an editor for authoring 2D and 3D content and compromises a game engine for executing and rendering the 3D application. Nevertheless, Unity3D by itself is not a mixed reality framework since it lacks support for tracking and interaction. It is rather designed for creating 3D video games and other interactive content. It offers a powerful render engine providing lighting, physics, network communication for collaboration and content distribution. Furthermore, it provides an integrated programming environment using C#, JavaScript or Boo while development can be done under Windows as well as Mac OS X. The final application can be built

2. BACKGROUND & RELATED WORK

– generally without changes – for various platforms such as Windows, Mac, iOS, Android, all major game consoles, Flash and web clients. For private and research purpose, Unity3D is available for free and applications can be deployed at no charge to Windows, Mac, iOS and Android. This makes this software a compelling component for scene management, rendering and distribution in a mixed reality framework.

Chapter 3

Framework Architecture

Regarding our motivation from Section 1.1, the aim was to develop a loosely coupled, modular mixed reality framework which can easily be adapted to support emerging devices and interaction techniques. Furthermore, multiple user in a distributed environment shall be supported, providing non-immersive to fully immersive mixed reality setups as well as handheld scenarios. The proposed software architecture borrows from best design practices, as illustrated in Figure 2.1.

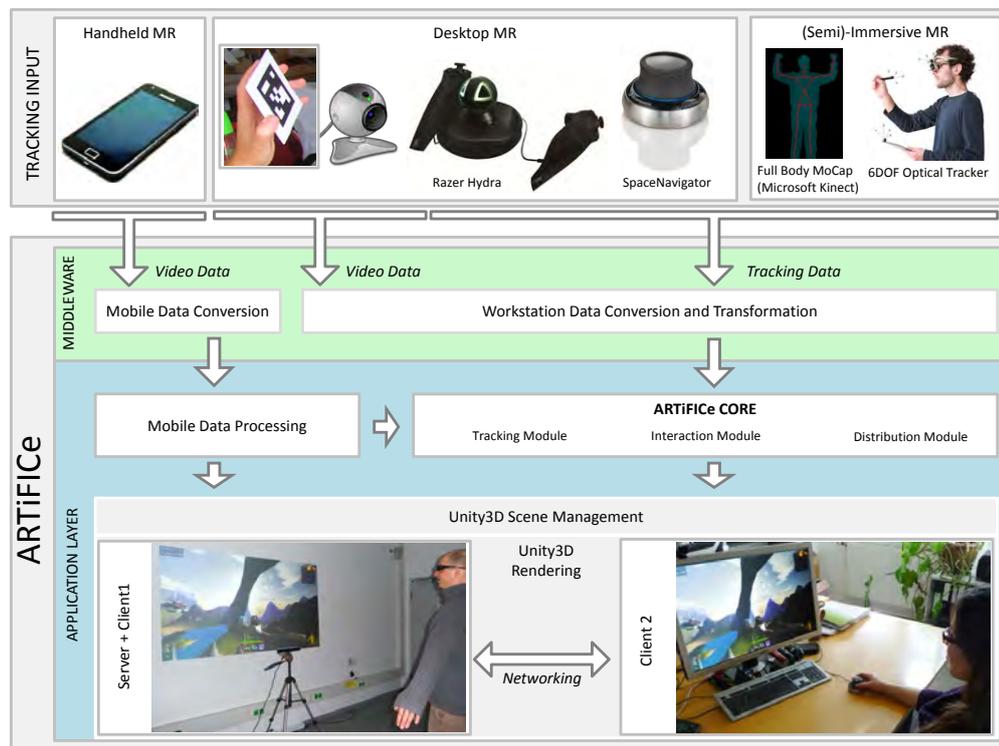


Figure 3.1: ARTiFiCe framework components and data flow.

3. FRAMEWORK ARCHITECTURE

An overview of the developed *Augmented Reality Framework for Distributed Collaboration* (ARTiFICe) with its components and the data flow is illustrated in Figure 3.1. Tracking data from the workstation-based input devices as well as from handheld devices are fed into ARTiFICe using the middleware layer that transforms all input data in a consistent way and delivers it to the application layer. The application layer is built on top of the external game engine Unity3D [167]. Within the application layer, the ARTiFICe core handles the tracking input data, provides interaction techniques and distribution support and delivers the data to the game engine's scene management. The virtual scene with real-time interaction is then visualized on different output devices using the game engine's rendering module. The ARTiFICe core defines a unified tracker object to provide the input data from the middleware that can be accessed for visualization and interaction. Furthermore, the ARTiFICe core comprises an interaction module with well-defined interfaces to integrate selection and manipulation techniques. Besides single-user 3D interaction, the co-presence of multiple users interacting with the same content at the same point in time opens up great possibilities for collaborative work. Therefore, a distribution module was integrated into the ARTiFICe core to enable real time user-managed collaboration for various hardware setups of two or more users over the network. It distributes the scene as well as user interaction in real time and was built upon the networking layer of Unity3D.

3.1 Base Infrastructure

ARTiFICe uses Unity3D, an "integrated authoring tool for creation of 3D videogames" [167], as base infrastructure for scene authoring, rendering and for its application layer.

3.1.1 Functionalities of Unity

The free to use license of Unity3D offers a powerful Application Programming Interfaces (API) to create projects in JavaScript, Boo and C#. These projects can be deployed without any further changes to multiple platforms, including Windows, OSX and Linux, iOS and Android, various game consoles and a special web player for online deployment. Unity's 3D rendering engine supports both DirectX and OpenGL. Furthermore, the Nvidia (previously Ageia) PhysX engine is included and supports real-time physics simulation such as object collisions and casts, forces and multiple joints. For 3D content, Unity natively provides only creation of very simple shapes, such as cubes, spheres and cylinders. More sophisticated 3D meshes can be imported using common formats such as .FBX, .OBJ, COLLADA, as well as models created in 3D Studio Max, Blender and Maya.

3.1.2 Core Concepts of Unity

The Unity scene management offers a rich GUI to place and arrange 3D objects, such as geometry, virtual cameras in space. All objects of a scene are organized in a hierarchical order that follows the basic principles of a 3D scene graph. Each object in the

hierarchy is represented by the Unity basic class *GameObject* that acts as a container for all kind of objects. Each *GameObject* can be enhanced by so called *Components* to control the *GameObject*'s transformation including position, rotation and scale, its appearance, rendering and physics behavior. Therefore, a *Component* is "attached" to a *GameObject*. While there is a magnitude of pre-defined *Components*, it is also possible to create specific behavior by implementing them in custom scripts using the Unity API and attaching those scripts to the *GameObject*. These two core concepts form the foundation for the development of ARTiFICe's core that resides as a script hierarchy within Unity's Integrated Developing Environment (IDE). For an in-depth description of Unity's functionality, the reader is referred to [167].

3.2 Middleware

To process the tracking input and to provide it to the application layer, the Artifice's middleware layer uses OpenTracker [46] to gather tracking data of various workstation-based input devices and Vuforia [163] for 6DOF estimation of a handheld device.

3.2.1 OpenTracker

OpenTracker [46] is an open-source software framework that serves as connection between the input devices and the application layer and communicates with the ARTiFICe core. It reads out tracking data from the input devices using appropriate drivers, transforms the data in a consistent format, fuses multiple tracking sources and finally delivers the data via a transport mechanism. To fetch tracking data from remote input devices, OpenTracker supports the *Virtual-Reality Private Network* [47] (VRPN) that is a device-independent and network-transparent framework for devices used in mixed reality systems. Thereby, it provides a hardware abstraction layer and eases the development and maintenance of hardware setups in a flexible manner. This is achieved by using an object-oriented design based on XML and utilizing standard XML tools for development, configuration and documentation. To describe the employed tracking configuration, a data flow graph is defined via a XML file complying to a predefined DTD. A multi-threaded execution model takes care of filters and transformations that are applied to the tracking data. The underlying data flow graph can be described by the following three XML node types:

Sources: This is the entry point for all tracking data. Typically, a source node is a wrapper of a specific device driver.

Filters: A filter node performs the actual work of processing the input data to be able to deliver it in a consistent way to the application layer. There is a great number of available filter nodes, such as geometric transformations, conversions to translate one data type into another or filters to merge tracking data from multiple inputs by combining them into a new data format.

3. FRAMEWORK ARCHITECTURE

Sinks: The sink node is mostly responsible for distributing the filtered data to the application that communicates with OpenTracker.

Extending OpenTracker On start-up, the XML configuration file is loaded and parsed to generate the data flow graph by dynamically instantiating the defined nodes. However, this convenient way to configure the interaction devices and to connect them to the application layer is only given if both hard- and software are fully integrated in OpenTracker. The native OpenTracker implementation does not provide support for i.e. Razer Hydra [142] and the 3D Connexion SpaceNavigator [147]. Since these devices have great potential to enable intuitive 3D interaction in a desktop mixed reality scenario, two novel source nodes were implemented as further described in Section 3.2.3.1. To further support Artoolkit markers [35, 87] as well as optical tracking and full body motion capturing, as outlined in Section 3.2.3, existing OpenTracker source nodes were used.

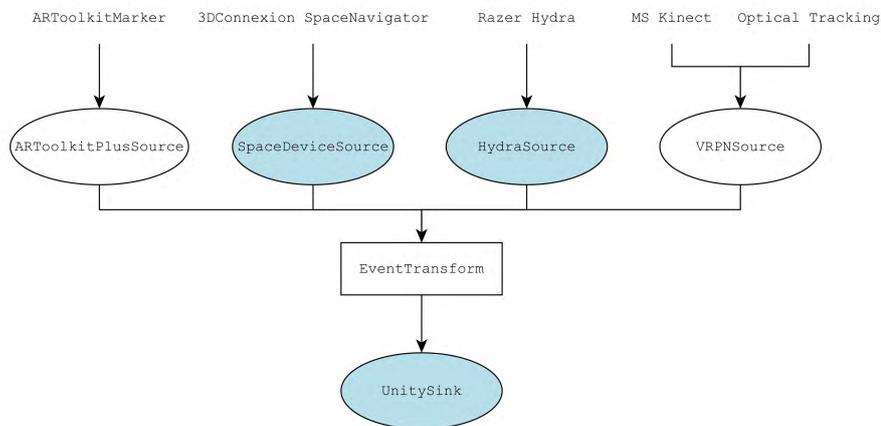


Figure 3.2: OpenTracker nodes with new ones marked in blue.

Furthermore, OpenTracker did not provide by default a sink node to communicate with Unity3D. Therefore, a new OpenTracker sink node *UnitySink* was implemented to provide a single sink for all tracking devices to link them with Unity. The *UnitySink* node is referenced during run-time by the ARTiFICe core for fetching tracking data to provide them within the application. An overview of the extended OpenTracker architecture that is employed within the middleware layer of ARTiFICe is depicted in Figure 3.2. An example XML configuration file is given in Listing 3.1.

3.2.2 Vuforia

Vuforia [163] is a software development kit (SDK) to create augmented reality applications for handheld devices. It uses natural features (see Chapter II.2) of planar or volumetric objects to determine in a frame wise manner the 6DOF pose of the handheld device’s camera, relative to the object. The object has to be registered using the Vuforia Target Management System in an off-line process before it can be tracked by the online

Vuforia processing pipeline. It provides native SDKs for Android with an Application Programming Interfaces (API) in Java and Java/C++ as well as for iOS in Objective C. The Vuforia AR Extension for Unity furthermore provides the pose tracking functionality within the Unity IDE. Currently, Vuforia is compatible with a broad range of mobile devices, such as the the iPhone (4/4S), iPad, and Android phones and tablets running Android OS version 2.2 or higher.

3.2.3 Supported Setups & Hardware

Using OpenTracker and Vuforia, a wide range of tracking input devices are linked to Unity3D to enable further mixed reality specific behavior, provided by the ARTiFICe core. For workstation-based devices, either existing OpenTracker source nodes were used or novel ones were implemented. To enable mobile devices, Vuforia was used by ARTiFICe. A comprehensive overview of all supported tracking devices by ARTiFICe is given in Table 3.1. Beyond this table, all devices that are natively supported by OpenTracker

Device-Name	Software Development Kit	
	Existing Node	New Node
ARToolkit Markers	OT <i>ARToolKitPlusSource</i>	
3D Connexion SpaceNavigator		OT <i>SpaceDeviceSource</i>
Razer Hydra		OT <i>HydraSource</i>
MS Kinect	OT <i>VRPNSource</i>	
Optical Tracking	OT <i>VRPNSource</i>	
Handheld Device	Vuforia	

Table 3.1: Interaction devices supported by ARTiFICe.

and by VRPN can be used within ARTiFICe as well. The flexible middleware concept allows configuration of all these devices in various combinations using a single OpenTracker XML configuration file. Configuration of mobile devices is treated separately using Vuforia. With the supported tracking input, ARTiFICe enables the creation of desktop-based, semi-immersive, full immersive as well as handheld mixed reality environments that are described in the following.

3.2.3.1 Desktop Mixed Reality

For desktop setups, ARToolKit [35] as well as ARToolkit+ [87] are tracking libraries providing projective-invariant planar bitmap patterns for 6DOF pose estimation that encode a unique number for distinguishing multiple markers (see Chapter II.2). ARToolkit is usually employed in desktop based mixed reality environments while ARToolkit+ enhances the original ARToolkit library and is optimized for usage on handheld devices. ARToolkit+ is used in ARTiFICe framework, which has been previously integrated into OpenTracker. To enable live video view within a deployed Unity project, OpenVideo [161], a data integration- and processing toolkit, is used. It acquires video

3. FRAMEWORK ARCHITECTURE

frames from the connected webcam that are subsequently processed by ARToolkit+. The video is then streamed into Unity3D to provide a view of the real world scene while interacting with the planar bitmap pattern. For more enhanced 3D interaction, the 3D mouse SpaceNavigator from 3D Connexion [147] was integrated by wrapping its native driver into a new OpenTracker source node. Furthermore, the two-handed interaction controller Razer Hydra [142] was integrated into OpenTracker, as described in detail in [122].

3.2.4 (Semi) Immersive Mixed Reality

Model based optical tracking, as described in detail in Chapter II.4 can be employed to track the user’s head and interaction device in a semi or fully immersive mixed reality environment. For room-sized environments, the passive optical tracking system [84] was integrated into ARTiFICe using VRPN [134]. The 6DOF pose tracking data is read by the existing OpenTracker *VRPNSource* node, transformed and provided to the ARTiFICe core using the *UnitySink* node.

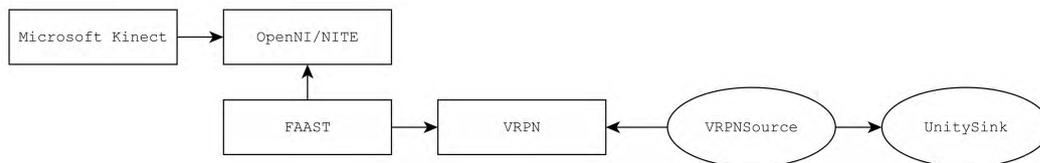


Figure 3.3: ARTiFICe’s processing pipeline of depth data for full-body motion tracking.

With emerging depth sensing technology, such as the Microsoft Kinect [127], markerless full-body motion tracking becomes more and more popular for user tracking and device-less 3D interaction in a mixed reality environment. Therefore, the Kinect was integrated using OpenNI/NITE [160, 162] and FFAST [165, 128]. OpenNI/NITE provides an API to access raw depth data as well skeleton data, which are calculated based on the depth data. FFAST runs as self-contained application and reads this data. It provides gesture recognition and supports streaming of the full body tracking data over VRPN. Using the *VRPNSource* node and the *UnitySink* node, this data is read and fed into the ARTiFICe core. The entire pipeline is depicted in Figure 3.3.

3.2.4.1 Handheld Mixed Reality

A modern mixed reality framework should support handheld devices to allow for mobile augmented or virtual reality setups. Due to its powerful properties and its fine-tuned integration into Unity3D, Vuforia [163] is integrated into the middleware layer of ARTiFICe. Over the ARTiFICe framework, it is interfaced to the ARTiFICe’s core to process the mobile tracking data, as described in Section 3.3.

3.3 Application Layer

The middleware components communicate with the application layer that comprises Unity3D and the embedded ARTiFICe core. Unity’s graphical user interface as well as its IDE are used for 3D scene authoring and application prototyping and its rendering engine is employed for 3D visualization. The ARTiFICe core comprises a Manager and a tracking-, interaction- as well as distribution module and is embedded into the Unity3D IDE. In Figure 3.4, a detailed view on the framework with its data flow and components is given.

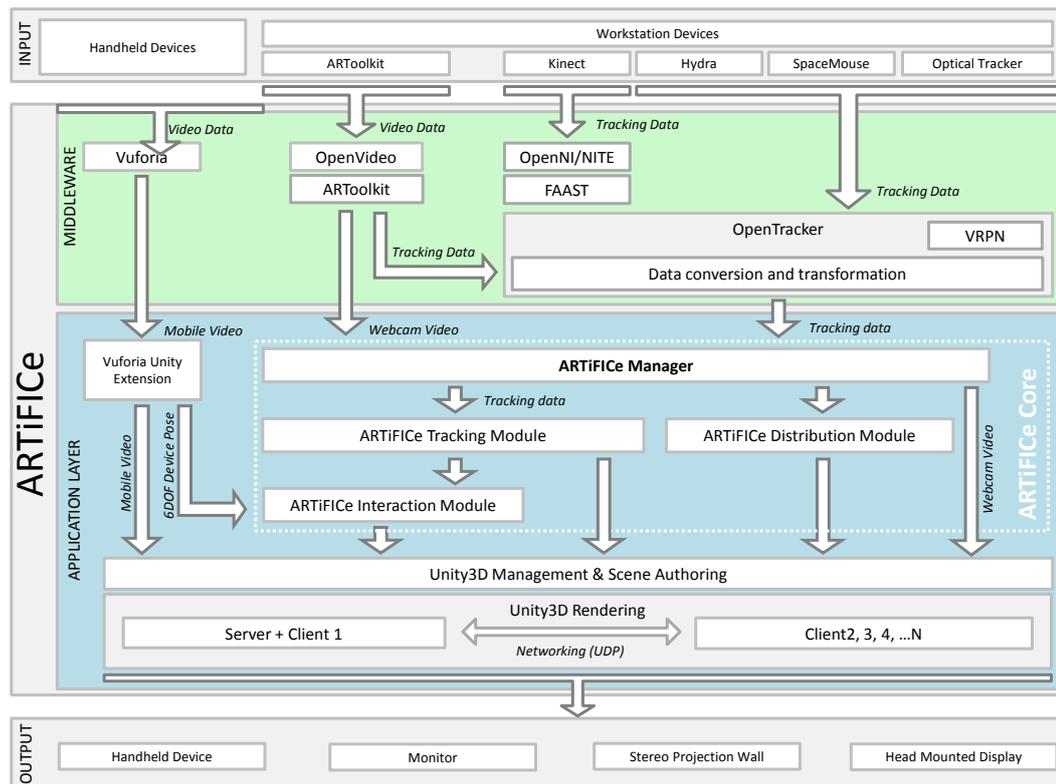


Figure 3.4: Detailed framework components.

3.3.1 The ARTiFICe Manager

The ARTiFICe *Manager* controls the data flow between middleware and application layer. Upon application start-up, it reads the OpenVideo and OpenTracker configuration files and loads the dependent tracking libraries. It starts an OpenTracker instance and an OpenVideo handler for ARToolkit+ marker tracking. It also closes OpenVideo and stops OpenTracker at application shutdown.

3. FRAMEWORK ARCHITECTURE

3.3.1.1 Tracking Module

The *Tracking Module* reads the tracking data of the connected input devices and feeds it into the transformation component of a Unity3D *GameObject*. The overall design of the tracking module is shown in Figure 3.5. It derives from *TrackBase* for workstation-based devices, respectively from *Vuforia.TrackerBehaviour* for handheld devices. Since these two classes inherit from the Unity3D base class *MonoBehaviour*, the deriving classes are capable to be attached to any scene object within the Unity3D hierarchy.

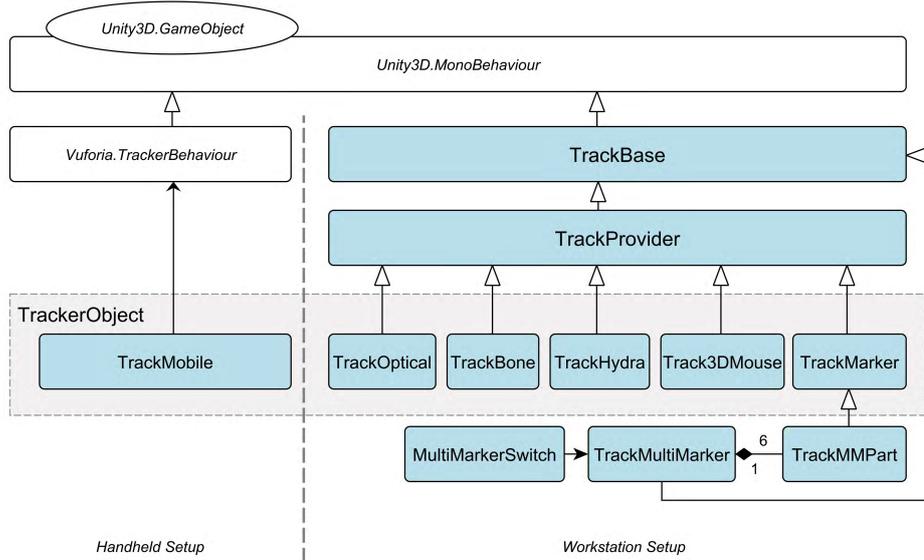


Figure 3.5: Tracking class hierarchy.

For each of the supported workstation-based input devices, a subclass was implemented to provide the specific tracking data depending on the attached devices. Upon application start, *TrackProvider* creates *ARTiFICe Trackers* through the *ARTiFICe Manager*, which is implemented as singleton. Each *ARTiFICe Tracker* is interfaced to the corresponding *OpenTracker Unity node*. For planar bitmap marker tracking, a multi-marker tracking support was implemented to be able to track cuboid-formed 3D objects and determine their absolute physical 6DOF pose. To access the handheld device, *Track-Mobile* reads from *Vuforia.TrackerBehaviour* that interfaces the Vuforia tracking core in Unity3D's IDE.

All tracking subclasses provide *Tracker Objects* that form a consistent tracking data layer and can be accessed by the *ARTiFICe's* interaction and distribution module for further processing.

3.3.1.2 Interaction Module

The raw tracking data of a connected input device can be accessed using a *Tracker Object*, as described in Section 3.3.1.1. It can be subsequently used for 3D object selection and

manipulation, as depicted in Figure 3.6.

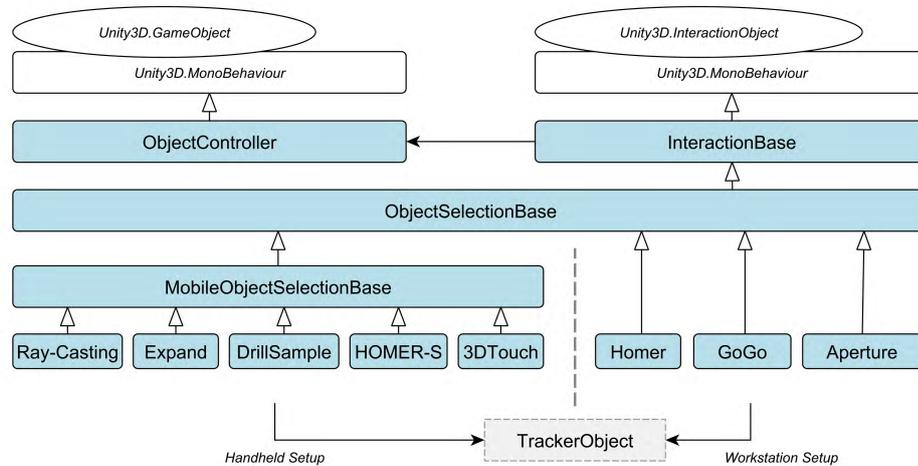


Figure 3.6: Interaction class hierarchy.

The data of the tracker object is processed by the specific interaction technique that can be attached to any scene object, i.e. to visually represent a virtual hand. Each concrete interaction technique inherits from the abstraction layer *ObjectSelectionBase* that provides a clean interface of data handling for workstation as well as handheld devices and offers a transparent layer to integrate new techniques into the framework.

At run-time, the concrete interaction technique determines the currently selected scene objects as well as calculates its absolute 6DOF pose. This data is then handed over in a uniform format to *ObjectSelectionBase* which is further processed by the *InteractionBase* class and delivered to all selected virtual scene objects. Virtual scene objects that are selectable must have the *ObjectController* class attached. By reading the data from *InteractionBase*, the *ObjectController* checks if the scene object to which it is attached to is selected and if it is, it manipulates the position and orientation depending on the given pose.

As concrete 3D interaction techniques, a number of state-of-the-art interaction techniques were implemented, such as a simple VirtualHand, GoGo [22], Aperture [21] and HOMER [24]. For 3D manipulation in a handheld mixed reality environment, the novel interaction techniques DrillSample, 3DTouch and HOMER-S, as described in Chapters III.3 and III.4, are integrated into the framework. As shown in Figure 3.6, the class *MobileObjectSelectionBase* acts as an interface for these interaction techniques. The class inherits from *ObjectSelectionBase* and provides a common layer to gain access to handheld specific hardware functionality, such as touch input.

3.3.1.3 Collaboration & Distribution

To provide multi-user support for interaction using different interaction devices and remote collaboration of one virtual scene, a collaboration and distribution module was

3. FRAMEWORK ARCHITECTURE

furthermore implemented. It is loosely coupled with the interaction module and enables distribution of both mobile and all workstation setups. The networking functions are based on the Unity3D network layer using the *User Datagram Protocol* (UDP) for communication. A client-server architecture is applied with a direct connection between the server and all clients, resulting in a *Star Topology*. For data exchange, remote procedure calls (RPC) and state synchronization are employed. To prevent data loss, the state synchronization is buffered.

An overview of the distribution module and its connection to the interaction module is given in Figure 3.7. The *NetworkBase* class provides functions to initialize the server and to connect a client to the server. All connected clients are managed by the *UserManager* class, implemented as singleton. To reduce necessary hardware for realizing a client-server application and to improve overall usability, one device can act simultaneously as server and client.

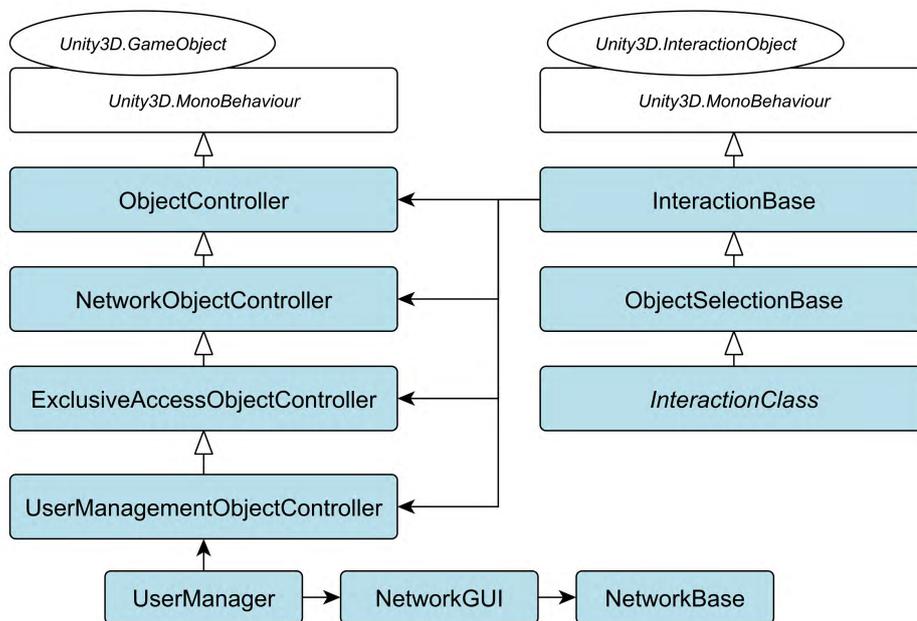


Figure 3.7: Distribution class hierarchy.

To enable multi-user collaboration of a virtual scene, all user-specific interaction must be distributed as well. Therefore, each selectable scene object must have a *NetworkObjectController* component attached that distributes selection and manipulation functionality over the network. To enable exclusive access to a scene object, *ExclusiveAccessObjectController* prevents simultaneous usage by multiple users. As long as a user selects and manipulates the scene object, it is locked for other users. To provide exclusive object access to a specific user, the *UserManagementObjectController* is used.

3.4 Workflow for Application Development

With the proposed middleware and application layer components, a new mixed reality application can be developed using the following steps.

1. A new Unity3D project is created and the ARTiFICe framework is added to the project by copying the sources into the project's folder hierarchy under *Assets*.
2. The desired workstation-based interaction devices are then configured using the single OpenTracker XML file. An example is given in Listing 3.1, configuring one ARToolkit+ marker as well as the SpaceNavigator as input devices. Both are filtered in terms of transformation to ensure a common orientation of the tracking input.
3. If the application is deployed as a handheld mixed reality setup, Vuforia is integrated into the Unity project, as described on the Vuforia developers page [163].
4. Virtual cameras, lights, interaction and selectable scene objects are created and added to the 3D environment using the Unity3D graphical scene management. They are encapsulated as Unity3D *GameObjects* and can be subsequently connected to the according classes of the ARTiFICe core modules.
5. Finally, the project is built and deployed to the desired platform using Unity's built-in deployment tool.

Listing 3.1: An OpenTracker example configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OpenTracker SYSTEM "opentracker.dtd">
<OpenTracker>
  <configuration>
    <ARToolkitPlusConfig camera-parameter="camera-calibration.cal" />
    <SpaceDeviceConfig />
  </configuration>
  <UnitySink name="Marker0">
    <EventTransform scale="1 1 1" rotationtype="euler" rotation="0 0 3.14159"
translation="0 0 0">
      <ARToolkitPlusSingleMarkerSource center="0 0" size="0.08 0.08" tag-id="0"/>
    </EventTransform>
  </UnitySink>
  <UnitySink name="SpaceMouse">
    <EventTransform scale="0.01 0.01 0.01" rotationtype="euler" rotation="1.57 0 0"
translation="0 0 0">
      <SpaceDeviceSource />
    </EventTransform>
  </UnitySink>
</OpenTracker>
```


Chapter 4

Developed Mixed Reality Environments

ARTiFICe was intensively tested and used within research projects as well as for teaching.

- The framework was applied as the technological foundation for the *Virtual and Augmented Reality* laboratory exercise in the graduate program of Vienna University of Technology from winter term 2011/12 on until now. In total, more than 150 students developed distributed and collaborative mixed reality applications with ARTiFICe, using several interaction techniques in combination with ARToolkit markers, 3D Connexion SpaceNavigator and Microsoft Kinect for Windows.
- ARTiFICe was employed for the laboratory exercise *Augmented Reality* as a part of the graduate program *Mobile Computing* at the University of Applied Sciences Upper Austria during winter term 2011/12 and 2012/13. With the help of the framework, more than 30 students developed a distributed and collaborative application for handheld mixed reality within just four weeks, using HOMER-S and 3D Touch.
- The framework is an integral component of research projects in the field of interaction and tracking at the Interactive Media Systems Group at Vienna University of Technology to enable rapid prototyping. Within the projects, ARTiFICe is subject to continuous development.

4.1 Test Setups & Environment

In the following sections, we demonstrate an excerpt of the setups that have been developed with ARTiFICe. The presented mixed reality environments feature different combinations of processing platforms and hardware for in- and output, and provide varying levels of immersion (see Chapter I.1). The framework was tested on various workstations, running Windows 7 (32/64bit). All parts of the framework, except Kinect and ARToolkit, can also be deployed on Mac OS X/iOS. The handheld mixed reality setup

4. DEVELOPED MIXED REALITY ENVIRONMENTS

was tested on multiple Android devices, all running a minimum of Android v2.2 featuring an ARMv7 architecture or higher.

4.2 Non-Immersive Mixed Reality

A *Non-Immersive* mixed reality environment usually consists of a non-stereoscopic screen through which the user observes the virtual scene, making the screen a window into the virtual world. In such a setup, the user is fully aware of the reality that surrounds him or her, resulting in a feeling of non-immersion. In the following, two typical non-immersive scenarios, a desktop as well as a handheld setup are presented.

4.2.1 Single & Multi-User Desktop Mixed Reality

Two mixed reality desktop applications were realized. In the first, as depicted in Figure 4.1a, a multi-user collaborative and distributed augmented reality simulation was developed using multiple ARToolkit+ markers as input and interaction devices.



(a) ARToolkit interaction.

(b) Interaction with Razer Hydra.

Figure 4.1: Two examples of desktop mixed reality setups.

A portion of the markers form a MagicBook [41] that was used for interactive story telling. The other portion of the markers acts as a cube that was employed as a multi-purpose interaction device, using the multi-marker tracking capabilities of the framework (see Section 3.3.1.1). All markers in the scene are centrally organized in one OpenTracker XML configuration file and were tracked by a low-cost off-the-shelf camera (*Logitech Webcam C905*). The virtual scene as well as any user interactions are distributed to all

connected clients using the ARTiFICe distribution module while the workstation of one user acts simultaneously as server and client.

The second desktop-based setup employs a Razer Hydra [142] as a high-precision 6DOF interaction device to realize a single-user virtual reality training environment. In an application for geometry education [122], virtual scene objects can be created, controlled and manipulated using the Hydra, as illustrated in Figure 4.1b. Thereby, spatial abilities as well as a deeper understanding of 3D geometry can be trained by using a low-cost setup that allows for seamless 3D manipulation.

4.2.2 Multi-User Handheld Mixed Reality

As an example for a non-immersive handheld mixed reality environment, a collaborative and distributed application was developed. It provides a multi-user augmented reality game in which users can interact with the physically driven virtual scene objects using HOMER-S. Again, the virtual scene as well as any user interactions are distributed to all connected clients using the ARTiFICe distribution module while the mobile device of one user acts simultaneously as server and client.

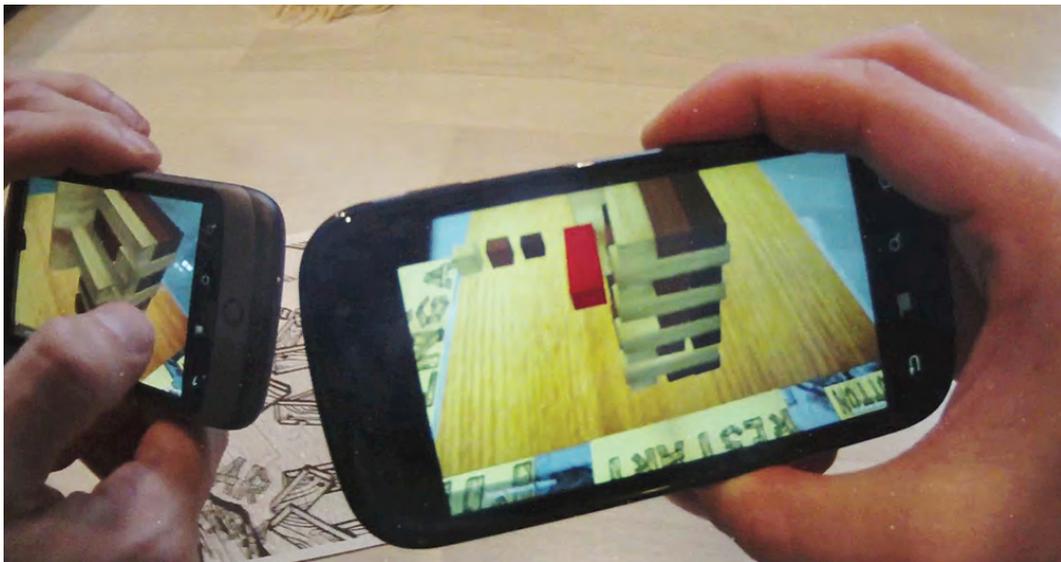


Figure 4.2: Multi-user collaborative and distributed handheld mixed reality.

As shown in Figure 4.2, the user on the left hand side currently translates a virtual brick in space while the user on the right observes this interaction. To enable 6DOF pose tracking, an arbitrary image is registered in an off-line process with the natural feature tracking toolkit [163]. At runtime, the image is used as playground and is augmented with the virtual scene that can be observed through the handheld's device screen. Multiple users can collaborate and interactively play together, either by pointing their phones on the same physical image or at different images at distributed locations that show the same motive.

4. DEVELOPED MIXED REALITY ENVIRONMENTS

4.3 Combined Non- & Semi-Immersive Mixed Reality

Furthermore, ARTiFICe can be employed to create collaborative and distributed mixed reality setups that offer different levels of immersion. In Figure 4.3, a collaborative and distributed multi-user setup is shown providing a non-immersive environment for User 1 and a semi-immersive setup for User 2. *Semi-Immersive* environments provide an increased amount of immersion by enabling stereoscopic viewing through shutter glasses and 3D interaction using mobile 6DOF devices, such as 3D pens (see Figure III. 2.1c) or motion capturing.



(a) Non-immersive setup using a stationary 6DOF interaction device. (b) Semi-immersive stereo projection setup with full body motion capture.

Figure 4.3: A distributed multi-user non & semi-immersive mixed reality setup.

The combined non- and semi-immersive distributed setup is achieved by supporting a different set of in- and output devices for each user. A game was developed as test application in which two users have to collaboratively control a flying bird through a virtual environment. While the first user (Figure 4.3a) views the scene on a screen and interacts with 3D Spacenavigator to control the attitude as well as clearing the bird's flight path using the GoGo interaction technique [22], the second user (Figure 4.3b) is provided with a stereoscopic scene view and controls the speed and direction of the virtual character by full body motion capturing and gesture recognition, using Microsoft Kinect [127] as input. Both users interact in different physical locations and are connected over the ARTiFICe distribution module.

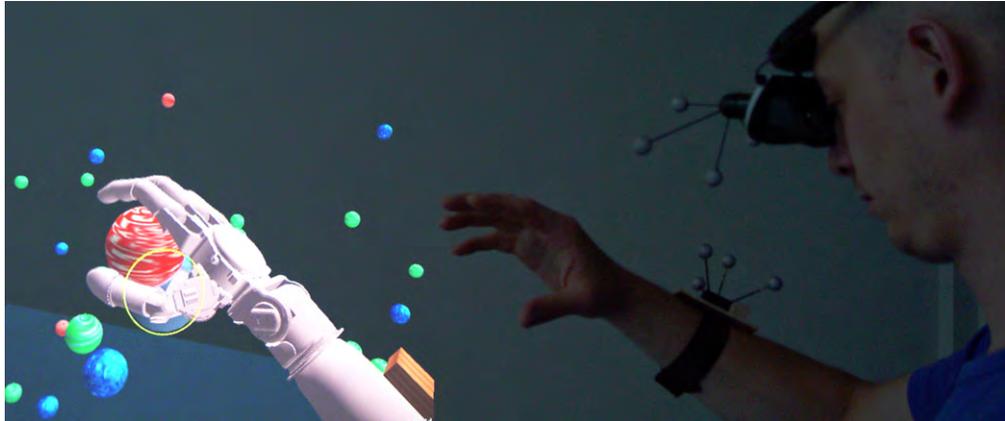
4.4 Combined Semi- & Full Immersive Mixed Reality

Furthermore, ARTiFICe has amongst others also been employed for serious game development. A virtual reality training was created based on ARTiFICe to support upper limb prosthesis patients in learning to control their myoelectric prostheses, even before they have access to the physical ones [139, 134]. The software consists of a server application to control the training parameters, and a client module to visualize the virtual environment to the user in a head mounted display (HMD).

In Figure 4.4, a test setup of this fully immersive application is shown. Both HMD and the user's upper arm are tracked using optical tracking. Thereby, the user is provided

4.4 Combined Semi- & Full Immersive Mixed Reality

with an egocentric scene view and can control the position and orientation of the virtual prosthesis.



(a) A detailed view of the immersive virtual reality



(b) The combined immersive and semi-immersive virtual reality

Figure 4.4: A distributed multi-user non & semi-immersive mixed reality setup.

The tracking data is sent to ARTiFICe through the OpenTracker VRPN node. An electromyographic (EMG) tracking device was integrated into the optical tracking target to detect muscle contraction for controlling grasping of the prosthesis, as shown in Figure 4.4a. The EMG data is sent via the wireless Bluetooth protocol to the workstation. As depicted in Figure 4.4b, the egocentric scene view can be displayed on a stereo projection wall for demonstration purposes to share the user's HMD experience for discussion and explanations.

Chapter 5

Summary

In this part, a flexible software framework named ARTiFICe is introduced to develop collaborative and distributed mixed reality applications. The framework follows a modular software architecture and features loosely-coupled, extendable modules for tracking, interaction and distribution. Built upon a state-of-the-art game engine Unity3D [167], the framework further provides high quality 3D rendering, physics support, a powerful graphical user interfaces for scene authoring and an integrated build tool to deploy the project for various hardware platforms. ARTiFICe's middleware is using Vuforia [163] and extends OpenTracker [46] to support tracking of various input sources, such as planar bitmap patterns, 3D mice, rigid body optical tracking targets as well as recently emerged, popular off-the-shelf devices, such as Microsoft Kinect, Razer Hydra and mobile devices running Android and iOS. The design of the middleware as well as the tracking module in ARTiFICe's application layer allow for a straightforward integration of new input devices. ARTiFICe's interaction module provides well-defined interfaces to integrate custom methods and offers a number of built-in techniques, including the proposed methods of Part III. Finally, ARTiFICe supports the distribution of scene content and user interaction to create remote mixed reality environments that can be shown on a wide range of devices, such as smartphones, stereo projectors and head mounted displays. Based on these functionalities, ARTiFICe provides the development of versatile mixed reality environments, ranging from non- to fully-immersive setups, that can run on different operating systems and platforms, including Windows and Android.

ARTiFICe was employed to create mixed reality environments for a number of scientific projects, including application development for the techniques that are presented in this thesis. Furthermore, the framework was used by more than 150 students during their university graduate program who were not familiar with mixed reality technology before. It allowed them to develop distributed applications within just a couple of weeks that incorporated different tracking devices and as well as interaction techniques. These results demonstrate the framework's applicability and usability for users, which are technically versed but do not have in depth knowledge in mixed reality. Thereby, it can support these non-experts to overcome the initial hurdles of creating advanced applications to create embodied mixed reality experiences. As existing toolkits and approaches

5. SUMMARY

have drawbacks regarding costs, usability, flexibility and extensibility, the results indicate that the implemented framework can act as foundation to further foster the simplification of application development and thereby the pervasiveness of mixed reality applications in everyday scenarios.

Conclusion

1	Findings & Outlook	199
1.1	Wide-Area Optical Tracking	200
1.1.1	Open Topics	201
1.2	3D Interaction	202
1.2.1	Open Topics	203
1.3	Creating Mixed Reality Environments	204
1.3.1	Open Topics	204

Chapter 1

Findings & Outlook

This thesis has focused on novel concepts and systems to leverage the applicability of mixed reality into unconstrained everyday environments. Therefore, we investigated concepts in the area of tracking, interaction and mixed reality frameworks, that resulted in the presented contributions, as depicted in Figure 1.1.

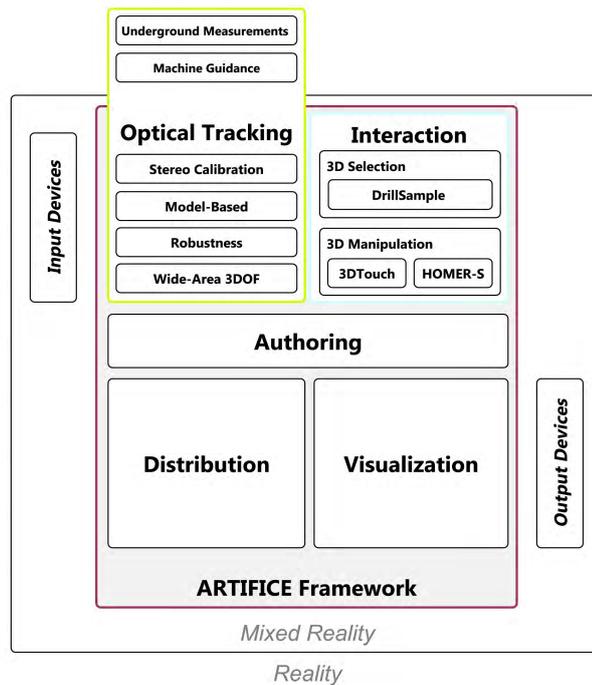


Figure 1.1: Investigated concepts, their relationship and the presented contribution.

For each of the investigated areas, we recapitulate our findings and give an outlook on open topics that are worthwhile to investigate and that we plan to conduct in the future.

1.1 Wide-Area Optical Tracking

The first part of this thesis has focused on optical tracking in large, unconstrained indoor environments. There, environmental conditions pose challenges in tracking volume coverage, tracking accuracy and disturbing interferences, such as static and moving lights, poor visibility and occlusions.

Our literature review revealed that state-of-the-art optical tracking systems are not capable to cope with the intended environments with a minimal vision hardware setup. Existing systems usually require a large amount of vision hardware to cover larger volumes and are sensitive to interferences, especially during target training and camera calibration. Therefore, they cannot provide accurate tracking without pre-conditioning the environment. To overcome the limitations of current approaches, we presented a robust and cost efficient wide area optical tracking system that estimates the 3D position of model-based targets up to $100m$ while requiring a minimal amount of two cameras. We extend the state-of-the-art in optical tracking systems by proposing a robust extrinsic stereo camera calibration, by introducing a highly re-configurable target design and by providing a software-based processing pipeline that enables the system to cope with large tracking distances, static and moving interfering lights, partly occluded targets as well as disturbances such as fog and dust during calibration as well as tracking.

To evaluate the developed hard- and software system, we conducted experiments in three different tracking scenarios that all feature large distances and unconstrained indoor environments. During the tests, we observed our system to robustly identify the target's model during stereo camera calibration and tracking in the presence of strong interfering lights, temporary occlusions as well as poor visibility, such as fog. The measurements of accuracy and stability up to $100m$ indicate that the proposed system outperforms competing optical tracking systems in terms of volume coverage, relative point accuracy and robustness. Furthermore, only a minimum of two cameras is required, leading to a significant reduction in system's cost and setup complexity. In addition, we demonstrated the system's abilities to act as a wide area tracking system for underground surveying tasks. This pushes the borders of optical tracking to a new application domain since state-of-the-art optical tracking approaches are exclusively designed and thus solely applicable for mixed reality applications. Our results indicate that our proposed system cannot compete with existing surveying measurement technologies in terms of relative point accuracy but outperforms existing systems in the following aspects. No manual sighting of a target is required, tracking of fast movements as well as of multiple targets at the same time can be provided and targets can be easily reconfigured to track static and portable objects as well as machines. Thus, our system acts as a first foundation for automated guidance for underground machine control.

We hope that our contributions help engineers and developers to foster the further emerging of mixed reality into everyday work and to improve automated surveying. For both application areas, a broad range of wide area tracking scenarios can be envisioned that are currently impeded by the limitations of state-of-the-art systems, such as user tracking at entertainment stages or in manufacturing workshops as well as for survey-

ing tasks such setting out, profile control, deformation monitoring, automated machine guidance.

1.1.1 Open Topics

Our evaluation revealed several open topics that we plan to address in future research.

- We plan to evaluate the relative point accuracy with different hardware setups using higher resolution cameras and lenses with smaller focal length to extend the field of view and thereby, the horizontal and vertical tracking coverage. Additionally, we will examine infrared LEDs with less radiant intensity to reduce the tracking target length. Both aspects can be beneficial especially for tracking at smaller distances up to $30m$.
- We will address the improvement of feature distribution in the camera image to enhance the estimation of external camera parameters in terms of robustness and accuracy. We found an unbalanced blob coverage of the artificially generated point features especially in the vertical dimension that is caused by limited human size and the length of the calibration target as well as by the natural boundaries of the physical environment, such as the ceiling and the ground. Therefore, we will investigate concepts to extract natural features from distinct environmental structures and fuse them with the blob features to increase the distribution along the edges and in the corner of the images. This approach requires a well illuminated environment with a sufficient amount of prominent geometrical structure that might be given in a standard indoor environment. In an underground scenario, where illumination is poor and geometric structures are mostly found around the front face, natural feature extraction would not significantly enhance the feature distribution in the camera images. Here, additional single IR-LED markers that are installed throughout the volume would be an adequate solution to improve the feature distribution. These single blob features could be autonomously detected and extracted using the proposed hardware interference filtering approaches from Chapter II.4. With these methods, we hope to achieve a more accurate calibration for stereo rigs with large baseline in both illuminated as well as poorly illuminated and non-cluttered environments.
- To obtain absolute 3D coordinates for surveying measurement tasks, linking the camera's coordinate system to the geo-reference coordinate system is required. The geo-reference coordinate system is obtained by geodesic measurements using a total station/theodolite. To determine the transformation matrix between the two coordinate systems, we plan to equip the tracking targets as well as additional stationary single point targets with geodesic prisms that are measured with a theodolite to obtain highly accurate geo-referenced 3D measurements.

1.2 3D Interaction

The second part of this thesis has focused on 3D interaction techniques in one-handed handheld mixed reality. We specifically investigated concepts for selection and manipulation of objects in dense mixed reality scenes. As tracking is the crucial foundation to enable interaction, *Inside-Looking-Out* optical 6DOF pose tracking is used as technological prerequisite for the presented interaction techniques.

To enable precise 3D object selection and manipulation (translation, rotation, scaling) on a handheld device, our literature research indicated that state-of-the-art interaction techniques usually use the multi-touch capabilities of the device in combination with complex multi-finger or -hand gestures. However, in a handheld mixed reality scenario where the user has usually only one hand available for interaction while the other one is holding the device, these approaches are not applicable and impede the intuitive usage as they require prior knowledge about the supported gestures. To overcome these limitations, we proposed three novel techniques for 3D interaction that employ the tracked device pose to highly reduce and thus simplify the user touch input.

For 3D object selection, we presented *DrillSample* that only requires one-finger tap gestures as input and splits the selection procedure into two steps. For object indication, Raycasting is employed that indicates the scene object(s) for later selection. In case of casting multiple objects, their full original 3D spatial context is preserved upon object indication. Thereby, the user is enabled to disambiguate and precisely select occluded objects or objects with high similarity in visual appearance. Finally, the desired object is selected within this refinement step by employing an one-finger tap gesture. The imprecise touch input of a finger that might yield ambiguous object indication is thereby compensated by the optional second refinement step. In comparison to state-of-the-art techniques, *DrillSample* provides precise selection of partly or fully occluded objects and the non-ambiguous identification of a desired object amongst visually similar ones by only requiring one-finger touch input. The conducted quantitative and qualitative evaluation revealed the strengths of *DrillSample* that outperformed the baseline techniques as it was found the best general purpose method for visible as well as partly and fully occluded objects, independent of their visual appearance.

For 3D object manipulation, the two novel methods *3DTouch* and *HOMER-S* were presented which both support translation, rotation and non-uniform scaling. *3DTouch* is based on multi-finger touch input and employs DOF-decomposition. Thereby, the integral 6DOF manipulation is split into the two tasks translation and rotation, enabling one finger to be sufficient to access all three DOFs of both tasks. Scaling is designed as another separate 3DOF task and requires a two-finger pinch gesture to allow for non-uniform transformations. *HOMER-S* decouples the manipulation process from any touch input and thus provides interaction beyond the (limited) screen dimensions. Therefore, it maps the estimated 6DOF device pose onto the object upon selection and employs real-world metaphors to enhance ease of use. *HOMER-S* applies DOF-integration for the 6DOF task translation and rotation and uses the 6DOF device pose to provide non-uniform scaling in a separate manipulation task. A comprehensive user study indicated

the strength of both techniques to intuitively translate and rotate objects. HOMER-S was found to be less accurate for 3D manipulation compared to 3DTouch but performed significantly faster for integral 6DOF manipulation tasks.

1.2.1 Open Topics

While investigating and developing the presented techniques, we have identified the following open topics in the context of 3D interaction.

- DrillSample was tested and evaluated in handheld mixed reality setups. However, the underlying algorithm can be applied to semi- as well as fully immersive environments. Thus, we plan to use and evaluate DrillSample in various mixed reality setups, using 6DOF input devices for object indication and selection in combination with stereoscopic viewing through shutter glasses or head mounted displays. Since the DrillSample visualization does not depend on display size but on the field of view of the user's output device, concepts such as the Image-Plane technique [62] can be employed to show the indicated objects in front of the user in space.
- We plan to further examine performance and usability of DrillSample for selecting objects in scenarios with various combinations of object density, size and distance. Therefore, we also consider to investigate using DrillSample with Cone-Casting to provide accurate selection of smaller objects at a larger distance.
- Our findings and the promising results of 3DTouch and HOMER-S motivate us to further evaluate the capabilities of both techniques. Therefore, we will investigate concepts to combine both techniques to enable context-aware manipulation to benefit from HOMER-S capabilities for rather coarse 3D manipulations and to exploit 3DTouch for fine-grained interactions.
- We plan to optimize the overall usability of HOMER-S to further exploit its potential. Therefore, we focus on improving the stability of the 6DOF device's pose during manipulation by applying filtering techniques to further reduce the intrinsic optical tracking jitter. This would yield an increased accuracy and might enhance the technique's potential to successfully perform fine manipulations as well. Given the direct mapping of the device's pose onto the selected object, rotations around the pitch-axis are limited. To solve for this issue, a non-direct mapping between the device's and object's orientation will be examined. Furthermore, we plan to provide more robust and view-independent pose tracking by incorporating natural feature tracking based on the surrounding scene geometry. Additionally, a temporal loss of the tracking pose might be compensated by fusing the inertial measurement data of the handheld device with the optical inside-out tracking data.

1.3 Creating Mixed Reality Environments

The third and last part of this thesis has focused on providing a framework to facilitate the development of compelling mixed reality environments. As this requires knowledge in all involved sub-domains, comprising tracking, interaction, scene authoring, 3D visualization and, optionally, network handling for distribution, the resulting entry threshold for application development is high. To minimize these initial hurdles and thereby, to leverage mixed reality technology for a broader everyday usage, an inexpensive novel toolkit ARTIFICE was presented that provides a powerful graphical interface to easy access and author the previously mentioned five modules.

ARTIFICE's framework design follows a modular software architecture and features loosely-coupled, extendable modules for tracking, interaction and distribution. Built upon a state-of-the-art game engine, the framework further provides high quality 3D rendering, physics support and an integrated build tool to deploy the project for various hardware platforms, including Windows and Android. To support a wide range of tracking input, we integrated and extended two middleware frameworks for workstation and mobile device support. Thereby, ARTIFICE is capable to integrate tracking input from planar bitmap patterns, 3D mice, rigid body optical tracking targets, Microsoft Kinect, Razer Hydra and mobile devices running Android and iOS. The frameworks interaction module provides well-defined interfaces to integrate custom methods and offers a number of built-in techniques, including the proposed methods from Part III. Finally, the developed distribution module supports the creation of collaborative and distributed mixed reality environments that can be visualized on a wide range of devices, such as smartphones, stereo projectors and head mounted displays.

We demonstrated the framework's capabilities of creating versatile mixed reality environments by presenting a number of examples of non-, semi- and fully-immersive setups. Finally, the framework was tested by more than 150 users who were technically versed but did not have in depth knowledge in mixed reality. Their results indicated that the framework is able to lower the initial hurdles of creating advanced applications and to develop embodied mixed reality experiences.

We hope that our contributions can support mixed reality developers in creating high quality, compelling virtual environments to further foster the pervasiveness of mixed reality applications in everyday scenarios.

1.3.1 Open Topics

As developing a software framework is a constant and on-going process, there are a number of open topics that are worthwhile to investigate in the future.

- We focus on improving mobile support and interaction. Therefore, we plan to assess, test and integrate different mobile middleware frameworks to provide mixed reality also on devices running iOS. Furthermore, we will examine concepts to enable distributed mixed reality across stationary and handheld devices. Here, we

1.3 Creating Mixed Reality Environments

aim at the flexible management of the employed 3D user interaction depending on the mixed reality setup and interaction device of each user.

- We aim on providing the novel framework as open-source project to developers and the research community.

PART VI

Appendix

Bibliography	209
List of Figures	223
List of Tables	227
A User Studies	229

Bibliography

- [1] Yehezkel Lamdan and Haim Wolfson. “Geometric Hashing: A general and efficient Model-Based Recognition Scheme”. In: *ICCV 88* (1088), pp. 238–249.
- [2] Rensis Likert. “A Technique for the Measurement of Attitudes”. In: *Archives of Psychology* 140 (132), pp. 1–55.
- [3] R.E. Kalman. “A new Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering* 82 (1960), pp. 35–45.
- [4] Merrill I. Skolnik. “Introduction to Radar Systems”. In: *Radar Handbook*. 1962, p. 2.
- [5] Wendell R Garner. *The Processing of Information and Structure*. L. Erlbaum Assoc., 1974.
- [6] M.E. Mündel. *Motion and Time Study: Improving Productivity*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1978.
- [7] S. Holm. “A simple sequentially rejective multiple test procedure”. In: *Scandinavian Journal of Statistics* 6.2 (1979), pp. 65–70.
- [8] Richard A. Bolt. “Put-that-there”. In: *ACM Voice and Gesture at the Graphics Interface* 14 (1980).
- [9] H.C. Longuet-Higgins. “A Computer Algorithm for Reconstructing a Scene from Two Projections”. In: *Nature* 293 (1981), pp. 133–135.
- [10] KS Arun, TS Huang, and SD Blostein. “Least-squares fitting of two 3-D point sets”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9.5 (1987), pp. 698–700.
- [11] Berthold K P Horn. “Closed-form solution of absolute orientation using unit quaternions”. In: *JOSA A* 4.4 (1987), pp. 629–642.
- [12] C. Harris and M. Stephens. “A combined corner and edge detector”. In: *Proceedings of the 4th Alvey Vision Conference*. 1988, pp. 147–151.
- [13] Mark R. Shortis and Clive S. Fraser. “A review of close range optical 3D measurement”. In: *Proceedings of 16th National Surveying Conference*. Barossa Valley, Australia, 1990.
- [14] K. Kanatani. “Computational Projective Geometry”. In: *CVGIP* 54.3 (1991), pp. 333–348.

BIBLIOGRAPHY

- [15] P. J. Besl and N. D. McKay. "A Method for Registration of 3-D Shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.
- [16] C Cruz-Neira, DJ Sandin, and TA DeFanti. "Surround-Screen Projection-Based Virtual Reality: the Design and Implementation of the CAVE". In: *20th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, New York, NY, USA, 1993, pp. 135–142. ISBN: 0897916018.
- [17] Oliver Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993. ISBN: 0-262-06158-9.
- [18] J. Liang and M Green. "JDCAD: a Highly Interactive 3D Modeling System". In: *Proceedings of Third International Conference on CAD and Computer Graphics*. 1994, pp. 217–222.
- [19] Paul Milgram, H. Takemura, A. Utsumi, and F. Kishino. "Augmented Reality: A class of displays on the reality-virtuality continuum". In: *Proceedings of Telem manipulator and Telepresence Technologies*. 1994, pp. 2351–34.
- [20] Greg Welch and Gary Bishop. *An Introduction to the Kalman Filter*. Tech. rep. Chapel Hill, USA: University of North Carolina, 1995, pp. 1–16.
- [21] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. "Aperture based Selection for Immersive Virtual Environments". In: *Proceedings of the 9th ACM Symposium on User Interface Software & Technology*. 1996, pp. 95–96. ISBN: 0897917987.
- [22] Ivan Poupyrev and Mark Billinghurst. "The Go-Go Interaction Technique: non-linear Mapping for direct Manipulation in VR". In: *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 1996, pp. 79–80.
- [23] J Rekimoto. "Tilting Operations for Small Screen Interfaces". In: *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 1996, pp. 167–168.
- [24] Doug A Bowman and Larry F Hodges. "An Evaluation of Techniques for Grabbing and Manipulating Objects in Immersive Virtual Environments Arm-Extension Ray-Casting". In: *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. 1997, pp. 35–38.
- [25] D.W. Eggert, A. Lorusso, and R.B. Fisher. "Estimating 3-D Rigid Body Transformations: a Comparison of Four Major Algorithms". In: *Machine Vision and Applications* 9.5-6 (Mar. 1997), pp. 272–290. ISSN: 0932-8092.
- [26] Richard Hartley and Peter Sturm. "Triangulation". In: *Computer Vision and Image Understanding* 68.2 (Nov. 1997), pp. 146–157. ISSN: 10773142.
- [27] Janne Heikkila and Olli Silven. "A four-step camera calibration procedure with implicit image correction". In: *IEEE Conference on Computer Vision and Pattern Recognition*. San Juan, 1997, pp. 1106–1112.

-
- [28] J S Pierce, A Forsberg, M J Conway, S Hong, R Zeleznik, and M Mine. “Image Plane Interaction Techniques in 3D Immersive Environments”. In: *Proceedings of the Symposium on Interactive 3D Graphics (I3D ‘97)*. 1997, pp. 39–43.
- [29] Hans-Jürg Fuchser. “Determining Convergences by photogrammetric Means”. In: *TUNNEL* 17.7 (1998), pp. 38–42.
- [30] P. Meer, R. Lenz, and S. Ramakrishna. “Efficient invariant representations”. In: *IJCV* 26.2 (1998), pp. 137–152.
- [31] Ivan Poupyrev, T Ichikawa, S Weghorst, and Mark Billinghurst. “Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques”. In: *Computer Graphics Forum (Wiley Online Library)* 17 (1998), pp. 41–52.
- [32] Doug a. Bowman and Larry F. Hodges. “Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments”. In: *Journal of Visual Languages & Computing* 10.1 (Feb. 1999), pp. 37–53. ISSN: 1045926X.
- [33] Klaus Dorfmueller. “Robust Tracking for Augmented Reality using Retroreflective Markers”. In: *Computers and Graphics* 23.6 (1999), pp. 795–800.
- [34] Klaus Finkenzyler. *RFID handbook: Radio-frequency identification fundamentals and applications*. New York, USA: John Wiley, 1999. ISBN: ISBN 0471988510.
- [35] Hirokazu Kato and Mark Billinghurst. “Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System”. In: *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR)*. IEEE, 1999, pp. 85–94.
- [36] David G. Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 1999, pp. 1150–1157.
- [37] Ivan Poupyrev and Tadao Ichikawa. “Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques”. In: *Journal of Visual Languages & Computing* 10.1 (Feb. 1999), pp. 19–35. ISSN: 1045926X.
- [38] P. Sturm and S. Maybank. “On Plane-based Camera Calibration: A general Algorithm, Singularities, Applications.” In: *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Fort Collins, Colorado, USA: IEEE Computer Society Press, 1999, pp. 432–437.
- [39] Bill Triggs, Philip F. McLauchlan, Richard Hartley, and Andrew Fitzgibbon. “Bundle adjustment - A modern synthesis”. In: *Vision Algorithms: Theory and Practise*. Ed. by W. Triggs, A. Zisserman, and R. Szeliski. Vol. 34099. Springer, 2000, pp. 298–372.
- [40] Zhengyou Zhang. “A Flexible new Technique for Camera Calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334.

BIBLIOGRAPHY

- [41] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. “The MagicBook: a Transitional AR Interface”. In: *Computers & Graphics* 25.5 (2001), pp. 745–753.
- [42] Carolina Cruz-Neira, Allen Bierbaum, Patrick Hartling, Christopher Just, and Kevin Meinert. “VR Juggler – An Open Source Platform for Virtual Reality Applications”. In: *Proceedings of IEEE Virtual Reality*. Reno, Nevada, USA: IEEE, 2001, pp. 89–96.
- [43] Gerd Hesina. “Distributed Collaborative Augmented Reality”. PhD thesis. Vienna University of Technology, 2001.
- [44] Jeffrey Hightower and Gaetano Borriello. “Location Systems for Ubiquitous Computing”. In: *IEEE Computer* 34(8).August (2001), pp. 57–66.
- [45] J. Lasenby and A. Stevenson. “Using Geometric Algebra for Optical Motion Capture”. In: *Geometric Algebra: A Geometric Approach to Computer Vision, Neural and Quantum Computing, Robotics and Engineering pages*. 2001, pp. 147–169.
- [46] Gerhard Reitmayr and Dieter Schmalstieg. “An Open Software Architecture for Virtual Reality Interaction”. In: *Proceedings of ACM Symposium on Virtual Reality Software & Technology (VRST)*. Banff, Canada, 2001, pp. 47–54.
- [47] Russel Taylor, Thomas C Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, and Aron T Helser. “VRPN: A Device-Independent, Network-Transparent VR Peripheral System”. In: *Proceedings of ACM Symposium on Virtual Reality Software & Technology (VRST)*. Banff, Canada, 2001.
- [48] Klaus Dorfmueller-Ulhaas. “Optical Tracking: From User Motion To 3D Interaction”. PhD Thesis. Vienna University of Technology, 2002.
- [49] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Prentice Hall, New Jersey, USA, 2002, 587ff. ISBN: 0-201-18075-8.
- [50] Mike Hazas and Andy Ward. “A novel broadband ultrasonic location system”. In: *Ubiquitous Computing* 2498.September (2002), pp. 264–280.
- [51] D. Scharstein and R. Szeliski. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”. In: *International Journal of Computer Vision* 47.1/2/3 (2002), pp. 7–42.
- [52] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, Miguel Encarnacao, Michael Gervautz, and Werner Purgathofer. “The Studierstube augmented reality project”. In: *Presence - Teleoperators and Virtual Environments* 11.1 (2002), pp. 33–54.
- [53] Grigore C Burdea and Philippe Coiffet. *Virtual Reality Technology*. 2nd. Wiley-IEEE, 2003. ISBN: 0471360899.
- [54] Robert van Liere and Jurriaan D. Mulder. “Optical tracking using projective invariant marker pattern properties”. In: *Proceedings of IEEE Virtual Reality*. IEEE Comput. Soc, 2003, pp. 191–198. ISBN: 0-7695-1882-6.

-
- [55] Ralitza Gueorguieva and John H. Krysta. “Move over anova: Progress in analyzing repeated-measures data and its reflection in papers published in the archives of general psychiatry”. In: *Archives of General Psychiatry* 61.3 (2004), pp. 310–317.
- [56] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. ISBN: 0521540518.
- [57] Robert van Liere and Arjen van Rhijn. “An experimental comparison of three optical trackers for model based pose determination in virtual reality”. In: *Proceedings of 10th Eurographics Conference on Virtual Environments (EGVE’04)*. Aire-la-Ville, Switzerland, 2004, pp. 25–34.
- [58] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691.
- [59] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. “DART: a Toolkit for Rapid Design Exploration of Augmented Reality Experiences”. In: *Proceedings of the 17th ACM Symposium on User Interface Software and Technology*. ACM Publications, 2004, pp. 197–206.
- [60] Gerard Medioni and Sing Bing Kang. *Emerging Topics in Computer Vision*. Ed. by Prentice Hall Professional Technical Reference. Upper Saddle River, NJ, USA, 2004. Chap. 2. ISBN: 0131013661.
- [61] A. Stelzer, K. Pourvoyeur, and A. Fischer. “Concept and application of LPM—A novel 3-D local position measurement system”. In: *IEEE Transactions on Microwave Theory and Techniques* 42 (2004), pp. 2664–2669.
- [62] Doug Bowman, Ernst Kruijff, Joseph J LaViola Jr., and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2005.
- [63] M. Fiala. “ARTag, a fiducial marker system using digital techniques”. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Washington, DC, USA: IEEE, 2005, pp. 590–596.
- [64] Raphael Grasset, Julian Looser, and Mark Billinghurst. “A Step Towards a Multimodal AR Interface : A New Handheld Device for 3D Interaction”. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2005, pp. 206–207.
- [65] Anders Henrysson, Mark Billinghurst, and Mark Ollila. “Augmented Reality on Mobile Phones: Experiments and Applications”. In: *The Annual SIGRAD Conference Special Theme – Mobile Graphics*. Linköping University Electronic Press, Linköping University, 2005, pp. 35–40.
- [66] Anders Henrysson, Mark Billinghurst, and Mark Ollila. “Virtual object manipulation using a mobile phone”. In: *Proceedings of the 2005 International Conference on Augmented Teleexistence (ICAT)*. ACM, 2005, p. 164. ISBN: 0473106574.

BIBLIOGRAPHY

- [67] Bodhi P. Nissanka. “The cricket indoor location system”. PhD Thesis. Massachusetts Institute of Technology, USA, 2005.
- [68] Jesús Rodríguez. *Vision 2030*. Tech. rep. Maastricht, Netherlands: European Construction Technology Platform (ECTP), www.ectp.org, 2005.
- [69] Tomas Svoboda, Daniel Martinec, and Tomas Pajdla. “A convenient multi-camera self-calibration for virtual environments”. In: *Presence: Teleoperators & Virtual Environments* 14.4 (2005), pp. 407–422.
- [70] Bill Glover and Himanshu Bhatt. *RFID Essentials*. O’Reilly Media, 2006. ISBN: 0-596-00944-5.
- [71] Sinem Guven, Steven Feiner, and Ohan Oda. “Mobile Augmented Reality Interaction Techniques for Authoring Situated Media On-Site”. In: *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, Oct. 2006, pp. 235–236. ISBN: 1-4244-0650-1.
- [72] M.S. Hancock, S. Carpendale, F.D. Vernier, and D. Wigdor. “Rotation and Translation Mechanisms for Tabletop Interaction”. In: *International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP ’06)*. IEEE, 2006, pp. 79–88. ISBN: 0-7695-2494-X.
- [73] Bing Jiang, Kenneth P. Fishkin, Sumit Roy, and Matthai Philipose. “Unobtrusive long-range detection of passive RFID tag motion”. In: *IEEE Transactions on Instrumentation and Measurement* 55.1 (2006), pp. 187–196.
- [74] Edward Rosten and Tom Drummond. “Machine Learning for High Speed Corner Detection”. In: *9th European Conference on Computer Vision*. 2006, pp. 430–443.
- [75] Pedro Santos and Andre Stork. “Ptrack: introducing a novel iterative geometric pose estimation for a marker-based single camera tracking system”. In: *Proceedings of IEEE Virtual Reality*. USA, 2006, pp. 149–156. ISBN: 1424402247.
- [76] Ferdi Alexander Smit, Arjen van Rhijn, and Robert van Liere. “GraphTracker: A Topology Projection Invariant Optical Tracker”. In: *Proceedings of the 12th Eurographics Conference on Virtual Environments*. 2006, pp. 63–70.
- [77] Klaus Chmelina. “Laserscanning in Underground Construction: State and Future of a Multi-Purpose Surveying Technology”. In: *Austria - China - International Symposium on Challenging Tunnel Construction*. Vienna, Austria: Institut fuer Interdisziplinäeres Bauprozessmanagement, 2007.
- [78] Raphael Grasset, Andreas Dünser, and Mark Billinghurst. “Human-Centered Development of an AR Handheld Display”. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. Nara, Japan: IEEE, 2007, pp. 177–180. ISBN: 9781424417506.
- [79] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. “Shallow-Depth 3D Interaction : Design and Evaluation of One-, Two- and Three-Touch Techniques”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2007, pp. 1147–1156. ISBN: 9781595935939.

-
- [80] Taehee Lee and Tobias Hollerer. “Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking”. In: *Proceedings of the International Symposium on Wearable Computers*. IEEE, Oct. 2007, pp. 1–8. ISBN: 978-1-4244-1452-9.
- [81] Manuel Loaiza, Alberto Raposo, and Marcelo Gattass. “A novel optical tracking algorithm for point-based projective invariant marker patterns”. In: *Advances in Visual Computing* 4841 (2007), pp. 160–169.
- [82] Fangfang Lu and Richard Hartley. “A fast optimal algorithm for L 2 triangulation”. In: *Computer Vision—ACCV 2007*. Springer, 2007, pp. 279–288.
- [83] Karen McMenemy and Stuart Ferguson. *A Hitchhiker’s Guide to Virtual Reality*. 1st. A.K. Peters, Ltd, Wellesley, MA, USA, 2007. ISBN: 13:978-1-56881-303-5.
- [84] Thomas Pintaric and Hannes Kaufmann. “Affordable Infrared-Optical Pose-Tracking for Virtual and Augmented Reality”. In: *Proceedings of Trends and Issues in Tracking for Virtual Environments Workshop, IEEE VR 2007*. 2007, pp. 44–51.
- [85] Arjen van Rhijn. “Configurable Input Devices for 3D Interaction using Optical Tracking”. PhD Thesis. Technische Universiteit Eindhoven, Netherlands, 2007. ISBN: 9789038608341.
- [86] F.a. Smit, A. van Rhijn, and R. van Liere. “Graphtracker: A Topology Projection Invariant Optical Tracker”. In: *Computers & Graphics* 31.1 (Jan. 2007), pp. 26–38. ISSN: 00978493.
- [87] Daniel Wagner and Dieter Schmalstieg. “ARToolKitPlus for Pose Tracking on Mobile Devices”. In: *Proceedings of 12th Computer Vision Winter Workshop (CVWW’07)*. Ed. by Michael Grabner and Helmut Grabner. 2007, pp. 139–146.
- [88] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. “SURF: Speeded Up Robust Features”. In: *Computer Vision and Image Understanding (CVIU)* 110.3 (2008), pp. 346–359.
- [89] Klaus Chmelina. “Tunnel Laser Scanning, Current Systems, Applications and Research Activities”. In: *Proceedings of the ITA - AITES World Tunnel Congress*. Agra, India, 2008, pp. 86–92.
- [90] Klaus Chmelina and Klaus Rabensteiner. “Laser Scanning Technology in Underground Construction”. In: *Proceedings of the Jubilee International Scientific and Technical Conference on Tunnel and Metro Constructions*. Sofia, Bulgaria, 2008.
- [91] Barry Kavanagh. *Surveying principles and applications*. 8th. Prentice Hall Inc., 2008. ISBN: 978-0132365123.
- [92] Roland Kuck, Jürgen Wind, Kai Riege, and Manfred Bogen. “Improving the AVANGO VR/AR Framework - Lessons Learned”. In: *5th Workshop of the GI-VR/AR Group*. Magdeburg, Germany: VDTC, 2008.
- [93] Leica Geosystems. *The Leica Absolute Interferometer: A New Approach to Laser Tracker Absolute Distance Meters*. Tech. rep. Unterentfelden, Switzerland, 2008, p. 11.

BIBLIOGRAPHY

- [94] Annette Mossel, Thomas Pintaric, and Hannes Kaufmann. *Analyse der Machbarkeit und des Innovationspotentials der Anwendung der Technologie des Optical Real-Time Trackings für Aufgaben der Tunnelvortriebsvermessung*. Tech. rep. Austria: Institute of Software Technology and Interactive Systems, Vienna University of Technology, 2008.
- [95] Thomas Pintaric and Hannes Kaufmann. “A Rigid-Body Target Design Methodology for Optical Pose-Tracking Systems”. In: *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology (VRST)*. ACM, 2008, pp. 73–76. ISBN: 978-1-59593-951-7.
- [96] WolfWings. *Barrel Dirtortion*. [Online Image]. 2008. URL: https://en.wikipedia.org/wiki/File:Barrel%5C_distortion.svg (visited on 03/01/2014).
- [97] WolfWings. *Pincushion Distortion*. [Online Image]. 2008. URL: https://en.wikipedia.org/wiki/File:Pincushion%5C_distortion.svg (visited on 03/02/2014).
- [98] Alan B. Craig, William R. Sherman, and Jeffrey D. Will. *Developing Virtual Reality Applications: Foundations of Effective Design*. Morgan Kaufmann Publishers Inc, 2009.
- [99] Thao Dang, Christian Hoffmann, and Christoph Stiller. “Continuous stereo self-calibration by camera parameter tracking.” In: *IEEE Transactions on Image Processing* 18.7 (July 2009), pp. 1536–50. ISSN: 1057-7149.
- [100] Barry Kavanagh. *Surveying with Construction Applications*. 7th. Prentice Hall Inc., 2009. ISBN: 978-0135000519.
- [101] GA Lee, Ungyeon Yang, Y Kim, D Jo, and KH Kim. “Freeze-Set-Go Interaction Method for Handheld Mobile Augmented Reality Environments”. In: *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST)*. ACM, 2009, pp. 143–146. ISBN: 9781605588698.
- [102] Ran Liu, Hua Zhang, Manlu Liu, Xianfeng Xia, and Tianlian Hu. “Stereo Cameras Self-Calibration Based on SIFT”. In: *International Conference on Measuring Technology and Mechatronics Automation*. Ieee, 2009, pp. 352–355. ISBN: 978-0-7695-3583-8.
- [103] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. “A Screen-Space Formulation for 2D and 3D Direct Manipulation”. In: *Proceedings of the Symposium on User interface software and Technology (UIST)*. ACM, 2009, p. 69. ISBN: 9781605587455.
- [104] Manuel Veit. “Influence of Degrees of Freedom’s Manipulation on Performances During Orientation Tasks in Virtual Reality Environments”. In: *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST)*. Vol. 1. 212. 2009, pp. 51–58. ISBN: 9781605588698.

-
- [105] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. “BRIEF: Binary Robust Independent Elementary Features”. In: *11th European Conference on Computer Vision (ECCV)*. Heraklion, Greece: LNCS Springer, 2010.
- [106] Jürgen Janssen and Wilfried Laatz. “Statistische Datenanalyse mit SPSS”. In: *Eine anwendungsorientierte Einführung in das Basissystem und das Modul exakte Tests 7* (2010).
- [107] Sven Kratz and Michael Rohs. “Extending the Virtual Trackball Metaphor to Rear Touch Input”. In: *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI)*. Ieee, Mar. 2010, pp. 111–114. ISBN: 978-1-4244-6846-1.
- [108] Anthony Martinet, Gery Casiez, and Laurent Grisoni. “The design and evaluation of 3D positioning techniques for multi-touch displays”. In: *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, Mar. 2010, pp. 115–118. ISBN: 978-1-4244-6846-1.
- [109] Takehiro Niikura, Yuki Hirobe, Alvaro Cassinelli, Yoshihiro Watanabe, Takashi Komuro, and Masatoshi Ishikawa. *In-Air Typing Interface for Mobile Devices with Vibration Feedback*. ACM, 2010, pp. 1–15. ISBN: 9781450303927.
- [110] Edward Rosten, Reid Porter, and Tom Drummond. “Faster and better: a machine learning approach to corner detection”. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 32 (2010), pp. 105–119.
- [111] Dieter Schmalstieg, Tobias Langlotz, and Mark Billinghurst. *Augmented Reality 2 . 0*. Ed. by Sabine Coquillart, Guido Brunnett, and Greg Welch. Dagstuhl S. Springer, 2010.
- [112] Amal Benzina, M Toennis, Gudrun Klinker, and Mohamed Ashry. “Phone-based Motion Control in VR: Analysis of Degrees of Freedom”. In: *Proceedings of the 2011 Conference on Human Factors in Computing Systems*. 2011, pp. 1519–1524. ISBN: 9781450302685.
- [113] A. Cohé, D Fabrice, and Martin Hachet. “tBox : A 3D Transformation Widget designed for Touch-Screens”. In: *Proceedings of the 2011 annual conference on Human Factors in Computing Systems*. 2011, pp. 3005–3008. ISBN: 9781450302678.
- [114] Wolfgang Hürst and Casper Van Wezel. *Multimodal Interaction Concepts for Mobile Augmented Reality Applications*. Springer, 2011, pp. 157–167.
- [115] Regis Kopper, Felipe Bacim, and Doug a. Bowman. “Rapid and accurate 3D Selection by Progressive Refinement”. In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, Mar. 2011, pp. 67–74. ISBN: 978-1-4577-0063-7.
- [116] Jens Puwein and Remo Ziegler. “Robust multi-view camera calibration for wide-baseline camera networks”. In: *IEEE Workshop on Applications of Computer Vision (WACV)*. 2011, pp. 321–328. ISBN: 9781424494972.
- [117] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradsk. “ORB: An efficient alternative to SIFT or SURF”. In: *International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2564–2571.

BIBLIOGRAPHY

- [118] R. Ortiz Alahi and P. Vandergheynst. “FREAK: Fast Retina Keypoint”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.
- [119] Jeffrey Cashion, Chadwick Wingrave, and Joseph J LaViola. “Dense and Dynamic 3D Selection for Game-based Virtual Environments”. In: *IEEE Virtual Reality*. Vol. 18. 4. Costa Mesa, USA: IEEE, Apr. 2012, pp. 634–42.
- [120] Klaus Chmelina, Josef Jansa, Gerd Hesina, and Christoph Traxler. “A 3-D Laser-scanning System and Scan Data Processing Method for the Monitoring of Tunnel Deformations”. In: *Journal of Applied Geodesy* 6.3-4 (Jan. 2012), pp. 177–185. ISSN: 1862-9016.
- [121] Florian Daiber, Lianchao Li, and Antonio Krüger. “Designing Gestures for Mobile 3D Gaming”. In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 2012, p. 3. ISBN: 9781450318150.
- [122] Zeller. David. “Physics driven 3D Dynamic Geometry Software for Elementary Education”. Master Thesis. Vienna University of Technology, 2012, p. 96.
- [123] Andy Field, Jeremy Miles, and Zoe Field. *Discovering Statistics Using R*. SAGE Publications, 2012. ISBN: 9781446200469.
- [124] Wolfgang Hürst and Casper Wezel. “Gesture-based Interaction via Finger Tracking for Mobile Augmented Reality”. In: *Multimedia Tools and Applications* 62.1 (Jan. 2012), pp. 233–258. ISSN: 1380-7501.
- [125] Hanno Jaspers, Boris Schauerte, and GA Fink. “Sift-based Camera Localization using Reference Objects for Application in Multi-camera Environments and Robotics.” In: *ICPRAM (2)*. 2012, pp. 330–336.
- [126] Anthony Martinet, Géry Casiez, and Laurent Grisoni. “Integrality and Separability of Multitouch Interaction Techniques in 3D Manipulation Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.3 (Mar. 2012), pp. 369–80. ISSN: 1941-0506.
- [127] Microsoft. *Kinect full body interaction*. 2012. URL: <http://www.xbox.com/en-US/kinect>.
- [128] Evan Suma, David Krum, Belinda Lange, Skip Rizzo, and Marc Bolas. “FAAST: The Flexible Action and Articulated Skeleton Toolkit.” In: *Proceeding of IEEE Virtual Reality*. Costa Mesa, USA: IEEE, 2012, pp. 247–248.
- [129] Accurex. *AICON DPA-Pro System*. [Online]. 2013. URL: <http://www.accurexmeasure.com/dpapro.htm> (visited on 29/11/2013).
- [130] AIA. *GigE Vision*. [Online]. 2013. URL: <http://www.visiononline.org> (visited on 08/01/2013).
- [131] Arduino. *Arduino IDE*. [Online]. 2013. URL: <http://www.arduino.cc/> (visited on 12/05/2013).
- [132] ART. *Advanced Real Time Tracking*. [Online]. 2013. URL: <http://www.art-tracking.de> (visited on 01/08/2013).

-
- [133] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab*. [Software]. 2013. URL: http://www.vision.caltech.edu/bouguetj/calib%5C_doc (visited on 09/01/2013).
- [134] Michael Bressler. “A Virtual Reality Training Tool for Upper Limb Prostheses”. Master Thesis. Vienna University of Technology, 2013, p. 115.
- [135] Geodata Group. *Gripper Camera - System Description and Data Sheet*. Tech. rep. Austria, 2013.
- [136] Faheem Ijaz, Hee Kwon Yang, Arbab Waheed Ahmad, and Chankil Lee. “Indoor Positioning: A Review of Indoor Ultrasonic Positioning systems”. In: *Proceedings of 15th International Conference on Advanced Communication Technology (ICACT)*. 2013, pp. 1146–1150.
- [137] MathWorks. *MATLAB ImageProcessingToolbox*. [Software]. 2013. URL: <http://www.mathworks.com/help/toolbox/images/> (visited on 01/12/2013).
- [138] NaturalPoint Inc. *OptiTrack*. [Online]. 2013. URL: <http://www.naturalpoint.com/optitrack/> (visited on 01/12/2013).
- [139] Andrei Ninu. “Prosthesis Embodiment : Sensory-Motor Integration of Prosthetic Devices into the Amputee’s Body Image”. PhD Thesis. Vienna University of Technology, 2013.
- [140] OpenCV. *Open Computer Vision Library*. [Software]. 2013. URL: <http://opencv.org/> (visited on 12/01/2013).
- [141] Thomas Pintaric and Hannes Kaufmann. *iotracker*. [Online]. 2013. URL: <http://www.iotracker.com> (visited on 01/12/2013).
- [142] Razer Inc. *Hydra*. [Online] <http://www.razerzone.com/gaming-controllers/razer-hydra>. 2013. URL: <http://www.razerzone.com/gaming-controllers/razer-hydra> (visited on 2013).
- [143] Can Telkenaroglu and Tolga Capin. “Dual-Finger 3D Interaction Techniques for Mobile Devices”. In: *Personal and Ubiquitous Computing 17.7* (Sept. 2013), pp. 1551–1572. ISSN: 1617-4909.
- [144] Khrystyna Vasylevska, Hannes Kaufmann, Mark Bolas, and Evan A. Suma. “Flexible Spaces : Dynamic Layout Generation for Infinite Walking in Virtual Environments”. In: *IEEE Symposium on 3D User Interfaces (3DUI)*. Orlando: IEEE, 2013, pp. 1–4.
- [145] Vicon. *Motion Capture*. [Online]. 2013. URL: <http://www.vicon.com/> (visited on 12/01/2013).
- [146] WorldViz. *PPT E Motion Tracking*. [Online]. 2013. URL: <http://www.worldviz.com/products/ppt/> (visited on 12/01/2013).
- [147] 3D Connexion. *SpaceNavigator*. [Online]. 2014. URL: <http://www.3dconnexion.de/> (visited on 05/06/2014).

BIBLIOGRAPHY

- [148] DirectIndustry. *Multi-sided 3D touch probe (MSP) for optical tracker, Leica T-Probe*. [Online Image]. 2014. URL: <http://www.directindustry.com/prod/hexagon-metrology/multi-sided-3d-touch-probes-msp-optical-trackers-5623-1132257.html> (visited on 12/04/2014).
- [149] Fraunhofer IGD. *InstantReality*. [Software]. 2014. URL: <http://www.instantreality.org> (visited on 10/05/2014).
- [150] ImInVR. *MiddleVR*. [Software]. 2014. URL: <http://www.imin-vr.com/middlevr/> (visited on 10/05/2014).
- [151] InSense. *IS-1200 System*. 2014. URL: <http://www.intersense.com/pages/21/13> (visited on 20/08/2014).
- [152] MOB Labs. *BuildAR*. [Software]. 2014. URL: <https://buildar.com> (visited on 20/05/2014).
- [153] Thorlabs. *Motorized Fast-Change Filter Wheel*. 2014. URL: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup%5C_id=2945 (visited on 07/20/2014).
- [154] K. Vasylevska and H. Kaufmann. "Influence of Vertical Navigation Metaphors on Presence". In: *Challenging Presence - Proceedings of 15th International Conference on Presence (ISPR 2014)*. Vienna, Austria, 2014, pp. 205–212.
- [155] ZigBeeAlliance. *ZigBee*. 2014. URL: <http://zigbee.org/> (visited on 20/08/2014).
- [156] Google. *Indoor Maps*. [Online]. URL: <https://www.google.com/intl/en/maps/about/explore/mobile/> (visited on 01/12/2013).
- [157] IndooRs. *Location Tracking*. [Online]. URL: <http://indoo.rs/> (visited on 12/01/2013).
- [158] Leica Geosystems. *Absolute Tracker AT901*. [Online]. URL: http://www.leica-geosystems.com/en/Leica-Absolute-Tracker-AT901%5C_69047.htm (visited on 02/12/2013).
- [159] Leica Geosystems. *T-Probe*. [Online]. URL: <http://www.leica-geosystems.com> (visited on 02/12/2013).
- [160] *OpenNI*. [Software] (Version 1.3.2.3). URL: <http://openni.org> (visited on 11/01/2011).
- [161] *OpenVideo*. [Software] (Version 1.0.0). URL: <http://rpm.icg.tugraz.at/> (visited on 01/08/2011).
- [162] PrimeSense. *NITE*. [Software] (Version 1.4.1.2). URL: <http://www.primesense.com/> (visited on 11/01/2011).
- [163] Qualcomm Inc. *Vuforia SDK*. [Software] (Version 2.8). URL: <https://developer.vuforia.com/resources/sdk/android/> (visited on 05/12/2013).
- [164] SensionLab. *Indoor Positioning and Navigation*. [Online]. URL: <http://www.sensionlab.com> (visited on 12/01/2013).

BIBLIOGRAPHY

- [165] Evan A. Suma, Belinda Lange, Skip Rizzo, David Krum, and Mark Bolas. *Flexible Action and Articulated Skeleton Toolkit (FAAST)*. [Software] (Version 0.08). URL: <http://projects.ict.usc.edu/mxr/faast/> (visited on 01/11/2011).
- [166] Ubisense. *Real-Time Localization Systems*. [Online]. URL: <http://www.ubisense.net> (visited on 12/01/2013).
- [167] Unity Technologies. *Unity3D*. [Software] (Version 4.3.4). URL: <http://www.unity3d.com/> (visited on 01/01/2014).
- [168] Virtools. *Virtools Dev User Guide*. [Online]. URL: <http://www.virttools.com> (visited on 02/01/2014).

List of Figures

I Introduction

1.1	The Milgram continuum describing the variations of mixed reality.	3
1.2	Components of a mixed reality system.	4
2.1	Investigated concepts, their relationship and the presented contribution. . .	5

II Wide-Area Optical Tracking

1.1	Tracking approaches, with the field of contribution marked bold.	15
2.1	The optical tracking pipeline.	22
2.2	Types of optical markers.	24
2.3	Taxonomy of model fitting depending on domain and property.	24
2.4	After perspective projection of the four points, the projective invariant properties of the cross ratio are expressed by $\lambda(A, B, C, D) \hat{=} \lambda(A', B', C', D')$. The points' collinearity is preserved as well, as $l \hat{=} l'$	26
2.5	An example of a passive 3D rigid body target.	28
2.6	Taxonomy of pose estimation.	29
2.7	The pinhole camera geometry with camera center C coincides with the coordinate system's origin. The image plane is placed with distance f in front of C	31
2.8	The principal point offset.	33
2.9	Two common types of radial distortion.	34
2.10	The Euclidean transformation between the world and the camera coordinate system.	35
2.11	The epipolar geometry.	37
2.12	The four possible solutions for P' , as combinations of rotations and translations.	39
2.13	A calibration taxonomy by dimension of the applied apparatus.	40
2.14	Reference targets for intrinsic and extrinsic camera calibration.	41
3.1	Tracking of a smartphone using Google Indoor Maps [156].	44

List of Figures

3.2	A simple four sensor Ubisense system [166].	45
3.3	Multiple target tracking using iotracker with 4 cameras, [84].	46
3.4	A tracking setup using the Prime41 system, [138].	46
3.5	The AICON DPA-Pro System, [129].	47
3.6	Leica Absolute Tracker AT901 with T-Probe, [148].	48
4.1	Key properties of the proposed optical tracking system.	51
4.2	Blobs at 50m distance with minimal/maximal focal length of $f = 12 / 36mm$	53
4.3	Overview over the system's workflow.	54
4.4	Coverage of stereo cameras	55
4.5	The 2D model design features projective invariant properties.	55
4.6	LED is coated with a translucent diffuse plastic sphere.	56
4.7	Intrinsic camera calibration with a retro-reflective pattern.	57
4.8	Trained background (left) and manual masking (right), [141].	58
4.9	Extrinsic calibration pipeline.	60
4.10	Resulting camera coordinate system for tracking.	61
4.11	Wavelengths of various light sources.	61
4.12	Radio module for target communication for luminance-based filtering.	62
4.13	Using a motorized filter wheel for wavelength-based filtering.	63
4.14	Pipeline to detect target features using hardware-based filtering.	64
4.15	Pipeline to obtain the target's model.	65
4.16	Pipeline for model identification.	66
4.17	Tracking pipeline.	67
4.18	The cabling of the hardware prototype.	69
4.19	Software architecture and modules.	71
4.20	User interface of semi-autonomous Model Trainer.	71
4.21	Examples of incorrect model recognition during training.	72
4.22	User interface of Controller to analyze data during calibration and tracking.	73
5.1	Wide area user tracking in a mixed reality setup.	77
5.2	Wide area user tracking in a mixed reality setup.	78
5.3	Target design for head tracking.	78
5.4	Target prototype attached on a HMD.	79
5.5	Corresponding blob traces used for extrinsic calibration.	80
5.6	Mean of relative accuracy $x_{RMS}(P)$ over all three calibrations.	81
5.7	3D position tracking from 5 – 30m.	82
5.8	Tracking situation in an underground environment.	83
5.9	Multiple unique target constellations.	83
5.10	3D position estimation of visible or invisible static and moving target's tips.	84
5.11	Developed target prototype.	84
5.12	Details of the developed target prototype.	85
5.13	Robust and dampness proof encasement of cameras and base station.	85
5.14	Test environment in a metro underground station.	86
5.15	Calibration with $d_{base} \approx 6m$	87

5.16 Target movement during accuracy and stability measurements.	88
5.17 $ \hat{\epsilon}_{bar} $ for all d_{base} and d_{track}	89
5.18 3D position tracking of a moving target through the entire volume.	91
5.19 Examples of modern underground machinery.	91
5.20 Details of the test environment.	92
5.21 Comparison of blob quality at 110m with an inter LED distance of 34cm. . .	93
5.22 A single optical target comprising the encased IR-LED attached to a reflective geodesic foiled target.	94
5.23 The IR-LED line target prototype for machine tracking.	94
5.24 Kinematic tracking of the horizontal target from 20 – 110m with $d_{base} \approx 3m$. .	98
5.25 The vertical target is partly occluded by an interfering light but can still be successfully identified, as indicated by the yellow crosses.	99
5.26 Both targets' models are fully identified and tracked despite heavy interfering light.	100
5.27 Both targets' models are fully identified and tracked during fog tests.	100

III User Interfaces for 3D Interaction

1.1 Interaction categories, with the fields of contribution marked bold.	109
2.1 An excerpt of 3D interaction devices.	113
2.2 A mobile phone acting as a window into the virtual world.	116
2.3 Taxonomy for egocentric object interaction in handheld mixed reality.	116
2.4 Taxonomy of immersive selection techniques classified by metaphor.	117
2.5 The Expand refinement view, courtesy of [119].	121
3.1 The two-step DrillSample technique.	125
3.2 DrillSample's two-step selection process.	127
3.3 State diagram for DrillSample selection.	129
3.4 Ray-Casting adapted to use it in a handheld mixed reality.	129
3.5 Sphere approximation of clones' size to calculate the optimal ray length. . .	132
3.6 User study procedure.	135
3.7 The three test scenarios of the performance user study.	138
3.8 Mean completion time per task and on average.	139
3.9 Mean selection steps per task and on average.	140
3.10 Users' average rating of Q3, Q4 and Q5.	141
3.11 Users' rating of Q6.	142
4.1 Touchless full 6DOF object manipulation using HOMER-S.	147
4.2 Examples of translations using 3DTouch.	150
4.3 Examples of rotations using 3DTouch.	150
4.4 6DOF translation and rotation using HOMER-S.	152
4.5 Floating GUIs of both techniques upon selection.	154

List of Figures

4.6	Supporting visualization depending on manipulation task and current accessible interaction axes.	154
4.7	The three test scenarios of the performance user study.	159
4.8	Mean completion time and mean number of interaction steps.	161
4.9	Mean completion time per task.	161
4.10	Mean number of interaction steps per task.	162
4.11	Users' average rating of Q3 & Q4.	162
4.12	Users' preferences given Q7.	163

IV Creating Mixed Reality Environments

1.1	Key components of mixed reality, with the contributions marked bold. . . .	171
2.1	Mixed Reality system architecture.	173
3.1	ARTiFICe framework components and data flow.	177
3.2	OpenTracker nodes with new ones marked in blue.	180
3.3	ARTiFICe's processing pipeline of depth data for full-body motion tracking.	182
3.4	Detailed framework components.	183
3.5	Tracking class hierarchy.	184
3.6	Interaction class hierarchy.	185
3.7	Distribution class hierarchy.	186
4.1	Two examples of desktop mixed reality setups.	190
4.2	Multi-user collaborative and distributed handheld mixed reality.	191
4.3	A distributed multi-user non & semi-immersive mixed reality setup.	192
4.4	A distributed multi-user non & semi-immersive mixed reality setup.	193

V Conclusion

1.1	Investigated concepts, their relationship and the presented contribution. . .	199
-----	---	-----

List of Tables

II Wide-Area Optical Tracking

2.1	Projective invariant features in the 2D domain.	25
5.1	Relative accuracy $x_{RMS}(P)$ of three independent calibrations.	81
5.2	Deviations and error of d_{bar}	89
5.3	Standard deviations $\hat{\sigma}(C)$ at different tracking distances d_{track}	90
5.4	Relative point accuracy and standard deviation $\hat{\sigma}_C$ for $d_{base} \approx 9m$	96
5.5	Empirical standard deviation $\hat{\sigma}_C$ for $d_{base} \approx 3m$	97
5.6	Comparison of relative point accuracy $x_{RMS}(P)$ and standard deviation $\hat{\sigma}(C)$ without (motor shut off) and under heavy vibrations (motor running).	97

III User Interfaces for 3D Interaction

3.1	Pre-Questionnaire	136
3.2	Post-Questionnaire	136
3.3	Evaluation of selection techniques in handheld mixed reality.	143
4.1	Post-Questionnaire	157
4.2	Users grouped by prior experience	158

IV Creating Mixed Reality Environments

3.1	Interaction devices supported by ARTiFICe.	181
-----	--	-----

Appendix A

User Studies

Selection in Handheld Mixed Reality

Page 01
G01

Selection in Handheld Mixed Reality

Welcome

Thank you very much in participating into this user study.

The study comprises 3 parts and will take approx. 20 minutes for completion:

1. Pre questionnaire
2. Practical exercise: Selection in Handheld MR
3. Post questionnaire

We will request only your gender and age as personal information. Your answers to the questions of the pre- and post questionnaire will be stored and analyzed anonymously.

1. What is your gender?

Female
 Male

2. How old are you?

years

Please proceed to the pre-questionnaire by hitting the "Next" button

Page 02
Pre

Selection in Handheld Mixed Reality

Pre-Questionnaire

Please answer all of the following questions, then proceed to the practical exercise.

3. About how often do you play video games?

Never Once a Year A few time a Year Once a Month Every Week Every Day

4. What % of your gaming is playing mobile 3D games?

0 10 20 30 40 50 60 70 80 90 100

Please specify:

5. Do you have a multi-touch Smartphone?

- Yes
- No
- No, but I regularly use one.

6. Do you have any experience with mobile Augmented Reality?

I have no or little mobile AR experience, but I have experience with Virtual Reality applications

None I used once a mobile AR app I regularly use mobile AR apps I develop mobile AR apps

7. Where do you rank your general 3D spatial skill?

Poor Below Average Average Above Average Excellent

8. Do you have any flexibility or pain issues with your primary hand, fingers or arm?

- No
- Yes, Please specify the pain issue.

Page 03
S01

Selection in Handheld Mixed Reality

Practical Exercise

During the practical part, you will be asked to count virtual objects. Please write down the numbers using paper and pen.

Upon completion of the practical test, please proceed to the post questionnaire.

Page 04
S03

Selection in Handheld Mixed Reality

Information of the Practical Part

9. Please enter the your User ID:

User ID is provided at the end of the practical exercise on the mobile screen.

UserID

10. Please enter, for each selection technique separately, the number of bricks that had covered the lowermost pink brick:

RAYCAST	Bricks
EXPAND	Bricks
DRILL SAMPLE	Bricks

Post-Questionnaire: Selection

Please answer all of the following questions.

11. How appropriate do you feel the time assigned for practice was?

Much too short Moderately too short Slightly too short Appropriate Slightly too long Moderately too long Far too long

12. How comfortable were you with using a mobile phone for task completion?

Very uncomfortable Moderately uncomfortable Slightly uncomfortable Neither Nor Slightly comfortable Moderately comfortable Very comfortable

13. How would you rate the RAYCAST selection technique in terms on how ...

	Extremely Poor	Poor	Below Average	Average	Above Average	Good	Excellent
... Easy it was to select a desired object?	<input type="radio"/>						
... Fast you could select a desired object?	<input type="radio"/>						
... Accurate the selection of a desired object was?	<input type="radio"/>						

14. How would you rate the EXPAND selection technique in terms on how ...

Extremely Poor Below Average Above Good Excellent

A. USER STUDIES

	Poor	Average	Average
... Easy it was to select a desired object?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... Fast you could select a desired object?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... Accurate the selection of a desired object was?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

15. How would you rate the DRILLSAMPLE selection technique in terms on how ...

	Extremely Poor	Poor	Below Average	Average	Above Average	Good	Excellent
... Easy it was to select a desired object?	<input type="radio"/>						
... Fast you could select a desired object?	<input type="radio"/>						
... Accurate the selection of a desired object was?	<input type="radio"/>						

16. Rank the three selection techniques in order of desired use (with 1 being the most desired).

Please drag and drop the selection techniques to the ranks.

- RAYCAST -
- EXPAND -
- DRILL SAMPLE -

17. When determining how much you like using a selection technique, how important in influence on your decision was ...

... How easy it was to select a desired object?

Very Unimportant	Unimportant	Slightly Unimportant	Neither Nor	Slightly Important	Important	Very Important
<input type="radio"/>						

... How fast you could select a desired object?

Very Unimportant	Unimportant	Slightly Unimportant	Neither Nor	Slightly Important	Important	Very Important
<input type="radio"/>						

... How accurate the selection of a desired object was?

Very Unimportant	Unimportant	Slightly Unimportant	Neither Nor	Slightly Important	Important	Very Important
<input type="radio"/>						

... The possibility to select hidden objects?

Very Unimportant Unimportant Slightly Unimportant Neither Nor Slightly Important Important Very Important

18. Regarding the visualization during the refinement process of the DrillSample technique, how helpful and useful was ...

... linear arrangement for spatial visualization of selected objects?

Strongly Useless Moderately Useless Slightly Useless Neither Nor Slightly Useful Moderately Useful Strongly Useful

... possibility to inspect selected objects (zoom, rotate)?

Strongly Useless Moderately Useless Slightly Useless Neither Nor Slightly Useful Moderately Useful Strongly Useful

19. Please state advantages and disadvantages you experienced when using DRILLSAMPLE technique for object selection.

Page 05
G02

20. Please feel free to add any comment about the study and the techniques.

Last page

Selection in Handheld Mixed Reality

Thank You!!!

We would like to thank you very much for helping us.

Close window

Manipulation in Handheld Mixed Reality

Page 01
G01

Transformation in Handheld Mixed Reality

Welcome

Thank you very much in participating into this user study.

The study comprises 3 parts and will take approx. 20 minutes for completion:

1. Pre questionnaire
2. Practical exercise: Transformation in Handheld MR
3. Post questionnaire

We will request only your gender and age as personal information. Your answers to the questions of the pre- and post questionnaire will be stored and analyzed anonymously.

1. What is your gender?

Female
 Male

2. How old are you?

years

Please proceed to the pre-questionnaire by hitting the "Next" button

Page 02
Pre

Transformation in Handheld Mixed Reality

Pre-Questionnaire

Please answer all of the following questions, then proceed to the practical exercise.

3. About how often do you play video games?

Never Once a Year A few time a Year Once a Month Every Week Every Day

4. What % of your gaming is playing mobile 3D games?

0 10 20 30 40 50 60 70 80 90 100

Please specify:

5. Do you have a multi-touch Smartphone?

- Yes
- No
- No, but I regularly use one.

6. Do you have any experience with mobile Augmented Reality?

I have no or little mobile AR experience, but I have experience with Virtual Reality applications

None I used once a mobile AR app I regularly use mobile AR apps I develop mobile AR apps

7. Where do you rank your general 3D spatial skill?

Poor Below Average Average Above Average Excellent

8. Do you have any flexibility or pain issues with your primary hand, fingers or arm?

- No
- Yes. Please specify the pain issue.

Page 03
M01

Transformation in Handheld Mixed Reality

Practical Exercise

Upon completion of the practical test, please proceed to the post questionnaire.

Page 04
M03

Transformation in Handheld Mixed Reality

Information of the Practical Part

9. Please enter the your User ID:

User ID is provided at the end of the practical exercise on the mobile screen.

UserID

Post-Questionnaire

Please answer all of the following questions.

10. How appropriate do you feel the time assigned for practice was?

Much too short Moderately too short Slightly too short Appropriate Slightly too long Moderately too long Far too long

11. How comfortable were you with using a mobile phone for task completion?

Very uncomfortable Moderately comfortable Slightly comfortable Neither nor Slightly comfortable Moderately comfortable Very comfortable

12. How would you rate the MULTI TOUCH manipulation technique in terms on how ...

	Extremely Poor	Poor	Below Average	Average	Above Average	Good	Excellent
... Easy it was to use?	<input type="radio"/>						
... Fast you could manipulate an object?	<input type="radio"/>						
... Accurate the manipulation was?	<input type="radio"/>						

13. How would you rate the DEVICE ORIENTATION manipulation technique in terms on how ...

	Extremely Poor	Poor	Below Average	Average	Above Average	Good	Excellent
... Easy it was to use?	<input type="radio"/>						
... Fast you could manipulation an object?	<input type="radio"/>						
... Accurate the manipulation was?	<input type="radio"/>						

14. Which manipulation technique do you prefer to ...

MULTI TOUCH DEVICE ORIENTATION I liked both the same

... Move an object in two dimensions (i.e. on a plane)?

... Move an object in three dimensions?

... Rotate an object?

... Move and rotate an object at the same time?

... Scale an object?

15. Rank the two manipulation techniques in order of overall desired use (with 1 being the most desired).

Please drag and drop the manipulation techniques to the ranks.

MULTI TOUCH

DEVICE ORIENTATION

16. When determining how much you like using a manipulation technique, how important in influence on your decision was how ...

... Easy it was to use?

Very Unimportant Unimportant Slightly Unimportant Neither Nor Slightly Important Important Very Important

... Fast you could manipulate an object?

Very Unimportant Unimportant Slightly Unimportant Neither Nor Slightly Important Important Very Important

... Accurate the manipulation was?

Very Unimportant Unimportant Slightly Unimportant Neither Nor Slightly Important Important Very Important

17. How would you rate intuitiveness of the MULTI TOUCH technique for ...

... Moving an object in two dimensions?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Moving an object in three dimensions?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Rotating an object?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Moving and rotating an object?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Scaling an object?

Extremely Poor Poor Below Average Average Above Average Good Excellent

18. How would you rate intuitiveness of the DEVICE ORIENTATION technique for ...

... Moving an object in two dimensions?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Moving an object in three dimensions?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Rotating an object?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Moving and rotating an object?

Extremely Poor Poor Below Average Average Above Average Good Excellent

... Scaling an object?

Extremely Poor Poor Below Average Average Above Average Good Excellent

19. When determining how intuitive a manipulation technique was for you, how important in influence on your decision was ...

... Prior knowledge of using multi touch interfaces?

Very Unimportant Unimportant Slightly Unimportant Neither Nor Slightly Important Important Very Important

... Possibility to manipulate an object (move, rotate, scale) by moving/rotating the mobile device?

Very Unimportant Unimportant Slightly Unimportant Neither Nor Slightly Important Important Very Important

20. Please state advantages and disadvantages you experienced when using **DEVICE ORIENTATION** technique for object manipulation.

Page 05
G02

21. Please feel free to add any comment about the study and the techniques.

Last page

Transformation in Handheld Mixed Reality

Thank You!!!

We would like to thank you very much for helping us.

Close window

Annette Mossel, Benjamin Venditti, www.imst.tuwien.ac.at, Interactive Media Systems Group, Vienna University of Technology - 2012