Werner Alexander Isop, BSc MSc

Extended Reality Interfaces For Teleoperation Of Aerial Robots

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Prof. Dr. Dieter Schmalstieg
Institute of Computer Graphics and Vision, Graz University of Technology, Graz, Austria

Examiner

Prof. Dr. Kiyoshi Kiyokawa
Nara Institute of Science and Technology, Ikoma - Nara, Japan

Graz, January 2020

To my family.

The scientist is not a person who always gives the right answers, but who asks the right questions.

Claude Levi-Strauss (1908 - 2009)

Abstract

In recent years, mobile touch devices were increasingly replacing or supporting desktop computing. However, touch devices require direct physical and visual attention, both of which are the scarcest resource in mobile situations. Research on augmented reality and wearable computing has long attempted to address these issues with alternative user interfaces that augment the users perception of the real world. Traditionally, this has been achieved by instrumenting a human user, or human operator, with head-worn displays, tracking devices or similar. Subsequently, trade-offs concerning ergonomics, interactions and data richness were unavoidable. In contrast, the focus of the research presented in this thesis is to bring together elements from visualization and interaction to combine them with mobile robotics. In the context of mobile computing, it proposes new types of teleoperation interfaces utilizing state-of-the-art extended reality technology, effectively operating highly mobile aerial robotic platforms. Important aspects and resulting design requirements are identified to facilitate interactions between the human operator and aerial robot. While, within this scope, the overall goal is to improve task performance, three contributions are presented. Experimental results indicate that if extended reality (either augmented-, mixed- or virtual reality) technology is purposefully used to support robotic teleoperation, task performance can be significantly increased. Moreover, common aspects with regards to visualization and interaction were identified in all three contributions. During this research it was found that such aspects can be expressively summarized inside an overarching classification framework. Importantly, the framework considers existing and well established notations, which are Milgram's Reality-Virtuality Continuum, the Situative Space Model and the Proxemic Notations. However, these frameworks include relevant aspects from a visualization-, Human-Computer Interaction-, or Human-Robot Interaction point of view individually. In contrast, the presented contributions clearly indicate a need of a combined and extended classification framework, but with additional focus on perception. As a consequence, a so called *perceptual distance* is introduced which could be measured between human operator, aerial robot and the according workspace.

With focus on operationality, it mainly considers spatial and geometric relations. As a result, this thesis structures the presented contributions inside a more extensive classification framework which we call the *Perceptual Distance Continuum*. Amongst others, the aspects of this continuum are mainly influenced by the physical distance of the human operator to the aerial robot and physical occlusions between them. Throughout identifying core aspects inside the continuum, resulting impacts on requirements to the experimental operator interface and the aerial robotic platform are highlighted. Ultimately, a guideline is provided to make future experimental- and framework-designs of extended reality interfaces for robotic teleoperation more efficient.

Keywords. Augmented Reality, Spatial Augmented Reality, Mixed Reality, Virtual Reality, Extended Reality, Teleoperation, Visualization, Human-Robot Interaction, Mobile Robotics, Aerial Robots, Micro Aerial Vehicles, Milgram‘s Reality-Virtuality Continuum, Situative Space Model, Proxemic Notations, Classification Framework

Kurzfassung

In den letzten Jahren haben mobile Touch-Geräte zunehmend das Desktop-Computing ersetzt oder unterstützt. Touch-Geräte erfordern jedoch direkte physische und visuelle Aufmerksamkeit, die in mobilen Situationen die knappste Ressource darstellen. Forschung im Bereich Augmented Reality und Wearable Computing hat lange versucht, diese Probleme mit alternativen Benutzeroberflächen zu lösen, welche die Wahrnehmung der realen Welt durch den Benutzer verbessern. Traditionell wurde dies durch Instrumentieren des menschlichen Benutzers, oder des Operators, mit am Kopf getragenen Displays, Tracking-Vorrichtungen und ähnlichen Geräten erzielt. In der Folge waren Kompromisse in Bezug auf Ergonomie, Interaktionen und Datenvielfalt unvermeidlich. Im Gegensatz dazu liegt der Fokus der in dieser Arbeit vorgestellten Forschung darauf, Elemente aus Visualisierung und Interaktion zusammenzuführen, um sie mit mobiler Robotik zu kombinieren. Im Zusammenhang mit Mobile Computing werden neue Arten von Teleoperationsschnittstellen vorgeschlagen, bei denen State-of-the-Art Extended Reality Technologie zur effektiven Steuerung hochmobiler Luft-gestützter Roboter zum Einsatz kommt. Wesentliche Aspekte und daraus resultierende Designanforderungen werden identifiziert, um die Interaktion zwischen dem Benutzer und dem Roboter zu erleichtern. Während das übergeordnete Ziel darin besteht, die Task-Performance zu verbessern, werden in diesem Zusammenhang drei Beiträge vorgestellt. Die experimentellen Ergebnisse zeigen, dass die Effizienz von Tasks erheblich gesteigert werden kann, wenn Extended Reality Technologien (entweder Augmented Reality, Mixed Reality oder Virtual Reality) gezielt zur Unterstützung der Roboter-Teleoperation eingesetzt werden. Darüber hinaus wurden in allen drei Beiträgen gemeinsame Aspekte in Bezug auf Visualisierung und Interaktion identifiziert. Bei dieser Recherche wurde festgestellt, dass solche Aspekte in einem übergreifenden Klassifikations-Framework aussagekräftig zusammengefasst werden können. Wesentlich ist, dass das Framework vorhandene und etablierte Notationen berücksichtigt, welche

Milgram‘s Realitäts-Virtualitäts-Kontinuum, das Situative Space Model und die Proxemik sind. Diese Frameworks behandeln jedoch relevante Aspekte für Visualisierung, der Mensch-Computer Interaktion oder der Mensch-Roboter Interaktion individuell. Im Gegensatz dazu weisen die vorgestellten Beiträge deutlich auf die Notwendigkeit eines kombinierten und erweiterten Klassifikations-Framework hin, wobei der Schwerpunkt jedoch zusätzlich auf der Sinneswahrnehmung liegt. Infolgedessen wird ein sogenannter *Wahrnehmungsabstand* eingeführt, der zwischen Benutzer, Luft-gestütztem Roboter und dem entsprechenden Arbeitsbereich gemessen werden kann. Mit Schwerpunkt auf Operationalität werden hauptsächlich räumliche und geometrische Beziehungen berücksichtigt. In der Folge strukturiert diese Dissertation die vorgestellten Beiträge in einem umfangreicheren Klassifikations-Framework, welcher als *Perceptual Distance Continuum* bezeichnet wird. Die Aspekte dieses Kontinuums werden unter anderem hauptsächlich durch die physische Entfernung des Operators zum Roboter und die physische Okklusion zwischen ihnen beeinflusst. Bei der Ermittlung der Kernaspekte innerhalb des Kontinuums werden die sich daraus ergebenden Auswirkungen auf die Anforderungen an die Versuchsaufbauten der Operator-Schnittstelle und der Roboterplattform hervorgehoben. Letztendlich wird ein Leitfaden vorgestellt, um zukünftige Experimentier- und Framework-Designs von Extended Reality Schnittstellen für die Roboter-Teleoperation effizienter zu gestalten.

Schlagwörter. Augmented Reality, Spatial Augmented Reality, Mixed Reality, Virtual Reality, Extended Reality, Teleoperation, Visualisierung, Mensch-Roboter Interaktion, Mobile Robotik, Luft-gestützte Roboter, Mikroflugzeuge, Milgram‘s Realitäts-Virtualitäts-Kontinuum, Situative Space Model, Proxemik, Klassifikations-Framework

AFFIDAVIT

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

The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Place	Date	Signature

EIDESSTATTLICHE ERKLÄRUNG

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

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

Ort	Datum	Unterschrift

Acknowledgments

I am grateful for all the people that I worked with and that supported me during my PhD at the Institute of Computer Graphics and Vision.

First of all, I thank my supervisor Prof. Dr. Dieter Schmalstieg, who gave me the freedom to follow my ideas and at the same time always provided guidance and support, not only from a scientific but from a human point of view, so that I did not get lost in the vastness of scientific and, most importantly, moral space. I also thank Prof. Dr. Kiyoshi Kiyokawa for agreeing to be my second assessor and examiner and for the valuable feedback and suggestions for finishing this work.

My thanks also go to my colleagues Gabriele Ermacora, Okan Erat, Denis Kalkofen, Christoph Gebhardt, Tobias Nägeli, Friedrich Fraundorfer and Otmar Hilliges. With all of them I had the pleasure to work on different research topics that make up this thesis. Further, I thank all my other colleagues, scientific and staff, I had the opportunity to work with over the years.

I am also grateful for the people close to me for providing their support and sticking with me, when I was too busy. Lastly, my deepest thanks go to my family, for all their love and support.

Contents

1 Introduction

Contents

1.1 Motivation And Problem Statement

In recent years, research on how humans operate machines and interact with digital information, became increasingly important. In the early age of digital computing, users or operators were mostly limited to direct and low-level instructions, for example console inputs based on keyboard commands. Nowadays, since also technology in the field of AR, MR and VR rapidly advanced, keyboard and mouse inputs are no longer the primary way to command machines, in particular robots. With rising popularity in the commercial sector, smart devices like smartphones or tablets are typically used to directly or indirectly teleoperate especially small-sized aerial robots, so called MAVs. However, such interfaces require direct physical and visual attention, both of which can quickly become a scarce resource in mobile situations. Instead, we are currently amidst a revolution of ways how to effectively combine MR-technology with more advanced touch-based [1], or even gesture-based [2], interfaces. The overarching goal stayed the same, which is to improve overall task performance and operator experience at the same time during teleoperation of aerial robots. Importantly, modern head-worn MR-devices, like Microsoft‘s Hololens [3], can present information registered with the 3D environment in a more mobile and compact way. Thus, they are, on one hand, beneficial for such applications. On the other hand, they still encumber the operator up to a certain point.

Regarding advances in VR-technology, state of the art Head-Mounted Displays (HMD), such as the Oculus Rift [4] or the HTC Vive [5] can offer hands-free approaches. Because they are typically coupled to a computational powerful stationary setup, a high degree of immersiveness can be achieved. They enable virtual environments, which are beneficial for teleoperation at far distances. However, they strongly encumber the operator due to their size, weight and limited range because of the fact that they are typically connected to the stationary setup via a tethered data link.

Addressing the problem of encumbering the operator, spatial AR [6] is an entirely unencumbering alternative, which relies on projectors to augment surfaces directly in the environment. However, projector setups are most commonly stationary and can not cover large areas with augmentations, even if they are mounted on a pan-tilt unit [7]. Emphasizing the benefits of XR interfaces if combined with mobile robotics, the project UFO [8] proposed concepts to combine spatial AR interfaces with aerial robots. The goal was to create an entirely novel experience, in which highly mobile aerial robots with physical or virtual on-board projection devices create movable personal projection screens on arbitrary surfaces in the environment (Figure 1.1). However, supporting the XR interfaces with highly mobile aerial robots resulted in a tradeoff between fully unencumbering the operator and limited physical constraints. It was found, that especially parameters like flight times, again limited computational power and increasing complexity of the overall framework affected the interaction between human operator and aerial robot.

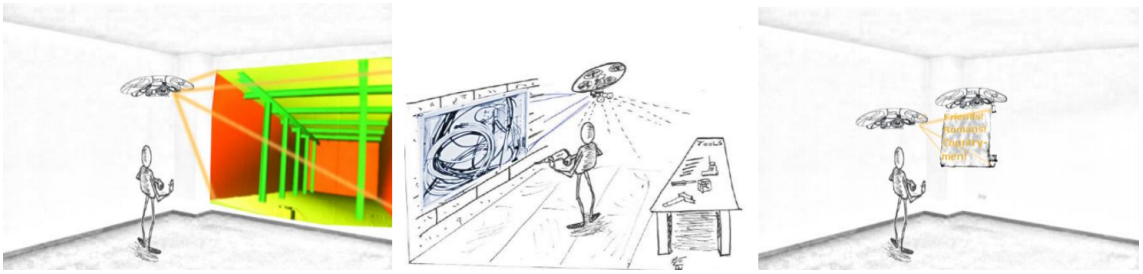


Figure 1.1: Original in-situ visualization and guidance concepts supported by Micro Aerial Vehicles as part of the project User‘s Flying Organizer (UFO).

The focus of the research presented in this thesis lies on investigating three usage-driven scenarios of utilizing recent XR technology for the purpose of effective teleoperation in typical use cases. The goal was to elaborate on what methods best suit the individual cases and in what qualitative and quantitative way task performance was improved. Further, questions arise for the existence of a general method in the context of XR interfaces for robotic teleoperation. Remarkably, there is currently no ultimately effective solution of a specific XR technology that is able to cover all kind of typical use cases. Instead, this thesis outlines and emphasizes that each XR technology has specific advantages and disadvantages for the design of future teleoperation interfaces. Additionally, the three different use cases structure the thesis based on physical distance and occlusion between

human operator and aerial robot. All resulting aspects are summarized in an overarching conceptual classification framework. A short introduction of the framework is given in Section 1.3 and a more detailed motivation and description is given in Chapter 3. Finally, with this classification framework, recommendations of suitable XR-technology and related important design aspects for different use cases are given as an outlook and supportive guideline for future related work.

1.2 Objective Of Research

The research topics of this thesis span several areas (Visualization, Interaction and Mobile Robotics) as well as different disciplines (Extended Reality, Interface- and Interaction-Design, System Design, Robotic Teleoperation). One major goal was to develop and study new ways for human operators to interact with visual content that is augmented onto the real world by highly mobile robots. If rephrased, the goal was to find the answer of the question: "How can humans co-exist with and benefit from increasingly intelligent aerial robots that have a non-humanoid nature and hence provide complementary physical and computational capabilities to the operator?" The interest is on fundamental, novel techniques for interaction between humans and aerial robots, but also between humans and virtual content delivered through the robot. The related research which is discussed in this thesis is driven by several challenging usage inspired scenarios and the developed algorithms and interfaces were implemented and tested during show-cases or statistically evaluated with user studies. The scenarios involve supportive visual information that is generated by a mobile robot for

- ... educational use cases in collocated, mobile and hands-free settings by means of spatial AR (Chapter 5).
- ... remote inspection scenarios of otherwise inaccessible areas (occluding structure) by combining mobile robotics and head-worn MR displays (Chapter 6).
- ... remote exploration of spatially constrained environments from far located distances by extending classical egocentric (camera live view) and virtual exocentric views (3D map view) with abstracted scene representation (Chapter 7).

The research presented in this thesis cover design, implementation and evaluation of advanced XR-methods and interaction techniques, combined with aerial robots. Requirements, methodology, and framework drastically changed depending on spatial aspects (e.g., geometry of the environment, distances, etc.), as well as *perceptual aspects* between a human operator and the aerial robot. As a result, a so called *perceptual continuum* was established, which serves as a classification framework, helping to structure the thesis. The goal of the *perceptual continuum* is to recommend preferred methodology, depending on where a potential use case is located inside this classification framework.

The research questions of this thesis can be listed as follows:

- What relevant methods exist already in the fields of Robotics (aerial robots), Interaction (teleoperation) and Visualization (XR interfaces) to support human operators, ultimately increasing situation awareness and task performance for typical use cases (Chapter 2)?
- What are resulting general requirements of any potential experimental platform and how do they relate to the context of this research (Chapter 4)?
- What visualization methodology can be used or extended in the presented use cases to provide support while a human operator is teleoperating an aerial robot? Is it even possible to improve overall task performance (Section 5.1, Section 6.1, Section 7.1)?
- Is there a lack of common important aspects amongst all use cases and, if so, how do they influence overall task performance?
- Does a more comprehensive classification framework (Chapter 3) that extends existing concepts (Chapter 2) help to summarize and emphasize the afore mentioned lack?
- Is it possible to apply the introduced classification framework to the presented use cases, and, more importantly, is it possible to create a guide which highlights important aspects and resulting design-requirements for future related work (Chapter 8)?

1.3 Structure Of The Thesis - Introduction Of A Classification Framework

It was found that two aspects played crucial roles during experimentation. On one hand this was found to be any potential *physical occlusion* (e.g., walls or other obstacles) between the human operator and the aerial robot and, on the other hand, the *physical distance* between them. Moreover, the experiments indicated that these two aspects have a great impact on the (visual-)perceptual senses of the human user, accompanied by potential effects on the effectiveness of interactions. The term *perceptual distance* is introduced, to emphasize the influence of physical distance on any human user‘s perceptual senses.

To give a better impression of the context of the classification framework, we now want to briefly look at the three presented contributions as part of this thesis. The first related contribution (Chapter 5) involves utilizing spatial AR techniques to provide in-situ guidance. At close physical distance, an aerial robot is teleoperated in a collocated situation by means of high-level interaction. It provides guidance to improve the human operator‘s task performance, whereas no physical obstacles were located in between. In a next step, the experimental design was extended with a MR interface for the purpose of remote inspection of hidden areas (Chapter 6). In this use case, the operator is still teleoperating the aerial robot at closer physical distance, but with an occlusion (wall) in

between. The last contribution finally involves teleoperation for exploration of challenging indoor environments (Chapter 7). This use case focuses on teloperation at any far physical distance, including a full occlusion between the human user and the aerial robot.

The presented contributions discuss how to improve task performance by utilizing (spatial-)AR, MR and VR techniques at different ranges of physical distance. They can be seen as covering the full range of a *continuous physical distance* between human user and aerial robot with bounded lower limit (collocated) and an unbounded upper limit (teleoperation at any far physical distance). Based on the afore mentioned terminology, we define the *Perceptual Distance Continuum (PDC)* as classification framework. It is used to establish a logical structure of the presented thesis and emphasizes important aspects, supporting the experimental- and framework-design of future related work.

This thesis is structured as follows. In a first step, the motivation is put in the context of investigating on novel user experiences of a human operator, specifically by means of XR technology for teleoperation of aerial robots. In this context, a discussion of related work motivates the utilized methods. Based on existing classification frameworks in related research fields (Milgrams RV-Continuum [9] in the context of visualization, the Situative Space Model [10] in the context of Human-Computer Interaction (HCI) and the Proxemic Notations [11] in the context of Human-Robot Interaction (HRI)), a lack of more extensive class definition to properly structure the presented contributions is highlighted. Details of the according experimental platforms, including software framework and physical setups of the aerial robots, are outlined in a separate chapter (Chapter 4). Consequently, the necessity for the PDC as more comprehensive classification framework for future related work is better motivated (Chapter 3).

After discussion of aspects of the PDC and well defining a classification, a relation to the presented contributions in the context of teleoperation at so called CLOSE- (Chapter 5), MEDIUM- (Chapter 6) and FAR-distance (Chapter 7) is established. Three different use cases were investigated on utilizing different types of XR technology and aerial robotic frameworks. Further, the experimental results are discussed in the individual chapters of the contributions with regards to overall task performance. A short reflection indicates if it was possible to even improve task performance by utilizing XR technology-based interfaces for aerial robotic teleoperation.

In a concluding chapter (Chapter 8), findings of the PDC are summarized and impact of most important aspects on overall requirements are discussed for individual use cases. Finally, based on the lessons learned from the PDC, the chapter reveals recommendations for suitable XR technology at different perceptual distances. The goal is to establish a helpful guideline for future work.

1.4 List Of Publications And Collaboration Statement

My work at the Institute for Computer Graphics and Vision led to the following peer-reviewed publications. For the sake of completeness of this thesis, they are listed in

chronological order along with the respective abstracts. An overview of the authors contributions of each individual publication is outlined. Additionally, contributions which are considered as part of the PDC contain references to the according chapters of this thesis.

1.4.1 2016

Micro Aerial Projector - Stabilizing Projected Images Of An Airborne Robotics Projection Platform (CLOSE distance teleoperation, Chapter 5)

W. A. Isop and J. Pestana and G. Ermacora and F. Fraundorfer and D. Schmalstieg
In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*
2016, Daejeon, South Korea

Abstract: A mobile flying projector is hard to build due to the limited size and payload capability of a micro aerial vehicle. Few flying projector designs have been studied in recent research. However, to date, no practical solution has been presented. We propose a versatile laser projection system enabling in-flight projection with feedforward correction for stabilization of projected images. We present a quantitative evaluation of the accuracy of the projection stabilization in two autonomous flight experiments. While this approach is our first step towards a flying projector, we foresee interesting applications, such as providing on-site instructions in various human-machine interaction scenarios.

Author's contributions: W. A. Isop designed and implemented the autonomous MAV flight system (later on called the *SLIM*), including the physical MAV setup, the small-sized laser-projection system and the ROS-based control-framework for (semi-)autonomous flight. W.A. Isop designed and implemented the model for calibration of the projection system. W. A. Isop further created the experimental design for projection-stabilization and evaluated the experimental data. W. A. Isop designed and conducted experiments for the use case of the Micro Aerial Projector acting as "Teaching Assistant". W. A. Isop and Dieter Schmalstieg mainly contributed to paper writing. G. Ermacora, J. Pestana and Friedrich Fraundorfer supported with evaluation of experimental data and paper writing.

1.4.2 2018

Drone-Augmented Human Vision: Exocentric Control For Drones Exploring Hidden Areas (MEDIUM distance teleoperation, Chapter 6)

O. Erat and W. A. Isop and D. Kalkofen and D. Schmalstieg
In: *IEEE Transactions on Visualization and Computer Graphics*
2018, Reutlingen, Germany

Abstract: Drones allow exploring dangerous or impassable areas safely from a distant point of view. However, flight control from an egocentric view in narrow or constrained environments can be challenging. Arguably, an exocentric view would afford a better overview and, thus, more intuitive flight control of the drone. Unfortunately, such an

exocentric view is unavailable when exploring indoor environments. This paper investigates the potential of drone-augmented human vision, i.e., of exploring the environment and controlling the drone indirectly from an exocentric viewpoint. If used with a see-through display, this approach can simulate X-ray vision to provide a natural view into an otherwise occluded environment. The user's view is synthesized from a three-dimensional reconstruction of the indoor environment using image-based rendering. This user interface is designed to reduce the cognitive load of the drone's flight control. The user can concentrate on the exploration of the inaccessible space, while flight control is largely delegated to the drone's autopilot system. We assess our system with a first experiment showing how drone-augmented human vision supports spatial understanding and improves natural interaction with the drone.

Author's contributions: W. A. Isop designed and implemented the improved version of the autonomous MAV flight system (later on called the *SLIM*), including a more lightweight physical MAV setup, the camera stream interface and the ROS-based control-framework for semi-autonomous user-controlled flight (localization and automated path-planning). This includes components 1, 2, 3, 5, 6 and the "HoloLens Comm Node" of component 4 according to Figure 4.8. Further, W. A. Isop collaborated with Okan Erat in interface-design, experimental-design/evaluation of results for the user-study and inter-sensor-calibration. W. A. Isop created the offline generated indoor model used for experimentation. W. A. Isop supported with paper writing.

1.4.3 2019

SLIM - A Scalable And Lightweight Indoor-Navigation MAV As Research And Education Platform

W. A. Isop and F. Fraundorfer
In: *RiE 2019, the 10th International Conference on Robotics in Education*
2019, Vienna, Austria

Abstract: Indoor navigation with micro aerial vehicles (MAVs) is of growing importance nowadays. State of the art flight management controllers provide extensive interfaces for control and navigation, but most commonly aim for performing in outdoor navigation scenarios. Indoor navigation with MAVs is challenging, because of spatial constraints and lack of drift-free positioning systems like GPS. Instead, vision and/or inertial-based methods are used to localize the MAV against the environment. For educational purposes and moreover to test and develop such algorithms, since 2015 the so called droneSpace was established at the Institute of Computer Graphics and Vision at Graz University of Technology. It consists of a flight arena which is equipped with a highly accurate motion tracking system and further holds an extensive robotics framework for semi-autonomous MAV navigation. A core component of the droneSpace is a Scalable and Lightweight Indoor-navigation MAV design, which we call the SLIM (A detailed description of the SLIM and related projects can be found at our website: https://sites.google.com/view/w-

a-isop/home/education/slim). It allows flexible vision-sensor setups and moreover provides interfaces to inject accurate pose measurements form external tracking sources to achieve stable indoor hover-flights. With this work we present capabilities of the framework and its flexibility, especially with regards to research and education at university level. We present use cases from research projects but also courses at the Graz University of Technology, whereas we discuss results and potential future work on the platform.

Author's contributions: W. A. Isop designed and implemented the improved version of the autonomous MAV flight system (later on called the *SLIM*), including a more lightweight physical MAV setup, according hardware- and software-interfaces and the control-framework for semi-autonomous user-controlled flight (localization and automated path-planning). W. A. Isop further created the experimental design to test the flight performance of the *SLIM* and evaluated the experimental data. W. A. Isop and Friedrich Fraundorfer contributed to paper writing.

Force Field-Based Indirect Manipulation Of UAV Flight Trajectories

W. A. Isop and F. Fraundorfer
In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2019, Macau, China

Abstract: For a variety of applications remote navigation of an unmanned aerial vehicle (UAV) along a flight trajectory is an essential task. For instance, during search and rescue missions in outdoor scenes, an important goal is to ensure safe navigation. Assessed by the remote operator, this could mean avoiding collisions with obstacles, but moreover avoiding hazardous flight areas. State of the art approaches enable navigation along trajectories, but do not allow for indirect manipulation during motion. In addition, they suggest to use egocentric views which could limit understanding of the remote scene. With this work we introduce a novel indirect manipulation method, based on gravitational law, to recover safe navigation in the presence of hazardous flight areas. The indirect character of our method supports manipulation at far distances where common direct manipulation methods typically fail. We combine it with an immersive exocentric view to improve understanding of the scene. We designed three flavors of our method and compared them during a user study in a simulated scene. While with this method we present a first step towards a more extensive navigation interface, as future work we plan experiments in dynamic real-world scenes.

Author's contributions: W. A. Isop designed and implemented the robotics framework, including the backend for physics simulation and the frontend for visualization. W. A. Isop designed and implemented the indirect manipulation method flavors. W. A. Isop further created the experimental design to compare the different manipulation method flavors and evaluated the experimental data. W. A. Isop designed and conducted experiments for the proof of concept real-world flights as part of the supplemental video. W. A. Isop and Friedrich Fraundorfer contributed to paper writing.

High-Level Teleoperation System For Aerial Exploration Of Indoor Environments (FAR distance teleoperation, Chapter 7)

W. Alexander Isop and Christoph Gebhardt and Tobias Nägeli and Friedrich Fraundorfer and Otmar Hilliges and D. Schmalstieg
In: *Frontiers in Robotics and AI*
Speciality of Human-Robot Interaction, 2019

Abstract: Exploration of challenging indoor environments is a demanding task. While automation with aerial robots seems a promising solution, fully autonomous systems still struggle with high-level cognitive tasks and intuitive decision making. To facilitate automation, we introduce a novel teleoperation system with an aerial telerobot that is capable of handling all demanding low-level tasks. Motivated by the typical structure of indoor environments, the system creates an interactive scene topology in real-time that reduces scene details and supports affordances. Thus, difficult high-level tasks can be effectively supervised by a human operator. To elaborate on the effectiveness of our system during a real-world exploration mission, we conducted a user study. Despite being limited by real-world constraints, results indicate that our system better supports operators with indoor exploration, compared to a baseline system with traditional joystick control.

Author's contributions: W. A. Isop, Christoph Gebhardt, Otmar Hilliges, and Dieter Schmalstieg contributed conception of the user interface and design of the study. W. A. Isop implemented the user interface and wrote the first draft of the manuscript. W. A. Isop and Tobias Nägeli designed and implemented the aerial robotic system. W. A. Isop and Christoph Gebhardt performed the statistical analysis of the study. W. A. Isop, Christoph Gebhardt, Friedrich Fraundorfer, and Dieter Schmalstieg wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

2

Background And Related Work

Contents

The purpose of this chapter is to introduce general terms with regards to the contributions and main research topics discussed as part of this thesis. Also related work to the according methodologies is outlined. Finally, related classifications and notations, supporting the PDC as overarching framework, are presented. A general view on the relevant core topics, which were mainly treated as part of this thesis are depicted in Figure 2.1, whereas also the PDC is put into proper context. While in this chapter, only an overview of the core components, utilized methods and overall goals of this research is given, the following chapter (Chapter 3) introduces a detailed description and definition of the PDC.

Concerning related work, it is noteworthy that this thesis mainly covers topics which lie in the fields of visualization (AR, MR and VR, with XR as the overarching term), HRI (assistive teleoperation interfaces) and mobile robotics (small-sized aerial robots), also including interdisciplinary aspects. For example, Card et al. [12] defines visualization as "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition", which emphasizes a close connection between visualized information and interaction of a human user. As a result, it is noteworthy that aspects, both in the field of visualization but also interaction, were considered. Specifically, regarding XR-based teleoperation interfaces for aerial robotic platforms in the context of HRI. Interaction with highly mobile aerial robots also suggests to present most relevant work in the field of mobile robotics. In relation to the PDC, also related work in terms of classification frameworks from the field of HRI and HCI were considered as necessary to better motivate the introduced concepts of the PDC framework. Finally, since the overall goal of the

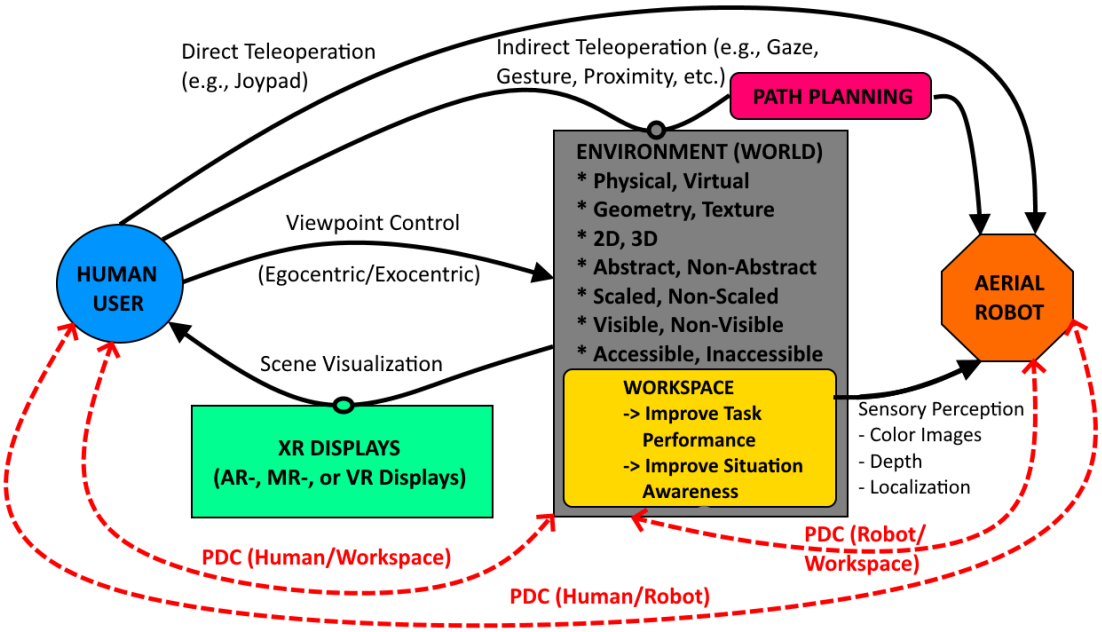


Figure 2.1: General view of utilized components (depicted in bold font), methods and overall goals inside the workspace during XR-supported aerial robotic teleoperation. In addition, the PDC between the relevant components is put into context (connectors depicted in red).

presented contributions was to improve task efficiency during teleoperation, also study evaluation methods for the according XR interfaces are summarized.

2.1 General Terms In Extended Reality

According to Franklin et al. [13], the difference between categories in Extended Reality can be expressed as follows.

- *Augmented reality (AR)* adds digital elements to a live view often by using the camera on a smartphone. Examples of augmented reality experiences include, for example, Snapchat lenses or the game Pokemon Go.

- *Virtual reality (VR)* implies a complete immersion experience that shuts out the physical world. Using VR devices such as HTC Vive, Oculus Rift or Google Cardboard, users can be transported into a number of real-world and imagined environments such as the middle of a squawking penguin colony or even the back of a dragon.

- *Mixed reality (MR)* combines elements of both AR and VR; real-world and digital objects interact. Starting with remarkable advances of HMD-based MR technology in the early 2000s (e.g., Kiyokawa et al. [14]), popularity of head-worn MR-devices

continuously increased up to more recent technology, like Microsoft's HoloLens [3] smart glass.

However, it seems that the term "Extended Reality" is relatively young and not widely established yet in the scientific community. According to industry [15], the term evolved as "An umbrella term encapsulating Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR), and everything in between. Although AR and VR offer a wide range of revolutionary experiences, the same underlying technologies are powering XR". Others refer to this terminology in a similar manner: "Three hot, interrelated technologies fall under the umbrella of extended reality: virtual reality (VR), augmented reality (AR), and mixed reality (MR)" (Goundner et al. [16]). Recently, only few related works in the field of visualization (Jantz et al. [17]) utilized this terminology to summarize AR, MR and VR under an overarching term. However, a most suitable definition seems to be again originated in [16], stating that: "Extended Reality is a term referring to all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables. It includes representative forms such as augmented reality (AR), augmented virtuality (AV) and virtual reality (VR) and the areas interpolated among them."

2.2 Extended Reality In Human-Robot Interaction

Effective human-robot collaboration requires that robots generate human-aware behaviors during planning [18] and interact with users during execution. Together, these processes enable common ground with respect to shared beliefs, desires, and intentions. However, in particular regarding natural interaction, for example state of the art language processing methods considerably limit the current scope of such interactions. This is primarily due to a mismatch between how humans and robots represent and communicate information, which makes it difficult for robots to, first of all, interpret human intentions and second, effectively express their own intentions. The resulting communication barrier [19] has been the focus of much past research in the field of HRI. Such barriers are particularly concerning when mixed human-robot teams are involved in collaborative tasks, for reasons of safety, efficiency and ease-of-use. This is emphasized in the Roadmap for U.S. Robotics [20], which outlines that *humans must be able to read and recognize robot activities in order to interpret the robots understanding.* Recent work in HRI has sought to address this challenge in a variety of ways, including the generation of intent-expressive motion [21] and explicable task plans [22] as implicit signals, as well as verbalization of intentions in natural language [23] and the use of explicit signaling mechanisms [24]. Recent advances in AR, MR and VR suggest an alternate method for mediating human-robot interactions, enabling communication of intent by leveraging the physical HRI space as a shared canvas for visual cues. While preliminary work in this field has been performed for many years (e.g., Sato's Interactive Hand Pointer for controlling a robot in a human

workspace [25]), recent technological advances in mixed and VR systems significantly enhance the feasibility and promise of this approach. Recently, systems have been developed to visualize trajectories of mobile wheelchairs and robots ([26] and [27]), with results suggesting that humans prefer to interact with a robot when it presents its intentions directly as visual cues. But while there has been much recent research in this area ([28], [29], [30] and [31]), most recent work presents passive, non-interactive systems with limited scopes. Moreover, these systems have not taken a true mixed reality perspective, as they have not considered the changing context of their environment while projecting information. Recent advances with AR and VR technology [32] have significantly advanced the state of what is possible within XR environments, and are now allowing researchers to even meaningfully incorporate XR in HRI ([33], [34] and [35]).

2.3 Extended Reality Interfaces For Teleoperating Aerial Robots

This section, in the following, provides background knowledge and most relevant related work of XR interfaces which are used to interact with, or teleoperate, small sized aerial robots. The section starts with discussing related aerial robotic platforms, which were also designed for extensive use in experimental indoor environments. While they all could be combined with XR interfaces, the focus is also on small size, low weight, sufficient flight times and enough computational power. Ultimately, these criteria make such platforms attractive for close interaction with-, or teleoperation by- human users in indoor environments. After related robotic platforms are presented, related work about interaction with small-sized aerial robots, XR interfaces and according interaction designs are discussed.

2.3.1 Experimental Aerial Robotic Platforms

Design, implementation and evaluation of small sized aerial robots (also called MAVs) for indoor use was widely investigated on during the last two decades. Bouabdallah et al. [36] present design and control of an MAV for indoor flights, whereas they do not provide results in free-flight but create a testbench to test their underlying control algorithm. Further, How et al. [37] introduce a real-time indoor autonomous vehicle test environment which they call RAVEN. They discuss design of their framework and evaluate performance of their vehicles during flight operation. More recent work addresses MAV design, control and localization based on vision algorithms. For example, Vempati et al. [38], in general discuss design and control of an MAV, specifically discussing non-linearities, to achieve a highly accurate model. In addition they suggest localization based on SLAM algorithms like PTAM. However, they use a considerably larger and heavier MAV frame with an all-up weight of about 1 kg without any additional sensory setup. Also, they do not focus on a full educational framework with flexible sensor setup. Moreover, their design does not allow for injection of ground-truth measurements from external sources, like motion

trackers. A more lightweight design is presented by Loianno et al. [39]. Their concept aims for an overall weight of below 250 g, also including a monocular vision camera for localization and navigation. However, their full setup does not include an RGBD-sensor which can be beneficial for mapping tasks in spatially constraint indoor environments if rich and more detailed maps are required [40]. Kushleyev et al. [41] introduce a very small and lightweight design as part of a swarm of agile MAVs. The ready-to-fly (RTF) weight of their design is below 100 g and they also localize their MAVs with an external motion tracker. However, their setup does not provide any computational unit onboard for processing more expensive tasks like online mapping or any additional vision-sensor. In comparison, a lightweight solution below 100 g is also available from DJI called Ryze Tello [42], whereas a monocular camera is attached onbaord providing live-streams. On the other hand, this solution provides only very limited pay-loads and is not capable of carrying more advanced vision sensors like RGBD or onboard projection devices like laser projectors. Besides, various off-the-shelf products aim for small-sized flight-control solutions and MAVs ([43], [44]), with fully integrated vision sensors [45], also emphasizing an educational context. On the other hand they do not provide a full educational- and research-framework, including software solutions for higher-level robotic tasks. Finally, none of these solutions discuss a flexible and scalable design for multiple configurations of computational units and sensors.

2.3.2 Interaction With Aerial Robots

Interaction with aerial robots, supporting a human user or operator, is an emerging research area. For example, Obaid et al. [46] proposed a flying robotic agent to help humans keep the environment clean, by persuading a user to pick up trash, leading him/her to the nearest trash bin and then communicating with him/her when the job is done. Cauchard et al. [47] proposed an elicitation study on how to naturally interact with aerial robots. Similarly, Obaid et al. [48] investigated user-defined gestural interactions to teleoperate an aerial robot. Emotional states to aerial robots have also been studied. Cauchard et al. [49] report that several of their participants compared the aerial robots to a pet, which entitles it to anthropomorphic status in that situation. Most off-the-shelf aerial robots do not promote close interaction due to fast rotating propellers being an immediate danger to humans. Some researchers have addressed the safety aspects of interacting with an aerial robot, such as the picocopter [50] and the collisions-resilient flying robot [51], where light-supported agents with cages are made safe to coexist with humans. A small aerial robot [52] to be used in a ball to manipulate speed and behavior for new sports interaction has also been proposed as a safe way for humans to interact with it. Bitdrone [53] is a tiny aerial robot that allows for many types of physical interaction between the user and the robot. Summarizing, a core aspect of robotic teleoperation by means of XR interfaces, especially at close distances, is safety, which must be considered at all circumstances.

2.3.3 AR Interfaces Based On Aerial Projection Devices

The first research challenge emphasized by this thesis was to provide a new natural user interface based on spatial Augmented Reality (AR). The term AR is used to describe systems that allow a user to perceive the real-world together with computer generated virtual information. Traditionally, superimposing graphics over the real-world has been achieved by requiring users to wear near-eye video or optical see-through head-mounted displays (HMDs). Despite decades of research, hardware is still cumbersome to wear and suffers from restricted display resolution, field-of-view and depth-of-field. Recent demands for mobility have led to the adoption of handheld devices [54] for AR, such as smartphones overlaying a live video feed with virtual graphics. While this approach removes the need for head-worn displays, it still requires users to constantly hold a device in the line of sight. A different approach is taken by spatial AR [6]. Since the early 2000s, it has been a popular topic for mobile interaction. However, most approaches for room-sized environments either cover all relevant surfaces with arrays of projectors or rely on steerable projection. This idea was pioneered with the everywhere display projector [55] and later extended with real-time reconstruction from commodity depth sensors in the Beamatron project [7]. Hörtner et al. [56] further introduced a spatial display paradigm without using a projector, but controllable moving visible objects in 3D physical space, the so called Spaxels.

Other work considers mobile projectors, which are handheld or worn on the users head or body. They can be combined with depth sensors [57] or inertial sensors [58] to react to the user‘s movement and, potentially, a changing environment. However, visual light projectors with sufficient brightness for spatial interaction usually require a stationary power source. Truly mobile, battery-powered projectors can only operate at very short distances.

Laser pointers concentrate the emitted energy in a single spot and, consequently, can achieve a significantly better contrast than conventional projectors with the same power budget. A steerable laser pointer can be used to point to a particular task location. A shoulder-mounted implementation of a steerable laser pointer has been used in a tele-assistance scenario [59]. However, the same authors later proof that a single point is not sufficient for conveying complex instructions [60].

Instead of a steerable single laser point, a scanning laser can be used. Maeda et al. [61] describe a head-mounted projective display with a scanning laser mounted co-axial to the observer‘s eye. Schwerdtfeger et al. [62] explore head-mounted and stationary scanning lasers for spatial augmentation. In both cases, the workspace is a major limitation.

Because of the obvious technical difficulties, there has been little work on flying projectors. Scheible et al. [63] demonstrate outdoor flying projection with a commercial visual light projector weighting 200g, mounted on a large octocopter platform with payloads of up to 3.5kg. These authors suggest a Human-Machine Interaction (HMI) scenario, but do not consider the problem of projection stabilization.

In terms of aerial projection platforms, the closest work to ours, presented in Chapter 5,

was introduced by Hosomizo et. al [64]. They suggest a flying projection platform and approach the problem of image stabilization by combining dead reckoning and computer vision. However, they, again, use a commercial, heavy projection system and off-board computation of the image stabilization. They do not report exact measurements of the uncompensated position of the projection. Furthermore they do not provide information about the distance to the projection surface during stabilization. Moreover, results for image stabilization are not evaluated in flight and only shown while the MAV is suspended from wires to overcome weight constraints.

In contrast, our work, presented in Chapter 5, introduces a small sized aerial robotic platform (MAP), which has all essential components - a single board computer (SBC), a vision camera and a projector - included onboard. We deploy a custom lightweight steerable laser projector, which is calibrated, and also evaluate performance of projection stabilization during flight. We use inter-sensor calibration of the laser projector with respect to the flight management controller‘s IMU and the motion tracking system. Additionally, we also propose to describe the intrinsics of the projector with a model which is similar to a camera pinhole model. This approach not only improves accuracy of image stabilization, but also makes it compatible to common computer vision algorithms. Further with the MAP, we combined the advantages of unencumbered users and full mobility in an everyday environment. While a proof of concept [63] hints at the great potential of this idea, our work addresses the research questions if its possible to put resulting new systems capabilities into practice. Specifically, benefits are that the human user can move freely in the environment and use both hands to interact with the physical and virtual world. Also, aerial robots augmenting the environment can move independently from the user, avoiding restrictions in terms of which parts of the environment can be used for projection and which parts of the scene can be observed from on-board cameras.

2.3.4 MR Interfaces For Remote Through-Wall Inspection

Existing work on occluded or remote space discovery with aerial robots proposes a variety of interaction techniques to steer the aerial robot and visualize the data coming from its sensors. Depending on the visualization of the sensor data, mostly from cameras, related work can be categorized into egocentric control and exocentric control. Moreover, the following related work is with focus on remote visualization techniques involving live video.

2.3.4.1 Egocentric Control

Egocentric control techniques visualize camera images from the first-person view of the aerial robot and immerse the user into the remote location currently occupied by the aerial robot. Using an HMD to display video from a aerial robot, Mirk and Hlavacs [65] created a virtual tourist application. However, the user was not given full control of the aerial robot to prevent crashes; only the user‘s head movements were translated into the yaw

rotation of the aerial robot. Hansen et al. [66] capture eye gaze, while the aerial robot pilot is looking at the camera stream. The 2D vector formed between screen center and the point gazed at on the screen is mapped to speed and rotation around a 3D axis in the aerial robot‘s local frame. As humans tend to rapidly change their gaze direction, this technique may be problematic for flight control whenever the pilot looses concentration.

Higuchi et al. [67] synchronize head movements of the user with a aerial robot, except for pitch and roll rotations. While this gives an intuitive control, the latency between the pilot‘s movements and response of the aerial robot can quickly create motion sickness. In addition, the motion dynamics of the aerial robot make it impossible for the aerial robot to exactly replicate the path taken by pilot‘s head, negatively affecting the spatial understanding of the human. As summarized by Chen et al. [68], egocentric robot control presents the user with the several problems, the most severe ones being narrow field of view (FOV), orientation and altitude misjudgement and a general lack of scene understanding.

2.3.4.2 Exocentric Control

In contrast to egocentric control, an exocentric control technique steers the aerial robot while the user is observing it directly. As discussed by Cho et al. [69], exocentric control is prone to accidents due to left-right confusion between human user‘s and aerial robot‘s local coordinate frames. Kashara et al. [70] tackle this problem by allowing users to control the aerial robot with a touch screen device in their own reference frame and mapping control commands into the aerial robot‘s local coordinates automatically. However, the users had to observe the aerial robot with the device‘s camera for pose estimation and move it on the 2D screen, which is not possible in the presence of occlusions. In addition, 2D gestures do not allow for an intuitive interface for generating a motion vector that is a combination of axes. Similar to Kashara et al., Hashimoto et al. [71] also provides a touch screen based control, but they place a camera at a fixed viewpoint to observe the robot. This is not feasible during an occluded scene investigation.

Saakes et al.[72] uses a camera to observe a ground robot from a third person view. In an unknown occluded environment, using another robot just increases the complexity. Sugimoto et al. [73] and Hing et al. [74] provides a visualization to observe the robot from an exocentric point of view. However, their systems limit the freedom of the viewpoint and makes it hard to relate surrounding colliders to the robot. Karanam et al. [75] use WiFi signals transmitted by aerial robots to monitor them behind the occluding structures.

Zollmann et al. [1] focuses on the spatial understanding problems that arise when the aerial robot is far away from the user. They use an exocentric AR display based on the backfacing camera of a handheld tablet. The aerial robot‘s altitude over the terrain and distance to the user is visualized in 3D on top of the video. However, if the aerial robot faces dense obstacles in close proximity, this technique does not provide a detailed enough visualization for accurate control. Bergé et.al. [76] create a synthetic point cloud resembling a 3D reconstruction obtained by a aerial robot and visualize it in immersive

VR. They also develop a method to evaluate the difficulty of finding a target.

All these techniques demonstrate the potential of using an exocentric viewpoint for aerial robot control, but do not allow for easy and intuitive navigation. Comparing direct and indirect manipulation for this purpose is the main contribution of the work, presented in Chapter 6.

2.3.5 XR Interfaces For Exploration Of Indoor Environments

Very recent work on XR interfaces for robotic teleoperation in indoor environments is for example discussed by [77]. In their work, they introduce high-level interaction methods based on gesture and language, but for ground-based robots. While they also suggest seamless switching between navigation, inspection and manipulation tasks, they use traditional egocentric 2D views and a 3D map to improve task performance. Recent work on XR interfaces for teleoperation, but with aerial telerobots is discussed by [78], [79], [2] and [80]. All contributions use state-of-the-art AR, MR or VR input devices, whereas they also design high-level interactions for their human-robot interface. Their overall goal is to improve task-efficiency when commanding aerial telerobots in indoor environments. Remarkably, they all compare their teleoperation systems against baseline systems (using traditional joystick or keyboard controls) and their independent variable in the study corresponds to what type of teleoperation interface the participants used. However, high-level XR interfaces are not necessarily connected to immersive HMD devices or based on natural gaze or language commands. Instead, like introduced in our work about high-level teleoperation in indoor environments [81], the remote operator can also refer to an interactive 2D scene topology created in real-time, which is supporting classical MR 2D and 3D views. Further, aerial exploration missions in challenging indoor environments are considered, where simple and robust input devices can be beneficial to improve task performance. Related work, which might be also of interest in connection to our work (details are discussed in Chapter 7), is introduced by [82]. The work presents three prototypes with different methods to control an aerial telerobot. Interestingly, they also make use of an abstract floor-plan representation of the scene. However, this plan is static and not autonomously created in real-time. Although related work already proposed abstract topological views for the control of teleoperation systems, we introduce a fully working teleoperation system that refers to an interactive scene topology, created in real-time during flight. This raises the interesting question, if the performance of our teleoperation system is also preserved when put into practice. Compared to a variety of related teleoperation systems with similar mission complexity ([83], [84] and [85]), we evaluate the performance of the system. We utilize XR visualization techniques, but using egocentric and exocentric MR views, combined with an abstract topological scene view for indoor exploration. Details about our system, experimental evaluation and results are outlined in Chapter 7.

2.3.6 Interface Evaluation Methods (User Study)

Evaluation of the designed XR interfaces for teleoperation was achieved by using the one-way analysis of variance (ANOVA) [86] or a paired samples t-test [87] if the measured quantity was parametric (normally distributed). Otherwise, measures were compared using Wilcoxon signed-rank test as, for example, questionnaire responses are non-parametric. Additionally, a 95% confidence interval (CI) [88] was given. The according studies aimed to examine the relationship between utilized interface- or system-conditions and task performance of the human operators. The samples involved up to 22 participants from Graz University of Technology (Austria). The tests were conducted to answer several research question, formulated as the following hypothesis:

- The introduced XR interface or system has a significant effect on the human operators mental load during teleoperation.
- The introduced XR interface or system has a significant effect on the human operators task time during teleoperation.
- The introduced XR interface or system has a significant effect on the human operators comfort during teleoperation.

Remarkably, the results presented in Chapter 6 and Chapter 7 indicate that there is a statistically significant positive effect of the designed XR interfaces or XR systems on the overall task-performance.

2.3.6.1 Collection Of Data

The data used for the studies to measure performance included mainly task-time measured in seconds, mental load (Nasa-TLX [89]) and users comfort during teleoperation. The subjects participating at the user-studies were students, researchers and employees of the Graz University of Technology. Their ages ranged from 22 to 32, whereas the samples included male and female participants.

2.3.6.2 Analysis Of Data

According to the work of Venkatesh et al. [90], researchers can examine relationships between two variables by comparing the mean of the dependent variable between two or more groups within the independent variable. During the studies, presented as part of this thesis, all participants were exposed to all designed interfaces or systems (within-subject design and full counterbalancing), with at most two conditions. In order, the means and variances of the sample data were compared with regards to the effect of the condition on the users performance metrics. The data analysis process included two stages. The first stage included analysis of the samples with respect to a 95% CI. The second stage included hypothesis testing with one-way ANOVA, paired samples t-test or Wilcoxon signed-rank test.

2.3.6.3 Choice Of Statistical Tests And CI

In the process of examining the relationship between variables, it is convenient using t-test or ANOVA to compare the means of two groups on the dependent variable (Green and Salkind [91]). The difference between these two tests is that the t-test can only be used to compare two groups, while ANOVA can be used to compare two or more groups if the data is parametric. When selecting the data analysis methods for the studies presented in this thesis, ANOVA was considered in the first step as the more general approach. Further, the advantage ANOVA has over t-test is that any potential post-hoc tests of ANOVA allow for better control of type 1 errors. In order, ANOVA was considered as the main evaluation method. In addition, since the CI is completely independent of the effects of heterogene variances, analysis of 95% CIs was provided, if possible. Finally, if the experimental data was found to be non-parametric (e.g., questionnaire responses), a Wilcoxon signed-rank test was used for significance testing.

2.4 Related Notations And Classification Frameworks

Regarding robotic teleoperation by means of XR technology, it is noteworthy, that the fields of visualization- and HRI lack of common terminology. Moreover, during the research presented in this thesis, it was found that a more interdisciplinary perspective seems to be missing. As a result, the PDC was established as a conceptual classification framework to first of all well structure the main contributions of this thesis. To better support and substantiate the PDC, closely related classification- and notation-frameworks are discussed in the following three sections. They involve Milgram‘s Continuum [9], which provides basic knowledge for the XR interfaces but from a visualization point of view. Another important notation framework are proxemics [11], but which discuss the interaction between a human user and a mobile robot based on spatial relations. The third, framework, which is the Situative Space Model, is originated in the field of HCI and emphasizes importance of a system‘s ”understanding of the interactive situation in which a given human agent is in, while also informing the system‘s possibilities for reacting to the behavior of the human agent or other events” [92].

2.4.1 Milgram‘s Reality-Virtuality Continuum

Milgram et al. [9] defined the Reality-Virtuality (RV) continuum, which allowed them to better relate methods in AR and VR to the real world (Figure 2.2). The continuum spans the two extremes from only presenting the unmodified real world to presenting a purely virtual world. Everything between these extremes can be classified as Mixed Reality (MR). Within the range of MR applications, AR is situated closer to the real world, because it mainly shows a real world representation, e.g., in the form of a video stream, which is modified by virtual objects.

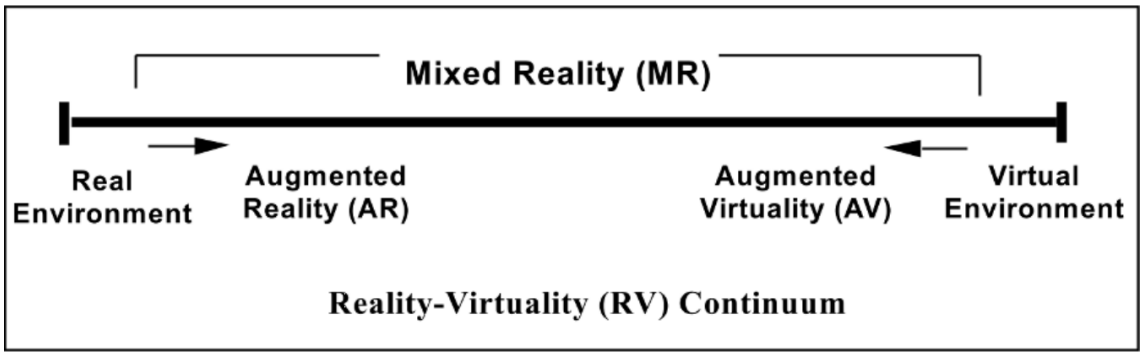


Figure 2.2: The RV Continuum encompasses all possible variations and compositions of real and virtual objects.

AR overlays virtual data directly on top of the real world. A common goal of AR is to communicate additional information about real world entities or to support certain tasks of a user. The overlaid information can consist of simple annotations, such as the names of the restaurants that are closest to the user. The overlays can also guide a user through maintenance and assembly tasks. In these examples, AR technology is used to visualize information regarding the surrounding real world environment.

The major aspects which are considered in the RV-continuum are discussed in the following, whereas the focus is on the introduction of a general classification of 7 displays (either monitor based or head-worn) in the context of MR.

In his work, Milgram accentuates that classification also depends on a variety of other aspects. He further emphasizes that, first of all, viewpoints are of importance (either the human user is egocentrically immersed or looking at the scene from exocentric perspective). Second, he raises the awareness for importance of conformal mapping which is usefull for see-through devices. At this point Milgram mentions that a conformal mapping is "much less critical for monitor-based, non-immersive displays", however this aspect can be vital if human users interact with aerial robots in specific use-cases. Consequently, it was also emphasized in the classification framework, introduced as part of this thesis. Summarizing, the taxonomy of the RV-continuum depends on three major properties:

- Reality: Real versus virtual representation of the scene.
- Directness: Primary world objects are either viewed directly or with utilizing "some electronic synthesis process".
- Immersion: Does the user need to feel completely inside a real or virtual scene?

Milgram also explains visual perceptual issues that arise if display classes are combined with interaction of the human user with the environment. It is noteworthy, that he also mentions aspects of remote-interaction with robotic agents, however in a very brief manner only. He further addresses mobile-telepresence and tele-existance [93]. However,

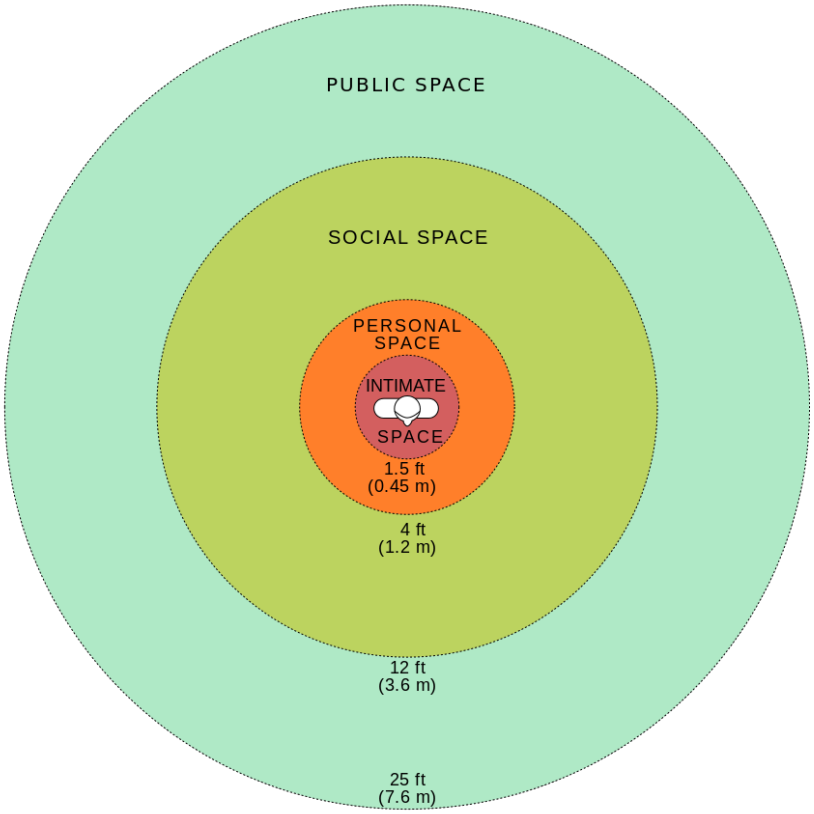


Figure 2.3: An overview of Edward T. Hall's interpersonal distances of man, subdivided into categories.

detailed aspects of visualization and HRI are not treated, especially regarding multisensory perception.

2.4.2 Proxemic Notations

Since proxemics play an important role in this work, this chapter shortly introduces basic terms. Hall [11] determined the limits of proxemic zones for humans, which are shown in Figure 2.3. He categorized distance as intimate, personal, social, or public. Since humans are incapable of accurately measuring distances, he also details how humans use their sense receptors to gauge the space between each other [94]. Not surprisingly, the boundaries fall at points of sensory shift. Hall acknowledges that he did his research on a small population with specific cultural and educational background. As a conclusion his subdivision of space was considered not perfectly accurate and lacked a detailed investigation of cross-cultural aspects. However, follow-up work [95] widely proofed his concepts. Nowadays proxmeics are an important basis for not only human-human interaction cases, but moreover if interaction between humans and robots is considered. Classical proxemic notations address perception with a focus on "interpersonal distance". Although, these aspects have

a strong subjective character and may depend on the indiviudal, a classification based on concrete spatial relations was established. The classification framework, introduced in this thesis, aims for respecting the proxemic notations with regards to perception, since there seems to be no other convenient notation which even enables dividing the physical surrounding of the human user spatially. Moreover, related work suggests a correlation between efficiency of interaction, perception of distances regarding visualization, and the far zones of interpersonal distances. As a result, the classification framework, presented as part of this thesis, also builds on metric interspatial relations between human users and aerial robots.

Connected to this, related work investigated on to what extent do humans comfortably approach to hovering aerial robots. Considering the afore mentioned work, social proxemics are well understood up to a certain point in classical HRI. In general, with regards to robots, Han and Bae [96] proofed that human users position themselves related to a teaching-robot, at a distance which is directly related the physical height of the robot. Further, with regards to commercial solutions, it must be noted that proximity gets of increased relevance if HRI, but with focus on flying robots, is concerned. Also, a in relation to the notations in classical HRI, similar cultural aspects were found and proofed by [97]. Abtahi experimented with an unsafe and a safe-to-touch drone, to check whether participants instinctively use touch for interacting with the safe-to-touch drones [98]. Interestingly, Han and Bae [99] found that human users approached closer to an aerial robot if flying at eye leveled height, compared to when flying overhead the human user during interaction. In order, also flight altitude is important with regards to social proximity for use-cases involving a human user and aerial robot. In general, it can be concluded that if interaction between human users and aerial robots is concerned, details of spatial- and perceptual aspects seem to be still highly underinvestigated.

2.4.3 Situative Space Model

The Situative Space Model (Pederson et al. [100], [92]) suggests that anything of interest with regards to the interaction between a human user and a system is constrained to happen within its realm for the HCI dialogue to technically work. From an egocentric perspective, human users have their own individual perspective of the surrounding environment. They are situated within a local place with their body and can manipulate a particular set of objects (pen and paper they have in their hands, or web pages on a laptop in front of them). Such objects are in order considered as a directly manipulable subset of all the objects that in some sense are available to the specific human. Depending on physical and virtual context, each object can be available in different ways: some are manipulable by the human user, others the human might only be able to examine, select (preparation for manipulation), or simply recognize as the objects they are, without an immediate possibility to manipulate them from where the humans are situated in space. On a slightly higher level of abstraction, it is possible to distinguish between an action

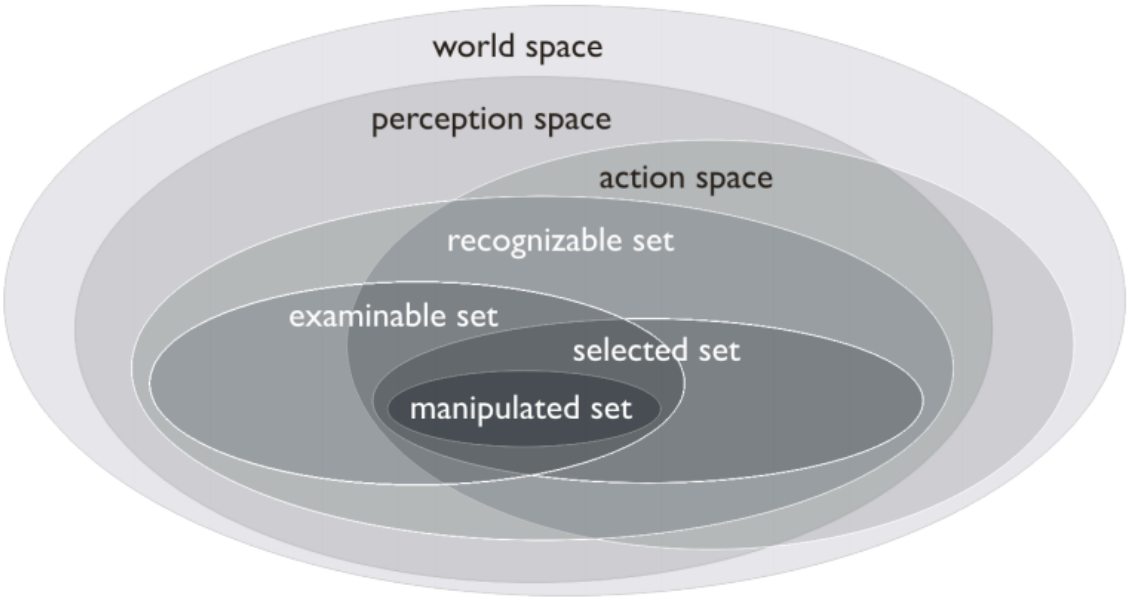


Figure 2.4: Depicted is an overarching space, including the set of all physical and virtual objects inside the Situative Space Model [10].

space and perception space for each human agent. As a result, the overarching space and the individual sets of the Situative Space Model can be summarized in a highly abstract representation (Figure 2.4). An extensive description of the model can be found in [10].

The work of Pederson was further used and adapted by later works. Surie et al. [101] investigate on an activity recognition approach based on the tracking of a specific human actor's current object manipulation actions. To this purpose, they adapt the situative space model and define space-categories which are depicted in Figure 2.5.

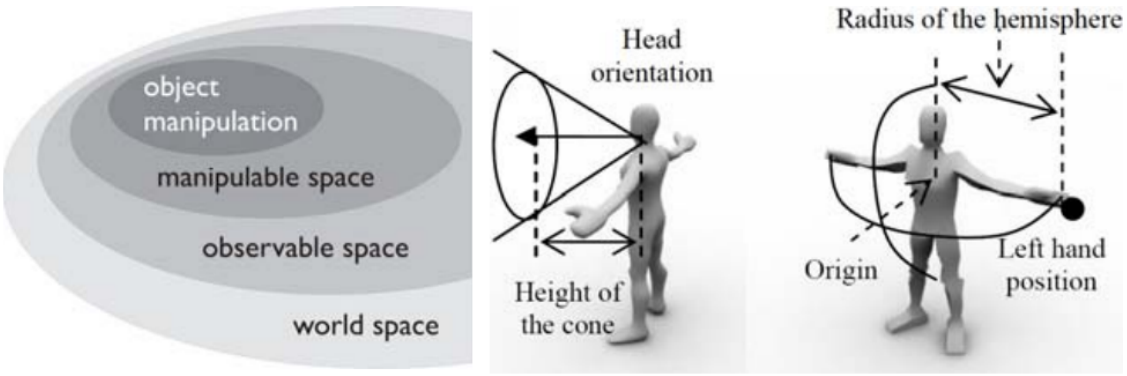


Figure 2.5: (left) Overview of the situative space model. (middle) Space which is observable (in the human users FOV). (right) Space that is manipulable (inside the human users reach).

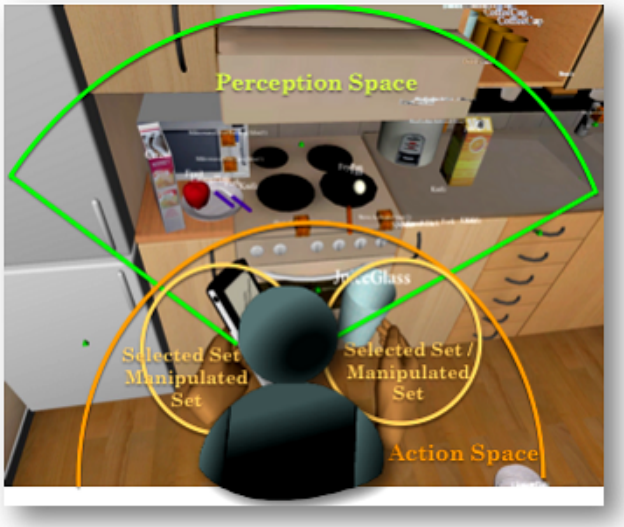


Figure 2.6: Perception space, action space, selected set and manipulated set captured from egocentric perspective in an VR-simulated kitchen (Surie et al. [102]).

While their overall goal was to make the model "functional" the categories can be described as follows:

The world space (WS) contains the set of all objects known to the system. In a real-world setting it might be determined by all objects carrying for example RFID tags. In a potential VR setting used for evaluation purposes, it is the set of all objects included in the virtual model of the environment.

The observable space (OS) is the set of objects within a cone in front of the human actor's eyes with this cone following the head movements. The height of the cone is limited by the walls in an indoor environment and visual occlusion further affects the number of objects within this space.

The manipulable space (MS) is the set of objects within a hemisphere in front of the human actor's chest. Such a shape is motivated by the fact that humans have two hands and the assumption that they manipulate objects within reach of their hands. The hemisphere follows the human actor's chest movements and its radius is equal to the maximum distance between the chest and a hand.
They further define the task of object manipulation when objects are manipulated by an actor. In this case, two events can be generated: grabbed and released. Although the actor can manipulate objects with both hands, we do at the moment not make any distinction between right and left. Object manipulations represent the operations performed by an actor during the accomplishment of an action (and activity).

The focus of the Situative Space Model is on object manipulation by human users, considered from an egocentric perspective. An example is shown in Figure 2.6 with an

implemented adaptive and personalized support in a virtual kitchen. In this example, the relationships between smart objects (so called containers, surfaces and actuators) were egocentrically modeled using the Situative Space Model. However, objects are considered to be either not observable, observable but not manipulable, or observable and manipulable. The Situative Space Model also mentions the aspect of occlusion in the users field of view [100] but in the context of visual perception only (visibility of objects). Moreover, the situative space does not clearly define interdependency of the human user with regards to an aerial robotic agent and its according workspace at full perceptual level. The aspects which are addressed do not discuss effects of all human senses, for example smell. Pederson also states that the result is a blunt categorization from a rough spatial perspective. For example, if an object is physically reachable by the human (for example in front of the chest) and the human user simply turns away his gaze, the object is, per definition, not observable.

3

Introduction Of The Perceptual Distance Continuum

Contents

The purpose of this chapter is to present details of the afore discussed classification framework. Introduced as the PDC in Section 1.3 to motivate the structure of this thesis, its main purpose is to help and guide the reader. While the presented PDC can only give design guidelines based on preliminary findings, it should also help to make the community aware of overlaps and gaps between the visualization-, interaction- and robotics world.

3.1 Main Aspects Of The PDC

The PDC requires involvement of *at least* one *aerial robot*, one *human user (operator)* and a spatially well defined and constrained *workspace*. These types of physically present objects are called *members* of the PDC. Since these members can be also "supported" by means of XR technology, visualization is important here. The reader should also keep in mind that the overall goal of the definition of the PDC is to help improving task performance of a human operator in similar use cases that are discussed as part of this thesis. It is further assumed that any involved robot requires interaction, because of not being able to act fully autonomously (for the majority of complex tasks this assumption is still valid nowadays [103]). Perception of the aerial robot (e.g., it's spatial location) from the human's POV is vital. With increasing autonomy, less interaction and visualization is required directly with the aerial robot, shifting attention to the perceived surrounding. However, the autonomy aspect has no direct effect on classification of the perceptual

distance itself with regards to visual and interaction aspects. More important is to consider that it is not relevant between which members the perceptual distance is measured. Also the term perceived surrounding needs clarification at this point by defining it as any potential workspace. Regarding all these aspects, the PDC covers potential *behaviours* of its *members*, whereas they can be either passive or active. While *active members* are enabled for *perception* (with all relevant perceptual senses), *augmentation*, *motion* and *manipulation* (direct or indirect) of a spatially defined space, *passive members* are not.

Summarizing the afore mentioned aspects leads to the following definitions:

- For the PDC members we assume at least one human operator (1:n) which must be able to directly perceive and interact with at least one aerial robot (1:n) that does not act fully autonomously and/or at least one workspace (1:n).
- Considered is at least one aerial robot, and such cases could be extended by more agents. Whereas in this thesis, discussions are limited to a single aerial robot, in general the PDC considers also multiple robotic agents (Chapter 8).
- Tasks and goals are well defined inside a spatially constrained workspace.
- The workspace could be physically shared between the human operator and aerial robot, meaning the operator is, in best case, *easily able to physically reach the aerial robot* and its surrounding or, in worst case, *not able to reach it at all*.
- The workspace could be either static or dynamically changing. Even if it is static, high mobility of the aerial robot is vital. This could be due to the fact that the workspace is hard to reach (e.g., high up in the air), or dynamically changing physical distances between human operators and aerial robots are required (e.g., as a result of potential safety concerns in a shared workspace).
- It is assumed that the aerial robot is able to have full reach inside the desired workspace, and no physical manipulation of the environment is required. The aspects related to this research (physical aerial manipulation) are not treated by this thesis. The presented use cases focus on either that perception (exploration) or augmentation of (projection onto) the environment is required.

Depending on the focus on the individual category of PDC members, a perceptual distance can relate to *passive members* or *active members*. Properties of a *PASSIVE* member are,

- It is not able to perceive (meaning it is able to visually perceive, smell, feel, etc.) its environment, AND
- It is not able to augment its environment, AND
- It is not able to inherently move or morph and, thus, static, AND

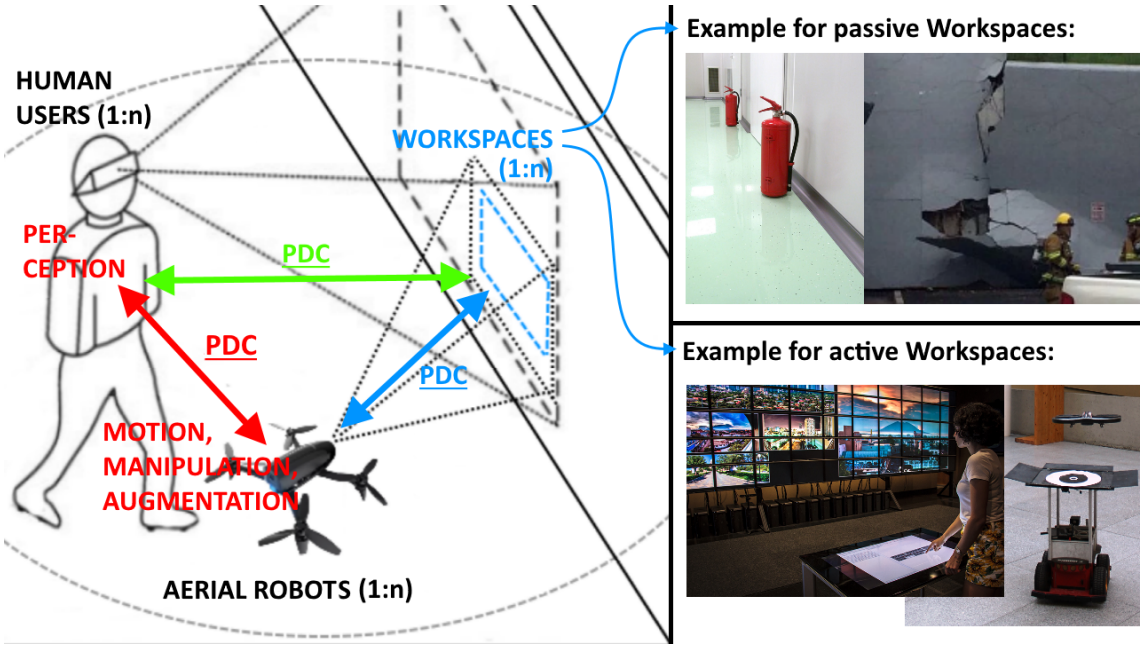


Figure 3.1: (left) Main members of a more general classification framework (human operator, aerial robot and workspace) including interdependencies and actions of members (perceive, augment, manipulate). The arrows in red indicate focus of the PDC. (right) Examples of passive and active workspaces.

- It is not able to manipulate its environment (either directly or indirectly via teleoperation).

whereas an *ACTIVE* member is defined with:

- It is able to perceive its environment, OR
- It is able to augment its environment, OR
- It is able to inherently move or morph, OR
- It is able to manipulate or teleoperate its environment.

We clarify potential interdependencies in this general definitions and introduce a focus of this thesis as part of the PDC. An overview of the perceptual relations between the main member categories of a general classification framework (human operator, aerial robot and its according workspace) is depicted in Figure 3.1. In addition, examples of a passive workspace (fire extinguisher, cracked wall) and active workspace (virtual wall, moving platform) are given.

According to Figure 3.1, the relation between all three members could be summarized as interdependency of ...

- HUMAN OPERATOR $\leftrightarrow$ AERIAL ROBOT

- AERIAL ROBOT ↔ WORKSPACE
- WORKSPACE ↔ HUMAN OPERATOR

The contributions presented as part of this thesis in Chapter 5, Chapter 6 and Chapter 7, mainly consider:

- The perceptual distance from one human operator to one aerial robot.
- Perception, augmentation and teleoperation (no physical manipulation of the environment) of members of the PDC.
- Passive workspaces, which are not able to perceive, augment or manipulate their environment. In addition, the workspaces were not able to move or morph.

In the most general sense, *perceptual distance* is a combination of spatial relations between a human operator and all other members of the PDC, from the perceptual POV of a human operator. It was found that the perceptual distance is mainly shaped by two aspects, which are:

- *Physical distance* between a human operator and an aerial robot.
- *Physical occlusion* of the aerial robot from the human operator's physical viewpoint.

Based on these two aspects, we identified perceptual distances inside the PDC, which are CLOSE, MEDIUM and FAR. Although this classification seems course-grained, it is sufficient to characterize the use cases studied in Chapter 5, Chapter 6 and Chapter 7.

3.1.1 Physical Distance

We start discussing the perceptual distance based on the physical space that separates the human operator and the aerial robot. For this purpose, we utilize the well established proxemic notations which have been extensively investigated in the field of HRI, but are still underinvestigated for aerial robots. These notations are also shortly introduced in Section 2.4. With this work, we would like to fuse aspects of both visualization- and interaction worlds considering aerial robots. The goal is to structure the contributions discussed as part of this thesis, which is based on the following considerations:

- It makes sense to utilize proxemic notations as part of the PDC to specifically classify closer perceptual distances. This is because of its tangible spatial definition.
- Besides vision, also other perceptual aspects motivate classification of a closer perceptual distance.
- The physical distance between human operator and aerial robot greatly influences perceptual aspects which are specifically relevant for visualization and interaction.

3.1.2 Physical Occlusion Of The Aerial Robot

A better definition of CLOSE perceptual distance requires an upper boundary for this category. We directly make it dependent on upper- and lower-boundaries inside the proxemic notations. Terminology of the proxemics in this case introduces CLOSE and FAR phases of the individual distance categories. In contrast, the PDC utilizes all close- and far phases of the proxemic notations, up to the public distance, to classify CLOSE and MEDIUM perceptual distance. Beyond the upper limit of the close phase of the public distance of the proxemic notations, the PDC defines a FAR distance, indepedent of any other related aspect.

An aerial robot would be perceived at CLOSE distance to the human operator, once inside the close phase at public distance of the proxemic notations, and, at very FAR distance, once the physical distance exceeds upper boundaries of the far public phase. Additionally, on a perceptual level, it makes sense to introduce and classify a distance which is in between the two extremes. We define such a MEDIUM distance as the space where the aerial robot is occluded, and its physical representation can‘t be directly seen by the human. For example, such cases could involve:

- The aerial robot is not physically seen at perceptual close distance, because the human operator‘s FOV does not cover the aerial robot.
- The aerial robot is not physically seen at perceptual close distance, because there is physical occluding geometry in between.
- The aerial robot is not physically seen because of occluding geometry, however, a virtual representation (either scaled or non-scaled according to a 1:1 coherence of virtual and physical world) can be recovered by means of visualization.

Occlusion depends on the occluder‘s properties. Such properties could be thickness, type of material, but also if the geometry is closed (i.e., not involving cracks or holes) and if it fully separates the human operator from the aerial robot. This defines at what level the senses are isolated from the aerial robot. The afore-mentioned aspects regarding the occluder are depicted in Figure 3.2.

CASE 1 (Figure 3.2a) represents the case when the aerial robot and its workspace is perceptually in a close range to the human operator. The only situation when the human operator does not see the aerial robot occurs, if the operator turns away the FOV of the physical viewpoint. Perception of the aerial robot may still be supported by a variety of other human senses, including touch, sound or even smell. The effort to get visual access to the physical representation of the aerial robot or its workspace is comparably low. Direct interaction with the aerial robot (either physically or by means of collocated teleoperation) is possible.

CASE 2: (Figure 3.2b) depicts another case when direct visual access suffers, since the aerial robot and its workspace are physically occluded. As the consistency of the occluder is a crucial aspect, we define a so called *weak occluder* with the following properties:

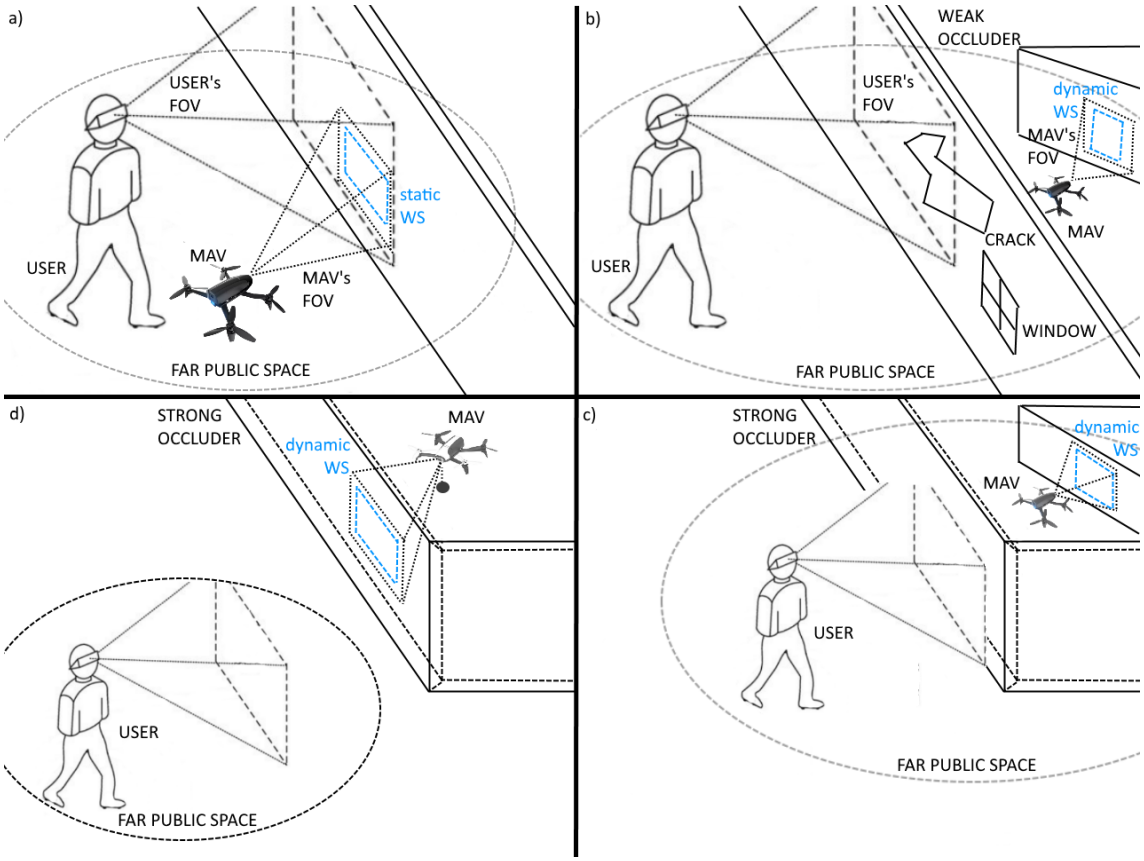


Figure 3.2: Exemplary situations to clarify use cases when the human operator is not able to see the physical representation of the aerial robot from physical viewpoints only.

- Thin physical or not fully closed structure, allowing for penetration of sound, but also other influences which could be harmful to the human operator.
- Damaged structure, involving cracks or holes allowing for penetration of sound and occasional physical visibility.
- Partly transparent structure which can be, for example, windows.

Note that direct physical access to the aerial robot and its workspace would take considerable effort of the human operator. Further, we still consider the aerial robot and its workspace to be inside the close phase at public distance of the proxemic notations. Relevance is discussed in Section 3.2 and Section 3.3.

CASE 3: (Figure 3.2c) Perceiving the aerial robot and its workspace is not possible at all due to presence of a *strong occluder*. A strong occluder has the following properties:

- A thick/massive physical structure. It fully isolates the human operator from the aerial robot on perceptual level.

- No damaged structure, not allowing for any penetration of sound or occasional physical visibility.
- Direct physical access to the aerial robot is not possible at all, even though the physical distance between human operator and aerial robot, including its workspace, is inside the close phase at public distance of the proxemic notations.

CASE 4: (Figure 3.2d) is similar to CASE 3, but also classification of the close phase at public distance ends (far public phase). This means that, assuming presence of strong occlusion of the aerial robot, the human operator is located at considerably increased physical distance. This, in turn, can have great impact on visual aspects and interaction aspects.

3.2 Aspects Related To Visualization

When changing from a closer physical distance to more far distances, perception of details of the environment could suffer, in particular, if a limited FOV is assumed. This was also found during the research discussed in Chapter 6, where inspection of more far distant areas of a scene was required. The human operator has to change the physical viewpoint, although this is only possible if the closer space is physically accessible.

Visualization could help at this point by extending the physical viewpoint by a virtual one. Potentially, with the help of the aerial robot, from a perceptual POV a lack of details at FAR distances is strongly dependent on the use case and the required level of detail in the perceived environment. The boundaries between a MEDIUM and FAR perceptual distance *without considering any physical occluder* could be considered as gradual. However, the proxemic notations classify multiple spatial ranges already, including an unbounded FAR public phase. The PDC relates to this classification by defining CLOSE perceptual distance as inside the close phase at public distance of the proxemic notations and FAR perceptual distance as beyond. Depending on the overall task, XR-supported teleoperation outside the limits of the close phase of the public distance could lead to problems regarding perception of other human operators.

Bringing physical distant space closer to the human operator was investigated in Chapter 6. A transitional interface (overview and detail) was used to switch between a physical and virtual viewpoint, zooming into scene details. Regarding the addressed MR setup, this could lead to problems. If virtual information is overlaid with the physical environment, the local physical surrounding and any virtual representation of the remote location must not lead to confusion of the human operator. By utilizing pure VR, this problem is tackled at the cost of losing perception of the local physical environment. If, for example, manipulation of an occluded closer physical space is required (e.g., disaster relief forces breaking through a wall), then visual access could be granted via overlay of a virtual representation of the occluded scene. For such use cases, a visual distortion of the true physical model may not be helpful due to a wrong spatial understanding. As a consequence, we argue

for a distinct classification of close or medium perceptual distance by requiring that a 1:1 coherence of the real physical world and any virtual information must be conserved [9]. We distinguish close and medium perceptual distance, based on the afore mentioned 1:1 coherence of real and virtual world, if occluding geometry is assumed and the human operator is still inside a proxemic range to other members of the PDC. This leads to another important aspect for separation between close and medium distance, since occlusion, like discussed in Section 3.1.2, in general means the human operator *has no direct visual access to the workspace without additional visualization- or XR-techniques.* Subsequently, visualization becomes mandatory to support the human operator when teleoperating the aerial robot.

FAR distance is defined as a physical upper boundary, beyond the close phase of the public distance of the proxemic notations. It further depends on which level of detail is required regarding the objects in the environment and the interactions between human operator and robot. This is important if no occlusions are present at all (Section 3.3).

3.2.1 Relation To The Presented Contributions

We now want to shortly relate the perceptual distance onto the presented contributions (Chapter 5 to Chapter 7). Chapter 5 considers a use case where the human operator is interacting at closer range with the aerial robot. This use case does not involve any manipulation task; rather, it concentrates on *perception* and *augmentation* of the environment as part of an in-situ guidance task. Without any occlusion, the aerial robot is hovering at close distances to the human operator (inside the far public phase of the proxemic notations, Figure 2.3). It further utilizes a projection device to augment the environment and, thus, is able to provide visual guidance, entirely without encumbering the human operator. However, utilizing spatial augmented reality to project arbitrary screens in the environment typically requires more effort in terms of required hardware and framework complexity, compared to encumbering the human with a head-mounted display. Additionally, HMDs require less effort to keep projections in the field of view of the human operator. Even if the long term goal might be to fully unencumber the human operator, both configurations still do not provide a perfect solution and trade-offs must be considered. Chapter 6 further investigates on this tradeoff, whereas the main contribution is on providing an XR interface to naturally command an aerial robot through an occluder. The interface is based on the Hololens, acting as a head-worn interactive MR display. Although, the human operator has no direct visual access to the workspace, the XR interface enables the operator to perceive a virtual 1:1 representation of the occluded environment. While the user study revealed naturalness of interaction, it must be noted that interaction happened inside the upper limit of the close phase of the public distance of the proxemic notations. In relation to the PDC and to aspects of visualization, a classification as MEDIUM distance emphasizes the conserved 1:1 coherence between real and virtual world. This aspect is crucial, since the mapping can be beneficial for inspection or

maintenance tasks of disaster relief forces. Beyond the MEDIUM distance, one condition that is mandatory for this classification can not be fulfilled any more. At FAR distance, visualization of a 1:1 coherent environment can not be established at all (full perceptual isolation due to occlusion), or is not helpful, because it would lack of details. Thus, other visualization techniques, like zooming or overview and detail must be utilized. These aspects are also reflected in the contribution, presented in Chapter 7, whereas the focus is on the design of a teleoperation interface and according underlying aerial robotic system. Since interaction is expected to happen at far distances, the interface utilizes purely virtual exocentric 3D- and abstract 2D map-views. Further, it combines MR egocentric views where virtual objects are overlaid over a camera live-stream of the aerial robot. Instead of an immersive interface, for the presented use case of aerial indoor exploration, a classical Window On the World approach (Mouse-Monitor-Keyboard combination) was utilized. The main goal here was to emphasize that an abstract topological scene representation in 2D, combined with according high-level controls, can be an effective interface for such FAR distance teleoperation applications. Mainly, due to the abstract 2D representation of the scene, immersiveness was not a primary aspect of the interface design.

3.2.2 Emerging Questions Regarding Design-Aspects

Based on the rich amount of aspects and interdependencies between them, the following design questions from a visualization point of view seem important:

- What type of XR (AR, MR or VR) is feasible for the different perceptual distances, and is there a preferred type of reality that is most beneficial for the specific use case?
- How does the technology affect the level of encumbering of the human and what level is acceptable?
- Which view modes are feasible, and are there most convenient ones for certain perceptual distances in specific uses cases?
- Are there combinations of realities and view modes which are efficient throughout all perceptual distances, or are they highly task-dependent?

In conclusion, it is not obvious how to extend teleoperation with visualization methods amongst the presented use cases at different physical distances. For example, if the proper reality (AR, MR or VR) and view mode has to be selected for the interface. Moreover, establishing a generalized method seems to be impossible if acceptable effort in terms of framework complexity must be guaranteed.

3.3 Aspects Related To Interaction

Section 3.2 discussed the main aspects of the PDC (Section 3.1) and their influence on visualization. We stated that, based on the physical distance, occlusion and affected visualization aspects, a classification of a perceptual distances into three classes (CLOSE, MEDIUM and FAR) is required. At the close perceptual distance, we defined that the human operator must be able to see a physical representation of the aerial robot. Perception is influenced by increasing distance and physical occlusions. At far distances, direct perception of the aerial robot is unavailable, because of lack of perceived visual details of the aerial robot and its surrounding scene (assuming 1:1 coherence is conserved). At a close perceptual distance, the human operator is further able to physically manipulate the aerial robot. The human can physically reach the location of the aerial robot and its workspace, being able to touch it. The aerial robot is inside the close phase at public distance of the proxemic notations and no occlusion is present. At medium distance, involving occlusions and still inside the close phase at public distance, direct physical interaction can not easily be established. Instead, the aerial robot must be teleoperated. This interaction, in turn, is still natural, if a 1:1 coherence of physical and virtual environment can be conserved.

3.3.1 Emphasizing Boundaries Of Medium Perceptual Distance

At medium perceptual distance, we observe significantly increased difficulty with direct interaction, for example, object manipulation. Poupyrev et al. [104] evaluate basic direct object manipulation techniques (Go-Go Pick-and-Place [105] and Ray-casting [106]) in virtual environments. Their findings clearly indicate an increasing task time (and decreasing overall task performance) for object re-positioning with increasing distance. Interestingly, for experimental evaluation, they defined an upper boundary for the manipulation distances of 6 *virtual cubits* [107]. Considering typical distances of the human arm-span ($1814.99 \pm 87.72mm$ [108]), this would accord to $6cubits \cdot \frac{1.81499m}{2} = 5.445m$. This is approximately in the middle of the close phase of the proxemic‘s public distance. Another interesting work with regards to purely virtual scenes was introduced by Plumert et al. [109]. During experimentation with subjects, it was found that estimating a bisected distance was significantly more accurate at closer distances ($3.0m$) and lacked accuracy at distances above $7.5m$.

Recent research of Whitlock et al. [110] presents results about the effect of distance on different interaction modalities (comparing voice, gesture and remote control). They compared the modalities at different distances in a range from 8 to 16 feet ($2.4m$-$4.9m$), while their findings also suggest considerations for designing efficient and intuitive interactions. However, it is remarkable that their experiments were designed for room-scale AR applications and further they also defined the maximum range for interaction inside the close phase at public distance of the proxemic notations.

Voida et al. [111] evaluated if object manipulation techniques in VR are applicable

to AR projection systems. They required users to reposition 2D objects projected onto surfaces. They state that the distance of the object from the user is a primary consideration their study results further indicate that "it is not always reasonable to expect a user to walk up to an object before they manipulate it", once the manipulated object is no longer inside the arms reach. This aspect also emphasizes separation between a close and medium distance based on a weak occluder, even if only comparably low effort would be necessary to physically reach the aerial robot.

At far physical distance perception and direct interaction with objects (either based on a physical, or 1:1 scaled virtual, representation) is significantly more difficult or even impossible. With regards to visualization, classification of such a perceptual distance can have two causes. First, a strong occluder between human operator and aerial robot is present, and a 1:1 coherence of physical and virtual scene representation can not be conserved any more. Second, the distance is outside proxemic notations (beyond definition of the public distance), so that perception of the scene lacks details. At far perceptual distance, it is impossible for the human operator to directly interact with (physically manipulate) the aerial robot, since the robot itself and the workspace are inaccessible. Further, direct (natural) object interaction based on a 1:1 coherence of virtual (or augmented) information and physical scene representation is impossible.

3.3.2 Relation To The Presented Contributions

In Chapter 5, interaction with a *flying robotic companion* at very close distances was considered, while no physical direct manipulation of the aerial robot was involved. Instead, it was commanded depending on the movements of the human operator, detecting if it approached too closely. Both human operator and aerial robot (active members) were able to share a workspace (passive member). The human operator was physically manipulating-, and the aerial robot was augmenting the workspace.

In Chapter 6, we extensively discussed through-wall interaction. The human operator is standing in close physical range, but the workspace was seperated by a weak occluder (thin structure, soft material, only partly occluded), and perception was not fully disabled. By utilizing a synthetic model of the environment, we were able to grant visual access to the otherwise hidden workspace of the aerial robot. Interaction happened at an approximate 1:1 coherence of real and virtual representation. Direct object manipulation (pick-and-place) was implemented. The physical distance between viewpoint of the human operator and the aerial robot (including its workspace) was always inside the close phase at public distance of the proxemic notations ($< 7.6m$). However, the presented through-wall interaction technique also raised the necessity for a transitional interface, to zoom into scene details on demand. As a consequence, the human operator was enabled to switch between physical viewpoint (1:1 coherence of physical and virtual scene) and a virtual viewpoint, where a 1:1 coherence was not conserved anymore.

Finally, Chapter 7 discusses teleoperation of an aerial robot based on purely virtual

data. Again, interaction was happening behind a weak occluder, meaning that the human operator was not fully perceptually isolated from the aerial robot and its workspace. However, purely virtual viewpoints were utilized. Thus, a 1:1 coherence of physical and virtual representation of the world was not conserved.

3.3.3 Emerging Questions Regarding Design-Aspects

The following important design questions arise from an interaction point of view:

- At what level is it meaningful to interact with the aerial robot (low-level vs. high-level interaction)?
- If direct object manipulation is concerned, are there interaction methods which work better for different kinds of perceptual distances in specific use cases?
- Is there a general interaction method which is efficient throughout all perceptual distances, or is this highly task-dependent?

3.4 Classification of the Distance Between Human Operator and Robot

As we can now better define the classes of the PDC (Figure 3.3), the following considerations are important:

- The PDC classifies CLOSE, MEDIUM and FAR perceptual distances, whereas CLOSE and MEDIUM perceptual distance are bounded, and the upper boundary of the FAR interval is open.
- Taking into account physical occlusions, the PDC introduces two different transitions from the CLOSE to the FAR phase. The reason is that, assuming an occluder between human operator and aerial robot (either strong or weak), the robot is hard to perceive or even not perceived at all. However, a natural interaction and scene perception could be still recovered with appropriate visualization and interaction techniques (XR interfaces), implying a MEDIUM distance. In the previous sections, we found that this is significantly harder to achieve with increasing spatial distance. If we assume a 1:1 coherence of physical and virtual scene, this recovery seems to work until a certain distance, whereas related work indicates that, beyond those spatial bounds, perception of distances and naturalness of interaction significantly drops.
- If a strong physical occluder is considered, and no recovery of any perceptual aspects, in particular visual aspects, can be granted, then the perceptual distance is classified as FAR, independent of the direct physical distance between human operator and aerial robot.

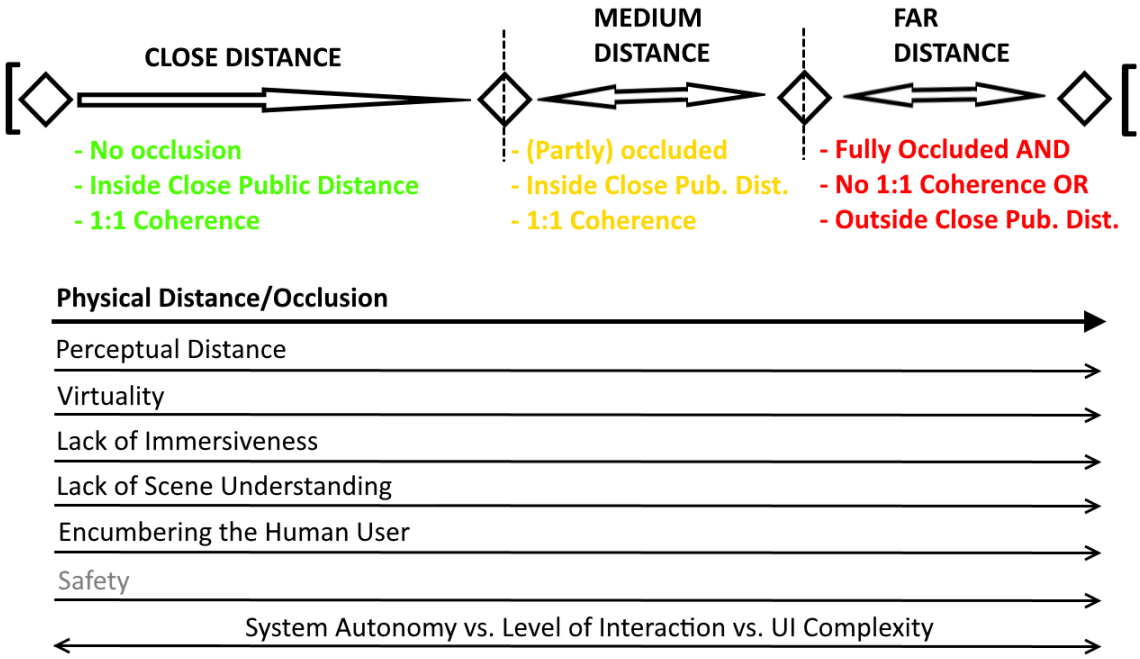


Figure 3.3: Classes of perceptual distance (CLOSE, MEDIUM and FAR) and according boundaries of the distance intervals.

- If a weak physical occluder is considered, and the human operator is still inside the proxemic notations, it can be assumed that perceptual aspects are still enabled up to a certain point. In that case, even if no artificial recovery of visual perception of- or interaction with the aerial robot is possible, the perceptual distance is classified as MEDIUM. Once beyond the close phase of the public distance, the distance must be classified as FAR.

3.4.1 Classification as CLOSE Perceptual Distance

For classification as CLOSE perceptual distance, we assume that the following conditions must be valid:

- The human operator‘s physical distance to the robot must be inside the limits of the public distance according to the proxemic notations.
- The human operator must be able to easily reach the physical location of the aerial robot.
- The human operator must be able to have direct visual access to the physical representation of the aerial robot with low-effort (e.g., simply by changing his field of view).

- No physical occlusions must be in between the human operator and the aerial robots including its workspace.
- The human operator must be able to perceive a physical/virtual 1:1 representation of the workspace at sufficient detail.

At CLOSE distance, all other sensory perceptions of the human are enabled. Besides of visual perception, this potentially includes sense of hearing (e.g., noise of propellers engines in terms of an aerial robot), tactile sense (considering touch-based interaction [98] or feeling the wind of the propellers), or even the olfactory sense (Gonzalez et al. [112]).

3.4.2 Classification as MEDIUM Perceptual Distance

For use cases classified as MEDIUM, we consider any occluding physical geometry which hinders the human operator to directly perceive the aerial robot. Further, we assume that the human operator is inside the close phase of the public distance related to proxemic notations. Again, we require a 1:1 coherence of the physical and virtual scene. It is remarkable, that the human operator is physically close to the aerial robot, although the sensory perceptions (e.g., sense of hearing) are partly, however not fully, disabled.

3.4.3 Classification as FAR Perceptual Distance

We classify interaction at perceptual FAR distance, if one of the afore mentioned conditions is not fulfilled. This means that either there is a strong occluder between human operator and aerial robot, or, the physical distance is out of proxemics, which requires classification as FAR distance, independent of presence or properties of any potential occluder. In addition, we require a 1:1 coherence between physical and virtual world for CLOSE and MEDIUM distance; at FAR distances, this coherence is not given.

4 Experimental Platform

Contents

The purpose of this chapter is to introduce the experimental aerial robotic platform that was used for the contributions discussed in Chapter 5, Chapter 6 and Chapter 7. The goal was to provide a low-weighted and scaleable aerial robotic platform which in order was called the *SLIM* (**S**caleable and **L**ightweight **I**ndoor-navigation **M**AV). Amongst others, the core design aspects of the *SLIM* were:

- Scalable enough to carry different types of sensors but also information-output devices like laser-projectors to create visual displays in the environment. The ultimate goal was to increase users task-efficiency, when teleoperating the *SLIM*, by means of visualization and natural interaction techniques.
- Low-weight design to provide adequate safety, enabling close flight to human users.
- Ability to robustly operate indoors, using self-localization based on an external motion tracker.

Amongst the different contributions which are namely the Mirco Aerial Projector (MAP) [113] and the Drone Augmented Human Vision (DAHV) [114], the *SLIM* platform underwent several software and hardware-upgrades resulting in different implementations (flavors) of the system. Details are also outlined in Section 4.3 and Section 4.4 of this chapter, whereas in the following we summarize most important aspects of the *SLIM*.

Indoor navigation with micro aerial vehicles (MAVs) is of growing importance nowadays. State of the art flight management controllers provide extensive interfaces for control and navigation, but most commonly aim for performing in outdoor navigation scenarios. Indoor navigation with MAVs is challenging, because of spatial constraints and lack of drift-free positioning systems like GPS. Instead, vision and/or inertial-based methods are used to localize the MAV against the environment. For educational purposes and moreover to test and develop such algorithms, since 2015 the so called *droneSpace* was established at the Institute of Computer Graphics and Vision at Graz University of Technology. It consists of a flight arena which is equipped with a highly accurate motion tracking system and further holds an extensive robotics framework for semi-autonomous MAV navigation. A core component of the droneSpace is a Scalable and Lightweight Indoor-navigation MAV design, which we call the *SLIM* (A detailed description of the SLIM and related projects can be found at our website: https://sites.google.com/view/w-a-isop/home/education/slim). It allows flexible vision-sensor setups and moreover provides interfaces to inject accurate pose measurements form external tracking sources to achieve stable indoor hover-flights. In the following, we present capabilities of the framework and its flexibility, especially with regards to research and education at university level. Further, use-cases from the project UFO [8], but also from students courses at the Graz University of Technology, are summarized. A discussion about potential future work on the platform as an outlook is discussed in Section 8.1 in Chapter 8.

4.1 Introduction

During the past decade, research and development on aerial robotic platforms, especially micro aerial vehicles (MAVs), became increasingly popular. With different types of physical MAV setups and underlying control framework, state of the art applications widely aim for autonomous, or at least semi-autonomous navigation. Though, commercial off-the-shelf systems are most commonly designed for acting autonomously in outdoor environments, whereas typical use-cases involve agriculture [115], parcel-delivery [116] or even search and rescue-missions [117]. Clearly, such use-cases put different demands on the MAV, compared to indoor flights. In outdoor environments spatial constraints play a subsidiary role and very often GPS-based localization and navigation is possible. Only few MAV platform designs exist which are focusing on indoor navigation mainly and more importantly even provide access to the appropriate interfaces to enable drift-free localization and navigation in indoor environments. This is especially important for educational purposes, if development and testing of vision based tracking or reconstruction algorithms is required. Moreover, the *SLIM* was extended with projection devices to serve as highly mobile visualization platform. To this purpose, an indoor flying-arena for MAVs, including physical MAV design and software-framework, was established at the Institute of Computer Graphics and Vision (ICG) at Graz University of Technology. Besides of serving as a framework for the research contributions in Chapter 5, Chapter 6 and Chapter 7, the

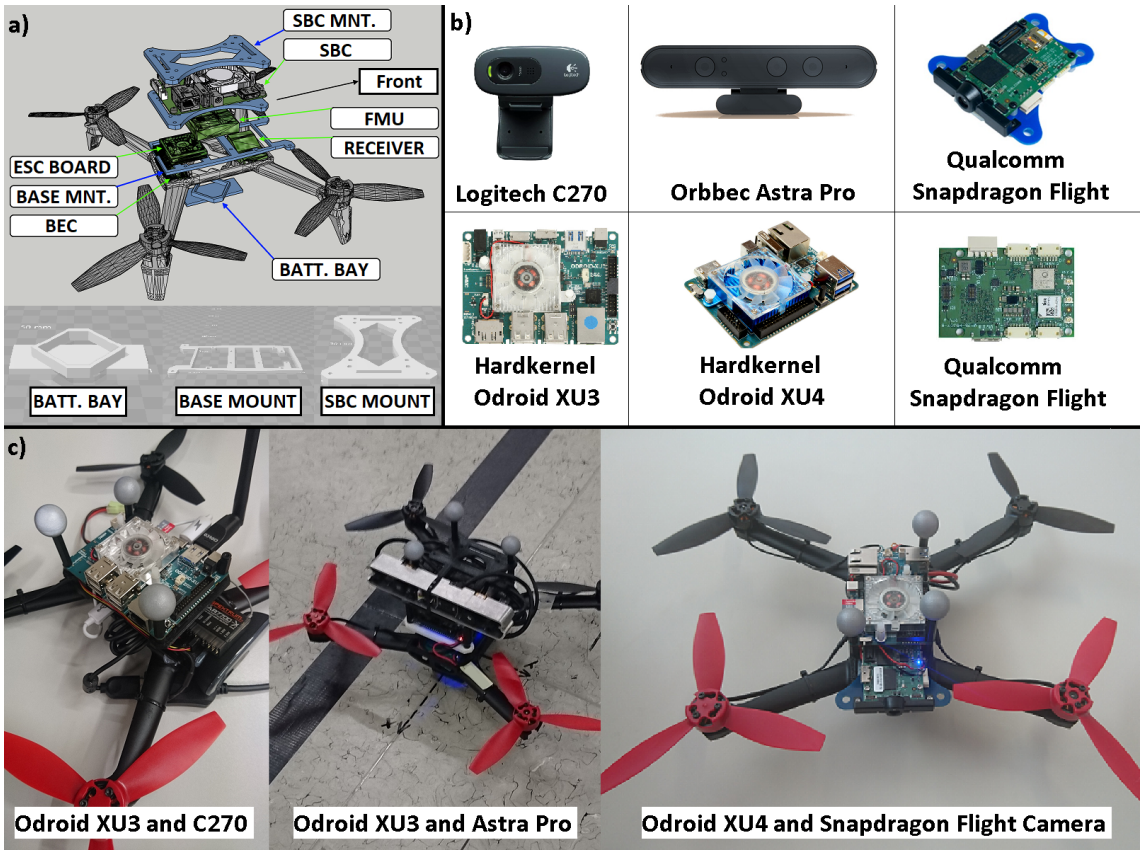


Figure 4.1: Overview of the *SLIM*. a) Principle layout of main components. Custom mounts are shown in blue and electronic components are represented in green colors. b) Utilized sensors and single-board-computers. c) Overview of different sensory setups (monocular camera, RGBD-sensor [120] and high-resolution monocular camera integrated into the Snapdragon Flight).

goal was to achieve a scalable educational platform for indoor flights, where Master and Bachelor-students are able to gain first hands-on experience in terms of MAVs. In the following, we present main design aspects of the *SLIM*, while we give details about the implementation of hardware and software. The MAV design is based on the PIXHAWK [118] open-source flight-controller and able to hold multiple vision-sensor configurations. For educational purposes, also drift-free pose measurements from a highly accurate motion tracking system can be used to localize the MAV against the environment. The physical MAV setup is complemented with an extensive ROS-framework [119] which serves for high-level control of the MAV and visualization of important data. In this chapter, an overview of the current capabilities of the platform is given. It was successfully used for indoor flight, also providing tracking-based navigation and online reconstruction of the environment during lecture courses.

4.2 The SLIM

In this section we present motivation and main aspects, which were considered when designing the *SLIM* as a research- and education-platform. Besides of achieving the challenging task of enabling indoor flight in a flexible setup, also easiness of use by unexperienced students played an important role. Additionally, the goal was to make experimental flights as safe as possible.

4.2.1 Design-aspects and Constraints of the physical MAV-Setup

Considering the general constraints of MAVs with regards to limited payload and flight times, finding a well balanced design is challenging. On one hand, larger MAVs with higher-all-up weight are capable of carrying more payload and typically have increased flight-times. On the other hand, their produced thrust, size of propellers and outer dimensions increase accordingly. Very often this is not practical, specifically if flights in more narrow indoor environments is concerned. Moreover, if flying close to humans is required or unavoidable for educational purposes, too large platforms could be too dangerous for practical use.

In general, well-selecting appropriate components for the given design requirements was challenging, since flexibility and maximum accessability of the individual interfaces excluded any off-the-shelf solution.

Platforms like the well known M100 from DJI or smaller sized MAVs like the DJI Mavic 2 [121] are capable in terms of flight time, easiness of steering and quality of the attached monocular vision camera. In addition, the M100 is provided with an extensive ROS framework, whereas the Mavic 2 is provided with a mobile SDK to enable development of custom applications. Still, with a diagonal wheelbase of 650 mm and 2.4 kg of all-up weight the M100 is considerably larger and was not assessed as practical for the indoor flights in narrow space. In comparison, the Mavic 2, with a diagonal wheelbase of 354 mm and 907 g of all-up-weight, is of smaller size but unfortunately does not provide a ROS API-directly, which would make it considerably more handy for educational purposes. In addition, the Mavic 2 is not easily hardware-customizable regarding more capable vision sensors like RGBD.

Additionally, there exist a variety of very small sized off-the-shelf platforms like the DJI Spark [122] or the crazy-fly from BitCraze [123], whereas their design aims for frame-wheelbases about the size of a humans hand. However, these platforms do not provide more powerful vision-based onboard sensors like RGBD and there is no easy access to hardware and flight controller functions. Also taking into account the very limited payload in general, this ultimately makes it more difficult to achieve position-stabilized hover-flight or environmental reconstruction in indoor environments.

To address the afore discussed problems, the main design goals for the *SLIM* were specified as follows:

- **Scaleable components** in terms of computational hardware and sensors, due to changing context of potential lectures or students demands. This involves extension by more capable vision sensors, like monocular cameras or even RGBD-sensors, to also enable object tracking or online reconstruction of the environment during flight.
- **Easily accessible** hardware components to increase easiness of setup, use and maintenance in case that components need to be replaced.
- **Open- and accessible** hardware interfaces. In addition the according interfaces should be open, meaning custom programmable, as well. For example, regarding the flight-management controller, injecting external pose measurements and high/low-level control commands are vital. The purpose is mainly to achieve a robust, position-controlled hover-flight indoors, but also let students touch the underlying structures of the flight management controller.
- **Open-source compatible** software frameworks should be used if possible, to increase easiness of use.
- **Lightweight** setup, with a maximum all-up weight below $700g$ to further increase safety and reduce turbulence in close proximity flight. The all-up weight includes any sensors for stabilized hover-flight and additional vision sensors for online environmental mapping. For details also refer to Section 4.2.1.4.
- **Acceptable flight-times** of $10min$ minimum, to provide enough time for experimenting during a lecture. This must be valid with an all-up weight, including any sensors for stabilized hover flight and additional vision sensors for online environmental mapping.
- **Safe operation** by using low-thrust engines and softer (smaller) propellers if possible. Additionally, possibility to attach safety guards to the frame and utilizing the safety-features of the flight-controller.

4.2.1.1 Scalability

To provide maximum scalability in terms of supporting multiple sensor-types and according computational hardware, a middle-sized frame design from off-the-shelf hardware was selected. The frame setup is based on the Parrot Bebop 2 [124] and has a diagonal wheelbase of 335 mm, whereas it is extended by custom mounts (Figure 4.1a) to hold various sensor types and single-board-computers (SBCs), shown in Figure 4.1b. The sensor-systems and small-sized computers, which were attached to the *SLIM* and also successfully used during lectures are listed in the following:

- Logitech C270 Pro Monocular Camera
- ORBBEC Astra Pro RGBD-sensor

- Snapdragon Flight Monocular Camera

whereas an overview of the used SBCs is listed below.

- Hardkernel Odroid XU3
- Hardkernel Odroid XU4
- Snapdragon Flight onboard SBC

An overview of the equipped sensors and available SBC configurations is also given in Figure 4.1b.

4.2.1.2 Easy-To-Access Hardware Components

The basic layout of the individual components to achieve stable hover flight is centered around the Bebop 2 frame. Use of the Bebop 2 frame is motivated in the following subsections, whereas a zoomed overview of the components placement in principal is shown in Figure 4.2. The overall goal was to make as many components accessible for replacement or repair. As a result the *SLIM* was designed based on multiple component layers, which are arranged in a stack-like structure. For example, if access to the receiver or battery emitting circuit (BEC) inside the frame is required, only the base mount has to be removed. The base mount is simply "clipped" into the stock base-frame and can be quickly removed. If connector cables of the engines are unplugged, it is further possible to remove all layers above the base-frame including ESC, Flight Controller, RC-Receiver, SBC and potentially connected vision sensors. The individual layers are separated by the custom mounts and connected via hexagonal plastic spacers and screws. Details about the assembly of the *SLIM* in a full RTF configuration are outlined in Section 4.2.5.

4.2.1.3 Open Hardware Interfaces

Access to hardware interfaces was of great important for our design, especially with regards to research and education. In general it can be stated, that closed off-the-shelf components might provide basic and robust functionality until some degree. Though, there are clear drawbacks in terms of letting students gather knowledge and experience, as customizable interfaces enrich educational and research potential. To this purpose, the main hardware components like Flight Controller, SBC and sensors were selected so that they provide maximum access to interfaces on hardware-level.

Connected to this, another important role plays open-source compatibility, because of its great flexibility and richness available solutions and support from the community. Accordingly, also the software frameworks were selected, whereas details are outlined in Section 4.2.3.

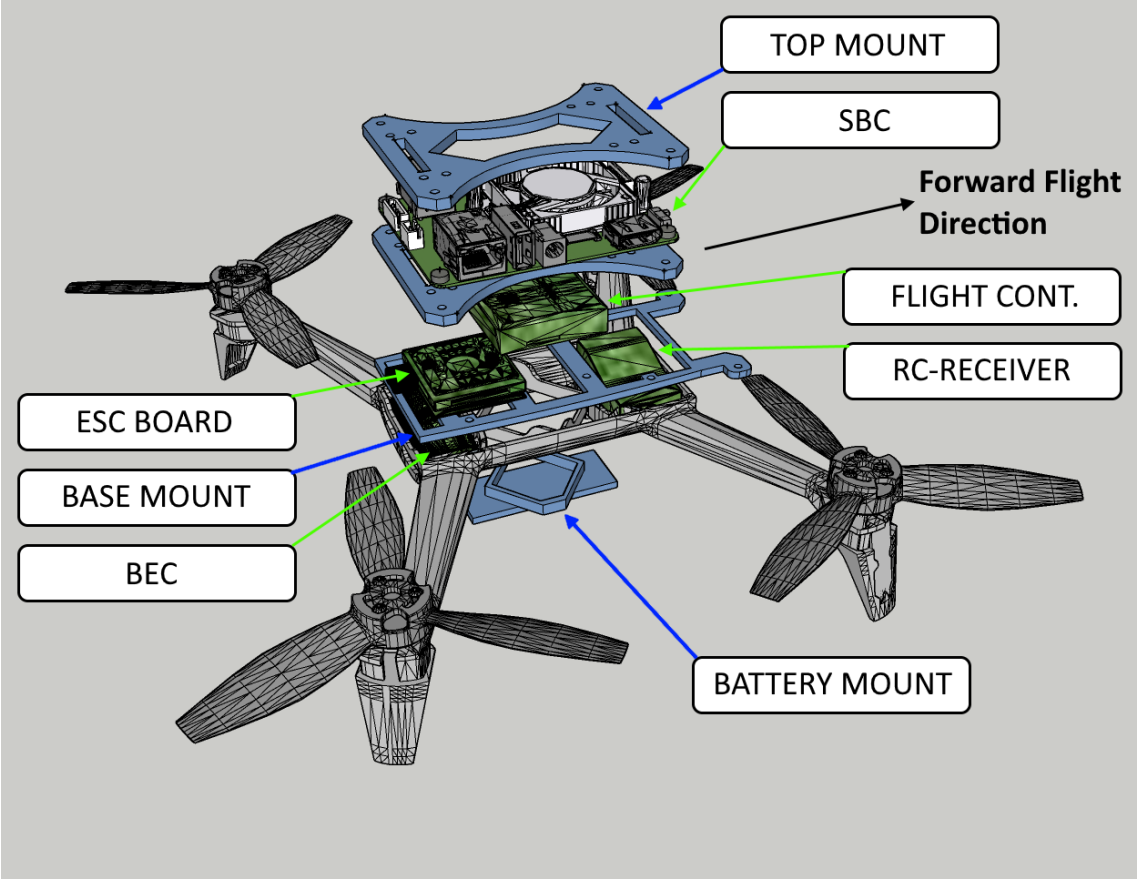


Figure 4.2: Principle layout for the placement of main components of the *SLIM*. Custom mounts are shown in blue, whereas electronic components are represented in green colors. The legs attached to the base frame are optional for the RTF setup.

4.2.1.4 Lightweight Setup

For safety reasons, especially when flying close to unexperienced students, designing the MAV as lightweight as possible, was a crucial aspect. On the other hand, like discussed in Sect. 4.2.1, a too small-sized MAV design significantly reduces its capabilities. In addition, given a constrained flight-height, the maximum flight velocity plays an important role. For better comparison, Falanga et al. [125] and Loianno et al.[39] presented aggressive MAV flight maneuvers with a max. velocity of up to 3.0 $\mathrm{m} \cdot \mathrm{s}^{-1}$ and 4.5 $\mathrm{m} \cdot \mathrm{s}^{-1}$ respectively. Whereas the design of Falanga et al. includes an all-up-weight of 0.83 kg, Loianno et al. introduce a lightweight design below 250 g and aim for similar capabilities like the *SLIM*, although both works include a monocular vision camera only. From design perspective, achieving online mapping during flight based on the richer RGBD-sensor data is still challenging in the weight-category below 250 g. As a result trade-offs have to be considered. To first of all define a baseline for weight- and velocity constraints, the maximum energy of MAVs for outdoor use was selected under consideration of the aviation law in Austria.

According to the Austrian Aviation Act [126], the maximum overall energy of a MAV during operation must not exceed

$$E = E_{kin} + E_{pot} \leq 79 \text{ J} \tag{4.1}$$

before it is subject to authorization. Since the height of the tracked flying space is below 3 m and considering the discussed works, the maximum MAV velocity during experimentation was defined to not exceed $v_{max} = 5.0 \text{ m} \cdot \text{s}^{-1}$. In addition a weight of 0.7 kg was defined as a competitive maximum all-up-weight, compared to the afore mentioned works. Based on this weight constraint a maximum tolerable flight-velocity $v_{tol.}$ was calculated for better comparison and is given in Equ. 4.3.

$$E_{kin} + E_{pot} \leq 79 \text{ J}, E_{kin} \leq 79 \text{ J} - 0.7 \text{ kg} \cdot 9.81 \text{ m} \cdot \text{s}^{-2} \cdot 3.0 \text{ m}, E_{kin} \leq 58.399 \text{ J} \tag{4.2}$$

whereas the maximum flight velocity is then given with

$$E_{kin} = 0.5 \cdot m \cdot v^2 \Rightarrow v_{tol.} \leq \sqrt{\frac{2 \cdot E_{kin}}{m_{AUW}}} \leq \sqrt{\frac{2 \cdot 58.399 J}{0.7 \text{ kg}}} \leq 12.917 \text{ m} \cdot \text{s}^{-1} \tag{4.3}$$

Regarding safety considerations, the *SLIM*'s maximum tolerable flight-velocity $v_{tol.}$ (Equ. 4.3), under the given weight constraint, was estimated to be far above the defined maximum of $v_{max} = 5.0 \text{ m} \cdot \text{s}^{-1}$.

As a result of the maximum all-up-weight and to start from a basic working setup, a semi-customized design was established. In stock configuration, the off-the-shelf Parrot Bebop 2 MAV [124] can provide flight times of up to 25 min and has an all-up weight of 0.5 kg. Thus, it is far below the required maximum of 0.7 kg regarding the need of mounting additional computational or sensor-units. Further, its frame and propellers provide enough stiffness and comparably low weight (43 g and 3 g each), compared to other carbon fiber frames of even smaller size (e.g. 190 mm-DroneArt MAV X-Frame with 26 g [127]). In addition, the frame provides enough hollow space in the center to potentially hold smaller hardware components. Moreover, its cheap price and availability makes it an attractive base for a semi-customized setup.

Based on the maximum all-up weight of $m_{AUW} = 0.7$ kg, in the following also the maximum flight-times should be discussed. As basic requirement for educational- but also for research purposes, a minimum of $t_{hover} \geq 10$ min hover-flight time was defined. To maximize flight-times, an efficient battery type was selected considering the energy to weight ratio. Evaluation and comparison of a wide range of battery types resulted in selection of the "DroneTec HD Power Battery For Parrot AR Drone 2.0". As it provides 3 cells (same as the stock Bebop 2), 2300 mA $\cdot$ h of capacity and due to its small size (71 x 37 x 34 mm) and low weight (146 g), it was selected as the best fit. Tbl. 4.1 shows a comparison of this battery type against other common types, typically used for MAVs at the scale of the *SLIM*. The given ratio divides the batteries energy by its weight, whereas

higher values indicate higher efficiency.

Type	Cells (S)	Energy (Wh)	Weight (kg)	Ratio
Turnigy Type 1	3	$2.2Ah \cdot 11.1V = 24.42$	0.188	129.89
Turnigy Type 2	3	$2.2Ah \cdot 11.1V = 24.42$	0.204	119.71
Turnigy Type 3	4	$2.2Ah \cdot 14.8V = 32.56$	0.247	131.82
DroneTec Type 1	3	$1.5Ah \cdot 11.1V = 16.65$	0.11	151.37
DroneTec Type 2	3	$2.0Ah \cdot 11.1V = 22.2$	0.141	157.45
DroneTec Selected	3	$2.3Ah \cdot 11.1V = 25.53$	0.146	**174.86**

Table 4.1: Comparison of the selected DroneTec battery against similar types.

Considering these boundary conditions, in order the the hover-flight time was estimated. In general, the hover-flight times of a Lithium-Polymer (LiPo) battery powered MAV, can be calculated based on the following parameters.

- C defined as the battery capacity given in mAh.
- V_n defined as the nominal battery voltage given in V. Considering LiPo batteries, the nominal voltage is defined with the cell voltage ($V_c = 3.7V$) times amount of cells S, thus $V_n = S \cdot V_c$.
- P_m defined as the electrical power given in W, required by one engine to lift a quarter of the total weight ($m_{AUW} = 0.7$ kg) of the MAV, since the *SLIM* setup includes 4 engines in X-configuration. It is further assumed that all 4 engines and props are identical and the center of gravity is roughly situated in the middle of the used frame. Beforehand an estimate of $P_{m,est.}$, was calculated in Equ. 4.4, whereas as a proof of concept a true value P_m was empirically measured during hover flight, assuming the worst case of $m_{AUW} = 0.7$ kg.
- P_e is defined as the electrical power given in W, which is required by the electronic components of the *SLIM* As a worst case assumption, P_e was estimated as the maximum power consumed by the SBC ($P_{SBC} = 5V \cdot 4A = 20$) and the Flight Controller ($P_{SBC} = 5V \cdot 1.0A = 5W$). An addition of $1.0A$ at $5V$ supply level was considered as safety margin ($P_{margin} = 5V \cdot 1A = 5W$).
- η can be defined as an efficiency-factor that takes into account energy losses from various sources. One major source is to keep the LiPo battery voltage level above a certain threshold when discharging. Another important source are losses from wires and electronic components. As a rule of thumb η is most commonly estimated with 0.8 for typical MAV configurations.

A crucial aspect for estimating hover-flight times is P_m, since it is the required power of each engine to let the MAV hover. An estimate $P_{m,est.}$ for one single engine given in

Watts can be derived based on a mathematical thrust-model for the static case [128], and is given in the following equation.

$$P_{m,est.} = K_{ad} \cdot \frac{\sqrt{F^3}}{r} = 0.3636 \cdot \frac{\sqrt{6.867^3}}{0.0762} = 10.7321 \text{ W} \tag{4.4}$$

whereas $K_{ad} = 0.3636$ is the air-density coefficient given with the nominal value under the assumption that the air pressure is 1atm and the room temperature is at 20 °C. $F = m \cdot g = \frac{1}{4} \cdot 0.7 \text{ kg} \cdot 9.81 \text{ m} \cdot \text{s}^{-2} = 6.867 \text{ N}$ is the thrust required for hover expressed in Newtons and $r = \frac{1}{2} \cdot 6 \cdot 0.0254 = 0.0762 \text{ m}$ is the radius of the propeller considering the 6inch propellers of the Bebop 2 model.

Noticeable is a deviation of true measured power (P_m empirically measured with 12 W) from the estimated power, required for hover-flight. Typically they are a result of unmodeled power losses, which can occur due to non-perfect stiffness of rotor-blades, efficiency of engines, power-electronics and the fact that in our model the propelled air follows a perfect cylindrical shape, which is not true in the real-world case. Although it is mentionable that deviations were in an acceptable range still, also considering resulting hover-flight times. Finally, a summary of the remaining parameters and their relation to the values used in the physical setup is given in Tbl. 4.2.

Parameter	**Value**	**Unit**
C	2300	$\text{mA} \cdot \text{h}$
V_n	11.1	V
P_m	12	W
P_e	30	W
η	0.8	n.a.

Table 4.2: Parameters and values for estimation of hover-flight time.

Based on the parameters and values listed in Tbl. 4.2, the estimated hover-flight time of the *SLIM* is then given with

$$\begin{aligned} t_{hover} &= \eta \cdot \frac{60}{1000} \cdot \frac{C \cdot V_n}{4 \cdot P_m + P_e} \\ t_{hover} &= 0.8 \cdot \frac{60}{1000} \cdot \frac{2300 \cdot 11.1}{4 \cdot 12 + 30} \\ t_{hover} &= 15.711 \text{ min} \end{aligned}$$

As a result, the estimated flight-time was considered to be above the required minimum of $t_{hover} = 15.711 \text{ min} \geq 10 \text{ min}$.

4.2.1.5 Safety

Concerning safety, the *SLIM* also provides basic features enabling operation and close-flight to unexperienced persons. One major aspect are the propeller guards (Tbl. 4.3, ID19)

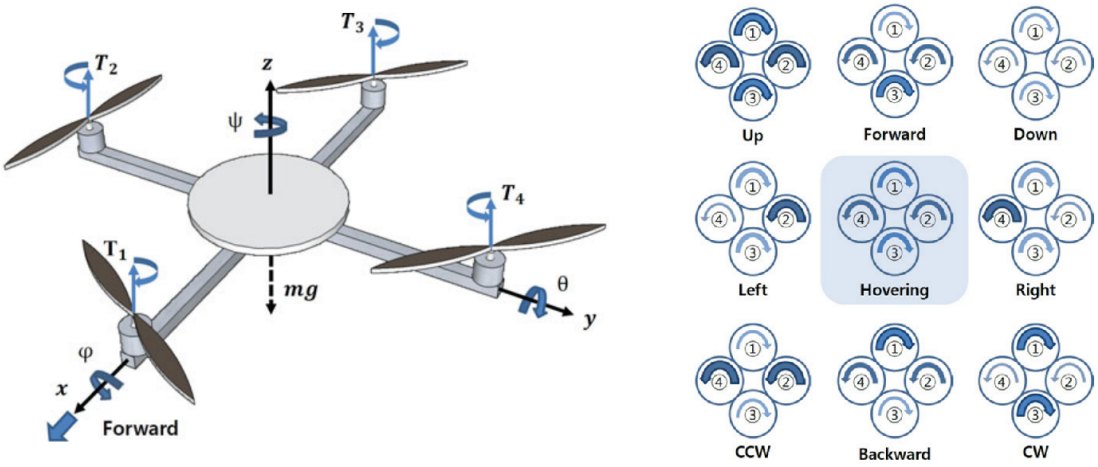


Figure 4.3: MAV configuration with a freed body diagram and schematic overview of motions resulting from different thrust configurations [129].

which were also customized and reduced in weight for the *SLIM* design. In addition, the PIXHAWK flight controller offers a safety-switch which enables the MAV's engines only after they were manually armed by an operator.

4.2.2 Modelling And Control of the MAV

The *SLIM* is setup in a common 4 propeller X-configuration including 4 engines that are individually positioned at the end of the stock Bebop 2 frame. The direction of motors is set in such a way that a pair of motors rotate clockwise while the other pair rotates counter clockwise as shown in Figure 4.3. This arrangement of motors is set in order to generate vertical lifting force to raise the MAV up in the air. To better understand the model, discussed in the this section, Figure 4.3 outlines two frames of reference which are the fixed world-frame (F_W) and the relative body frame of the *SLIM* F_B. The transformation of the *SLIM*'s body frame F_B relative to the world frame F_W is defined with ${}^{W}T_B$. All frames are right handed with the x-axis pointing in forward-flight direction and the z-axis pointing upwards in opposite direction of earth's gravity vector. They are defined to in order derive the equations of motion for a 6 degree of freedom (DOF) configuration of the *SLIM*.

4.2.2.1 Degrees of Freedom

Like described by Choi et al. (Figure 4.3), the *SLIM* in the presented configuration can be controlled via angular movements along 3 different axis, which are roll-angle θ (rotation around the x-axis), pitch-angle ϕ (rotation around the y-axis) and yaw-angle ψ (rotation around the z-axis). In order, the following in-air maneuvers can be performed by the MAV: hovering, pitching forward or backwards, rolling left or right and turning around

the z-axis. Hovering can be achieved if all engines turn at the same speed and in order produce the same vertical thrust. Roll- and pitch movements can be achieved if the speed of one pair of motors is changed, while the other pairs turning speed remains constant. Turning around the z-axis is achieved by altering the speed of the two pairs of engines. The orientation angles θ, ϕ and ψ are expressed as Euler angles.

4.2.2.2 Model of the MAV

The model of the MAV can be defined with the equations of motion. They are expressed as Newton-Euler equations, whereas they reflect the combined translational and rotational dynamics of a rigid body. Assuming a simplified point mass model of the MAV the equations are given in Equ. 4.5 and Equ. 4.6 and reflect the MAV's 6DOF,

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) + \frac{U_2}{J_x}, \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{J_z - J_x}{J_y}\right) + \frac{U_3}{J_y}, \ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{J_x - J_y}{J_z}\right) + \frac{U_4}{J_z} \tag{4.5}$$

$$\begin{aligned} \ddot{z} = \frac{U_1}{m}\cos\phi\cos\theta - g, \ddot{x} &= \frac{U_1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ \ddot{y} &= \frac{U_1}{m}(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \end{aligned} \tag{4.6}$$

whereas J_x, J_y and J_z reflect the inertia terms on the main diagonal of the inertia matrix J, m parametrizes the point mass and g is the earths gravity constant. The equations are simplified, based on Beard et al. [130], whereas coriolis terms and time-derivates of the rotational component of the transformation matrix ${}^{W}T_B$ are neglected. Accordingly, the 4 control inputs U_i of the model can be defined as given in Equ. 4.7.

$$\begin{aligned} U_1 = b\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right), \quad U_2 &= b\left(\omega_4^2 - \omega_2^2\right) \\ U_3 = b\left(\omega_3^2 - \omega_1^2\right), \quad U_4 &= d\left(\omega_4^2 + \omega_2^2 - \omega_3^2 - \omega_1^2\right) \end{aligned} \tag{4.7}$$

whereas ω_i is the angular velocity of each rotor, b is the thrust coefficient and d is the drag coefficient. U_1 can be interpreted as the overall thrust force applied to the MAV along the z-axis in the center of its body-frame F_B. U_2 and U_3 lead to pitch- and roll torques respectively, while U_4 is leading to the torque around the z-axis.

4.2.2.3 Control of the MAV

Since the *SLIM* is designed to fly indoors mainly, turbulences were expected to disturb flight-performance. For sakes of robustness and due to the already existing architecture of the PIXHAWK the integrated linear controllers were utilized to achieve stabilized flight. They consist of nested PID controllers, which can be expressed in general by

$$C_{PID}(s) = K_p + K_i \cdot \frac{1}{s} + K_d \cdot s \tag{4.8}$$

whereas K_p, K_i and K_d are the proportional-, integral- and differential gain parameters of the controller and typically tuned based on the Ziegler-Nichols method [131]. The linear position control approach for the MAV can be separated into altitude control (z-axis) and control of the horizontal movement (x/y-axis). Control of the angular position around the z-axis (yaw-angle ψ) is directly achieved by the PIXHAWK's inner attitude controller, whereas control of x,y and z-position is achieved by the integrated position control loop. Details are discussed in the following.

First of all, the MAV's rotor dynamics play a crucial role, whereas they can be approximated with a first order system, including a linear coefficient K_M and a time constant τ_M resulting from the inertia of the rotors and engines. In the Laplace domain the dynamics are then given with

$$G(s) = \frac{K_M}{\tau_M s + 1} \tag{4.9}$$

Laplace transformation of the translational equations of motion ($\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ in Equ. 4.6) and combination with the linearized rotor dynamics (Equ. 4.9) results in the transfer functions for displacement in x-,y- and z-direction. Different modelling approaches exist here, whereas a common approach is to approximate the transfer functions for displacements by a double integrator combined with first- and second-order systems (Seidel [132], Joyo et al. [133]). Thus, assuming hovering condition ($\frac{U_1}{m} - g = 0 = const.$ with $\phi = \theta = 0$) for vertical displacement and considering a small-angle approximation for the horizontal displacement, we can express the according transfer functions given in Equ. 4.10. Noticeable are the internal dynamics for attitude stabilization G_{xy} for pitch- and roll-angles, which are approximated by a second-order system [134]. The according gain parameters can then be defined with $K_{xy} = g$ and $K_z = \frac{1}{m}$.

$$G_{xy}(s) = \frac{K_{xy}}{s^2}\left(\frac{\omega_0^2}{s^2 + 2D\omega_0 s + \omega_0^2}\right), G_z(s) = \frac{K_z}{s^2}\left(\frac{K_M}{\tau_M s + 1}\right) \tag{4.10}$$

Combining these transfer functions with the PID-control approach expressed in Equ. 4.8, the transfer functions of the closed loop system can then be expressed with

$$T_{xyz}(s) \quad = \quad \frac{G_{xyz}(s) \cdot C_{PID}(s)}{1 + G_{xyz}(s) \cdot C_{PID}(s)} \tag{4.11}$$

4.2.2.4 Experiments for Position Stabilization

For sakes of completeness, performance of the position stabilization of the *SLIM* was evaluated in relation to the model and control approach discussed in the previous section. The setup used for experimenting included the C270 camera and the Odroid XU3, whereas details about the different *SLIM* configurations are shown in Figure 4.1. In a first step, datasets necessary for evaluation and further controller tuning were taken during flight and it is remarkable that the proportional gain of the position control loop of the PIXHAWK in the first step was set to $MPC\_XY\_P = MPC\_Z\_P = 1$. No additional integral

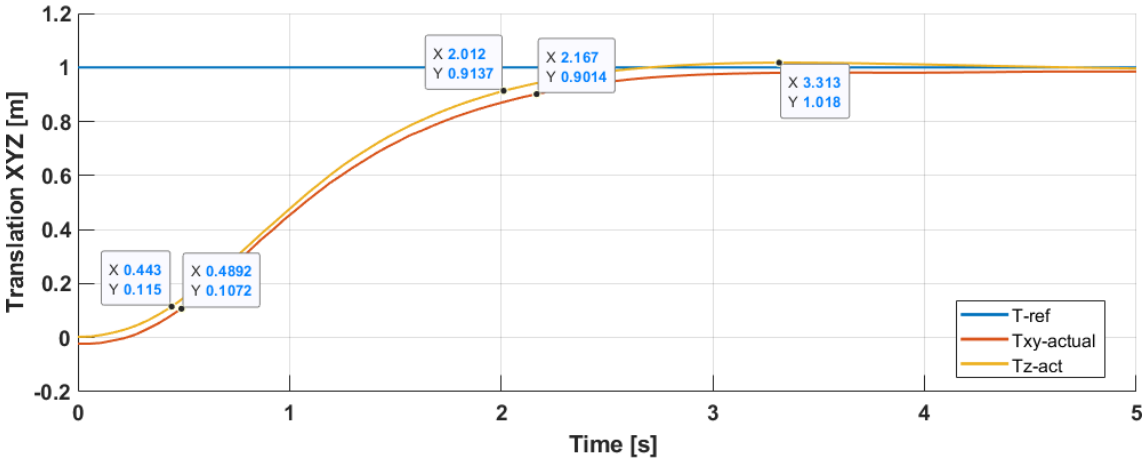


Figure 4.4: Shown are step responses with tuned controller gains for translation in x/y- and z-direction. Remarkable are adequate rise times ($t_{r,x} \simeq t_{r,y} \simeq 1.57$ s and $t_{r,z} \simeq 1.68$), small overshoot ($PO < 5\%$) and sufficiently small steady-state error ($e_\infty < 5\%$).

or derivative gain was used ($K_p = 1, K_i = K_d = 0 \rightarrow C_{PID}(s) = 1$) and all other gain parameters (e.g. for the feed-forward path and state estimator) were left at default configuration. The procedure for taking the datasets is described in the following. First the MAV was commanded to achieve stable hover flight at 1 m with zero heading. After the MAV stabilized around the given setpoint, step inputs with a relative change of 1 m were applied in x, y and z-direction accordingly. Data for the applied position command and resulting actual position of the MAV was recorded at 60Hz. The step responses with tuned controller gains applied to the system described by Equ. 4.11 are shown in Figure 4.4 and in good approximation show PT2 behaviour. They reflect adequate closed loop control performance of the fully modelled MAV, including nested control loops and internal system dynamics.

4.2.3 Design of the Software Framework

The main motivation of the software design was compatibility to the selected hardware components in Sect. 4.2.1, which also provide open and accessible interfaces for the *SLIM* platform. This involves typical components used for research with MAVs, like the open source PIXHAWK flight controller, but also computers with more computational power like the Odroid SBCs from Hardkernel (XU4, XU3). They well support UNIX-based open-source operating systems (Ubuntu), whereas support for commercial operating systems like Microsoft Windows is not as extensive. Consequently, with the aspects of scalability and open-source compatibility in mind, the Robot Operating System (ROS) [119] was selected as state of the art framework. Not only because it is open-source and well supported under UNIX based operating systems, but also because of its rich ecosystem.

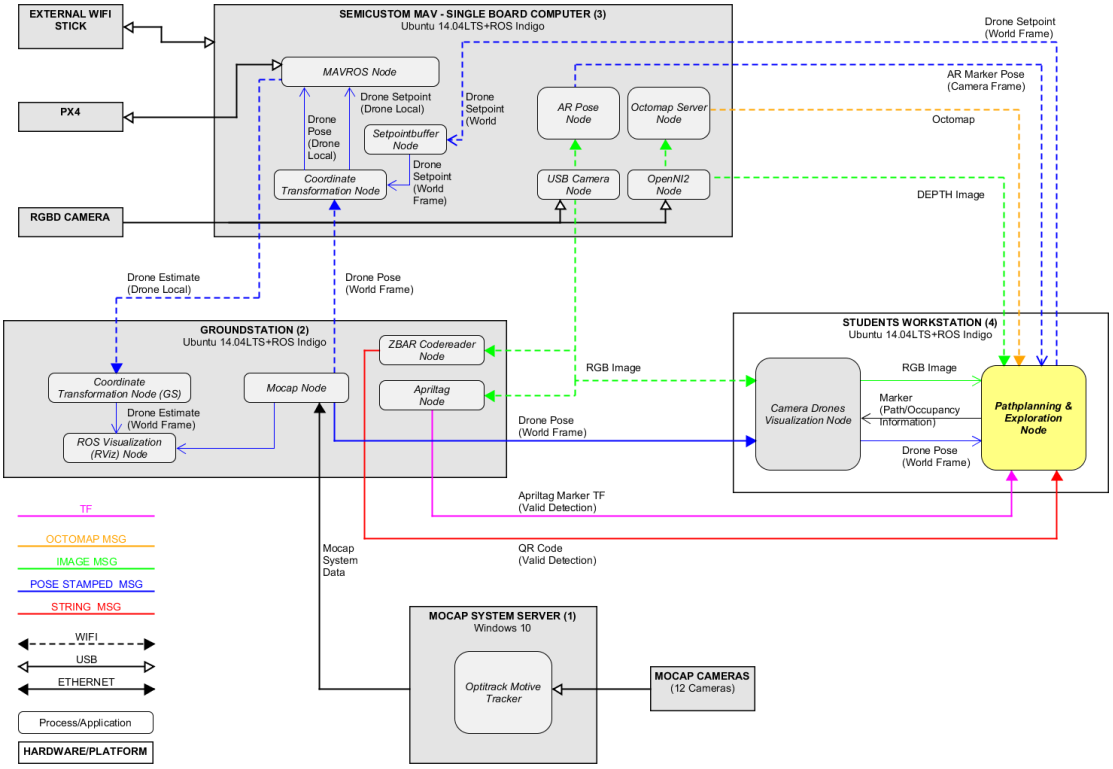


Figure 4.5: Core components of the *SLIM*'s software-framework.

4.2.4 Architectural Overview And Utilized Methods

The main aspect for design of the software architecture was to provide maximum flexibility with regards to research and educational projects. Consequently, a minimum set of core-functionalities were integrated as individual components in a ROS-framework. They are listed in the following:

- Low-Level Flight Control
- Localization
- High-Level Flight Control (Path-Planning, Exploration)
- Environmental Mapping
- Object detection

An overview of the components and the distributed ROS-messages is given in Figure 4.5.

4.2.4.1 Low-Level Flight Control

For low-level flight control, the *SLIM* uses the PIXHAWK as open-source flight management controller. A schematic overview of the control architecture is shown in Figure 4.6.

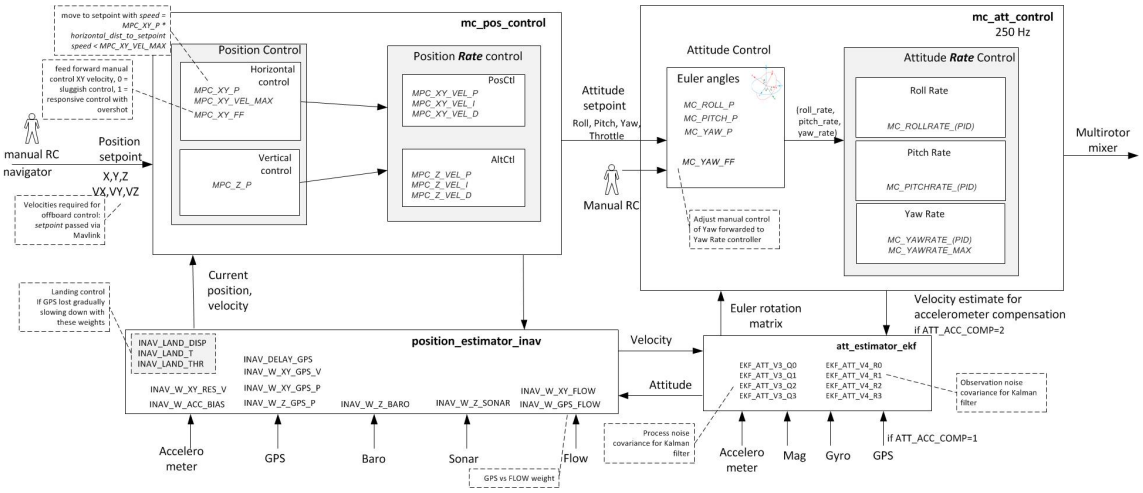


Figure 4.6: Overview of the PIXHAWK's control architecture [118].

For the *SLIM*, the internal attitude-control architecture of the PIXHAWK was utilized, including attitude rate control and attitude control (**mc_att_control**). In addition, the PIXHAWK provides an outer PID position control loop (**mc_pos_control**) to achieve stable hover flight and tracking of a position trajectory respectively.

4.2.4.2 Localization And State-Estimation

For state estimation, the *SLIM* uses the so called INAV state estimator [135] of the PIXHAWK, which fuses attitude estimates based on IMU measurements (**att_estimator_ekf**). Due to the selected flight management controller, it is possible to feed external pose measurements into the state-estimator of the PIXHAWK. The pose measurements are supposed to be drift-free and are typically based on vision methods. Drift-free measurements are vital for the estimation of the MAV's current pose since estimates from purely inertial measurements significantly drift over time, thus making collision-free navigation impossible.

In case of the *SLIM*, two different types of external pose measurements were used, which are pose measurements from an external motion tracking system [136] and pose estimates, derived from RTABMap SLAM framework [137].

4.2.4.3 High-Level Flight Control (Navigation, Exploration)

For high-level flight control, a custom package to generate position trajectories was implemented, whereas a library for cubic spline interpolation was utilized [138]. Additionally, a surrounding PID-position controller was implemented and combined with the inner control architecture of the PIXHAWK to achieve trajectory tracking. A full trajectory control approach, including generation of velocity and acceleration trajectories, was not used for

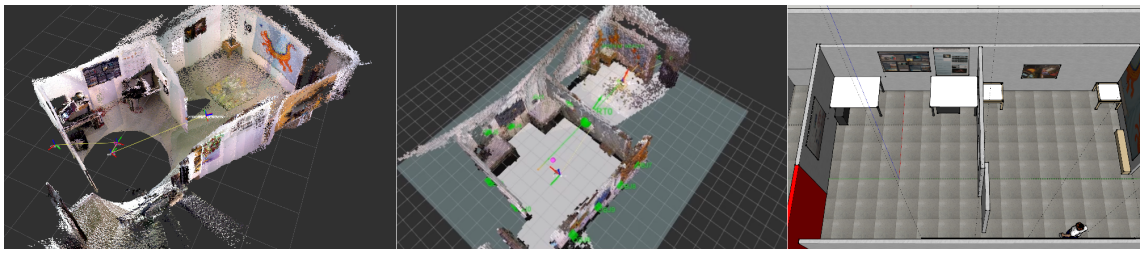


Figure 4.7: Example reconstructions of the droneSpace lab environment. (left) Dense reconstructed point-cloud (middle) Online-reconstructed colored octomap with resolution of $5cm$. (right) CAD model of the indoor environment with realistic textures.

experimentation yet. Connected to this, we outline details about future work on the *SLIM* platform in Section 8.1.

4.2.4.4 Localization and Environmental Mapping

To achieve environmental online-mapping, the *SLIM* platform mainly utilizes the octree-based mapping frameworks which is namely Octomap [139]. The most recent version of the *SLIM* provides also online-mapping and creation of octree-structures onboard (running on the XU3 SBC), while the created octrees are only sent to a groundstation for visualization purposes. The smallest octree resolution that was used during experimentation while still running onboard was $5cm$.

For SLAM-based localization and creation of dense point clouds the *SLIM* platform utilizes the RTAB-Map framework ([140] and [141]), whereas RTAB-Map is also capable of creating colored occupancy grids assuming that RGBD-sensor data is available. Examples of reconstructed point clouds and colored octomaps are shown in Figure 4.7. In addition the framework provides methods to localize the MAV against the created map. Although it has to be mentioned that for the sake of robustness the contributions discussed in Chapter 5, Chapter 6 and Chapter 7 used pose measurement of the external Optitrack tracking system for localization.

4.2.4.5 Object Tracking

To achieve tracking of objects, as part of reserach experiments or educational courses, two different approaches were added to the *SLIM* platform. This on one hand included fiducial marker tracking based on the ARToolkit [142] and furthermore on Apriltags [143]. Typical objects that were highlighted during experimentation included general objects of interest (OOIs). For example, artificial hazardous areas, narrow flight passages or artificial victims.

4.2.5 Implementation Of The SLIM (Basic Version)

Resulting from the design aspects, discussed in Section 4.2.1, the implementation of the *SLIM*, including a detailed part list and assembly instructions are discussed in this section.

4.2.5.1 Firmware and Parameters for Flight-Control

For the presented *SLIM* platform the *v1.3.4 Release* was selected as the PX4 firmware version. The main reason was that up to this version the INAV estimator was well maintained and supported. Besides of robust state estimation, this release in general included major stability improvements [144].

It is noteworthy, that adaptions had to be made to the stock parameters of the firmware to in order achieve acceptable control performance during experimentation. Those parameters are in the category of the attitude-rate controller (MC), multicopter position controller (MPC) and the inertial state estimator (INAV). For sakes of completeness they are discussed in the following.

Control Parameters: In preliminary experiments it was found that the stock configuration for the control parameters of the DJI F330 frame provide acceptable results in terms of control performance. Although, it is important to note that a tuned version of these control settings was used for the works discussed in Chapter 5 and Chapter 6. The following parameters were tuned based on the stock control-parameter configuration for the DJI F330 frame:

- $MC\_PITCHRATE\_P = 0.143$ (increased by 10%)
- $MC\_ROLLRATE\_P = 0.143$ (increased by 10%)
- $MC\_YAWRATE\_P = 0.22$ (increased by 10%)
- $MPC\_XY\_P = 1.25$ (increased by 25%)
- $MPC\_Z\_P = 1.25$ (increased by 25%)
- $INAV\_W\_XY\_VIS\_P = 7$ (increased by 40%)
- $INAV\_W\_Z\_VIS\_P = 7$ (increased by 40%)

State Estimation Parameters: In addition to the control parameters, the following INAV parameters were adapted and **set to zero** to reduce sensor noise and are mandatory to achieve stable flight indoors:

- Gain for barometric height measurements: $INAV\_W\_Z\_BARO$
- Gain for sonar height measurements: $INAV\_W\_Z\_SONAR$
- Gain for GPS measurements: $INAV\_W\_Z\_GPS$

- Gain for sonar height measurements: $INAV\_W\_Z\_BARO$
- Weight for Magnetometer-based attitude estimation: ATT_W_MAG

Setting the magnetometer weight to zero is important, since indoors the heading estimation could suffer from serious noise. This as a result from surrounding ferro-magnetic objects. In the same way, measurements from the barometric sensor can significantly suffer from noise if flying indoors, due to turbulences close to ground.

4.2.5.2 Assembly Instructions

In the following detailed instructions for assembly of an RTF configuration are discussed. The according parts list with the part ID is given in Tbl. 4.3.

Engines (ID6), Propellers (ID5) and connector cables are mounted with the according fasteners in stock configuration to the frame.

Starting from the bottom, the *SLIM* is designed to land and takeoff on the attached battery. This has two advantages as it saves additional weight of the frames legs and also makes it easily attach-/detach-able for recharging. The battery (ID10) is fixed to the custom battery-bay mount (ID2), part of the Bebop 2 frame (ID1), via a Velcro-Fastener (ID18).

As the Bebop 2 frame provides a hollow space in the center, it makes placement of smaller sized components, like the SBEC (ID13) and the RC-Receiver (ID9), more efficient.

On top of the stock Bebop 2 frame, we attach another custom extension mount (ID3). It is responsible for holding the ESC (ID7), the Flight Controller (ID12) and the SBC (ID13). The ESC is mounted with screws (ID17) at the backside of the frame, whereas the Flight Controller is mounted in the same way and centered in the middle. The SBC is mounted above, via 4 hexagonal spacers (ID18). Finally, the SBC is attached with a serial connection to the flight controller (ID16), a WiFi link (ID14) and one additional vision sensor via USB.

The third custom mount (ID4) is optional and able to hold the additional vision-sensor, for example an RGBD-Sensor (ID15).

4.2.5.3 Parts List

This section discusses a full RTF setup of the *SLIM*, including the Hardkernel Odroid XU3 single-board-computer and the ORBBEC Astra sensor configuration to achieve online environmental mapping during flight. The full parts list, with according weights, required quantity and availability for ordering, is shown in Tbl. 4.3. The presented setup can be estimated with an overall weight of approx. $603.5g$ which is below the required maximum weight of $700g$, including safety buffers for potentially more payload.

ID	Description	Type/Brand	Specs	Quant.	Wght.	∑Wght.	www
1	Bare Frame	Parrot Bebop 2	Stock	1	43	43	Link
2	Battery-Mount	Custom	3mm/PLA	1	8	8	—
3	Base-Mount	Custom	3mm/PLA	1	9	9	—
4	Top-Mount	Custom	3mm/PLA	2	12	24	—
5	Propeller	Parrot Bebop	Stock	4	3	12	Link
6	Engines	Parrot Bebop	Stock	4	18	72	Link
7	ESC	Afro 4in1	20A	1	20	20	Link
8	SBEC	Hobbyking	5V/5A	1	18	18	Link
9	RC-Receiver	R6007SP	Sepktrum	1	4	4	Link
10	Battery	AR2.0	3S/2300mAh	1	146	146	Link
11	IR Markers	Motive	19mm	3	3	9	Link
12	Flight Cont.	Pixfalcon	Micro	1	16	16	Link
13	SBC	Hardkernel	XU3	1	45	45	Link
14	WiFi Dongle	Realtek	WN1000	1	2.5	2.5	Link
15	RGBD-sensor	Orbbec	Astra Pro	1	110	110	Link
16	MicroUSB-cable	System-S	11cm	1	10	10	Link
17	Screw	Plastic	M3	4	0.5	2	Link
18	Spacer	Hexagonal	15mm/M3	8	0.5	4	Link
18	Velcro	Fastener	20x50mm	1	1	1	Link
19	Prop-Guards	Custom	PLA	4	12	48	—

Table 4.3: S.L.I.M parts list for physical setup including RGBD-Sensor.

4.3 Implementation For Remote Through-Wall Inspection

A detailed overview of our experimental system architecture which was utilized for the contribution presented in Chapter 6 (including hardware components, software components and data flow between them) is shown in Figure 4.8. The system builds on an earlier version of the SLIM framework, originally described by Isop et al. [113] and is based on six main components:

We use an (1) Optitrack motion tracking system consisting of a server system with 12 cameras to externally localize the drone. The Optitrack is connected to a (2) ground-station, which further communicates to the (3) drone‘s on-board computer, an Odroid XU3, and the (4) HoloLens via WiFi. For our user study, we complemented the system with a remote control user interface including a (5) joypad for steering and a (6) visualization station. All components communicate via Ethernet or WiFi.

The software components are integrated via ROS [119] nodes. We use Unity 3D for visualization on the HoloLens and the ROS tool *RViz* for monitoring on the ground-station.

The motion capturing node on the ground-station relays UDP packages from the Optitrack system, which describe timestamped poses of the tracked objects, to the Odroid. The Odroid transforms the poses into local coordinates of the drone and forwards them to

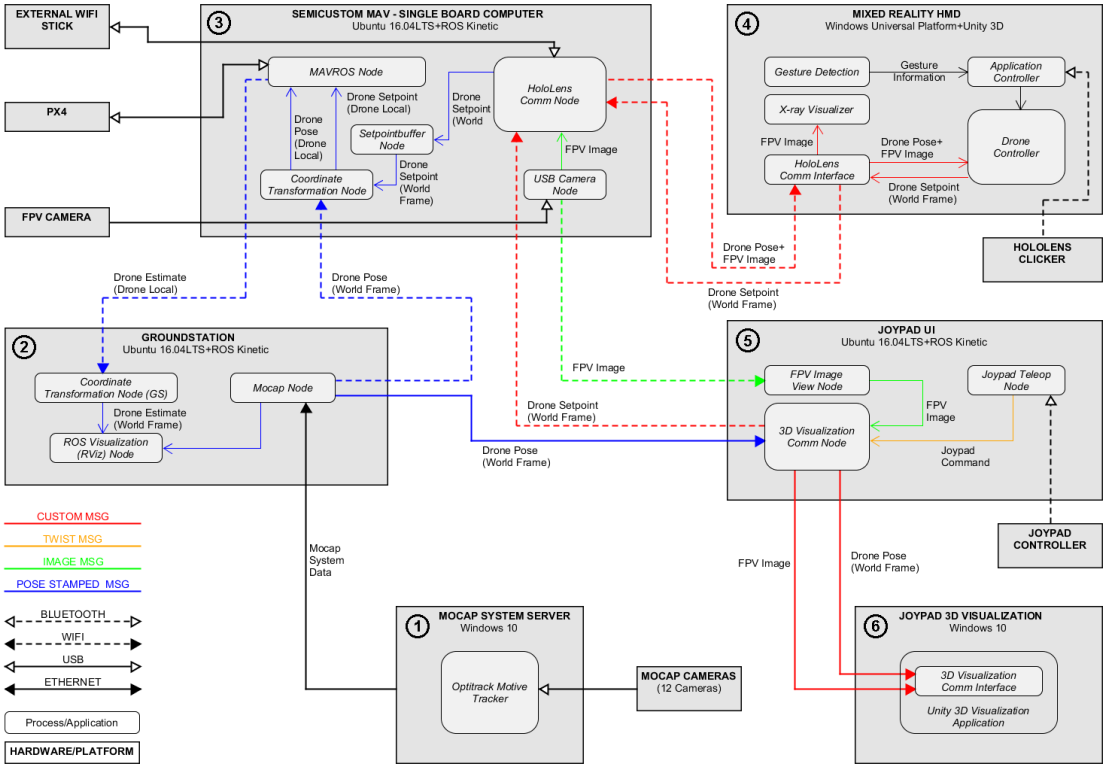


Figure 4.8: Overview of main hardware and software components of our experimental setup including (1) the motion tracking system, (2) a ground-station, the (3) on-board computer of our semi-customized drone, (4) the HoloLens, (5) a joypad interface, and (6) a visualization station.

the MAVROS node, a ROS wrapper to communicate with the PX4 via publish/subscribe messages. It is responsible for acquiring IMU data, pose updates, target coordinates (set-points), internal pose estimates, etc.

The drone controller node on the HoloLens maps gestures into target drone position and visualizes the drone's current position and target positions in the mixed reality view. Setpoints can be generated either by the HoloLens interface or by the joypad interface.

4.3.1 Aerial Robot setup

The drone (Figure 4.9), which has a frame with 25cm diameter and weighs 450g, uses a semi-customized design with rotors and frame taken from a Parrot Bebop 2 platform. The flight time is about 11-15 minutes, while running all relevant components and tasks. We added a PX4 Pixfalcon autopilot as a low-level flight control unit and an Odroid XU3 single-board processor computer.

The forward-looking camera captures image data at 30Hz with 640x480 resolution and delivers it to the Odroid via USB. The video is streamed to the HoloLens in MJPEG format, annotated with timestamp and camera poses to allow precise image-based rendering.

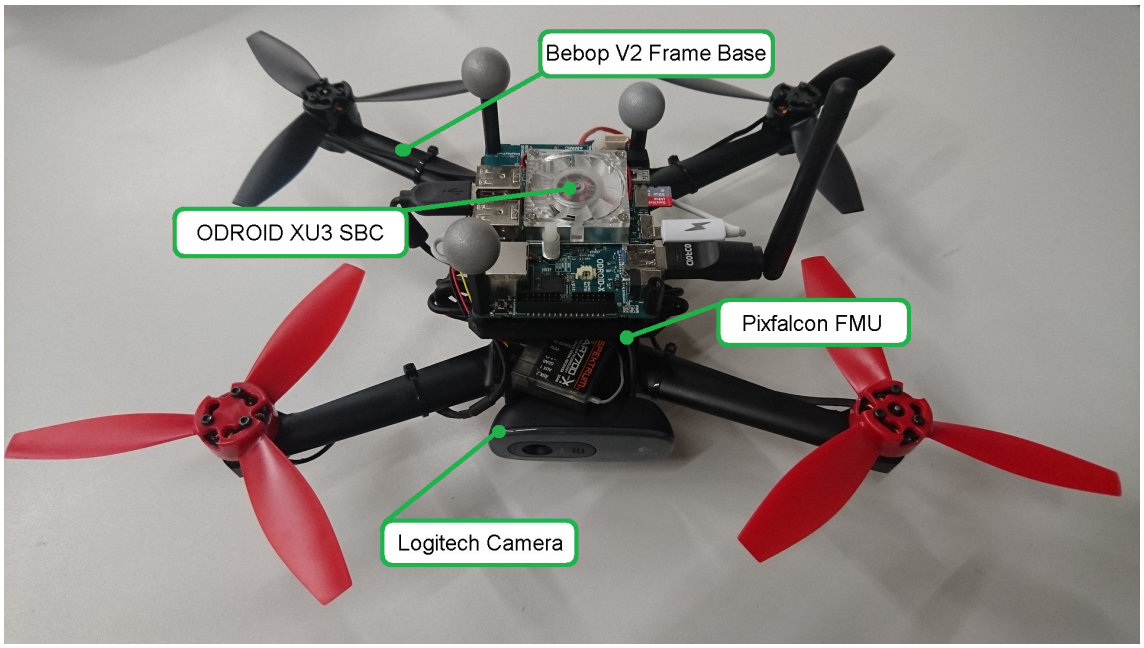


Figure 4.9: Experimental drone setup including the main components. The camera is mounted facing forward. Inside the frame, the autopilot is located. The battery is mounted on the bottom to balance weight distribution. The onboard computer is located on top.

All high-level tasks, including processing of image data, estimated poses from the motion tracker and control commands received from the pilot run on-board and are implemented in the ROS framework.

4.3.2 Flight management control

For localization of the drone, we use the external Optitrack system with 12 ceiling-mounted tracking cameras, covering an area of roughly $5 \times 4 \times 3m$. The Optitrack system provides pose estimations at 120Hz, which are delivered over WiFi to the Odroid at a latency of ˜25ms. The serial link from the Odroid to the PX4 adds another ˜10ms of delay. The system time between Optitrack server, ground-station and PX4 is synchronized based on NTP using the Chrony service.

Designing a drone for autopilot-controlled flight at low heights in small confined spaces is challenging, because of the imminent danger of hitting obstacles. We combined several measures to ensure safe operation. Since high flight speed was not a primary goal, we used low-thrust engines (taken from a Bebob 1 platform) and soft materials for the propellers. This produces less turbulences when flying close to walls and around objects. We further relied on the ability of the PX4 to use pure inertial navigation for short periods, when the measurements from the motion capture system are noisy or intermittent. The pose

updates are buffered on the Odroid to minimize the occasions where the PX4 switches unintendedly from autonomous flight mode into manual mode if the Wifi link stalls or drops position updates from the Optitrack.

4.3.3 Control of the Aerial Robots movements

Control of the drone is based on measuring its 6DOF pose by the motion capture system in world coordinate representation. We make use of the PX4 inertial estimator to fuse the motion capture data with the inertial sensors of the PX4, deriving 3D position $[x, y, z]$ and the yaw θ required for the drones‘s position control. These measurements, obtained at discrete times $i = 0 \dots n$, are denoted as Y_i.

For position control, we use the internal linear control approaches of the PX4. The methods consist of an inner attitude rate PID (proportional/integral/derivative) controller with pitch, roll and yaw angular velocities as inputs. This control loop is enclosed by an attitude P-controller with attitude setpoints for roll, pitch and yaw angles and throttle as reference input. The inner control loop is nested in a position control loop, which takes 3D position $[x, y, z]$ and yaw θ as reference inputs H_i, which can, for example, be derived from the HoloLens interaction. The yaw reference is directly fed into the inner attitude control loop.

$$Y_i = \{x_i, y_i, z_i, \theta_i\} \quad (i = 0...n) \tag{4.12}$$

$$H_i = \{x_i, y_i, z_i, \theta_i\} \quad (i = 0...n) \tag{4.13}$$

$$E_i = H_i - Y_i \tag{4.14}$$

The derived position error, given in Equation 4.14, is calculated in every iteration i and fed into the control structure of the PX4. We use aggressive controller gains, which are based on the default gains of the more heavyweight DJI F330 model, to establish fast response times and accept slight overshooting of approximately 5%, when the drone‘s actual position converges towards the given setpoint.

4.4 Implementation For Aerial Indoor Exploration

For the contribution presented in Chapter 7, again a modified version of the Parrot Bebop 2 [124] is utilized. It is compact and suitable for narrow indoor spaces and offers open-source software [145] for low-level flight control. For reliable experimentation, retro-reflective markers are attached to the MAV for outside-in localization using an Optitrack infrared motion capturing system. An overview of the physical setup can be found in Chapter 7, Figure 7.6.

With all on-board sensors attached, the outer dimensions of the UAV are $32.8 \times 38.2 \times 25cm$, and it weighs $750g$, with flight times of up to 10 minutes. On top of our UAV, we mount a customized RGBD sensor rig ($250g$), consisting of an ASUS Xtion Pro Live sensor

($FOV_{hor.} = 58°$, $FOV_{vert} = 45°$) and a WiFi stick, connected via USB to an ODROID XU3 single-board computer.

The individual components of the safe exploration system are implemented in ROS [119]. The real-time motion planner is implemented in MATLAB (Robotics Toolbox), utilizing the FORCES Pro real-time solver [146].

During our experiments, the UAV was navigating at a default flight height $z_{takeoff} = 1.25m$. The height range for projecting 2D to 3D data was selected with $z_{min} = 0.1m$ to $z_{max} = 2.5m$, covering typical room heights.

5 CLOSE Distance Teleoperation Utilizing Spatial AR

Contents

Major improvements in terms of how humans interact with machines and digital information are still ongoing. In the last years, interfaces based on direct-touch or devices with gesture recognition have come to maturity. Mobile portable devices, like smartphones, and wearables, like head mounted displays, are becoming widespread. However, they require either visual or physical attention and constrain the user.

Spatial augmented reality [6] tries to evade those constraints, but is strongly dependent on projection devices with significant weight, which therefore have to be considered as stationary. As a consequence, it is difficult to cover wide projection areas.

We propose to address these limitations by combining augmented reality and mobile robotics into a new form of HMI. Specifically, we introduce a small semi-autonomous micro aerial vehicle (MAV) with an onboard lightweight laser-projection system and visual sensors, called the Micro Aerial Projector (MAP) (Figure 5.2). We think of it as a robotic companion, which follows the user and is able to project supportive information in the 3D environment. Figure 5.1 shows an example where the MAP assists a student solving mathematical problems by projecting results into the environment.

The design of the MAP requires mastering several challenges. First, safety considerations require that the MAP has to be small sized and as lightweight as possible. It should provide enough payload for all required input, output and computational units. Furthermore, it has to offer sufficient flight dynamics and flight times for being able to follow the user for an adequate amount of time. To meet these requirements, we introduce a novel laser projection system built from scratch. It is small, lightweight and can project a set of basic symbols for a wide variety of HMI scenarios.

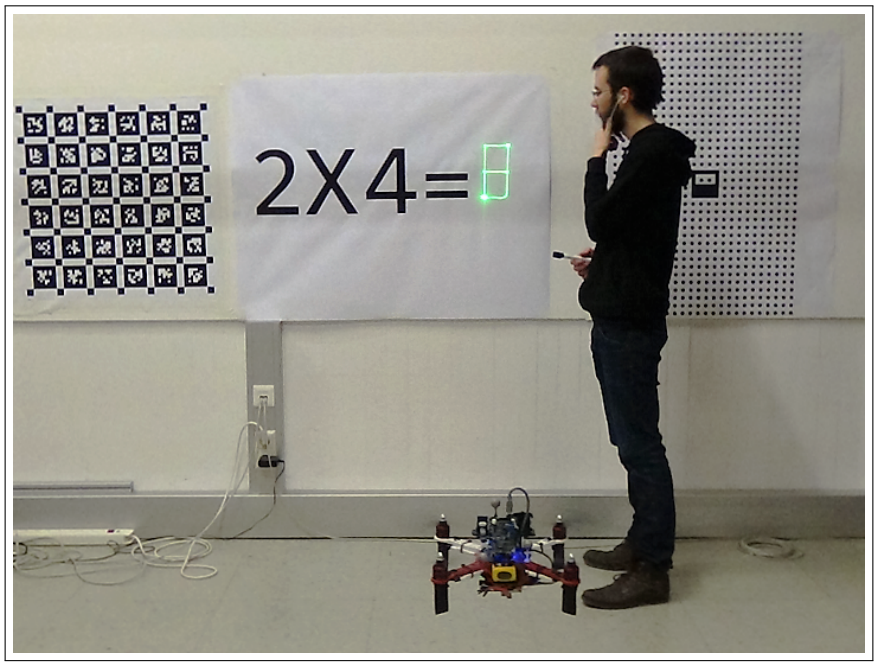


Figure 5.1: The MAP supporting a student by recognizing an equation and projecting the result during hover flight.

Second, the quality of airborne projection is influenced by the stability of the MAV. Changes to position and orientation of the drone, while hovering or during dynamic flight, as well as vibrations emerging from the rotors during flight, must be considered. Thus, a stabilization method is necessary to compensate at least for significant movements of the drone. To this aim, we propose a simple but robust feedforward correction approach, which is able to tackle image fluctuations by deflecting the projector's laser beam. The feedforward correction algorithm is based on pose estimates of an Optitrack motion tracking system. We evaluate quality of the projection stabilized by directly using the pose estimates from the tracking system and compare it to utilizing sensor fusion with the IMU, which is implemented in the inertial state estimation of the onboard flight management controller.

The contributions of this work are the following. As part of the proposed scenarios, this paper represents our first step towards combining the fields of mobile robotics, spatial augmented reality and HMI, focusing on the MAP as a small sized and potent flying projection platform. We introduce a novel lightweight projection system built from scratch, complemented by a projector calibration model. Our system is able to project a steady pattern from a moving drone. We quantitatively evaluate the accuracy of the steady pattern projection during autonomous hovering and dynamic flight. Furthermore, we examine the overall system capabilities, discussing the difficulties and limitations when putting the MAP into practice.

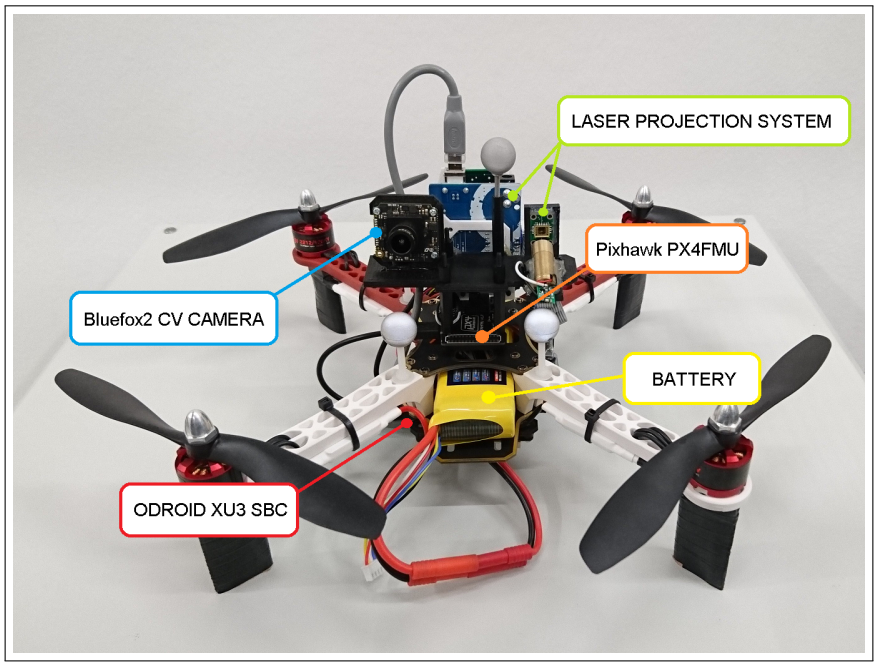


Figure 5.2: MAV platform of our experimental setup including the main components. The laser projector together with the camera is mounted on top. Below, the flight management controller is located. The battery is mounted in the middle to balance weight distribution. The onboard computer is located at the bottom.

5.1 AR Teleoperation Interface For In-Situ Guidance

This section describes our system setup including the MAP and discusses its characteristics and limitations. We use an Optitrack motion tracking system based on eight tracking cameras. The Optitrack server is connected to the groundstation via Ethernet. The groundstation is connected to the MAP through WiFi. The laser projection system and the PX4 autopilot are mounted on the MAP and interfaced to the onboard computer via serial links. Figure 5.3 shows an overview of the setup.

In Figure 5.2, we show the ready-to-fly MAP. The battery is mounted in the middle for balanced weight distribution. Below the frame, the on-board computer is positioned. Above the battery, the low-level flight management controller is located. The projection system and the camera are mounted on top, facing in forward flight direction. The camera is inclined to keep the rotors out of the field of view (FOV).

5.1.1 Micro-Aerial Vehicle

The MAV, which is of 33cm frame diameter and 1200g weight, uses a semi-customized design with engines, rotors and frame taken from a DJI F330 platform. It is powered by a single 2700mAh battery with 14.8V. The flight time is acceptable with about 11 minutes, while running all relevant components and tasks for stabilization of the projection. As

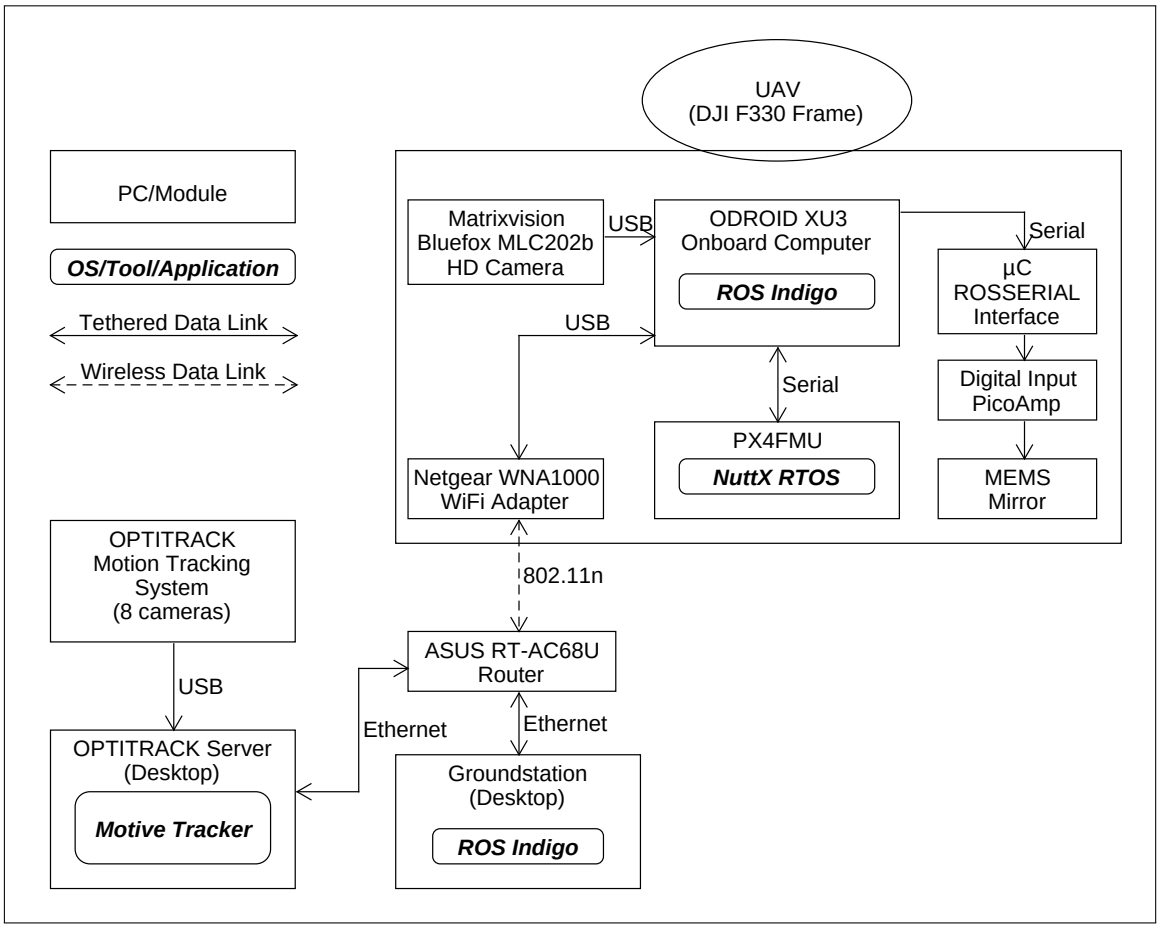


Figure 5.3: The main components of our setup include the MAP itself, the on-board computer and the custom laser projector. We use an Optitrack motion tracking system combined with a groundstation to control the MAV.

an onboard SBC we use an ODROID XU3 with a Samsung Exynos 5422 processor. We added a Pixhawk PX4 flight management controller [147] as a low-level flight control unit including an inertial navigation estimator [135]. The ODROID is connected to the PX4 via a serial link and communicates with a ground-station via WiFi. It captures image data up to 24.6Hz with 1280x960 resolution from a forward-looking MatrixVision Bluefox2 camera connected via USB 2.0. The MAP uses the camera for sensing context-related information in the environment, enabling it to interact with the user (Figure 5.1). All high-level tasks, including processing of image data, processing estimated poses from the motion tracker and sending feedforward correction data to the laser projection system, run onboard and are implemented in the ROS framework [119].

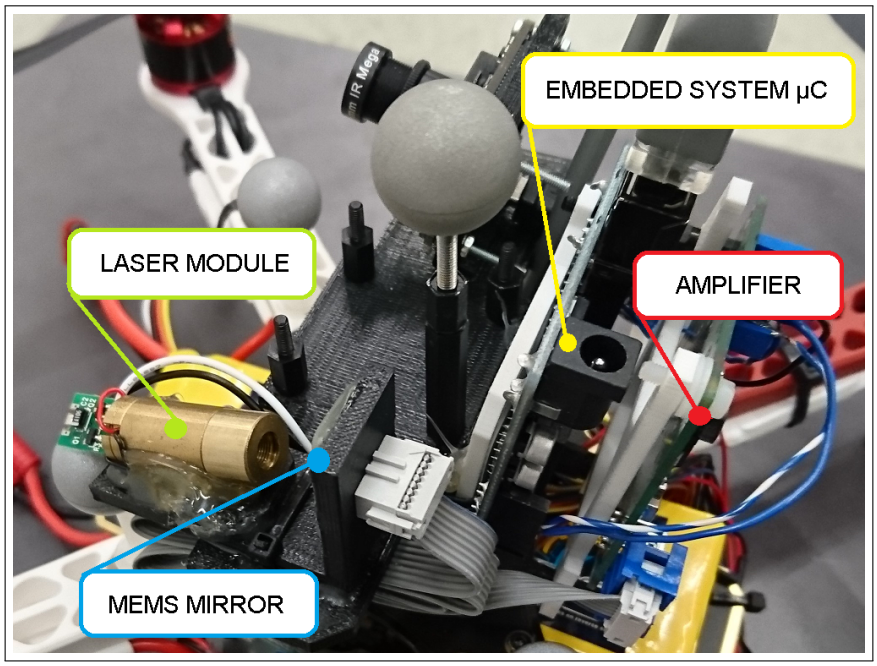


Figure 5.4: Detailed overview of the laser projection system. The emitted light of the laser module is reflected by a forward facing MEMS mirror, interfaced to an amplification stage that converts commands from an embedded system (μC) into appropriate voltages and steers the mirror in two directions.

5.1.2 Laser Projection System

We custom-built a laser projector (Figure 5.4) with a weight of approx. 100g from the following components: A 5mW laser module emits green light at a wavelength of 532nm. Due to the human eyes sensitivity to this wavelength, it improves contrast perception. The laser beam is deflected by a two-axis (tip/tilt) MEMS mirror [148]. Its reflective surface can be steered in a range of $\pm 5°$ by applying a bias differential driving scheme. To generate the required DC bias voltage, we use an amplifier interfaced to a microcontroller (μC) via SPI, the SPI clock frequency is 1MHz. The μC is connected to the ODROID via a serial link running at 57.6Kb/s and relying on the ROS-serial package. To draw images with the laser, we use a vector graphics approach. For a demonstration of projections, refer to Figure 5.6. We send an array of ROS messages with x/y coordinates to the laser, whereby a rate of 100Hz is selected to improve connection reliability of the ROS serial link. The projector coordinates are defined by α_x and α_y and internally represented as angles at which the mirror is steered in x/y direction. Typical latencies of sending the messages from the SBC to the μC are 11ms. Sending position commands from the μC to the amplifier is currently done at a rate of 2kHz. As the complexity of measuring the time delay between sending commands from the μC and actually steering the MEMS mirror is significant, the delay is treated as unknown.

5.1.3 Flight Control Of The MAV

For flight control of the MAV, we use an Optitrack motion tracking system providing the PX4 flight management controller with low latency pose estimations. We use eight stationary cameras, covering an area of roughly $5 \times 4 \times 3m$ length, width and height. Poses from the Optitrack motion tracker are derived by the Optitrack server with a latency of 10ms. Transfer to the groundstation with Ethernet adds a latency below 1ms. To relate to measurement rates from a vision camera, poses are then delivered to the SBC via WiFi at 20Hz. Interfacing the PX4 flight management controller is also done via serial link running at 57.6Kb/s with below 10ms of delay. The system time between the Optitrack server, ground-station and the PX4 is synchronized via NTP/Chrony [149]. For a more detailed overview of the system, also including rates and latencies between the individual components, refer to Figure 5.7.

5.1.4 Pose Estimation For Stabilization Of Projection

In addition to directly using the estimated poses from the motion tracking system, we further utilize an inertial position estimation approach on the low-level flight management controller to reduce noise and obtain interpolated position estimates at higher rates to improve stabilization accuracy of projections. We further exploit the noise filtering characteristics of the estimator to reduce flickering of the projected images and increase visual quality.

5.2 Stabilization Of Projected Images

In this section, we concentrate on the problem of stabilizing projected information suffering from movements of the MAV. We propose a method for stabilization and describe our implementation deployed on the onboard computer.

5.2.1 Coordinate Frames and Transformations

Figure 5.5 shows the reference frames of our experimental setup: The world, the region of interest (ROI) on the wall, the MAP and the laser projector. Note that all coordinate frames are right handed, the coordinate frame of the projector and the ROI are in OpenCV conventions, letting the z-axis point towards the wall. We stabilize the projected image using the 4×4 homogeneous transformation matrix T_{PR} from the ROI on the wall with respect to the projector (Eq. 5.2), consisting of the rigid transformations T_{PM} from the MAP wrt. the projector, T_{MW} from the world wrt. the MAP and T_{WR} from ROI wrt. the world coordinate frame, which is tracked by the motion tracking system:

$$T_{PR} = T_{PM} \cdot T_{MW} \cdot T_{WR} \tag{5.1}$$

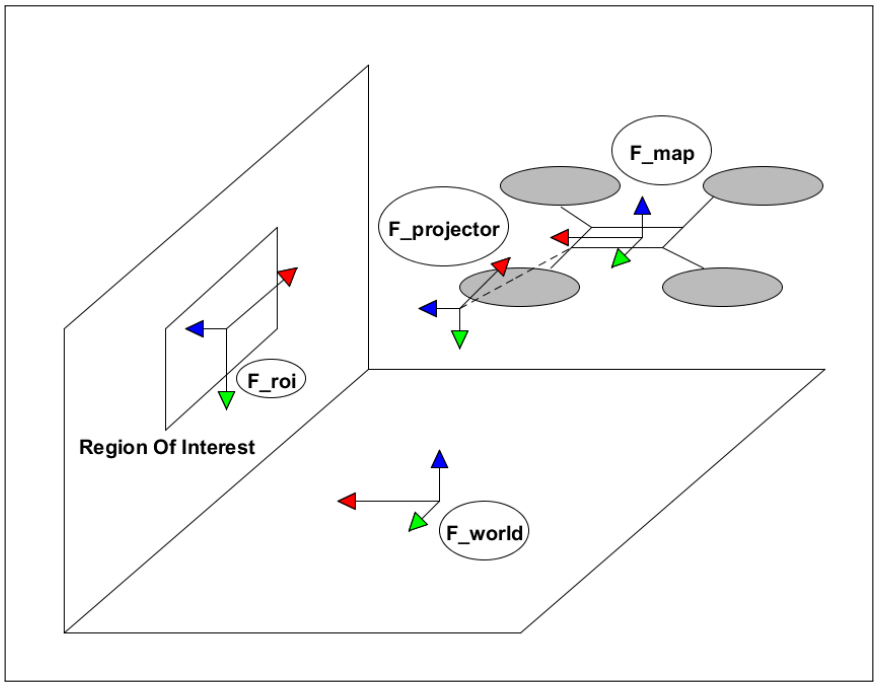


Figure 5.5: Overview of coordinate frames in our experimental setup. Using right handed coordinate convention, the x-axis is colored red, y-axis is green and z-axis is blue.

A 3D point P_R in the frame of the ROI can be transformed to the projector frame with

$$P_P = T_{PR} P_R \tag{5.2}$$

and further on into image space of the laser.

5.2.2 Laser Projector Model

As our projection system is based on a MEMS mirror, it shows nonlinear relations between applied DC voltage and mirror inclination of the individual axis [148]. To compensate for those nonlinearities, while still being able to relate to common computer vision algorithms, we suggest a pinhole camera model with nonlinear lens distortion to describe the characteristics of the projector. In our approach, we transform a 3D point P_P given in the projector frame (Eq. 5.2) into the projectors coordinates.

By first normalizing the point P_P, we have

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} P_{Px,n} \\ P_{Py,n} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{P_{Px}}{P_{Pz}} \\ \frac{P_{Py}}{P_{Pz}} \\ 1 \end{pmatrix} \tag{5.3}$$

Applying radial distortion with the Taylor approximation to an arbitrary function $L(r)$ [150] and neglecting all coefficients K_n except for the second-order term results in

$$\begin{pmatrix} x'' \\ y'' \end{pmatrix} = L(r) \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} (1 + K_{2x} r^2) \cdot x' \\ (1 + K_{2y} r^2) \cdot y' \end{pmatrix} \tag{5.4}$$

where x'' and y'' are the distorted coordinates, K_{2x} and K_{2y} are the distortion coefficients for the second-order terms and r is the radial distance $r^2 = x'^2 + y'^2$. We use two different distortion coefficients for the second-order terms of x and y axis, because we expect different nonlinear behaviors in terms of the individual axis of the MEMS mirror. This would be the equivalent to describing a distorted lens with an ellipsoid shape.

Finally, we apply calibrated intrinsics from the pinhole camera model, described in Section 5.2.3, to receive the actual coordinates in our laser image space.

$$\begin{aligned} \alpha_x &= f_x \cdot x'' + c_x \\ \alpha_y &= f_y \cdot y'' + c_y \end{aligned} \tag{5.5}$$

The projector coordinates α_x and α_y are directly used as the inputs for the laser projection interface to steer the mirror in x/y direction.

5.2.3 Laser Model Calibration

For the calibration of the model, we take a set of 6DOF poses of the MAP in the world coordinate frame. For each pose, we project a symmetric point grid pattern onto a planar wall and measure the 3D coordinates of the projected points in world coordinates. A desired grid of laser image points, gets distorted on a planar projection surface due to nonlinearities in the MEMS mirror. Compared to common camera calibration procedures, where a non-distorted checkerboard is used to calibrate for the intrinsics of the camera, our approach considers the opposite. The desired points in the laser space are the true grid points and get distorted in real world.

We use the Ceres Solver [151] and define the reprojection error in the laser image space as optimization residual. This has the advantage that the projection surface does not need to be known, as the calibration 3D projected points are directly measured. The natural choice would be to compare the 3D projected point positions from the calibration dataset against the projection of the laser commands into the real world. However, this would pose problems due to the non-existence of an analytic inverse for the camera distortion model, which needs to be coded into the residual.

Our solution is to utilize a reprojection error residual in the laser image space (Eq. 5.6). As a consequence, we need to propagate the measurement covariance into the laser image. This results in the forward covariance propagation from the measured 3D point to the laser image through the backprojection operation. Thus, our calibration respects the maximum likelihood principle and can be used to estimate a proper calibration parameter covariance through the backward transport of covariance theorem [150].

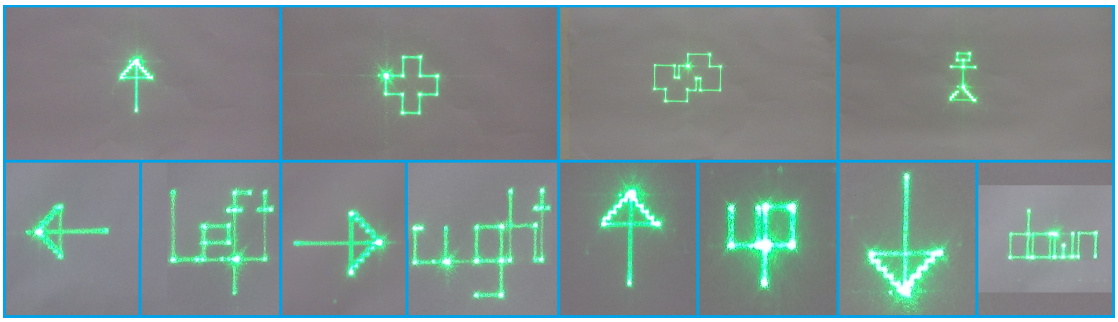


Figure 5.6: Shown is a set of symbols which can be projected by the projection system. Included are an arrow for instructing users, a red cross representative for a point of interest and the logo of the Graz University of Technology. Additionally we can project directions and letters, for example, to guide a user inside of a building.

Our calibrated parameters are defined by

$$\mathbf{r}(\mathbf{T_{PR}}, \mathbf{K}, \mathbf{D}) = \frac{\boldsymbol{\alpha_{img,cmd}} - \hat{\boldsymbol{\alpha}}_{\boldsymbol{img,backpr.}}}{\boldsymbol{\sigma}} \tag{5.6}$$

where T_{PR} transforms a 3D point measured on the wall into the laser projector frame, K and D include the calibrated intrinsic parameters and distortion coefficients of the laser model, $\alpha_{img,cmd}$ are the commanded points in the projector image plane, $\hat{\alpha}_{img,backpr.}$ are the points in the projector image plane derived from the backprojection and σ is the previously discussed propagated covariance of our 3D points, measured with the motion tracker, through the backprojection operation.

5.2.4 Compensation via Feedforward Correction

For stabilization of the projected information, we want to use a simple but robust feedforward correction algorithm. We define our desired projection in 3D, which is represented by a projected point on a wall. The position and shape of the target projection surface are known and used to calculate P_R. The desired 3D point P_R is defined in the ROI frame F_{roi} (please also refer to Figure 5.5 and Figure 5.10a). During our experiments, based on this desired point, we derive our projector coordinates and steer the mirror towards it. We do this in the following way: We transform P_R into coordinate frame of the MAP F_{map} using $T_{MW} \cdot T_{WR}$ based on poses from the motion tracker. Next, we use the transformation T_{PM}, calibrated from MAP to projector, to derive coordinates of the defined point in the laser projector frame $F_{proj.}$. Including the intrinsics and radial distortion from our projector model, we finally steer the mirror to the position of the 3D point P_R. According to Figure 5.9, while we calculate the coordinates for correction on the SBC and forward commands to the laser interface (time delays t_1 to t_3), the MAP is moving. Due to the delays of the laser interface, the actual projection happens when the MAP has moved already $(t_3 + \Delta t)$. Thus, we are not able to compensate for the exact position of

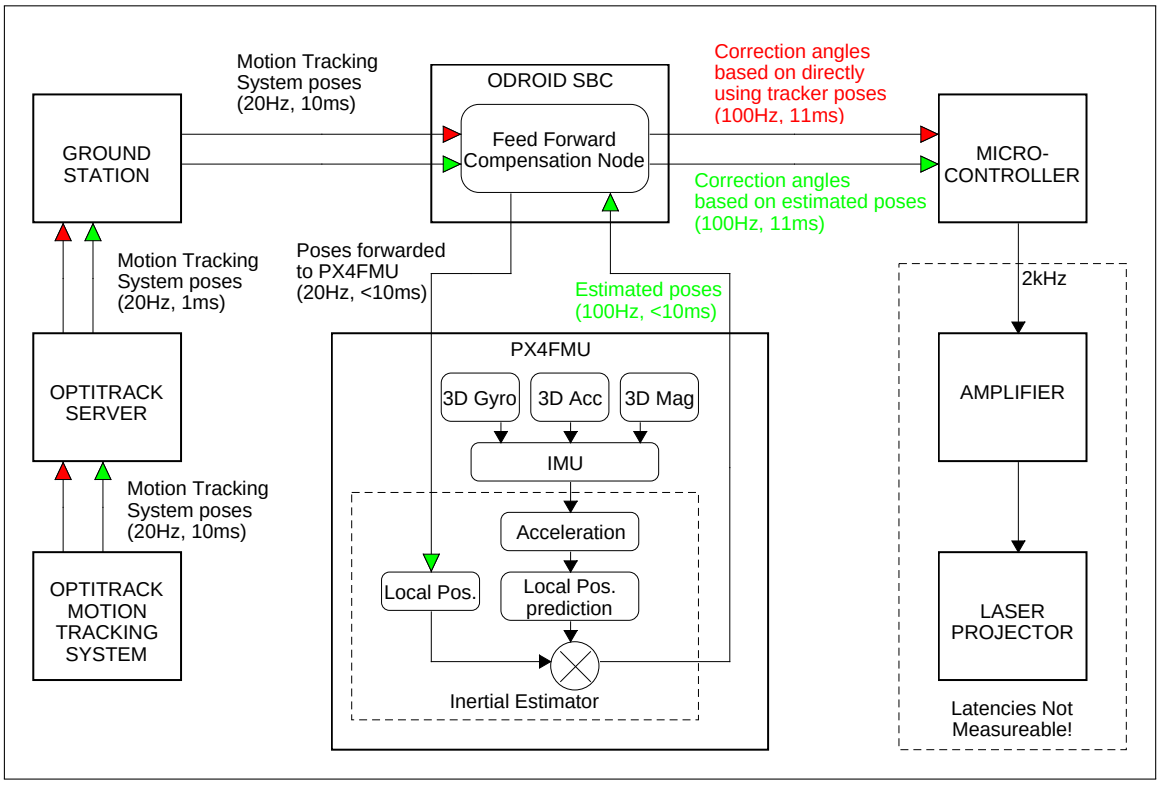


Figure 5.7: Overview of rates and latencies of the experimental setup. The red path shows the poses which are transferred in terms of the direct compensation approach. The green path indicates transfer of poses for compensation via the inertial estimator.

the desired 3D point P_R, which results in an offset of the projection. This problem affects all compensation methods used in our experiments.

For an improved stabilization approach, including noise filtering and interpolation between rare pose readings, we utilize the inertial state estimator of the onboard flight management controller (Figure 5.7). The poses from the motion tracking system are sent at a rate of 20Hz to the SBC and arrive with approximately 21ms of delay. The inertial state estimator fuses measurements of the IMU, running at high rates, with poses from the motion tracker. The state estimator predicts and corrects position in x, y and z for the current time step, whereby predictions are based on accelerometer measurements. It filters out noise and interpolates the motion tracker poses, which are derived from the flight management controller at a rate of 100Hz. This estimated pose is again used to transform the coordinates of the 3D point P_R into the frame of the laser projector. For correction, the updated coordinates in laser space are finally forwarded to the laser projection interface via the serial link. The latency is thereby approximately 11ms.

Figure 5.8 shows a timing diagram between the individual system components with intermediate rates and latencies.

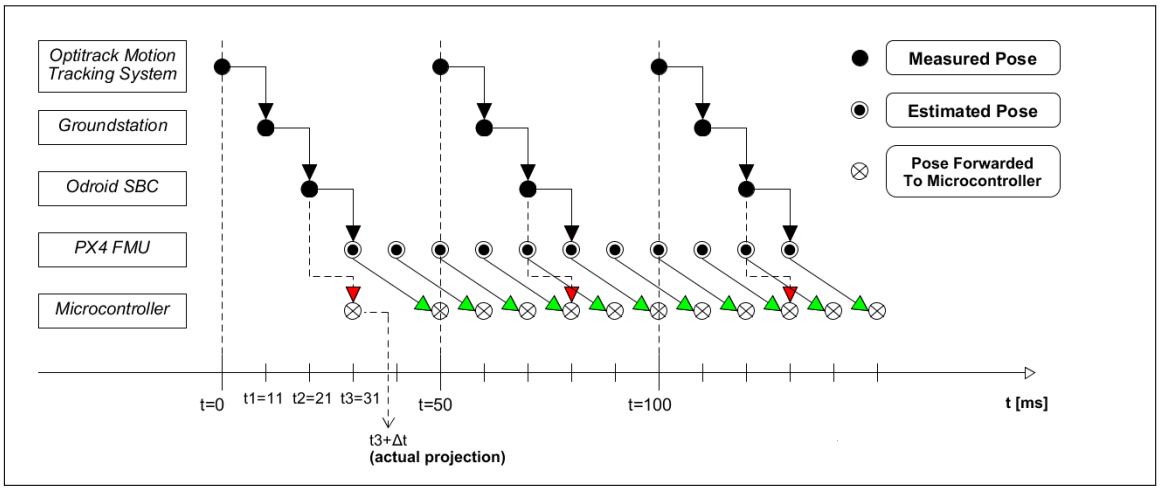


Figure 5.8: Timing of the poses used for feedforward compensation. The red arrows indicate poses of the motion tracking system directly used for compensation, whereas green arrows are poses derived from the inertial estimator.

5.3 Experimental Results

In this section, we analyze the performance of the stabilized projection during flight. We report on four experiments: We elaborate on the accuracy of feedforward stabilization based on poses directly derived from Optitrack and compare the results with compensation by using the fused poses from the inertial estimator of the low level flight management controller. Both methods are applied during hover flight and also during dynamic flight. An example for the MAP's actual position trajectories is shown in Figure 5.10b.

5.3.1 Hover Flight

For experiments during hover flight, we want to position the MAP in air so that the 3D point P_R can be "seen" by the FOV of the laser projector. At the beginning of the experiment, we place the MAP at the origin of the world frame facing the wall. When the MAP takes off, it is commanded to a hovering height of 0.65m, again to be able to project close to the 3D point. The MAPs attitude setpoint is oriented towards the ROI. As we can localize the pose of the ROI on the wall with the motion tracking system, we use our onboard feedforward correction algorithm to project the desired point into the origin of the ROI. At the same time, we project the uncompensated point $\alpha_x/\alpha_y = (0, 0)$ in projector coordinates which represents the disturbance due to the movements of the MAV. To quantify the effects of the compensation, we capture image data of the ROI with an external camera over 60s with a resolution of 1920x1080 at 25fps and detect the uncompensated and compensated points in the image. The ROI is thereby of size A3 with WxH of 420x297mm. The 1500 frames are downsampled by a rate of 10 which results in 150 point pairs (compensated and uncompensated points) in every experiment. In every

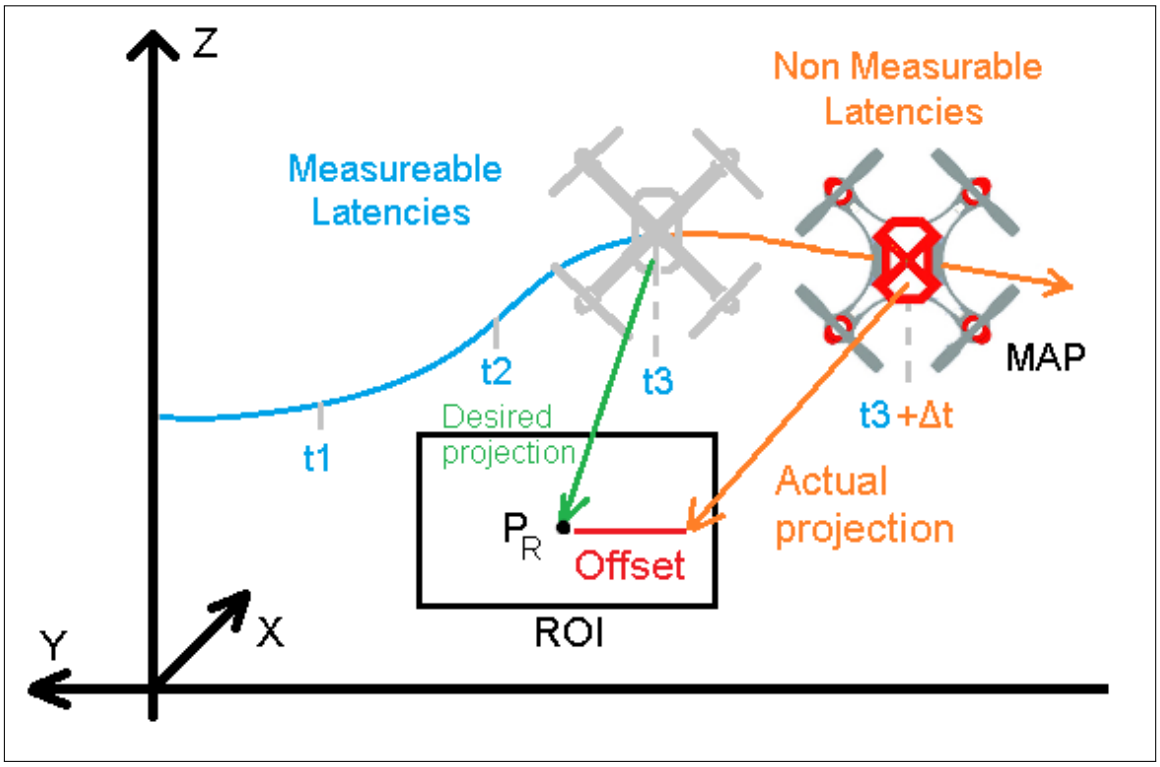


Figure 5.9: Effect of the laser interfaces unknown latency. After the coordinates for correction are derived and forwarded to the laser projector, any movement of the drone results in an offset of the projection.

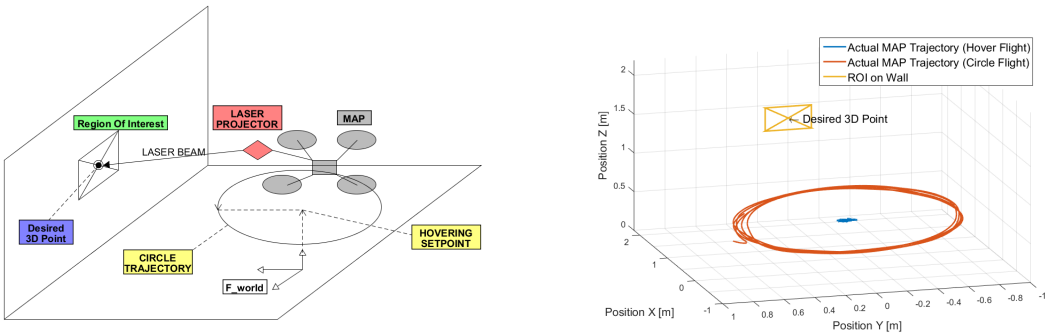
image, we use the Euclidean norm of x/y coordinates in F_{roi} and measure the distance of the projected uncompensated points and compensated points to the desired 3D point P_R. We derive mean error and standard deviation ($\bar{X}$ and σ).

5.3.2 Dynamic Flight (Circle Flight)

For dynamic flight we calculate a full 3D elliptical position trajectory with the center located at $x/y = (0, 0)m$ in the world frame and $0.65m$ above ground. In the x/y plane, we command a circular trajectory with a diameter of 1.4m and with an angular speed of $36°/s$. During flight, we set the attitude of the MAP towards the walls ROI and adjust the height to keep the 3D point P_R approximately in the laser projectors FOV. The wall is located at a distance of 2.18m towards positive x-direction in the world frame.

5.3.3 Experimental Results

Table 5.1 shows a summary of the four experiments. We provide mean and standard deviation of the uncompensated projected points compared to the compensated ones during hover flight $\bar{X}(\sigma)_H$ and circle flight $\bar{X}(\sigma)_C$. Both experiments are evaluated either with



(a) Sketch of the experimental setup. **(b)** Actual position trajectories.

Figure 5.10: The experiments of the MAP in detail. (a) A sketch of the experimental setup. (b) The actual position trajectories of the MAP during circle and hover flight based on a view from behind.

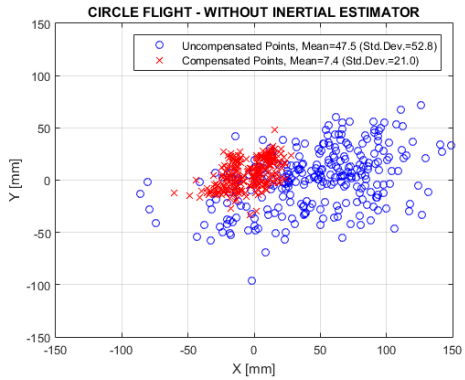
Mean(Std.Dev.) [mm]	uncomp.	comp.
$\bar{X}(\sigma)_{H,noinav}$	23.5 (38.7)	5.2 (6.6)
$\bar{X}(\sigma)_{H,inav}$	27.7 (40.3)	8.6 (5.0)
$\bar{X}(\sigma)_{C,noinav}$	47.5 (52.8)	7.4 (21.0)
$\bar{X}(\sigma)_{C,inav}$	66.6 (56.0)	7.3 (19.3)
$\bar{X}(\sigma)_{S,noinav}$	221.9 (54.9)	7.5 (8.4)
$\bar{X}(\sigma)_{S,inav}$	203.9 (47.4)	9.8 (5.8)

Table 5.1: Error characteristics of feedforward compensation.

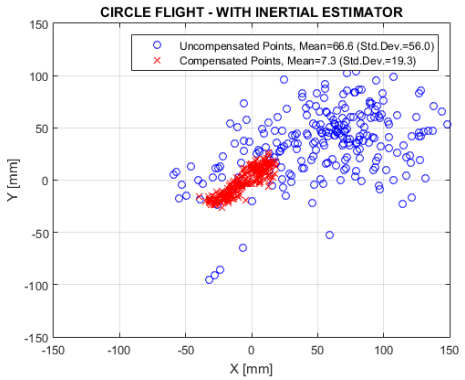
or without using inertial estimates.

In Figure 5.11a, uncompensated and compensated projected points during the circle flight are shown. The results are based on directly using the pose estimates from the motion tracking system. It is clearly visible that the dynamic error, resulting from the movements of the drone, is significantly reduced by the feedforward correction. Offsets to the desired point in the origin of the axis are a result of movements of the MAP, errors in the calibration of the transformation T_{PM} and inaccuracies of the estimated model of the laser projector. Figure 5.11b shows results of the same experiment, but the feedforward compensation is based on poses from the inertial estimator. It is obvious that the filtering characteristics of the inertial estimator reduce noise and also help to smooth out the movements of the projected image on the plane. This improves accuracy, which is also reflected in the standard deviations $\sigma_{C,noinav}$ and $\sigma_{C,inav}$. Moreover, also visual quality is significantly increased.

We took a set of measurements positioning the MAP at random static poses in front of the ROI, to inspect on the static error of the feedforward compensation. The uncom-



(a) Results of compensation based on directly using Motion Tracker.



(b) Results of compensation using estimated poses.

Figure 5.11: Measured projections of uncompensated and compensated points during circle flight. (a) Results of compensation based on directly using poses from the Optitrack motion tracking system. (b) Results of compensation using estimated poses from the flight management controller.

pensated point is steered randomly to outer regions of the FOV, which also means that the MEMS mirror almost reaches its maximum inclination angles. It reflects accuracy of the calibration and influences of nonlinearities of the laser model.

It is noticeable that the error characteristics of compensated points $\sigma_{H,noinav}$ and $\sigma_{H,inav}$ during hover flight are slightly below the static error characteristics $\sigma_{S,noinav}$ and $\sigma_{S,inav}$. This is due to the fact that, in hovering position, the dynamics of the MAP are low and the origin of the laser projector can be kept quite close to the desired 3D point. Therefore, inaccuracies in our estimated nonlinear laser model have less influence on the compensated points. Furthermore, the mean error during hover flight without using estimates ($\bar{X}_{H,noinav}$) is significantly lower. This is considered to be a result of the low disturbance characteristics during this specific flight experiment.

5.3.4 Use Case Scenario

Considering the MAP as a robotic companion, it should be able to support the inexperienced user in certain HMI scenarios. Therefore we implemented a "teaching assistant" scenario as a showcase. The MAP is able to support a student with solving basic equations (Figure 5.1) by autonomously detecting the content of the equation, solving it and projecting the result next to it. Please also refer to the supplementary video[1].

In detail, the following scenario is shown: The MAP first approaches a given equation and reads it during flight by utilizing a neural network based approach for text recognition. Then, the user approaches the equation and tries to solve it, while the MAP moves away to keep distance. In our example, the user fails on the first attempt. When the user writes

[1]Supplementary video - Micro Aerial Projector: https://youtu.be/0A2EtgqAMNE

down a result and moves away from the equation, the MAP comes in to check the result. Depending on validity of the result it projects "$\surd$" for correct or "X" for wrong on the wall to signal the user validity of the result. Then it moves away again. The user should write down the correct result and move away from the equation. The MAP moves in again to check and signals if correct result was detected by projecting on the wall. Finally, the drone approaches a default waypoint and projects the correct result next to the equation. During the waiting phases, the MAP signals that the user should interact with the scene by projecting "hold" on the wall.

6 MEDIUM Distance Teleoperation Utilizing MR

Contents

Small sized aerial robots with cameras attached, also called camera drones, can easily fly into locations which are impassable or too dangerous for humans to reach. Using camera drones allows exploring such areas from a safe distance, providing essential data for diverse applications, such as rescue missions, infrastructure inspection or just photographic exploration.

Depending on the situation, a drone pilot may choose between two principal modes of flight control. In either mode, a conventional handheld controller is used to steer the drone, but the viewing differs between modes: In *exocentric viewing* mode, the pilot observes the drone from the ground while steering. In *egocentric viewing* – or first-person – mode, video from the drone‘s on-board camera is streamed to the pilot to inform the steering. Recently, wearing a head-mounted display to watch the streaming video has become a popular enhancement of the egocentric mode.
Obviously, piloting a drone in exocentric mode is difficult in the presence of significant occlusions, and it becomes impossible when the drone is exploring the inside of a building. In this case, the pilot is forced to use an egocentric mode based on streaming the image from the on-board camera. But navigation in a narrow environment while relying exclusively on an on-board camera with a potentially limited field of view can be difficult.

If only an egocentric view is available, flight control is more difficult than necessary. An exocentric view, showing the drone‘s surrounding from the pilot‘s rather than the drone‘s point of view, would clearly be preferable.

Virtual reality (VR) can provide a synthetic exocentric view by combining the live video with a 3D model of the occluded environment from any viewpoint, independent

of user‘s physical viewpoint. Using image-based rendering of the drone‘s video stream delivers a realistic impression with live updates [152]. By coupling the drone autopilot to the user‘s gaze direction, the experience is redefined from remotely piloting a drone to perceiving the occluded world with *drone-augmented human vision*. As a special case of VR, augmented reality (AR) additionally adds the illusion of X-ray vision: A pilot wearing a see-through display can make the walls or other occluders partially transparent to reveal the area currently observed by the drone.

In addition to supporting a virtual viewpoint, AR also allows users to investigate the scene from their physical viewpoint and spatially relate occluded geometry with the visible world. In case of a disaster scenario, a rescue team can quickly locate an imminent danger, such as a fire or explosion behind an occluder, and proceed with caution.

Spatial relationships between visible and occluded geometry become especially important when infrastructure modifications are needed. For example, drilling a hole in a wall without damaging the cables or pipes located on the occluded side of the wall requires estimating their positions from the visible part of the wall.

We demonstrate the first proof-of-concept implementation of drone-augmented human vision. We couple an indoor drone with a head-mounted display (HMD) to deliver an exocentric perspective on the drone, letting the pilot control the drone via gaze direction. We present a first experiment showing how virtual exocentric visualization supports spatial understanding and thus enables exploration and natural interaction with the drone. In a second experiment, we use VR (non-see through) for its virtual viewpoint nature and compare it with the physical viewpoint that is additionally provided by AR (see-through).

6.1 MR Teleoperation Interface For Through-Wall Inspection

6.1.1 Interface design

Our drone-augmented human vision system lets the pilot control a drone inside an occluded space indirectly, via an exocentric visualization provided in a see-through HMD (Microsoft HoloLens). While the drone travels in the remote environment, the video frames streamed from the on-board camera are projectively texture-mapped onto a geometric model of the scene. The scene is rendered from user‘s current perspective, as measured by the built-in self-localization of the HMD.

In addition, a virtual representation of the drone is rendered at the position reported by the physical drone, to give the pilot an overview of the physical configuration of the occluded space. The interior scene with partial texture mapping is made to appear inside a ”cutaway” magic lens that appears as a hole in the occluding wall structure.

For flight control and navigation in the occluded space without hitting obstacles, we introduce two interaction techniques, called *pick-and-place* and *gaze-to-see*. Moreover, we introduce *overview-and-detail*, a transitional interface [153] to reveal details on demand.

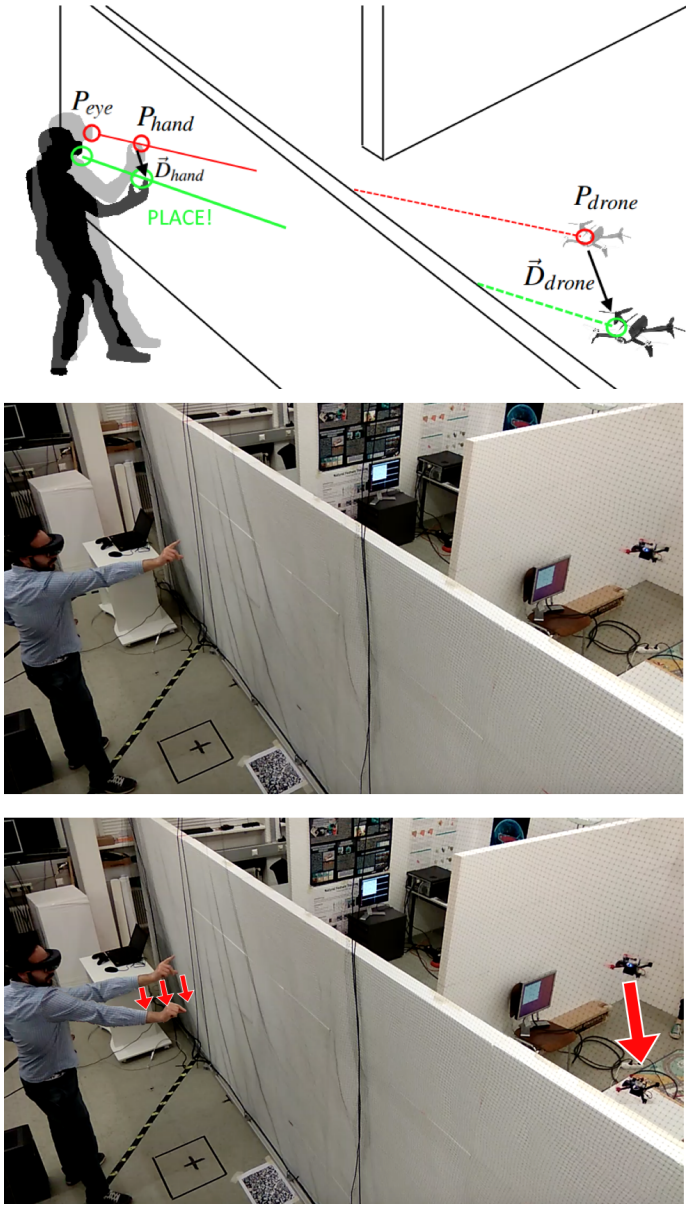


Figure 6.1: Pick-and-place steering.

6.1.1.1 Pick-and-place

This interaction technique allows users to pick a drone by looking at it and applying a pinch gesture. After picking the drone, moving one's hand repositions the drone in 3D space, as illustrated in Figure 6.1. The hand movement is scaled proportionally to the distance of the picked object, as in the scaled-world-grab technique proposed by Mine et al. [154]. More formally, the displacement vector $\mathbf{D}_{drone}$ of drone's position P_{drone} in $\mathbb{R}^3$

is calculated as

$$\mathbf{D}_{drone} = \frac{\|P_{drone} - P_{eye}\|}{\|P_{hand} - P_{eye}\|} \cdot \mathbf{D}_{hand}$$

where P_{eye} and P_{hand} represent the positions of the eye and the hand, respectively, while $\mathbf{D}_{hand}$ indicates hand's motion vector in $\mathbb{R}^3$ (Figure 6.1. P_{eye} and P_{hand} are directly provided by HoloLens, whereas P_{drone} is received from the drone tracking system. Note that, depending on factors like dominant eye, HMD position on the head or the distance of the currently focused object, P_{eye} measurement may be subject to brittle calibration. However, during our experiments, users did not indicate that they needed (re-)calibration.

6.1.1.2 Gaze-to-see

Using the view vector and eye position provided by HoloLens, one can calculate the point of interest P_{gaze} a user is gazing at by intersecting the viewing ray with the scene model. Knowing gaze position allows to predict which part of the occluded scene a user is interested in. Therefore, in this interaction technique, the drone focuses on the high level goal of the user and automatically repositions to observe the area around the user's point of interest with its on-board camera. Let $\mathbf{N}_g$ be the normal vector at P_{gaze}, and let $\mathbf{Z} = \{0, 0, 1\}$ denote the up-axis of the scene. The drone is positioned at

$$P_{drone} = P_{gaze} + \frac{\mathbf{N}_g - (\mathbf{N}_g \cdot \mathbf{Z}) \cdot \mathbf{Z}}{\|(\mathbf{N}_g - (\mathbf{N}_g \cdot \mathbf{Z}) \cdot \mathbf{Z})\|} \cdot x$$

$$if \, \|\mathbf{N}_g \cdot \mathbf{Z}\| < \|\mathbf{N}_g\| \cdot 0.9$$

Unless we are looking at a horizontal surface, the drone will reposition x meters away from the point of interest along a displacement vector corresponding to the surface normal projected to a horizontal plane (Figure 6.2).

In our experiments, we set x to 0.5 meters for ensuring a close-up view of the surface. The drone's yaw orientation is adjusted to align with the negative displacement vector. In case the user looks at a horizontal surface, the drone is positioned between the user and the point of interest, mimicking the user's view vector in the horizontal plane. If the calculated position is not inside the safe flight zone, the repositioning terminates at the nearest border of the permitted flight zone.

6.1.1.3 Overview-and-detail

By visualizing the occluded scene and the drone from user's perspective, our system allows a drone pilot to better understand the spatial relationships between scene geometry, drone and the pilot's body. However, this visualization lacks details, as the drone can be far away and both camera and display suffer from a rather limited field of view. Therefore, we introduced an overview-and-detail technique, which fills the gap between egocentric and exocentric drone control modes in the form of a transitional interface [153] using image-

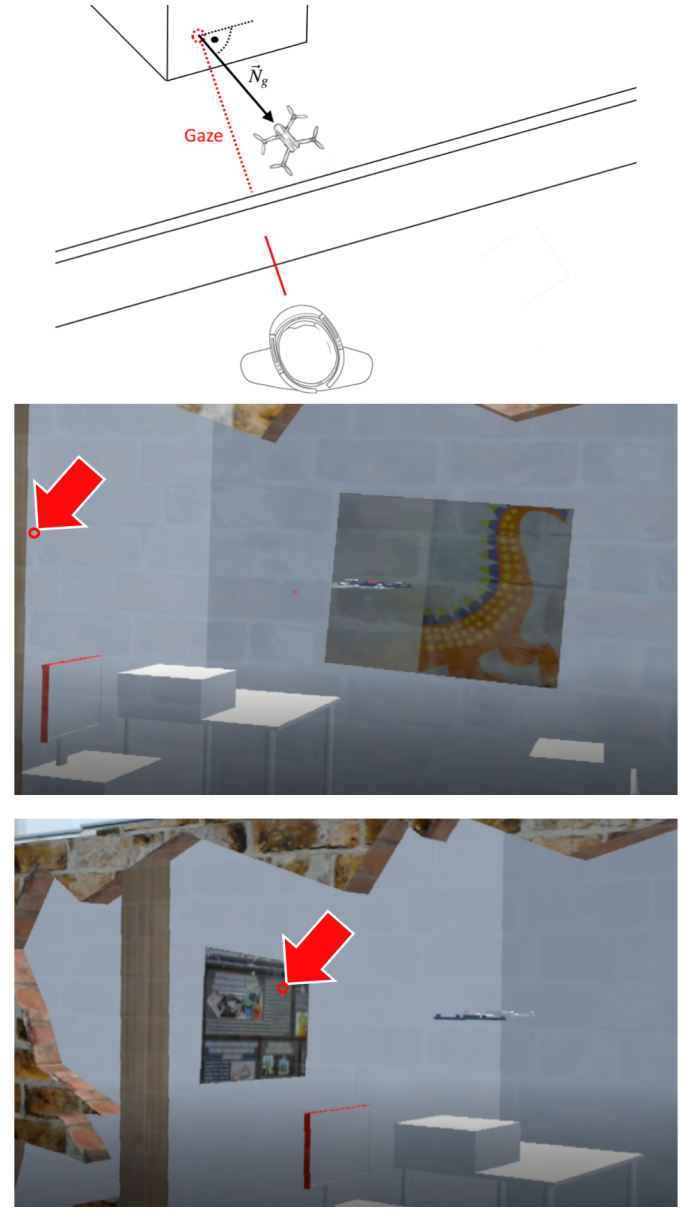


Figure 6.2: Gaze-to-see steering.

based warping [155]. After steering the drone to a point of interest, users are given option to either virtually move closer to the drone or to the currently gazed-at surface point in the occluded scene, by selecting the corresponding interface hotspot. During the detail visualization, we apply the occluding wall structure to clip the zoomed detail geometry in order to avoid confusion between real and occluded virtual geometry (Figure 6.3). Zooming in is achieved by positioning the virtual hole in front of the gazed point while preserving the relative transformation between the virtual hole and the user‘s camera view. The

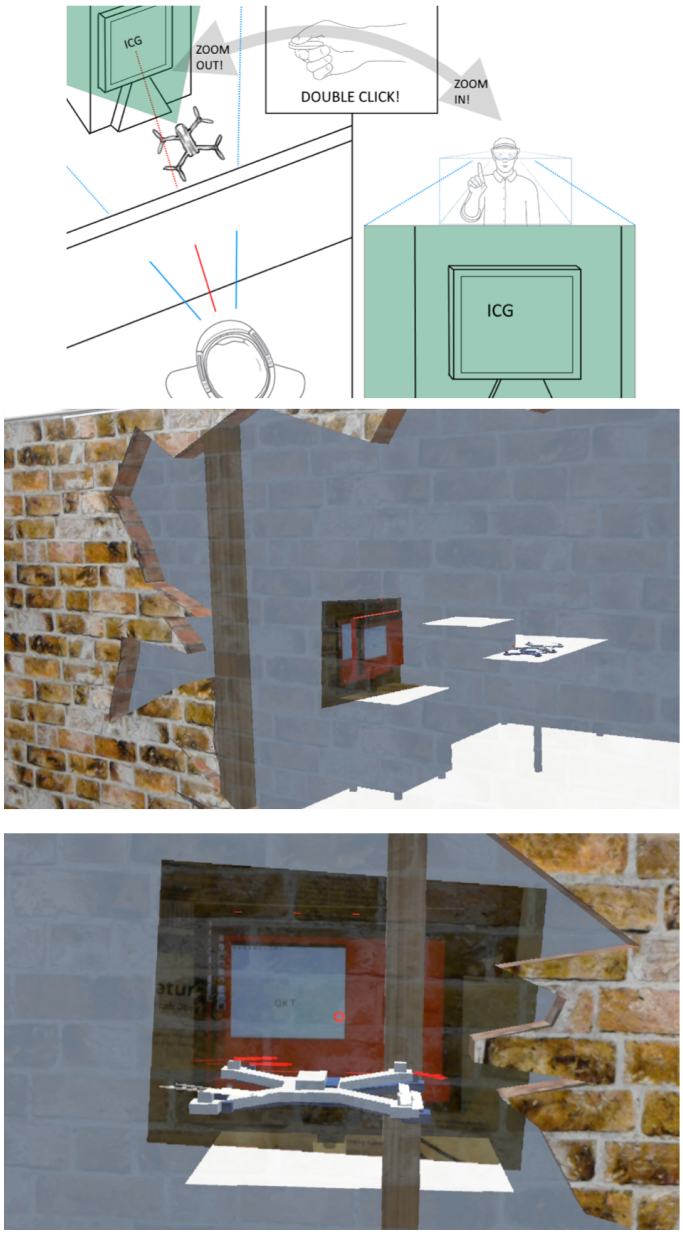


Figure 6.3: Overview-and-detail.

position of the virtual hole is computed in the same way as the positioning of the drone in gaze-to-see interaction.

6.1.1.4 Precomputed path planning

For our experiments, we wanted to relieve the pilot as much as possible from path planning, providing the illusion of augmented vision without concerns about flight safety. However,

a fully featured path planning is computationally expensive and can be brittle. Since we track the drone externally, rather than by SLAM, we can pre-compute the necessary path planning information from the floor plan. In our test environment, we divided the space into three regions, two rooms connected by a corridor.

If the pilot issues a repositioning command that requires changing the region, the path planning first approaches a predefined waypoint at the boundary before progressing to the neighboring region. Overall, our path planning is simplistic, but works instantaneously and reliably prevents accidents due to hitting obstacles or walls of the scene. A more realistic path planning based on SLAM would run an A* algorithm on a map of the environment that has already been explored by the drone.

6.1.1.5 Joypad control

Alternatively to the path planning, the drone can be controlled via a joypad. In this case, a custom ROS node integrates the inputs from four axes of the joypad and converts them into a 3D position and yaw of the drone. We derive the position reference commands by integration of the joypad‘s linear axis commands J_i over the time intervals between discrete times i. The position error E_i in this case is given as $E_i = J_i - Y_i$.

To enable a fair comparison between the exocentric interaction techniques introduced in section 3 and the joypad interface, we added advanced features to the joypad interface, which go beyond what is conventionally available in commercial drone control.

First, we provide drift-free stabilization of the MAV position during navigation in the scene. This kind of stabilization is not available when using off-the-shelf drone technology. Conventional tracking and stabilization, especially in the x-y plane, is usually based on optical-flow or inertial sensors, which suffer from drift over time. With the drift-free tracking, we also enable a basic level of disturbance rejection against turbulences which occur during flight in narrow parts of the scene.

Second, we chose Mode-2 axis mapping on the joypad, which is a well-known and widely accepted mapping for drone control. It is also the default configuration in a variety of off-the-shelf drone products, e.g., the Parrot AR Drone 2.0, the Parrot Bebop 1/2, and the DJI Marvic. Mode-2 mapping employs the left joystick for commanding vertical velocity and velocity around the rotational z-axis of the drone. No direct thrust control is required by the user, reducing cognitive load. The right joystick controls the translational velocity in X and Y direction.

Third, we created a safe-guard for the use by introducing artificial boundaries inside the scene, so the user is not able to crash the drone into walls or hit obstacles. Before each experiment, the user was informed that crashing the MAV is not possible. We presented visual feedback when the user hits the artificial boundaries via warning message, and we visualized the valid flight areas inside a 3D perspective view with green bounding boxes (Figure 6.4). If the user hit the boundaries, the drone did not fully stop, but continued movement along the boundary with the resulting speed vector. Thus, the user was able to

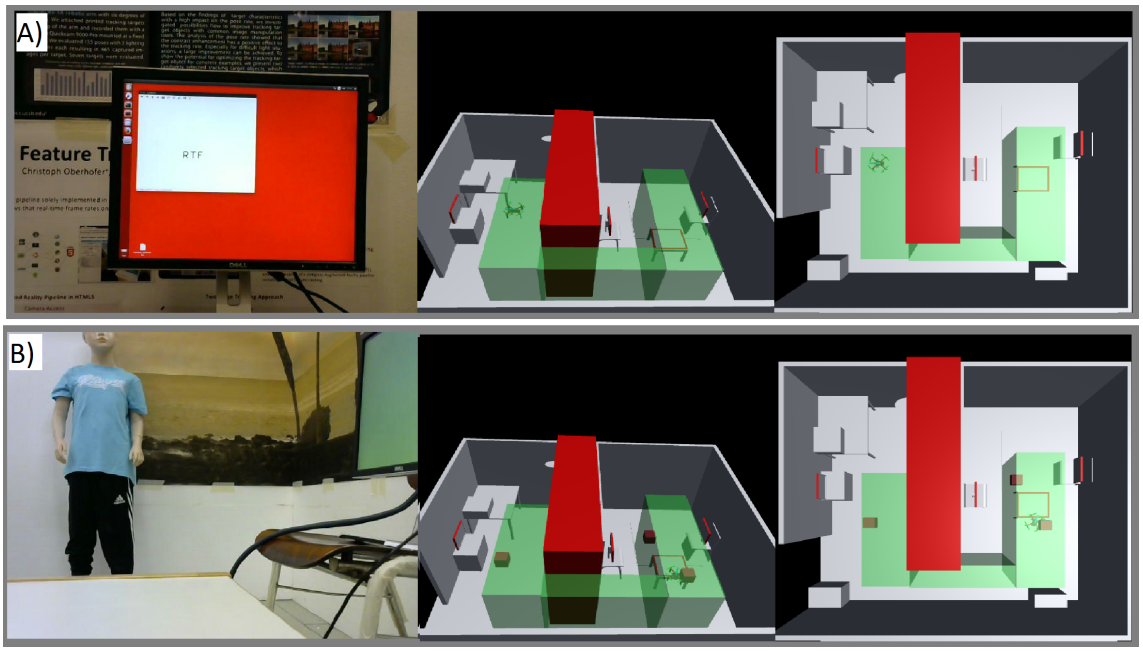


Figure 6.4: The first-person view and two additional views of the flight space were available for EGO user during both tasks. The red zone in the middle of the 3D model indicates a restricted flight zone, where drone's position is confined to remain inside boundary, while the green zone delimits the allowed flight space. (a) The user's view while engaged in the screen-reading task. The model shows three screens with red border, but only two of them are active per user. (b) The user's views during the drone positioning task. Cubes in 3D model indicate the target positions to be reached by the drone with a tolerance of 10 cm.

"slide along" the artificial boundaries. Another safety mechanism allowed the joypad user a simple and safe transition between the rooms. Once the user approached the narrow corridor between the rooms, the drone was automatically transported to the other room. We did not impose any limit in z-direction, so the user was able to safely transit between the rooms at any flight height.

6.1.1.6 Head-mounted display

The pilot interface runs on the HoloLens. Its tinted visor holds transparent combiner lenses, in which projected images are shown to the user. We rely on the built-in SLAM system of the HoloLens to provide continuous self-localization. In order to register the localization data reported by the HoloLens with the Optitrack coordinates (OC), we use a Vuforia tracking target. The tracking target is placed on the floor in front of the occluding wall, which corresponds to the plane $Z = 0$ in OC. The transformation between OC and tracking target was calibrated offline. Using the Vuforia SDK for HoloLens, we obtained the transformation from the origin of the HoloLens SLAM tracking to the tracking target at startup time and concatenated to the OC transformation. Thus, a drone pose reported

in OC can be transformed into HoloLens coordinates.

6.1.1.7 X-ray vision

We apply AR X-ray vision while providing the user with an exocentric interface for nearby remote scenes. We use the Unity 3D game engine for rendering the scene geometry on the HoloLens. A stencil masking technique is applied to render X-ray visualization only where the virtual geometry is observed through the virtual hole in the wall.

Images for first-person view are streamed from the drone-mounted camera as MJPEG, annotated with the drone‘s pose when the frame was taken. The MJPEG is decoded and uploaded as a texture to the GPU of the HoloLens to generate the Mixed Reality view. For each fragment displayed on the HoloLens, the texture is sampled during the shading process by projecting fragment positions in world space with the view projection matrix of the drone‘s camera. In order to eliminate virtual geometry from being rendered between the occluding wall and the user, fragments with world coordinates that are located behind the wall plane are discarded.

6.2 Experimental Results

We conducted two user studies to collect quantitative and qualitative data on the performance and scalability of our system.

6.2.1 Physical viewpoint study

First, we were interested in the users‘ spatial awareness using the exocentric viewing interface and X-ray vision, compared to a standard egocentric interface that lets the pilot control the drone with a joypad. Specifically, we studied the case in which the user is in-place investigating the occluded scene, which is close (e.g., behind a wall) but cannot be reached from the current viewpoint. To ensure a fair comparison, we supported the joypad user not only with the live egocentric video from the drone, but also with a screen-based 3D visualization of the hidden space, showing real-time updates of the drone‘s position. We formulated our hypotheses as follows:
H_1: *”Steering a drone for collecting information in distant spaces is faster with the exocentric interface than using a common joypad interface.”*
H_2: *”Steering a drone for positioning in distant 3D spaces is faster with the exocentric interface than using a common joypad interface.”*
H_3: *”Steering a drone for collecting information and positioning in distant 3D spaces is more intuitive with the exocentric interface than using a common joypad interface.”*

Study design and tasks In order to test our hypotheses, we chose interaction mode as an independent variable with two conditions: Exocentric interface (EXO) and egocentric interface (EGO). In addition, we selected completion time as a dependent variable.

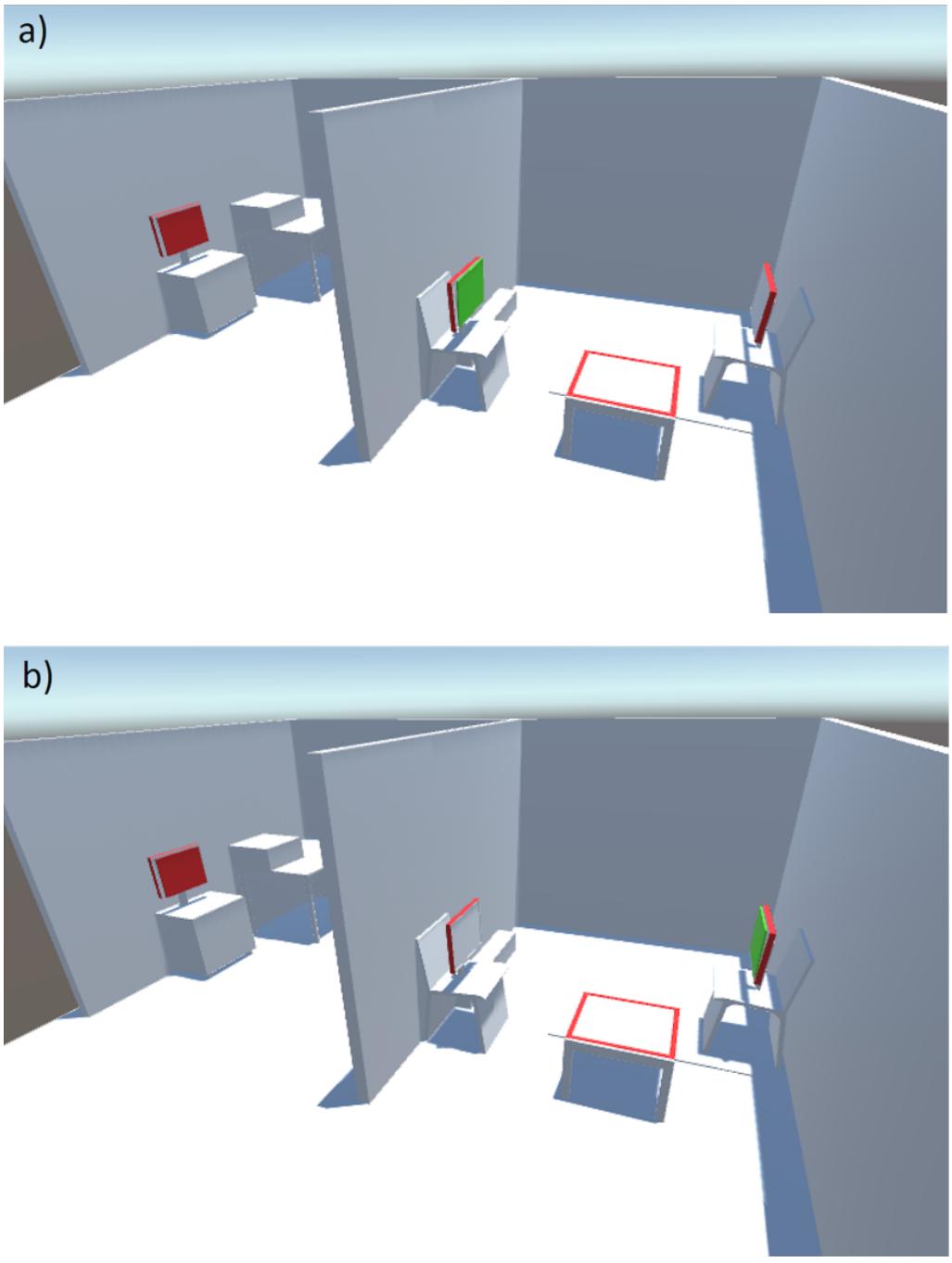


Figure 6.5: By altering the position of the green monitor, two different flight paths are generated per user.

Workload was measured using NASA TLX [89], and overall preferences of the users were assessed via semi-structured interviews. Based on a within-subjects design, participants

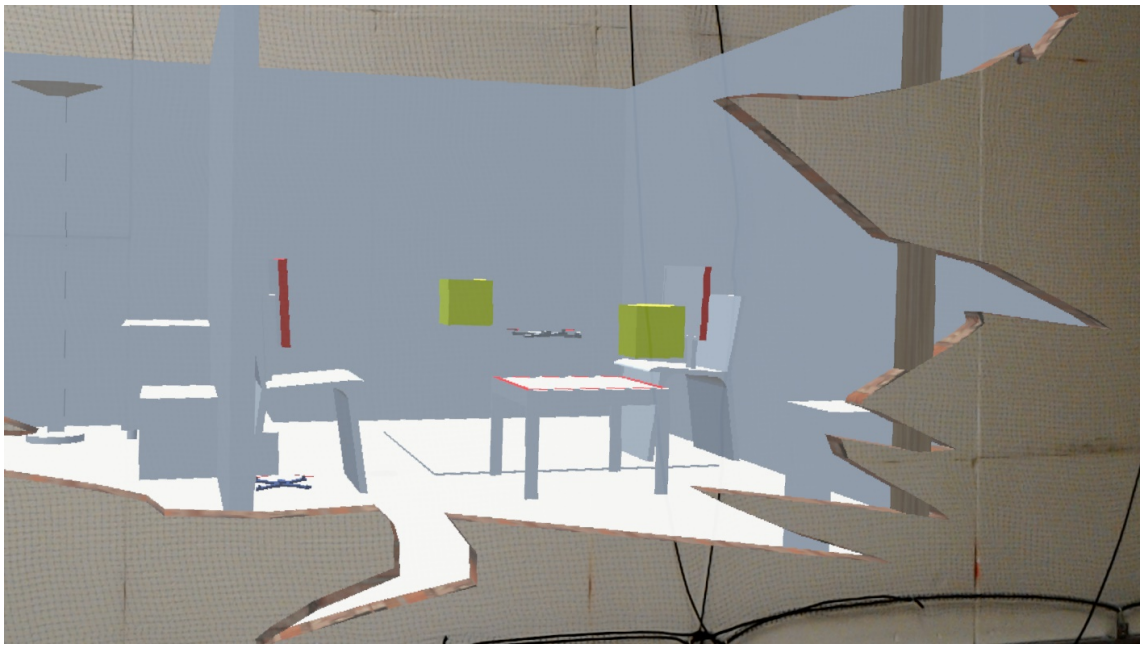


Figure 6.6: As part of positioning the drone task, target positions are visualized as yellow boxes in the EXO interface.

were given two instances of a search-and-explore task to be accomplished with either of the interaction methods, in randomized order.

Reading text on monitors We asked subjects to steer the drone with both interfaces and report random texts displayed on two monitors positioned in different places of the occluded space. The monitors showed different background colors (red and green) to uniquely identify them from an arbitrary distance. During the training, users were informed of the positions of the red and green monitors in the 3D environment model. The environment model contains three physical monitors, but only two of them were active at any time in order to necessitate different flight paths (Figure 6.5). The time spent to read from each monitor was recorded as soon as the user reported the text correctly. We asked participants to use the gaze-to-see interaction technique, and we suggested to additionally use the overview-and-detail technique in EXO.

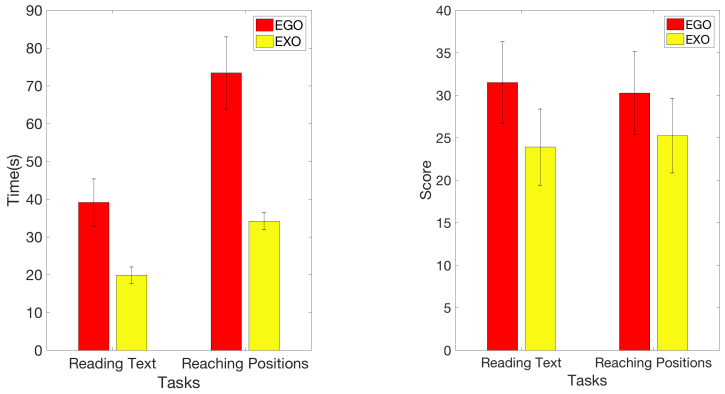
Positioning of the drone In this task, participants were expected to position the drone at three known target locations, which were visualized as boxes in the 3D models shown in both interfaces (Figure 6.4b and Figure 6.6). We logged the time spent to visit the target locations, whenever the system reported that the drone approached a target to within 10 cm tolerance. As the task involved accurate and fast positioning of the drone for this tasks, we suggested to the EXO users to use the pick-and-place technique.

Participants Ten participants (0 female, $\overline{X}$ =23.1 (sd=2.07) years old) volunteered in our experiment. All of them had extensive experiences with mobile devices, none was a regular drone pilot.

Experimental setup Participants performed the tasks while standing in front of a wall completely occluding the flight zone. In the EXO condition, participants wore a HoloLens for seeing through the wall. In the EGO condition, a joypad was used to steer the drone, while a monitor (19 inch) was used to display the video stream delivered by the camera of the drone. EGO users were also provided with 3D views of the flight zone from different perspectives (top and top-side view), displayed on a second monitor (15 inch) (Figure 6.4). A laptop was used to record the participants‘ qualitative and quantitative input during the experiment.

Procedure Participants were brought to the participant zone and informed about the setup of the experiment environment without giving detailed information about the flight zone. After the briefing, we assessed their demographics and explained how to use both interfaces. Participants were allowed to practice both interfaces, until they expressed confidence to use them.

Participants were asked to accomplish the tasks in randomized order, to eliminate training effects. For the text reading task, the position of the green monitor was changed to alter the flight path from the first to the second condition. After finishing each task with one interface, participants filled in the NASA TLX. Upon the completion of all tasks for both interfaces, participants filled out a preference questionnaire, and a semi-structured interview was conducted. Sessions lasted ≈50 min.



(a) Time Measurements. **(b)** NASA TLX scores.

Figure 6.7: Results. a) Average time spent on the tasks with our two interfaces. EXO users performed much faster in both of the tasks, with similar performance, as indicated by the standard error b) NASA TLX scores of the both interfaces for the given tasks.

Results The task completion time was evaluated using paired t-tests, and the TLX data was analyzed using pairwise Wilcoxon signed-rank tests. The t-tests revealed significant differences between EXO and EGO interface for both the reading task (p=0.001347) and the reaching positions task (p=0.002369), in terms of task completion time. On average, EXO task completion times were less than half of EGO (Figure 6.7a). In the text reading task, EXO took 19.85 seconds average (standard error 2.2 seconds) to read the texts on both monitors, whereas EGO took 39.1 seconds on average (standard error 6.3 seconds). For reaching the given 3D positions, EXO users completed the task on average in 34.2 seconds (standard error 2.3 seconds). EGO took 73.4 seconds on average (standard error 9.57 seconds).

According to the overall scores of the NASA TLX forms, for both of the tasks, users found EXO to have a slightly better usability than EGO. For the first task, users gave an average score of 24 for EXO and 32 for EGO, whereas, for second task, EXO scored 25 and EGO scored 30 (Figure 6.7b). Probably due to the small number of participants, the TLX data did not show significant differences between the interfaces. However, we found a noticeable trend in the TLX data towards the HoloLens interface for the reading task (Z=1.68, p=0.105).

Relatively high deviations in task completion time of EGO suggest that EGO requires a good 3D interpretation or experience with joypad control. In contrast, EXO seems to efficiently leverage human abilities, resulting in consistent performance, specifically for pick-and-place.

In the informal feedback during the post-interview, users commented on their preferences. All the participants stated that they would prefer EXO for the given tasks or similar task for investigation of the occluded space. Verbal feedback from the interviews for both conditions included:

- I felt more confident of being precise when using EXO, specifically using pick-and-place.
- I was feeling inside the scene with EXO.
- Depth feeling was amazing with EXO.
- I confused my orientation with EGO.
- I couldn‘t decide which view to concentrate on with EGO.
- Pick-and-place was cool, natural and accurate.
- Observing the drone from a distance, but still being able to get close to it, was pleasant.

On EGO, several users commented that the joypad axis confusion between drone‘s local frame and global frame during steering was difficult. They also sometimes confused buttons, a problem that may be overcome with longer training. Nonetheless, the direct manipulation in EXO was more easily adopted. Users also criticized the limited field of

view of EGO and reported a confusion of heights. Finally, they found that they could not easily decide which view (camera image or perspective views) to concentrate on.

On EXO, one user stated he preferred the precision of the joypad interface for collecting boxes, and several users found the HoloLens pick gesture inconvenient. However, both comments were likely caused by the unreliable gesture detection provided on the HoloLens. We hope that a future update of the HoloLens SDK will include a more stable gesture detection, which directly will make our pick-and-place interface appear more convenient and more precise. In summary, the results of our experiment allow to accept H_1, H_2, Furthermore, we partially accept H_3 based on the trend towards EXO provided by the user comments and the data retrieved from TLX questionnaires.

6.2.2 Virtual viewpoint study

The physical viewpoint experiment demonstrated the use of exocentric interaction techniques at close distances. If the drone is further away from the user, drone control by hand gestures obviously becomes increasingly sensitive to fine-motor control of the hand and to tracking errors. We empirically verified that, indeed, satisfactory drone control with gestures is not possible at distances of 20m or more.

However, since our exocentric (X-ray) interface uses the physical environment – the brick wall – only to provide relative motion cues to the user, a VR interface using the same setup is also possible. In VR, the head-mounted display is operated in a non-see-through mode, and the user is placed in a purely virtual environment, with the exception of the texture-mapped remote video stream. This setup can always place the user‘s virtual viewpoint in convenient proximity to the drone to allows direct manipulation. The VR interface is also necessary if physical proximity to the drone is not possible, for example, in dangerous environments.
We speculated that the virtual viewpoint (VV) interface would perform similar as the physical viewpoint (PV) interface (the latter is essentially the same as EXO in the previous experiment). We formulated our hypotheses as follows:
H_4: *"Users will perform similar in terms of execution time for a virtual viewpoint as for a physical viewpoint"*
H_5: *"A virtual viewpoint does not affect how a user completes the tasks, while being away from the scene"*

We tested these hypotheses by repeating the previous experiment with VV and PV conditions, as follows.

Procedure In VV, participants performed the tasks while standing completely away from the occluded space. The visor of the HoloLens was entirely covered with a blinder to disable its see through display nature and turn it into a VR device. At the beginning of the experiment, VV users witnessed an animated camera transition from their current physical viewpoint to the virtual viewpoint at the remote location. The animation gave

them the impression of flying to the target zone and landing where they had to perform the experiment.

In contrast, PV users were standing just behind the occluding brick wall like in the physical viewpoint study. Compared to the first study, we had a slightly larger flight space with the same floor plan characteristics. In the virtual viewpoint study, again ten participants (0 female, $\overline{X}$ =27.5 (sd=2.33) years old) volunteered in our experiment. All of them had extensive experiences with mobile devices, none was a regular drone pilot (different subjects from the physical viewpoint study).

Results In the text reading task, the PV condition took 48.62 seconds average (standard error 2.5 seconds) to read the texts on both monitors, whereas the VV condition took 43.37 seconds on average (standard error 2.9 seconds). For reaching the given 3D positions, PV users completed the task on average in 44.03 seconds (standard error 3.72 seconds). VV took 41.21 seconds on average (standard error 1.53 seconds). It should be noted that flight times are slightly increased compared to the first study due to the enlarged space and longer paths.
According to the overall scores of the NASA TLX forms, for both of the tasks, users found PV to have a slightly better usability than VV. For the first task, users gave an average score of 23 for PV and 26 for VV, whereas, for the second task, PV scored 25 and VV scored 27. While users commented to perceive both systems as almost identical for completing the tasks, they reported to prefer the PV condition more due to its see through-visualization capability.

The results let us accept H_4 and H_5.

6.3 Discussion About Limitations

We propose using real scale interactions for steering remote drones. This enables simple control of the drone with low cognitive effort. Based on the feedback of users and the quantitative results of our experiments, we believe that pick-and-place interaction is useful for quickly positioning the drone when fully automatic navigation is not enough. While wearing the HMD, users have stereo vision to perceive depth. In addition, users can quickly change their viewpoint by simply moving around in a natural way to understand where an object is located in 3D. In contrast, a traditional desktop interface requires several scene manipulations to understand the 3D position of an object in the scene, especially when the object is floating in the air. Simple and natural exploration of the position of the drone in 3D space enables quick understanding of spatial relations, which is a fundamental requirement for navigating the drone in 3D.

Our pick-and-place technique uses a single target point to position the drone. While we could continuously sample points along a path defined by the user, we restrict the number of waypoints to a single start point and end point to ensure a precise placement and to avoid unnecessary drone motion. Mapping any user motion directly to the position

of the drone would not allow the user to search for the final position, while the drone follows the user‘s hand motion.

While pick-and-place can be used to precisely place a drone in 3D space, the gaze-to-see technique can be used to continuously explore and search the environment. Gaze-to-see is a high-level, goal-oriented interaction between drone and human with low cognitive requirements. It provides a tool for quickly observing a region of interest without dealing how to position to drone.

Both of our interaction techniques outperform the traditional egocentric interface for controlling a drone. Note that the significant time difference observed between our experimental conditions are not the result of different reaction times, such as the time spent on moving the head when wearing an HMD versus pressing a button on joypad. The differences can rather be largely attributed to the user‘s efforts towards fine-tuning the position of the drone to solve the task. For example, finding the correct pose for the drone to read a small text clearly while experiencing motion blur during the movement phase takes more time with EGO. In contrast, EXO users can easily assume a convenient pose thanks to gaze-to-see technique.

Apart from the motion blur, no text rendering artifacts were disturbing the EGO users, as can be seen in (Figure 6.4a). In contrast, the EXO users experienced both motion blur and slight artifacts due to the limited resolution of HMD (Figure 6.3). We expect that, with better HMD quality, the advantages of EXO may even be more pronounced.

Similarly, during the positioning drone task, EGO users had difficulties understanding if the drone was at the correct position from the given perspective views, whereas EXO users quickly identified the right position by virtue of the stereoscopic view.

Despite the good performance of EXO, we noticed a number of limitations during the experiments, which we describe in the following, along with recommendations for overcoming them based on our experience with the system.

Limited resolution. Our placement precision depends on the distance. As the drone moves away from the user, the increased distance affects the precision of pick-and-place. In addition, when the surface is far away from the user, it is hard to gaze at it. This provides a challenge for selecting the drone with pick-and-place interaction, and it makes it harder to position the drone in front of the right surface during gaze-to-see interaction. This limitation arises, as humans cannot keep their head stable at millimeter-level accuracy. These limitations are solved when the user is virtually teleported to a viewpoint close to the drone, as demonstrated by virtual viewpoint study. In fact, the virtual viewpoint technique can be seen as a generalization of the overview-and-detail technique. The user can always use the VV mode to virtually move closer to the drone and thus increase the precision. The blinder on the visor may not even be necessary, as implied by user‘s preferences.

Projection error. When the outside in tracking is not precise enough, misregistration causes the projected images do not line up properly with the 3D model. In addition, if the poses are not synchronized with the camera images, the error is further increased.

However, these problem can be overcome by better tracking, ideally incorporating dense reconstructions obtained in real time from a drone equipped with suitable sensors, such as structure-from-light sensors or stereo cameras.

Tracking error. Depending on the tracking accuracy of the system, the drone may position itself slightly off the target destination, although the results would be still visualized as if the drone was at the correct location. During tasks requiring accurate spatial positioning, such as drilling a hole at the right spot, the user may be misled. A hybrid interface showing both the exocentric synthetic view and the egocentric video stream side by side may partially alleviate this problem.

Reconstruction error. Gaze-to-see can be strongly affected by wrongly estimated surface normals, if the 3D model is automatically reconstructed using structure from motion algorithms. However, many exploration tasks do not require photorealistic rendering and tolerate heavy low-pass filtering of normals to suppress unwanted outliers.

Eye calibration error. Like any ray-picking technique, pick-and-place performance is affected by eye calibration. Without a good estimation of the eye position, any deviations of physical eye and virtual camera will be magnified by the projected distance, letting the picked virtual object drift from the hand after some displacement. During our experiments, we noticed that users coped with such situations by simply releasing their grip and quickly re-picking the drone, essentially improvising a form of clutching to minimize the aggregation of unwanted drift.

3D interaction. The mathematics of scaled world grab imply that when the user moves an object away from be body, movement precision will drop quickly. As a remedy, users can re-adjust their virtual viewpoint to move closer to the target location or look at the drone from a different perspective in order to control the drone more precisely. Likewise, if surfaces face away from the user, gaze-to-see requires first assuming a rotated virtual viewpoint to look at the target position.

Aerodynamic restrictions. Aerodynamics of the aerial robot restrict it from quickly adapting into a new given position. Therefore, gaze-to-see interaction technique had to be limited to discrete position commands instead of continuous.

Selecting the remote scene. Assuming a virtual viewpoint is natural for immersive VR users, while AR user must switch from their physical viewpoint to a virtual one. This can lead to confusion between real and virtual objects. The overview-and-detail technique mostly avoids such confusion, but introduces the restriction that users can only move closer to a point they are already gazing at. While this is sufficient for a number of tasks, choosing a new viewpoint relative to gaze has clear limitations. In particular, gazing becomes less precise and more difficult at larger distances.

However, common techniques such as world-in-miniature [156] (WIM) can be used to easily overcome this limitation. Using a gesture, users can obtain a miniaturized copy of the scene in front of them, in the same orientation as their current viewpoint. It is straightforward to apply scene manipulation techniques from traditional desktop interfaces to a WIM. Users can rotate the WIM towards the desired view and apply clipping or

transparency to expose interior structures. They can apply exocentric selection of movement targets in the WIM rather than in the egocentric perspective. In case of a rescue operation, the use of a WIM extends towards a remote control center overview of multiple drones and rescuers from an exocentric perspective.

7

FAR Distance Teleoperation Utilizing Virtual Scenes

Contents

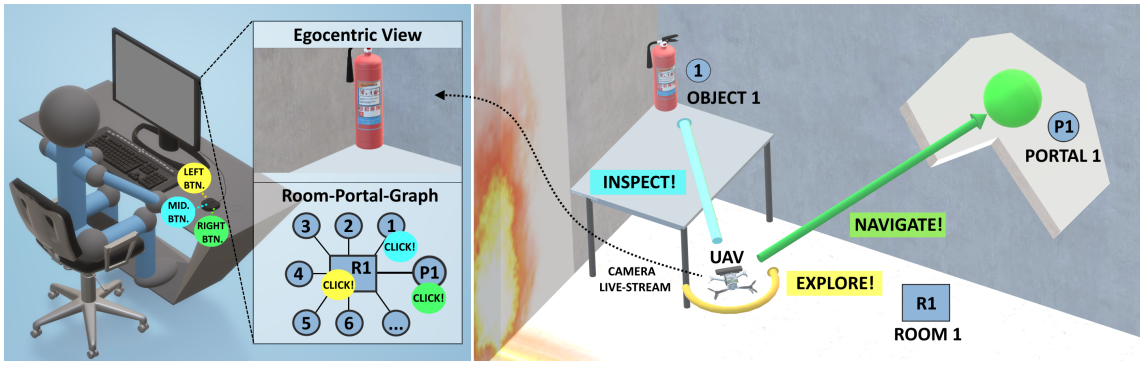


Figure 7.1: High-Level teleoperation system. (left) The room-portal graph displays an interactive topological view of an indoor environment, created in real-time, to facilitate automation. (right) Conceptual illustration of the aerial telerobot, implemented as unmanned aerial vehicle (UAV), and its according high-level tasks.

Teleoperation of small-sized aerial robots in indoor environments is important for applications like search-and-rescue or exploration missions. A recurring problem in such applications is lack of situation awareness and consequently decreasing overall task performance ([157, 158]).

One important aspect is that with an increasing amount of scene details, operators struggle to comprehend the visualized information of the teleoperation system [159]. While it is required that the system presents the information in a way that does not overwhelm

the operator, also the LOA play a crucial role. Increasing autonomy of the system can improve operators task performance by reducing their mental load. The goal is to free up the operators to be engaged in other important high-level tasks [160], such as navigation or identification of victims or hazards. However, related work has shown that true full autonomy is still hard to accomplish for complex missions [103]. This emphasizes difficulty of an optimal level of autonomy for a teleoperation system. As a tradeoff, approaches were introduced in which operators can explicitly adjust the autonomy of the system to the desired level [161, 162]. Unfortunately, such approaches typically require a handover to low-level demanding tasks [163]. While trading off task automation and manual control is task-specific and remains non-trivial to date, our system, on one hand, suggests to automate all low-level tasks. On the other hand, high-level tasks can be accessed via an interactive scene with reduced details. Yet, the question remains how such a system effects aerial exploration missions in a real-world setting.

To this aim, we introduce a fully working teleoperation system. The system uses a small-sized aerial telerobot to perform the challenging task of indoor exploration (Figure 7.1). In particular, our system is capable of: (i) indoor navigation in the presence of narrow spaces and wind-turbulence without collision; (ii) automated repetitive exploration and navigation of structured spaces; (iii) detection and navigation of constrained spaces, like narrow gateways or windows, without collision; (iv) and detection of OOIs, like victims or fire extinguishers. To relieve the operator, the system automates all low-level mission-critical tasks (see Figure 7.5). However, we allow the operator to override non-mission-critical, high-level objectives. This results in a design where the system usually runs at the highest LOA (*highest autonomy*), but can be effectively supervised at collaborative level (*high autonomy*) if necessary (see Table 7.2)[1].

The operator supervises the teleoperation system using a multi-view GUI which consists of a traditional egocentric scene view, an exocentric scene view, and a complementary topological graph view of the scene, called the room-portal graph (RPG) (see Figure 7.2). The RPG represents the scene as a subdivision of rooms connected by portals, creating a clearly distinguishable spatial structure of a typical indoor environment. The RPG reduces scene details and allows fast comprehension of important scene structure and objects during an exploration mission. It is interactive and lets the operator improve time-performance and resolve system failures, for example false detection of OOIs.

To understand the task effectiveness of our teleoperation system in a real-world setting, we conducted a user study. Participants accomplished an exploration mission using our proposed system and, in comparison, using a baseline system with traditional joystick control. While results indicate increased task performance and comfort with the outcome of our experiments, our findings provide evidence that our system can better support operators with challenging aerial indoor exploration missions.

In summary, we contribute:(i) a fully working teleoperation system for aerial indoor

[1]The reported LOA are based on the ALFUS framework of Huang et al. [164, 165].

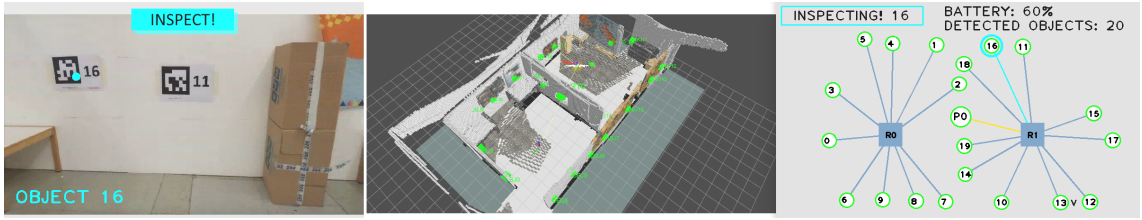


Figure 7.2: Implementation of the RPG as part of our high-level teleoperation system. It is presented during an inspection task, after full exploration of two connected rooms. (left) Egocentric virtual live view from the on-board camera of the UAV, highlighting an inspected object. (middle) Exocentric virtual 3D view of the reconstructed scene. (right) Interactive, topological RPG of the same scene, with two rooms (represented as squares), detected objects including a portal. Objects are shown as round labels with leader lines. Inspected objects are highlighted.

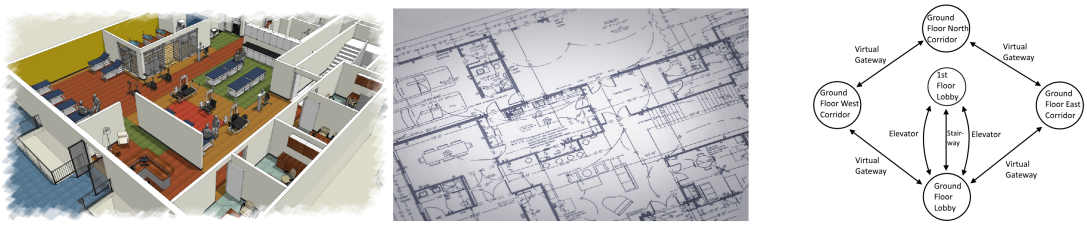


Figure 7.3: Example map views of complex office environments with gradual loss of details. left) Full 3D map view. middle) Floorplan in 2D. right) Topological view [166].

exploration missions, including a real-time interactive scene topology that enables supervisory control of high-level tasks, and (ii) the empirical evidence that such a system effectively supports human operators during realistic exploration missions.

In this chapter, we present details of design rationals and implementation of our system, as well as limitations. Followed by reporting on experimental design and results of our user study, we finally conclude our paper and propose interesting directions for future work. An extensive overview on related work, including a brief history of teleoperation and current state-of-the-art systems can be found in Chapter 2.

7.1 Design of an XR Teleoperation Interface For Indoor Exploration

The design of our teleoperation system is governed by the needs of aerial exploration. It focuses on exploration of civil buildings with constrained indoor spaces and repeating room geometry. Example representations of an office building are shown in left and middle Figure 7.3 (3D map and 2D floorplan).

Typically, an exploration mission would require to navigate inside the building and detect OOIs (fire extinguishers or trapped victims). For such applications teleoperation systems can be helpful, if disaster relief forces are not able to reach inside such buildings,

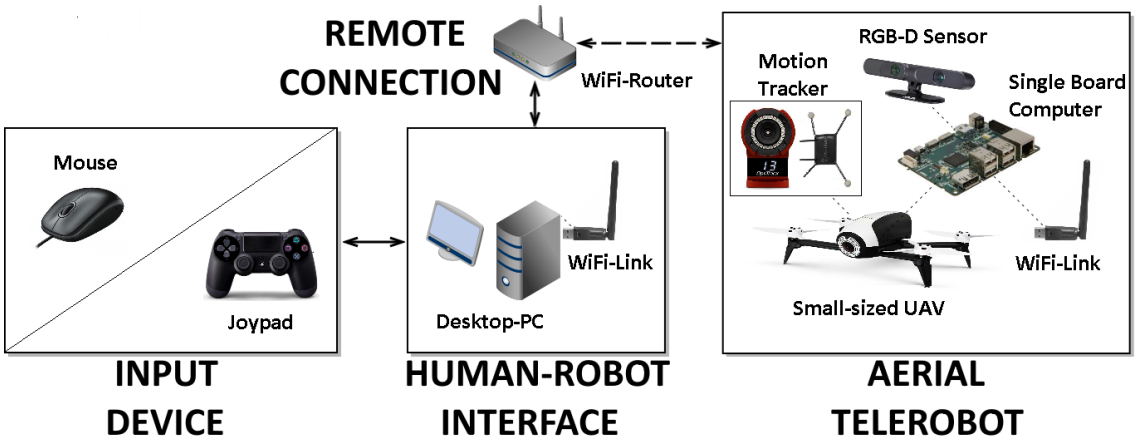


Figure 7.4: Overview of the main components of the teleoperation system design: The aerial telerobot which is a small-sized UAV and localized by a motion tracker [168]. A remote connection between the telerobot and the human-robot interface which runs on a desktop PC. The input devices used for manual control of our teleoperation system.

and assessment of the situation is required [167].Our teleoperation system uses the same main components as state-of-the-art systems (Figure 7.4):

- Aerial Telerobot: Our telerobot is a small-sized UAV, holding various sensors (cameras and inertial sensors) and actuators to perform the challenging task of aerial indoor exploration. Additionally it is equipped with an onboard computer to transfer sensor and actuator data to the human robot interface via a wireless remote connection.

- Human Robot Interface: Our human robot interface includes all software components for processing the sensory data and flight-control of the telerobot. Further it holds the interactive scene topology (RPG) which is enabled by the underlying system components (Section 7.2.3).

- Input Device: The design of our system considers a simple and cost-effective input device sending manual high-level commands to the human-robot interface.

7.1.1 Teleoperated Aerial Exploration Of Indoor Environments

Indoor space is typically limited and room exploration may require passing through narrow passages or so called portals, which can be hallways or windows. As a consequence, for our teleoperation system we designed a highly mobile and rather small sized UAV as telerobot. The design of our telerobot focuses on core functionalities which are vital for indoor exploration. On a higher task-level, our telerobot provides functions for **room exploration**, **object inspection**, and **navigation of portals**. However, such high-level tasks entail a variety of low-level functionalities with increased complexity (Figure 7.5). Also, it is

important to distinguish between mission-critical tasks and **non-mission-critical tasks**, whereas **mission-critical tasks** have to be solved by the teleoperation system under all circumstances and at all time. If the system fails with a mission-critical task this could lead to serious damage of the telerobot and potentially end the overall mission. For our system design we define the following low-level mission-critical tasks which are vital for indoor exploration:

- Localization: The telerobot has to be able to localize itself against the environment at all time. A failure in self-localization would typically result in that the telerobot collides with its environment.
- Landing/Take-off: Based on a robust localization and proper control of speed and acceleration, the telerobot provides assistive features like take-off and landing.
- Hold Position: Due to the turbulences that occur in the indoor environment, our design has to consider methods for stabilizing the telerobot while in-air and rejecting disturbances. Disturbances can occur due to flying close to obstacles or passing through portals.
- Collision-free Path Planning: Path- and motion-planning ensures collision-free navigation inside the indoor environment. It is vital if navigation between objects is required (waypoint-based navigation).
- Live-Video Stream And 3D Mapping: It is based on a robust acquisition of sensory data, whereas abstraction into a topology requires the 3D data. Since 3D Mapping also provides minimum understanding of the remote scene to the human operator, these tasks must not fail.
- Portal Detection And Evaluation: We detect portals by analyzing single depth images, in which we recognize the contour of the portal in 3D and estimate size (minimum diameter) and the normal vector of the contour at the geometric center. Once a potential portal is detected, it must be evaluated correctly.

On top of the low-level tasks, we introduce high-level non-mission-critical tasks. These are difficult cognitive tasks, where a human operator can still improve overall performance. The high-level tasks can be summarized to one automated indoor exploration task, autonomously executed by the system at highest LOA. In particular, the system uses an automated search strategy to explore one single room, identifies objects and portals on the fly and is able to navigate the safest portal. Implementation details are given in Section 7.2. Non-mission-critical tasks are considered as the following:

- Room Exploration and Abstraction: Room exploration is based on a state-of-the-art rapidly exploring random trees (RRT) exploration algorithm. On lower level this requires collision-free navigation. In parallel the system has to tackle the challenging

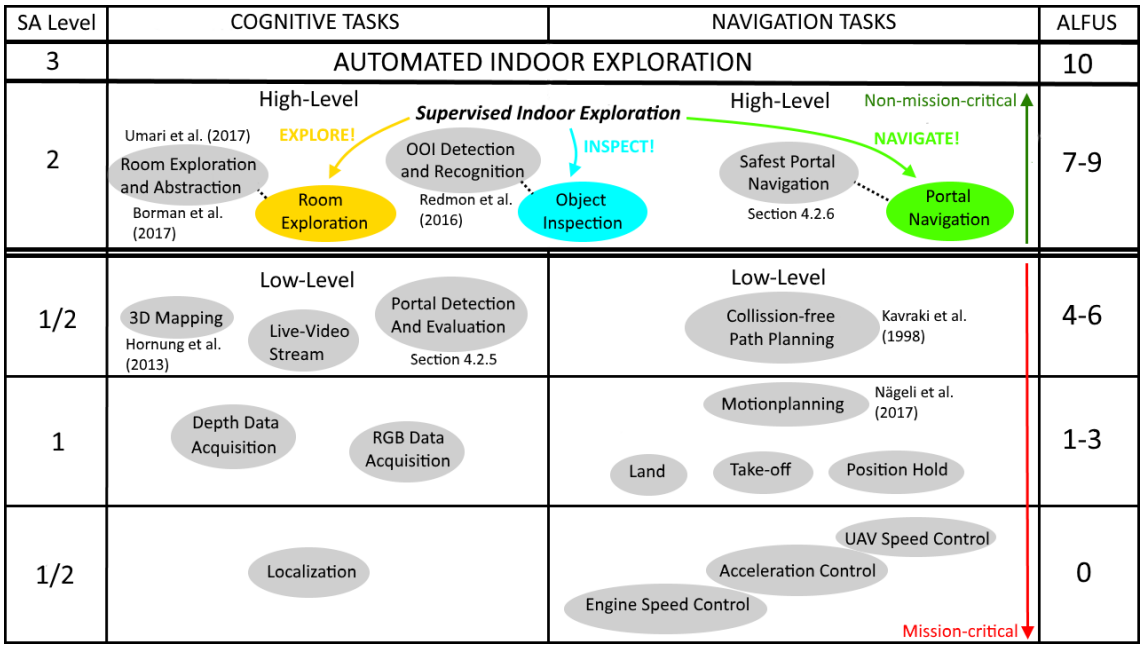


Figure 7.5: Overview of high-level and according low-level sub-tasks that have to be solved by our teleoperation system. Noteworthy is the separation between cognitive- and navigation-tasks, definition of mission-critical tasks, levels of situation awareness [158], relation to the ALFUS [165] and the recovery behaviours, triggered by high-level interactions of the operator (Explore!, Inspect! and Navigate!).

tasks of detecting portals for navigation and OOIs. Once a full room is explored it is abstracted into a node and added to the scenes topology.

- Object Detection And Recognition: For object detection our system design aims for using state-of-the-art real-time detection algorithm.

- Safest Portal Navigation: After room exploration the system navigates the safest detected portal.

However, if the system fails with one of the high-level tasks, the operator can intervene by commanding high-level recovery behaviors in the GUI of the human-robot interface (Figure 7.1). In detail the operator can: trigger a simplistic but robust search strategy (**Explore!**), select a preferred portal over the other (**Navigate!**) or correct for object detection by close inspecting the object and/or registering the object manually (**Inspect!**) (Figure 7.7). Noteworthy is that our Inspect! command is motivated by the overview and detail paradigm, also used in the work of Seo et al. [169] to improve effectiveness of teleoperation.

7.1.2 Human-Robot Interface

Our human-robot interface is designed to support the human operator during teleoperation. Core design aspects are typical essential tasks during aerial indoor exploration, limitations of the telerobot and usage of an untethered remote connection.

7.1.2.1 Levels of Autonomy And Approaches For Control

Our proposed scenario for aerial indoor exploration involves rather complex tasks, like object recognition and path planning. Such tasks have to be executed at the same time and involve mission-critical tasks like collision-free navigation. Due to the complexity of the tasks, the design of our system assumes that true full autonomy is not feasible. For our scenario a human operator is necessary to support the system with complex cognitive tasks on higher level. However, these tasks are non-mission-critical. The purpose is to avoid the lumberjack-effect and avoid sudden passing of control to the operator. If tasks fail on higher level, the telerobot is not damaged and able to continue with the overall mission. As a consequence, in accordance to related work Valero-Gomez et al. [170], we design a supervisory control approach for our system which adapts the ALFUS framework [165]. Details about task definitions, high-level interactions to supervise the system with recovery behaviors and relation to LOA are presented in Figure 7.5 and Table 7.2. Importantly, hazardous regions in challenging indoor environments require the usage of an untethered remote connection. Consequently, potential sudden network dropouts and time delays during control strongly motivate supervisory control.

7.1.2.2 Graphical User Interface

The user interface is one vital design aspect of our full high-level teleoperation system. Moreover, its design is based on the complex interplay with the underlying system components, whereas the overall goal is to improve teleoperation during aerial exploration missions. Yanco et al. [171] summarizes core design aspects to improve overall task performance which are 1) using a map; 2) fusing sensor information; 3) minimizing the use of multiple windows; and 4) providing more spatial information to the operator. In addition, [172] discusses several window layouts in a standard paradigm. Besides of the rich variety of designs found in related work, a very common window layout is placement of exocentric map views on the bottom half of the screen whereas egocentric live camera views are placed on top.

The design of the GUI is also based on a standard layout, whereas we keep all view windows at equal size. It includes a traditional egocentric live view on top and a 3D map view on the bottom half of the screen. The purpose is to provide a minimum of spatial understanding to the operator. For the 3D view we use grid-map representations as they are a more robust in the presence of network delays and sudden dropouts [84]. We place the view of the interactive scene topology (RPG) side by side to the traditional views

to avoid occlusions or switching. The RPG is motivated by exploration of structured human environments, which can have complex and repetitive geometry (e.g. office buildings). While the structure of such environments motivates a topological representation of the environment, related work clearly supports the use for navigating robots. Other motivational aspects are extensively discussed by Johnson [173]. Amongst other benefits, the work states that a topological representation is suitable for telerobots which have to navigate reliably from one place to another without the need of building an accurate map. While this is not valid during exploration of the environment, clear benefits occur for repeated navigation from one object to another after exploration. Johnson [173] also points out that the topology supports affordances (opportunities for interactions) and poses a human-like representation of the environment. Based on the concept of an ecological interface [172], we designed visualization of objects that support affordances, but do not overwhelm the operator [159]. Consequently, we define general OOIs, which are detected during the exploration mission and highlighted in the topological scene view. Based on these considerations and avoiding to overwhelm the operator with too rich scene details in the traditional views (left and middle Figure 7.3), our design leads to the RPG which poses an interactive topological 2D map of the indoor environment. Implementation details can be found in Section 7.2.2.

7.1.3 Input Device

The design of our high-level teleoperation system includes a topological scene view which is represented in 2D. Because the topology supports affordances, we make OOIs explicit for interaction during flight. Motivated by the 2D representation, we consequently selected a 2D mouse as input device. Besides of being robust and simple to setup (e.g., no need for calibration), other advantages are shorter pre-training phases and cost-effectiveness [174]. The mouse holds three buttons, whereas the operator can trigger three high-level recovery behaviours (Figure 7.1 and Figure 7.9) of the aerial telerobot (Section 7.2.1).

7.2 Implementation of the XR Teleoperation System

To solve the challenging tasks that occur during aerial indoor exploration missions, we implemented the following components as part of our high-level teleoperation system:

- **Aerial telerobot** represented as a small-sized UAV. The UAV is equipped with a sensory setup to acquire RGB-video and depth data. The data is transferred to a desktop PC via the remote connection.

- **Human-robot interface** to facilitate control of the aerial telerobot by providing different views of the remote scene. Based on these views, the operator controls the telerobot in a supervisory manner via high-level interactions. It further holds the underlying system components that are responsible for flight control, 3D mapping,

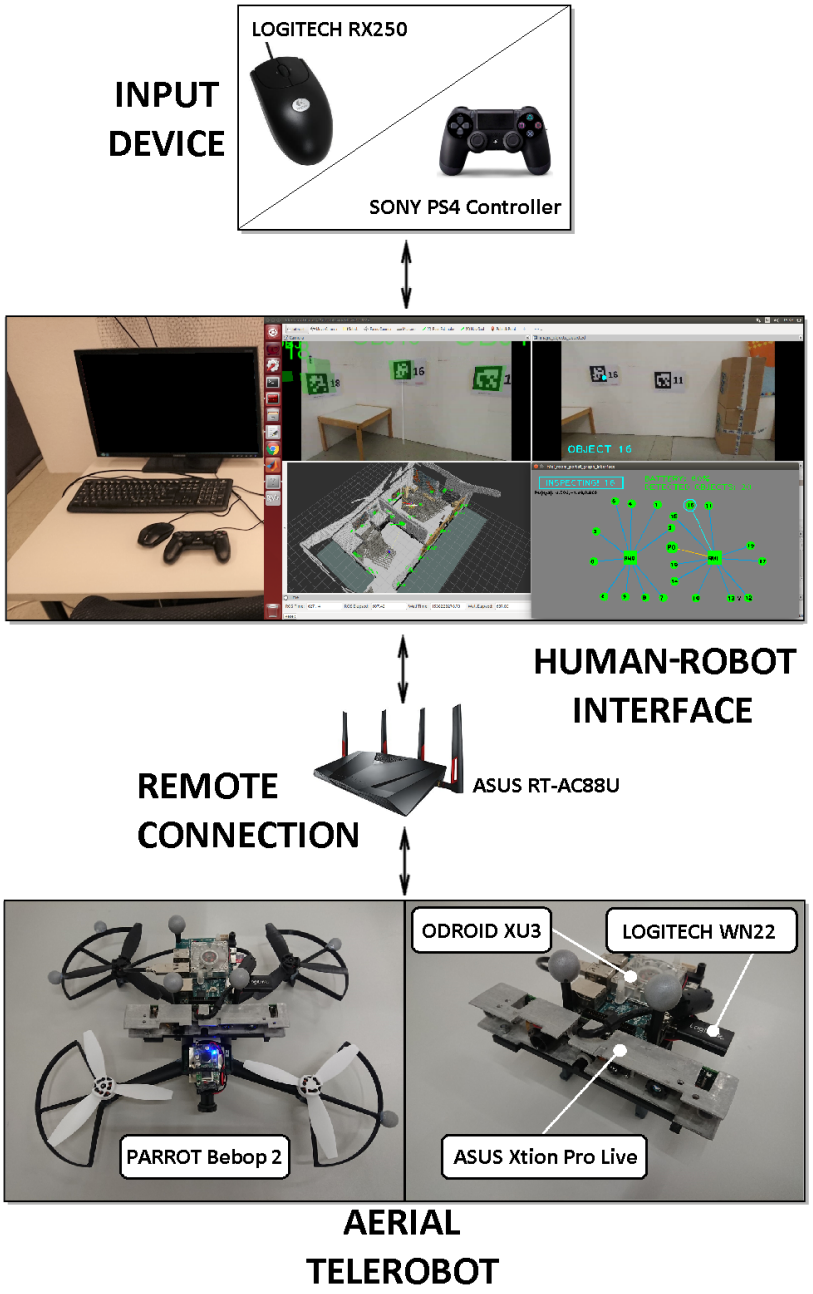


Figure 7.6: Implementation of the main components of the high-level teleoperation system: The UAV based on a Parrot Bebop 2 with onbaord single-board-computer and sensory setup. The remote connection implemented as ASUS RT-AC88U wireless router. The GUI, including the RPG, implemented on a desktop PC in ROS. The input devices implemented as a Logitech RX250 mouse and a PS4 controller.

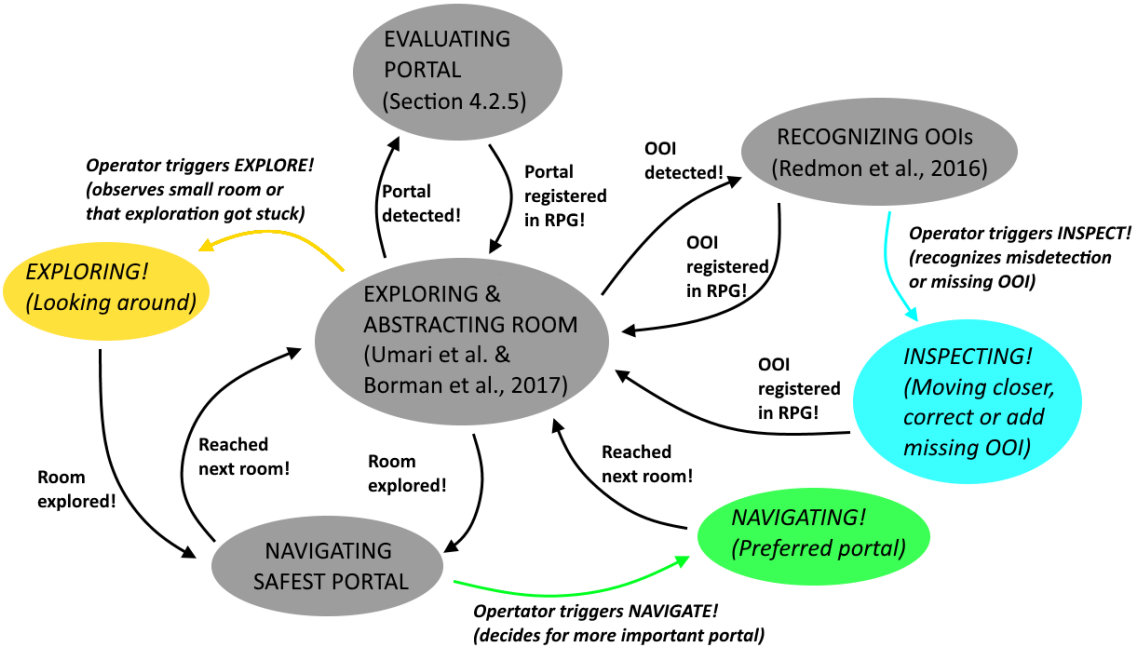


Figure 7.7: State diagram of the software framework of our teleoperation system.

abstraction, detection of portals, and object detection in real-time. Remarkably, the components are vital for enabling the interactive scene topology of the human-robot interface. Thus, they are also essential to enable the high-level interactions (Explore!, Inspect!, Navigate!).

- **Input device** that sends manual inputs to the human-robot interface. It is implemented as a simple and cost-effective mouse to interact with the RPG. To compare our system against traditional controls, we use a joypad controller for our user study (Section 7.4.1).

While the physical setup of the teloperation system is shown in Figure 7.6, we give an overview of the software implementation (represented as state diagram), high-level interactions (Explore!, Inspect!, and Navigate!) and according recovery behaviours in Figure 7.7.

7.2.1 Aerial Telerobot (UAV)

For the aerial telerobot (UAV) of our system we use a modified Parrot Bebop 2 [124]. It is compact, suitable for narrow indoor spaces and offers open-source drivers [145] for low-level flight control. For reliable experimentation, we attach retro-reflective markers for outside-in localization using an Optitrack Flex 13 infrared motion capturing system. An overview of the physical setup is shown in Figure 7.6.

With all on-board sensors attached, the outer dimensions of the UAV are $32.8 \times 38.2 \times$

$25cm$, and it weighs $700g$, with flight times of up to approx. 10 minutes. On top of our UAV, we mount a customized RGBD sensor rig ($250g$), consisting of an ASUS Xtion Pro Live sensor ($FOV_{hor.} = 58°$, $FOV_{vert} = 45°$) and a Logitech WN22 WiFi stick, connected via USB to an ODROID XU3 single-board computer. During our experiments, the UAV was navigating at a default flight height of $z_{takeoff} = 1.25m$.

7.2.2 Human-Robot Interface And Input Devices

In this section, we give details about the human-robot interface which enables the operator to high-level control our teleoperation system. As one vital component it holds the RPG as interactive scene topology which is created, based and, thus, strongly dependent on the complex interplay of its underlying system components (Section 7.2.3). While we motivate a supervisory control approach in Section 7.1.2.1, in the following we discuss implementation details and the correspondence to the LOA. Furthermore, we give details about the traditional baseline system with a joystick as input device in Section 7.2.2.2. It runs on low autonomy level, and the operator has manual control over the system. We compare the two different systems against each other and report on results in Section 7.4.1.

7.2.2.1 High-Level Teleoperation (RPG Condition)

High-level teleoperation of our system is enabled by the RPG to let an operator effectively supervise our system on high LOA (Table 7.2). We intentionally do not provide low-level access, so that the operator is not burdened with demanding mission-critical tasks (ALFUS 1-6). This also means that the system must achieve all mission-critical tasks even in a challenging indoor environment. The system usually operates on highest LOA (ALFUS 10), but we let the operator switch to a lower collaborative level (ALFUS 7-9), if supervision is required. This is particularly relevant if the underlying system components do not perform satisfactorily, e.g., when object recognition fails [175].

For the RPG (Figure 7.2), we combine a traditional egocentric view (on-board camera of the UAV) with an exocentric 3D map view. The views include visual aids for current pose of the UAV, view frustum of the onboard camera, the online reconstructed 3D environment and invalid flight zones. The purpose is provide a basic spatial understanding of the scene. We extend the ego- and exocentric views with an interactive topological view, the RPG. It consists of rooms (nodes) and portals (edges) to other rooms or OOIs (e.g., a fire extinguisher or a victim). OOIs registered in the RPG are highlighted in real-time. Once an interactive OOI is highlighted, the operator can use 2D mouse inputs to supervise the system by a reduced set of high-level interactions (**Explore!**, **Navigate!** and **Inspect!**). This triggers recovery behaviours (Figure 7.9 and implies switching from highest LOA (ALFUS 10) to collaborative level (ALFUS 7-9).

The Explore! command lets the system more effectively uncover smaller rooms. During this task, our system autonomously detects OOIs and adds them as interactive nodes to the RPG topology. If a false detection occurs, the operator can use the Inspect! command

to move closer. If one of the detected objects is selected, a safe path is generated between the current location of the UAV and the object. After the system navigates close to the false detection, the operator can inspect the situation in a close-up egocentric view and determine further action. During exploration of a room, also portals which are safe for navigation are detected and highlighted (Section 7.2.3.4) automatically. Detected portals add a new node and edge to the RPG. In case of multiple detections, the operator is able to select a preferred portal to trigger navigation into the adjacent room (Navigate!). A picture sequence of the recovery behaviours is shown in Figure 7.9, whereas we evaluated our system during real-world flights. Details about the physical setup of the aerial telerobot are discussed in Section 7.2.1.

The goal of the RPG is to provide a topological map that is a human-like representation of the environment. Since it provides natural interactions for commanding the system and describing the environment, it facilitates and eases human-robot-interaction [173]. Moreover, its purpose is to reduce scene details in the presence of cluttered traditional views (left and middle Figure 7.3). However, the concept of the RPG has also limitations which we detail in Section 7.3.

7.2.2.2 Traditional Direct Teleoperation (JOY Condition)

To compare the effect of our high-level teleoperation system against a state-of-the-art baseline system, we implemented traditional joystick controls. For our study we define it as condition JOY. In this condition, the operator uses a joypad to command the UAV at lower ALFUS (Table 7.2) with a high-level of interaction on sides of the human operator (ALFUS 1-3). At this level, the system takes care of automatic take-off, position stabilization and landing. Besides, the operator is also responsible for mission-critical tasks.

To achieve fair comparison against the RPG, we added a visualization of flight zone boundaries to help the operator prevent collisions. The boundaries are displayed in the horizontal plane via a color-coded surface at the height of the UAV. Operators must not exceed this indicated boundary and get color-coded feedback, if they are close to exceed the maximum flight height. The surface turns orange, if the UAV is close to the height boundary, which means the distance of the geometric center of the UAV to the upper boundary is smaller than the height of the UAV. The surface turns red, if the distance is smaller than half of the height of the UAV, indicating that the operator has to steer the UAV downwards immediately.

The joypad is used in MODE-2 configuration, allowing the operator to give direct motion commands. In this configuration, the left rudder stick controls translational and rotational velocity of the z-axis of the UAV, and the right rudder stick gives acceleration inputs along the x-axis and y-axis of the UAV.

7.2.3 Underlying System Components

This sections describes the underlying components of our high-level teleoperation system. They are implemented as part of the human-robot interface on a Desktop PC and responsible for exploration, flight planning and navigation, 3D mapping of the environment, and highlighting of OOIs. Since they even enable the RPG as interactive scene topology, the effectiveness of our full system strongly depends on their performance. Thus, they must be emphasized as core for interaction. The aerial telerobot supports with automated indoor exploration and the human operator can trigger recovery behaviors via the RPG. Subsequently, if a non-mission-critical task fails, time performance could be improved. The recovery behaviors are designed in a supervisory manner so that the human operator can effectively supervise the system with difficult tasks on higher-level. Their purpose is illustrated throughout the following use cases:

- Explore: After take-off, the UAV autonomously starts exploring the current room using an RRT-based exploration method [176]. If the operator decides that the room seems rather small or the exploration fails to fully explore the room, the operator can on demand trigger a simple recovery behavior. In that case the UAV explores the local environment by flying a circular trajectory. Once a room is fully explored we use the implementation of Bormann et al. [177] for room-segmentation.

- Inspect: During exploration of a room, the telerobot autonomously detects portals and OOIs, like victims or fire-extinguisher. However, if the operator feels that an object was misdetected, the operator can command the telerobot to move closer to a detected OOI or portal for verification.

- Navigate: During room exploration, the telerobot detects portals which are safe to navigate. However, if multiple safe portals are detected, the human operator might intuitively prefer one portal over the other for navigation. In such cases the operator can manually trigger portal navigation.

7.2.3.1 Room Exploration

At the beginning of every mission, the UAV ascends to a default flight height (Section 7.4.1). After reaching the default height, the system starts to autonomously explore the local environment (Figure 7.8, Step 1). For local exploration of a single room, we use a frontier detection algorithm, based on rapidly-exploring random trees [176]. If no failure cases occur, we consider the system to work on highest LOA (ALFUS 10).

Once the UAV takes off, we start detection of local frontiers by taking into account the occupancy map constructed online. First, we project 3D occupancy information into 2D, since this helps to clearly define boundaries of a single room. We project occupied cells into the 2D map. Second, we let a local frontier detector discover valid navigation points, which are derived from a rapidly growing random tree biased toward unexplored regions

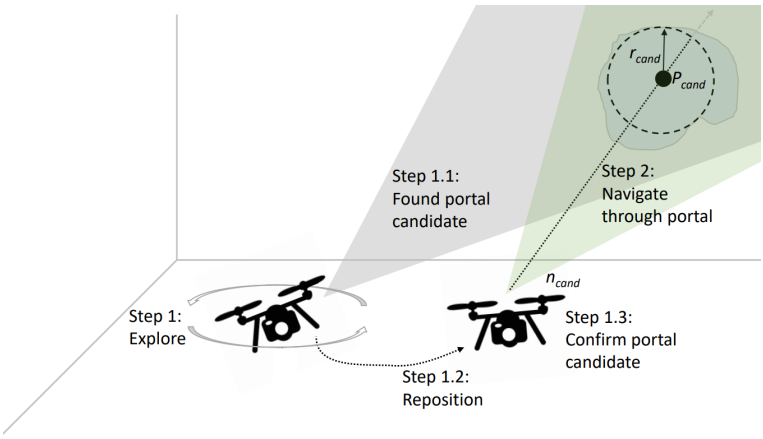


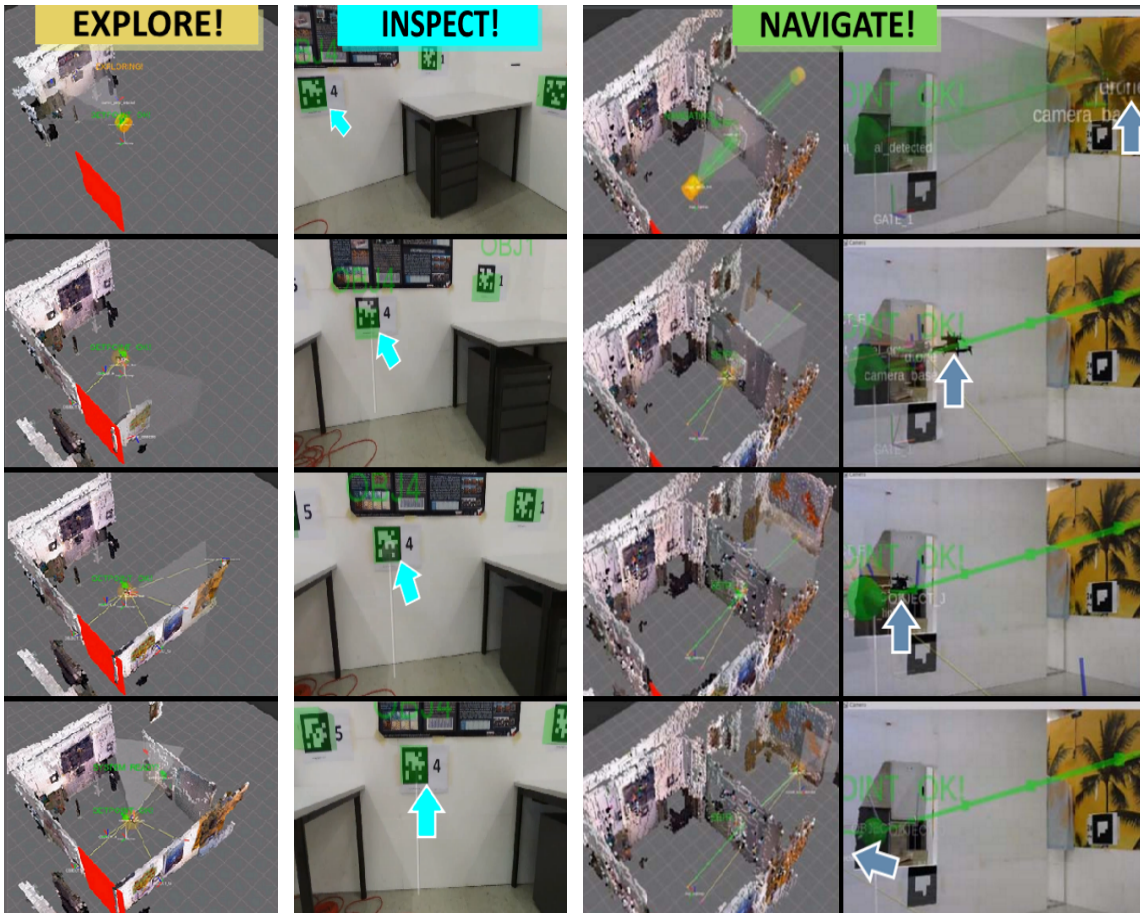
Figure 7.8: Method for detecting and evaluating potential safe portals directly from depth-image data. The UAV first explores the close environment and, if a safe-portal candidate is detected, positions itself to confirm that the portal candidate is safely traversable.

of the occupancy map. Third, we directly steer the UAV toward the detected point, incrementally exploring the local environment. These steps are repeated, until no new frontier points are detected and the room is locally fully explored. To abstract the local room and to further obtain room information we make use of the segmentation approach presented by Bormann et al. [177]. Note that we assume the range and FOV of our depth sensor to be wide enough to cover the close environment and detect potential obstacles, when navigating at default height. For simplicity, we assume that there are no additional obstacles between the UAV and the detected room boundaries. The operator is further able to manually override frontier detection by selecting the abstract room representation of the RPG (triggering Explore! and switching from highest- to collaborative LOA). This prompts the system to execute a more efficient circular trajectory.

7.2.3.2 Room Navigation

To enable collision-free navigation through portals from one room to another, we use a global path planning approach based on probabilistic road maps (PRM) [178]. The global path planner generates a PRM based on the occupancy grid map [139]. The PRM is represented as a set of 3D points given in world coordinates. For an example of generated paths, please refer to Figure 7.9c.

The PRM is passed to a real-time motion planning framework Gebhardt et al. [179], Nägeli et al. [180, 181]. The motion planner involves a model predictive controller (MPC), which produces smooth motion trajectories for the UAV when moving along the global path. Following a receding-horizon MPC formulation, at each timestep Δt, a locally optimal path with N steps and a duration of $N\Delta t$ is computed. This optimization problem is re-evaluated at every sampling instance T_s, leading to a closed-loop behavior. Thus, we



a) b) c)

Figure 7.9: Exocentric virtual views of the aerial robotic system during execution of the recovery behaviours. a) The UAV is commanded to explore its surrounding by flying a circular trajectory and simultaneously builds a 3D model of its environment from RGBD data. b) The UAV is commanded to close inspect a detected object for verification. c) The UAV is commanded to navigate through a safe portal to the adjacent room along an autonomously planned path, shown in green. The UAVs position is marked with blue arrows.

make use of the disturbance rejection characteristics of the MPC to stabilize the UAV during the mission. Stabilization against turbulence is necessary when flying close to objects or passing through portals. The real-time motion planner is implemented in MATLAB (Robotics Toolbox), utilizing the FORCES Pro real-time solver [146].

7.2.3.3 Environmental Reconstruction

To provide the operator with basic environmental understanding during navigation (Section 7.2.3.1), we make use of the RTABMap reconstruction framework ([141], [140]). It represents the reconstructed geometry as a colored occupancy grid map and is capable of

loop-closure detection. The grid map is created by fusing depth- and RGB-data from the onboard sensor setup of the UAV (Figure 7.6) and visualized in the exocentric view.

7.2.3.4 Detecting And Highlighting Objects Of Interest

For our experimental setup, we introduce different types of OOIs which are commonly present in exploration scenarios. These objects can be hazardous areas (location of fire extinguishers, broken power lines, gas leaks), human victims (embodied by human-like mannequins) or portals (potentially narrow passages), which connect adjacent rooms. The OOIs are automatically highlighted as virtual overlays in the GUI to direct the operators attention toward them. This requires automatic object detection and registration of the observed object positions in world coordinates. Noteworthy, we use a true relative scale between objects in the current design of the RPG. We detect objects either using the YOLO framework for object detection [182] or by simply marking them with Apriltag markers [183] during the user study (Section 7.4.1).

Indoor environments can typically be structured into wider open areas (rooms) and more narrow spaces (portals) connecting rooms [166]. During the exploration task, our goal is to detect and visualize potential portals. Making rooms and portals explicit is vital in our scenario, since they support navigation. Interactive highlighting, helps operators to get a clearer understanding of the environment and make an educated decision on which part of the environment to explore next. The portal detection proceeds as follows (Figure 7.8): During exploration of the close environment (Step 1), we detect discontinuities in the depth images captured by the RGBD sensor. If the discontinuities form a closed, bounded region with minimum radius r_{cand} and depth d_{min} (measured from the centroid P_{cand} of the entry surface), the region is selected as a portal candidate (Step 1.1). This intermediate step is necessary to ensure the portal can be safely traversed, as looking at portals from larger offset angles would result in shadowed regions behind the portals. Based on the surface geometry of the portal candidate, we derive P_{cand} and the corresponding normal vector $\mathbf{n}_{cand}$. The normal $\mathbf{n}_{cand}$ is oriented perpendicularly to this entry surface and has its origin in P_{cand}. In Step 1.2, the UAV is commanded to align the x-axis of its local coordinate frame F_{UAV} with $\mathbf{n}_{cand}$. The distance to the portal candidate d_{cand} is calculated based on the minimum radius r_{cand} and the narrower vertical field of view of the depth sensor FOV_{vert}. d_{cand} can be expressed as $d_{cand} = r_{cand} / \tan(FOV_{vert})$.

7.3 XR Teleoperation System Limitations

The teleoperation system presented in this work has also several limitations. The most important limitations are discussed in the following:

- Telerobot: Besides of there is room for improvement of our physical design (weight, size and computational onboard power), also the ability to morph and adapt to challenging environments could be added. Speaking of passing narrow portals or

gaps, also highly dynamic maneuvers [125] are currently not possible but could be interesting for future work. Another limitation of the telerobot is the exploration algorithm. While we make use of an RRT-based approach to explore a single room but at constant flight height, a more powerful approach would involve full 3D exploration. Additionally, a gimbal could help to resolve constraints with the cameras limited FOV, making room exploration more efficient.

- Wireless Remote Connection: Due to the usage of an untethered remote connection between the telerobot and the human-robot interface, typical problems could occur like limited bandwidth and sudden connection dropouts. While in-field applications would require a much more sophisticated (and expensive) setup, in our implementation we considered commodity hardware only. However, it must be stated that due to usage of a powerful WiFi router, comparably short ranges, and non-overlapping/non-populated channels the effects during the user study could be reduced to a minimum.

- Supervisory Control of high-level tasks: The supervisory control approach of our system aims for effectively resolving failures of high-level tasks. However, this is only valid if the telerobot is capable of handling all low-level mission-critical tasks without failure.

- Human-Robot Interface: An essential component of our human-robot interface is the RPG, serving as interactive scene topology. The focus of its design is to supplement traditional views by supporting affordances and reducing scene detail. Thus, overwhelming the operator should be avoided. However, several aspects could not be considered in this work. While in our current RPG design we use a true relative scale of rooms, portals and objects, we did not elaborate on different layouts of the objects inside the RPG view or adapting its orientation relative to the 3D view. We also did not yet investigate on proper placement of the simplistic 2D objects in case they overlap or on altering their shapes and size. Future work would also include a zooming function for wider areas and adding important details on demand. Such helper functions could display size and volume of the selected room or distance between the telerobot and according OOI if selected with the input device.

- Input device: The design of our teleoperation system supports a robust and simple-to-use input device which is also cost effective. As a consequence we utilize a traditional 2D mouse with three buttons. These are dedicated to our three high-level interactions (Figure 7.1) to trigger recovery behaviors. However, the design of interactions and button mappings could be still improved by evaluating different layouts toward optimum usability. Further, utilizing a mouse with more degrees of freedom [184] could improve support for multi-floor exploration or manual steering of a camera gimbal with the attached joystick.

- Multi-floor environments: To be able to explore multi-floor environments, our sys-

tem would require further components. For instance, the system would need to be able to detect stairways [185]. In addition, the robustness of the untethered remote connection would have to be improved. While the implementation of our current system uses commodity hardware, systems with increased power and higher penetration of structures are for example presented by Patra et al. [186]. Additionally, like introduced for nuclear power plant inspection [187, 188], one or multiple additional telerobots could be used as mobile wireless transmission relays, retaining reliable communication.

7.4 User Study

The purpose of our study is to investigate the effect of our high-level teleoperation system on operator performance during real-world exploration missions. We considered the different teleoperation systems as strongest baseline for our study conditions, whereas we compare our high-level teleoperation system, including the RPG (Section 7.2.2.1), against a traditional baseline system with direct controls (Section 7.2.2.2). Table 7.1 gives an overview of the experimental conditions, type of systems, and view modes, whereas (Figure 7.10) summarizes results of our user study.

A core aspect of our study is that, despite a variety of related work has shown semi-autonomous systems positively effecting task performance, however it is unclear if this holds in a realistic setting where a system has to generate an interactive abstract topological scene view in real-time during flight missions. While operators with traditional direct controls can issue commands based on their quasi-instantaneous human cognition, operators of the semi-autonomous system need to wait until it processes, abstracts and outputs (visualizes) the abstracted information. This raises the question if such systems are still able to improve task performance over traditional control approaches in a realistic setting, where operators potentially need to wait until information is available in the topological view.

7.4.1 Experimental Design

In the following sections, we summarize the experimental design of our user study, including study conditions and tasks. Besides, we give details about study procedure, participants and accordance of the study to the local legislation.

7.4.1.1 Conditions

The main objective of our study was to assess the effect of the two user interface conditions, RPG and JOY, on operators task times, mental load and general comfort during a real-world indoor exploration mission. We based our study on within-subjects design and varied the order of the conditions using full counterbalancing. We defined task completion time, mental load and general comfort of the operator as main task performance

metrics. We formulated the following hypothesis for the user study and report on results in Section 7.4.2:

- H_6: The operator‘s task time decreases in RPG.
- H_7: The operator‘s mental load decreases in RPG.
- H_8: The operator‘s general comfort increases in RPG.

7.4.1.2 Tasks

According to Bhandari et al. [189], typical indoor exploration tasks involve investigation of the unknown environment and evaluation of hazardous areas to minimize human risk. We designed our study so that participants had to fulfill similar tasks in our experimental setup (Figure 7.11). We assumed a situation where the operator is far from the indoor space and has no prior knowledge of it. To ensure a basic degree of validity, we discussed the design of our experimental task-design with a local fire brigade. As a conclusion, the firefighters confirmed the validity of our task design and further emphasized usefulness of our system for assessment of a stable but still potentially dangerous situation. As an example use case, they specified the on-site inspection of a damaged chemical recovery boiler where an imminent explosion cannot be ruled out.

The indoor exploration task of our study comprises three subtasks, which had to be completed by each participant in each of the conditions. During this task, participants had to fully explore the environment and find all OOIs. In particular, participants were told to:

- Find all 19 hazardous areas marked with fiducial markers.
- Find the safe portal.
- Find the victim.

The placement of objects was altered in a controlled fashion to avoid learning effects. An overview of the experimental indoor environment is given in Figure 7.11.

7.4.1.3 Procedure

Before each experimental session, an introduction to the teleoperation system was given to the participant by the experimenter. Preliminary questions were asked to identify eyesight restrictions. The evaluation procedure of each experimental condition can be split into three phases. In a training phase, participants learned to use the system of the specific condition. This phase ended when participants reported to be comfortable in using the system. In the second phase, participants had to accomplish the indoor exploration task as fast as possible. For each participant, we captured screen recordings and measured the task completion time using a stop watch. The task was considered to be completed

when the system detected all safe portals, hazardous areas and victims (RPG condition) or users verbally confirmed to the experimenter that they found all of those objects (JOY condition). In both conditions, users were aware of the number of objects they already identified as well as of the objects they still need to find. Finally, participants were asked to fill out a NASA-TLX [89] task-load questionnaire (Scale: 0-100) as well as a custom questionnaire with respect to their experience in the respective condition. The custom questionnaire contained 8-point Likert items (ranging from 1, "strongly disagree", to 8, "strongly agree") asking participants about accuracy and smoothness of control as well as their perception of control over the system and their general comfort during the task.

7.4.1.4 Participants

A total of 23 participants were invited, 20 of them successfully finished the given tasks in all conditions. 3 participants had to stop the study due to technical problems and their results have been excluded. We invited 17 male participants and 3 female participants which were either students or researchers in the field of computer science or electrical engineering at Graz University of Technology (age: M=27.6, SD=3.98).

7.4.1.5 Ethics Statement

The presented study was carried out in accordance with the World Medical Association's Declaration of Helsinki, as revised in 2013 [190]. The study did not involve any medical experiments and further, no biometric data was taken from participants. We did not take any personal data from participants besides age, whereas all taken data was fully anonymized. In general, the study was conducted in accordance with the local General Data Protection Regulation (GDPR) in Austria and all participants gave written informed consent via an IRB consent-form. As per the local legislation, no IRB approval was required for our particular study type.

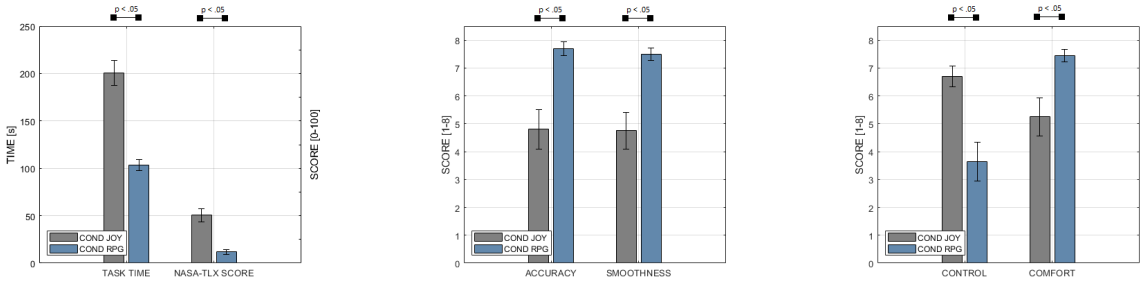


Figure 7.10: Our study results indicate significantly decreasing task times (Scale: 0-250s) and decreasing NASA-TLX score (Scale: 0-100) with our high-level teleoperation system (condition RPG). Based on an even 8-point Likert scale (Agreement-Scale: 1 Strongly Disagree - 8 Strongly Agree), we managed to retain general comfort during operation, compared to our baseline system with traditional joystick controls (condition JOY). In addition, participants reported increasing accuracy of control and smoothness of control.

7.4.2 Results

In each of our 20 sessions, we tested the teleoperation system in both conditions, JOY and RPG. This resulted in a total of 40 valid runs. For each participant, we took screen recordings and measured the task completion time during the flight. After finishing the flight for one condition, participants were asked to fill out the NASA-TLX score as well as a custom questionnaire. This questionnaire contained several 8-point Likert items asking participants about the accuracy of control, the smoothness of control, their perception of control over the system and their comfort in general during the task. We report mean, standard deviation and interval estimates for a 95% confidence interval. For significance testing, we use a paired samples t-test for task execution time as the data is normally distributed. All other measures are compared using Wilcoxon signed-rank test as questionnaire responses are non-parametric.

The main findings of our study are summarized in Figure 7.10. Statistical testing revealed that the task completion time was significantly lower for the RPG- ($M = 103.7s, SD = 13.7s$) compared to the JOY-condition ($M = 200.1s, SD = 30.58$, $t(19) = 12.01, p < 0.001$). In addition, a significant effect of conditions on mental load, as determined by NASA-TLX, has been revealed ($Z = 210.0, p < 0.001$). Again, RPG ($M = 11.75, SD = 6.43$) caused a significantly lower mental load than JOY ($M = 50.71, SD = 16.41$).

In our custom questionnaire, we asked participants about their perception of the tested user interface. Unsurprisingly, the perceived level of control in conditions decreased with the increasing LOA from JOY ($M = 6.7, SD = 0.87$) to RPG ($M = 3.65, SD = 1.63$). Wilcoxon signed-rank test showed that these differences are significant ($Z = 169.0, p < 0.001$). In contrast, participants perceived RPG ($M = 7.7, SD = 0.57$) to be significantly more accurate than JOY ($M = 4.8, SD = 1.67$, $Z = 0.0, p < 0.001$). Similarly, perceived smoothness of control was higher for RPG ($M = 7.5, SD = 0.51$) compared to JOY ($M = 4.75, SD = 1.55$). Again, differences are significant ($Z = 0.0, p < 0.001$). Finally, perceived general comfort was significantly higher in the RPG condition ($M = 7.45, SD = 0.51$), compared to JOY ($M = 5.25, SD = 1.62$), with ($Z = 0.0, p < 0.001$). This lets us accept H_8, which is supported by a significantly higher task completion confidence in RPG ($M = 7.8, SD = 0.41$), compared to JOY ($M = 6.8, SD = 1.06$, $Z = 0.0, p < 0.001$).

7.4.3 Discussion

Overall, we were able to support all of our three hypotheses, implying that our high-level teleoperation system is successful in supporting the operator during aerial exploration missions in challenging indoor environments. Remarkably, our teleoperation system reduced task execution times by 48.28% and task load by 76.82% compared to the JOY condition. Moreover, results indicate an increase in general comfort by 41.90%. We attribute the significant differences between conditions to the interplay of the RPG-view and the autonomous system. However, further research is necessary to differentiate the influence

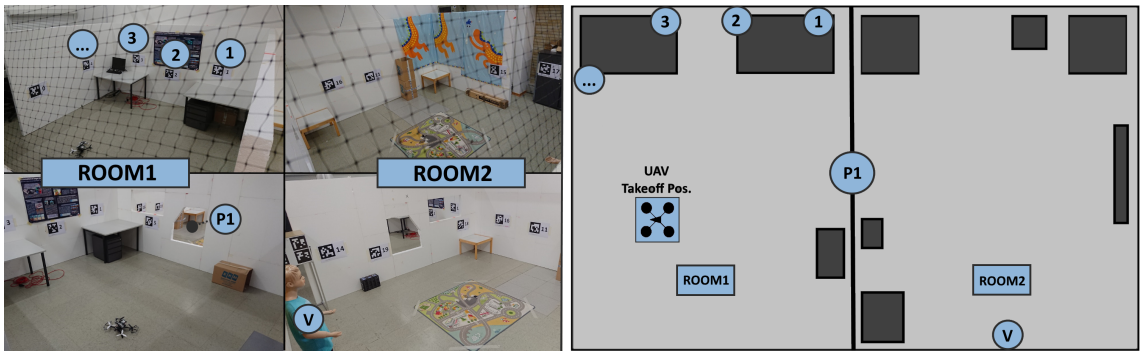


Figure 7.11: (left) Physical setup for our experimental evaluation. Note the two rooms, connected via a safe portal and the OOIs including a victim. (right) The same environment, represented as a floor-plan in 2D.

of the autonomous system and the topological scene view on results.

Although, participants conducted real-world flights to solve the posed exploration task, the study took place in a controlled environment. For instance, localization of the UAV was achieved with a motion capture system. However, on-board localization methods like SLAM have proven to be sufficiently accurate and fast to be used for UAV position estimation (Weiss et al. [191], Mur-Artal and Tardós [192]). In addition, due to limited lab space, the environment of our study did only comprise two rooms. Nonetheless, we believe that differences between conditions further evolve in favor of our system in wider- or multi-floor environments. The reason is that it is evidently harder to gain a good spatial understanding of larger compared to smaller environments. Thus, operators will benefit more from the RPG view in larger spaces, as the RPG abstracts the environment in an easy-to-understand manner. Furthermore, the task of our study was a simplification of complex real-world search and rescue missions. However, it is likely that our system even better supports operators in more complex task scenarios. For instance, research has shown that topological views, like the RPG, are beneficial if an environment is fully explored and operators are required to repetitively navigate between OOIs Johnson [173]. With regards to our system, the reinspection of an OOI could easily be performed by triggering its visualization in the RPG. The telerobot would then autonomously renavigate to the specific room and object. Due to mentioned reasons, we argue that, despite limitations, our experimental setting is an ecologically valid approximation of a real-world exploration mission.

Summarizing, our study has shown that high-level teleoperation systems with an on-the-fly created interactive scene topology are still able to better support operators in real-world settings, compared to systems using traditional controls. In this context, please also refer to our supplementary video[1].

[1]Supplementary video - High-Level Teleoperation System: https://youtu.be/1vXYoes1-lY

	JOY Condition	**RPG Condition**
Type of Control	Traditional Direct	High-Level Supervisory
LOA	1-3	7-10
RPG View	No	Yes
EXO View	Yes	Yes
EGO View	Yes	Yes

Table 7.1: User Study Conditions

ALFUS	**Level Definitions**	**Operating The UAV**
[10] Approaching 0% HRI	Highest level of autonomy, high complexity, all missions, extreme environment	Maximum autonomy level. Full autonomous exploration of the environment including object detection. The user can still improve task performance by switching to the collaborative levels (e.g., navigation of one preferred portal over the other or inspection of an OOI for verification).
[7-9] Low-level HRI	Collaborative level of autonomy, high complexity missions, difficult environments	High autonomy level involving non-mission-critical tasks. Functioning of repetitive low-level tasks is guaranteed (collision-free navigation). The operator can switch to this level, to supervise the system if it fails with complex tasks on highest level. **Minimum level that the operator can access in the RPG condition.**
[4-6] Medium-level HRI	Medium level of autonomy and complexity of missions, multi-functional missions, moderate environment	No operator access at this level for RPG condition.
[1-3] High-level HRI	Low level of autonomy, low level tactical behaviour, simple environments	No operator access at this level for RPG condition. **We use that range of levels in the JOY condition.**
[0] 100% HRI	Lowest level of autonomy, full manual remote control by the operator	No operator access at this level for RPG condition.

Table 7.2: Relation between the ALFUS and operating the UAV of our teleoperation system.

8 Conclusion And Outlook

Contents

This thesis presents three XR interfaces for teleoperation of aerial robots, originally motivated in a project context [8]. The focus of the contributions lies on supporting a human user or human operator during teleoperation of an aerial robot, by means of spatial AR (Chapter 5), MR (Chapter 6) and VR (Chapter 6 and Chapter 7) interfaces. The interfaces were tested and evaluated in three different use-cases, including in-situ guidance, remote inspection and aerial exploration of indoor environments. To better structure this thesis and highlight relevant common aspects amongst all presented contributions, we established the PDC, discussed in Chapter 3.

In the remainder of this chapter we first conclude the SLIM as the main experimental platform and basis for this thesis. Further, we highlight its benefits for experimentation with human users in constrained indoor spaces. We then discuss the presented XR interfaces, utilizing the different versions of the SLIM, within the context of the PDC. Results of the user studies, presented in Chapter 6 and Chapter 7, indicate that the designed XR teleoperation interfaces improved overall task performance of the human operator compared to the baseline conditions. It must be added that the spatial AR interface of the MAP, presented in Chapter 5, has not been evaluated regarding task performance in a more detailed user study. However, please note that the MAP and the according framework was a first essential step towards the SLIM. Finally, we provide lessons learned from our contributions, considering aspects highlighted by the PDC, and summarize guidelines and recommendations to support future work in Section 8.4.

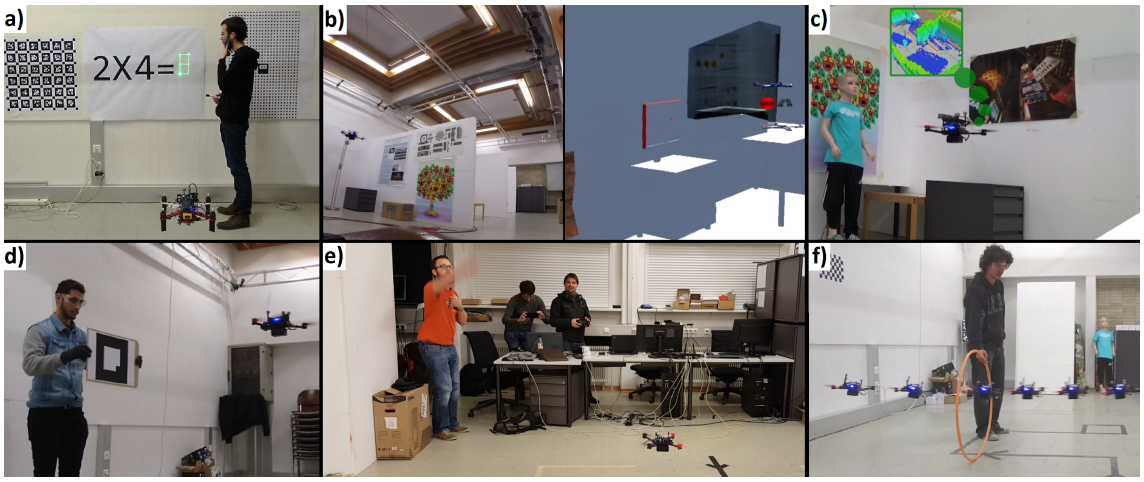


Figure 8.1: Example use cases of the *SLIM*. a) An earlier version of the *SLIM* acting as teaching assistant [113]. b) The *SLIM* during experimentation for the DAHV [114]. c) Victim detection during the camera drones' rescue challenge. d) Marker-based visual servoing. e) Avoidance of a thrown object. f) Hula-Hoop visual tracking and passing through.

8.1 Experimental Platform

The experimental platform, used to combine the presented XR interfaces with aerial robots, was introduced as *SLIM* in Chapter 4. The requirements of the platform were derived from the teleoperation use-cases. The design of the platform aimed for being versatile for research-, but also for educational purposes.

Since 2015, *SLIM* was used in various projects. These were either research projects or lectures and students projects. Research contributions which were utilizing the *droneSpace* and the *SLIM* platform were the Micro Aerial Projector [113] (Figure 8.1a) and the Drone Augmented Human Vision [114] (Figure 8.1b).

Additionally, a lecture where the *SLIM* platform was extensively used is called *Camera Drones* and was established in Winter Term 2016/2017 at Graz University of Technology. During this term, students had to work on individual projects based on a reference implementation. Each student received a *SLIM*, equipped with different vision sensors and SBCs. Amongst others, projects included visual servoing based on fiducial markers (Figure 8.1d), collision avoidance when throwing a reflective marker towards the *SLIM* (Figure 8.1e) and a hula-hoop flight (Figure 8.1f).

In Winter Term 2017/2018, students had to compete in an indoor MAV rescue challenge (Figure 8.1c). A maze was set up in the *droneSpace* as part of an artificial disaster scenario. The task for each team was to explore the environment and report back the 3D-positions of found victims. Based on a reference implementation, the teams had to solve individual sub-tasks, which included localization, path-planning and navigation, 3D-mapping and detection of fiducial markers using the onboard RGBD-sensor of *SLIM* (Figure 4.1c). Thus, students gained valuable experience in **control**, **sensing**, **navigation** and also

environmental mapping with MAVs.

In conclusion, the *SLIM* served as versatile experimental platform, especially for experimentation with human users in indoor spaces. The platform is constantly improved and extended; a brief overview on future work should given in the following.

One improvement is to achieve trajectory tracking by feeding higher-derivative reference signals into the PIXHAWK flight controller. Based on a position trajectory, discretized velocity- and acceleration inputs could be derived and directly fed into the inner control architecture. Ultimately, the tracking accuracy could be improved.

Another potential improvement of the *SLIM* involves reducing weight to improve flight times and further improve safety. As a future step, aluminum-based sensor housings could be replaced by 3D-printed parts. Additionally, sensor cables could be shortened to save space and all-up weight.

Finally, self-localization via SLAM-based algorithms, like discussed in Section 4.2.3, could be considered for future work. In spatially constrained indoor environments an external motion tracking system is still vital as a backup-system in case that localization fails.

8.2 Extended Reality Interfaces Inside The PDC

Our findings indicate that it is possible to significantly increase efficiency for inspection- and exploration tasks. In the context of teleoperation, this can be achieved at different physical distances by utilizing advanced XR interfaces. However, we found that the physical distance between the human operator and the aerial robot is not the only important aspect influencing task-efficiency. Instead, a rich variety of aspects related to visualization, HRI and HCI were affecting the experiments, whereas *perception* had the strongest influence.

The PDC classifies three perceptual distances between different member categories. These member categories are human operator, aerial robot and workspace. By utilizing the terminology of *perception*, we emphasize that dependency between the members of the PDC is strongly, but not purely, depending on visual aspects. Instead we found that it is rather beneficial to consider a full perceptual spectrum.

In the following, the most relevant aspects for experimental- and framework design are summarized:

- Task-category and overall requirements to increase performance for a specific use-case.
- Resulting *members* involved in the PDC and if they are *active* or *passive*.
- Spatial and geometric relations/constraints in between the relevant members of the PDC (distance, occlusion, type and size of workspace).

- Multi-sensory perception of either physical or virtual information inside the PDC (preferred reality AR, MR or VR and physical- or virtual-viewpoint).
- Required autonomy of the involved robotic agent and resulting level of interactions (high-level vs. low-level) and utilized interaction methods.
- Ergonomics (up to what level is it possible/convenient to encumber the user, robotic-based spatial AR vs. advanced wearable displays).
- Safety concerns if relevant in that use-case.

Based on the major aspects summarized above, we were able to classify the use-case into the three main perceptual distances CLOSE, MEDIUM and FAR. We believe that, by introducing the PDC, requirements for the presented contributions and also future related work become more transparent.

8.2.1 Room For Improvement At CLOSE Perceptual Distance

In Chapter 5, we present the MAP, which is able to stabilize the projection during hover and even dynamic flight and increases visual quality by utilizing the onboard hardware. Combined with compact size and acceptable flight times, the MAP is able to guide and support the user in various scenarios.

Limiting factors were the narrow field of view of 10° of the laser projector and dependency on the motion tracking system. We were able to overcome limitations related to the FOV, by utilizing the dynamics of the MAP to steer the laser approximately towards the desired projection region. In addition, we introduced a laser projection model, which was able to compensate for nonlinearities. With mean projection error characteristics of below 1cm at a maximum distance of approximately 2.9m, we are able to compensate for higher dynamics of the system and improve visual quality of projected symbols.

Future work would involve to implement state prediction considering the time delays of the system to predict the movement of the MAP to better compensate the latency of the laser projector interface. As a next step, the position and shape of the target projection surface could be estimated online in combination with the onboard vision camera, as these are currently known variables used to calculate P_R (3D point in the frame of the ROI). With such an extension, the MAP could dynamically adapt to non-planar, skewed or tilted projection surfaces, and even adapt the projected symbol to the current user viewpoint. We also foresee using SLAM or optical flow for navigation to improve mobile capabilities and make the platform more flexible for future interaction scenarios.

8.2.2 Room For Improvement At MEDIUM Perceptual Distance

Chapter 6 presents a prototypical system to discover a remote or occluded scene in an intuitive way by visualizing live imagery streamed from a camera aerial robot in a three-

dimensional, exocentric context. To control the exploration, we have implemented experimental high-level interaction techniques that control the aerial robot indirectly, by relating to the enclosing space in which the aerial robot is flying rather than the aerial robot‘s own local coordinate system and flight parameters, such as speed or altitude. This gives the user the impression of being present next to the aerial robot, or having X-ray vision when using a see-through display. Our experiments confirm that this style of interaction is efficient compared to conventional remote piloting and that it is at attractive for users. This work represents only the beginning of our work. While we have argued that aerial robot-augmented human vision is technically feasible, we have relied on simplifying assumptions with respect to several challenging technical aspects: The aerial robot does relies on external tracking for autopiloting and does not reconstruct the environment dynamically. Instead, we use a 3D model of the environment that was created offline. Moreover, path planning is simplistic and does not scale up to real unknown locations. Many technical parameters, such as image resolution, flight times, display field of view etc. are not yet satisfactory. Nonetheless, we are confident that ongoing technical developments in several fields will turn out in favor of the proposed system design. Potential next steps could involve testing and enhancing of the interaction metaphors and enhancing image-based rendering of the model of the remote scene.

8.2.3 Room For Improvement At FAR Perceptual Distance

In the work presented in Chapter 7, we demonstrate a fully working teleoperation system for aerial exploration missions. It improves task performance by using an interactive scene topology, whereas related work motivates using topological representations for robotic teleoperation. However, in contrast to related work, we for the first time investigate on how task performance is effected if the topology is created in real-time during actual indoor flight missions. The overall goal of our system was to reduce task times and mental load of the operator while conserving general comfort. To elaborate on the expected improvement, we evaluated our teleoperation system with a user study under real-world conditions. We compared our high-level teleoperation system against a traditional baseline system with joystick control. Results indicate that our system positively effects task performance and operators comfort during aerial exploration of challenging indoor environments.
In future work we would like to address the limitations of our system (Section 7.3) and conducted user study (Section 7.4.3). Further we would like to evaluate our system in more complicated or even multi-floor environments, for which abstraction has a potentially larger benefit in terms of overall task performance.

8.3 Lessons Learned From The PDC

This section details benefits and problems which arose in the individual experimental use-cases. They are discussed more detailled in the following sections.

8.3.1 CLOSE Perceptual Distance

The MAP is a representative for interaction at CLOSE distance in the perceptual distance continuum. According to the terminology introduced in Chapter 3, this leads to a PDC classification as follows:

- **Task-Category - In-situ guidance:** The task of the user is to solve an equation which is written on a wall. In the given passive workspace, the aerial robot guides the user by double-checking results of the equation and providing feedback about correctness.

- **Involved PDC-members:** Besides the presence of one human operator manipulating the workspace, one aerial robot augmented the workspace, while the workspace itself can be classified as passive.

- **Perceptual distance**: The presented use-case involves CLOSE physical interaction between human operator and aerial robot. This is because the user is able to directly see the aerial robot, meaning there is no physical occlusion between them. Additionally, interaction happens inside the range of the proxemics which enables full multi-sensory perception of the aerial robot.
  Problems: Since there is no physical separation between the human operator and the robot, and both share the same workspace, safety concerns must be considered. Basic safety measures are recommended for the aerial robot (propeller guards) but also for the human operator (safety goggles).
  Benefits: An underlying aspect for CLOSE classification is that, if the user does not currently observe the aerial robot (e.g., the aerial robot is not in the field of view of the user), the aerial robot could be still perceived as CLOSE, as a variety of other senses work here: Tactile senses (rotor wind), sense of hearing (rotor noise) or even smell [112]. Also, due to the fact that the human operator is present CLOSE to the physical workspace, demands to the visualization support to establish scene understanding are low (e.g., comapared to interaction at FAR distance discussed in Chapter 7).

- **Reality-Mode:** The reality of this use-case is spatial AR; projected screens are generated on sides of the aerial robot.
  Problems: Several problems and limitations can occur here, which make more complex scenarios highly challenging. The limited flight times of the aerial robot limit overall task time. Localization of the user is achieved via an external motion tracking system. Only full self-localization of the aerial robot and tracking of the user would enable a highly-flexible workspace environment.
  Benefits: The human operator is able to freely move inside the workspace, assuming that the aerial robot is able to dynamically adapt to the changing workspace. In addition, the user is fully unencumbered; no additional devices are

necessary for the presented task.

- **Size of Workspace, View-Mode and Mapping of Physical to Virtual Scene:** Since the given workspace is comparably small, it can be fully covered by the FOV of the human operator. The human operator is able to look at the scene from a physical egocentric view. Further, the projected screens are augmented on top of the real-world, but, in this case, do not aim for changing the visual appearance of the physical representation (Section 1, Figure 1.1). This means that the underlying geometry preserves a 1:1 mapping between physical and virtual elements, also supporting the CLOSE distance classification.
  Problems: Due to the physical egocentric viewpoint and external generation of visual elements, problems could occur due to shadowing of projections and coverage of the workspace, if this is not intentionally used for interaction. Problems could occur, if the location of the workspace must be dynamically changed. Such use-cases could be treated with an additional virtual exocentric viewpoint. As discussed by Higuchi et al. [193], this could lead to other problems due to out of body experience of the human operator.
  Benefits: Given the full coverage of the workspace by the human‘s FOV, no exocentric viewpoint is mandatory.

With the MAP, we were able to present a working interaction scenario, where we used mobile spatial augmented reality to assist the user in terms of accomplishing a task. One important aspect which was not treated in detail were proxemics when the MAP was flying close to the user. On one hand, Kim et al. [194] state that human operators feel more comfortable at CLOSE distances during collaborative tasks. On the other hand, a minimum flight distance had to be kept to the human operator and additional safety measures (goggles, arm protectors) were taken to guarantee adequate safety of the involved human operator. Further, the MAP dynamically adapted its distance and only approached the workspace once the user moved away. This could be seen as respecting the personal workspace of the human operator.

8.3.2 MEDIUM Perceptual Distance

The work discussed in Chapter 6 is representative for interaction at MEDIUM distance in the perceptual distance continuum. According to the terminology introduced in Chapter 3, this leads to the following PDC:

- **Task-Category - Through-Wall Inspection:** The task of the user is to inspect OOIs placed in a spatially constrained environment within the help of the aerial robot. The user is not physically present inside the scene, but is standing behind an occluding wall. The human operator is not able to easily physically reach the aerial robot. By projecting the live view of an onboard mounted camera onto a

predefined geometry model of the environment, the human operator is able visually access hidden areas of the environment to inspect them. Two task-categories were elaborated on. For the first task-category, the user is told to give indirect positioning commands to the let the UFO inspect text displayed on screens. For the second task, the user is told to directly command the aerial robot to a pre-defined 3D positons inside the environment (pick and place interaction).

- **Involved PDC-members:** Again, one human operator is involved (manipulating/teleoperating the aerial robot), one aerial robot visually perceived the workspace, whereas again the workspace itself can be classified as passive.

- **Perceptual distance - MEDIUM**: The presented use-case involves interaction at MEDIUM distance between human operator and aerial robot. The user is not always able to directly see the aerial robot, meaning there is some physical occluder between human operator and aerial robot. This has great impact on classification inside the PDC. Interaction for the discussed inspection-tasks happened inside the range of the close phase of the public space and interaction techniques like pick-and-place can be still used effectively assuming a 1:1 mapping of physical and virtual environmental representation. Remarkably, the 1:1 mapping is not mandatory from a pure visualization point of view, since the virtual scene representation including the occluder and human operator could be physically moved to potentially any remote location. However, this 1:1 coherence is required since it is strongly vital for inspection or maintenance tasks. Thus, it is also emphasized by the PDC.
  Problems: Interaction at MEDIUM distances potentially leads to a significant increase in terms of required visualization support to improve scene understanding (e.g., X-ray vision), compared to the use-case discussed in Chapter 5.
  Benefits: An underlying aspect regarding perceptual MEDIUM distance is that although the user does not directly see the aerial robot, perception of the aerial robot could be recovered due to the provided visualization techniques (MR interface). Additionally, since there is a weak occluder between the human operator and the robot, requirements in terms of safety are lower compared to the teleoperation at CLOSE distance.

- **Reality-Mode:** The reality modes for the use-case in this chapter are MR and VR. Visualization support is provided by a head-worn MR smartglass (HoloLens [195]). For the MR mode, a virtual scene representation is overlaid with the close physical scene (wall/occluder). A virtual live-camera view taken inside the physical scene is projected onto the virtual representation of the hidden scene.
  Problems: The reality mode for this use-case reveals problems and limitations. Again, limited flight energy ressources of the aerial robot, but also the head worn display, limit overall task time for such scenarios. In addition, localization of the user is achieved via an external motion tracking system, and the virtual representation of

the hidden scene is based on a predefined model and not reconstructed online. For more complex scenarios, self-localization of the aerial robot and online reconstruction of the environment would be vital. This would put significantly higher demands on the hardware. In addition, the user is encumbered.
Benefits: The utilized MR-technology enabled intuitive scene visualization (1:1 mapping of physical and virtual scene) and interaction inside a CLOSE physical range providing visual access to an otherwise inaccessible scene. In addition, the MR-technology enabled the user to seamlessly switch to and interact with purely virtual scene representations. Thus, also interaction from FAR distant remote locations can be achieved.

- **Size of Workspace, View-Mode and Mapping of Physical to Virtual Scene:** Since the given workspace is now considerably larger, it can not be fully covered by the FOV of the human operator any more. A transitional interface is introduced to switch between exocentric view and egocentric view-mode (overview-and-detail technique).
  Problems: A 1:1 mapping of physical and virtual scene limits maximum distance for interaction. This supports a separation from the CLOSE or MEDIUM distance to the FAR distance. At increased physical distance, effectiveness of interaction techniques and perception of the aerial robot becomes increasingly challenging (Poupyrev et al. [104] and Plumert et al. [109]). In this case, techniques from a purely virtual viewpoint (world in miniature) and more advanced interaction techniques need to be utilized.
  Benefits: The X-ray vision combined with the physical viewpoint behind the wall helps to improve scene understanding and naturalness of interaction at CLOSE distances. Thus, effectiveness for spatial tasking can be increased inside hidden areas which are close to the human operator.

8.3.3 FAR Perceptual Distance

Finally, the work presented in Chapter 7 explores interaction at FAR perceptual distance in the PDC. According to the terminology introduced in Chapter 3, this leads to the following PDC classification:

- **Task-Category - Exploration Of Spatially Constrained Indoor Environments:** The task of the user is to fully explore an artificial indoor disaster scene by teleoperating the aerial robot. The user is not physically present in the scene and located behind a strongly occluding wall. The human operator is not able to easily reach the aerial robot physically. The goal is to find OOIs, like potential victims, inside the disaster scene. Two conditions were elaborated as part of a study. In the first condition, the aerial robot is directly commanded via a joypad, and, in the second condition, the high-level interface is used. Results indicated that the abstract

scene visualization combined with the high-level interactions significantly increased overall task-efficiency.

- **Involved PDC-members:** One human operator is involved (manipulating/teleoperating one aerial robot). The aerial robot visually perceived the workspace. The workspace is considered as passive member.

- **Perceptual distance - FAR**: The presented use-case involves interaction at FAR distance between human operator and aerial robot. The user is not able to directly see the aerial robot, meaning there is at least one physical occluder between human and aerial robot. Interaction for the discussed task happened inside the range of the proxemics, although no direct manipulation of the aerial robot is involved. Further, a 1:1 mapping of physical and virtual environmental representation is not given.
  Problems: Interaction at FAR distances requires more visualization support to improve scene understanding. Compared to the MEDIUM use-case, multiple view modes were utilized, common for such teleoperation tasks (egocentric live-camera view, exocentric 3D map view, abstract 2D representation). Also, due to the non-existing 1:1 mapping of physical and virtual representation, naturalness of direct object manipulation suffered.
  Benefits: Any physical occlusion (either weak or strong) between the human operator and the aerial robot at MEDIUM or FAR perceptual distance decrease demands on the safety measures.

- **Reality-Mode and Mapping of Physical to Virtual Scene:** No specific XR view-modes were utilized. Scene visualizations were purely virtual and without any immersion. Visualization-support is provided by virtual 2D-egocentric FPV and 3D-exocentric maps. No switching between the view-modes is required, and all view-modes are accessible to the user on demand at any time.
  Problems: Due to the missing 1:1 relation between physical and virtual scene, demands on the interface with regards to scene understanding and situational awareness are high. On one hand, immersive scene perception could be enabled by utilizing an HMD. On the other hand, compared to MEDIUM and CLOSE distance, fully recovering rich scene perception and enabling natural navigation through the scene is significantly more difficult. Naturally moving through the explored virtual scene (using a 1:1 mapping) would either require a sufficiently large workspace or advanced interaction techniques, like redirected walking [196].
  Benefits: Assuming that the aerial robot is capable of accomplishing the commanded high-level task, reduced scene detail could be also beneficial. Overwhelming the human operator with unnecessary information can be prevented, while situation awareness is conserved or even improved. If intervention of the human operator is required, access to more detailed scene information might be

necessary. It must be noted that finding a task-dependent optimum between autonomy, scene details and interaction is hard to achieve.

- **Size of Workspace and View-Mode:** Although the physical space used for experimentation was the same as in the MEDIUM condition, the workspace discussed in Chapter 7 is of increased size. No direct visual perception of the aerial robot and the workspace is possible any more. As a result, the interface design combines an egocentric view-mode with an exocentric 3D map, both being purely virtual and presented side-by-side.

For remote teleoperation use cases, the preferred reality would be VR to improve immersiveness and immediate scene understanding.

8.4 Guideline And Recommendations

Based on the lessons learned from the previous section, we now summarize the most important findings after reflection on the relationship of the PDC and the presented contributions. An overview of the findings is presented in Figure 8.2. The primary focus of this guideline is on recommended **reality-mode**, **viewpoint** and **supportive visualization devices (displays)**. The following discussion should help to improve **easiness, intuitiveness and overall task-performance for future designs of XR-supported teleoperation interfaces**.

8.4.1 CLOSE Perceptual Distance

For teleoperation at CLOSE distance, all XR modes (spatial AR, MR and VR) are feasible. Utilizing spatial AR involves a projection device, which is able to augment the physical environment (workspace) of the human operator. In the context of ubiquitous computing, typical setups involve handheld see-through displays (smart devices like phones or tablets). A major drawback of such devices is that the human operator has to hold the display and can‘t use both hands for interaction with the workspace. Alternatively, the human operator could utilize a head-worn smart display to enable hand free interaction with the workspace. However, in both cases the human operator is encumbered with the device. In Chapter 5, we were able to overcome the afore discussed limitations by means of spatial AR. By utilizing a highly mobile projection device, we proofed that in-situ guidance of a fully unencumbered human operator is technically feasible. Clear drawbacks occur due to increased system complexity. In addition to the computational power and hardware that is necessary to generate arbitrary projection screens, first, the mobile projection platform has to be flight-controlled at all time (including all connected limitations, like flight times, safety, etc.). Second, it might be required that the projected screens have to be kept inside the human operator‘s FOV. For this purpose, the mobile projection

Categories	PERCEPTUAL DISTANCE CONTINUUM								
	CLOSE			MEDIUM			FAR		
PHYSICAL DISTANCE TO MEMBER	Physical distance inside close public phase			Physical distance inside close public phase			Physical distance outside close public phase OR...		
OCCLUSION OF MEMBER	No occluders			Occluders but 1:1 coherence of virtual and real scene			... Occluders and NO 1:1 coherence		
PERCEPTION/REACHABILITY OF MEMBER	Fully perceivable, easily reachable			Not fully perceivable OR not easily/not reachable			Not perceivable and not reachable at all		
VISUALIZATION ASPECTS	*SAR*	*MR*	*VR*	*SAR*	*MR*	*VR*	*SAR*	*MR*	*VR*
TECHNICALLY FEASIBLE REALITY MODE?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SUGGESTED BY (EXAMPLES)	Isop et al. [113]	Hedayati et al. [78]	Higuchi et al. [193]	UFO [8]	Whitlock et al. [110]	Erat et al. [114]	???	???	Isop et al. [81]
RECOMMENDED REALITY MODE	SAR OR MR			MR			VR		
FEASIBLE SUPPRTIVE DEVICES (DISPLAYS)	MOBILE PROJECTOR, HANDHELD SEE-THROUGH DEVICE, HEADWORN SEE-THROUGH DISPLAY, HEADWORN VR-DISPLAY			MOBILE PROJECTOR, HANDHELD SEE-THROUGH DEVICE, HEADWORN SEE-THROUGH DISPLAY, HEADWORN VR-DISPLAY			MOBILE PROJECTOR, HANDHELD SEE-THROUGH DEVICE, HEADWORN SEE-THROUGH DISPLAY, HEADWORN VR-DISPLAY		
RECOMMENDED SUPPORTIVE DEVICES (DISPLAYS)	MOBILE PROJECTOR or HEADWORN SEE-THROUGH DISPLAY			HEADWORN SEE-THROUGH DISPLAY			HEADWORN VR-DISPLAY		
FEASIBLE VIEWPOINT	EGO (physical), EXO (virtual)	EGO (physical or virtual), EXO (virtual)	EGO (virtual), EXO (virtual)	EGO (physical), EXO (virtual)	EGO (physical or virtual), EXO (virtual)	EGO (virtual), EXO (virtual)	EGO (physical), EXO (virtual)	EGO (physical or virtual), EXO (virtual)	EGO (virtual), EXO (virtual)
RECOMMENDED VIEWPOINT	EGO (physical), EXO (virtual)			EGO (physical or virtual), EXO (virtual)			EGO/EXO (virtual)		
IMMERSIVENESS	HIGH (natural perception of the real environment)			MED (immersiveness could be recovered if 1:1 coherence is conserved)			LOW (clear limitations and challenges without 1:1 coherence)		
IMMEDIATE SCENE UNDERSTANDING	HIGH			MED			LOW		
INTERACTION ASPECTS	*SAR*	*MR*	*VR*	*SAR*	*MR*	*VR*	*SAR*	*MR*	*VR*
NATURALNESS OF THE INTERACTION	HIGH (natural interaction with the real environment)			MED (naturalness can be recovered if 1:1 coherence is conserved)			LOW (natural interaction difficult at more far distances and no 1:1 coherence)		
DIRECTNESS OF INTERACTION WITH WORKSPACE	HIGH (Isop et al. [113])			MED (Erat et al. [114])			LOW (Isop et al. [81])		
RECOMMENDED LEVEL OF CONTROL OVER THE ROBOT	LOW			MED			LOW to HIGH (on demand)		
LEVEL OF ENCUMBERING THE USER	LOW	MED	HIGH	LOW	MED	HIGH	LOW	MED	HIGH
REQUIRED LEVEL OF ENCUMBERING THE USER	LOW			MED			HIGH		
SYSTEM ASPECTS	*SAR*	*MR*	*VR*	*SAR*	*MR*	*VR*	*SAR*	*MR*	*VR*
INTERFACE COMPLEXITY (REQUIRED SUPPORT OF SCENE UNDERSTANDING and CONTROL OF AERIAL ROBOT)	LOW			MED			HIGH		
OVERALL SYSTEM COMPLEXITY	MED	MED	MED	HIGH	MED	MED	HIGHEST	MED	MED
REQUIREMENTS FOR SAFETY	HIGH			MED			LOW		

Figure 8.2: Summary of our findings after reflecting the PDC as classification framework on relevant aspects of the presented contributions. The aspects are separated between the individual perceptual distances, whereas most important aspects are highlighted in red.

platform has to keep track of the human operator‘s body (or, at least, head-pose), while at the same time the projections have to be adapted accordingly, if the human operator changes its FOV. For future work, findings of this thesis indicate that separation of the tasks (projection, self-localization and tracking of the human operator) onto two aerial robots is highly recommended. This would greatly decrease limitations in terms of all-up-weight and flight times. However, fully unencumbering the human operator then comes at the cost of overhead in framework complexity. **Spatial AR by means of highly mobile projection** devices can be recommended as the preferred reality mode at CLOSE perceptual distance. This is true if an increased system complexity is acceptable and **fully unencumbering the human operator** is required. If encumbering is acceptable up to some point, and safety is a major concern, instead, head-worn see-through displays as supportive devices (MR smart glasses) could be recommended instead.

Further, if a workspace limited to the FOV of the human operator is considered, an **physical egocentric viewpoint** is clearly recommended. In addition, the work of Higuchi et al. [67] suggests to use **virtual exocentric viewpoints**, provided by the aerial robot. While their work emphasizes benefits of an exocentric viewpoint for self-perception, it could be vital in the context of the PDC if the workspace exceeds the FOV of the human operator.

Immediate scene understanding and interaction with any potential workspace at CLOSE distance is naturally given without any support. Thus, VR is clearly the least preferred reality mode at this distance. Moreover, supportive devices like VR-HMDs may significantly encumber the human operator and also limit the workspace.

At **CLOSE distance safety requirements are highest inside the PDC**. According safety measures must be taken since the aerial robot is potentially flying close to the human operator.

8.4.2 MEDIUM Perceptual Distance

For teleoperation at MEDIUM distance, all XR modes are feasible. If spatial AR is concerned, this distance would require, at least, two aerial robots. One aerial robot is acting in the workspace behind any potential occluder. The second aerial robot would need to hold the projection device and locally augment the environment of the human operator to establish the XR interface. This, in turn, greatly increases overall system complexity which makes spatial AR the least preferred view mode her. Like introduced in Chapter 6, an **MR setting with an according head-worn display is recommended** instead. **If constraints of the PDC are respected, an MR setting could recover scene understanding and natural interaction**, even if the workspace is occluded and physically not directly reachable. The MR setting also enables natural interaction with and perception of the physical world locally on the human operator‘s side of the occlusion. This is valid, if a 1:1 coherence of real and virtual scene, and teleoperation

inside the proxemic notations can be established. Also, encumbering the human operator at a medium level must be accepted. Clearly, the MR interface could be replaced with VR technology. While a VR display could provide more computational power, drawbacks are a more limited workspace of the human operator behind the occluder and higher level of encumbering.

Besides of the physical egocentric viewpoint of the human operator, additional virtual viewpoints can be explicitly recommended. While both viewpoints help to improve scene understanding of the occluded workspace, the virtual egocentric viewpoint can give access to scene details. A virtual exocentric view on the occluded scene can help to improve overall scene understanding (overview-and-detail).

At MEDIUM distance, safety concerns are lower due to the presence of occlusion. However, **depending on the characteristics of the occluder, still safety requirements at moderate level are recommended**.

8.4.3 FAR Perceptual Distance

Although all reality modes are technically feasible, a **FAR distance teleoperation emphasizes usage of a VR setting**. Since the human operator is fully perceptually isolated from the workspace, interaction and scene understanding inherently lack of naturalness. Using a second or even multiple aerial robotic agents, this problem could be also addressed by means of spatial AR. However, the overall system complexity would increase to an unacceptable level. By utilizing a head-worn MR device, this overhead could be reduced. Nevertheless, **the benefit of being able to observe the physical surrounding is not helpful at FAR distance.** Furthermore, even if the MR display is morphed into a purely virtual one, **MR devices typically have less computational power compared to VR setups.** But this is a crucial aspect, as the **requirements to support the human operator‘s scene understanding and overall system complexity are highest at this distance. If encumbering the human operator is acceptable, a VR setting can be recommended.** If the XR interface is **required to not encumber the human operator, common Window on the World displays could be used.** We addressed these aspects in the work presented in Chapter 7. Albeit, non-immersiveness of the interface may lead to lack of immediate scene understanding, this could be compensated with providing simple scene visualizations, reducing details to a minimum (lower interface complexity). In combination with high level interactions, this leads to improved task performance for specific use cases (aerial exploration of indoor environments). On the downside, this requires that the system is robust enough to safely operate at higher automation level at all time. The overall system complexity of the aerial robot significantly increases.

At FAR distance, the physical representation of the workspace and the aerial robot is neither visible nor accessible by the human operator. As a result, since 1:1 coherence between physical and virtual scene can not be established, **purely virtual egocentric**

and virtual exocentric views remain as recommended, and moreover feasible, viewpoints.

Finally, at FAR distance, **safety concerns and according requirements to safety measures are lowest** in the PDC.

8.5 Outlook

The use cases for teleoperation at the three different distances of the PDC, presented as part of this thesis, reveal interesting future work on XR interfaces. Remarkably, considering that the visual-perceptual aspect is the strongest (however, amongst many other aspects) two extreme examples become apparent, which extend the three presented use cases to a potential future development on state-of-the-art XR technology. In the first example, all senses and capabilities, which would mean the full "being", of the human operator are perfectly transferred to the FAR distant workspace. For example, represented by an Avatar [197] or Surrogate [79]. Thus the operator would be remotely fully integrated into the FAR distant space. The second extreme example would require that the remote environment is fully physically "replicated" (e.g., morphing environment by means of mobile robots [198]) at a CLOSE workspace or surrounding of the human operator, which, in a very basic sense, relates to the idea of the Holodeck [199]. Even if both extreme cases are not entirely feasible yet because of technical limitations, a classification inside the PDC still holds. If we consider the "Surrogate-case" and the human operator would be fully transferred to the remote location (however fully perceptually isolated from the local environment), interaction at CLOSE distance could be established since the 1:1 coherence between physical and "virtual" world is guaranteed by the Surrogate. The same is valid for the "Holodeck-case", because naturalness of the CLOSE environment perceived by the operator could be recovered by a perfect physical replication of the FAR distant environment. Remarkably, in the latter case the human operator stays fully unencumbered, whereas it remains unclear which case would involve a higher degree of system complexity. In a broad sense, Chapter 5 represents an early stage of the Holodeck-case, whereas Chapter 6 and Chapter 7, in the context of robotic teleoperation, represents early work towards a fully functional surrogate.

In the presence of growing importance of research with regards to augmenting human senses, this thesis presented three contributions. While they emphasize that purposefully supporting teleoperation with state of the art XR technology is widely feasible and can lead to significantly improved overall task performance, clear limitations become apparent if it comes to system complexity and encumbering the human operator. Moreover it is important to distinguish between different physical distances, whereas evidence was found that they strongly affect the interaction between a human user, robotic agent, and the according visual methods for augmentation. While this thesis discusses visualization and interaction aspects mainly, however interesting future work would relate to investigating on a more fine grained classification of the PDC in relation to physical distance and

occlusion. Since technology will evolve over the years, a trend towards the two extreme cases "Holodeck" and "Surrogate" seems to be logical. This, in turn, would strongly support a more intense investigation on all related aspects of the PDC, resulting in a more detailed and complete classification framework for future related research on XR interfaces for robotic teleoperation.

A

List of Acronyms

AR ... Augmented Reality
DAHV ... Drone Augmented Human Vision
FOV ... Field Of View
HCI ... Human-Computer Interaction
HMD ... Head Mounted Display
HMI ... Human-Machine Interaction
HRI ... Human-Robot Interaction
LOA ... Levels Of Autonomy
MAP ... Micro Aerial Projector
MAV ... Micro Aerial Vehicle
MPC ... Model Predictive Control
MR ... Mixed Reality
NUI ... Natural User Interface
OOI ... Object Of Interest
PDC ... Perceptual Distance Continuum
POV ... Point Of View
SAR ... Spatial Augmented Reality
SLIM ... Scalable and Lightweight Indoor-navigation MAV
UFO ... User‘s Flying Organizer
UI ... User Interface
VR ... Virtual Reality
XR ... Extended Reality

Bibliography

[1] Stefanie Zollmann, Christof Hoppe, Tobias Langlotz, and Gerhard Reitmayr. Flyar: AR supported micro aerial vehicle navigation. *TVCG*, 20(4):560–568, 2014. (page 1, 18)

[2] Baichuan Huang, Deniz Bayazit, Daniel Ullman, Nakul Gopalan, and Stefanie Tellex. Flight, camera, action! using natural language and mixed reality to control a drone. ICRA, 2019. (page 1, 19)

[3] Rod Furlan. The future of augmented reality: Hololens-microsoft's ar headset shines despite rough edges [resources_tools and toys]. *Ieee Spectrum*, 53(6):21–21, 2016. (page 1, 13)

[4] Manuela Chessa, Guido Maiello, Alessia Borsari, and Peter J Bex. The perceptual quality of the oculus rift for immersive virtual reality. *Human–computer interaction*, 34(1):51–82, 2019. (page 2)

[5] Crystal Maraj, Jonathan Hurter, Schuyler Ferrante, Lauren Horde, Jasmine Carter, and Sean Murphy. Oculus rift versus htc vive: Usability assessment from a teleportation task. In *International Conference on Human-Computer Interaction*, pages 247–257. Springer, 2019. (page 2)

[6] Oliver Bimber and Ramesh Raskar. *Spatial augmented reality: merging real and virtual worlds*. AK Peters/CRC Press, 2005. (page 2, 16, 67)

[7] Andrew Wilson, Hrvoje Benko, Shahram Izadi, and Otmar Hilliges. Steerable augmented reality with the beamatron. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 413–422. ACM, 2012. (page 2, 16)

[8] Institute of Computer Graphics and Vision. Users flying organizer (ufo) - semi-autonomous aerial vehicles for augmented reality. `https://www.tugraz.at/institute/icg/research/team-schmalstieg/research-projects/ufo-users-flying-organizer/`, 2018. [Online; accessed 29-October-2018]. (page 2, 44, 125)

[9] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, 1995. (page 5, 21, 36)

[10] Thomas Pederson. *From Conceptual Links to Causal RelationsPhysical-Virtual Artefacts in Mixed-Reality Space*. PhD thesis, 2003. (page 5, 25)

[11] Edward T Hall. A system for the notation of proxemic behavior. *American anthropologist*, 65(5):1003–1026, 1963. (page 5, 21, 23)

[12] Mackinlay Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999. (page 11)

[13] The Franklin Institute. What's the difference between ar, vr, and mr? `https://www.fi.edu/difference-between-ar-vr-and-mr`, 2018. [Online; accessed 29-October-2018]. (page 12)

[14] Kiyoshi Kiyokawa, Yoshinori Kurata, and Hiroyuki Ohno. Elmo: An enhanced optical see-through display using an lcd panel for mutual occlusion. *International Symposium on Mixed Reality (ISMR)*, pages 186–187, 2001. (page 12)

[15] Qualcomm. Extended reality. `https://www.qualcomm.com/invention/extended-reality`, 2018. [Online; accessed 29-October-2018]. (page 13)

[16] J. P. Gownder. Breakout vendors: Virtual and augmented reality. `https://www.forrester.com/report/Breakout+Vendors+Virtual+And+Augmented+Reality/-/E-RES134187`, 2016. [Online; accessed 17-August-2016]. (page 13)

[17] Jay Jantz, Adam Molnar, and Ramses Alcaide. A brain-computer interface for extended reality interfaces. In *ACM SIGGRAPH 2017 VR Village*, page 3. ACM, 2017. (page 13)

[18] Tathagata Chakraborti, Subbarao Kambhampati, Matthias Scheutz, and Yu Zhang. Ai challenges in human-robot cognitive teaming. *arXiv preprint arXiv:1707.04775*, 2017. (page 13)

[19] Gal A Kaminka. Curing robot autism: A challenge. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 801–804. International Foundation for Autonomous Agents and Multiagent Systems, 2013. (page 13)

[20] Henrik I Christensen, T Batzinger, K Bekris, K Bohringer, J Bordogna, G Bradski, O Brock, J Burnstein, T Fuhlbrigge, R Eastman, et al. A roadmap for us robotics: from internet to robotics. *Computing Community Consortium*, 2009. (page 13)

[21] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communication of intent in assistive free flyers. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 358–365. IEEE, 2014. (page 13)

[22] Yu Zhang, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explainability and predictability for cobots. *CoRR abs/1511.08158*, 2015. (page 13)

[23] Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. Going beyond literal command-based instructions: Extending robotic natural language interaction capabilities. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. (page 13)

[24] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communicating directionality in flying robots. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 19–26. IEEE, 2015. (page 13)

[25] Shin Sato and Shigeyuki Sakane. A human-robot interface using an interactive hand pointer that projects a mark in the real work space. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 589–595. IEEE, 2000. (page 14)

[26] Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, and Achim J Lilienthal. That's on my mind! robot to human intention communication through on-board projection on shared floor space. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015. (page 14)

[27] Atsushi Watanabe, Tetsushi Ikeda, Yoichi Morales, Kazuhiko Shinozawa, Takahiro Miyashita, and Norihiro Hagita. Communicating robotic navigational intentions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5763–5769. IEEE, 2015. (page 14)

[28] Florian Leutert, Christian Herrmann, and Klaus Schilling. A spatial augmented reality system for intuitive display of robotic data. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 179–180. IEEE Press, 2013. (page 14)

[29] Shayegan Omidshafiei, Ali-Akbar Agha-Mohammadi, Yu Fan Chen, Nazim Kemal Ure, Shih-Yuan Liu, Brett T Lopez, Rajeev Surati, Jonathan P How, and John Vian. Measurable augmented reality for prototyping cyberphysical systems: A robotics platform to aid the hardware prototyping and performance testing of algorithms. *IEEE Control Systems Magazine*, 36(6):65–87, 2016. (page 14)

[30] Jinglin Shen, Jingfu Jin, and Nicholas Gans. A multi-view camera-projector system for object detection and robot-human feedback. In *2013 IEEE International Conference on Robotics and Automation*, pages 3382–3388. IEEE, 2013. (page 14)

[31] Matthew Turk and Victor Fragoso. Computer vision for mobile augmented reality. In *Mobile cloud visual media computing*, pages 3–42. Springer, 2015. (page 14)

[32] Sean O'Kane. Microsoft used this adorable robot to show off new hololens features. *The Verge*, 2015. (page 14)

[33] Rasmus S Andersen, Ole Madsen, Thomas B Moeslund, and Heni Ben Amor. Projecting robot intentions into human environments. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 294–301. IEEE, 2016. (page 14)

[34] Tathagata Chakraborti, Sarath Sreedharan, Anagha Kulkarni, and Subbarao Kambhampati. Alternative modes of interaction in proximal human-in-the-loop operation of robots. *arXiv preprint arXiv:1703.08930*, 2017. (page 14)

[35] Catherine Diaz, Michael Walker, Danielle Albers Szafir, and Daniel Szafir. Designing for depth perceptions in augmented reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 111–122. IEEE, 2017. (page 14)

[36] Samir Bouabdallah, Pierpaolo Murrieri, and Roland Siegwart. Design and control of an indoor micro quadrotor. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4393–4398. IEEE, 2004. (page 14)

[37] Jonathan P How, Brett Behihke, Adrian Frank, Daniel Dale, and John Vian. Real-time indoor autonomous vehicle test environment. *IEEE control systems*, 28(2): 51–64, 2008. (page 14)

[38] Anurag Sai Vempati, Vipul Choudhary, and Laxmidhar Behera. Quadrotor: design, control and vision based localization. *IFAC Proceedings Volumes*, 47(1):1104–1110, 2014. (page 14)

[39] Giuseppe Loianno, Chris Brunner, Gary McGrath, and Vijay Kumar. Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu. *IEEE Robotics and Automation Letters*, 2(2):404–411, 2017. (page 15, 49)

[40] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010. (page 15)

[41] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013. (page 15)

[42] DJI. Ryze tello. `https://www.ryzerobotics.com/de/tello`, 2018. Accessed: 10-12-2018. (page 15)

[43] Infineon. Educopter. `https://www.infineon.com/cms/en/applications/consumer/multicopters-and-drones/`, 2019. Accessed: 01-03-2019. (page 15)

[44] Intel. Intel aero. `https://software.intel.com/en-us/aero`, 2019. Accessed: 01-03-2019. (page 15)

[45] Qualcomm. Snapdragon flight. `https://worldsway.com/product/dragon-drone-development-kit/`, 2019. Accessed: 01-03-2019. (page 15)

[46] Mohammad Obaid, Omar Mubin, Christina Anne Basedow, A Ayça Ünlüer, Matz Johansson Bergström, and Morten Fjeld. A drone agent to support a clean environment. In *Proceedings of the 3rd International Conference on Human-Agent Interaction*, pages 55–61. ACM, 2015. (page 15)

[47] Jessica R Cauchard, Kevin Y Zhai, James A Landay, et al. Drone & me: an exploration into natural human-drone interaction. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 361–365. ACM, 2015. (page 15)

[48] Mohammad Obaid, Felix Kistler, Markus Häring, René Bühling, and Elisabeth André. A framework for userdefined body gestures to control a humanoid robot. *International Journal of Social Robotics*, 6(3):383–396, 2014. (page 15)

[49] Jessica R Cauchard, Kevin Y Zhai, Marco Spadafora, and James A Landay. Emotion encoding in human-drone interaction. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 263–270. IEEE, 2016. (page 15)

[50] Paul Robinette, Alan R Wagner, and Ayanna M Howard. Assessment of robot guidance modalities conveying instructions to humans in emergency situations. Georgia Institute of Technology, 2014. (page 15)

[51] Adrien Briod, Przemyslaw Kornatowski, Jean-Christophe Zufferey, and Dario Floreano. A collision-resilient flying robot. *Journal of Field Robotics*, 31(4):496–509, 2014. (page 15)

[52] Kei Nitta, Keita Higuchi, and Jun Rekimoto. Hoverball: augmented sports with a flying ball. In *Proceedings of the 5th Augmented Human International Conference*, page 13. ACM, 2014. (page 15)

[53] Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. Bitdrones: Towards using 3d nanocopter displays as interactive self-levitating programmable matter. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 770–780. ACM, 2016. (page 15)

[54] Dieter Schmalstieg and Daniel Wagner. Experiences with handheld augmented reality. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 3–18. IEEE, 2007. (page 16)

[55] Claudio Pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *International conference on ubiquitous computing*, pages 315–331. Springer, 2001. (page 16)

[56] Horst Hörtner, Matthew Gardiner, Roland Haring, Christopher Lindinger, and Florian Berger. Spaxels, pixels in space-a novel mode of spatial display. In *SIGMAP*, pages 19–24, 2012. (page 16)

[57] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 441–450. ACM, 2011. (page 16)

[58] Keisuke Tajimi, Teppei Konishi, Nobuchika Sakata, and S Sishida. Stabilization method for a hip-mounted projector using an inertial sensor. *Advances inWearable Computing*, pages 37–44, 2009. (page 16)

[59] Nobuchika Sakata, Takeshi Kurata, Takekazu Kato, Masakatsu Kourogi, and Hideaki Kuzuoka. Wacl: Supporting telecommunications using wearable active camera with laser pointer. In *ISWC*, volume 2003, page 7th. Citeseer, 2003. (page 16)

[60] Nobuchika Sakata, Takeshi Kurata, and Hideaki Kuzuoka. Visual assist with a laser pointer and wearable display for remote collaboration. 2006. (page 16)

[61] Taro Maeda and Hideyuki Ando. Wearable scanning laser projector (wslp) for augmenting shared space. In *SIGGRAPH Sketches*, page 109, 2004. (page 16)

[62] Björn Schwerdtfeger, Daniel Pustka, Andreas Hofhauser, and Gudrun Klinker. Using laser projectors for augmented reality. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 134–137. Citeseer, 2008. (page 16)

[63] Jürgen Scheible, Achim Hoth, Julian Saal, and Haifeng Su. Displaydrone: a flying robot based interactive display. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*, pages 49–54. ACM, 2013. (page 16, 17)

[64] Yoshito Hosomizo, Daisuke Iwai, and Kosuke Sato. A flying projector stabilizing image fluctuation. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pages 31–32. IEEE, 2014. (page 17)

[65] David Mirk and Helmut Hlavacs. Using drones for virtual tourism. In *International Conference on Intelligent Technologies for Interactive Entertainment*, pages 144–147. Springer, 2014. (page 17)

[66] John Paulin Hansen, Alexandre Alapetite, I Scott MacKenzie, and Emilie Møllenbach. The use of gaze to control drones. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 27–34. ACM, 2014. (page 18)

[67] Keita Higuchi and Jun Rekimoto. Flying head: a head motion synchronization mechanism for unmanned aerial vehicle control. In *CHI'13 Extended Abstracts*, pages 2029–2038. ACM, 2013. (page 18, 137)

[68] Jessie YC Chen, Ellen C Haas, and Michael J Barnes. Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6):1231–1245, 2007. (page 18)

[69] Kwangsu Cho, Minhee Cho, and Jongwoo Jeon. Fly a drone safely: Evaluation of an embodied egocentric drone controller interface. *Interacting with Computers*, 2016. (page 18)

[70] Shunichi Kasahara, Ryuma Niiyama, Valentin Heun, and Hiroshi Ishii. extouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proc. ACM TEI*, pages 223–228, 2013. (page 18)

[71] Sunao Hashimoto, Akihiko Ishida, Masahiko Inami, and Takeo Igarashi. Touchme: An augmented reality based remote robot manipulation. In *Proc. ICAT*, 2011. (page 18)

[72] Daniel Saakes, Vipul Choudhary, Daisuke Sakamoto, Masahiko Inami, and Takeo Lgarashi. A teleoperating interface for ground vehicles using autonomous flying cameras. In *Proc. ICAT*, pages 13–19, 2013. (page 18)

[73] Maki Sugimoto, Georges Kagotani, Hideaki Nii, Naoji Shiroma, Fumitoshi Matsuno, and Masahiko Inami. Time follower's vision: a teleoperation interface with past images. *IEEE Computer Graphics and Applications*, 25(1):54–63, 2005. (page 18)

[74] James T Hing, Keith W Sevcik, and Paul Y Oh. Development and evaluation of a chase view for uav operations in cluttered environments. *Journal of Intelligent & Robotic Systems*, 57(1):485–503, 2010. (page 18)

[75] Chitra R Karanam and Yasamin Mostofi. 3d through-wall imaging with unmanned aerial vehicles using wifi. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 131–142. IEEE, 2017. (page 18)

[76] Louis-Pierre Bergé, Nabil Aouf, Thierry Duval, and Gilles Coppin. Generation and vr visualization of 3d point clouds for drone target validation assisted by an operator. In *Computer Science and Electronic Engineering (CEEC), 2016 8th*, pages 66–70. IEEE, 2016. (page 18)

[77] Robert Valner, Karl Kruusamäe, and Mitch Pryor. Temoto: Intuitive multi-range telerobotic system with natural gestural and verbal instruction interface. *Robotics*, 7(1):9, 2018. (page 19)

[78] Hooman Hedayati, Michael Walker, and Daniel Szafir. Improving collocated robot teleoperation with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 78–86. ACM, 2018. (page 19)

[79] Michael E Walker, Hooman Hedayati, and Daniel Szafir. Robot teleoperation with augmented reality virtual surrogates. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 202–210. IEEE, 2019. (page 19, 139)

[80] Jesse Paterson, Jiwoong Han, Tom Cheng, Paxtan Laker, David McPherson, Joseph Menke, and Allen Yang. Improving usability, efficiency, and safety of uav path planning through a virtual reality interface. *arXiv preprint arXiv:1904.08593*, 2019. (page 19)

[81] Werner Alexander Isop, Christoph Gebhardt, Tobias Nägeli, Friedrich Fraundorfer, Otmar Hilliges, and Dieter Schmalstieg. High-level teleoperation system for aerial exploration of indoor environments. *Frontiers in Robotics and AI*, 6:95, 2019. (page 19)

[82] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Designing planning and control interfaces to support user collaboration with flying robots. *The International Journal of Robotics Research*, 36(5-7):514–542, 2017. (page 19)

[83] Kwangsu Cho, Minhee Cho, and Jongwoo Jeon. Fly a drone safely: Evaluation of an embodied egocentric drone controller interface. *Interacting with computers*, 29 (3):345–354, 2017. (page 19)

[84] Maik Riestock, Frank Engelhardt, Sebastian Zug, and Nico Hochgeschwender. User study on remotely controlled uavs with focus on interfaces and data link quality. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3394–3400. IEEE, 2017. (page 19, 107)

[85] John Thomason, Photchara Ratsamee, Kiyoshi Kiyokawa, Pakpoom Kriangkomol, Jason Orlosky, Tomohiro Mashita, Yuki Uranishi, and Haruo Takemura. Adaptive view management for drone teleoperation in complex 3d structures. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 419–426. ACM, 2017. (page 19)

[86] CARL N von Ende. Repeated-measures analysis. *Design and analysis of ecological experiments*, 8:134–157, 2001. (page 20)

[87] R Clifford Blair and James J Higgins. Comparison of the power of the paired samples t test to that of wilcoxon's signed-ranks test under various population shapes. *Psychological Bulletin*, 97(1):119, 1985. (page 20)

[88] Taro Yamane. Statistics: An introductory analysis. 1973. (page 20)

[89] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988. (page 20, 92, 120)

[90] Viswanath Venkatesh, Susan A Brown, and Hillol Bala. Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS quarterly*, pages 21–54, 2013. (page 20)

[91] Neil J Salkind and Samuel B Green. *Using SPSS for Windows and Macintosh: Analyzing and understanding data*. Pearson Prentice Hall, 2005. (page 21)

[92] Thomas Pederson, Lars-Erik Janlert, and Dipak Surie. a situative space model for mobile mixed-reality computing. *IEEE pervasive computing*, 10(4):73–83, 2011. (page 21, 24)

[93] Susumu Tachi, Hirohiko Arai, Taro Maeda, Eimei Oyama, Naoki Tsunemoto, and Yasuyuki Inoue. Tele-existence in real world and virtual world. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 193–198. IEEE, 1991. (page 22)

[94] EM Griffin. *A first look at communication theory*. McGraw-Hill, 2006. (page 23)

[95] O Michael Watson. *Proxemic behavior: A cross-cultural study*, volume 8. Walter de Gruyter GmbH & Co KG, 2014. (page 23)

[96] Ilhan Bae and Jeonghye Han. Does height affect the strictness of robot assisted teacher? In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 73–74. ACM, 2017. (page 24)

[97] James A Landay, Jessica R Cauchard, et al. Drone & wo: Cultural influences on human-drone interaction techniques. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6794–6799. ACM, 2017. (page 24)

[98] Parastoo Abtahi, David Y Zhao, LE Jane, and James A Landay. Drone near me: Exploring touch-based human-drone interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):34, 2017. (page 24, 42)

[99] Jeonghye Han and Ilhan Bae. Social proxemics of human-drone interaction: Flying altitude and size. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 376–376. ACM, 2018. (page 24)

[100] Thomas Pederson, Lars-Erik Janlert, and Dipak Surie. Towards a model for egocentric interaction with physical and virtual objects. In *Proceedings of the 6th Nordic*

Conference on Human-Computer Interaction: Extending Boundaries, pages 755–758. ACM, 2010. (page 24, 27)

[101] Dipak Surie, Thomas Pederson, Fabien Lagriffoul, Lars-Erik Janlert, and Daniel Sjölie. Activity recognition using an egocentric perspective of everyday objects. In *International Conference on Ubiquitous Intelligence and Computing*, pages 246–257. Springer, 2007. (page 25)

[102] Dipak Surie, Berker Baydan, and Helena Lindgren. proxemics awareness in kitchen as-a-pal: tracking objects and human in perspective. In *2013 9th International Conference on Intelligent Environments*, pages 157–164. IEEE, 2013. (page 26)

[103] Somaiyeh MahmoudZadeh, David MW Powers, and Reza Bairam Zadeh. *Autonomy and Unmanned Vehicles: Augmented Reactive Mission and Motion Planning Architecture*. Springer, 2018. (page 29, 102)

[104] Ivan Poupyrev, Tadao Ichikawa, Suzanne Weghorst, and Mark Billinghurst. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, volume 17, pages 41–52. Wiley Online Library, 1998. (page 38, 133)

[105] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 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*, pages 79–80. ACM, 1996. (page 38)

[106] Richard H Jacoby, Mark Ferneau, and Jim Humphries. Gestural interaction in a virtual environment. In *Stereoscopic Displays and Virtual Reality Systems*, volume 2177, pages 355–365. International Society for Optics and Photonics, 1994. (page 38)

[107] Ivan Poupyrev, Suzanne Weghorst, Mark Billinghurst, and Tadao Ichikawa. A framework and testbed for studying manipulation techniques for immersive vr. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 21–28. ACM, 1997. (page 38)

[108] SL Reeves, Chaowadee Varakamin, and CJ Henry. The relationship between armspan measurement and height with special reference to gender and ethnicity. *european Journal of clinical nutrition*, 50(6):398–400, 1996. (page 38)

[109] Jodie M Plumert, Joseph K Kearney, James F Cremer, and Kara Recker. Distance perception in real and virtual environments. *ACM Transactions on Applied Perception (TAP)*, 2(3):216–233, 2005. (page 38, 133)

[110] Matt Whitlock, Ethan Harnner, Jed R Brubaker, Shaun Kane, and Danielle Albers Szafir. Interacting with distant objects in augmented reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 41–48. IEEE, 2018. (page 38)

[111] Stephen Voida, Mark Podlaseck, Rick Kjeldsen, and Claudio Pinhanez. A study on the manipulation of 2d objects in a projector/camera-based augmented reality environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 611–620. ACM, 2005. (page 38)

[112] J González-Jiménez, JG Monroy, and JL Blanco. Robots that can smell: motivation and problems. In *Submitted to 15th international symposium on olfaction and electronic nose (ISOEN) Google Scholar*, 2013. (page 42, 130)

[113] Werner Alexander Isop, Jesus Pestana, Gabriele Ermacora, Friedrich Fraundorfer, and Dieter Schmalstieg. Micro aerial projector-stabilizing projected images of an airborne robotics projection platform. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5618–5625. IEEE, 2016. (page 43, 62, 126)

[114] Okan Erat, Werner Alexander Isop, Denis Kalkofen, and Dieter Schmalstieg. Drone-augmented human vision: Exocentric control for drones exploring hidden areas. *IEEE transactions on visualization and computer graphics*, 24(4):1437–1446, 2018. (page 43, 126)

[115] Paolo Tripicchio, Massimo Satler, Giacomo Dabisias, Emanuele Ruffaldi, and Carlo Alberto Avizzano. Towards smart farming and sustainable agriculture with drones. In *Intelligent Environments (IE), 2015 International Conference on*, pages 140–143. IEEE, 2015. (page 44)

[116] Jesus Pestana Puerta, Michael Maurer, Daniel Muschick, Devesh Adlakha, Horst Bischof, and Friedrich Fraundorfer. Package delivery experiments with a camera drone. 2017. (page 44)

[117] Mario Silvagni, Andrea Tonoli, Enrico Zenerino, and Marcello Chiaberge. Multipurpose uav for search and rescue operations in mountain avalanche events. *Geomatics, Natural Hazards and Risk*, 8(1):18–33, 2017. (page 44)

[118] Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39, 2012. (page 45, 58)

[119] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system.

In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009. (page 45, 56, 62, 66, 70)

[120] ORBBEC. Astra pro. `https://orbbec3d.com/product-astra-pro/`, 2018. Accessed: 10-12-2018. (page 45)

[121] DJI. Marvic. `https://www.dji.com/de/mavic-2`, 2018. Accessed: 10-12-2018. (page 46)

[122] DJI. Spark. `https://www.dji.com/de/spark`, 2018. Accessed: 10-12-2018. (page 46)

[123] Bitcraze. Crazyflie. `https://www.bitcraze.io/crazyflie-2/`, 2018. Accessed: 10-12-2018. (page 46)

[124] Parrot. Bebop 2. `https://www.parrot.com/eu/drones/parrot-bebop-2`, 2018. Accessed: 10-12-2018. (page 47, 50, 65, 110)

[125] Davide Falanga, Elias Mueggler, Matthias Faessler, and Davide Scaramuzza. Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5774–5781. IEEE, 2017. (page 49, 117)

[126] Austro Control. Regulations for unmanned aerial vehicles. `https://www.austrocontrol.at/en/aviation_agency/licenses__permissions/flight_permissions/rpas`, 2018. Accessed: 10-12-2018. (page 50)

[127] Conrad. Lightweight carbon fiber frame. `https://www.amazon.com/DRONE-ART-Quadcopter-Lightweight-X-Design/dp/B07F27FGTZ`, 2018. Accessed: 10-12-2018. (page 50)

[128] S. Ramamurthy. Thrust models. `https://de.scribd.com/document/101910324/Static-Thrust-for-Propeller`, 2018. Accessed: 10-12-2018. (page 52)

[129] Young-Cheol Choi and Hyo-Sung Ahn. Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. *IEEE/ASME transactions on mechatronics*, 20(3):1179–1192, 2014. (page 53)

[130] Randal Beard. Quadrotor dynamics and control rev 0.1. 2008. (page 54)

[131] John G Ziegler and Nathaniel B Nichols. Optimum settings for automatic controllers. *trans. ASME*, 64(11), 1942. (page 55)

[132] M Sc Carsten Seidel. Entwurf und stabilitätsanalyse der höhenregelung und wandvermeidung des finken ii quadrokopters. (page 55)

[133] M Kamran Joyo, D Hazry, S Faiz Ahmed, M Hassan Tanveer, Faizan A Warsi, and AT Hussain. Altitude and horizontal motion control of quadrotor uav in the presence of air turbulence. In *2013 IEEE Conference on Systems, Process & Control (ICSPC)*, pages 16–20. IEEE, 2013. (page 55)

[134] Ruesch Andreas. Dynamics identification & validation, and position control for a quadrotor. *Swiss Federal Institude of Technology Zurich, Spring Term*, 2010. (page 55)

[135] iNavFlight. inav. `https://github.com/iNavFlight/inav/wiki/Developer-info`, 2018. Accessed: 10-12-2018. (page 58, 70)

[136] Optitrack. Flex 13. `https://optitrack.com/products/flex-13/`, 2018. Accessed: 10-12-2018. (page 58)

[137] Ilmir Z Ibragimov and Ilya M Afanasyev. Comparison of ros-based visual slam methods in homogeneous indoor environment. In *Positioning, Navigation and Communications (WPNC), 2017 14th Workshop on*, pages 1–6. IEEE, 2017. (page 58)

[138] Kluge. Cubic-spline library. `https://kluge.in-chemnitz.de/opensource/spline/`, 2018. Accessed: 10-12-2018. (page 58)

[139] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013. (page 59, 114)

[140] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014. (page 59, 115)

[141] Mathieu Labbe and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3): 734–745, 2013. (page 59, 115)

[142] Mark Billinghurst and Hirokazu Kato. Collaborative mixed reality in proceedings of the first international symposium on mixed reality (ismr99). mixed reality–merging real and virtual worlds, 1999. (page 59)

[143] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407. IEEE, 2011. (page 59)

[144] Lorenz Meier. Px4 firmware release 1.3.4. `https://github.com/PX4/Firmware/releases/tag/v1.3.4`, 2018. Accessed: 10-12-2018. (page 60)

[145] Mani Monajjemi. Bebop autonomy driver. Homepage, 09 2015. Retrieved November 1, 2018 from `https://github.com/AutonomyLab/bebop_autonomy`. (page 65, 110)

[146] Alexander Domahidi and Juan Jerez. Forces pro: Code generation for embedded optimization, 2016. (page 66, 115)

[147] Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *2011 IEEE International Conference on Robotics and Automation*, pages 2992–2997. IEEE, 2011. (page 70)

[148] Veljko Milanovic, Gabriel A Matus, and Daniel T McCormick. Gimbal-less monolithic silicon actuators for tip-tilt-piston micromirror applications. *IEEE journal of selected topics in quantum electronics*, 10(3):462–471, 2004. (page 71, 73)

[149] Miroslav Lichvar Richard Curnow. User guide for the chrony suite. `https://web.archive.org/web/20150907232505/http://chrony.tuxfamily.org/manual.html`, 2015. Accessed: 2015-09-14. (page 72)

[150] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. (page 73, 74)

[151] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`, 2012. (page 74)

[152] Ulrich Neumann, Suya You, Jinhui Hu, Bolan Jiang, and JongWeon Lee. Augmented virtual environments (ave): Dynamic fusion of imagery and 3d models. In *IEEE Virtual Reality, 2003. Proceedings.*, pages 61–67. IEEE, 2003. (page 84)

[153] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. The magicbook-moving seamlessly between reality and virtuality. *IEEE Computer Graphics and applications*, 21(3):6–8, 2001. (page 84, 86)

[154] Mark R Mine, Frederick P Brooks Jr, and Carlo H Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings SIGGRAPH*, pages 19–26, 1997. (page 85)

[155] Markus Tatzgern, Raphael Grasset, Denis Kalkofen, and Dieter Schmalstieg. Transitional augmented reality navigation for live captured scenes. In *2014 IEEE Virtual Reality (VR)*, pages 21–26. IEEE, 2014. (page 87)

[156] Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *CHI*, volume 95, pages 265–272, 1995. (page 99)

[157] Kristen Stubbs, Pamela J Hinds, and David Wettergreen. Autonomy and common ground in human-robot interaction: A field study. *IEEE Intelligent Systems*, 22(2), 2007. (page 101)

[158] Jennifer L Burke and Robin R Murphy. Situation awareness and task performance in robot-assisted technical search: Bujold goes to bridgeport. *citeseer. ist. psu. edu/burke04situation. html*, 2004. (page 101, 106)

[159] J Alan Atherton and Michael A Goodrich. Supporting remote manipulation with an ecological augmented virtuality interface. In *Proc. of the AISB Symposium on New Frontiers in Human-Robot Interaction, Edinburgh, Scotland*, pages 381–394, 2009. (page 101, 108)

[160] Michael A Goodrich, Timothy W McLain, Jeffrey D Anderson, Jisang Sun, and Jacob W Crandall. Managing autonomy in robot teams: observations from four experiments. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 25–32. ACM, 2007. (page 102)

[161] David J Bruemmer, Donald D Dudenhoeffer, and Julie L Marble. Dynamic-autonomy for urban search and rescue. In *AAAI mobile robot competition*, pages 33–37, 2002. (page 102)

[162] Sebastian Muszynski, Jörg Stückler, and Sven Behnke. Adjustable autonomy for mobile teleoperation of personal service robots. In *RO-MAN, 2012 IEEE*, pages 933–940. IEEE, 2012. (page 102)

[163] Hope Reese. Autonomous driving levels 0 to 5: Understanding the differences. Homepage, 2016. Retrieved November 1, 2018 from `https://www.techrepublic.com/article/autonomous-driving-levels-0-to-5-understanding-the-differences/`. (page 102)

[164] Hui-Min Huang, Kerry Pavek, James Albus, and Elena Messina. Autonomy levels for unmanned systems (alfus) framework: An update. In *Unmanned Ground Vehicle Technology VII*, volume 5804, pages 439–449. International Society for Optics and Photonics, 2005. (page 102)

[165] Hui-Min Huang, Kerry Pavek, Brian Novak, James Albus, and E Messin. A framework for autonomy levels for unmanned systems (alfus). *Proceedings of the AUVSI's Unmanned Systems North America*, pages 849–863, 2005. (page 102, 106, 107)

[166] Daniel P Kun, Erika Baksane Varga, and Zsolt Toth. Ontology based navigation model of the ilona system. In *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000479–000484. IEEE, 2017. (page 103, 116)

[167] Stefan Lichiardopol. A survey on teleoperation. 2007. (page 104)

[168] Optitrack. Flex 13. Homepage, 2019. Retrieved March 1, 2019 from `www.optitrack.com/products/flex-13/`. (page 104)

[169] Stela H Seo, Daniel J Rea, Joel Wiebe, and James E Young. Monocle: Interactive detail-in-context using two pan-and-tilt cameras to improve teleoperation effectiveness. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 962–967. IEEE, 2017. (page 106)

[170] Alberto Valero-Gomez, Paloma De La Puente, and Miguel Hernando. Impact of two adjustable-autonomy models on the scalability of single-human/multiple-robot teams for exploration missions. *Human factors*, 53(6):703–716, 2011. (page 107)

[171] Holly A Yanco, Jill L Drury, and Jean Scholtz. Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction*, 19(1):117–149, 2004. (page 107)

[172] Curtis W Nielsen, Michael A Goodrich, and Robert W Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, 23(5): 927–941, 2007. (page 107, 108)

[173] Collin Johnson. Topological mapping and navigation in real-world environments. 2018. (page 108, 112, 122)

[174] António Espingardeiro. Human performance in telerobotics operations. In *Advanced Materials Research*, volume 403, pages 772–779. Trans Tech Publ, 2012. (page 108)

[175] Zdeněk Materna, Michal Španěl, Marcus Mast, Vítězslav Beran, Florian Weisshardt, Michael Burmester, and Pavel Smrž. Teleoperating assistive robots: A novel user interface relying on semi-autonomy and 3d environment mapping. *Journal of Robotics and Mechatronics*, 29(2):381–394, 2017. (page 111)

[176] H. Umari and S. Mukhopadhyay. Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1396–1402, 09 2017. doi: 10.1109/IROS.2017.8202319. (page 113)

[177] Richard Bormann, Florian Jordan, Wenzhe Li, Joshua Hampp, and Martin Hägele. Room segmentation: Survey, implementation, and analysis. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1019–1026. IEEE, 2016. (page 113, 114)

[178] Lydia E. Kavraki and Jean-Claude Latombe. Probabilistic roadmaps for robot path planning, 1998. (page 114)

[179] Christoph Gebhardt, Benjamin Hepp, Tobias Nägeli, Stefan Stevšić, and Otmar Hilliges. Airways: Optimization-based planning of quadrotor trajectories according to high-level user goals. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2508–2519. ACM, 2016. (page 114)

[180] Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Trans. Graph.*, 36(4):132:1–132:10, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073712. URL `http://doi.acm.org/10.1145/3072959.3073712`. (page 114)

[181] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges. Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. *IEEE Robotics and Automation Letters*, 2(3):1696–1703, 7 2017. doi: 10.1109/LRA.2017.2665693. (page 114)

[182] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. (page 116)

[183] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, May 2011. doi: 10.1109/ICRA.2011.5979561. (page 116)

[184] Razor. Orbweaver. Homepage, 2015. Retrieved March 1, 2019 from `https://www.razer.com/gaming-keyboards-keypads/razer-orbweaver-chroma`. (page 117)

[185] Jeffrey A Delmerico, David Baran, Philip David, Julian Ryde, and Jason J Corso. Ascending stairway modeling from dense depth imagery for traversability analysis. In *2013 IEEE International Conference on Robotics and Automation*, pages 2283–2290. IEEE, 2013. (page 118)

[186] Rabin K Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric A Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. In *NSDI*, volume 1, page 1, 2007. (page 118)

[187] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, et al. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013. (page 118)

[188] Qihao Zhang, Wei Zhao, Shengnan Chu, Lei Wang, Jun Fu, Jiangrong Yang, and Bo Gao. Research progress of nuclear emergency response robot. In *IOP Conference Series: Materials Science and Engineering*, volume 452, page 042102. IOP Publishing, 2018. (page 118)

[189] Subodh Bhandari, Steven Viska, Harsh Shah, Callie Chen, Guiseppe Tonini, and Scott Kline. Autonomous navigation of a quadrotor in indoor environments for

surveillance and reconnaissance. In *AIAA Infotech@ Aerospace*, page 0717. AIAA SciTech Forum, 2015. (page 119)

[190] WMA DECLARATION OF HELSINKI. Ethical principles for medical research involving human subjects. Homepage, 2013. Retrieved November 1, 2018 from `https://www.wma.net/policy/current-policies/`. (page 120)

[191] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011. (page 122)

[192] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5): 1255–1262, 2017. doi: 10.1109/TRO.2017.2705103. (page 122)

[193] Keita Higuchi, Tetsuro Shimada, and Jun Rekimoto. Flying sports assistant: external visual imagery representation for sports training. In *Proceedings of the 2nd Augmented Human International Conference*, page 7. ACM, 2011. (page 131)

[194] Yunkyung Kim and Bilge Mutlu. How social distance shapes human–robot interaction. *International Journal of Human-Computer Studies*, 72(12):783–795, 2014. (page 131)

[195] Nick Statt. Microsoft's hololens explained: How it works and why it's different. `https://www.cnet.com/news/microsoft-hololens-explained-how-it-works-and-why-its-different/`. Accessed: 2015-01-24. (page 132)

[196] Niels Christian Nilsson, Tabitha Peck, Gerd Bruder, Eri Hodgson, Stefania Serafin, Mary Whitton, Frank Steinicke, and Evan Suma Rosenberg. 15 years of research on redirected walking in immersive virtual environments. *IEEE computer graphics and applications*, 38(2):44–56, 2018. (page 134)

[197] Nicolas Ducheneaut, Ming-Hui Wen, Nicholas Yee, and Greg Wadley. Body and mind: a study of avatar personalization in three virtual worlds. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1151–1160. ACM, 2009. (page 139)

[198] Hiroo Iwata, Hiroaki Yano, Hiroyuki Fukushima, and Haruo Noma. Circulafloor [locomotion interface]. *IEEE Computer Graphics and Applications*, 25(1):64–67, 2005. (page 139)

[199] Stefan Marks, Javier E Estevez, and Andy M Connor. Towards the holodeck: fully immersive virtual reality visualisation of scientific and engineering data. In *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*, pages 42–47. ACM, 2014. (page 139)