Enabling Ubiquitous Augmented Reality with Crowdsourced Indoor Mapping and Localization

Marius Noreikis
Enabling Ubiquitous Augmented Reality with Crowdsourced Indoor Mapping and Localization

Marius Noreikis

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall T2 of the school on 10th June 2019 at 12 noon.

Aalto University
School of Electrical Engineering
Department of Communications and Networking
Mobile Cloud Computing
Supervising professor
Professor Yu Xiao, Aalto University, Finland

Preliminary examiners
Professor Fan Ye, Stony Brook University, USA
Professor Petteri Nurmi, University of Helsinki, Finland

Opponent
Professor Romit Roy Choudhury, University of Illinois at Urbana-Champaign, USA

Aalto University publication series
DOCTORAL DISSERTATIONS 86/2019

© 2019 Marius Noreikis

ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Unigrafia Oy
Helsinki 2019

Finland
Author
Marius Noreikis

Name of the doctoral dissertation
Enabling Ubiquitous Augmented Reality with Crowdsourced Indoor Mapping and Localization

Publisher School of Electrical Engineering
Unit Department of Communications and Networking
Series Aalto University publication series DOCTORAL DISSERTATIONS 86/2019
Field of research Networking Technology

Permission submitted 18 January 2019 Date of the defence 10 June 2019
Monograph ☐ Article dissertation ☒ Essay dissertation

Abstract
With a proliferation of sensor-rich small form factor devices such as smart glasses and smartphones, augmented reality (AR) applications attracted tremendous interest from both, industry professionals and academics. AR applications enrich the real-world view, seen by a user, with additional information such as computer-generated 3D artifacts that blend seamlessly with real-world objects. Although popular AR applications, especially AR games, are already used by millions of people, enabling shared and ubiquitous AR experiences is still challenging. It is still highly challenging to provide persistent AR experience which aligns artificial objects seamlessly with designated real-world places and allows multiple users to simultaneously perceive the same objects. Furthermore, enabling truly ubiquitous AR requires AR applications to work in arbitrary environments, while users access the applications via commodity devices such as smartphones.

In this dissertation, we focus on enabling technologies for ubiquitous multi-user AR applications for indoor environments. We observe that an accurate, real-time localization system is required, in order to provide ubiquitous AR experience indoors. Consequently, we investigate the applicability of computer vision-based techniques for efficient indoor mapping and study how the maps can be used to enable accurate six-degrees-of-freedom positioning, suitable for AR-based applications.
Specifically, we investigate applicability of visual crowdsourcing for mapping and providing accurate and infrastructure-less indoor localization and navigation services. Furthermore, we develop mobile AR applications that use the developed indoor positioning services. We solve the challenge to enable energy-efficient and accurate real-time position and facing direction tracking, which is required to enable seamless AR experiences. Finally, we focus on deployment of the developed real-time AR-based systems on a hierarchical edge cloud environment. In particular, we focus on initial computing capacity planning that satisfies the Quality of Service requirements of the developed systems. In this dissertation we conduct empirical studies in order to answer the research questions. We develop a practical indoor mapping and localization system and a smartphone application that uses the localization system for AR-based indoor navigation. The results of this work provide basis for enabling ubiquitous AR experience within entertainment, productivity and social applications.

Keywords Augmented Reality, Indoor Mapping, Visual Crowdsourcing, Indoor Navigation, Capacity Planning

ISSN (printed) 1799-4934 ISSN (pdf) 1799-4942
Location of publisher Helsinki Location of printing Helsinki Year 2019
Pages 142

Preface

I would like to express special gratitude to my supervisor, Prof. Yu Xiao, who gave me the opportunity to join her research group in Aalto University. I thank her for guiding me through the vast field of scientific research and supporting my work during all stages of preparing this dissertation. Her invaluable advice, encouragement, and tremendous help made this dissertation possible.

I greatly appreciate the help of Prof. Antti Ylä-Jääski, with whom I started my doctoral research, and thank him for his great hints and invaluable support for the publication of my research.

My warmest thanks to my co-authors Dr. Jiang Dong, Dr. Yuki Sato, Prof. Yuming Jiang, Prof. Yang Chen, and Jiyao Hu. Your hard work, valuable comments, and insightful discussions resulted in the best publications included in this work. It was my great pleasure working with you.

I also wish to thank a number of colleagues who made my life as a doctoral candidate a fun, thrilling, and exciting experience. Thank you Vilen Looga, Sanja Šćepanović, Siddharth Rao, Pranvera Kortoçi, Petr Byvshev, Gopika Prem sankar, Truong-an Pham, Maria Montoya Freire, Chao Zhu, and Annika Stuke for lovely evenings, cheerful discussions around a lunch table, and for sharing the burden of PhD life.

I would like to thank to all my friends in Finland and around the world.

Special thanks to Prof. Petteri Nurmi from the University of Helsinki and Prof. Fan Ye from Stony Brook University, who made the effort to pre-examine my dissertation. Their insightful comments and valuable feedback helped me to improve the quality of this work and to produce a dissertation that meets the high standards of the scientific community.

My gratitude goes to Prof. Romit Roy Choudhury from University of Illinois at Urbana Champaign who graciously accepted to act as my opponent in the public defense of my dissertation.

I appreciate the financial support of the School of Electrical Engineering, Helsinki Institute of Information Technology, and the Nokia Scholarship Foundation.

Over the last three years, I have learned that pursuing a doctoral degree is a thrilling yet difficult task. I have also learned that completing the dissertation
Preface

would have been impossible without the help and support of the people closest to me. I send my warmest thanks to my loving family for their endless support, patience, and encouragement. My lovely fiancée Eglė stood by me and supported me during the toughest times and inspired to continue with my work. Therefore, I dedicate this thesis to them.

Espoo, April 19, 2019,

Marius Noreikis
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>3</td>
</tr>
<tr>
<td>Contents</td>
<td>5</td>
</tr>
<tr>
<td>List of Publications</td>
<td>7</td>
</tr>
<tr>
<td>Author's Contribution</td>
<td>9</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>13</td>
</tr>
<tr>
<td><strong>1. Introduction</strong></td>
<td>15</td>
</tr>
<tr>
<td>1.1 Research Questions and Scope</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Methodology</td>
<td>18</td>
</tr>
<tr>
<td>1.3 Contributions</td>
<td>20</td>
</tr>
<tr>
<td>1.4 Structure</td>
<td>21</td>
</tr>
<tr>
<td><strong>2. Background</strong></td>
<td>23</td>
</tr>
<tr>
<td>2.1 Indoor Mapping and Localization</td>
<td>23</td>
</tr>
<tr>
<td>2.1.1 Spatial Indoor Mapping</td>
<td>23</td>
</tr>
<tr>
<td>2.1.2 Mapping with 3D Point Clouds</td>
<td>25</td>
</tr>
<tr>
<td>2.1.3 Indoor Localization</td>
<td>27</td>
</tr>
<tr>
<td>2.1.4 Navigation Methods for Indoors</td>
<td>30</td>
</tr>
<tr>
<td>2.2 Crowdsourced Indoor Mapping</td>
<td>30</td>
</tr>
<tr>
<td>2.2.1 Mobile and Visual Crowdsensing</td>
<td>31</td>
</tr>
<tr>
<td>2.2.2 Different Types of Mobile Crowdsensing</td>
<td>32</td>
</tr>
<tr>
<td>2.2.3 Crowdsourced Mapping Techniques</td>
<td>32</td>
</tr>
<tr>
<td>2.3 Enabling Technologies for Augmented Reality</td>
<td>34</td>
</tr>
<tr>
<td>2.3.1 Tracking Methods for AR</td>
<td>35</td>
</tr>
<tr>
<td>2.3.2 Energy Efficient AR</td>
<td>37</td>
</tr>
<tr>
<td>2.4 Capacity Planning for Real-Time Systems at the Edge</td>
<td>37</td>
</tr>
<tr>
<td>2.5 Summary</td>
<td>38</td>
</tr>
</tbody>
</table>
3. Enabling Ubiquitous Augmented Reality with Crowdsourced Indoor Mapping and Localization 39
   3.1 Crowdsourced Mapping .................................. 39
      3.1.1 Indoor Mapping from Crowdsourced Data .......... 40
      3.1.2 Efficient Mapping with Visual Crowdsourcing ...... 42
   3.2 Pose Tracking for Seamless AR .......................... 44
      3.2.1 Seamless and Energy-Efficient AR ................. 45
      3.2.2 High-Accuracy Tracking with SLAM ............... 47
   3.3 Capacity Planning for Real-Time Compute-Intensive Systems 48
      3.3.1 Capacity Planning Framework ........................ 49
      3.3.2 Planning the Amount of Required Edge Nodes ...... 50
   3.4 Open Questions .......................................... 51

4. Conclusion 53

References 55

Errata 67

Publications 69
List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.


Author’s Contribution

Publication I: “ViNav: A Vision-based Indoor Navigation System for Smartphones”

The author of this thesis, together with the first author of this paper, proposed, designed, and implemented a crowdsourcing-based indoor navigation system. He developed algorithms for constructing navigation meshes from structure-from-motion (SfM) models and proposed a mechanism for updating the models. He also implemented an image-based localization service and designed floor change detection and multi-floor navigation algorithms and evaluated their performance. He and the first author of this paper conducted an evaluation of the developed system in an office building and a supermarket. He also evaluated the system's scalability and the deployment scenarios for public and edge cloud environments.

Publication II: “SnapTask: Towards Efficient Visual Crowdsourcing for Indoor Mapping”

The author of this thesis proposed, designed, and developed a guided participatory visual crowdsourcing system for efficient floor plan reconstruction. He proposed algorithms for generating floor plans, analyzing floor plan coverage and identifying locations for crowdsourced data collection. He also developed a system for reconstructing texture-less surfaces and, together with the other authors, designed an algorithm for adding the detected surfaces to SfM models. He evaluated the developed system in a library building.
Author's Contribution


The author of this thesis proposed, developed, and evaluated an augmented reality (AR) mobile application for indoor navigation. He designed a sensor fusion algorithm for combining image-based positioning with tracking based on inertial sensors. He studied the power consumption of AR applications and optimized the tracking algorithm to reduce energy consumption. He evaluated the developed algorithms and conducted a user study to evaluate the performance of the application in a building belonging to the Department of Computer Science at Aalto University.

Publication IV: “Low-cost mapping of RFID tags using reader-equipped smartphones”

The author of this thesis implemented a mobile application for accurately locating and tracking a smartphone in a 3D space, which is a necessary component of radio-frequency identification (RFID) mapping. Together with the first author of the paper, he designed and implemented a low-cost RFID tag mapping system and conducted a field test of the developed system in a university library.

Publication V: “QoS-oriented Capacity Planning for Edge Computing”

The author of this thesis proposed a capacity planning framework for estimating the initial cloud capacity required to satisfy an application’s quality of service requirements. He implemented tools for measuring an application’s resource utilization patterns and estimating the number of edge cloud nodes required. He formulated and proved three design assumptions that improve the utilization of edge cloud nodes. He evaluated the proposed framework by deploying an AR-based navigation system.

Publication VI: “Edge Capacity Planning for Real Time Compute-Intensive Applications”

The author of this thesis designed and implemented an edge node count estimation algorithm based on queuing theory to supplement the previously developed capacity planning framework. He evaluated the proposed models with multiple cloud setups to confirm the applicability of the models for the node count estimation problem. He evaluated the node count estimation algorithm by planning the capacity for an AR-based navigation and information system.
Language check

The language of my dissertation has been checked by a Scribbr editor Rebecca. I have personally examined and accepted/rejected the results of the language check one by one. This has not affected the scientific content of my dissertation.
Abbreviations

**6DoF** Six-Degrees-of-Freedom

**AoA** Angle-of-Arrival

**AR** Augmented Reality

**CAD** Computer-Aided drawing

**CNN** Convolutional Neural Network

**CPU** Central Processing Unit

**GNSS** Global Navigation Satellite Systems

**GPU** Graphics Processing Unit

**IMU** Inertial Measurement Unit

**MCS** Mobile Crowdsensing

**ORB** Oriented features from accelerated segment test and Rotated Binary robust independent elementary features

**PoI** Point of Interest

**QoI** Quality of Information

**QoS** Quality of Service

**QR** quick-response codes

**RANSAC** Random Sample Consensus algorithm

**RFID** Radio-frequency identification

**RSS** Received Signal Strength

**SfM** Structure-from-Motion

**SIFT** Scale Invariant Feature Transform
Abbreviations

**SLAM** Simultaneous Localization and Mapping

**SURF** Speeded-Up Robust Features

**UWB** Ultra Wide Band

**VCS** Visual Crowdsensing
1. Introduction

Aside from being a prominent topic in science fiction for decades, ideas about altered reality have also long existed within the industrial and science communities. In the early 90s, T.P. Caudell coined the term “augmented reality” and together with D.W. Mizell published an article describing the first AR application, which was meant for efficient delivery of various wiring instructions to factory workers [36]. In general, AR refers to superimposing computer-generated information into a real-world view that is seen by a user. Over the past few years it has gained lots of attention from both industry professionals and academics due to the release of small form factor AR smart glasses, such as Vuzix Blade [12] and Microsoft HoloLens [8], as well as an increasing number of AR applications developed for widely used smartphones. As a result, the global market for AR is forecasted to reach over $61 billion by 2023 [14].

Nevertheless, AR is still a developing technology, and many technical challenges must be solved to offer immersive and ubiquitous AR experiences. Ubiquitous AR should offer persistence of AR artifacts regardless of the environment, allowing the placement of persistent AR objects in large outdoor and indoor spaces. It must ensure that multiple users can perceive the same AR objects that are correctly positioned within a real-world view and must support pervasive consumer devices. Current smart devices, such as smartphones, are capable of rendering stunning 3D visualizations at a high frame rate, enabling immersive life-like augmentation. However, techniques for aligning the rendered 3D objects with the real-world view and ensuring their persistence are still lacking. So far, most AR applications for smartphones operate locally with the help of fiducial and pre-trained markers. When a user scans such a marker, the AR experience starts, although it is limited to a radius of a few meters around the marker. Because of the need to install markers, this approach is not suitable for enabling AR in large spaces, such as outdoor parks and multi-floor offices.

Global navigation satellite systems (GNSS) have been used to enable AR in large outdoor environments, where the AR device obtains a position from a GNSS and estimates its facing direction using a digital compass. However, the accuracy of GNSS positioning and compass sensors is often insufficient to achieve accurate placement of 3D objects. The challenges related to 3D artifact
Introduction

placement are even greater in indoor environments, where GNSS signals are not available. This forces AR developers to position AR artifacts in approximate or even arbitrary positions around the AR application user, fundamentally disabling shared and multi-user AR experiences, as two people cannot perceive the same augmented scene. Therefore, an accurate and reliable positioning system must be established to enable truly persistent and shared AR experiences in indoor environments.

In recent years, various techniques have been proposed for indoor positioning. A considerable amount of research has been conducted on positioning based on radio signals. Researchers have proposed using Wi-Fi fingerprinting techniques [18], Bluetooth beacons [89] or ultra-wideband (UWB) devices [59]. However, accurate positioning systems based on radio signals require extensive deployment of radio beacons which is rarely feasible in large indoor areas. To overcome this issue, some studies have shown that geomagnetic field measurements can be used to calculate a user’s location within a premises, in this way eliminating the need for additional hardware [84]. However, systems based on fingerprinting of radio signals, or magnetic fields can initially provide only the position of a user and require the user to move for a certain period of time to estimate the facing direction. This is a major drawback, as AR applications require instant six-degrees-of-freedom (6DoF) positioning, which includes both the position and facing direction of a user within a 3D space. Recently vision-based positioning systems have attracted a lot of attention [110]. Not only do they provide 6DoF positioning but they are generally also infrastructure-less. Furthermore, for positioning, they use images from a phone’s camera, which is invariably enabled during the use of an AR application. These unique features make computer vision-based positioning a choice worth considering for implementing the 6DoF positioning required for AR. However, several important challenges must still be addressed before vision-based positioning can be adapted for ubiquitous AR.

First, most vision based positioning systems [54, 136, 93] require 3D maps of the environment. Constructing such a map requires an extensive site survey, which involves collecting photographs and videos of the environment. This is a laborious process, as every major indoor surface and object must be captured in the camera footage. To address the issue, several studies have proposed the use of visual crowdsourcing [62], which helps to collectively capture and process images from the same environment. Second, vision-based positioning algorithms are notably more compute-intensive than other positioning techniques, mainly due to complex processing of large quantities of visual data. This poses a challenge for continuous positioning, especially with computing power-constrained and energy-constrained devices such as smartphones. Finally, computer vision-based positioning systems have not yet been widely adopted and it is yet unclear what the best deployment scenarios for such systems are. In particular, techniques for assessing the initial required computing capacity for deploying such systems are significantly lacking.
In our work, we aim to solve the challenges of efficiently obtaining indoor maps using computer vision-based algorithms and enabling six-degrees-of-freedom positioning for AR applications such as AR-based indoor navigation. Furthermore, based on the developed positioning system, we propose ways of implementing mobile AR applications that require low power consumption and achieve high accuracy in embedding augmented information into the real-world view. Finally, we study practical deployment scenarios of such systems in hierarchical edge cloud environments and develop techniques for calculating the initial computing capacity required before deploying the real-time image-based positioning system. In the subsequent sections, we elaborate on the research questions, scope, and contributions of this dissertation.

1.1 Research Questions and Scope

In this dissertation, we ask the following question: How can robust indoor maps be constructed using visual crowdsourcing and how can such maps be used to enable ubiquitous AR applications such as AR-based indoor navigation? In particular, we focus on developing computer vision-based mapping for 6DoF indoor positioning and navigation, designing accurate tracking algorithms for AR-enabled mobile applications, and on planning the deployment of the whole system in an edge cloud infrastructure. Since the research scope is extremely broad, we divide the research question into the following more detailed and more concrete sub-questions:

**RQ1** How can efficient crowdsourced spatial mapping for indoors be enabled?

The main objective is to design and develop quick and efficient mapping methods that are suitable for commodity devices such as smartphones. In addressing this question, we also study how the indoor maps can be further enhanced by detecting and adding floor interchange points and other points of interest (PoIs), such as rooms, shops, or discount products. Finally, we study how to develop accurate 6DoF localization methods based on obtained maps.

**RQ2** How can accurate and efficient tracking algorithms for mobile AR applications be developed?

Here, we focus on enabling continuous, accurate, and energy-efficient tracking of the position and facing direction of a smartphone in a 3D space. Regarding AR-enabled applications, we focus on two application scenarios: a pedestrian navigation scenario, where energy efficiency and the best possible user experience are the main priorities, and an inventory mapping scenario, where accurate 3D tracking is most important.

**RQ3** How can such a real-time AR navigation system be deployed in an edge cloud environment to ensure a seamless user experience for multiple simultaneous users?

Here, we focus on planning the initial edge cloud capacity to
satisfy the system’s quality of service requirements. This question is often overlooked, as most prior research focuses on computation offloading and requests scheduling rather than initial system capacity planning.

To answer the aforementioned questions, it is important not only to design theoretical models and low-level algorithms but also to develop prototypes of feature-full systems to evaluate results. Thus, we begin our research by designing and developing a practical indoor mapping system that accomplishes all the required steps from obtaining raw input photographs to providing accurate and robust positioning and navigation services. In a further step, we study how the services should be consumed by a mobile client application to ensure efficiency and a high-quality user experience. Finally, we conduct research on edge capacity planning to aid deployment of the developed systems.

The broad scope of this dissertation inevitably means omitting certain topics and challenges. In our work, we focus on vision-based indoor positioning and navigation, which is most suitable for ubiquitous AR. Therefore, we do not thoroughly investigate the applicability of other positioning systems, such as those based on Bluetooth beacons [89] or geomagnetic field fingerprinting [84]. However, the developed crowdsourced mapping and positioning solution can easily be used with the aforementioned technologies, either to improve system accuracy or to provide maps for positioning systems that require initial floor plans.

In this dissertation, we assemble practical proof-of-concept systems. Thus our study does not involve improving existing photogrammetry algorithms, such as structure from motion (SfM), or multi-view geometry. Similarly, we do not invent our own low-level visual tracking techniques but rather use state-of-the-art algorithms showing that they are applicable for solving the challenges in question. Finally, since the technology developed in this work directly applies to AR applications for large indoor environments, we do not focus on outdoor areas, as we believe that indoor environments are the most challenging to enable ubiquitous AR experiences.

### 1.2 Methodology

In order to answer the research questions set out in this work, we relied on empirical research methods. We designed and implemented proof-of-concept prototype systems that were later evaluated through field studies. Developing prototypes contributes to an understanding of how the system behaves in real-world environments and allows identification of corner cases. Furthermore, we aim to provide practical systems that can be used by, and thus can contribute to, the scientific community.

Figure 1.1 shows the research methodology applied throughout this dissertation. We began by identifying and designing the required algorithms, which were
Introduction

Figure 1.1. Research methodology

later assembled into proof-of-concept prototypes. We then tested the prototype and collected its performance metrics, which were later analyzed to understand the system’s performance. Finally, we began a new iteration to improve the algorithms and the overall system performance using the previous observations. While the aforementioned methodology applies throughout our entire work, we outline some additional research methods used in this dissertation below.

To analyze the created crowdsourced maps and floor plans, we used computer-aided drawing (CAD) representations of floor plans. We overlaid and aligned the generated maps with the existing CAD floor plans, which enabled us to verify the correctness of the geometry of the produced floor plans and to estimate what percentage of the total area was reconstructed. We employed a laser range finder to measure errors in SfM reconstructions. We measured distances between walls and other static obstacles in situ and compared them to corresponding distances in the generated floor plan. A similar method was used to assess positioning system accuracy, for which we used the laser range finder attached to a smartphone and a set of ground-truth reference points. To investigate the power consumption of AR mobile applications we used a Monsoon power monitor [15]. We physically hooked the device up to a smartphone battery and executed the mobile application in different usage scenarios. Besides quantitative evaluation, we also conducted qualitative analyses. For example, for Publication III, we conducted a qualitative user study and analyzed users’ feedback to evaluate the application’s performance and the user experience. For Publication V and Publication VI, we simulated an edge environment within a public cloud. We ensured that the simulation indicated typical edge behavior, such as having a very low latency between edge nodes and between edge and client applications. The simulations were necessary to test edge deployment scenarios with hundreds
Introduction of simultaneous users in order to thoroughly evaluate the capacity planning algorithms.

1.3 Contributions

The dissertation is a summary of six scientific publications that address the research questions. The contributions of each publication are briefly explained below.

Publication I demonstrates the design, implementation, and evaluation of a crowdsourcing-based mapping and navigation system for smartphones. We demonstrate the feasibility of using crowdsourced photographs, videos, Wi-Fi fingerprints, and inertial sensor data to build complete indoor maps. We use SfM techniques and model partitioning to enable fast image-based indoor positioning and employ constructed 3D models to provide reliable path planning for navigation. We present algorithms to supplement the built models with automatically detected PoIs and floor changing points. The evaluation of the system in a supermarket and a multistory office building proves its competitive performance for indoor mapping and navigation.

Publication II presents the design, implementation, and evaluation of a guided visual crowdsourcing system for efficient indoor mapping. We designed a crowdsourcing task-generation algorithm for collecting visual data with high quality of information (QoI) to minimize the number of photographs required for indoor mapping. The designed system also uses crowdsourced annotations to overcome the limitations of SfM algorithms for reconstructing texture-less surfaces, such as glass walls and panels. Through an extensive evaluation we show that our solution comprehensively reconstructs indoor areas and outperforms conventional opportunistic and participatory crowdsourcing techniques.

Publication III proposes to combine the previously developed image-based positioning with tracking based on inertial sensors to provide continuous and accurate tracking for AR applications. We further investigate the power consumption of AR-based applications and present an adaptive, context-aware tracking algorithm that offloads heavy processing of computer vision algorithms to a cloud and minimizes the number of image-based positioning requests required. We implemented an energy-efficient AR-based indoor navigation application for smartphones and conducted a user study which shows the usefulness of our application and proves that it can enable participants to easily navigate indoors.

Publication IV presents the design, implementation, and evaluation of a low-cost radio-frequency identification (RFID) tags mapping system. The system employs an off-the-shelf RFID-reader-equipped with a smartphone. Using the angle of arrival (AoA) concept, it locates RFID tags attached to inventory items. We used image-based positioning and visual-inertial-odometry-based tracking to provide precise positioning for a reader-equipped smartphone during the mapping stage and to ensure accurate placement of the located tags in an AR
view after the mapping is complete.

Publication V presents a capacity planning framework for hierarchical edge clouds that is oriented toward quality of service (QoS). The framework takes into account network latency constraints and the diverse resource requirements in relation to central processing unit (CPU), graphics processing unit (GPU), and network bandwidth. We develop tools for measuring utilization of the aforementioned resources for server applications and for estimating the required computing capacity before deploying the applications. The framework considers the deployment of compute-demanding real-time systems with a main focus on edge layer capacity planning.

Publication VI proposes a refined algorithm for estimating the required computing capacity for deploying real-time systems on edge clouds. We develop an M/M/k queue-based model and, through extensive evaluation, prove its applicability for estimating required edge node counts with fixed CPU/GPU capacities on each node. We demonstrate the applicability of the developed algorithm by deploying a real-time AR navigation system, while following the obtained capacity plan. Through the evaluation we indicate how our solution minimizes the required number of edge nodes needed to satisfy the specified QoS requirements.

1.4 Structure

This dissertation is structured as follows. Chapter 2 surveys state-of-the-art indoor mapping and localization techniques, reviews visual crowdsourcing systems, and provides relevant background information on AR and computing capacity planning. Chapter 3 summarizes the main contributions of the dissertation. Chapter 4 concludes the work. The original publications are presented after Chapter 4.
2. Background

This section presents the background necessary for understanding the field this dissertation addresses. We first introduce spatial indoor mapping, including indoor localization and navigation methods. We then introduce mobile crowdsourcing and summarize state-of-the-art techniques for crowdsourced mapping. Then, we present AR and its enabling technologies. Finally, we present the background of, and current research related to, edge computing capacity planning.

2.1 Indoor Mapping and Localization

Indoor mapping is an essential step for enabling indoor localization services. Accurate and detailed indoor maps help to enable accurate localization and seamless navigation and to identify PoIs. It is important to note that in order to provide the aforementioned services, the maps should contain additional information besides a typical two-dimensional floor plan image meant only for visualization purposes. For example, a map may contain positions of Bluetooth or UWB beacons if it is to be used for radio-beacon-based localization [79]. It could also be combined with 3D point clouds for vision-based localization [137]. Furthermore, an indoor map should include walkable areas and information on obstacles to aid indoor navigation. In the following subsections, we introduce prominent spatial mapping methods for indoors, including computer vision-based ones, present techniques for indoor localization, and briefly describe state-of-the-art methods for indoor navigation.

2.1.1 Spatial Indoor Mapping

Indoor mapping is a widely researched topic for which numerous techniques have been developed. Pedestrian walking traces were used in previous studies as to construct floor plans [22, 37, 109]. Trace-based floor plan reconstruction uses inertial sensors built into mobile devices, including accelerometers, gyroscopes, magnetometers, and pressure sensors. Readings from accelerometers are used to estimate walking speed and to count steps, while gyroscopes and magnetometers
provide current heading. Pressure sensors are typically used to detect changes in elevation. They can detect floor change points, such as stairs and elevators. The process of tracking a user by utilizing only inertial sensors on-board a mobile device is commonly called *dead-reckoning* [78]. The obtained sensor data is fused together to obtain a continuous walking trace. Typically, the traces are aligned using reference points, such as last-known GNSS positions [94], Wi-Fi access points [109] or Bluetooth beacons [113]. Such walking traces can be used to detect corridors, rooms, and other indoor structures. However, individual walking traces have accuracy problems, due to the measurement integration drift of inertial sensors [82]. Elhamshary et al. [43] have proposed using multiple traces along with trace segment classification and clustering techniques to align the traces and devise accurate indoor floor plans.

In order to improve mapping quality, other studies have developed indoor mapping tools that utilize additional sensors besides inertial ones. *Batmapper* [154] combines traces from inertial sensors with acoustic mapping to infer room boundaries and detect doors. *Walkie-Markie* [124] measures received signal strength from Wi-Fi access points to accurately fuse walking trajectories, and *iFrame* [113] utilizes Bluetooth and Wi-Fi radios to detect open spaces and obstacles. *Kintter* [52] employs camera sensors to capture images and extracts line segments from the images to supplement mapping algorithms. Measurements of ambient magnetic fields were also utilized for indoor mapping [129]. *MaLoc* [135] uses magnetic field fingerprinting to map indoor areas for localization. Other researchers proposed mapping approaches based on landmarks and PoIs. A landmark may be e.g. a payment terminal with a known location [18], or a room or any other indoor structure within a building [121]. Identifying and using landmarks as anchors helps in aligning collected dead-reckoning-based or radio-signal-based traces.

In recent years, **3D point clouds** have been widely used on personal mobile devices for both mapping [86] and localization [93]. A 3D point cloud is a set of points scattered in a three-dimensional space. A point may have color or other information associated with it and is typically independent of other points. Nevertheless, by analyzing groups of points, it is possible to infer contours [66] and consequently rigid 3D structures [46]. This means that we can extract obstacles from point clouds and use them for navigation.

An efficient way of capturing 3D point clouds is to use laser scanners and depth cameras. Such technologies allow accurate measurement of distances between sensing devices and real-world objects. Zhang et al. [146] have presented efficient algorithms to enable accurate mapping with laser light technology. Huang et al. [74] have proposed using Kinect sensor [151] for mapping with an unmanned aerial vehicle, and Endres et al. [44] have designed a 3D mapping system for robots, which relies on depth images. Currently, however, such techniques can only be used with a limited number of devices, since depth sensors are not commonly available on commodity smartphones.

Computer vision-based methods such as SfM have been proposed for obtain-
Figure 2.1. Pipeline for calculating SfM point clouds

Structure from motion is a popular photogrammetry technique that has been widely adopted for constructing 3D point clouds from a set of 2D input images. The output of the SfM process is a sparse 3D point cloud with 3D points generated from visual features extracted from the images. The features are commonly detected on the edges and surfaces of real-world objects. Thus, from the generated models it is easy to infer structure of real-world objects that were visible in the input photographs.

Several tools and libraries, such as VisualSFM [133] and OpenMVG [103] have been developed to implement SfM algorithms. A typical SfM pipeline consists of three steps, as shown in Figure 2.1. Below, we briefly explain each of the steps.

1. **Feature extraction.** An SfM process begins by extracting visual features from input images. Visual features may represent corners, edges [67], or gradient changes [91, 27] in input images. The feature extraction process consists of identifying features within input images (*extraction*) and assigning descriptive vectors to the identified features (*description*). Algorithms for extracting Scale-invariant feature transform (SIFT) features [91] are commonly used in scientific work. As the name suggests, these algorithms identify the corresponding features, regardless of their position, scale, or transformation. With Affine-SIFT [142] invariance was introduced for affine transformation, making
feature detection more robust against significant viewpoint changes. However, SIFT extraction is a compute-intensive process. Therefore, researchers have suggested processing with a GPU [58] or using more efficient feature extractors such as speeded-up robust features (SURF) [27] or Oriented features from accelerated segment test and Rotated Binary robust independent elementary features (ORB) [119].

2. **Feature matching.** The extracted features are compared, i.e. matched across all images. A typical image may contain thousands of features, thus, exhaustive $O(M \cdot N^2)$ matching techniques, where $M$ is the number of images and $N$ is the number of features, are impractical. Cheng et al. [40] have proposed cascade hashing technique to speed up the matching process, and Xu et al. [138] have designed a GPU-based algorithm that matches images several magnitudes faster than CPU-based techniques.

Using a random sample consensus algorithm (RANSAC), a fundamental matrix is computed between the two images with feature matches, to account for errors in the feature extraction and matching stages and to obtain *inlier* matches. The *inlier* matches are then used for 3D point cloud reconstruction.

3. **Reconstruction.** During this step, 3D points and positions and facing directions of the cameras that captured the input images are calculated from the feature matches. A combination of a 3D position and a 3D facing direction is commonly referred to as a 6DoF pose, camera pose, or simply pose. Structure from motion reconstruction is an iterative process. It begins with the computing of points and poses from two initial images, and the point cloud is gradually formed by adding the remaining images one at a time. After an image or a set of images is added to the reconstruction, *bundle adjustment* [143] is performed to minimize reconstruction errors.

When a point cloud is built using SfM, the 3D points are stored within a local coordinate system. Consequently, scaling an SfM model to a geographical coordinate system is an important step before the model can be used for mapping. Gao et al. [51] have used traces collected by inertial sensors and tailored sensing tasks to scale SfM point clouds, and Price et al. [111] have proposed scaling SfM point clouds and estimating ground level by detecting people in the input photographs. Modern SfM tools allow manual scaling of a model with the use of ground control points [5]. A user must either provide 3D coordinates of selected points within a set of images or choose several camera positions and enter their real-world coordinates [10]. The whole model is then adjusted to best fit the defined points.

There are several important advantages of using SfM for crowdsourced indoor mapping. First, SfM works with an arbitrary number of unordered input photographs and is robust against outlier photographs. For example, a photograph capturing a very different view than the views captured by others would not be
Table 2.1. Commonly used indoor localization techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Related research</th>
<th>Requirements and positioning method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure-based</td>
<td>[89] [139] [156] [141] [101] [105]</td>
<td>Pre-installed additional hardware devices (beacons) or labels with known fixed locations. Mobile device obtains its position by detecting the installed beacons/labels.</td>
</tr>
<tr>
<td>Fingerprinting</td>
<td>[18] [84] [73] [68] [92]</td>
<td>A site survey to collect existing fingerprints, e.g., Wi-Fi, Bluetooth, magnetic, and light. Matching detected fingerprints against a database of previously collected ones.</td>
</tr>
<tr>
<td>Dead-reckoning</td>
<td>[149] [41] [78] [147] [55]</td>
<td>An existing floor plan or a site survey to build one. Uses inertial sensors of a mobile device to track user from a known reference point, e.g., last known position obtained from GNSS.</td>
</tr>
<tr>
<td>Vision-based</td>
<td>[93] [136] [54] [87] [80] [99]</td>
<td>Site survey to identify distinctive landmarks or to build 3D maps. Triangulation of matched visual feature points between a query image and the previously-built map.</td>
</tr>
</tbody>
</table>

used for reconstruction. Furthermore, the ability to process unordered images makes SfM compatible with crowdsourced data. Second, SfM-based point clouds contain feature information associated with each 3D point. This helps to enable accurate 6DoF positioning using only the point cloud and a query image that must be localized [93]. Due to the additional information attached to points, in the literature and in this dissertation, SfM point clouds are often referred to as SfM models.

Simultaneous localization and mapping (SLAM) is another technique that works in a similar way to SfM. However, SLAM is tailored for real-time visual input processing. Therefore, it only accepts sequential sets of images and must skip or dramatically simplify the bundle adjustment. Consequently, it achieves lower point cloud accuracy and is typically paired with other inputs, such as inertial sensor readings, to supplement construction of point clouds.

2.1.3 Indoor Localization

In this section, we briefly introduce state-of-the-art approaches for locating a mobile user indoors. Table 2.1 presents various indoor localization techniques which we summarize below.
In order to enable positioning, several researchers have proposed using specialized **infrastructure**, such as visual labels [105, 101], RFID tags [139, 141] or Bluetooth beacons [89] in indoor areas. For example, a user position can be obtained by receiving a signal from a nearby Bluetooth beacon [89]. The location of the user would then be the same as the already known position of the identified beacon. Other approaches use received signal strengths and AoAs of signals from multiple beacons to improve localization accuracy [156]. Montañés et al. [101] have presented a localization system that allows users to scan quick response (QR) codes or near field communication tags to obtain their positions indoors, and Muñoz-Salinas et al. [105] used several visual markers to estimate the position and facing direction of a user. Infrastructure-based approaches demonstrate steady and high localization accuracy. However, these methods are not cost efficient, as dense infrastructure deployments are required to achieve high accuracy.

**Fingerprinting** based techniques rely on various signals already present in indoor environments and do not require the use of additional hardware. To enable positioning, such systems first construct a fingerprint map by taking measurements at positions with known coordinates. In this way, each fingerprint is associated with a unique fixed location. Afterwards, a mobile device can locate itself by capturing a new set of fingerprints and searching for similar ones within the fingerprint database. The prerecorded location of the closest match is assumed to be the current position of the device. For example, regarding Wi-Fi fingerprint-based techniques, researchers have proposed collecting Wi-Fi service set identifiers and measuring the received signal strength (RSS) at multiple known locations within a building to build a fingerprint map [18]. Other researchers have proposed measuring indoor geomagnetic fields [84], visible light intensity patterns [73] and even the electromagnetic field of building’s power network [92] for creating the fingerprint maps. Fingerprinting-based methods mitigate the need for additional hardware and have low processing overheads. However, creating an off-line fingerprint map is labor intensive, since it generally requires extensive site surveys to ensure high accuracy. In order to prevent burdensome site surveys, Pulkkinen et al. [112] have suggested generating synthetic fingerprints based on wireless access point locations and Wi-Fi signal propagation.

**Dead-reckoning** techniques use inertial sensors, such as accelerometers, gyroscopes, and compasses, to track the movement of a mobile device. They typically detect steps, measure step length, and estimate walking direction to obtain positions of users indoors [78]. However, inertial sensors quickly accumulate errors and without proper calibration cannot provide reliable positioning over longer distances. In order to reduce heading direction errors, Zhou et al. [155] designed an accurate and automatic attitude detector A³ that calibrates a phone’s 3D orientation by measuring gravity and compass attitude when the phone is stationary. Shen et al. [125] improved the idea with sensor fusion that uses a 3D North vector obtained from a digital compass as an anchor point. To account
for errors resulting from estimation of linear acceleration, Hemminki et al. [69] have proposed to combine readings from gyroscope and accelerometer to mitigate inertial noise caused by device rotation. Other researchers have suggested using additional equipment for sensor calibration. Chen et al. [39] have observed that Bluetooth beacons can be used to calibrate the walking traces, and Giarré et al. [55] have proposed deploying UWB receivers as calibration points. Zhang et al. [149] have developed CIMLoc, a dead-reckoning mapping and localization system that uses digital maps and particle filtering [70] to circumvent errors during the dead-reckoning. Dead-reckoning based approaches do not require additional infrastructure. However, they quickly accumulate errors and without proper calibration cannot provide reliable positioning over longer distances. Furthermore, in order to quickly obtain an initial location, they must rely on other positioning systems such as infrastructure or fingerprinting-based systems.

**Vision based** localization systems originate from the field of robotics, where they have proved highly beneficial for guiding autonomous robots [53]. Vision-based techniques determine the position and facing direction of a mobile device using a single supplied photograph or a sequence of images captured by the device’s camera. Lee et al. [85] developed an efficient mapping and 6DoF localization algorithm for challenging indoor environments, suitable for autonomous vehicles equipped with a forward-viewing monocular camera. Zhang et al. [150] proposed a localization approach for autonomous robots where a monocular camera is aimed at ceilings. The researchers argued that ceilings undergo fewer changes over time and that the upward view is less obstructed in compare to forward-looking views.

The main benefit of vision-based systems is their ability to immediately calculate 6DoF poses, as opposed to solutions based on radio signals, which typically only provide 3DoF positions. However, vision-based techniques are compute-intensive, cannot be used in completely dark environments, and most of them require initial site survey in order to construct visual landmark databases for retrieval-based techniques or point cloud models for model-based methods. Here, we present infrastructure-free indoor localization solutions for smartphones, which do not require pre-installed fiducial labels but rather use captured views of their current surroundings.

A retrieval-based positioning method captures a picture of the surroundings and compares it to a 2D landmark database that was built beforehand. The algorithm then retrieves images from the database that are visually similar to the query image. Afterwards, it either triangulates the location of the query image using the visual feature matches or directly uses the positions attached to the retrieved images. Gerstweiler et al. [54] used a retrieval-based method to locate a mobile user with the help of a 2D landmarks database consisting of advertisements, company logos, and posters found within the surroundings. Li et al. [87] have proposed capturing a short video for localization and extracting keyframes containing well-observed landmarks. This allows the use of several images to calculate a more accurate pose.
Advances in deep learning and convolutional neural networks (CNN) have helped to improve image retrieval for indoor localization. Kendall et al. [80] have developed PoseNet, a CNN that employs transfer learning [132] and image plane regression to predict 6DoF image poses from a single query image. Their approach achieved a real-time performance of 5 milliseconds to process a query image. Melekhov et al. [99] have proposed using hourglass-shaped networks and analyzing image context to improve 6DoF localization.

Other researchers have proposed building 3D models of indoor environments to enable localization. Xu et al. [136] employed SfM models and image-based localization to extract geometric constraints from crowdsourced photographs in order to reduce the ambiguity that results from Wi-Fi fingerprint-based positioning. Lu et al. [93] combined SfM models with Wi-Fi-based localization by employing Wi-Fi for coarse localization and then using nearby image matching to triangulate a final, fine-grained location. However, high illumination intensity caused by indoor lighting and scarce visual features must be taken into account by the image processing algorithms. Vision-based positioning techniques have proved to be highly suitable for AR applications, since the localization mechanism provides the 6DoF pose required for augmented scene alignment. In this dissertation, we use model-based vision techniques for indoor positioning.

2.1.4 Navigation Methods for Indoors

One of the key uses of indoor maps is to enable indoor navigation services. A navigation service should provide users with instructions about the shortest paths or the paths with the shortest travel times between given source and destination points. Previous work has suggested using grid-based and graph-based methods for indoor path planning [48]. Grid-based systems divide an area into fixed-size cells, with each cell representing a traversable area, obstacle, or PoIs. Path planning within a grid can be lightweight and fast. However, its efficiency and accuracy depend on the size of the grid cells [104]. Graph-based approaches offer higher path planning accuracy, but building an initial navigation graph is a compute-intensive task [140].

Tools and algorithms for mesh-based navigation have recently been developed [9]. These allow accurate and fast path planning within a 3D model, avoiding obstacles and allowing navigation for multiple floors. Mapping based on 3D point clouds is ideal for mesh-based approaches, as point clouds can be converted to 3D meshes using surface reconstruction techniques [28]. In our work, we adopt mesh-based methods to provide navigation services.

2.2 Crowdsourced Indoor Mapping

As discussed in a previous section, multiple techniques have been proposed for indoor mapping, ranging from radio signal to computer vision-based techniques.
Nevertheless, all techniques require site surveys, and the performance of the mapping algorithms usually depends on the amount of data collected. Crowdsourcing [128] is a widely accepted approach for collecting large amounts of data at a low cost.

In this section, we introduce mobile and visual crowdsensing and discuss their advantages and their roles in crowdsourced mapping frameworks. We then review the important work on crowdsourced indoor mapping.

### 2.2.1 Mobile and Visual Crowdsensing

The term “crowdsourcing” was coined from the words “crowd” and “outsourcing” and is defined as a process of obtaining required data or services from a group of individuals, especially via the internet [64]. A similar term “crowdsensing” fits into the general crowdsourcing concept and refers to using individuals, especially their mobile devices, to collectively measure a certain phenomenon [144]. Due to the similarity of the terms, they are often used interchangeably in the research literature. A more recent term “mobile crowdsensing” (MCS), introduced by Ganti et al. [50], refers to collectively sensing data using only mobile devices such as smartphones. A typical smartphone is equipped with a wide range of sensors including inertial, barometric pressure, audio, and imaging sensors. This makes MCS a very popular approach for studying various phenomena [72, 102]. The main advantages of MCS are inexpensive system deployment, a wide range of measurements and much faster data acquisition than data collection with a single device.

A subset of MCS, “visual crowdsensing” (VCS), has attracted a lot of interest in recent years [62]. In VCS, mobile device cameras are the main source of data. The collected data commonly consists of photographs or videos of interesting views and objects taken by crowdsourcing workers. In many cases, readings from other sensors, such as GNSS receivers or accelerometers may be used to supplement the collected visual data, for example, to attach orientation and location information to photographs.

Compared to general MCS, VCS has several important differences. First, over the same period of time, VCS collects data amounts that are several magnitudes larger than datasets collected from inertial sensors or radio signal receivers. For example, a typical smartphone can record a video at a rate of 72Mbps\(^1\). Even though the visual data is more complex to process, it is also more informative and can provide a broader context than other MCS approaches [62]. Second, the process of visual data collection is dependent on position and facing direction and therefore requires more effort to achieve high sensing coverage. One must photograph an object from the perspective of each of its sides to collect complete information about the object [134]. Moreover, depending on the required level of detail, the photographs may need to be taken up close or from far away. In order to achieve higher sensing coverage, previous works have proposed

\(^1\)https://www.gsmarena.com/samsung_galaxy_s9-review-1734p8.php
using users’ mobility patterns [76, 63]. Lastly, ensuring data reliability and redundancy is more important in VCS due to high processing costs and a high probability of collecting redundant/unreliable data. We can identify three main factors that lead to the production of unreliable data in VCS: 1) low quality cameras, that fail to capture detailed photographs or perform poorly in different environments; 2) challenging environments such as dark or intensely lit areas; and 3) undesirable motion, such as quickly moving objects, and instability or shaking of the capturing device. To ensure reliable data collection, Guo et al. [61] have suggested using inertial and light intensity sensors to filter out blurry and incorrectly exposed photographs.

2.2.2 Different Types of Mobile Crowdsensing

Researchers commonly distinguish between two types of MCS: participatory sensing and opportunistic sensing. According to Burke et al., participatory crowdsensing algorithms develop and command sensor networks composed of mobile devices to collect, analyze, and share knowledge among public and professional users [32]. When users take part in a participatory crowdsensing activity, they are given clear instructions regarding the type of data to collect and how to collect it. They may perform the assigned tasks voluntarily or they may be provided with incentives.

With opportunistic crowdsensing, on the other hand, data is collected without any intervention from the user. The main advantages of opportunistic crowdsensing over participatory crowdsensing include significantly lower burdens on users, less requirement for incentives, unbiased data collection and fewer privacy concerns. However, according to Ma et al. [95], the sensing quality of opportunistically collected data is lower and many more participants are needed to achieve sufficient sensing coverage.

In order to combine the benefits of both sensing paradigms, hybrid crowdsensing approaches have recently been introduced [25]. Researchers have proposed collecting initial data opportunistically and then involving the data contributors in participatory sensing activities to improve the quality of the collected data.

2.2.3 Crowdsourced Mapping Techniques

Opportunistic sensor data collection has been widely studied for 2D floor plan reconstruction. A number of sensor-based systems have been proposed for generating indoor maps by using crowdsourced pedestrian walking traces. MapGENIE [109] uses foot-mounted inertial measurement units (IMUs) to collect crowdsourced walking traces and exploits structural building information to build indoor floor plans, while other mapping solutions use only data obtained from smartphone sensors for mapping. PiLoc [94] and ALIMC [153] use traces collected from smartphones and scan for Wi-Fi signals to anchor and segment the collected traces. They also use last-known GPS positions to align aggregated
Background

traces to the global coordinate system.

Several researchers have proposed using RSS from Wi-Fi access points for crowdsourced mapping. Jiang et al. [77] associated rooms within a building with Wi-Fi fingerprints to obtain a room adjacency graph. Afterwards, they could infer a floor plan by combining walking traces collected from hallways. Ahn et al. [18] have shown that it is possible to obtain a radio map by collecting Wi-Fi fingerprints alone without considering walking traces. The researchers utilized payment terminals to reference the collected fingerprints and showed that the system can improve the map over time as the number of collected measurements increase.

Other sensors, such as magnetometers and Bluetooth radios have also been used for opportunistic crowdsourced mapping. By collecting available Bluetooth signals, iFrame improved crowdsourced walking-trace-based mapping. They used Wi-Fi access points and nearby Bluetooth devices as anchor points to calibrate the deviations in dead-reckoning trace collection. Crowdsourced geomagnetic field readings have been used by Zhang et al. [145] in their mapping and navigation tool GROPING. The researchers have suggested that magnetic field observations are more stable than Wi-Fi fingerprints and have proposed building an indoor map using readings only from a magnetometer and a gyroscope. Besides collecting data opportunistically, GROPING also allows users to mark intersection points on a digital map, thus mitigating errors that may be caused by gyroscope-based turn estimation.

With advancements in the quality and availability of mobile device cameras, researchers began investigating the use of VCS data in the form of still photographs and videos for indoor map reconstruction. However, VCS data must be collected using participatory sensing, since a user must point the camera of a device toward the surroundings and ensure that the camera viewfinder is not blocked. Nevertheless, it provides very rich contextual information on the captured environment.

Chen et al. [38] have proposed using keyframes from videos to robustly align crowdsourced walking traces. They designed a mapping system called CrowdMap, which involves a 2-step image matching process. In the first step, the extracted frames are quickly matched using color-indexing histograms [130], shape matching [42] and wavelet decomposition [24]. Since the techniques used in the first step are sensitive to illumination changes and camera parameters, in the second step, the matched results are refined using SURF features [27] for precise matching. With this approach, the researchers achieved a 90% F-measure for reconstructing hallway shapes. Similar to CrowdMap, Travi-Navi [152] compares image histograms to match photographs taken along walking traces, in order to align the traces. Additionally, Travi-Navi provides a guider-follower navigation system once the mapping is done. The system tracks a user who follows a previously recorded trace to reach a desired destination. IndoorCrowd2D [37] uses crowdsourced images to obtain image vector bundles, which are later projected into a grid to infer indoor hallways. It also uses the collected images to
generate indoor panoramas to provide interior views of buildings.

Other researchers have used VCS to detect physical landmarks. Gao et al. [51] have developed an indoor mapping system *Jigsaw*, that employs SFM to reconstruct indoor landmarks from VCS images. *Jigsaw* extracts facade contours from images and reconstructed 3D point clouds. It asks users to execute tasks such as making a turn and capturing an image, or taking a walk and capturing another image. This enables SFM models to be scaled to the real-world coordinate system. To improve mapping speed and make the system more robust, Gao et al. [52] have proposed angle and stride-length calibration of dead-reckoning and single-image localization-based landmark detection.

Several researchers have also focused on the efficiency of the mapping process. Reliable task assignment for spatial crowdsourcing workers was studied by Zhang et al. [148]. Researchers have suggested estimating the confidence levels of the workers in order to predict successful completion of the assigned tasks, thereby minimizing the number of task assignments required. Zheng et al. [152] have suggested that blurry photographs negatively impact VCS data collection by introducing useless data. Therefore, the researchers proposed using inertial sensors to predict the quality of the collected images and the optimal moment for performing the image capture.

### 2.3 Enabling Technologies for Augmented Reality

AR is defined as an interactive system that combines and registers real and virtual objects in real time [98]. In other words, AR enhances the real world with artificial objects that are virtually indistinguishable from physical ones. To use AR, one must have an AR-capable device such as a heads-up display [36], smart glasses [116] or a handheld device, e.g., a smartphone.

In recent years head-mounted AR devices developed by companies such as Microsoft [8], Vuzix [12] and MagicLeap [7] have gained significant popularity. Such devices provide a hands-free AR experience through see-through displays, which has made them popular in industry. However, AR tools such as smart glasses are not yet widely used. Smartphones are currently the most widely used devices for AR. Numerous studies have been conducted on AR smartphone applications [75] in various fields, including education [19], entertainment [117] and industry [47].

In general, AR applications can be divided into two categories: location-independent and location-aware. Location-independent applications create AR experiences that do not depend on a user’s location, including applications for displaying virtual content [6] and certain AR games [60]. Location-aware applications use spatial awareness to show content that is relevant for a user. A typical example is a facility maintenance application that displays nearby smoke detectors and previous inspection notes to maintenance personnel [81].

In order to provide AR experiences, a mobile AR application should generally
support the following functionalities: it should capture video and display it on screen, render natural-looking artificial objects, seamlessly align the rendered objects onto the video feed in real time, and allow interaction with the augmented objects. A common smartphone can easily capture high definition video to display on its screen. It is also typically equipped with a GPU, capable of rendering realistic 3D artifacts, with which a user can interact via a touchscreen. However, aligning the real and augmented worlds is still the most difficult challenge for enabling AR [98].

In order to align the two scenes, a mobile device must identify its position and orientation, also known as a 6DoF pose, with respect to the real world. For example, considering the AR maintenance application, a mobile device must be able to estimate its 6DoF pose with an accuracy of less than a meter and with a facing direction error of less than 5 degrees. Otherwise, maintenance annotations in AR would not be positioned on top of the actual equipment that needs maintenance. Furthermore, the device must continuously update the pose estimation to account for device movement so that the AR annotations stay aligned to the real world objects. The process of continuously updating the device’s pose is commonly referred to as tracking. A typical tracking accuracy should be equal or better than the accuracy of initial 6DoF localization. Tracking differs from the localization techniques presented in Section 2.1.3 insofar as it must calculate relative pose updates with very high accuracy in real time. This is challenging for computing-power-constrained and energy-constrained mobile devices. In the next sections, we summarize popular and state-of-the-art tracking techniques for AR and introduce contemporary solutions for energy-efficient AR.

2.3.1 Tracking Methods for AR

Pose estimation was identified as the main challenge facing AR researchers in the very first surveys on AR technologies [26]. Since then, various techniques have been proposed for aligning real-world and artificial scenes [30]. Early AR applications used GNSS to obtain locations and device compasses to estimate orientations of mobile devices [83]. While this enabled quick estimation of a device’s pose and provided continuous tracking, the accuracy of the method was only sufficient for displaying AR labels on large objects such as statues and buildings. Moreover, the tracking method could not be applied indoors due to the lack of a GNSS signal.

To mitigate accuracy problems and to provide robust 6DoF tracking for indoor environments, Newcombe et al. have proposed using depth sensors, such as Kinect, which can estimate the distance to nearby objects and construct depth images at a high frame rate [108]. They used a fast projective data association algorithm [29] to estimate the relative pose of a device from a stream of depth images. The method achieves high tracking accuracy and works in low lighting conditions. However, it is not suitable for commodity smartphones that contain monocular cameras.
Computer vision-based algorithms have become popular for tracking with smartphones equipped with monocular cameras [98]. Several researchers have proposed using fiducial markers attached within the environment [114, 118]. A fiducial marker is an easily distinguishable label placed in an area. Such markers include signs, QR codes, pattern-rich textures, and images of distinctive objects and sceneries. Whenever a marker is observed by a mobile device’s camera, the device recognizes the marker and calculates its relative pose with respect to the marker. It then renders augmented content on top, or very close to, the marker. The approach provides high accuracy and reliable tracking as long as the marker is seen. Several free and commercial toolkits, such as ARToolKit [3] and Vuforia [11], have been developed for marker-based AR [23].

Other researchers have proposed eliminating the need for markers by developing marker-less tracking methods. Visual SLAM [49] has become popular for marker-less vision-based tracking. It originates from mobile robot navigation systems, whereby a robot maps the environment and at the same time positions itself within the map using a monocular camera. The SLAM algorithm processes a continuous stream of camera images and, by detecting changes in pixel intensity (optical flow) or the locations of detected visual features, estimates the 6DoF pose of the camera. In contrast to SfM, Visual SLAM operates in real time, as it either skips the bundle adjustment step or only performs partial adjustment.

Schops et al. [122, 45] have proposed a SLAM tracking algorithm that uses optical changes between pixels of two subsequent color images captured in real time. By detecting changes in pixel intensity values, the researchers could estimate the pose of a mobile device and a semi-dense depth map of the surroundings. Other types of Visual SLAM methods are based on detecting natural features such as corners, edges, interest points, or regions [49]. Mur-Artal et al. [106] developed a visual SLAM method that can use any object or piece of scenery for pose alignment provided it has enough visual features. The algorithm extracts ORB features [119] from the keyframes, matches them across subsequent frames, and triangulates the matched points to obtain a 6DoF pose. To ensure the robustness of the tracking algorithm, the researchers introduced real-time loop closure based on pose graph optimization and a relocalization mechanism to quickly account for tracking failures.

Texture-less environments and fast camera movement can significantly degrade the accuracy of Visual SLAM tracking. Gerstweiler et al. [54] have proposed using 2D visual markers and tracking based on inertial sensors to supplement Visual SLAM tracking. The 2D markers are used as anchor points and the IMU-based tracking ensures seamless switching between SLAM maps. While IMU sensors are useful for developing tightly coupled visual-inertial tracking systems [107], the researchers have proposed initializing the system with a few seconds of purely vision-based tracking to determine biases of the gyroscope and accelerometer before initiating inertial-sensor-based tracking. Afterwards, the IMU sensors are used to estimate the scale of the scene and the movement velocity of the camera, which helps to prevent drifts in visual SLAM.
pose estimation and aids loop closure. Visual-inertial SLAM tracking has been adopted for use in several commercial AR software development kits, including Google’s ARCore [1] and Apple’s ARKit [2].

2.3.2 Energy Efficient AR

Energy efficiency is another important challenge in relation to AR, which has been highlighted in recent reports [88, 123]. Due to continuous usage of a mobile device’s camera and heavy image processing algorithms, the device’s power consumption increases notably when using AR apps.

Computation offloading has been suggested as a viable option for increasing the energy efficiency of AR applications [20]. Researchers have proposed offloading parts of AR application, such as tracking and mapping, to an edge server and have introduced collaborative processing to share processing results with multiple AR clients, thus minimizing the device’s power consumption. Other researchers have proposed optimizing image processing algorithms to minimize computation costs. Rublee et al. [119] introduced an efficient visual feature detector and descriptor for object detection and patch tracking, while Sattler et al. [120] have proposed an efficient vocabulary-based visual feature-matching algorithm.

2.4 Capacity Planning for Real-Time Systems at the Edge

Edge clouds are small data centers or computing nodes co-located with wireless network access points. In contrast to public clouds, edge nodes and services deployed at the edge are reachable within a single network hop, which ensures minimum network latency for service users. The low latency is a necessary requirement for real-time applications, such as AR applications that retrieve 6DoF positions computed on the server side. However, the scalability of edge computing resources is very limited, which presents unique challenges for deploying services on edge clouds. Consequently, not only is intelligent task offloading and scheduling required but careful initial required computing capacity planning must be conducted before deploying the edge cloud infrastructure. It is important to deploy enough resources so that running services satisfy QoS requirements. At the same time, edge cloud utilization should be maximized to prevent costs related to under-provisioned hardware.

Previous works on capacity planning have proposed capacity planning solutions based on benchmarking or analytical methods. Gonçalves et al. [57] have proposed employing benchmarking methods to estimate the required capacity. The required workload is first deployed on an initial capacity environment. After the workload finishes, depending on whether or not the required QoS has been met, the cloud capacity is decreased or increased until an optimal configuration is obtained. Advantages of benchmarking based capacity estimation techniques
include applicability to the capacity planning of multiple resources and accurate capacity plans regarding real-world infrastructure. However, they require executing extensive simulations every time when either an infrastructure or an application changes.

In order to prevent simulations, heuristics based and analytical capacity planning methods were proposed. In their study of business-driven capacity planning, Candeia et al. [33] used heuristics based on instance utilization and queue networks [21] to develop capacity planning solutions that would increase the profits of software-as-a-service providers. Carvalho et al. [35] have used analytical models for capacity planning. The researchers have proposed using queuing theory [157] to estimate the computing capacity required to meet defined service-level agreements.

Most analytical approaches considered capacity planning for CPU. However, contemporary server applications not only require CPU but also GPU, memory, and network resources. Carvalho et al. [34] have considered CPU and memory requirements for capacity planning, and Song et al. [126] have added network utilization. Brunnert et al. [31] have suggested specifying application resource usage profiles before deploying cloud applications. While the profiles help to identify which resources the application requires, this imposes additional burdens on application developers.

2.5 Summary

We have reviewed the state-of-the-art research on indoor mapping, positioning, tracking, and capacity planning. So far, a considerable amount of research has been done on indoor mapping and positioning, with smartphones being the most commonly used devices for these purposes. Although numerous positioning systems indicate promising performance, in this work we focus on vision-based mapping and positioning techniques, as they integrate perfectly with AR use cases. Nevertheless, a fusion of positioning techniques, especially combining positioning and tracking methods, is essential for providing a seamless AR experience. Thus, non-vision-based methods, such as dead-reckoning are still highly relevant in this research field. Consequently, we have also introduced contemporary work on pose estimation for AR, which is the main challenge for enabling any kind of AR applications.

In the next chapter, we present how the introduced contemporary work can be applied, and how we supplement it with novel algorithms to address the challenges that remain for enabling efficient indoor mapping and ubiquitous indoor AR.
3. Enabling Ubiquitous Augmented Reality with Crowdsourced Indoor Mapping and Localization

In this chapter, we summarize the contributions of each publication and show how they address the research questions introduced in Chapter 1. Table 3.1 illustrates the research questions addressed in each publication and briefly summarizes the research results. In Section 3.1, we address RQ1 by elaborating on the design and implementation of systems that solve the challenges of efficient crowdsourced mapping and accurate indoor positioning. In Section 3.2, we focus on using the developed systems to provide seamless AR experiences to address RQ2. Section 3.3 describes the challenges of hierarchical edge cloud capacity planning and addresses RQ3 by presenting an edge capacity planning framework used for deploying the developed AR systems. We end this chapter with a discussion on future work.

3.1 Crowdsourced Mapping

Indoor maps are highly beneficial for localization and wayfinding in indoor environments such as airports, shopping malls and office buildings. The maps allow users to easily find and navigate to places of interest. More importantly, combined with 6DoF-positioning systems, fine-grained indoor maps are key for enabling ubiquitous AR experiences. However, despite the numerous mapping techniques discussed in Chapter 2, the availability of fine-grained indoor maps is still low. There are several reasons for such slow progress in indoor mapping. To provide a high-quality user experience, the maps have to be extremely detailed and must include obstacles, walking paths, and all important PoIs. Additionally, the maps must be kept up to date and modifications must be easy to incorporate when indoor environments change. Finally, indoor mapping must be cheap and efficient and preferably conducted using crowdsourced data. This has motivated us to develop an efficient vision-based indoor mapping and navigation system that uses crowdsourced data collected by smartphones for mapping and provides positioning and navigation services to mobile clients.

Our solution to RQ1 consists of two parts. First, in Section 3.1.1, we explore methods and tools for enabling crowdsourced indoor mapping and localization.
In Section 3.1.2, we investigate how crowdsourced data for indoor mapping can be collected efficiently and how crowdsourcing can be used to address the challenge of reconstructing texture-less surfaces.

### 3.1.1 Indoor Mapping from Crowdsourced Data

Previous works [51, 38, 152] have demonstrated the feasibility of using VCS and computer vision techniques, in particular SfM, to build crowdsourced indoor maps and to enable image-based positioning. As presented in Section 2.1.2, SfM techniques are highly suitable for crowdsourced mapping. However certain key challenges must be solved before SfM can be used to map large areas. First, reliable map-updating mechanisms are necessary to keep maps up to date, as indoor environments tend to change over time. For example, furniture is rearranged inside office buildings, exhibitions change in museums and new stores are opened in shopping malls. Secondly, mapping of texture-less areas that include reflective surfaces and plain-colored walls is extremely challenging for current SfM techniques, which can result in incomplete and disconnected indoor models. Moreover, approaches that combine multiple SfM models, which are required for multi-floor buildings, have not been widely explored. Lastly, there is a significant need for methods to automatically detect and add PoIs to the maps.
In Publication I, we tackle the above-mentioned challenges by presenting the design, implementation, and evaluation of ViNav, a mapping and navigation system based on crowdsourcing. ViNav builds 3D models of areas of interest and keeps them up to date. The system improves the accuracy and completeness of map reconstructions using walking traces and enables mapping in multistory buildings. In addition, ViNav enhances maps by detecting and adding PoIs and enables fast and accurate image-based positioning and navigation services within the mapped area. Below, we present in detail each contribution of this Publication.

The first contribution of Publication I is the building of accurate and up-to-date indoor maps from crowdsourced visual and inertial sensor data. We proposed the map construction process that uses crowdsourced image data, such as photographs and videos, to build SfM models represented as 3D point clouds. To ensure the accuracy of the models over time, we designed an algorithm to update the point clouds when the environment changes. The algorithm expands the SfM models with new input photographs and uses information from the cameras’ fields of view and ray tracing algorithms [56] to remove obsolete points from the 3D point clouds. To obtain visual maps and enable navigation, we built 2D maps and 3D meshes by projecting the 3D point clouds onto a ground plane and extracting obstacles represented by clusters of points. We also used an A* (A star) path-finding algorithm to provide navigation within meshes. However, using only SfM tools for indoor mapping presents the following challenges: 1) SfM algorithms cannot reconstruct featureless or reflective surfaces such as glass walls or mirrors, and any reflections may introduce unwanted artifacts in the 3D point clouds; 2) Most staircases and elevators are very challenging and in most cases impossible to reconstruct with SfM techniques, which means that separate and disconnected SfM models must be built for each floor of a building. ViNav solves this challenge by collecting walking traces obtained from accelerometer and gyroscope readings and adding the collected traces to the SfM models. The system uses images taken along the traces to calibrate them and to prevent an accumulation of errors from inertial sensors. We used the collected traces to remove any artifacts caused by reflections along the walking paths. Furthermore, we collected pressure sensor readings along the traces to identify floor interchange points, which allowed us to align the SfM models and provide multi-floor wayfinding and navigation.

As a second contribution, we present techniques to enrich the generated maps with automatically detected PoIs. We used optical character recognition techniques [4] to extract text, such as room numbers and discount product names, from images and added the extracted pieces of text to the models as PoIs, based on nearby feature matching. This enables users to immediately search and navigate to PoIs once the mapping process is complete.

In addition to generating maps, state-of-the-art mapping systems also provide positioning services. The third contribution of this work is an infrastructure-less indoor localization system. ViNav uses SfM models to enable accurate
image-based positioning. In addition to 3D points, an SfM model includes the positions and facing directions of the cameras that took the input photographs. Whenever a new image is added to the model, we can retrieve the 6DoF pose that corresponds to the position and facing direction of the device that took the photograph. Thus, we can obtain accurate positioning of the person who captured the photograph within the area.

One of the main challenges of image-based localization is the fact that SfM algorithms are difficult to scale and involve heavy processing, especially when they are applied to large datasets. To tackle this challenge, we use model partitioning and image selection for feature matching. We use Wi-Fi fingerprints collected along walking traces to perform coarse fingerprint-based localization before choosing nearby partitions and images for SfM-based localization. This technique dramatically reduces the time needed to calculate image-based positions and enables seamless AR experiences in large spaces. We describe ways of enabling seamless AR further in Section 3.2.

Through extensive evaluation, we demonstrate that ViNav is suitable for building-scale mapping and positioning and that it achieves competitive performance compared to the state-of-the-art. We evaluate ViNav in two different environments: a three-story office building and a supermarket. The sum of the total area of the evaluation was roughly 4900 m², including a library, corridors, a restaurant and lobby areas, and supermarket aisles. We show that maps created from crowdsourced inputs are accurate and suitable for indoor way finding. We achieved 0.78 meters error in mapping obstacles in the office building and 0.68 meters overall distance error in the outline of the supermarket building. ViNav located all floor interchange points in the office building, including stairs and an elevator, with an average distance error of 2.13 meters. Furthermore, ViNav localized a user indoors within 2 seconds and demonstrated accurate localization with a position error of 2 meters in the office building and 1 meter in the supermarket. The facing direction error in both cases was less than 6 degrees.

### 3.1.2 Efficient Mapping with Visual Crowdsourcing

The previous subsection describes how crowdsourced visual and sensor data can be used for indoor mapping. However, when it is used for mapping, VCS introduces several new challenges.

First, as discussed in Section 2.2.1, the data collection process in VCS is dependent on sensing position and sensing direction. Hence, a VCS participant must collect visual data from all sides of an object to achieve full coverage of the object. Consequently, the sensing task becomes highly laborious and requires considerable involvement of participants. In indoors scenarios, VCS participants must capture photographs inside all the rooms and photograph nearby objects from all visible sides. While taking videos helps to achieve the required coverage faster, the participant must still memorize where data has already been collected and plan collection paths in advance. Second, assuming that the VCS data is
Enabling Ubiquitous Augmented Reality with Crowdsourced Indoor Mapping and Localization

to be used for SfM-based indoor map reconstruction, the amount of collected data is important. Too little data will result in incomplete SfM models and deficient maps, whereas data redundancy will significantly increase the model construction time as SfM algorithms demonstrate exponential execution time. Therefore, to prevent semantic redundancy, the data collected should have high QoI. We use QoI to refer to the amount of useful information that the dataset contributes to map reconstruction in terms of area coverage. The last challenge is the fundamental limitations of SfM algorithms, which is that they cannot reconstruct featureless surfaces, such as glass panels, monochromatic textureless walls, or highly reflective objects. Therefore, the built maps may be lacking certain structural information, which reduces the completeness of the maps and affects navigation path-finding performance.

In Publication II, we solve the aforementioned challenges by designing and implementing an efficient indoor mapping system, which we call SnapTask. SnapTask helps VCS participants to collect VCS data with high QoI and achieves efficient mapping by minimizing the number of photographs needed to fully cover a particular area. It ensures completeness of the built models by using crowdsourced annotations of featureless surfaces in order to overcome the limitations of SfM in reconstructing such objects.

Several previous researchers have focused on collecting VCS data with high QoI and have proposed using readings from inertial sensors to predict the quality of collected images [61, 152]. While this helps to remove blurry and wrongly exposed photographs from input data, it does not improve area coverage. Other researchers have proposed measuring the experience levels of crowdsourcing participants before assigning them VCS tasks [148, 71]. However, estimating participants’ skills requires additional effort and does not guarantee high QoI data. In contrast to the previous work, SnapTask employs a guided participatory crowdsourcing paradigm, whereby it generates locations for the next photograph collection tasks in such a way that photographs taken at the generated locations notably increase the area coverage with the minimum number of collected photographs.

Our system uses SfM models built using the techniques described in Section 3.1.1 as input and iteratively constructs an SfM model that fully covers the whole area. The iteration begins once a set of photographs is added to a model. SnapTask extracts walkable and non-walkable areas (obstacles) from the model and computes the fields of view of the cameras that were used to collect the input photographs. Using a flood-fill-based algorithm, our system finds an empty (neither covered by obstacles nor seen by any camera views) area where new photographs must be taken to efficiently increase the area coverage. It then sends the location to a mobile application used by a VCS participant. When the participant arrives at the location, the application instructs them in how to collect a set of images in situ. When the collected images are uploaded to the server, an SfM model with increased coverage is built and a new iteration begins. The process continues until the system determines that the outer bounds of the
area are fully reconstructed and the area within the bounds is fully covered by either obstacles or walkable areas.

Whenever a set of images fails to contribute toward increasing the area coverage, SnapTask generates a different kind of crowdsourcing task known as an annotation task. The annotation tasks ask participants to take photographs of featureless or reflective surfaces such as glass walls and mirrors. The photographs are then uploaded to an online annotation tool, which we ask other participants to use to annotate the featureless objects. We cluster the annotations to obtain reliable boundaries of featureless surfaces and project unique feature-full textures onto annotated surfaces in the photographs. The photographs are then processed with the SfM pipeline and the projected textures result in 3D points that represent featureless objects in the SfM model. In this way, obstacles that cannot be reconstructed by conventional SfM techniques can still be added to the SfM model and eventually to the map.

In Publication II, we demonstrate that SnapTask can be used for efficient mapping of indoor venues with crowdsourced data. We evaluated our solution in a university library by comparing its performance with conventional opportunistic and participatory crowdsourcing techniques. For that, we collected three sets of images. The first set was collected using conventional opportunistic crowdsourcing techniques, meaning that the images were captured automatically by a mobile device without intervention from a VCS participant. The second set was collected through participatory crowdsourcing, where participants were asked to capture a number of arbitrary photographs in an area. The third set was obtained using SnapTask to collect images. We used each set to reconstruct maps of the library and compared the reconstructed outer walls of the library (area bounds), the total coverage of traversable areas and obstacles, and the number of photographs needed for the reconstructions. The evaluation shows that SnapTask fully reconstructed the outer walls of the venue and achieved 98.14% in total area reconstruction. Furthermore, in terms of area coverage, it outperformed participatory and opportunistic crowdsourcing approaches by 20.72% and 34.45%, respectively, using the same amount of input images. The evaluation shows that SnapTask enables efficient mapping of indoor venues and can greatly improve the efficiency of mapping systems such as ViNav.

### 3.2 Pose Tracking for Seamless AR

The previous section describes methods for efficient indoor mapping and accurate 6DoF image-based localization, which form a basis for enabling indoor AR applications. Another key part of the AR system is a mobile application that provides the AR experience to end users. However, several challenges have to be solved in order to develop accurate and efficient mobile AR applications. First, the mobile application must continuously and accurately track the movement of a phone with respect to the real world. Otherwise, augmented objects would
appear misplaced, and their locations would seem to fluctuate, which would result in a poor user experience. While we can use the previously developed image-based localization to obtain an accurate initial pose of a device, the same technique is not applicable for continuous tracking. This is because the localization request takes around one second to process, while a typical human reaction time is several magnitudes faster [16]. AR libraries such as ARKit and ARCore provide real-time vision based SLAM tracking for mobile devices. However, such SLAM algorithms impose high power consumption for energy-constrained smartphones. Even without visual SLAM tracking, AR applications are known to use excessive amounts of power [13], thus, the second challenge is to develop energy-efficient tracking algorithms, in order to allow use of AR applications over long periods of time.

In this section, we focus on RQ2. We propose using the developed mapping and localization system to implement mobile AR applications, including AR-based navigation and AR-based inventory management. In Section 3.2.1 we solve the challenge of providing an accurate yet energy-efficient mobile AR indoor navigation application, while in Section 3.2.2 we focus on a scenario in which high accuracy takes precedence over energy efficiency.

3.2.1 Seamless and Energy-Efficient AR

Accurate and continuous tracking of a device's pose is an essential component of seamless AR experiences. However, localization based solely on servers is not sufficient for continuous tracking, due to server-imposed latency and the necessity to update the pose in real time. Previous works have proposed fusing SfM-based positioning with SLAM techniques [131] or performing image-based pose calculation on the mobile device [54]. However, such approaches impose heavy processing on the client side and are not suitable for supporting large maps due to device memory constraints. To mitigate the aforementioned problems, in Publication III we describe how we designed and implemented a continuous real-time 6DoF pose tracking algorithm by fusing the image-based localization developed in Section 3.1, which ensures localization within large maps, with pose tracking based on local inertial sensors, which helps to make tracking energy-efficient. We used accelerometer and gyroscope readings, as well as a step-detection algorithm, to calculate the poses of a mobile device between two subsequent image-based localization responses. In this way, our algorithm ensures that AR objects remain stable with respect to the real world when the mobile device moves. Furthermore, we used inertial sensors for orientation detection to reduce the orientation errors of image-based localization, since a smartphone can very accurately estimate its orientation with respect to Earth's gravity. We show that the estimation is more accurate than image-based orientation alone.

In Publication III we further analyze the energy consumption of AR apps and demonstrate that state-of-the-art AR applications consume as much as three
times more energy than a typical interactive non-AR application, such as Google Maps. Even though the largest amount of energy is consumed by a camera sensor, compute-intensive image processing algorithms also heavily contribute to energy consumption. Using inertial sensors for tracking and offloading image processing to a server helps to minimize mobile device energy consumption. However, tracking with inertial sensors leads to an accumulation of errors over time. Thus, image-based positioning must be frequently conducted to mitigate the errors and recalibrate the sensors. However, computation offloading involves energy consumption for wireless data transfers. Thus, in order to lower the energy consumption imposed by computation offloading while maintaining high tracking accuracy, in Publication III we design a context-aware tracking algorithm that minimizes the number of localization requests.

We consider a scenario in which a mobile application is used to navigate indoors from a particular position to a selected PoI. The navigation instructions are shown in real time as AR objects, and the user can reach the desired destination by following the instructions. In a such scenario, when a user is stationary or moves along a straight line, the sensor tracking errors accumulate slowly. Therefore, we designed our algorithm to perform image-based localization after a user makes a significant turn $\delta$ or walks a certain number of steps $T$. Our evaluation shows that, in our test environment, optimal values are $\delta \geq 40$ degrees and initial $T = 20$ steps. With these values, the algorithm helps to significantly reduce the number of image-based localization requests.

In order to evaluate the accuracy of our tracking algorithm and the user experience of the AR mobile applications that use the algorithm, we developed an AR-based indoor navigation application for smartphones. We evaluated the application in an office building belonging to the Department of Computer Science at Aalto University, where we conducted a user study. We asked 17 participants to use the application. Each participant had to use the application to locate her or himself in the building, to search for a desired destination, to plan the navigation path, and, using the application in AR mode, to follow an augmented arrow to arrive to the planned destination. During the study, every participant managed to successfully reach their desired destinations within the building. Eighty-eight percent of the respondents found that the application was sufficiently accurate and 82% of respondents found it easy to use. In addition, all respondents agreed that AR is useful or very useful for this type of application, with 88% stating that following the augmented arrow was helpful for finding the destination. The study revealed two important conclusions: 1) the fusion of image-based localization and tracking based on inertial sensors provides sufficiently accurate tracking for AR-based pedestrian navigation services; and 2) AR applications are found to be useful and highly welcomed by users. Furthermore, we observed the energy consumption of the application and discovered a more than 20% reduction in total energy usage when using the context-aware tracking algorithm. In conclusion, we show that our algorithm can achieve high tracking accuracy and reduced energy
consumption.

### 3.2.2 High-Accuracy Tracking with SLAM

The previous section describes an energy-efficient tracking solution for AR-based pedestrian navigation, which combines image-based localization and inertial-sensor-based continuous tracking algorithms. However, for certain types of AR applications, the accuracy of the tracking algorithm takes precedence over energy efficiency. In Publication IV, we design and implement a low-cost solution for mapping and locating ultra-high-frequency RFID tags attached to items on shelves. Most previously proposed RFID mapping solutions require readers with multiple antennas [100] or fixed-position readers [115, 96]. Our solution uses a commercial single-antenna off-the-shelf RFID reader attached to a smartphone to scan for nearby RFID tags. We locate the positions of the tags based on the AoA concept, using the positions and facing directions of the reader-equipped smartphone as the AoA of the RFID tags. Therefore, the accuracy with which the proposed system can map inventory items depends significantly on how accurately the device’s pose can be calculated.

To achieve the necessary high localization and tracking accuracy within a venue, in Publication IV we explore the possibility of combining the image-based localization presented in Section 3.1 with readings from both the inertial and visual sensors of a smartphone. As discussed in Section 2.3.1, previous research has demonstrated the feasibility of using a smartphone’s camera feed for Visual SLAM [131, 54]. In addition, AR tools such as ARKit and ARCore further improved the pose tracking accuracy of a device by utilizing inertial sensors together with Visual SLAM. In our work, we combine visual and inertial tracking using ARCore with image-based localization to achieve high-accuracy 6DoF localization and pose tracking of a reader-equipped smartphone.

We developed an inventory scanning mobile application that uses the aforementioned localization and tracking method to implement the RFID tag localization algorithm. When a user walks around the scattered RFID tags, the smartphone continuously scans for RFID tag identifiers. When a tag is found, its identity number and the current position and facing direction of the smartphone (reader) are stored in a database. Each record is stored as a vector in a 3D space, which represents the direction from where the RFID response comes. The directions are then used to compute the positions of discovered RFID tags. The algorithm takes all the directions that belong to the same tag identifier and computes the centroid of their intersections. The centroid represents the 3D position of the discovered RFID tag, which is also stored in a database. However, accurate estimates of the centroid require the RFID tags to be discovered from multiple angles, which may not be practical in real-world scenarios. Moreover, the phone may be facing an RFID tag with a direction angle offset when the AoA is calculated, which leads to inaccuracies of the estimation of the centroid. In order to improve the tag localization accuracy we developed a bias removal algorithm,
based on reference RFID tags with known positions. After the scanning is completed other users can use the application to see the discovered RFID tags in an AR view.

We evaluated the proposed RFID mapping solution in a library where we attached 75 RFID tags to books in the shelves. First, we enabled image-based localization using the mapping system described in Section 3.1.1. Then, we evaluated the image-based localization and ARCore tracking accuracy within the library premises. The image-based localization accuracy indicated a 1 meter position error and a 6 degree facing direction error. The ARCore based tracking demonstrated a mean position error of 0.28m (std = 0.15) and a facing direction error of 2.84 degrees (std = 0.83). This indicates that we can obtain the position of the RFID reader with less than 1.3 meter error and less than 9 degrees facing direction error. After applying the bias removal algorithm, we show that with such tracking accuracy the proposed RFID mapping algorithm can correctly locate 84% of RFID tags placed within shelf cells of size 87cm x 38cm x 26cm. The results indicate that our solution performs competitively against other state-of-the-art solutions while eliminating the need for additional hardware.

3.3 Capacity Planning for Real-Time Compute-Intensive Systems

In Sections 3.1 and 3.2, we developed an AR system based on client-server architecture. However, in order to make the system available for wide adoption, we need to ensure that system QoS requirements remain satisfied with numerous users simultaneously interacting with the real time AR system. Clearly, a single server system is not sufficient to support many users. A public cloud environment, which offers easily accessible vast amounts of computing resources, is commonly considered for deploying client-server applications. While public cloud providers offer a cheap, readily available and scalable approach for deploying server applications, their centralized architecture imposes high network latency. This poses a major challenge for real-time systems, such as AR navigation, that provide latency critical services to end users. To mitigate latency issues, researchers have proposed deploying such services at the edge of the Internet, close to service users [127]. As discussed in Section 2.4, edge capacity planning is still challenging, and careful planning must be performed to ensure high QoS of the deployed edge services. In this work, we define QoS in terms of server response delays. While numerous research efforts have been made in developing server task scheduling [90] and computation offloading mechanisms [97], little has been done to provide reliable estimates of the initial edge cloud computing capacity needed to satisfy QoS constraints.

Motivated by the need for practical tools and methods for edge capacity planning, in this section we tackle the challenge outlined in RQ3. In Publication V, we propose a novel hierarchical cloud capacity planning framework, while in Publication VI, we present an improved, more general edge node count estima-
Figure 3.1. Workflow of the capacity planning framework

The aim of our proposed capacity planning framework is to provide accurate estimates of initial hierarchical edge cloud capacity that satisfy the required QoS. Current research on cloud capacity planning has developed methods that focus on CPU capacity estimation [35, 57]. By contrast, our framework considers diverse resource requirements including CPU, GPU, and network utilization. However, having diverse resource requirements poses a challenge for accurately determining resource usage levels for a given workload. Brunert et al. [31] have proposed delivering resource usage profiles while developing the cloud applications to indicate correct usage levels. To prevent the burden of specifying resource usage levels, our framework use resource profiling to determine the resource utilization levels for different server tasks.

The design principles of the framework are based on three key assumptions
that we formulate and prove in Publication V. First, we showed that edge nodes are more suitable than public clouds for executing latency sensitive tasks. Second, we prove that smart combinations of service requests yield better resource utilization and minimize the number of required processing nodes. For example, processing of GPU-intensive requests can be executed on a node that is concurrently processing CPU-intensive requests without affecting the completion times for either kind of request and improving the overall node utilization. Third, we showed that several low-capacity computing nodes outperform a single high-capacity node, even when the sum of processing capacity of the low-end nodes is equal to the processing capacity of the high-end node. This information helps in selecting appropriate performance machines to deploy as edge nodes.

According to the aforementioned principles, we designed a framework consisting of six steps (see Figure 3.1). The framework takes as an input an already-developed system that consists of a set of microservices implemented as Docker\(^1\) containers and QoS requirements expressed in terms of maximum allowed response delays. It uses automated profiling (step 2) and task category classification (step 3) to identify which microservices should be deployed on public or edge clouds. Step 4 identifies the microservices that should be placed on the same computing nodes to maximize resource utilization. We determine the required edge node counts in step 5, before testing the entire planned deployment in step 6. The final capacity plan contains results from every step of the framework and can be used to accurately estimate the capacity required for a hierarchical cloud deployment.

Execution of the framework requires the use of the three key tools that we develop in this work. We developed the profiler tool (1) for gathering resource usage patterns during step 2 and the benchmarking tool (2) for determining the maximum number of simultaneous service users supported by a particular system deployment in steps 5-6. In Publication VI, we designed and developed a node count estimation tool (3), which is used in step 5. This tool calculates the minimum number of edge nodes to support the required number of simultaneous users, given QoS constraints. We further elaborate on the node count estimation tool in the next section.

### 3.3.2 Planning the Amount of Required Edge Nodes

The most critical part of the previously presented capacity planning framework is estimating the required edge capacity. Edge capacity is defined by the number of computing nodes within the edge layer. Previous research has suggested the use of benchmarking-based approaches to obtain optimal numbers of edge nodes [57]. However, benchmarking requires extensive simulations when system requirements change or a new functionality is introduced in the system. Therefore, our work aims to avoid benchmarking and to provide a more general node count estimation solution. Therefore, we propose using queuing theory and

\(^{1}\)https://www.docker.com/get-started
an M/M/\(k\) model. The model assumes exponentially distributed request arrival and processing times, \(k\) nodes for processing requests, and an infinite queue where incoming requests are queued when all \(k\) nodes are busy. In Publication V, we show that smart combinations of requests maximize resource utilization and minimize the number of required edge nodes. In order to apply the M/M/\(k\) model to the combinations of requests, we employed a weighted average approach, which shows consistent results assuming that the service is running for a long period of time.

The edge count estimation tool takes combinations of requests, request arrival rates, and the required QoS, which is expressed in terms of mean required response delays, as an input. When planning the required node counts, it considers both CPU and GPU requirements. It uses the output from the previously developed profiler tool to obtain resource usage patterns, categorize the requests, and estimate the mean processing times for the requests. It then calculates the node counts, with fixed CPU/GPU capacities on each node, needed to satisfy the given QoS. The developed tool is fast to execute, which represents an advantage over benchmarking-based solutions. For example, it can be used in dynamic deployment scenarios where QoS requirements often change or in scenarios when a new part of a resource pool must be assigned to support new requirements.

We evaluated the performance of the proposed capacity planning framework and the node count estimation tool by planning capacity for the AR-based navigation and information system introduced in Section 3.1. The system under evaluation supports image-based localization, video streaming from the server to clients, information retrieval, browsing, and real-time object recognition in AR mode. Therefore, it requires both CPU and GPU computing resources with different resource usage profiles for each request type. For example, image-based localization requires high CPU and GPU usage, whereas object recognition requires mostly GPU resources. The evaluation shows how the capacity planning framework helps to plan the required capacity of the compute-intensive real-time AR navigation system. Through extensive simulations, we demonstrated that the proposed M/M/\(k\) model can estimate edge node counts with 88% accuracy. The evaluation proved that our solution minimizes the amount of edge nodes required to deploy real-time compute-intensive systems while also satisfying QoS requirements.

### 3.4 Open Questions

In this section we discuss potential areas for future research based on the results of this dissertation.

First, our indoor mapping and positioning system is based on traditional SfM algorithms. Throughout the work, we use SIFT features, but other types of features, notably ORB [119], that demonstrate comparable performance and much faster processing times could also be used by the mapping system. Furthermore,
current advances in machine learning and the introduction of CNNs have proved extremely useful for tackling computer vision problems. As discussed in Section 2.1.3, CNNs have been successfully used for image-based localization and 6DoF pose estimation [65, 80]. The main advantage of CNN-based localization is that processing times are faster than those of conventional SfM. Thus, it is worth investigating the applicability of localization based on neural networks for the developed mapping system.

Second, we believe that fusing image-based localization with other types of localization techniques, such as those based on Bluetooth beacons or magnetic fields, would help to improve overall positioning accuracy. The mapping and localization system presented in Publication I could take advantage of existing positioning infrastructure to accelerate the selection of partitions for image-based localization and to improve robustness in texture-less areas.

Third, as well as using point clouds obtained via SfM, we could also use 3D point clouds obtained from Visual SLAM. Visual SLAM point clouds are generated on-the-fly and are more efficient to obtain than SfM point clouds. However, they are typically not as accurate as SfM point clouds. Nevertheless, they can supplement SfM point clouds by utilizing the SfM point cloud as an anchor and registering smaller Visual SLAM point clouds to it in order to speed up the mapping process.

Fourth, proper incentive mechanisms are needed to ensure rapid and consistent collection of crowdsourced data. In Publication II, we developed mechanisms for guiding crowdsourcing workers and for assessing the quality of the collected data. These can be used to design novel incentive mechanisms and location-based participant selection. We believe that such work would greatly accelerate progress in indoor mapping of public spaces.

Finally, privacy considerations regarding the collected crowdsourced data and overall system security are not within the scope of this dissertation. Recent regulations on personal data protection, such as the General Data Protection Regulation, enforces strict rules for processing personally identifiable data. It is highly relevant for vision-based crowdsourcing systems, as a photograph may disclose personal information about an individual or multiple individuals. Therefore, certain measures must be taken to properly inform participants and novel algorithms must be developed for data anonymization. Furthermore, there is a need for methods to ensure secure data transfer and storage mechanisms in order to prevent malicious users or third parties from accessing or tampering with collected data.
4. Conclusion

Over the last few decades, AR has attracted much attention from both academics and industry professionals. Recent developments in AR demonstrate great advances, but several important challenges remain to be addressed to enable seamless and ubiquitous AR experiences. This dissertation addresses the challenges of constructing indoor maps using visual crowdsourcing and using the maps to enable accurate indoor positioning, which are the fundamental elements of ubiquitous, multi-user AR. In particular, we focus on efficient vision-based indoor mapping, accurate and efficient device position and orientation tracking, and planning computing capacity for the deployment of such real-time AR systems.

To address the research questions, we conduct empirical studies involving the design, implementation, and evaluation of practical mapping and navigation systems suitable for enabling AR experiences in large indoor spaces. First, we develop an image-based indoor mapping and positioning system that uses crowdsourced data collected from imaging and inertial sensors. We also present guided crowdsourcing mechanisms for efficient indoor mapping. Second, we demonstrate how the developed mapping system, combined with position and facing direction tracking on a mobile device, enables energy-efficient and rigorous AR applications. Third, we propose techniques for planning computing capacity before deploying real-time systems in hierarchical edge cloud environments. We present a capacity planning framework that considers diverse resource requirements and reliably estimates the number of edge computing nodes needed to satisfy the QoS requirements of the previously developed AR navigation services.

The evaluation of the developed systems indicates their applicability for enabling ubiquitous AR and demonstrates the accuracy and efficiency of the developed image-based indoor navigation system. We believe that the results of our work can contribute to the development of a new wave of ubiquitous AR applications that would increase productivity, provide convenient services, and improve entertainment in society.
References


References


References


References


References


Errata

Publication IV

The phrase “utilize a 6 points algorithms” in the System Overview and Algorithms section should be “utilize a 6 points algorithm”.