DISS. ETH NO. 25756

Lifelong Mapping and Robust Place Recognition Using Cameras

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Marcin Tomasz Dymczyk

MSc ETH Electrical Engineering and Information Technology

born on August 10, 1989
citizen of Poland

accepted on the recommendation of

Prof. Dr. Roland Siegwart, Examiner
Prof. Dr. José Neira Parra, Co-examiner
Prof. Dr. Paul Newman, Co-examiner

2019

Autonomous Systems Lab
Department of Mechanical and Process Engineering
ETH Zurich
Switzerland

© 2019 Marcin Tomasz Dymczyk. All rights reserved.

Abstract

The ability to precisely localize a robot within the 3D space is a core capability for mobile robotics. Knowing a robot's precise pose permits to execute regular missions, accurately follow a precomputed trajectory or avoid obstacles, a skillset necessary for robotic tasks such as goods delivery, cleaning, inspection or navigating Micro Aerial Vehicles (MAVs). Additionally, estimating a pose in a joint map is vital for collaboration, where agents need to know their relative locations. The very same technology has already proven to be exceptionally useful in various consumer products, such as augmented reality (AR) or virtual reality (VR) devices. Motivated by these widespread applications, the goal of this thesis is to develop robust, reliable mapping and place recognition algorithms that cope with lifelong applications and challenging conditions. We focus on camera-based perception, as vision provides remarkably rich information about the surroundings using compact, lightweight and cost-efficient sensors.

The applications of mapping and place recognition algorithms, no matter if on a mobile robotic platform or a consumer device, have to adhere to specific constraints, such as limited computational resources or memory. Consequently, most of current state-of-the-art mapping solutions only support environments with a limited scale, often struggling to cover more than a single floor of a building. Long-term scenarios repeatedly mapping the same area create noisy maps with ever-growing size leading to imminent failures. Even if the maps are successfully constructed, the place recognition methods deliver subpar performance in challenging conditions, *i.e.* when illumination, weather or viewpoint vary. Common compact hand-crafted descriptors do not offer enough discriminative power to deal with such variability. To address those issues, this thesis presents robust, scalable solutions that will support real-world long-term deployment and robust place recognition irrespective of environment conditions.

The first contribution of this thesis addresses the scalability issues by summarizing multiple mapping sessions into a compact representation, a *Summary Map*. We propose a solution that removes the noise and clutter from multi-session maps, but still retains persistent features that are deemed useful to relocalize in the future. This allows us to reduce the size of the map and keep it bounded even in lifelong applications. Furthermore, we introduce methods that permit to evaluate features solely based on their appearance and discard some of them immediately after detection, thus reducing the computational footprint of mapping.

Even a map containing an optimal set of *summarized* features will not be useful if we cannot localize against it. Thus, we propose several methods that boost the performance of state-of-the-art place recognition approaches using visual con-

text. We show that augmenting hand-crafted descriptors brings significant gains in performance, particularly in challenging conditions.

The efforts centered around addressing the large-scale and long-term mapping and place recognition problems required a development of the infrastructure to conduct research. Therefore, we introduce a mapping and localization system called *maplab* that offers flexible interfaces and can be considered a tool for prototyping Simultaneous Localization and Mapping (SLAM) algorithms. It has been extensively tested in numerous applications, including mobile devices and robotic platforms, such as MAVs or wheeled robots. It offers a core set of algorithms that provide a jump-start into the world of vision-based mapping.

The theoretical contributions of the thesis are validated in real-life scenarios on a wide variety of robotic platforms. We demonstrate that we can maintain reliable vision-based maps over extended periods of time and keep the map size bounded. The maps could often be reduced by more than 90% without compromising place recognition performance. Our proposed place recognition methods deliver a boost in retrieval precision and recall, while still remaining lightweight. Finally, our mapping system has been open-sourced and successfully used by hundreds of researchers, promoting visual-inertial mapping and localization in the robotics community. Together, our contributions constitute a complete mapping and localization solution for mobile robotics, designed to deliver excellent performance in challenging conditions and robustness in lifelong applications.

Zusammenfassung

Die Fähigkeit eines Roboters sich exakt im dreidimensionalen Raum zu lokalisieren ist eine Kernkompetenz in der mobilen Robotik. Eine präzise Positionierung des Roboters erlaubt es zum Beispiel einfache Missionen auszuführen, einem vorberechneten Pfad zu folgen oder Hindernisse zu umgehen. Diese Fähigkeiten werden unter anderem benötigt um Güter auszuliefern, zu navigieren oder Aufgaben wie Reinigung, Inspektion zu meistern. Des Weiteren ist die Positionsschätzung in kombinierten Karten eine grundlegende Voraussetzung für die Zusammenarbeit zwischen Agenten, die ihre relative Position wissen müssen. Genau die gleiche Technologie hat sich bereits in verschiedenen Produkten, wie zum Beispiel in Geräten für die Virtuelle oder Erweiterte Realität (VR/AR), bewährt. Durch diese weit verbreiteten Anwendungen motiviert, wollen wir in dieser Dissertation eine robuste und zuverlässige Kartografierung und Lokalisierung entwickeln, die langfristige Anwendungen in herausfordernden Bedingungen ermöglicht. Dabei konzentrieren wir uns auf die kamerabasierte Wahrnehmung, da diese basierend auf einem kompakten, leichten und günstigen Sensor bemerkenswert dichte Informationen über die Umgebung liefert.

Unabhängig davon, ob auf einem Roboter oder tragbaren Mobilgerät, die Algorithmen zur Kartographierung und Lokalisierung müssen unter ähnliche restriktiven Bedingungen arbeiten, wie zum Beispiel beschränkte Rechenleistung oder Arbeitsspeicher. Infolgedessen unterstützen die gegenwärtigen Spitzentechnologien meist nur eine limitierte Kartengrösse und habe Schwierigkeiten mehr als ein einzelnes Stockwerk eines Gebäudes abzudecken. In langfristigen Szenarien wird wiederholt dasselbe Gebiet kartographiert, was meist zu ungenauen und unweigerlich wachsenden Karten führt und letzten Endes zum Versagen des Systems. Auch wenn es dem System erfolgreich gelingt eine derartige Karte zu erstellen, so leidet die Leistungsfähigkeit der Lokalisierung meist unter herausfordernden Bedingungen, hervorgerufen durch schwierige Lichtverhältnisse, Wetter oder Unterschiede im Betrachtungswinkel. Die gängigen, handgefertigten Deskriptoren liefern nicht genügend aussagekräftige Übereinstimmungen um eine solche Variabilität zu meistern. Dadurch motiviert, wollen wir diese Probleme angehen indem wir skalierbare Lösungen einbringen, welche es durch robuste Lokalisierung ermöglichen ein System unabhängig von der Umgebung, langfristig und unter realen Bedingungen einzusetzen.

Das Problem der Skalierbarkeit eines solchen System wird im ersten Beitrag dieser Dissertation erforscht indem mehrere Kartografierungsmissionen in einer kompakten Representation zusammengefasst werden, einer komprimierten Karte zur Lokalisierung. Wir präsentieren ein Lösung, die Rauschen und Artefakte,

hervorgerufen durch ungenaue und inkorrekte Verarbeitung der Sensordaten, aus Karten, die aus mehreren Datensätzen erstellt worden sind, entfernt. Dabei werden persistente Landmarken, welche als nützlich für die zukünftige Lokalisierung erachtet werden, beibehalten. Dies erlaubt es uns die Kartengrösse zu reduzieren und in langfristigen Anwendungen unter einer gewissen Obergrenze zu halten. Des Weiteren führen wir neue Methoden ein, die es erlauben markante Bildpunkte, einzig basierend auf deren Erscheinungsbild, bereits kurz nach deren Berechnung auszusortieren. Dies ermöglicht es den Rechenaufwand entscheidend zu reduzieren.

Allerdings ist eine Karte, auch wenn sie ein optimales Set von Landmarken enthält, nur hilfreich wenn man sich erfolgreich dagegen lokalisieren kann. Deshalb, präsentieren wir mehrere Methoden, welche durch das Anreichern der Deskriptoren mit Bildkontext die Leistungsfähigkeit des aktuellen Standes der Technik noch weiter erhöht. Wir zeigen dabei auf, dass diese Art von Anreichern von handgemachten Deskriptoren eine signifikante steigerung der Leistungsfähigkeit hervorruft, gerade unter schwierigen Bedingungen.

Unsere Forschung im Bereich der Skalierbarkeit und Langlebigkeit von Karten, wie auch in des Problems der Lokalisierung, erfordern die Entwicklung einer passenden Infrastruktur. Aus diesem Grund haben wir *maplab* entwickelt, ein System für die Kartografierung und Lokalisierung, welches flexible Schnittstellen bietet für die Erforschung von Algorithmen zur Simultanen Positionsbestimmung und Kartenerstellung (SLAM). Das System wurde ausgiebig und auf vielen verschiedenen Robotersystemen, wie kleine Multikopter, Fahrzeuge und Mobilgeräte, getestet. Maplab bietet eine Sammlung von grundlegenden Algorithmen, die einen schnellen Einstieg in die Welt der Kamera-basierten Kartografierung ermöglichen.

Die theoretischen Beiträge dieser Dissertation werden in Anwendungen aus der realen Welt und auf einer Vielfalt von Robotersystemen getestet. Wir zeigen auf, dass wir Kamera-basierte Karten zuverlässig und über einen längeren Zeitraum instand erhalten und dabei eine Obergrenze für die Kartengrösse durchsetzen können. Die Grösse der Karten kann dabei meist um 90% reduziert werden, ohne die Leistungsfähigkeit der Lokalisierung signifikant zu verschlechtern. Die präsentierten Lokalisierungsmethoden erreichen eine Steigerung der Trefferquote und Genauigkeit, benötigen dabei aber nur begrenzt mehr Rechenleistung. Wir haben maplab, unser System zur Kartografierung, als Open-Source Software veröffentlicht, woraufhin es von hunderten von Nutzern adaptiert wurde. Wir hoffen damit einen Beitrag zur weiteren Verbreitung von Kamera-basierter Kartografierung und Lokalisierung in der Forschungsgemeinschaft zu leisten. Der Beitrag dieser Dissertation besteht aus einer kompletten Kartografierungs und Lokalisierungslösung für mobile Roboter und zeichnet sich vor allem durch ihre ausgezeichnete Leistung und Robustheit unter schwierigen Bedingungen aus.

Acknowledgements

This doctoral thesis would not have been possible without help, support and contributions from numerous people. First and foremost, I would like to express my sincere gratitude to Prof. Roland Siegwart, the head of Autonomous Systems Lab of ETH Zürich, who gave me an opportunity to write a PhD thesis within his lab. Thank you for the great trust towards all of us in the team, optimism and open-minded approach to our ideas. I truly appreciate how freedom in conducting the research triggers great collaborations and a friendly atmosphere among all lab members. Thanks to Prof. Dr. José Neira and Prof. Dr. Paul Newman for accepting to co-supervise this doctoral thesis and for developing the state-of-the-art of visual-based methods which constituted a solid base for my investigations.

I would also like to express my deepest gratitude to all my supervisors at Autonomous Systems Lab of ETH Zürich. Thank you for bringing motivation, countless hours of discussion and good atmosphere to our team. I would like to thank Dr. Paul Furgale and Dr. Mike Bosse, who initiated my fascination in visual-based mapping and localization. Simon Lynen, who created the exciting vision of a multiagent visual-inertial mapping system and initiated collaborative effort towards this goal. Then, Dr. Igor Gilitschenski, Dr. Cesar Cadena and Dr. Juan Nieto who were always open for discussion and provided help and invaluable feedback on my work and research directions. I particularly appreciate all the long days (and nights) spent working together just before submission deadlines.

It is hard to name all members of Autonomous Systems Lab who contributed to this thesis. All of you provided invaluable help, motivation and inspiration. For me personally, it was an amazing experience to collaborate with such a unique team of talented roboticists. I would like to highlight the contributions of the mapping team: Thomas Schneider, Marius Fehr, Titus Cieslewski, Mathias Bürki, Timo Hinzmann, Florian Tschopp, Andrei Cramariuc, Mathias Gehrig and Abel Gawel – I truly appreciate all you efforts. It was a pleasure to work and brainstorm ideas together. It was also exciting to deploy our algorithms on various robotic platforms, with a particular help from the rotary wing team, including Dr. Michael Burri, Helen Oleynikova, Dr. Zachary Taylor and Alex Millane. Furthermore, I am grateful to Dr. Janosch Nikolic, Dr. Jörn Rehder, Pascal Gohl and Prof. Dr. Stefan Leutenegger for introducing me to the world of calibration and estimation respectively. My understanding of visual-based localization was vastly extended by discussions with Prof. Dr. Torsten Sattler. Finally, I would like to thank Dr. Renaud Dubé, Fadri Furrer, Dr. Dario Bellicoso and Karen Bodie for fruitful discussions and plain fun when travelling, hiking or skiing.

The results of the thesis would not have been the same without enormous support from the administrative and technical staff at the lab. Big thanks to Luciana Borsatti, Cornelia Della Casa, Michael Riner, Konrad Meyer, Amir Melzer and Stefan Bertschi for taking care of ASL. Particularly when the tight deadlines approached, your contributions were invaluable.

Over the course of the doctoral studies, I have supervised several extremely talented and motivated students. I would like to thank all of you for your true dedication to the projects. I express my particular gratitude to Paul-Edouard Sarlin, Fabian Blöchliger, Kevin Egger and Antonio Loquercio for their hard and inspiring work.

I would also like to thank the Sevensense Robotics team. In recent months, being close to the industrial applications of the mapping technology brought a lot of satisfaction and excitement. Special thanks to Renaud Dubé, who planted in me the idea of entrepreneurship while sitting on the plane heading to ICRA 2018 in Brisbane. I truly believe that we are only at the very beginning of the Sevensense journey now.

My friends outside of academia brought inspiration and worked hard to keep my work-life balance reasonable. Big thanks to my friends in Poland, Marta and Bartosz, Martyna and Piotr, and friends in London, Krzychu and Natalia. I also wanted to thank my friends in Zürich, Beata, Tunç and Ola. Additionally, special thanks to my flatmate Dawid, for exploring Zürich together during our early days here and tolerating random robotic bits and bobs all over our flat.

Lastly, I would like to express my deep gratitude to my family. Special recognition goes to my mother, who always supported me, motivated me to aim high and provided true love. Without her, I would probably have never ended up being who I am now. Thanks to my grandfather who always emphasized the value of education and hard work. Thank you to my father, who sparked my interests in engineering by making me understand how things work. Finally, an enormous thanks to my girlfriend Julia, who provided unconditional love, optimism and support even in the toughest days. Your creativity and dedication are impossible to value.

May 7, 2019 *Marcin Dymczyk*

Financial Support

The research leading to the results presented in this thesis has partially been funded by Google's Project Tango and Google's Visual Positioning System.

Contents

Abstract **i**

Zusammenfassung **iii**

Acknowledgements **v**

Preface **1**

1 Introduction **3**
1.1 Motivation and Objectives 4
1.2 Approach 8

2 Contributions **13**
2.1 Part A: Efficient Vision-Based Mapping 13
2.2 Part B: Robust Place Recognition 22
2.3 List of Publications 25
2.4 List of Supervised Students 27
2.5 List of Open-Source Software 29

3 Conclusion and Outlook **31**
3.1 Part A: Efficient Vision-Based Mapping 31
3.2 Part B: Robust Place Recognition 34

A. EFFICIENT VISION-BASED MAPPING **37**

Paper I: The gist of maps-summarizing experience for lifelong localization **39**
1 Introduction 40
2 Related Work 41
3 Methodology 43
4 Experiments 47
5 Results 48
6 Conclusions 52
7 Acknowledgments 52

Paper II: Keep it brief: Scalable creation of compressed localization maps **55**
1 Introduction 56
2 Methodology 58
3 Experiments 63
4 Conclusions 69
5 Acknowledgments 70

Paper III: Erasing bad memories: Agent-side summarization for long-term mapping **71**
1 Introduction 72
2 Related Work 74
3 Methodology 75
4 Experiments 81
5 Conclusions 86
6 Acknowledgments 88

Paper IV: Will it last? Learning stable features for long-term visual localization **89**
1 Introduction 90
2 Related Work 92
3 Predicting feature reliability 93
4 Methodology 94
5 Experiments 97
6 Conclusion 102

Paper V: maplab: An open framework for research in visual-inertial mapping and localization **107**
1 Introduction 108
2 Related work 110
3 The maplab framework 110
4 Use-cases 115
5 Using maplab for research 122
6 Conclusions 123
7 Acknowledgement 123

B. ROBUST PLACE RECOGNITION **125**

Paper VI: Efficient Descriptor Learning for Large Scale Localization **127**
1 Introduction 128
2 Related Work 129
3 Methodology 130
4 Experimental Setup 135
5 Results 137
6 Conclusions 140

Paper VII: LandmarkBoost: Efficient Visual Context Classifiers for Robust Localization **141**
1 Introduction 142
2 Related work 144
3 Methodology 145
4 Experimental evaluation 151
5 Conclusions 156

Bibliography **159**

Curriculum Vitae **179**

Preface

This thesis is a *cumulative doctoral dissertation* and as such contains the most relevant publications of the author, grouped into two major parts and attached at the end.

Chapter 1 is an introduction to the thesis that acquaints the reader with the motivation of the research and defines an intersection of the contributing publications. We also explain the importance of visual mapping and localization technology and explain the relevance of the thesis within the field. Furthermore, we present the high-level goals of the research and the approaches taken to address the challenges. Then, in Chapter 2, we introduce the contributing publications and explain how they relate to the overall picture of robust and efficient localization and mapping. Moreover, we highlight the relationships between the publications, guiding the reader along the timeline of our investigation. Finally, in Chapter 3, we summarize the results of the doctoral research and provide an outlook to the future.

Chapter 1

Introduction

Over the last decade, advancements in the field of mobile robotics have been particularly pronounced. More and more tasks that were traditionally performed by humans can now be executed by robots. This is particularly noteworthy in situations where those tasks are tedious, require particular precision or pose the human health and life at danger. The robots are also becoming a part of our daily routines, with robotic vacuum cleaners or smart assistants inhabiting our homes. Furthermore, mobile platforms will soon become ubiquitous in crowded spaces, coping better and better with dynamic obstacles and unstructured environments. This widespread use of mobile robotics poses challenges on algorithms used to guarantee safety and reliability of such devices. In the end, we expect robots to be always at our disposal and to *never* be lost.

There are numerous applications of mobile robotics that evolved owing to advancements in mapping technology. The industry starts to deploy mapping and localization for indoor logistics, warehouse automation, surveillance and inspection of various structures. Agriculture is one of the rising markets, where monitoring the crops by MAVs and fixed wing planes will soon increase the efficiency and lower the usage of pesticides and fertilizers. A significant media coverage is dedicated to autonomous driving, where estimating the precise location of a vehicle within annotated maps helps to stick to the traffic regulations and avoid collisions. Finally, the steadily decreasing cost of the sensing and computation hardware brings SLAM technology to the households. Examples include the entertainment sector, with various AR and VR applications, but also mobile phone mapping apps, robotic lawn mowers and vacuum cleaners.

One of the core technologies that catalyzed this mobile robotics revolution are algorithms related to mapping and localization. They provide spatial awareness and knowledge of the global pose to the robotic platforms, a *sine qua non* requirement for path planning, navigation and manipulation. Another factor that contributed to the popularity of mobile robotics is the availability of compact, power efficient

computational units. We also should not omit the recent developments in SLAM-related sensor technology that include more sensitive global-shutter cameras, low-noise Microelectromechanical System (MEMS) Inertial Measurement Units (IMUs) and 3D Light Detection and Ranging (LiDAR) sensors.

There are multiple sensing modalities that can be used to perform pose estimation, mapping and place recognition. Outdoors, one of the most natural approaches would be to use the Global Positioning System (GPS). Its limited accuracy, however, does not permit to perform precise navigation. Additionally, we want robots to operate in GPS-denied environments as well. Many contemporary industrial applications depend on 2D and 3D LiDARs. These sensors provide accurate measurements, but are much larger and heavier than cameras, contain moving parts and are prohibitively expensive for many cost-sensitive applications.

This leads us to believe that camera-based systems coupled with inertial sensors are a viable solution for a wide range of industries. Such configuration closely resembles human anatomy, *i.e.* our eyes provide visual cues and our inner ear various inertial measurements. By processing the readings of the visual-inertial setup, state-of-the-art odometry systems provide accurate, high-frequency pose estimates. Most of those algorithms simultaneously construct a visual map of the environment in a process called mapping. The map might serve several purposes, including tasks such as inspection or path planning. In the context of this thesis, we consider such prior visual map to be particularly useful for relocalization during subsequent visits to the same area. The process of mapping and localization, while reliable in controlled spaces, remains an open challenge in large-scale, dynamic environments and scenarios that require lifelong map maintenance. Those limitations hinder the expansion of SLAM to even wider range of applications.

The goal of this thesis is to develop versatile vision-based mapping and localization algorithms suited for large-scale and long-term deployment of mobile robots. Our research focuses on dealing with GPS-denied, unstructured, dynamic environments and multiple mapping sessions in varying conditions. We want to guarantee that our system accurately constructs a visual map, maintains it over extended periods of time and manages to recognize the same place irrespective of the viewpoint, illumination or weather. We aim at creating a system that can be deployed on a wide variety of robotic platforms used at Autonomous Systems Lab, see Fig. 1.1, with a minimal engineering effort. We believe that fulfilling those requirements is a key to push further vision-based mobile robot autonomy.

1.1 Motivation and Objectives

The growing number of applications of mobile robots and pose estimation systems in general sparked an unprecedented interest in the vision-based mapping and localization. The wide variety of applications calls for robustness both indoors and outdoors, which remains an open challenge for state-of-the-art algorithms. Contemporary mobile robots, evolving from low-volume expensive research platforms to high-volume industrial and consumer devices, increase the demand for SLAM

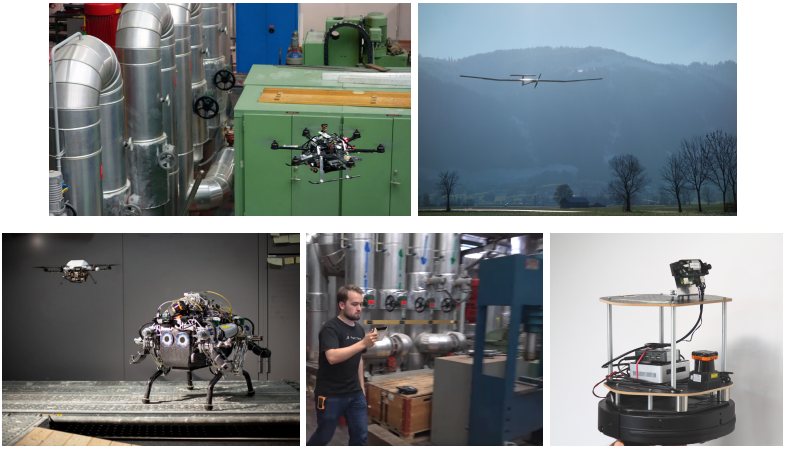


Figure 1.1: A broad variety of applications of vision-based mapping and localization technology within the Autonomous Systems Lab. The algorithms developed within this doctoral thesis, and centered around the *maplab* framework [187], aim at addressing the needs of all the depicted platforms. From left to right, top to bottom: MAVs used for inspection in industrial, GPS-denied environments [27], AtlantikSolar autonomous solar-powered UAV [162], collaboration between flying and legged platforms using vision-based place recognition [59], visual-inertial estimation and mapping on a mobile consumer device [61] and a wheeled robot relocalizing daily within a busy office space [52].

solutions that are easy to deploy and cost-efficient. Particularly the price considerations drive the development towards affordable, generic hardware instead of proprietary sensors that are expensive to manufacture. Furthermore, certain applications, *e.g.* mobile phones, tablets and smaller robots, put additional, hard constraints on the size and weight of the sensor suite. Finally, the rapid progress in machine learning and artificial intelligence brings revolutionary algorithms that commonly depend on images as input. All of those factors motivate us to focus on vision sensors and, consequently, vision-based mapping and place recognition. We believe that the synergy of mapping and place recognition is essential for successful deployment of visual SLAM [29]. Now, we would like to take a closer look at those two building blocks to better understand the challenges in front of us.

The first building block empowering applications of vision-based mapping and localization is precise odometry estimation and map building. The odometry system processes sensor measurements and outputs real-time 6-Degree-of-Freedom (DoF) pose estimates of the agent. Often, a map of the environment is built simultane-

Figure 1.2: A pair of images from Oxford RobotCar dataset [137] that were taken in the same place. It proves how changing illumination conditions cause dramatic variations of appearance. Robust place recognition algorithms must cope with such challenging cases. In this exact example, state-of-the-art approaches, such as SIFT [132] or LIFT [230], fail to establish correct correspondences.

ously. The knowledge of a precise pose is necessary to properly control the robot, navigate to waypoints or place and display 3D objects in AR and VR. Certain scenarios require precision in the order of centimeters or millimeters, *e.g.* when interacting with industrial environments or to guarantee a fully immersive experience to the user. In those applications, externally mounted sensors are an alternative to inside-out tracking and visual odometry. They deliver excellent precision, but require time-consuming installation and setup. Furthermore, we want our algorithms to work everywhere, without limiting them to specifically preconfigured environments. To alleviate those constraints, we want to deliver a stand-alone odometry and mapping solution that uses cost-efficient vision-based sensing. The agents should be able to work reliably even in busy spaces full of dynamic objects and in presence of illumination changes. We also want the constructed maps to remain compact, without redundant elements or dynamic objects that are not persistent in the environment. This drive to increase the system efficiency, improve the scalability and reduce the memory consumption results in various algorithms developed to maintain visual maps over multiple sessions. We focus on map summarization techniques that extract a gist of multi-session maps, discarding the noisy, outdated or unnecessary information.

The second building block is the place recognition capability. Recognizing a place and matching it to the prior map has many applications, but we would like to highlight the two we consider the most fundamental. First, as the local odometry estimates drift, at some point they do not provide any reliable information about the agent's pose in the environment. One way to correct them is to recognize the place in a prior map or in a map being currently constructed. Realizing that *"we have seen this place before"* adds additional constraints and thus reduces the error. Secondly, place recognition algorithms permit to register an agent in the prior map, *i.e.* estimate where it is within a previously created representation.

To achieve this goal, algorithms need to identify a correspondence between the current view and the map, which proves challenging when the appearance has changed considerably, as depicted in Fig. 1.2. A straightforward approach is to use easy-to-be-recognized patterns, such as April Tags [166], and mount them in predefined spots of the environment, but that is infeasible for large-scale outdoor spaces. To address the shortcomings of using fiducial markers, the current state-of-the-art methods depend on natural features of the environment and describe them using hand-crafted [121, 132] or learned features [44, 196, 230]. The features can vary from global image descriptors, descriptors of smaller image regions to those embedding only local image patches. The features of the query image and the map are then matched, often using a nearest neighbor search in the descriptor space, and if they are similar enough and geometrically consistent, a loop closure is confirmed. The place recognition system is therefore as robust as its descriptors. We expect them to be invariant to changes, *e.g.* in illumination, elements of the scene structure, weather or seasons. Our goal is to build on top of the current state-of-the-art by researching approaches that improve the description of images, facilitate the correspondence search and ultimately make the place recognition more robust.

Finally, with the growing number of applications of vision-based mapping systems and continuous advancements of related algorithms, the frameworks themselves grow in complexity. Numerous research directions are being investigated by the robotics and computer vision community. We believe the research on vision-based mapping and localization could significantly profit from a common set of basic algorithms and interfaces, that help to reuse existing implementations and compare the results more easily. Over the time of the doctoral studies, we have developed a framework that can be used by researchers working with visual mapping and localization, dense reconstruction, scene understanding, as well as those who deploy such methods on robotic platforms. Moreover, each of the implementations contributing to this doctoral thesis is implemented and evaluated using this system. The framework, called *maplab*, has been published and open-sourced to the benefit of entire community.

Overall, the visual mapping and localization technology is getting more and more mature and becoming widespread in industrial solutions, service robotics and consumer devices. It is a key to guarantee spatial awareness with a limited cost, weight and computational power. We want to make it more reliable and robust in real-life environments, including crowded indoor and outdoor spaces, in the presence of dynamic objects and changes of the weather, seasons or illumination. We aim to construct compact maps that summarize well the environment and do not contain objects that are likely to disappear. Finally, we strive to improve the place recognition algorithms to ease the registration of a device within a previously constructed map.

In other words, the overarching goal of this doctoral thesis is to advance the current state-of-the-art in efficient vision-based mapping and robust place recognition with the particular objectives:

Mapping and localization at scale Perception is one of the limiting factors of mobile robotics. So far, the environment size and the long-term deployment posed a big challenge for perception systems. Our goal is to research the algorithms that permit to reliably estimate the agent's pose even in large environments. We also want to localize the agent in the prior map, no matter what the conditions are. Furthermore, we want a consistent map constructed out of past experiences of the agent that are co-registered and jointly refined.

Building maps in dynamic environments While the current mapping and localization algorithms perform reasonably well in controlled environments of a limited size, they struggle in the presence of noise or dynamic objects. We aim to deal with dynamic or temporary objects and changes in the environment. We want to construct maps that only contain persistent elements of the robot's surroundings, but reject everything that is not deemed stable. Consequently, we will make the place recognition process more robust and strip the maps off redundant or noisy elements.

Reliably recognizing places Contemporary mobile platforms mostly use sparse, hand-crafted features to construct maps and to perform visual place recognition. These techniques are often not robust enough to deal with the changes in illumination and viewpoint. They commonly fail with changing seasons or simple day-night cross-localization. We propose methods that aim to help the sparse feature matching and, as a result, improve the localization performance.

Developing core tools for mapping The researchers focusing on vision-based mapping and localization lack efficient tools to prototype their algorithms and evaluate them. No widespread standard means that the experiments often require reimplementing many of the basic algorithms which is redundant, error-prone and simply time consuming. Furthermore, the lack of reusable implementations of basic algorithms affects the credibility of reported results, making it difficult to reproduce them.

1.2 Approach

To present our contributions leading towards the aforementioned goals, we group our work in two parts. In Part A we study the algorithms that enable efficient, large-scale and long-term mapping. The utility of the mapping process, however, is severely limited without localization and place recognition, *i.e.* a capability to associate new measurements with the existing map. We therefore examine several approaches to improve the reliability and performance of place recognition in Part B. Overall, we assume to deal with a client-server type of a system architecture, where an agent can request a prior map from the mapping backend that maintains a consistent global representation. The data flow in such a system is presented in Fig. 1.3.

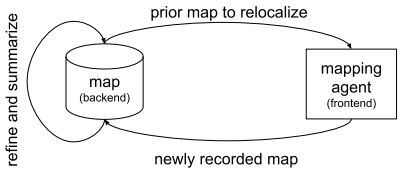


Figure 1.3: The diagram depicts the basic architecture of the mapping system we consider. The agent requests maps from the backend and localizes within them. At the same time, it collects new mapping data that is fed back to the backend and can be integrated into a global map. The backend periodically refines and summarizes the global map to reject outdated content. Such system scales well to multiple agents and requires limited computational power on the agent side, which corresponds to constraints of mobile robotics.

Part A: Efficient Vision-Based Mapping

The goal of this part of the thesis is to enable efficient vision-based mapping, with a focus on large-scale, lifelong applications. We investigate various centralized and decentralized system architectures and aim to reduce the requirements such as computational power, memory usage, storage or bandwidth. We take a close look at maps created in long-term scenarios. We want them to be compact and to contain only the necessary information for future place recognition queries. Our motivation is to let robots out of confined, often controlled, environments to large, unstructured areas with dynamic objects.

We therefore propose a novel approach to deal with multi-session and long-term maps with noisy measurements. We introduce the *Summary Map*, a compact map definition that is tailored to the needs of the localization system and contains a compressed representation of the environment. Along with the representation itself, we propose a handful of algorithms that permit to create such maps based on the past observation statistics, area-specific features and raw appearance. Our feature selection methods are data-driven, as *repeatability* and *matchability* of natural visual features are hard to model. Finally, to facilitate our experiments, we introduce a graph-based mapping and localization system capable of building large-scale multi-session maps.

Map summarization Once a map has been constructed, our goal is to relocalize within it when visiting the same place again. Clearly, the map will most probably contain some noise or outdated data, a consequence of dynamic or non-persistent objects appearing in the sensor data stream. Such spurious map elements bloat the size of the visual map and might negatively affect the localization procedure. In our initial method (Paper I), we propose a heuristic summarization approach that permits to discover which features of the map were reliable in the past. We have also investigated what are the

limits of summarization and how many features in the *Summary Map* do we need to reliably relocalize and to correct the drift of local odometry. The approach achieves 10-fold map size reduction with a marginal localization performance loss. Subsequently, we have extended the method proposed in [167] and introduced a scalable optimization-based approach (Paper II), that improves the summarization quality and runtime.

Feature persistence prediction The previous summarization methods can be extended by using directly the local feature statistics and appearance instead of past observation data. It is clear that we humans can quite reliably judge which objects in the image will most probably move or completely disappear in a short period of time. Inspired by results presented in [85], we have proposed two methods to select features directly on the frontend side. We have answered the question whether we can use traditional statistical methods (Paper III) or deep learning based on the local appearance (Paper IV) to directly find out which features are worth retaining. These feature rejection techniques reduced the computational footprint of the mapping process and permitted to significantly limit the bandwidth when transferring a new map to the backend.

Efficient map structure We believe that efficient large-scale and long-term mapping requires a custom-tailored map structure. We propose a sparse visual-inertial map that efficiently indexes keyframes, inertial data and triangulated 3D landmarks (Paper V). It seamlessly supports multiple mapping sessions that can be co-registered in a single coordinate frame, but still remain easy to handle separately. This allows to perform efficient Least Squares (LS) batch optimization and place recognition based on 2D-3D matching [185]. The proposed map structure directly supports experience-based localization [35] or several appearance definitions per 3D landmark [25]. Finally, our map can be augmented with additional sensor modalities such as GPS, wheel odometry or LiDAR measurements to broaden the range of its possible applications.

SLAM research testbed Our map summarization experiments demanded a modular system that would include an efficient, but also extensible map structure. Unfortunately, most of off-the-shelf robotic mapping and localization frameworks focus primarily on accuracy and runtime, lacking true extensibility. While the mapping frontend should indeed minimize the resource usage, we believe that the offline backend of the system should facilitate performing experiments and prototyping algorithms. We have therefore developed a system (Paper V), a *de facto* side-product of the research efforts, that is easy to use and provides a user with core SLAM-related utilities.

Part B: Robust place recognition

The second part of the thesis addresses the problem of robust place recognition, focusing on long-term and large-scale scenarios. Even a perfect map has limited

value if algorithms are not able to establish correspondences between a query image (or a sequence of images) and the prior map elements. The type of correspondences can vary from image-to-image matching [5, 96], region detectors [143] to local, sparse features [3, 44, 121, 132]. We have decided to rely on sparse 2D-3D matching [185], as it has proven to be efficient and generally yields accurate 6-DoF poses even within city-scale models [94, 147, 186].

Most of the contemporary place recognition systems in robotics rely on local hand-crafted features [3, 121, 181], mostly due to the computational constraints. Such features struggle, however, with illumination, appearance or seasonal changes and even slight viewpoint differences negatively affect the matching result. We believe the place recognition performance has to improve significantly to make robots reliably navigate outdoor environments or crowded, dynamic spaces. In this thesis, we introduce methods that help 2D-3D feature matching and still remain lightweight to satisfy the constraints of real-time robotic applications.

Machine learning for feature retrieval The lightweight features used by robotic SLAM systems are often binary-valued, which makes the correspondence search within large-scale models challenging [215]. To address this issue, Lynen *et al.* [134] proposed a method to project binary descriptors to obtain compact, real-valued descriptors. The linear projection relied on a classic Principal Component Analysis (PCA). Instead, we introduce a novel linear projection method that is trained in a supervised fashion and uses the triplet loss formulation (Paper VI). The training procedure aims at maximizing the discriminability and matchability of resulting projected descriptors. The new projection scheme significantly improves the descriptor retrieval statistics and yields more robust place recognition, particularly in challenging environments. We also explore alternative methods to perform 2D-3D matching. Instead of a conventional nearest neighbor search, we propose a classification-based solution (Paper VII) that scales well to tens of thousands of landmarks and increases the correspondence search quality.

Understanding visual context Even with an improved projection method, the small size of a receptive field of common descriptors limits the place recognition performance. Many features in man-made environments are locally identical (*e.g.* light switches) which leads to outlier matches and pose estimation failures. To address this issue, we propose novel methods that take into account a wider visual context of a keypoint but still seamlessly integrate into existing mapping systems. In Paper VI, we introduce a method that embeds the context around a keypoint using a Convolutional Neural Network (CNN) and fuses it together with a local descriptor pattern during the descriptor projection step. We demonstrate that local context boosts the retrieval performance especially when visual aliasing is present. Then, Paper VII proposes a boosted classifier that takes into account not only the local descriptor, but also the other features in the image plane.

Capturing the relationships of visual features Generally speaking, most state-of-the-art place recognition methods rely on matching pairs of visual features and then performing a geometric verification of some sort. They fail, when the features are not discriminative enough, as many outliers are retrieved and the geometric verification step cannot cope with a low inlier ratio. We believe, however, that geometrically-encoded reoccurring patterns of features (*e.g.* "there is always a plant next to the light switch") are much more unique and provide better quality matches. Our boosting-based approach (Paper VII) encodes those geometric relationships between features in the image plane. It generates context regions all over the image plane and looks for reoccurring patterns for each of the salient visual features. The regions are anchored relative to the feature location and are scale, translation and rotation invariant. This novel approach to feature matching yields a significant improvement over the binary descriptor baseline.

Chapter 2

Contributions

This chapter presents our contributions to the field of vision-based mapping and place recognition, particularly focusing on long-term and large-scale scenarios. The list of publications is divided into two sections corresponding to the split proposed in Chapter 1. In Section 2.1, we introduce the contributions related to efficient, large-scale and lifelong mapping algorithms. Then, in Section 2.2, we focus on robust place recognition and localization. Each publication is introduced with a research context, motivation and the link with the high-level thesis objectives. We also present the specific contributions of each paper within the thesis. Furthermore, we introduce the dependencies and relationships between the contributing papers.

Additionally, in Section 2.3 all of the contributed publications are listed. In Section 2.4 we present all student projects supervised during the course of doctoral studies. Finally, in Section 2.5 we briefly introduce the major open-source software contributions.

2.1 Part A: Efficient Vision-Based Mapping

The following publications are focusing on vision-based mapping algorithms that cope with large-scale, lifelong scenarios. Handling those conditions is vital for successful long-term mobile robot deployment. Additionally, we introduce a reliable research-oriented mapping system that can support multiple mapping sessions and large scale maps. It is an open-source project available at `https://github.com/ethz-asl/maplab/` with a comprehensive set of demonstrations. Most contributions presented in this chapter are part of the *maplab* framework.

Paper I

Marcin Dymczyk, Simon Lynen, Titus Cieslewski, Michael Bosse, Roland Siegwart, Paul Furgale, "The gist of maps-summarizing experience for lifelong localization". In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015.

Context

State-of-the-art mapping and localization solutions mostly focus on small interior spaces and a single mapping session that is reused for localization after a short period of time [20, 27, 109, 174]. This limits the applicability of visual mapping and localization to rather short deployment timeframes and confined environments with controlled illumination. We consider this a very severe restriction for certain applications, where multiple autonomous visits to the same environment are to be expected. Examples include service robotics for cleaning, inspection and maintenance.

Our goal is to address the large-scale and long-term localization problem, assuming existence of dynamic objects and noise in the environment. We want the algorithms to rely on inexpensive cameras and IMUs, maintain an up-to-date map and run in real-time on computationally constrained platforms, such as low-cost mobile robots or mobile consumer devices. Additionally, our aim is to support multiple mapping sessions in a crowd-sourced mapping scenario that would be reliably co-registered and merged together. Our system should be able to cope with real-life environments, *e.g.* shopping malls or university hallways. We also want to guarantee that a resulting long-term map would be of constant-size, capturing the stable elements of the environment. We call such map a *Summary Map*.

In contrast to the existing research [14, 15], we do not focus on the fiducial marker placement problem or just geometric solution for the feature selection. In the feature-based visual pipelines we consider, future feature utility is strictly correlated with the keypoint and descriptor properties that are hard to model. We have therefore decided to employ a statistical approach, where past mapping *sorties* are used to predict the visual feature persistence and matchability.

Contribution

The paper proposes a heuristic approach to *summarize* a sparse visual-inertial multi-session map. The algorithm looks for features, *i.e.* triangulated 3D landmarks, that are deemed good for localization during subsequent visits to the area. It highly scores features that are consistently reobserved and matched, assuming these belong to permanent objects that are easy to redetect by common feature detectors. The method, however, cannot remove too many features as that would leave many of the keyframes underconstrained. To mitigate this danger, we exercise special care to guarantee even coverage of the entire mapped area. The paper also introduces an approach to remove redundant keyframes from the map by detecting

overlapping viewpoints. Overall, efficient pruning of both features and keyframes allows us to reduce the map size and keep it bounded.

The proposed method was evaluated on several datasets recorded in a dynamic public environment. We have demonstrated that iterative summarization process leads to a stable place recognition performance. Even though the places change between the *sorties*, the map size is kept constant and the persistent features are preferred whereas the dynamic objects (*e.g.* humans, temporary installations) are discarded. The final constant-size map contains contributions from each of the recordings and the first mapping session contributes the most. Quantitatively, the map created by common mapping algorithms can undergo 10-fold 3D point reduction with marginal localization performance loss.

Interrelations

The paper introduces the map summarization methodology on the backend side of the mapping system, where all the maps from the agents are collected. This backend map summarization method was adopted by automotive applications [26, 154], aerial [27] and ground [163] robotic platforms. The methodology was later revisited and reformulated as an optimization problem in Paper II.

The proposed backend summarization approach reduces the bandwidth from the backend to the agents when trying to relocalize, but, assuming a lifelong mapping scenario, the agents still need to stream a newly recorded map to the backend. Our subsequent publications, Paper III and Paper IV, introduce the frontend summarization, where subpar quality features are immediately discarded after the detection. The problem posed in the paper is closely related to [15], but instead of artificial markers we used natural features with lower repeatability.

Paper II

Marcin Dymczyk, Simon Lynen, Michael Bosse, Roland Siegwart, "Keep it brief: Scalable creation of compressed localization maps". In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015.

Context

The contributions presented in Paper I permitted our mapping system to maintain a constant size map even after collecting multiple mapping sessions. This led to a successful deployment of the framework on mobile devices [135] and mobile robots (Paper III) in an office environment to obtain a reliable map that contains persistent features of the environment. The reductions in storage, bandwidth and localization query time were significant and helped to reduce the computational requirements of the entire system.

The methodology, however, presents a few shortcomings. The heuristic feature selection process is relatively slow when applied to large-scale maps. Moreover, it requires tedious parameter tuning which is application-dependent and is not

intuitive to the users. Finally, extending it with a different set of feature selection criteria is not straightforward and requires a significant design overhaul. This led to researching more formal approaches to construct a *Summary Map*.

Contribution

The paper poses the backend map *summarization* as a binary optimization problem. Each 3D point has a corresponding binary variable that implies if we should retain it (if 1) or discard it (if 0). Our basic approach uses integer linear programming (ILP) problem definition, where each 3D point has an assigned scalar score in the score matrix. The score value corresponds to our perceived quality of the feature which may correspond, for example, to the number of observations. This value, however, can be easily tailored to the specific use-case. We then solve a problem related to the maximum coverage problem – we want a minimum set of 3D points that covers evenly the keyframes of the map and maximizes the total score. We also let users define additional constraints to the problem, such as the total number of features or specific coverage guarantees.

We have also introduced an extended version of this approach. It uses the integer quadratic programming (IQP), where we can additionally define pairwise scores for the 3D points. This enables us to model relationships between features, *e.g.* promote even distribution of features in the image plane. As a result, the geometric pose recovery (we use perspective-n-point by Kneip *et al.* [110]) is better constrained and outputs more accurate solutions.

The integer programming problems are NP-complete and the cost of summarizing might be prohibitive for large-scale maps. The paper introduces therefore a partitioning scheme that lets us solve smaller subproblems and only subsequently a larger global problem for further refinement. The partitioning is based on a graph-cut algorithm [106] applied to the covisibility graph and introduces minimal performance loss.

Overall, the paper improves on Paper I, making the *Summary Maps* even smaller in size. As a consequence, the correspondence search results in fewer outliers, making the entire place recognition system less computationally expensive – an important step towards lifelong deployment on mobile robots.

Interrelations

This paper introduced the optimization based map *summarization* that was later used in large-scale and long-term mapping applications, such as [26]. It also got integrated in the long-term mapping experiments (Paper III), where a wheeled robot kept revisiting a crowded office space daily. The optimization-based map summarization was applied in [59, 135] to reduce the size of the map and speed up the correspondence search problem when localizing. The implementation of the method is included within the open-source *maplab* framework (Paper V).

The methodology of this paper builds on top of [167] where 3D point clouds were reduced in a rigid and non-scalable fashion. Our approach was further extended

by Van Opdenbosch *et al.* [218], where the authors suggest not to constrain the optimization problem using the total number of features, but rather care about the number of bits required to encode the feature's descriptor. Additionally, they introduce efficient compression and coding of local and global descriptors to further reduce the bandwidth necessary for transferring the map.

A video demonstrating the map partitioning process and subsequent integer-based optimization to prune the 3D features and summarize the map is available at `https://youtu.be/noRXxuJtRGA`.

Paper III

Marcin Dymczyk, Thomas Schneider, Igor Gilitschenski, Roland Siegwart, Elena Stumm, "Erasing bad memories: Agent-side summarization for long-term mapping". In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016.

Context

Lifelong mapping of large environments, as used by various service and inspection robots or within autonomous driving, produces very large amounts of data, even if raw sensor measurements are discarded. In Paper I and Paper II, we address the problem of an unbounded growth of the map in the backend, where maps from all mapping sessions are merged and refined together. We have proven that it is possible to reduce the number of 3D landmarks in the map by over 90%, keep it constant over long timeframes and still retain good localization performance. At the same time we have also reduced the bandwidth when sending back the map from the server to the client that attempts to localize. However, the bandwidth in the opposite direction, when streaming a newly constructed map from the client to the server, remained unchanged.

We believe these transmissions can be reduced by selecting the features on the agent side just after detection, before sending them to the server. There are multiple factors that can tell us if a certain 3D landmark will be valuable in the future for relocalization. A careful statistical examination would point to the predictor variables particularly useful for this feature selection. We expect, for example, that better constrained features will have a more accurate 3D position and will be more commonly used for subsequent localization.

Contribution

In this paper, we perform a thorough statistical analysis of predictor variables that can be used to decide locally, without any past statistics, which features are worth keeping in the map for subsequent localization. Our framework automatically selects predictors out of numerous candidates that let us estimate the expected quality of a 3D landmark. Examples of such input variables include the track length, maximum disparity angle, Z-coordinate or the number of observations. We have

also included and extended the methodology presented in [85] to binary descriptors, where a random forest taking descriptors as input can also provide useful information to this feature quality regression problem. Such classification of descriptors is beneficial, as only certain appearance patterns are unique enough to be reliably matched when relocalizing. Overall, we propose a system that combines backend summarization algorithms with a newly developed frontend summarization and lets us reduce the data transmission in both directions.

The performance of the system was thoroughly evaluated on datasets collected using a wheeled robotic platform in a highly dynamic office environment. In total, over 30 datasets spanning over several months were used to perform the experiments. The datasets with features matched in a brute-force fashion provided the training data and feature quality labeling. We have set up a pipeline that combines the backend summarization from Paper II with the proposed frontend summarization and proved its usability. Additionally, we have also verified which frontend and backend map summarization ratios are a good compromise between the map compactness and the localization quality.

Interrelations

This paper extends the methodology of backend summarization presented in Paper I and Paper II. The newly introduced frontend summarization permits to summarize the maps while collecting data and before sending it to the mapping backend. This reduces the bandwidth from the client to the server, making the transfers in both directions bounded and complementing the previously described approaches.

This paper relies on a traditional statistical analysis of predictor variables to subselect the features locally, without a global map or past observation statistics. With the advent of deep learning, significant performance improvement may be expected by directly regressing the feature quality from the image patches. In the end, humans can also predict feature persistence just by understanding the semantics, *i.e.* if a point belongs to a static or a dynamic object. In Paper IV, we introduce a method to evaluate feature quality solely using the appearance.

Paper IV

Marcin Dymczyk, Elena Stumm, Juan Nieto, Roland Siegwart, Igor Gilitschenski, "Will it last? Learning stable features for long-term visual localization". In *3D Vision (3DV), 2016 Fourth International Conference on*, 2016.

Context

Selecting 3D features in the frontend, as introduced in Paper III, is particularly advantageous in the crowd-sourced mapping scenarios with computationally constrained devices. Being able to discard some portion of features directly after detection limits the footprint of feature matching and subsequent mapping. Additionally, it also reduces the amount of data collected and transferred when up-

loading a new map to the backend. That is particularly important for consumer devices, where this reduction lowers both the power consumption and the mobile data usage.

Compared to the traditional statistical methods, modern deep learning techniques taking images as input promise even better prediction results of the feature matchability. Intuitively, for most of the features humans can tell whether they belong to a persistent or a dynamic object in the environment. This property, on the other hand, is very hard to model using conventional methods. We therefore decided to apply a CNN to the raw image information in order to regress the feature matchability. Our aim was to improve the performance of the method in Paper III and move beyond manually engineered predictor variables. We also expect, given enough training data, that the learning-based method will generalize better to unseen environments.

Contribution

In this paper, we propose a method using a CNN to predict feature matchability based on the appearance information. First, to train a network, we introduce the procedure to construct a training set out of multiple mapping sessions collected manually using a mobile device [118]. Then, a mapping framework (Paper V) is used to merge all the maps together, exhaustively match the features and find out which of them are particularly reliable for long-term mapping and localization.

Next, we use the network [116] as a classifier to evaluate the local features. We also consider several summarization ratios to understand what reduction rate is optimal for target applications. Finally, we verify how the resulting maps perform in the long-term relocalization scenarios.

Given a monocular setup, we also propose to use structure-from-motion filtered depth maps [172, 211] to further enhance the results. We believe that the shape of the structure around the features is correlated with the feature matching quality. In particular, occlusion corners (keypoints located on the structure edges) might have limited repeatability and look very different depending on the exact viewpoint. Those hypotheses were confirmed by the evaluations presented in the paper.

The experiments used over 25 datasets collected manually using a mobile device in a highly-dynamic environment. Comparisons with state-of-the-art approach, presented in [85], demonstrated superior performance of our method both within the same environment and when generalizing to another area. Our algorithm allowed to discard about 50% of detected keypoints with nearly zero localization performance loss.

Interrelations

This paper introduces a learning-based approach that generalizes better than our previous method presented in Paper III. Instead of using conventional predictors, mostly based on the map geometry and covisibility graph, we regress the feature matchability directly from the appearance. We therefore avoid using any hand-

engineered features and make it easier to adapt the system to any environment.

The contribution might also be seen as an improvement over the state-of-the-art predicting matchability method [85], where using random forests to classify SIFT descriptors [132] was proposed. We believe that the gain in performance we report is a result of using a more powerful classifier on the raw image patches, instead of a hand-crafted descriptor pattern with a very limited size.

The preprocessing of the training set and exhaustive matching of 3D features was implemented within the *maplab* framework (Paper V).

Paper V

Thomas Schneider*, Marcin Dymczyk*, Marius Fehr*, Kevin Egger, Simon Lynen, Igor Gilitschenski, Roland Siegwart (*contributed equally), "maplab: An open framework for research in visual-inertial mapping and localization". In *IEEE Robotics and Automation Letters 3.3 (2018): 1418-1425*, 2018.

Context

Various research directions within the Autonomous Systems Lab and a wide variety of robotic platforms [27, 52, 67, 92, 162] motivated an effort towards a common visual mapping and localization framework. We believe that robotic systems we deploy, even though their environments differ significantly, can actually share core mapping algorithms. Understanding that it is worth to reuse system components and map representations, we have developed a framework that offers robust basic tools, flexible interfaces and permits to leverage those existing implementations. Furthermore, as we were also interested in the basic research related to computer vision, we wanted the system to expose low-level sensor measurements and mapping data.

To the best of our knowledge, at the time of development there was no open-source framework that combined reliable odometry, fast loop closure running on lightweight computing platforms with an option to perform multi-session, large-scale mapping. Additionally, as we wanted the system to become a research testbed, we needed a tool that offers a flexible user interface, automating our experiments, parameter sweeps and facilitating introspection and troubleshooting. Finally, a large number of projects depending on the system required modularity, where it is easy to manage and swap out specific system components.

Contribution

The framework can be perceived as a by-product of the research performed by the contributing authors. The organic development and continuous extension of its capabilities made the software architecture particularly suitable for research and prototype development. The visual-inertial map structure is both efficient and enables various algorithms to access the low-level data. It also offers many

high-level interfaces that permit to prototype SLAM-related algorithms in a time-efficient manner.

The system also integrates several standard algorithms with state-of-the-art performance. The maps are constructed using sparse feature tracking and estimates from a visual-inertial odometry framework ROVIO [20]. Maps can be loopclosed using binary descriptors [121] and the user's pose can be estimated using an inverted multi-index that scales well with the size of the map [135]. By using efficient retrieval algorithms, we are able to perform global localization even multiple times a second on our robotic platforms. The framework also offers a refinement step, where a visual-inertial weighted least-squares optimization yields an accurate batch solution for the map structure. Finally, we also include various dense reconstruction tools [60], that are particularly useful for subsequent navigation, inspection and path planning.

maplab also aims at becoming a standard research platform for visual and visual-inertial mapping. The development of an algorithm is simplified as most basic search, retrieval, geometry and query algorithms are already available. Consequently, prototyping a new method resembles writing pseudo-code. The system offers several tools to verify the results of own experiments. It is possible, for example, to use built-in data processing tools that can import, export and directly plot various variables. To understand the map structure better, it is also possible to visualize 3D information in RViz, a visualization tool of Robot Operating Systems (ROS) [175].

The system has been extensively tested on various platforms, including mobile consumer devices, wheeled robots, flying robots, legged robots and robotic arms. This wide variety of applications led to a development of a plugin system, that facilitates swapping out certain components or adding own extensions (*e.g.* a map quality evaluation module [146]) that can be easily reused by others. The inputs and outputs of the system are compatible with ROS, which is supported by common robotic platforms. The framework also supplies interfaces to other tools that are commonly used, such as Kalibr [177] for sensor calibration, Voxblox for voxel-based mapping [164] and CMVS [70] for patch-based dense reconstruction.

Interrelations

The *maplab* framework integrates the contributions of papers being part of this thesis: Paper I, Paper II, Paper III and Paper VI. By design, it is suited for large-scale multi-session experiments [54]. We have also used it for merging multiple visual maps together which was necessary to create the training sets for learning based approaches: for the CNNs introduced in Paper IV and for the boosting framework in Paper VII.

Additionally, the framework was used to perform research and experiments that led to several co-authored publications. It acted as a testbed for map quality evaluation [146], autonomous car SLAM systems [26], robust dense reconstruction [60], topological mapping [19], collaborative navigation of flying and walking robots [59] and hierarchical localization [184].

A video demonstrating the capabilities of *maplab*, particularly the multi-session and large-scale mapping scenarios is available at `https://youtu.be/CrfG4v25B8k`.

2.2 Part B: Robust Place Recognition

In this section, we present contributions related to the robust place recognition capability. We believe it is an essential feature for reliable visual mapping systems. Intuitively, for humans the benefits of realizing that they have seen a certain place before help them to navigate, save time or perform various tasks in the area.

Similarly, localization within a prior map brings several benefits for the robotic applications. Knowing an accurate pose within a well-mapped environment helps to navigate and perform manipulation tasks. This becomes particularly useful for inspection or industrial robotics. The ability to close loops also helps to reduce the drift of open-loop odometry estimators. Finally, understanding spatial relationships across visits or agents leads to multi-agent collaboration or persistence useful for AR and VR experiences.

Place recognition, however, still remains a challenging task. We want to improve its robustness in case of difficult illumination conditions, presence of dynamic objects and environments with self-similar structures. These are the scenarios where compact hand-crafted descriptors [3, 121, 181], commonly used on platforms with limited resources, fail to establish correspondences and localize.

Paper VI

Antonio Loquercio, Marcin Dymczyk, Bernhard Zeisl, Simon Lynen, Igor Gilitschenski, Roland Siegwart, "Efficient Descriptor Learning for Large Scale Localization". In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2017.

Context

Most of visual mapping algorithms used in robotics rely on hand-crafted descriptors [3, 121], primarily due to the strict computational constraints. It is questionable whether the pattern used to compute descriptor values is actually optimal. Furthermore, the receptive field is rather small (for BRISK [121] radius of 16 pixels) which leads to poor discriminability. Ultimately, many corners and keypoints of man-made structures look nearly identical within this local area (*e.g.* identical door handles in the entire building).

Another motivation is related to projection of binary features to the Euclidean space. Generally, we project binary features to make it easier to perform large nearest neighbor searches [134, 215]. Projection using linear PCA [102] might be suboptimal and non-linear methods [188] pose a large computational burden. There exists, therefore, a strong motivation to improve binary feature matching using alternative projection methods that would be still compatible with a nearest neighbor search in the Euclidean space.

Contribution

The paper researches various approaches to perform linear projection of binary descriptors to the Euclidean space. The projected descriptors have several advantages, such as easy, well-researched nearest neighbor search and reduced size. Furthermore, the projection can also improve the retrieval performance when compared to a Hamming distance search in all original dimensions.

We propose a trainable linear projection method that is both easy to train and straightforward to deploy in the existing systems. The feature matching results of our method improve even over the compute-heavy non-linear kernel-PCA [188] descriptors of the same dimensionality. Additionally, we have researched approaches to include the visual context within the projected descriptors, as in [195]. We propose to use an embedding of a larger image patch around the local features and combine it with a hand-crafted descriptor in a common, learned projection step. This increases the discriminability of our augmented descriptors, particularly for features that are affected by visual aliasing.

We have thoroughly evaluated the proposed methods on indoor datasets from a museum and office environments. The proposed linear projection method significantly improves the quality of descriptor matching and brings true positive matches closer in the projected descriptor space. The results further improve if we augment the descriptors with context embeddings. Our descriptors yield over 40% higher precision values at given recall when compared to the BRISK-based [121] baseline implementation [187], which translates to improved place recognition.

Interrelations

The paper is a follow-up of previous contributions [77, 134] that strive to improve the binary descriptor retrieval and resulting place recognition performance. Here, we address the problem of efficient binary descriptor retrieval [215]. Firstly, we look for an optimal, learned projection method for descriptors, similar to [171]. We also revisit the methodology presented in [195]. Because the proposed method uses wider visual context, it can be considered as an approach similar to [184], where we explicitly separate the search in the domain of visual context from the subsequent local feature matching. This contribution follows the real-time goals of the *maplab* system (Paper V).

The idea of using feature's visual neighborhood to robustify feature matching was also revisited by us later. In this contribution, we model the visual neighborhood as a rectangle of a fixed size while in Paper VII we demonstrate how more distant context relationships in the image plane improve the matching quality.

Paper VII

Marcin Dymczyk, Igor Gilitschenski, Juan Nieto, Simon Lynen, Bernhard Zeisl, Roland Siegwart, "LandmarkBoost: Efficient Visual Context Classifiers for Robust Localization". In *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*, 2018.

Context

The methodology presented in Paper VI, while efficient, demonstrated certain disadvantages. Visual context defined as a rigid, immutable neighborhood of a fixed size tends to fail if not carefully tuned for a specific environment, taking into account texture, distance of the camera from 3D structure and object sizes. We have therefore decided to investigate more flexible ways to augment the 2D-3D correspondence search with visual context and geometric relationships between groups of features. We expect such configurations of multiple features to be more discriminative and less sensitive to visual aliasing in the environment. We consider self-supervised discovery of informative context relationships particularly tempting as it does not need to be re-tuned for each use-case.

Our target application was crowd-sourced mapping. Due to the privacy concerns, our algorithms could only depend on local hand-crafted features, but never use or store images, image regions or rich global embeddings. Thus, we are discarding the full image information directly after feature detection and extraction. This design is compatible with our hardware, where local descriptors are computed on a dedicated co-processor, without costly processing of raw images on a main central processing unit (CPU). Our system architecture permits for expensive training for each map tile in the backend, where the available computational power is large. The frontend device, on the other hand, handles the lightweight classifier establishing reliable 2D-3D correspondences in real-time.

Contribution

The paper introduces a novel method that poses the 2D-3D feature matching problem [185] as classification, instead of a commonly-used nearest neighbor search. It revisits the idea of Joint-Boosting [212] and applies it to localization by classifying 2D observations to the classes corresponding to 3D landmarks in the map. The boosting algorithm trains an efficient multiclass classifier that is particularly fast to evaluate. The specific Joint-Boosting architecture shares the weak learners among classes, which significantly reduces the final classifier complexity and speeds up its execution. We have further improved the scalability of the algorithm, enabling it to scale to tens of thousands of classes instead of original 21.

Our algorithm performs matching not only using local hand-crafted binary descriptors [3], but also by aggregating the features that lie within the visual context regions. The raw image information is not used. This time, we formulate the context as regions in the image plane, with random size and position relative to the keypoint location. We generate a large number of regions and describe them using standard techniques, such as bag-of-words [41]. The boosting algorithm discovers useful regions and discards the non-informative ones in a self-supervised setting.

The paper provides thorough evaluation on indoor and outdoor datasets, including large scale Zürich Old Town with over 16km of trajectories [54]. We show that LandmarkBoost achieves superior place recognition performance compared to raw nearest neighbor search of the same feature type [187] while retaining real-time capabilities when localizing.

Interrelations

This paper fits the overall picture that we have drafted in Section 1.1, where several agents localize within a prior map that is maintained and provided by the mapping backend. This paradigm also matches the structure of the *maplab* system (Paper V), where the backend is a long-term large-scale multi-session mapping system. Given this architecture, we expect the queries to require minimal processing on lightweight hardware, but we assume the backend to offer significant computational power.

This contribution presents an alternative to Paper VI that proposed to augment binary features with visual context to improve their discriminative power. We follow up on this idea, but implement the feature embedding and matching in a data-driven fashion. Our methodology also demonstrates how *maplab* can be utilized to provide refined visual maps to build training sets for machine learning. Its multi-session and large-scale capabilities permit to embed enough variability to develop algorithms for robust localization.

The idea of exploring the relationships and patterns between features was earlier proposed in [231] and later in [234]. We extend those methods by demonstrating that it is possible to capture even very distant relationships in the image plane. In contrast to the past approaches that encoded the relationships as descriptors, our method adapts a boosting classifier presented in [213] and originally used for object classification to match the query frame to the database.

2.3 List of Publications

Over the course of the author's doctoral studies the following peer-reviewed publications were published. The list is sorted by first author's name and the year of publishing.

- F. Blöchliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. In *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018

- M. Bürki, M. Dymczyk, I. Gilitschenski, C. Cadena, R. Siegwart, and J. Nieto. Map management for efficient long-term visual localization in outdoor environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 682–688. IEEE, 2018

- T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Y. Siegwart. Map api-scalable decentralized map building for robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 6241–6247. IEEE, 2015

- M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale. The gist of maps-summarizing experience for lifelong localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 2767–2773. IEEE, 2015

- M. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart. Keep it brief: Scalable creation of compressed localization maps. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2536–2542. IEEE, 2015
- M. Dymczyk, T. Schneider, I. Gilitschenski, R. Siegwart, and E. Stumm. Erasing bad memories: Agent-side summarization for long-term mapping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4572–4579. IEEE, 2016
- M. Dymczyk, E. Stumm, J. Nieto, R. Siegwart, and I. Gilitschenski. Will it last? learning stable features for long-term visual localization. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 572–581. IEEE, 2016
- M. Dymczyk, I. Gilitschenski, J. Nieto, S. Lynen, B. Zeisl, and R. Siegwart. Landmarkboost: Efficient visual context classifiers for robust localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2018
- P. Fankhauser, M. Bloesch, P. Krüsi, R. Diethelm, M. Wermelinger, T. Schneider, M. Dymczyk, M. Hutter, and R. Siegwart. Collaborative navigation for flying and walking robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2859–2866. IEEE, 2016
- M. Fehr, M. Dymczyk, S. Lynen, and R. Siegwart. Reshaping our model of the world over time. In *IEEE Int. Conf. on Robotics and Automation*, pages 2449–2455. IEEE, 2016
- T. Hinzmann, T. Schneider, M. Dymczyk, A. Schaffner, S. Lynen, R. Siegwart, and I. Gilitschenski. Monocular visual-inertial slam for fixed-wing uavs using sliding window based nonlinear optimization. In *IEEE International Symposium On Visual Computing*, pages 569–581. Springer, 2016
- T. Hinzmann, T. Schneider, M. Dymczyk, A. Melzer, T. Mantel, R. Siegwart, and I. Gilitschenski. Robust map generation for fixed-wing uavs with low-cost highly-oblique monocular cameras. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3261–3268. IEEE, 2016
- A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart. Efficient descriptor learning for large scale localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 3170–3177. IEEE, 2017
- H. Merzić, E. Stumm, M. Dymczyk, R. Siegwart, and I. Gilitschenski. Map quality evaluation for visual localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 3200–3206. IEEE, 2017
- P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, 2018

- T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 3(3): 1418–1425, 2018

The doctoral studies also resulted in several workshop papers and presentations.

- M. Fehr, T. Schneider, M. Dymczyk, J. Sturm, and R. Siegwart. Visual-inertial teach and repeat for aerial inspection. *arXiv preprint arXiv:1803.09650*, 2018

- M. Dymczyk, I. Gilitschenski, R. Siegwart, and E. Stumm. Map summarization for tractable lifelong mapping. In *RSS Workshop on Geometry and Beyond*, 2016

- M. Dymczyk, M. Fehr, T. Schneider, and R. Siegwart. Long-term large-scale mapping and localization using maplab. *arXiv preprint arXiv:1805.10994*, 2018

- G. Zhou, B. Bescos, M. Dymczyk, M. Pfeiffer, J. Neira, and R. Siegwart. Dynamic objects segmentation for visual localization in urban environments. *arXiv preprint arXiv:1807.02996*, 2018

2.4 List of Supervised Students

During the author's doctoral studies a significant effort was spent on supervising student projects. Below, all supervised students and projects are listed. For projects that resulted in a publication a citation is provided.

Master Thesis

Master student, 6 months full time

- Jari Kruth (Spring 2015): "Design and Implementation of a distributed Vision-based parking Spot Management System"

- Nicolas Degen (Spring 2015): "Local Optimisation for Large Scale, Offline Visual-Inertial Maps"

- Marius Fehr (Spring 2015): "Reshaping our model of the world over time" [60]

- Partha Gosh (Spring 2015): "Improving Scalability of Bundle Adjustment"

- Kevin Egger (Spring 2016): "Vision Based Person Following for a Quadrupedal Robot"

- Antonio Loquercio (Spring 2016): "Efficient Descriptor Learning for Large Scale Localization" [130]

- Jonas Eichenberger (Spring 2016): "Robust visual SLAM for omnidirectional cameras"
- Alessandro Simovic (Autumn 2016): "Transparent and verifiable knowledge manipulation and computation outsourcing"
- Fabian Blöchliger (Spring 2017): "Topomap: Topological mapping and navigation based on visual slam maps" [19]
- Dimitris Gryparis (Spring 2017): "Generative methods to place categorization"
- Flavio Eiholzer (Autumn 2017): "Learning descriptors for visual sequence matching"
- Thomas Eppenberger (Autumn 2018): "Dynamic obstacle avoidance using a stereovision sensor"
- Benjamin Hahn (Autumn 2018): "Segment-based relocalization using stereo cameras"
- Sebastian Ratz (Autumn 2018): "Vision-LiDAR localization"
- Josua Boegli (Autumn 2018): "Development of an online SLAM System for maplab"
- Hendrikx Hubertus (Autumn 2018): "Combining vision and LiDAR for robust SLAM"

Semester Thesis

Master student, 3-4 months part time

- Antonio Loquercio (Spring 2015): "Image-based localization"
- David Hoeller (Spring 2015): "Integration of Google Street View into Multi-agent Mapping"
- Andreas Pfrunder (Spring 2015): "Map Tile Streaming for Indoor Localization"
- Ralph Aeschimann (Spring 2015): "Deformation of Large-Scale Dense Point Clouds"
- Marius Huber (Autumn 2015): "Deep Learning for Map Summarization"
- Mathias Gehrig (Autumn 2015): "Daily Autonomous Mapping of an Indoor Environment"

- Hamza Merzić (Spring 2016): "Map quality evaluation for visual localization" [146]
- Nektarios Lianos (Spring 2016): "Feature Quality Learning for Long-Term Localization"
- Yawei Ye (Autumn 2016): "Feasibility Study of GNSS Outlier Rejection Using Additional Pose Information"
- Stefanie Nacht (Spring 2016): "Predicting reliable features for long-term localization from semantic information"
- Flavio Eiholzer (Spring 2017): "Semantic mapping for VI maps: Enriching sparse maps with semantic information"
- Wojciech Witoń (Spring 2017): "Deep Image Retrieval on a portable GPU platform"
- Guoxiang Zhou (Spring 2018): "Dynamic Objects Segmentation for Visual Localization in Urban Environments" [235]

"Perception and Learning for Robotics" course project

2 Master students, 3-4 months part time

- Paul-Edouard Sarlin, Frédéric Debraine (Spring 2018): "Leveraging Deep Visual Descriptors for Hierarchical Efficient Localization" [184]

2.5 List of Open-Source Software

Throughout the doctoral studies, the author actively developed and co-developed software packages used for visual place recognition and mapping. The most notable open-source contributions include:

- `maplab` – a visual-inertial mapping and localization system [187], used as testbed for most of the author's publications, https://github.com/ethz-asl/maplab/,
- `aslam_cv2` – a robotics-oriented computer vision library that includes camera and distortion models, multiview geometry, SLAM data structures and feature tracking, https://github.com/ethz-asl/aslam_cv2/,
- `hierarchical_loc` – a hierarchical localization package that uses efficient, deep-learned global image descriptors to improve the matching process using classical local features [184], https://github.com/ethz-asl/hierarchical_loc/.

Chapter 3

Conclusion and Outlook

This doctoral thesis researched the topics related to vision-based mapping and localization, focusing on large-scale and long-term scenarios. We would like to provide a brief summary of the major findings of the studies and draft our views on the future developments. We believe important research directions are emerging right now in the topics tightly related to both parts of the dissertation.

3.1 Part A: Efficient Vision-Based Mapping

In the first part of the thesis, we have presented approaches that address the challenges of long-term and large-scale mapping in dynamic environments. We have introduced map summarization techniques that efficiently reduce the size of sparse, vision-based maps. We summarize the maps both in the frontend, by discarding features directly after detection, and in the backend, based on geometry and past matching statistics. The proposed methods permit to significantly lower the map size, often by more than 90%, without compromising the localization performance. The resulting representation, a *Summary Map*, relaxes the requirements on bandwidth, storage and computational power. The proposed methods successfully resolve the severe limitations of multi-session large-scale SLAM, making the maps compact and easy to deploy.

Furthermore, we have introduced an open-source visual-inertial mapping system, *maplab*. The framework is meant to be used as a versatile solution for different types of platforms, from flying and wheeled robots to handheld devices. The system integrates all core functionalities necessary to construct maps and relocalize within them, including odometry, loop closure, batch LS optimization and introspection tools. The algorithms are extensively tested and were successfully deployed on numerous platforms in varying environments. Because of standardized interfaces agnostic to specific sensor configurations, we were also able to demonstrate collaborative robotic scenarios. Besides, the *maplab* framework was developed as a

research testbed. We have constructed the system following the modular principle, so that new developments can easily be integrated into the existing codebase. The prototyping and research work are further facilitated by a collection of stable core algorithms and a broad range of map queries.

We believe our research and development efforts demonstrate the value of the visual-inertial mapping and localization technology in robotics. These sensor modalities complement each other and are more versatile than other commonly used sensing modalities. It is clear, however, particularly given the recent progress within machine learning, that mapping systems will further evolve and improve. We perceive the following directions as the possible next developments in the field:

Topological Mapping Revisited The problem of scalability of SLAM has been researched for almost two decades now. A particular attention was dedicated to sub-mapping [120, 209], where a set of local maps is maintained instead of a single, global metric representation. Subsequent works further developed this idea by proving its favorable properties [119] and introducing methods for estimation [228], localization [10], efficient large-scale loopclosure [58], fusing other sensory information [81, 158] and building hybrid metrical-topological representations [21]. These approaches, however, are not commonly leveraged by the contemporary mapping and localization systems [122, 174, 187], which struggle with city-scale models. We believe that sub-mapping could solve the scalability issues, lift the memory constraints and prohibitive cost of large scale batch optimization. The compact local maps can be easily transferred and used for localization on-demand. This approach interfaces seamlessly with 3D reconstruction [150] and planning algorithms and helps to navigate even in very complex environments. As shown in [19], planning a path in a resulting topological representation is reliable and efficient. The easy-to-handle local sub-maps are then used to localize, correct for drift and perform local obstacle avoidance. In a nutshell, while a global metric map remains a tempting concept, it is often not strictly necessary and very expensive to maintain.

Multi-modal Perception The visual-inertial sensor suite has found a wide variety of applications. In certain cases, however, vision-based estimation and mapping pipelines fail. This might have various reasons, *e.g.* motion blur, inconsistent tracking, lack of texture or poor illumination. Additionally, the state cannot be propagated using just inertial sensors for extended periods of time. Consequently, failure of the visual pipeline means a critical failure of the entire system. We therefore believe additional sensor modalities will be used to mitigate this threat and to extend a visual-inertial setup. For example, using LiDARs helps in low-light and low-texture scenarios, where vision itself is not providing enough useful measurements. The development of solid-state LiDARs is expected to further reduce the manufacturing cost, weight and size of laser-based rangefinders which will contribute to their widespread use. Similarly, event cameras may solve the pitfalls of visual-based estima-

tion in the presence of dynamic maneuvers [176]. They capture single pixel events instead of complete frames at a very high frequency and do not suffer from motion blur. The unconventional visual sensing, *e.g.* the SCAMP cameras [30], open up a new field of high-frequency hardware-accelerated computing directly on the image sensor chip. Those bleeding edge approaches might change the way we implement low-level computer vision algorithms, such as feature detection and tracking. Another promising sensor type is a millimeter-wave radar, as it is insensitive to appearance changes (contrary to cameras) or difficult atmospheric conditions such as heavy rain (contrary to LiDARs). Already now, radar odometry permits to achieve accurate ego-motion estimates and remains reliable when classic methods fail [34].

Semantic Understanding and Deep Learning Understanding the semantics of the scene will benefit the mapping process and permit to build smaller and more persistent maps. The hand-crafted features and algorithms we are using right now will shift towards semantically labeled primitives and objects [69], and use them for mapping [141], as they are more reliable to (re-)detect and less prone to changes in viewpoint and illumination. Semantics also provide richer and more compact descriptors. We believe that higher-level representations will replace the sparse or dense features we depend on in most of the contemporary systems, as it was already proposed for LiDAR data [48]. Understanding the semantics also means removing the dynamic objects from the scene that most probably will not reappear again [18]. Intuitively, when memorizing a place humans do not pay attention to the specific cars or passersby. Rejecting the dynamic parts of the scene also helps estimation, as large moving objects might provide inconsistent visual measurements and negatively effect the visual odometry estimates [12]. Finally, semantics and adversarial methods are a key to appearance transfer between various environmental conditions [173] and to building a single canonical appearance representation in the map [40]. An interesting approach is presented in [117], where the appearance transfer is further augmented with learned features of the generative adversarial network (GAN) discriminator for robust place recognition.

Towards End-to-End Mapping All of the aforementioned algorithms replace a single component or blocks of components of the conventional mapping and localization system. We would like to pose a question if it would be possible to replace the entire system and directly regress the agent's trajectory given a sequence of sensor measurements. Recent publications demonstrate initial promising results in this direction [39, 124, 224]. State-of-the-art methods still struggle, however, to generalize well to various environments and deliver metric accuracy on par with model-based approaches. Furthermore, a vast mount of training data is necessary to obtain those less-than-satisfactory results. The advances in the field of machine learning will at least partially address those shortcomings. While it is unclear whether data-driven solu-

tions will indeed outperform the conventional methods, as they did in object classification [182] or semantic segmentation [9], we believe that hybrid geometric-learning systems might soon represent state-of-the-art.

3.2 Part B: Robust Place Recognition

In the second part of the thesis, we have addressed the core problem of visual place recognition: establishing correspondences between a query frame and a prior map. Our contributions enhance place recognition performance in challenging environments and enable robots to localize reliably, estimate inter-robot transformations and reach desired global waypoints. We propose two methods that are drop-in replacements for systems using conventional hand-crafted descriptors. Both methods use machine learning to improve the retrieval of database features that are matching the query. Additionally, both methods augment local features with visual context, but define it in two diverse ways. Using context improves the retrieval performance and leads to more 2D-3D correspondences, fewer outliers and, consequently, better localization accuracy and robustness.

The first method introduces a novel linear projection method of binary descriptors that replaces the one used in [134, 187]. We can perceive it as a drop-in improvement with zero added runtime cost. Besides that, the paper introduces an approach that augments the projected descriptors with visual context, defined as a fixed region with a CNN-based embedding. The contextual information helps to match features even when the hand-crafted descriptor pattern is not discriminative enough. The improvement is particularly striking when dramatic changes in viewpoint and illumination are present.

The second contribution lifts the limitations of the rigid context definition of the first method. Our motivation was to enable capturing additional, more distant relationships between features in the image plane. Our boosting-based algorithm mines for informative geometric feature patterns in the entire image plane and embeds them into weak learners. The core of the method is a classifier that performs robust 2D-3D matching and is inexpensive to evaluate.

The contributions of this thesis improve state-of-the-art in efficient place recognition in challenging environments. We have achieved significant gains by using machine learning and data-driven approaches while retaining real-time capabilities of the system. We expect the place recognition to further benefit from exploring the following research directions:

Hierarchical place recognition Current methods try to solve the 6-DoF place recognition problem in a single-step, by matching query image to the features that belong to a global, possibly large, prior map. We believe this procedure should be decomposed into estimation of a coarse location prior, *e.g.* using image retrieval or GPS when available. Given a prior, the accurate 6-DoF pose should be then estimated based on matching to the local structure only. This approach addresses visual aliasing issues and resembles what humans do to maintain spatial awareness. This method, using off-the-shelf descriptors,

is investigated in [184] and the benefits of this coarse-to-fine paradigm are demonstrated.

Semantics Understanding the scene semantics will bring several benefits for the place recognition task. First of all, semantics are stable in the presence of illumination or viewpoint variations compared to the raw appearance [200]. This leads to more robust scene description and, consequently, more reliable matching of the query frame to the prior model [76]. Some recent works already explore the semantics by relying on off-the-shelf scene segmentation and object detection algorithms [75, 189]. Secondly, as mentioned above in Section 3.1, semantics also bring a possibility to reject dynamic objects and grasp better the relationships between elements of the scene. For example, it would be beneficial to understand the effects of foliage occluding parts of the structure, or reflections in puddles.

Learned features Robotic systems still primarily rely on weak, hand-crafted descriptors for place recognition. Those features work reliably in controlled environments, but often fail in the presence of appearance and illumination changes. Often, it is not possible to match images taken across sunny and cloudy conditions. Learning compact feature descriptors might be an answer to those issues. Already now, approaches such as [44, 88] deliver significant performance gains, particularly for challenging day-night matching. The future development of deep learning and more comprehensive training datasets are expected to further boost performance of similar methods.

End-to-end Localization The limits of deep learning and the optimal split between traditional model-based approaches and learned "black boxes" are a common discussion topic in the robotics community. Deep learning becomes more and more ubiquitous and in certain applications end-to-end approaches demonstrate significant benefits. There have been some attempts to take advantage of these advancements for robust place recognition, *e.g.* in [107] for single images and in [38] for sequences. The performance of those approaches remains, however, inferior to the conventional methods relying on sparse, often learned, features. We believe that in the long run, end-to-end place recognition might demonstrate a significant edge, particularly where a combination of broad visual context, unique local features and semantics is necessary to reliably estimate an accurate pose.

Part A

EFFICIENT VISION-BASED MAPPING

Paper

The gist of maps-summarizing experience for lifelong localization

Marcin Dymczyk, Simon Lynen, Titus Cieslewski, Michael Bosse, Roland Siegwart, Paul Furgale

Abstract

Robust, scalable place recognition is a core competency for many robotic applications. However, when revisiting places over and over, many state-of-the-art approaches exhibit reduced performance in terms of computation and memory complexity and in terms of accuracy. For successful deployment of robots over long time scales, we must develop algorithms that get better with repeated visits to the same environment, while still working within a fixed computational budget.

This paper presents and evaluates an algorithm that alternates between online place recognition and offline map maintenance with the goal of producing the best performance with a fixed map size. At the core of the algorithm is the concept of a Summary Map, a reduced map representation that includes only the landmarks that are deemed most useful for place recognition. To assign landmarks to the map, we use a scoring function that ranks the utility of each landmark and a sampling policy that selects the landmarks for each place. The Summary Map can then be used by any descriptor-based inference method for constant-complexity online place recognition. We evaluate a number of scoring functions and sampling policies and show that it is possible to build and maintain maps of a constant size and that place-recognition performance improves over multiple visits.

Published in:
Robotics and Automation (ICRA), 2015 IEEE International Conference on, 2015
DOI: 10.1109/ICRA.2015.7139575

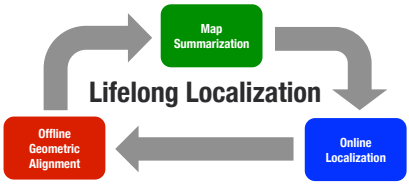


Figure 4.1: Summarizing experience for lifelong localization: Online localization provides a stream of new experiences that contain new but also redundant information. We propose a combination of scoring functions and sampling policies that can identify and select relevant data to be used to produce a Summary Map. This Summary Map contains the gist of previous experiences to unlock the best performance for online place recognition and localization.

1 Introduction

Robust vision-based localization based on pre-built maps forms the backbone of modern navigation algorithms both for robotics and mobile device applications. As demonstrated in many studies of visual navigation [68, 153], a static map is insufficient for lifelong localization given the *continuous* change of scene appearance around us. We believe that the stream of observations provided by the localizing agents should be used to *continuously* update, improve and enhance the map for reliable localization over longer periods.

In addition to incorporating environmental changes, repeated visits to the same environment should increase localization robustness and quality by providing statistics about which map data is most reliable. While we would like to be able to store and localize against a large library of previous experiences of a place (as in [35]), in practice, an online system will be subject to memory, computation, and bandwidth constraints on map data. Consequently, we need to develop mapping and localization systems that fluidly incorporate new data and exhibit better performance with every visit to the same place, while using only a fixed memory and computational budget.

In this paper, we apply these general concepts to the problem of *metric* place recognition. In metric place recognition, the system recovers the pose of the agent with respect to a coordinate frame within our map. This is in contrast to *topological* place recognition systems that simply identify previously visited places. This difference guides our system design as we have to take care that the geometry of landmarks in the map is able to constrain the agent's pose when revisiting places.

This paper presents a methodology for *recursive, continuous map summarization* that reinforces stable landmarks within the agent's map *and* adapts to newly discovered or changing aspects of the environment. After a run of online localization, we incorporate the new data to produce a *Summary Map* based on running

statistics updated after every visit. In the most general case, we apply a *scoring function* for landmark and trajectory segments to evaluate their usefulness for localization, and then use a *sampling strategy* to select the landmarks that will be retained for online operation. New parts of the map which are not reobservations of existing structure are saved on a "waiting list" to give them a chance to be confirmed by additional evidence. This process allows us to distinguish noise (single observation) from persistent changes to the environment (multiple observations). The fractions of the map that are found either to be redundant with other observations or never re-observed are discarded. Careful design of both the scoring and the sampling methods makes it possible to remove noise or outdated data from the map while simultaneously accommodating to changes in the environment. The resulting algorithm repeatedly integrates new observations into the map, while maintaining an overall map size constant with respect to the area covered. Because only the most reliable information is incorporated into the *Summary Map*, we show that the overall localization quality improves as we incorporate new data, despite the fact that the map size remains constant.

The contributions of this paper are as follows:

- we present the general methodology for producing *Summary Maps* as a solution for lifelong localization;
- we apply this methodology to the specific problem of metric place recognition;
- we evaluate a number of scoring functions and sampling strategies to determine the combination that produces the best performance given a fixed map size;
- we demonstrate the performance of the algorithm on 10 datasets gathered over 3 days in a dynamic, frequently changing environment (crowded university building); and
- we show that, using a suitable choice of scoring function and sampling strategy, we can produce a metric place recognition system that exhibits enhanced performance as an agent revisits an environment multiple times, while maintaining a constant size map.

For inference, we use the Placeless Place Recognition system of Lynen *et al.* as it was recently shown to be superior to other state-of-the-art approaches [134]. However, we believe that our results could be applied to any voting [95] or bag-of-words [198] approach based on descriptors attached to landmarks.

2 Related Work

This review of related work will focus primarily on advances in lifelong learning for vision-based mapping, localization, and place-recognition. In this paper we focus on long-term visual navigation where reliable data-association is the most

challenging question and leave the related research of optimal beacon placement given perfect data-association [4, 15, 222] aside.

One closely related research track is focused on continuous simultaneous localization and mapping (SLAM), and methods for updating the SLAM map over repeated visits to the same environment. Konolige *et al.* describe a system that combines metric SLAM with bag-of-words-based place recognition to create an anytime mapping system based on stereo keyframes [113]. The system maintains a constant map size for a particular environment by marginalizing out keyframes to maintain a uniform density [2]. Johannsson et al. take a similar approach but sidestep the marginalization and simply discard redundant views [99]. Similar approaches that seek to maintain a constant pose graph density have also been proposed for 2D [115] and 3D [31] laser data as well as for general SLAM graphs [91]. However, the main focus of all of these approaches is to keep the offline map optimization tractable during continuous operation. Our approach also considers the complexity of offline optimization, but this is not the primary focus. Instead we concentrate on increasing the success of place recognition, while keeping the number of landmarks in the map fixed. Furthermore, these approaches aggregate sensor data into keyframes or scans, removing all of this data when it is deemed redundant. In contrast, our *Summary Map* method prunes data at the landmark level; this fine-grained approach allows images that contain mostly ephemeral data to still contribute to the statistics about static parts of the scene.

The Experience-Based Mapping of Churchill *et al.* [35] uses stereo visual odometry, while simultaneously processing many localizers that try to associate the current image stream with previous runs in the environment. When too few localizers are successful, the system flags the experience as novel and saves the visual odometry to localize against in the future. In many scenes, the number of experiences will stabilize after several visits. However, in areas that exhibit significant and random daily change (say, an overland parking lot with a limited number of visual cues), the number of experiences representing an area can grow without bound. Again, this is a side effect of aggregating data into experiences, rather than considering statistics at the landmark level.

Another research track has been focused on directly being able to match scenes under extreme environmental change (day/night, winter/summer). SeqSLAM [148] and its successor SMART [169] compute the difference between downsampled, patch-normalized image sequences, using a sequence of query images to find a match within a sequence of database images. Naseer *et al.* [157] take a similar approach using a full image descriptor, then searching for a lowest-cost path through the database sequence that matches a query sequence. While these approaches produce extremely impressive results under challenging appearance changes, there are two shortcomings. First, it is not clear how to update the image sequence database over multiple visits to an environment. Second, these algorithms once again aggregate data into whole images, ignoring the fine-grained information about stable environmental features that may be present in part of an image.

Although FAB-MAP [42] has been one of the most widely successful place recognition algorithms, its performance degrades when places are visited multi-

ple times [138], because the probability mass gets distributed between the multiple visits if they have not been correctly associated to belonging to the same place. Paul and Newman developed a method to update the prior place distribution for FAB-MAP, resulting in increased place recognition performance over time [168]. However, this method only updates the prior and does not maintain a database of places between runs, limiting its usefulness for continuous operation within a fixed environment. In contrast, Maddern *et al.* [136, 138] propose a method to mitigate the degrading performance with their CAT-Graph algorithm. They maintain place-recognition performance over time and keep a fixed computational complexity by discarding full images with low information content. Once again, this algorithm discards full images and is not able to derive benefit from stable image components.

The closest related work is that of Johns and Yang who propose an algorithm that learns feature co-occurrence statistics at places over time, rather than storing a new place when the appearance changes [100]. The approach works with or without using image sequence information and demonstrates impressive performance on challenging datasets, even under viewpoint changes. However, the algorithm does not include a methodology for removing data and so the storage and computational complexity will grow without bound during continuous operation.

3 Methodology

Our goal is to obtain a compact map that extracts the gist from all observations of an area while maximizing the information content in the *Summary Map* for localization and place recognition purposes. We start with a single consistent map which we want to periodically extend or update with experience from new datasets that are captured by agents. Each batch of new information undergoes a joint summarization step with the old map, keeping the map size constant but aiming to improve localization performance. This recursive map update method responds to dynamic changes as well as develops a more complete representation of the environment as new and varying viewpoints are captured over time.

The subsequent map extension proceeds in two steps: in the first step a new dataset is co-registered against the existing map using the loop-closure framework of Lynen *et al.* [134]. Loop-closures in this framework are found using a continuous representation of appearance based on individual landmarks (similar to [185]) instead of matching key-frames which is commonly done in related work [136]. The second step keeps the map-size constant by scoring and sampling to determine which parts of the new run are worth keeping and which parts of the old map are to be discarded. Generally, the size of a map can be reduced in two ways:

1. Reducing the number of landmarks, i.e. removing map features that are outdated, not useful, redundant or recognized as noise.

2. Reducing the number of posegraph segments, i.e. deleting redundant, overlapping pieces of trajectories.

In visual-inertial maps, the bulk of the data is given by the 3D-landmarks, which are used for metric localization. Even though in principle only a few well constrained landmarks are sufficient for robust localization and pose-estimation, online systems commonly estimate the camera-pose from hundreds of landmarks to be robust against noise and incorrect data-associations. As agents traverse novel areas of the environment or capture new viewpoints within the map we might however not always be able to associate all their visual observations correctly with the reference map. Consequently, a fraction of the landmarks estimated by the agents is redundant with existing structure. Most of the missed data associations can be recovered and redundant landmarks merged by place-recognition in the map-summarization process. This merging process improves the quality of the map by acquiring additional samples of the visual landmark appearance for better place recognition and using more constraints on landmark positions to reduce their uncertainty. The biggest gain in data reduction comes from landmark sparsification. Given the higher map-quality, all but the most relevant information required for localization can be discarded from both the old and the new map. To further reduce the amount of data, the system identifies runs that are mostly redundant with other runs, so that some part of them can be discarded.

3.1 Landmark sparsification

Once a map reaches the maximum permitted size, the *Summary Map* is generated using a *sampling* policy, which follows a *scoring* function based on the importance of landmarks. The final quality of the *Summary Map* is determined by the parameters and algorithms used in the *scoring* and *sampling* processes and which require careful analysis. We evaluate different landmark sparsification methods by first looking into the process of scoring landmarks within a single run.

Scoring by observation count *i.e.* the number of times a particular landmark has been observed is a simple scoring metric which was also used by Muehlfellner *et al.* [154]. Intuitively, a frequently observed landmark is likely to be a good feature, because it is easily re-observed, re-matched and most probably stable over time. However, a substantial drawback occurs if some regions of the map are visited significantly more often than others, since the landmark observation count is then biased towards these regions. Removing landmarks based on observation count can lead to situations in which we are unable to perform place recognition in the rarely visited regions due to the lack of landmarks.

Constraining the sampling process helps to resolve the issue of too aggressive pruning of trajectories from rarely visited areas. By imposing a constraint on the sampling process the order of landmark removal is changed wrt. what the scoring function initially defined. Before deleting a candidate landmark, we first verify whether all the map keyframes observing the landmark would still be well-constrained. Ideally, determining how "well-constrained" a keyframe is should take into account factors such as the geometric layout and the quality of the landmarks. For simplicity in our experiments, we use a lower-bound on number of visual observations per frame. Fixing a lower-bound on the observation count per keyframe

has the tendency to regularize the sparsification process, by ensuring that even places visited only once will still retain enough information to localize.

Scoring by covariance of the landmark's position, which can be estimated from the least-squares optimization is expected to convey more information about landmark usefulness than observation count. For example, the covariance takes into account the geometric configuration of the bundle-adjustment problem. A scalar score can be calculated from the 3×3 position covariance matrix using its trace or largest eigenvalue. However, a significant disadvantage of this method is in the computational cost of obtaining the covariance matrices due to the inversion of the Jacobian matrix of nearly the entire bundle adjustment problem. The complexity of matrix inversion is relatively high and grows more than quadratically with the number of landmarks in the map.

Sampling based on descriptor stability which can be derived from the standard deviation of descriptors from all the observations of a landmark is another policy in our system. A good landmark is expected to have observations with similar descriptors that show little variance over the observations even between different datasets. The sampling policy will not consider landmarks with a high descriptor variance since they are either a result of incorrect data associations or otherwise unstable features (e.g. highly lighting-dependent).

3.2 Multi-dataset landmark sparsification

So far we described the choices for our scoring and sampling policy based on the data a single dataset provides. Over time however multiple datasets will be merged into the global map.

Observations across multiple datasets are valued more than the ones only in a single dataset since they more accurately capture the information for long-term localization. We therefore suggest to use the number of trajectories a landmark was observed from as the most important factor when selecting the landmarks and augment it only in a second step with the number of observing keyframes.

However, a score on the number of observing trajectories does not solve the problem of bias towards regions which were visited more often. Instead, we propose a score that relates the actual number of trajectories to the expected number of trajectories to observe a landmark:

$$\text{score} = \frac{\text{Number of observer trajectories}}{f(\text{Expected number of observer trajectories})}$$

The choice of f depends on our goals:

- $f(x) = x$ will weight the landmarks by the probability of reobserving them $p(\text{observing}|\text{visiting the area})$.

- $f(x) = 1$ will cause weighting according to the actual trajectory observer count and may lead to rarely visited areas being dropped.

Information from the landmark covisibility graph is used to estimate the *expected* number of landmark observer trajectories. The scoring function queries the covisibility graph to retrieve all landmarks which are usually co-observed with the landmark to score and then searches these co-visible landmarks in all other trajectories. If a significant percentage (we use a threshold of 10%) of such landmarks are observed by a trajectory, we expect that the landmark in question should have been observed in this trajectory as well.

3.3 Trajectory sparsification

In addition to sparsifying the landmarks, it is important to remove certain segments of the accumulated trajectory as well. Each merge of a new dataset will add a new trajectory to the map, which constraints the newly added landmarks with metric inertial constraints. But because trajectories themselves do not aid localization, it is desirable to keep only a limited number of them in the map in order to limit the global memory budget. Removing trajectory segments does not necessarily have any adverse effects on the map – if a certain area is rigidly constrained by multiple trajectories, removing one of them will not significantly change the structure. In section 3.1 we mentioned that our sampling policy is driven by a lower-bound on the minimum number of keypoints per keyframe. Simultaneously, we limit the total number of landmarks in the *Summary Map*. With a growing number of merged trajectories (which implies a higher number of keyframes), the constraint on the minimum number of observations per keyframe however will eventually have to be violated to keep the map size below the fixed bound. Before the usefulness of a trajectory segment can be determined, we want to evaluate if the information that got captured is novel, redundant or noise. In order to allow this distinction, new data is kept in a sliding window of datasets where novel observations can be confirmed by additional evidence. Once the dataset leaves the sliding window, every section of the trajectory gets tested if it adds uniquely-observed structure to the map. The trajectory scoring and removal can be handled as follows:

- If a trajectory does not observe substantial amounts of novel structure: landmarks coobserved with other trajectories are associated with them and thus remain in the sliding window. The entire original path can then be deleted.

- If a trajectory contains unique structure information: the relevant sections are separated and stored as individual trajectories while discarding the rest of the original path.

Our sliding window works as a buffer for map landmark candidates where landmark re-observations allow the algorithm to separate trajectory segments which capture novel structure of the environment from landmark observations that are non repeatable and thus treated as noise.

We identify pieces of the trajectory which observe completely new structure (e.g. visiting a new place or using a viewpoint very different from previous ones) by estimating how many keypoints per keyframe are co-observed by other trajectories.

Landmarks	Length	Size	Num used	% used
39,071	112 m	22.4 MiB	2,485	6.36%
40,879	116 m	23.4 MiB	1,093	2.67%
39,492	118 m	23.8 MiB	1,095	2.77%
46,383	113 m	22.7 MiB	1,076	2.32%
48,552	116 m	23.4 MiB	574	1.18%
55,426	116 m	22.3 MiB	892	1.61%
42,324	114 m	27.1 MiB	589	1.39%
45,794	113 m	30.2 MiB	1,240	2.71%
49,687	116 m	31.5 MiB	956	1.92%

Table 4.1: The datasets used to highlight the summarization process. The number of landmarks, length and memory consumption of the maps is roughly equal. For the last columns, "use" denotes the number of landmarks that were used from these datasets in the final summary-map in comparison to their initial count. Note: The total size comprises of descriptors, IMU data, landmark as well as pose information.

4 Experiments

4.1 Experimental setup

To evaluate the performance of the proposed scoring functions and sampling policies we collected 10 datasets in a large public hall on different days. We used 9 of these datasets to build the *Summary Map*, see Table 4.1, and the remaining dataset for evaluation respectively. Each dataset is approximately 100 m long, covering a cafeteria and working spaces where we expect significant appearance changes. The datasets were processed using the monocular visual-inertial batch optimization of the Google Tango [118] project, and co-registered using the placeless-loopclosure algorithm by Lynen *et al.* [134] within the Map API framework [36].

4.2 Evaluation Methodology

We would like to evaluate the influence of different scoring functions and sampling policies on the place recognition and localization performance as more experiences from other datasets are added to the *Summary Map*.

Furthermore, we are interested in the fraction of landmarks that can be removed in the summarization process out of a single dataset while still allowing robust place recognition. To evaluate the place recognition performance we count the number of successful keyframe localizations of an independent *evaluation* dataset to the *Summary Map*. A localization is considered successful (true positive) if the camera pose from the independent *evaluation* dataset can be recovered from the local structure with a translational error of less than 0.5 m compared to ground-truth. The ground-truth reference is determined by co-registering both the original (non-summarized) and the new datasets using loop-closure detection and bundle-adjustment. However not every keyframe in a dataset is localizable; for instance, when the viewpoints do not overlap with the *Summary Map*. Therefore, we use

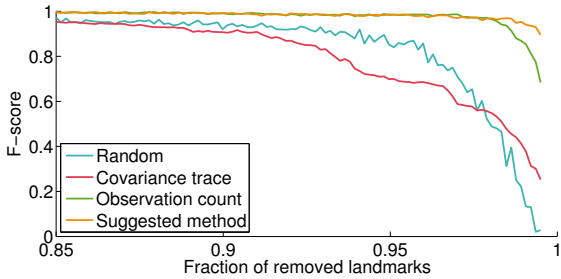


Figure 4.2: F-scores of place recognition performance when using different scoring functions and sampling policies to create a *Summary Map* from multiple datasets. The X-axis denotes the fraction of landmarks that were removed during the summarization process.

the subset of keyframes which are localizable against the full map as ground-truth.

During the evaluation, we apply a scoring function to the reference map and remove landmarks according to the sampling policy. We then query every keyframe from the *evaluation* dataset against the *Summary Map*. Precision and recall values are computed from the number of successful localizations in comparison to the ground-truth values from the non-summarized maps. We then compute the F-Score as a final performance measure:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

.

Besides the evaluation of scoring functions and sampling policies, we evaluate the relocalization performance to *Summary Maps* built from multiple runs. Starting from a single dataset, more datasets are iteratively merged and sparsified to keep the number of landmarks constant. After every iteration of the summarization process we run the performance evaluation as described above.

As a baseline we also evaluate a sampling policy that picks points at random.

5 Results

We evaluate the relocalization performance for different scoring functions and sampling policies used to create the *Summary Maps*.

In the first experiment a single dataset is summarized using different strategies and then used as a basis for localization. Figure 4.2 shows the localization performance for the different sampling functions as the summarization process removes more and more landmarks.

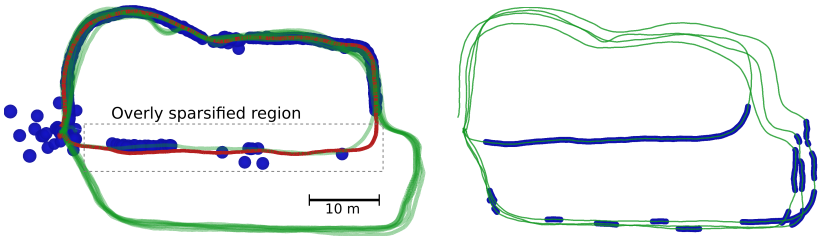


Figure 4.3: Left: The uneven distribution of trajectories (green lines) over the map leads to an uneven distribution of observations per landmarks. Scoring solely based on observation count causes under-representation of rarely visited areas. Blue dots denote results from the localization, with some of them being outliers. The *evaluation* trajectory is denoted by a red line. After the summarization using raw observation count, localization is no longer possible in the overly sparsified region.
Right: Detection of keyframes (blue) that provide enough new information over the base-map to justify being kept for later localization even if their dataset is moved out of the sliding window. The continuous segment in the middle of the map is a novel trajectory, while short segments in the bottom and bottom-right capture drastic changes in visual appearance of already mapped areas.

As shown in Figure 4.2 our proposed method works better than a scoring function based on landmark covariance as well as the method based solely on landmark observation count. The gain from our suggested method is especially significant for the sparsification ratios over 98%. As mentioned in Section 3 the method based on the landmark covariance has a very high computational cost. Furthermore, it was not possible to compute the covariance for landmarks only appearing in degenerate configurations where the Jacobian was rank-deficient.

When using a scoring method solely based on observation count, we observe a bias towards sampling points that were observed more often just because they were visited more often than others. Figure 4.3 (left) demonstrates how the scoring function that only uses the landmark observation count negatively affects the place recognition performance for such maps. The map structure in the rarely visited areas is completely sparsified which makes it impossible to localize there. As discussed in Section 3 we can remove this bias by introducing an additional constraint on the sampling process, a lower-bound on number of visual constraints per keyframe.

The final experiment evaluates the localization performance to a *Summary Map* which was created by iteratively merging new experiences using our suggested method, while keeping a constant map size.

Despite limiting the *Summary Map* to contain less data than a single one of

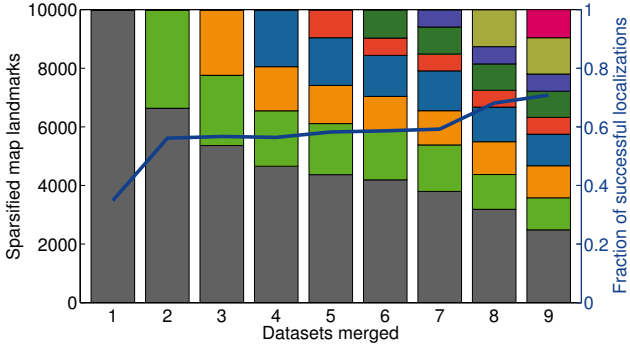


Figure 4.4: This figure shows the increase in localization performance as more datasets become of the *Summary Map* which has a constant size of 10000 landmarks. The fractions of the bars provide information about the origin dataset of the landmarks the *Summary Map* is composed of.

the original datasets, we can see in Figure 4.4 that results continue to improve with merges of additional data. Thus we can conclude that our method efficiently summarizes information, selecting the unique and reliable map elements. The bar plot in Figure 4.4 depicts the dataset origins of the landmarks in the *Summary Map* for every iteration. Each dataset which is merged and summarized provides a piece of useful information to the *Summary Map*. One can observe that some datasets contribute less than others, *i.e.* . datasets that were recorded on the same day as a previously merged dataset, which indicates that they contain mostly redundant information already present in the *Summary Map*.

Figure 4.5 shows the landmarks used for localizing the *evaluation* dataset in the *Summary Map*. Since the total number of landmarks is constant, we see that as new datasets get merged the fractions that each dataset contributes to the *Summary Map* are shifted. Even after performing all the iterative summarization, the initial dataset still provides the largest number of landmarks to the *Summary Map*. This behaviour is expected since the first dataset observes most of the static structure in the environment.

Figure 4.5 also shows that new datasets fill the missing information in the *Summary Map* that was left incomplete from previous datasets. Some of the initially uncovered sections of the *evaluation* dataset were covered as the last datasets became part of the *Summary Map*. Color represents the latest dataset a particular landmark was observed by. When a landmark is matched to the visual observations of a new dataset, the color of the corresponding point is updated. Thus, the better the matching and localization procedure is, the more points are expected to

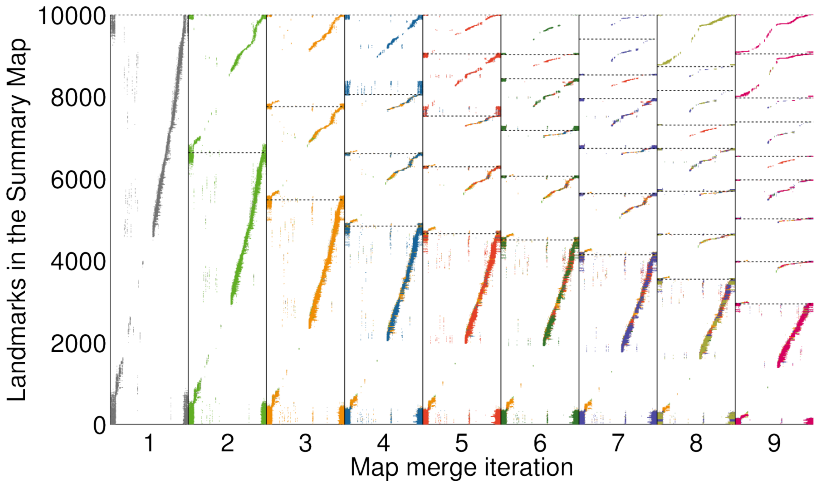


Figure 4.5: A visualization of the fixed number of landmarks that form the *Summary Map*. Within each vertical strip corresponding to a map merge iteration, the X-axis represents keyframes within the evaluation dataset, and data points correspond to landmarks used for localization of each keyframe. As landmarks from additional datasets become part of the *Summary Map*, the fractions that each dataset provides shift (separated by dashed horizontal lines). The colors denote the dataset from which a landmark was last observed. The better the data association and loop-closure the more landmarks should change color as a new dataset is merged. As more and more datasets become part of the *Summary Map*, parts of the *evaluation* dataset that were initially not localizable, become localizable.

change color as a new dataset becomes part of the *Summary Map*.

As discussed in Section 3, to keep a constant map size the parts of the trajectories that do not cover new areas and contain redundant data can be discarded. We propose keeping new datasets in a sliding window in which their observations can be confirmed or identified as noise. Figure 4.3 (right) shows the identification of keyframes within such unique regions. The keyframes marked in blue should be retained even after the corresponding dataset is moved out of the sliding window. The experiment demonstrates that our algorithm correctly selected the places that were not visited before, namely the trajectory segment in the middle of the plot. When repeatedly visiting the same regions large parts of the trajectory are identified as redundant and not kept in the map as shown in Fig. 4.6 which leads to an approximately constant map size. As a result, reliable place recognition in such

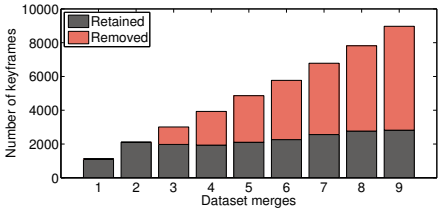


Figure 4.6: Illustration of the dataset sliding window performance. Trajectory sparsification was performed for datasets going out of the sliding window of size $n = 2$. Only the keyframes which observed completely new structure in the map were retained. The resulting map keyframe count is approximately constant over multiple merges of datasets covering the same trajectory.

uniquely seen regions is still possible at a later time. Figure 4.3 (right) shows some other areas which were previously covered by other datasets. These regions, however, underwent such drastic changes in visual appearance (crowded areas of the cafeteria) that they provide enough new information to justify being kept for later localization.

6 Conclusions

In this paper we presented an algorithm that alternates between online place recognition and offline map maintenance with the goal of producing the best place recognition and localization performance from a fixed map size. We propose and evaluate *scoring functions* that rate the utility of every landmark and *sampling policies* which are used to pick the most useful information to be combined into a Summary Map. We showed how this reduced map, despite having only a fraction of the original size, can be maintained over time while constantly providing better place-recognition performance as more experience about an environment is gathered. This *Summary Map* can then be used by any descriptor-based inference method for constant-complexity online place recognition.

Future work will focus on identifying dynamic scenes and related changes in the appearance in a more flexible way such that the algorithm can correctly classify them.

7 Acknowledgments

The research leading to these results has received funding from Google's project Tango and the European Union Seventh Framework Programme FP7/2007-2013, Challenge 2, Cognitive Systems, Interaction, Robotics, under grant agreement No

269916, V-Charge and grant agreement no 610603, EUROPA2. Thanks to Will Maddern for helpful discussions about the current state of the art and the reviewers for their constructive feedback.

Paper II

Keep it brief: Scalable creation of compressed localization maps

Marcin Dymczyk, Simon Lynen, Michael Bosse, Roland Siegwart

Abstract

Robust, scalable localization unlocks path-planning, obstacle avoidance as well as manipulation and thus is a core competency for many robotic applications. However, as we leave the lab and move out in the world, models of the environment no longer span distances of meters but kilometers in length. Now, gigabytes instead of megabytes of memory are required to hold the model of the environment required for localization. Discarding data and keeping the map representation compact is thus essential for any meaningful application.

This paper presents and evaluates a map compression algorithm that approaches this data-reduction as an constrained optimization problem. At the core of the algorithm is the concept of a Summary Map, a reduced map representation that includes only the landmarks that are deemed most useful for place recognition. To assign landmarks to the map we have to satisfy the conflicting goals of map coverage and localizability as well as our tight memory budget. While using an optimization approach for compression is not novel, in this paper we propose adaptations to drastically reduce the computational requirements. Our approach improves scalability from trajectories of a few tens of meters manageable by the state of the art to virtually unlimited dataset sizes in our system. We evaluate the performance of various compression levels as well as several methods for selecting the best localization landmarks from outdoor datasets.

Published in:
Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, 2015
DOI: 10.1109/IROS.2015.7353722

1 Introduction

Being able to reliably estimate and track the pose of a robot with respect to an existing map at a high frequency is fundamental to enable path-planning, obstacle avoidance, manipulation and repeatable inspection. Designing systems that provide highly accurate localization at a large scale while running on platforms with limited computational resources is thus a key problem in robotics. One crucial aspect for localization systems, regardless of whether they are used for agile flying robots, manipulation tasks or inspection, is that they provide a (nearly) perfect registration to a global model covering the area of interest.

Using visual and inertial sensors, *i.e.* , a camera and an inertial measurement unit (IMU), enables localization under a wide range of conditions without relying on the availability of GPS or WiFi signals. State-of-the-art methods for online, low-latency visual (inertial) localization assume the existence of a 3d model of the scene. This model is often pre-built from a set of database images using Structure-from-Motion (SfM). Since SfM reconstructs each 3d scene point from multiple 2d visual features, each such 3d landmark can be associated with a set of local image descriptors [185]. Given a novel image together with features [3, 13] extracted from it, the camera pose w.r.t. to the model can be computed from 2d-3d matches between image features and landmarks using relative [160] or absolute [79] pose solvers. In order to match the features against the 3d landmarks, approaches from robotics can be used to first retrieve relevant database images [42, 72]. The image features are then matched against the landmarks observed in the retrieved keyframes [6, 47]. Alternatively, matching can also be done directly against the full model [94, 125, 147, 191, 226].

Localization approaches that keep the full scene model on a mobile device or robot [6, 47, 94, 127, 226] are usually restricted to small workspaces due to the high memory requirements of storing the model consisting of several thousands if not millions of 3d-points. More scalable approaches assume the availability of an external server, which is used to send the relevant parts of the model to the device [7] or even perform the actual localization [147, 219, 220]. Instead of handling large scale models by outsourcing computation to a server, an alternative is to compress the global model into a compact map by removing unnecessary scene geometry. We call these compact maps Summary Map*s* since they contain only the most important information of an environment required for localization. This allows localization from building scale models on clients having computational resources similar to micro aerial vehicles or mobile phones. Besides reducing storage requirements of the model on the robot, such compression also unlocks online transmission of the map during the course of the mission.

The goal of the summarization process is to convey as much localization information as possible while reducing the model to a compact representation. Compressing maps by selecting informative landmarks for visual localization was also proposed earlier [123] using greedy heuristic selection.

In order to provide localization in all locations of the map the goal of any such compression is to find the most compact subset of the 3d-points while providing

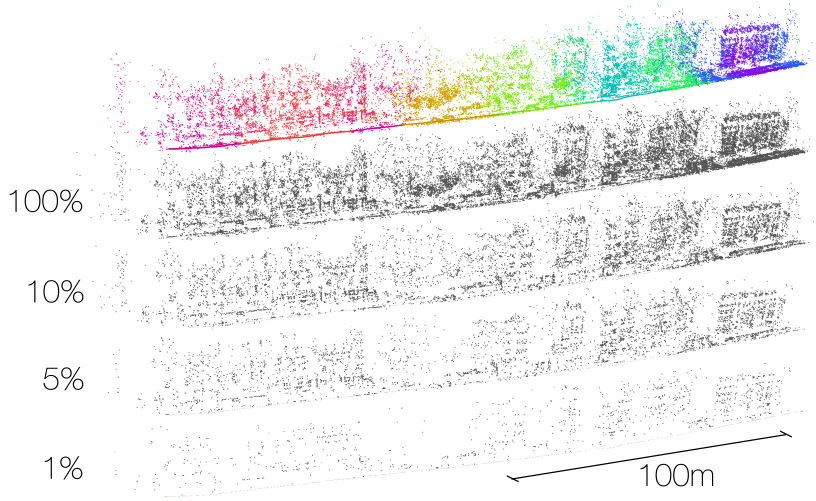


Figure 5.1: Different compression ratios of a visual-localization map. The selection of the most useful landmarks for localization is formulated as a constrained optimization problem. While this has a clear advantage over greedy selection strategies, the problem becomes infeasible to compute already for medium sized problems. In this work we thus partition the problem into sub parts (top) which are solvable and yield the near-optimal localization map given by the optimization based landmark selection.

at least k points for localization as required by perspective-n-point algorithms. These requirements are related to the maximum cover problem which has been shown [104] to be NP-complete. Therefore Li et al. [126] as well as our previous work [50] employ heuristic greedy strategies to choose the geometry to be preserved in the *Summary Map*. In both approaches the number of images in which a landmark is observed is used to derive the probability of a landmark becoming part of the *Summary Map*. Using these heuristic methods, impressive compression results up to 99% can be achieved, while still providing high localization quality. However given the sub-optimality of the greedy algorithms, the analysis of the performance as well as the precise design of the heuristic is problematic. We are therefore interested in approaching map summarization as a well defined optimization problem.

Integer linear programming (ILP) with binary variables is a well suited approach for SfM models to finding the optimal subset in coverage problems such as the best

set of cameras [86, 140]. Similarly ILP can be applied to compute the best subset of 3d-points to retain in the localization model [167]. Park et al. [167] improve the ILP results by incorporating the feature distribution in the image and solving the resulting problem using quadratic programming (QP). We use the methods proposed by Park et al. [167] as a baseline for the methods presented in this paper.

While using QP for the optimization can lead to improved localization results, the process of enumerating solutions using the branch-and-bound algorithm has high computational complexity and thus very limited scalability: In the experiments of Park et al. [167] it took more than 30 minutes to solve a QP problem with 30,000 landmarks corresponding to ≈ 100 m trajectory. For the datasets we have evaluated we found that we couldn't solve problems with a landmark count exceeding 50,000 on the available computer with 16 GB RAM. In order to solve summarization problems of reasonable size we propose to apply a subdivision algorithm similar to what Mauro et al. [140] have applied to view selection.

The contributions of this paper are as follows:

- we present new methods for map compression with improved performance over the methods presented in [50]
- we evaluate the quality of the *Summary Map*s and compare different algorithms on datasets from a dynamic environment with challenging lighting conditions
- we demonstrate that our method performs better than the current state-of-the-art presented in [167]
- we suggest a partitioning algorithm that makes our algorithm scalable – it can handle a virtually infinite input map size (in contrast to state-of-the-art methods)
- we show that the partitioning causes only marginal (if any) quality loss compared to solving the full problem
- we demonstrate that the suggested method is able to reduce the map to less than 1% of the initial size while still providing high localization quality.

2 Methodology

Our goal is to obtain a compact map that extracts the most useful information from a 3d-model for a given a area while maximizing the information content in the *Summary Map* for localization and place recognition purposes. The employed place-recognition system localizes a frame by establishing correspondences between 2d image features and 3d landmarks by descriptor matching [134]. The resulting 2d-3d matches are then used to robustly estimate the camera pose via the perspective-n-point (PnP) problem in a RANSAC [62] scheme. It is important that the model summarization process considers the underlying principle of

the localization system, namely having a minimum set of landmarks visible in all frames and a sufficiently high distribution of features in the images. These two desired properties of the final model are constraints to the summarization process and thus require a non-trivial algorithm to be applied during data-reduction.

2.1 Landmark sparsification

Greedy method. In our previous work [50] we used an iterative reduction algorithm by alternating between *scoring* and *sampling* policies. In every iteration each landmark was assigned a score, that was based on the number of keyframes they were observed in and covariance derived from the least-squares bundle adjustment optimization. These scores then controlled a sampling process which was constrained by two factors:

- A minimum number of keypoints should be visible in each keyframe to ensure sufficient landmark constraints in all map parts for PnP pose recovery.
- Low descriptor variance to discourage inclusion of landmarks in the *Summary Map* if their appearance is unstable.

Even though this approach provides satisfying results for both indoors and outdoors datasets, there is no guarantee on solution optimality.

Integer Linear Programming. Due to the greedy nature of the above algorithm it is not possible to obtain information about how close a given solution is to the optimal set of landmarks. Combining a scoring function with a constrained sampling process can, however, be directly mapped to an integer linear programming problem, for which optimal solutions can be found efficiently:

$$\begin{aligned} &\text{minimize } \mathbf{q}^T\mathbf{x} \\ &\text{subject to } \mathbf{A}\mathbf{x} \geq b\mathbf{1} \\ &\qquad \sum_{i=1}^{N} x_i = n_{desired} \\ &\qquad\qquad \mathbf{x} \in \{0,1\}^N \,, \end{aligned}$$

where $\mathbf{x}$ is a binary vector whose ith element is one if the ith point is kept in the model or zero otherwise. $\mathbf{A}$ is an $M \times N$ visibility matrix where M is the number of images and N the number of points in the model. b denotes the minimum number of 2D-3D correspondences to keep for each keyframe and $\mathbf{q}$ is a weight vector which encodes e.g. the number of observations or any other type of score per landmark. We build on the work of Park et al. [167] which proposed to use ILP for this selection process, but add an additional constraint that allows to exactly specify the total number of landmarks in the resulting model. This constraint allows to exactly specify the memory requirements of the resulting *Summary Map*, without the need to iteratively adjust $\mathbf{q}$ and b.

Imposing this maximum model size requirement conflicts with the constraint of having a minimum number of observations for every keyframe. The existance of conflicting constraints renders the ILP problem infeasible and thus requires the addition of slack variables. The optimizer can relax the slack variables when it is otherwise not possible to satisfy a certain constraint, their utilization, however, adds a significant cost to the optimization objective s.t. the constraints are satisfied as closely as possible. The ILP problem with slack variables has the following form:

$$\begin{aligned} \text{minimize } & \mathbf{q}^T\mathbf{x} + \lambda \mathbf{1}^T \boldsymbol{\zeta} \\ \text{subject to } & \mathbf{A}\mathbf{x} + \boldsymbol{\zeta} \geq b\mathbf{1} \\ & \sum_{i=1}^{N} \mathbf{x}_i = n_{desired} \\ & \mathbf{x} \in \{0,1\}^N \\ & \boldsymbol{\zeta} \in \{\{0\} \cup \mathbb{Z}^+\}^M. \end{aligned}$$

The same variety of scoring functions as for the greedy approach can be used with ILP by incorporating their results into the linear term weight $\mathbf{q}$.

Integer Quadratic Programming. By adding quadratic terms to the ILP problem formulation, more complex objectives can be formulated.

$$\begin{aligned} \text{minimize } & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} + \lambda \mathbf{1}^T \boldsymbol{\zeta} \\ \text{subject to } & \mathbf{A}\mathbf{x} + \boldsymbol{\zeta} \geq b\mathbf{1} \\ & \sum_{i=1}^{N} \mathbf{x}_i = n_{desired} \\ & \mathbf{x} \in \{0,1\}^N \\ & \boldsymbol{\zeta} \in \{\{0\} \cup \mathbb{Z}^+\}^M. \end{aligned}$$

Here again, we augmented the formulation of Park et al. [167] with a constraint on the resulting number of landmarks. Additionally, we propose to compose the quadratic term weight matrix $\mathbf{Q}$ differently: Park et al. [167] propose to set the element $\mathbf{Q}_{ij}$ proportional to the number of frames where the landmarks with indices i and j are co-observed. This formulation, however, penalizes **all** landmarks that are commonly observed which biases the selection process towards rarely observed landmarks, which is detrimental for localization.

We propose to change this formulation by taking the localization process into consideration which utilizes the resulting *Summary Map*. Due to geometric dilution of precision it is beneficial for pose recovery to prefer a sufficient spread of

observations in the image which is covered by the following formulation:

$$\mathbf{Q}_{ij} = \frac{\sum_{k=1}^{m} \max\left(0, \frac{d_{min} - \|p_{k,i} - p_{k,j}\|}{d_{min}}\right)}{m},$$

where $p_{k,i}$ denotes the position of the i-th landmark in the k-th keyframe (out of m). The quadratic term represents an average distance (relative to d_{min}) between the two keypoints (in pixels in the image plane), but only if they are closer to each other than the threshold d_{min}. In general, matrix $\mathbf{Q}$ will penalize landmarks that project to nearby image location in observing keyframes.

The improved modeling enabled by using a QP solver leads to higher localization quality for a given map size compared to ILP (See Section 3).

2.2 Summarization problem partitioning

Motivation. QP based model summarization has shown to outperform both greedy and ILP based optimizations and is therefore the preferred method for compression. The complexity of mixed-integer programs, however, grows significantly faster with the number of variables since each landmark introduces an additional binary variable. This property makes even relatively small maps very difficult to solve. In order to circumvent this problem and increase scalability to the envisioned city-scale maps, we propose to partition the problem into sub-graphs which can be solved individually. The partitions should be as independent as possible in order to limit the errors introduced by this approximation. Our method splits the problem using graph-cut, where the edge weight between keyframes is given the number of landmark co-observations.

Methodology. The partitioning algorithm splits the problem into subpieces by first defining a graph with key-frames as nodes. Edges between all key-frames observing at least n_{min} common landmarks are added with a weight corresponding to number of co-observed landmarks. The graph is then partitioned into N partitions (N is estimated based on the number of landmarks in the full map) using a graph-cut that minimizes the sum of weights of the cut edges. We adjust the partitioning algorithm to balance the size of partitions and penalize non-contiguous and singleton partitions. Only adding edges with a minimum edge weight to the problem significantly reduces the complexity of graph partitioning procedure while only marginally influencing the final solution. The resulting sub-problems have minimal interdependencies and thus allow obtaining a solution from the partitioned optimization which is close to the optimal (non-partitioned) solution (see Fig. 5.2 for an example).

Given that the summarization process is carried out on sub-problems the problem formulation needs to be updated to

$$\mathbf{A}\mathbf{x} + \boldsymbol{\zeta} \geq \min\left(b, c_{i,j}\right)\mathbf{1},$$

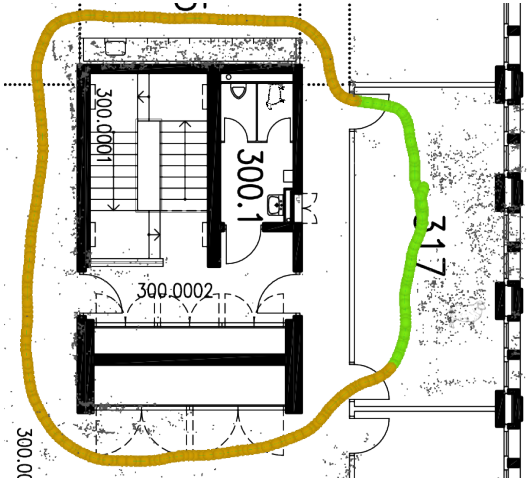


Figure 5.2: Partitioning the model compression task by cutting the problem into sub-parts having the smallest number of landmark co-observations. Splitting the problem this way ensures that the errors introduced by the partitioned compression are minimized. In this example a trajectory passes through two separate areas which, after applying the proposed partitioning, can be compressed individually.

where $c_{i,j}$ denotes the number of landmarks visible in keyframe i which belong to the partition j.

Each resulting sub-map can now be summarized separately, yielding a few times smaller LP or QP optimization problems, depending on the chosen number of partitions. Reducing the problem sizes has a significant influence on the computational requirements as both LP and QP problems grow significantly faster than linearly with the number of map landmarks N.

Quality loss with high degrees of sparsification. Splitting the optimization problem into sub-parts introduces an approximation of the visibility constraints at the partition boundaries. Even though the partitioning is carried out along regions with low covisibility constraints, key-frames near partition boundaries observe landmarks which belong to neighboring regions. Given that the summarization is performed on each sub-problem individually, these neighbor-landmarks cannot be considered by the constraint requiring a minimum number of landmark observations per key-frame. This approximation leads to the situation in which landmarks in two neighboring sections are removed simultaneously while leaving the observing key-frame insufficiently constrained. Naturally, the errors introduced

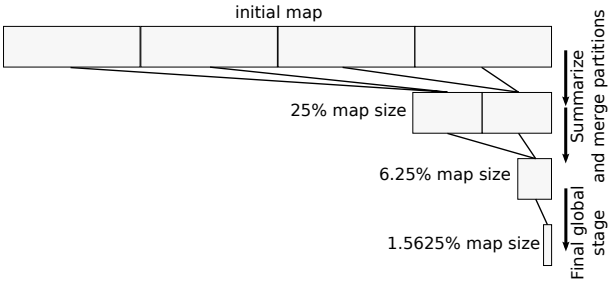


Figure 5.3: In order to reduce errors introduced by solving subparts of the problem individually, we formulate the summarization as an iterative process. At every step of the optimization only a fraction of the total desired landmark removals is carried out. This limits the approximation errors of the constraints in the optimization problem while allowing the algorithm to scale even to very large models.

by this approximation depend on the amount of landmarks removed per optimization step. In order to circumvent this problem we perform summarization as an iterative process in which at every step the landmark reduction ratio is controlled to limit the approximation error. Furthermore the final steps of reduction in our algorithm are carried out on fewer partitions to further reduce the influence of the approximation. A scheme of the iterative summarization procedure is presented in the Fig. 5.3.

3 Experiments

3.1 Experimental setup

We build a model from a single run through a street using a sequence of images and IMU data in a visual-inertial least-squares optimization (bundle-adjustment). The resulting 3d model consists of 134,297 3d points associated with 821,655 descriptors, reconstructed from a trajectory of 204 m. We apply the model summarization scheme discussed in Section 2 such that at least $b = 30$ of the selected landmarks are visible in each of the 1,453 keyframes. For constructing the quadratic term of a QP problem, a d_{min} of 40 pixels was used. The compression scheme reduces the global map down to 1,074 points (0.8%) with 50,628 descriptors, reducing the memory requirements from 52.7 MB to 3.1 MB. We use the Metis [106] library to subdivide the problem in a desired way, including contiguity and partition balancing requirements. The resulting optimization problem is formulated as mixed-integer LP/QP problem and solved using the Gurobi [82] library.

We record an additional dataset for evaluation of the localization accuracy given

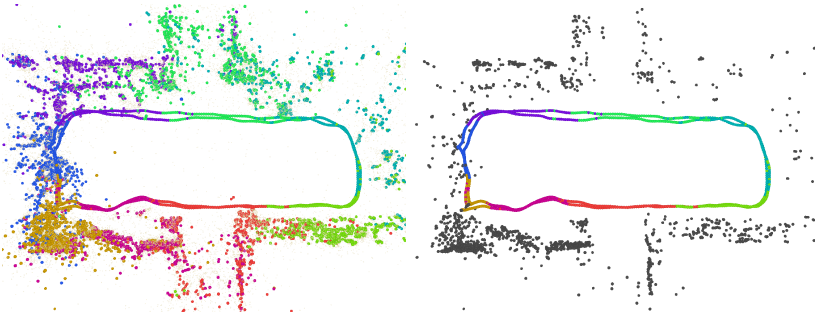


Figure 5.4: Left: A result of compression on sub-partitions. In each sub-map only 10% of the initial landmarks are retained. **R**ight: After the sub-map compression the overall problem is small enough to allow global optimization. The resulting map contains only 3% of the initial landmarks while providing nearly the same localization quality as the full map. Note that the partitioning is done according to the number of co-observed landmarks and not based on a spatial division.

	Vertices	Initial Landm.	Length	Initial Size
Map mission	1,453	134,297	204 m	52.7 MB
Query mission	1,350	107,822	200 m	39.8 MB

Table 5.1: The outdoor datasets used to evaluate different summarization strategies. Both were collected using a Google Tango device at different lighting conditions and varying viewpoints. The number of landmarks, number of vertices and length of the maps are roughly equal.

the *Summary Map* where we perform moderate view-point changes compared to the first dataset. We obtain a ground truth trajectory by using the batch loop-closure algorithm of Lynen et al. [134] to find candidate 2d-3d matches for each frame in the evaluation sequence. These candidates are then filtered using spatial verification [185] before we associate the observations from the evaluation dataset with the model landmarks. Finally we run a full-batch visual-inertial bundle adjustment (BA) to align the evaluation trajectory with the model. After alignment, the poses of the evaluation trajectory are used as ground truth. All experiments were done on an Intel Xeon E3-1245 CPU with 3.3 GHz. A video of the system localizing in real-time from the *Summary Map* covering a 204 m long trajectory can be found in the supplementary material.

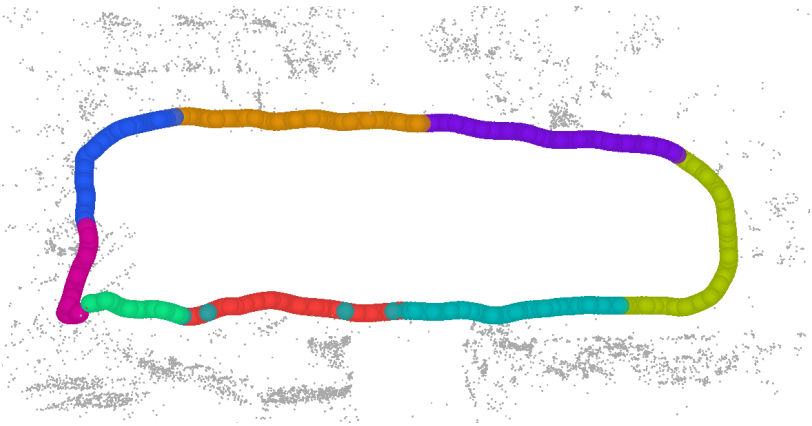


Figure 5.5: Partitioning of an indoors dataset (965 keyframes) into 8 subproblems. Note how the partitions correspond to the geometrical structure of the environment. The plot is a top-down view on the 3d map.

3.2 Partitioning effects

We first want to analyze the effects of partitioning the optimization problem into subparts. As discussed in subsection 2.2 the approximation given by the partitioning of the problem leads to errors related to the number of removed landmarks per summarization step (i.e. removing over 95% of landmarks in one step leads to a significant degradation of localization performance). The approach to iteratively apply summarization and reduce the number of partitions with a final global summarization as the problem size shrinks yields results that are only marginally worse than the global optimization (Fig. 5.6). The subdivision of the problem, however, reduces the computational time significantly and thereby allows summarization of meaningful model sizes.

3.3 Compression and summarization performance

After having determined the best strategy to subdivide the problem into smaller pieces which are feasible to solve we are interested in the performance of different algorithms to select the best subset of landmarks. Using an ILP based optimization method provides a map which has a significant higher rate of successful localizations compared to the greedy method proposed by Dymczyk et al. [50] especially at high compression rates. For the evaluated dataset the computationally more expensive QP based method of Park et al. [167] did not provide an improvement over the

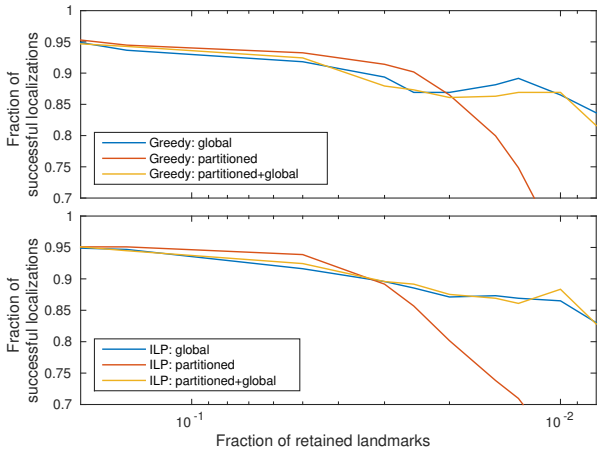


Figure 5.6: Comparison of different partitioning approaches for greedy and ILP-based summarization (*global*: single global optimization; *partitioned*: single partitioned optimization till the landmark count goal; *partitioned+global*: single partitioned optimization up to 95% removal rate, then global optimization). When partitioning the problem and summarizing these in one step, leads to a degradation in localization performance once more than 95% of the initial landmarks are to be removed. When running a final global refinement on the initially partitioned problem after an initial landmark reduction by 90% the resulting *Summary Map* provides a localization performance equal to the global optimization method without partitioning.

Method	Ratio	Localized frames	Inlier ratio	Avg runtime
Greedy	0.90	1,214	0.342	2.01 s
	0.95	1,195	0.328	4.08 s
	0.99	885	0.282	5.85 s
ILP	0.90	1,215	0.342	0.94 s
	0.95	1,188	0.332	1.62 s
	0.99	918	0.292	2.14 s
Ref. QP	0.90	1,211	0.341	90.84 s
	0.95	1,191	0.333	112.52 s
	0.99	925	0.290	101.25 s
Prop. QP	0.90	1,213	0.341	93.58 s
	0.95	1,189	0.332	103.39 s
	0.99	968	0.292	97.23 s

Table 5.2: Results, 128 partitions, outdoor dataset. 134,297 total initial landmarks observed by 1,453 keyframes. The threshold on the error for counting a localization attempt as successful is 10 cm wrt. ground-truth.

ILP result. Changing the objective function of the QP problem as proposed in this paper provides a significant improvement over both the ILP as well as the QP based results of Park et al. [167]. Due to the nature of the underlying absolute pose solvers used for localization a few points are sufficient to localize. Therefore the differences between the various methods for landmark selection only start playing a role once the compression ratio surpasses 98%. For example our method is able to compute a model compressed by 99% which allows an additional 43 keyframes (out of 1,453) to be successfully localized within a 10 cm threshold compared to the model computed by the current state-of-the-art QP approach [167]. Even when further increasing the compression ratio to 99.2% the model still allows localizing over 65% of the key-frames. Thus for the outdoor dataset of 200 m length we only need about 3.1 MB of data to allow updates from the map for most of the time. The reduction from 52.7 MB to 3.1 MB is especially beneficial for streaming such localization maps to mobile devices using WiFi of mobile networks. Also for the number of PnP RANSAC inliers (a metric for the reliability of the pose-estimate) the advantage of the proposed QP method over the state-of-the-art is shown especially for very high compression rates (See Fig. 5.7).

3.4 Timing

We have measured the time consumption of the different evaluated methods (See Table 5.2). Fig. 5.8 demonstrates the computational cost of the ILP and mixed-integer QP optimization with respect to the problem size (which grows with the number of landmarks).

Depending on factors such as the size of the QP problem, the weights of the quadratic terms and their number the solving time can vary heavily (from milisecond range up to problems infeasible even for high-end PCs). For problems reduced to about 500-1,000 landmarks, however, the problem is immediately relaxed and

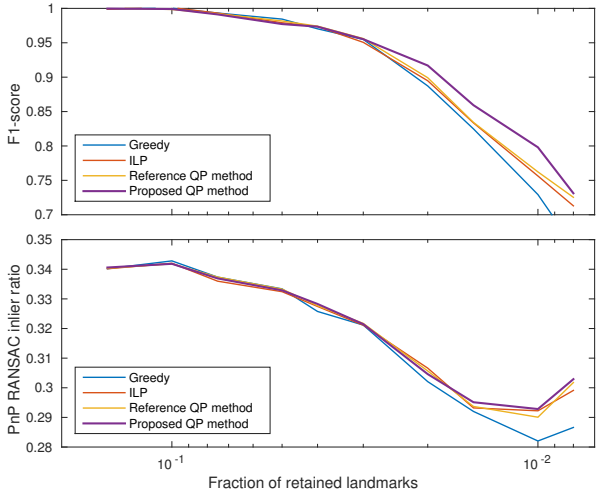


Figure 5.7: Top: Localization F-scores for the evaluated methods, depending on the fraction of removed landmarks. The proposed QP based method for compression outperforms other approaches for compression rates exceeding 98%. **Bottom:** PnP RANSAC inlier ratio from the localization queries: Here all optimization based approaches achieve similar performance wrt. the greedy landmark selection strategy. A smaller number of inliers correlate to less correspondences and thus a decrease in accuracy of perspective-n-point algorithms. In extreme cases a low number of inliers can make the localization impossible, when less than n matches to the model can be established. An increasing inlier ratio for the highest landmark removal ratios is caused by limited number of matches returned by the localization engine (close to n of PnP). In such situation, any hypothesis of PnP RANSAC will yield n out of all matches marked as inliers, resulting in a relatively high inlier ratio. Note: values in both plots are averaged over 50 independent iterations.

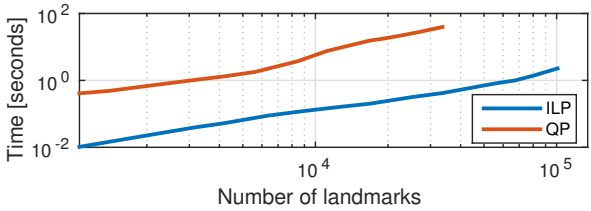


Figure 5.8: Relationship between the number of map landmarks and ILP/QP summarization time: The super-linear growth of the computational complexity motivates the suggested sub-division of the optimization problem. The initial computational complexity of QP and the increase with the number of landmarks makes any but the smallest problem too complex to be solved on a PC.

the branch&bound algorithm takes no more than few seconds to find an optimal solution.

4 Conclusions

In this paper we focused on map compression by selecting the most informative landmarks for localization. The resulting maps require only a fraction of the original size but provide nearly the same localization quality compared to the full model. During the compression process two competing constraints need to be satisfied: Removing uninformative landmarks to achieve the desired compression ratio while making sure that localization remains possible in all areas of the map. While in our previous work [50] we tried to find a solution using a greedy algorithm and heuristics, in this work we approach map compression as an optimization problem. Building on the work of Park et al. [167] we evaluate both integer linear programming (ILP) as well as mixed integer quadratic programming (MI-QP) as solvers for this task. We show that using the additional cost-terms enabled by the QP formulation both the solutions of the ILP and the greedy method are outperformed. Our parameterization of the QP problem furthermore improves localization performance over the results of Park et al. [167] especially for very high compression rates exceeding 98%. However, given the computational complexity of QP solving the problem for most real-world datasets is infeasible. We therefore propose a constrained partitioning of the map into sub-graphs which are compressed individually. By running the partitioning as an iterative process we ensure that the partitioning has no undesired effect on the final result. This divide-and-conquery technique drastically improves scalability from trajectories of a few ten meters manageable by the state of the art to virtually unlimited dataset sizes in our system: Compressing a map with 30,000 landmarks takes less than 60 seconds compared to 30 minutes as reported by [167] on similar hardware. Still, given the increase in the

solver time for the QP solver compared to the IPL solution, we want to emphasize that for some problems already the ILP solution might be sufficient.

For future work we would like to combine the results from this publication with the iterative multi-session map compression proposed in our earlier work [50]. Particularly for larger datasets the partitioning approach will also allow us to leverage our cloud-backend [36] such that we can distribute the compression over multiple machines.

5 Acknowledgments

The research leading to these results has received funding from Google's project Tango.

Paper III

Erasing bad memories: Agent-side summarization for long-term mapping

Marcin Dymczyk, Thomas Schneider, Igor Gilitschenski, Roland Siegwart, Elena Stumm

Abstract

Precisely estimating the pose of an agent in a global reference frame is a crucial goal that unlocks a multitude of robotic applications, including autonomous navigation and collaboration. In order to achieve this, current state-of-the-art localization approaches collect data provided by one or more agents and create a single, consistent localization map, maintained over time. However, with the introduction of lengthier sorties and the growing size of the environments, data transfers between the backend server where the global map is stored and the agents are becoming prohibitively large. While some existing methods partially address this issue by building compact summary maps, the data transfer from the agents to the backend can still easily become unmanageable.

In this paper, we propose a method that is designed to reduce the amount of data that needs to be transferred from the agent to the backend, functioning in large-scale, multi-session mapping scenarios. Our approach is based upon a landmark selection method that exploits information coming from multiple, possibly weak and correlated, landmark utility predictors; fused using learned feature coefficients. Such a selection yields a drastic reduction in data transfer while maintaining localization performance and the ability to efficiently summarize environments over time. We evaluate our approach on a data set that was autonomously collected in a dynamic indoor environment over a period of several months.

Published in:
Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, 2016
DOI: 10.1109/IROS.2016.7759673

1 Introduction

Recent developments in robotics have pushed forward numerous new applications, where robots can provide invaluable help, e.g. by reaching previously inaccessible places or performing tasks more precisely than humans. A vital requirement for these capabilities is to accurately estimate the robot's pose with respect to the global environment, as well as other agents. Often, robots need to operate in GPS-denied areas, without any embedded localization beacons. This motivates the need for precise, high-frequency pose estimation made possible by using visual-inertial sensors paired with a localization framework.

Many mapping and localization scenarios consist of repeated visits to the same environment by one or more agents. In a typical setting, a mapping agent creates a map of the current session locally [36, 178]. Such a map is then transferred to a mapping backend which collects data from all the agents that is used to build and maintain a single consistent map of the environment [63, 178]. When a certain place is revisited, new information may be observed and used to refine the map quality or fill possible gaps. Once an agent needs to localize itself against the existing global map, it can request the information from the backend and place recognition can be used to retrieve a current pose [7, 135, 147]. In the case of mobile robotics, we can assume that the information is exchanged using a wireless transmission system. Thus, there is a considerable limitation in the network's bandwidth, which affects both the backend-to-agent transmission of the localization map as well as the agent-to-backend transmission of the raw, local map.

To overcome these bandwidth limitations, we would like to reduce the data flow between the backend and the agent (and vice versa). Existing work has mostly aimed at reducing the size of the localization map, by creating a so called Summary Map, which is then sent to the agents [49, 50, 154] (illustrated by the top block in Fig. 6.1). These approaches take advantage of repeated visits to the same location and derive statistics out of the collected data in an offline processing step. Their goal is to select a set of landmarks that permits reliable place recognition while ensuring coverage of the entire environment and keeping the transfer size as small as possible.

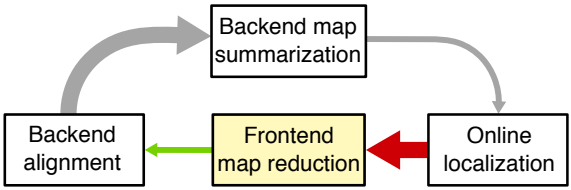


Figure 6.1: We propose a frontend map reduction method that can be used to reduce the data flow from the mapping agent to the mapping backend.

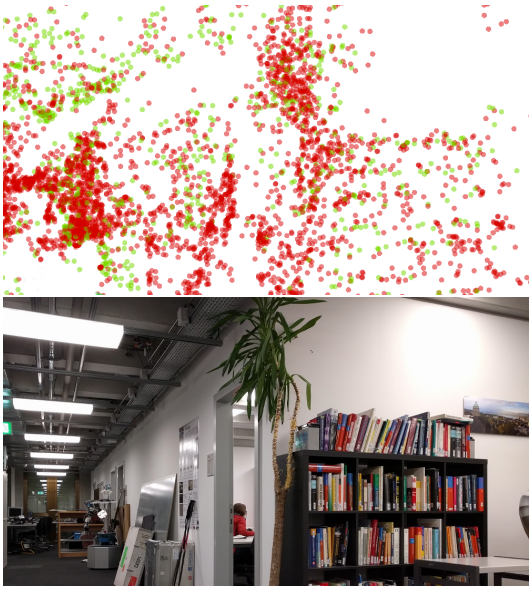


Figure 6.2: Above: the result of the proposed method – features expected to be stable are marked in green and feature expected to be volatile and hard to redetect are marked in red. It is worth to note that the algorithm considers the ceiling structure as a source reliable landmarks, but rejects objects on the floor and the plant, as visible in the photo of the same area (below).

In this paper, our primary goal is to reduce the data flow from the mapping agent to the mapping backend (illustrated by the bottom half of Fig. 6.1). When not reduced, those transfers can reach sizes of over 50MB per 100m trajectory in a system using relatively compact 512-bit visual descriptors. To limit this bandwidth burden, we propose a method that only requires data from the current mapping session to predict and select those landmarks which are most likely to be useful for localization in the long term. To perform the prediction, we have come up with several metrics, called predictors, that we deem to describe aspects of this long-term consistency. The method then learns the relative influence of each predictor based on training data, therefore combining numerous (sometimes weak) sources of information in order to infer the overall quality of each landmark.

The proposed methodology is evaluated using a dataset recorded by an autonomous robot in a highly-dynamic office space (depicted in Fig. 6.2). The data

was collected over a period of several months, with varying lighting conditions and numerous objects appearing and disappearing from the mapped area. The evaluation not only shows improvement over the state-of-the-art, but also provides insight into the relative merit of a variety of proposed landmark quality predictors (based on properties such as geometry, persistency, and appearance). By using the method, we are able to reduce the agent-backend data flow by 80% with a marginal localization recall drop of about 5%.

2 Related Work

Vision-based place recognition systems enable retrieval of precise 6-DoF pose information using relatively low-cost sensors, in GPS-denied environments, and without using any external beacons. Current state-of-the-art place recognition approaches range from using holistic image retrieval [149], over features obtained using deep-learning techniques [205], to local point features [73]. Working with local point features (landmarks), has the advantage of providing an intuitive framework for building up a 3D model of the environment, and these are therefore used in this work. This map can then serve as a database to localize from [185],[134].

In general, this process has a strong dependence on the quality of the underlying landmark detection and description. As a result, a range of work has been done to investigate and quantify the quality of visual descriptors. For example, in [33], the stability of feature descriptors over viewpoint and lighting changes is investigated, providing a measure of robustness. Similarly, the work of [221] focuses on learning repeatable detectors over drastic illumination changes. This work relies on training images capturing the same view-point over multiple illumination conditions.

Furthermore, in relation to localization, existing work has explored similar ideas of evaluating the quality of features (or landmarks) with respect to the stability and reliability of being detected repeatedly. For example, in [233] and [111] a subset of landmarks is selected based on the uniqueness of the descriptors. Additionally, the work of [85] evaluates the likelihood of a SIFT descriptor to be matched in subsequent observations, in a localization framework similar to the one presented in this paper.

In the context of robotics, and generally within the applications where the computational power is limited, binary descriptors such as BRISK [121] have gained popularity. While these descriptors are more compact, they also contain less information which reduces their discriminative power required for a robust matching. This motivates the need for methods that score the information content of such keypoints for localization, as proposed in this paper.

Perhaps the most similar work to that presented here is by Buoncompagni et al. [24], where image features are scored based on distinctiveness, repeatability, and detectability; then combined to give an overall saliency score. However, the weights used therein to combine these scores are hand-tuned, rather than learned automatically as we do here. Moreover, the work presented in [24] is applied to the task of object detection and recognition, and is therefore missing metrics specific

to long-term localization.

Applications to long-term mapping scenarios certainly promote the need for map maintenance and selection of only the most relevant landmarks. Existing work has mostly aimed at maintaining the map over long time-horizons [50], removing outdated information by using a change-detection metric [223] or aggregating the experiences to eventually encompass all possible conditions [35]. As state-of-the-art systems push to cover larger and larger areas over longer time-scales, there exists a growing interest in reducing the amount of data that needs to be stored or transferred [167]. One of the ideas is to compress the global model into compact representations, so-called Summary Maps [154], [49], effectively subsampled versions of the centrally maintained map that are useful for localization. These approaches, however, assume access to all past sessions and history, which is generally only possible on a centralized mapping server.

Our goal is to take inspiration from existing summarization methods and deploy them in an online fashion on the agent. As a consequence, we would only send the most valuable parts of the local map to the backend, reducing the required transmission bandwidth and processing time. The main contributions of the paper are as follows:

- A presentation and evaluation of novel metrics for predicting a landmark's relevance.
- An extension of the matchability prediction approach presented in [85] to binary descriptors.
- A general framework, which learns the relative influence of proposed landmark quality predictors, in order to rank sets of landmarks in an online fashion.
- An evaluation of the entire approach in a realistic mapping scenario using data collected autonomously, spanning over several months.
- An analysis of the possible compression ratios that can be used on the agent-side while avoiding a significant loss in localization performance.
- A study of the proposed method used as a part of the system presented in [50] to confirm its advantageous long-term properties – getting better over time.

3 Methodology

Our approach to reducing the number of landmarks while keeping the most relevant ones is twofold. On the agent-side, we obtain a ranking of current landmarks, which serves as a decision criteria for selecting the ones that are most suitable for being sent to the server. This approach is described in the first subsection. Afterwards, we revisit an Integer Linear Programming based approach for keeping informative

landmarks on the backend-side. This approach is then described in the second subsection.

3.1 Summarization on the agent-side

The merit of a given landmark for localization depends on various aspects. This work aims at exploring some of these by considering different features related to landmark observations in order to infer the respective significance of each landmark. Furthermore, by evaluating such features in conjunction, even relatively weak correlations in the data can be exploited to boost results. Thus, several evaluation metrics are proposed, and later combined in a regression framework. Using the results thereof, a landmark ranking policy can be defined such that only a subset of the most relevant landmarks is selected while still maintaining relocalization quality during future sessions.

Before evaluating the ranking features of each landmark i, we first define the following notation (further illustrated in Fig. 6.3):

- $p_{GL,i} = \left(p_{Gi,x}, p_{Gi,y}, p_{Gi,z}\right)$ denotes the position of the landmark i in the global frame G,
- $p_{GV,j}$ denotes the position of the keyframe j in the global frame G,
- $b_{i,j} = \dfrac{p_{GLi} - p_{GVj}}{\|p_{GLi} - p_{GVj}\|_2}$ denotes a unit-length bearing vector corresponding to the ray between the feature i and the keyframe j,
- $\mathcal{S}_i$ is an ordered set of keyframes observing the feature i, i.e. if $j \in \mathcal{S}_i$ then j observes i,
- $m_{i,j}$ is the the keypoint measurement of the keyframe j, which got associated with the landmark i,
- $m'_{i,j} = (m_{x,i,j}; m_{y,i,j})$ represents the reprojection of a landmark i into the keyframe j, using the corresponding projection and distortion models.

Predictor features

We will now present candidate features related to the observation of each landmark, which may aid in the prediction of its relevance with respect to localization. The goal is to model the probability of consistently redetecting and matching a landmark based on the local information, such as geometry and the descriptor pattern, from a single visit to its environment. We assume any long-term mapping data, such as previous landmark observations, is not available to us. The actual relationship between the proposed predictor variables and the empirical probability of redetecting the landmark will be evaluated in Section 4. The list of candidate features proposed in this paper is by no means complete, but we believe that it

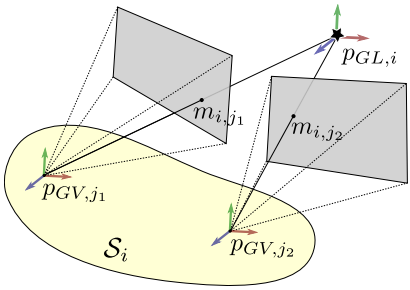


Figure 6.3: A landmark located at $p_{GL,i}$ is observed from a set $\mathcal{S}_i$ of keyframes located at $p_{GV,j}$. For each keyframe, the location of the landmark in the image plane is given as $m_{i,j}$.

captures a range of relevant metrics, and demonstrates the value of combining several sources of information (even weak ones) for landmark selection.

Track length: We will begin by looking at properties related to local tracking information about each landmark during the current traversal. Track length is one of the simplest and most intuitive features, widely used by existing map reduction methods [49, 154, 167]. Its underlying assumption is: the more frames landmark i is reobserved in, the more probable it is that it can be observed from a large area and that it can easily be redetected. It is given by

$$\phi_i^l = |\mathcal{S}_i| \tag{6.1}$$

Distance traveled while observing the landmark: This is a variant of the track length predictor, which accounts for effects such as varying velocity or keyframe selection and is obtained as

$$\phi_i^d = \sum_{j \in \mathcal{S}_i} \|p_{GV,j+1} - p_{GV,j}\|_2 \tag{6.2}$$

Distance traveled between the two most distant keyframes on a track: Similarly, this brings robustness against trajectories that are meandering and have a large length even though they only cover a relatively small area. We believe this measure is complementary to the ones presented above. Its computation requires maximization over all keyframes observing the same landmark, i.e.

$$\phi_i^\Delta = \max_{j,j' \in \mathcal{S}_i} \|p_{GV,j'} - p_{GV,j}\|_2 \tag{6.3}$$

Maximum angle between observation rays: Additionally, we propose to not only use the distance along the track, but also the maximum angle spanned by

all observation rays from the keyframes to the landmark. If this angle is relatively wide, the area where the landmark can be observed is likely to be large as well. As every $b_{i,j}$ is of unit length, the predictor can be efficiently calculated as a dot product:

$$\phi_i^b = \max_{j,j' \in \mathcal{S}_i} \cos^{-1} b_{i,j} \cdot b_{i,j'} \tag{6.4}$$

Mean reprojection error: Furthermore, apart from the landmark track geometry, it is also worth to consider the consistency of the map in the landmark's locality. The mean reprojection error of the landmark into all of the observing keyframes might be considered as a metric that contains additional information about the utility of a landmark. A landmark that is triangulated from many observers with distinct positions (given that it has no incorrect keyframe associations) is likely to have a small reprojection error.

$$\phi_i^\epsilon = \frac{\sum_{j \in \mathcal{S}_i} \|m_{i,j} - m'_{i,j}\|_2}{|\mathcal{S}_i|} \tag{6.5}$$

Gravity constraint: Our aim is to select not only those landmarks which have a high chance of being redetected and matched, but those which also tend to be stable over longer periods of time. We therefore propose a predictor based on what we refer to as the gravity constraint. It disambiguates between landmarks that belong to objects which are located *on* something (and not necessarily anchored) from objects which are *hanging* from something and must be anchored, otherwise they would fall down. The basic concept is illustrated in Fig. 6.4. We believe the objects that are not anchored, such as objects placed on a desk, will prove to be less reliable for long-term localization. Assuming g is the gravity vector pointing down:

$$\begin{aligned} g &= (0, 0, -1) \ , \quad \|g\|_2 = 1 \ , \\ \phi_i^g &= \frac{\sum_{j \in \mathcal{S}_i} b_{i,j} \cdot g}{|\mathcal{S}_i|} \end{aligned} \tag{6.6}$$

Vertical coordinate: In the same vein, we also include a predictor that relates to the specific position of each landmark in the mapped environment. The vertical coordinate of the landmark's position may provide additional information about its long-term stability. For example, in our office environment, objects that are either on the floor (e.g. parcels) or within a person's reach (e.g. things on the table) change frequently, while objects located higher up tend to be more stable. As this quantity might not be linear with the landmark quality, we expect to use some nonlinear transformation f before using this feature in our model (see Section 4.3).

$$\phi_i^h = f(p_{Gi,z}) \tag{6.7}$$

Descriptor appearance classification: Following the methodology presented in [85] for SIFT descriptors, we propose a similar method for classifying the BRISK

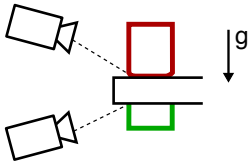


Figure 6.4: The idea behind the gravity constraint predictor is that clutter cannot fly. Objects seen from below (the green rectangle in the image) must be fixed or anchored, and are therefore expected to be relatively persistent. On the other hand, features seen from above (the red cup) are not necessarily anchored and can be easily displaced or removed from the environment.

descriptors using Random Forests. We believe that information about the appearance pattern will be complimentary to the other predictor variables that are based on map geometry. Instead of the binary classification proposed by Hartmann et al., we have formed 5 classes corresponding to the landmark quality. We used Gini impurity as a split criterion and 100 trees in total. To prevent overfitting, which is very likely with the binary data, we limited the depth of a tree to 14. The output of the Random Forest is transformed as follows to form a single scalar that can be fed to the regression framework:

$$\phi_i^{rf} = \sum_{c=1}^{5} c \cdot P(\text{class} = c) \tag{6.8}$$

Regression

We would like to combine all of the features presented above into a single, scalar landmark relevance score, that can be used to rank the landmarks according to the predicted quality and select a subset of them. One way of achieving this is by using a regression algorithm and fitting the coefficients based on a labeled training set.

The training set labeling: We propose to label the training set based on the past evidence of observing a particular landmark in multiple datasets covering the same trajectory over longer periods of time. Calculating the ratio of the number of datasets in which the landmark ℓ_i was observed over the total number of datasets $|\mathcal{D}|$ approximates the empirical probability of observing the landmark in a new dataset:

$$P(\ell_i|\mathcal{D}) \approx \frac{\text{\# of datasets observing the landmark}}{\text{total \# of datasets}} \tag{6.9}$$

The landmark correspondences between the datasets can be established using an appearance-based feature matching algorithm. It is worth noting that since all

the datasets were recorded when following the same trajectory, each portion of the environment is equally represented, and therefore persistent and reliable landmarks should be redetected in each traverse.

Feature selection: To confirm the true significance of the predictors presented above, we apply the Lasso regression method (see (6.10)). Depending on the value of λ, the method penalizes the absolute value of the coefficients β and might eventually set them to 0. We therefore cross-validate over the values of λ, look for the model with the highest prediction accuracy and observe which coefficients are not equal to zero. Assuming X is a matrix of predictors, y is the dependent variable and N is the number of samples, the Lasso regression can be formalized as:

$$\min_{\beta} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{6.10}$$

Regression model: After performing the feature selection, we fit our final model using ridge regression. This technique penalizes the L-2 norm (instead of L-1 in the Lasso method) of the coefficients and is suitable for handling collinearity in the predictor variables. This is beneficial here, as many of the proposed predictor variables are strongly correlated (e.g. the distance along the track with the maximum angle between the rays).

3.2 Summarization on the backend-side

While the summarization on the frontend-side mostly focuses on information immediately available to the agent, the backend-side may additionally incorporate information from multiple agents and earlier traversals through the same environment. For the backend landmark selection we used the method proposed in [49] which uses an Integer Linear Programming optimization to find the desired subset of landmarks. The problem can be described as:

$$\begin{aligned} \text{minimize } & \mathbf{q}^T \mathbf{x} + \lambda \mathbf{1}^T \boldsymbol{\zeta} \\ \text{subject to } & \mathbf{A}\mathbf{x} + \boldsymbol{\zeta} \geq b\mathbf{1} \\ & \sum_{i=1}^{N} \mathbf{x}_i = n_{desired} \end{aligned} \tag{6.11}$$

where $\mathbf{A}$ is the covisibility matrix, b is the desired number of landmarks per keyframe and $\boldsymbol{\zeta}$ is a slack variable. Each landmark is represented as a binary variable x_i and the goal is to find a subset of $n_{desired}$ landmarks. The landmark score $\mathbf{q}$ is based on the number of observing datasets, descriptor stability, as well as the within-dataset track length.

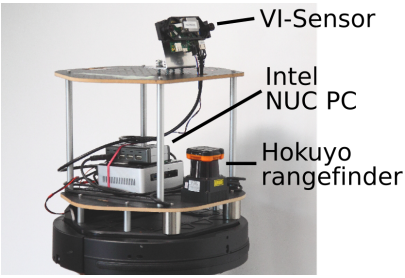


Figure 6.5: The Turtlebot robot used to acquire the datasets.

4 Experiments

In order to evaluate our methodology, we used a Turtlebot[1] robot, autonomously collecting datasets in an office environment. We first present the experimental setup (Sec. 4.1) followed by a discussion on the preparation of the training set (Sec. 4.2), and a subsequent evaluation of the regression model (Sec. 4.3). We compare the presented approach to other selection methods (Sec. 4.4) and evaluate the long-term performance of the integrated summarization system (Sec. 4.5).

4.1 Experimental setup

The data for all the experiments and evaluation has been collected using a Turtlebot robot, depicted in Fig. 6.5. The navigation and local obstacle avoidance used during data collection was based on the Hokuyo 2D laser rangefinder and the stock ROS navigation stack [139]. Additionally, the robot was equipped with a VI-Sensor [159] to build visual-inertial maps for the long-term map maintenance experiments. Mapping data was collected over a 150 m long trajectory, twice a day, over a period of several months. Since the datasets are recorded within a functioning office environment, there are many examples of dynamic objects that are often moving, appearing, or disappearing on a daily basis (see Fig. 6.2). In addition, the lighting conditions vary significantly for an indoor environment, depending greatly on the weather conditions and the artificial light inside the building.

The datasets were then fed into the mapping system described in [36], to perform evaluations and verify how the proposed agent-side map reduction method affects the short-term and long-term place recognition performance. For all of the following evaluations, we used a threshold of 40 cm when computing the precision of the pose retrieval. The ground-truth poses were obtained using a full-batch visual-

[1] www.turtlebot.com

Predictor	Coefficient	R^2 change if excluded
Track length (frames), ϕ_i^l	0.0050	−0.0011
Total distance, ϕ_i^d	0.0095	−0.0043
Max. distance, ϕ_i^Δ	0.0101	−0.0061
Max. angle, ϕ_i^b	0.1109	−0.0019
Mean reproj. error, ϕ_i^ϵ	−0.0040	−0.0001
The gravity constraint, ϕ_i^g	0.1954	−0.0334
Z-coordinate, ϕ_i^h	0.0174	−0.0307
BRISK random forest, ϕ_i^{rf}	0.1165	−0.0232

Table 6.1: The table shows the influence of the predictor variables in our regression model. As all predictors are normalized, large absolute values of coefficients imply stronger impact of the corresponding predictor. The change in the coefficient of determination (R^2) when excluding a specific predictor, provides an idea about the unique contribution of that predictor to the entire regression model. Thus, the larger the drop, the more unique the predictor's contribution is.

inertial bundle adjustment of the trajectories, as is often done when no external motion tracking system data is available [49, 135].

4.2 Training dataset

The proposed method is a data-driven approach, and therefore we need a training set which consists of multiple mapping sessions. We have used 31 maps recorded by the Turtlebot, aligned them together, identified commonly observed landmarks (using [134]), and jointly refined the maps using our visual-inertial least-squares optimization. Eventually, the landmarks in the dataset could be automatically labeled using the empirical probability measure described in Section 3.

Furthermore, the datasets were split into two pieces: a training part and an evaluation part. The sets of landmarks used for training and evaluation are therefore completely disjoint, but they still come from a similar environment (illustrated in Fig. 6.6). In this way, we can test how the proposed method generalizes e.g. within a single building.

4.3 Regression model

First, we examine the relationship between each feature and the landmark observation count, as depicted in Fig. 6.7. It can be seen that all the predictor variables are correlated with the landmark labeling and only the z-coordinate of the landmark exposes a strong non-monotonic relationship. This observation was confirmed by the cross-validation procedure of the Lasso regression, where only the coefficient for the landmark's z-coordinate variable was set to zero (at $\lambda = 4.249 \cdot 10^{-4}$).

Instead of just removing this predictor variable, we decided to add a non-linear transformation that will make the relationship between the landmark label and

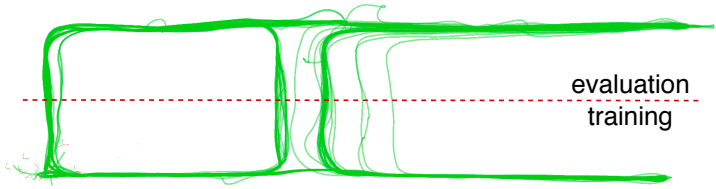


Figure 6.6: Split of the collected maps between the part used for fitting the regression parameters and for all the evaluations. The complete dataset consists of 31 trajectories, each about 150 m long.

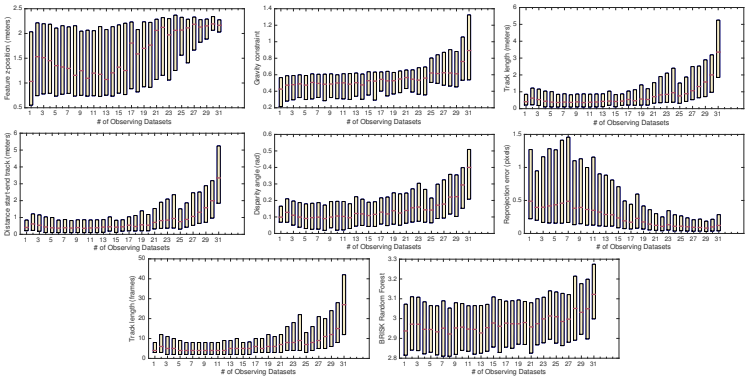


Figure 6.7: The relationship between the proposed predictor variables and the the training set labeling. Yellow bars represent the interquartile range and the violet marker is the median value. The only predictor variable that is showing a strong non-monotonic relationship with the labeling is the z-coordinate of the landmark position. We therefore propose to use a non-linear transformation of this predictor before fitting the regression coefficients.

its z-coordinate monotonic. We used $\phi_i^h = \max(p_{GLi,z}, \theta_z)$, with $\theta_z = 1.5$ m. While this might be perceived as a strongly hand-crafted feature, it actually has an interpretation: The landmarks located around $h = 1.5$ m often belong to dynamic objects, which is rarely true for the landmarks close to the ceiling. After applying the transformation, the Lasso regression did not reject any of the predictors and we were able to obtain the model coefficients using Ridge regression.

The Ridge regression algorithm was able to fit to the training model and reported a coefficient of determination $R^2 = 0.1265$, meaning that over 12% of the variance is explained by the predictor variables. While this result may seem low, we need to take into account that we are predicting a very complex phenomenon using only weakly correlated variables. Table 6.1, displays the values of each coefficient for the normalized predictors. We also include the change in the R^2 value after excluding each of the predictor variables – the larger the change is, the more information was brought by the predictor over all the other ones. We can notice that excluding the track length-related predictors is not causing a large change in R^2, as there is always another measure that captures similar properties. On the other hand, removing the Random Forest prediction or the gravity constraint reduces the R^2 significantly, suggesting these features provide an orthogonal source of information.

Finally, we also present the regression result against the evaluation set labeling in Fig. 6.8. The results suggest that we should be able to filter out the best landmarks with good precision (the first quartile value of the 31st class is larger than the third quartile value of the first 15 classes).

4.4 Comparison with other selection methods

The regression output can be used to rank the landmarks of the map and select a suitable subset for localization. We furthermore want to analyze how the place recognition performance is affected by the map reduction using the following approaches: the proposed method, random selection and the method based solely on the track length (which was used in [50]). The results of this evaluation are presented in Fig. 6.9. While all three methods perform similarly up to a 60% reduction, the differences are significant at reduction ratios of 80% and higher. Our proposed method outperforms both the random selection and the track length based methods, having an edge in the F1-score values of 10% for 90% landmark reduction and 30% for 98% landmark reduction when compared to the latter.

4.5 Iterative mapping and summarization

A backend designed for long-term mapping needs to incorporate new data from agents over time and iteratively resummarize the global map ensuring its size to stay bounded. In this context, it is interesting to investigate an optimal ratio between agent-side and backend-side summarization. In other words, we seek the optimal trade-off between minimal data upload to the backend while still providing enough information to create accurate localization maps of the environment.

For this reason, we have built a series of maps with varying agent-side and

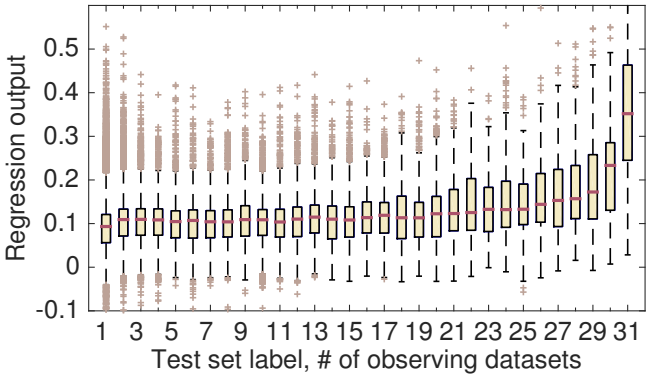


Figure 6.8: Regression results over the evaluation dataset. Yellow bars represent the interquartile range, the violet line is the median value, and beige crosses denote outliers. The regression model output, which corresponds to the predicted landmark quality, shows a significant correlation with the evaluation set labeling. Moreover, the distributions of the very good and very bad landmark score are separable – the Q1 value of the best landmarks is larger than the Q3 value of the worst.

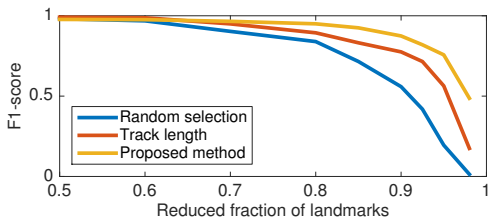


Figure 6.9: Comparison of place recognition performance based on a map from a single session. The landmarks of the map were reduced using different methods and up to varying degrees of compression. We use the F1-score to measure the performance, which is a typical data retrieval measure combining precision and recall.

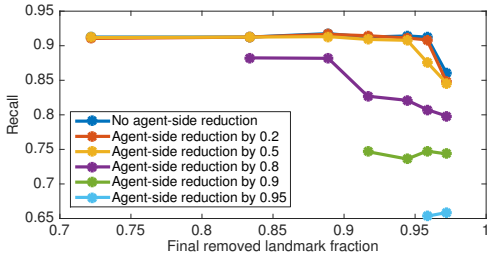


Figure 6.10: Place recognition recall values for different levels of summarization on the agent and in the backend. We can see that the agent-side summarization causes almost no recall drop when reducing the map by 50%. Even a more aggressive reduction, by 80%, causes a mere drop of 5% of the recall. This indicates that agent-side summarization can significantly reduce data uploads to the backend while maintaining similar place recognition quality.

backend-side summarization levels from a single dataset. A second dataset, recorded in the same environment, is used to localize against the built maps. An evaluation of the F1-score against the "fraction of retained landmarks", as shown in Fig. 6.10, indicates that an agent-side data reduction of up to 50% is feasible while only marginally affecting the place recognition recall. Even a reduction of 80%, which vastly reduces the required transmission bandwidth, only causes a drop of 5-10% in the F1-score, which may still be acceptable in many applications.

One fundamental question still remains unanswered. Can we incrementally build and localize from a map that was built by repeatedly merging in new agent data followed by a resummarization? To investigate this, we first apply agent-side summarization to a local agent map, then merge it into the existing global map of the backend. Next, the map is resummarized in the backend. This process is repeated for a set of 30 datasets as shown in Fig. 6.1.

Again, we evaluate the recall by localizing each mission against the current global map before merging it in. The results are presented in Fig. 6.11. It can be seen that the recall values stabilize within the region of 75-90% after gathering a sufficient set of stable landmarks. With this experiment, we validate that agent-side summarization can significantly reduce transferred data while still maintaining good localization performance.

5 Conclusions

In this paper we have presented an algorithm that selects a subset of landmarks which are more likely to be consistently redetected during subsequent localization attempts. The landmark selection procedure is based solely on locally available

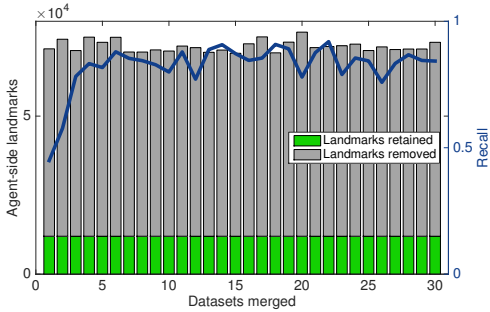


Figure 6.11: The proposed agent-side summarization combined with backend summarization yields good place recognition performance at reduced data transfer. The map on the agent-side is reduced to 12,000 landmarks (about 20%) after each session and then used to build the global map in the backend, limited to 6,000 landmarks. Initially, the *experience* of the mapping sessions is summarized and the recall results are improving to reach the 75-90% range and stabilize.

map data, rather than requiring information gained only after several visits to an area. By relying on this reduced subset of landmarks, the amount of data transferred between agents and the map backend can be throttled without corrupting localization. The proposed method scores landmarks by fusing a combination of novel and existing predictor variables using coefficients provided by a regression framework. The approach was evaluated in a long-term, iterative mapping scenario, using data collected by our autonomous office mapping platform. We show that we can drastically reduce the data transfers (by about 80% for agent to backend) while maintaining comparable localization results. In addition, we show that our method performs well in an iterative mapping process, leveraging long-term experience, and integrating easily with the backend summarization methods of [49]. This data-driven approach brings a benefit over hand-crafted variable fusion and also allows us to learn which variables are the most informative. In our experiments, we show that the newly introduced gravity constraint feature and the maximum angle between keyframe-landmark rays, as well as the proposed binary descriptor classification have the largest influence. While some of the proposed predictor variables may be environment specific, the presented methodology is generalizable to different environments.

In future work we would like to identify additional landmark quality predictors and deploy the system in a large-scale, multi-agent mapping application. We also believe that modern machine learning techniques such as convolutional neural networks employed directly on the image data, may provide complementary predictors to the currently proposed features. We can furthermore imagine learning

and maintaining a set of place-specific regression coefficients.

6 Acknowledgments

We would like to thank Mathias Gehrig for the preparation of the Turtlebot, our autonomous mapping platform. The research leading to these results has received funding from Google's project Tango.

Paper IV

Will it last? Learning stable features for long-term visual localization

Marcin Dymczyk, Elena Stumm, Juan Nieto, Roland Siegwart, Igor Gilitschenski

Abstract

An increasing number of simultaneous localization and mapping (SLAM) systems are using appearance-based localization to improve the quality of pose estimates. However, with the growing time-spans and size of the areas we want to cover, appearance-based maps are often becoming too large to handle and are consisting of features that are not always reliable for localization purposes.

This paper presents a method for selecting map features that are persistent over time and thus suited for long-term localization. Our methodology relies on a CNN classifier based on image patches and depth maps for recognizing which features are suitable for life-long matchability. Thus, the classifier not only considers the appearance of a feature but also takes into account its expected lifetime. As a result, our feature selection approach produces more compact maps with a high fraction of temporally-stable features compared to the current state-of-the-art, while rejecting unstable features that typically harm localization. Our approach is validated on indoor and outdoor datasets, that span over a period of several months.

Published in:
3D Vision (3DV), 2016 Fourth International Conference on, 2016
DOI: 10.1109/3DV.2016.66

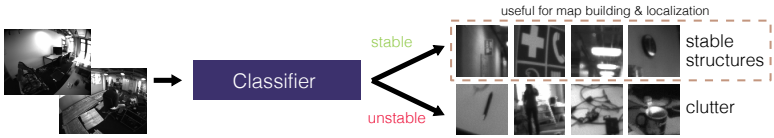


Figure 7.1: An overview of the proposed algorithm: A classifier is used to decide whether a particular visual feature is expected to be persistent or not. Our method uses full image information as input and helps to maintain compact stable-over-time maps that can be used for life-long localization.

1 Introduction

Generating sparse 3D models from camera images is a key component of state-of-the-art simultaneous localization and mapping (SLAM) systems [43]. Usually, mapping relies on detecting image features, *i.e.* distinctive points in an image, tracking them, triangulating 3D landmarks and characterizing them using descriptors. Recent approaches tend to use such 3D models of a given area to perform visual localization and improve pose estimates [147] [135].

Localization systems rely on matching points from a query frame to the previously constructed 3D model [185]. As time passes and the map gets older, however, the matching becomes more and more difficult as the scenery is subject to appearance changes. In this paper, we consider life-long mapping scenarios, where the visual appearance can change due to lighting, seasonal, or visual and structural changes. Additionally, with the growing time-scale of the mapping scenario, the amount of data that needs to be stored, processed, or transferred becomes prohibitive. Figure 7.2 (a) presents a dataflow diagram of a mapping system where maps are transmitted from the frontend to the backend. We would like to reduce this stream of maps by means of online feature selection, as in Figure 7.2 (b).

Existing approaches to feature selection are not well suited for the life-long scenario we consider. Some approaches use the past data to find features which are reliably redetected [167], but ignore their actual semantic meaning. Other methods attempt to classify hand-crafted descriptors [85] to guarantee a high probability of matching. But even if a feature is well-recognizable and a good matching seems likely, it might be unreliable for life-long localization, *e.g.* , because it belongs to an object that is moved frequently or may even entirely disappear (such as a pen or a mug on a desk, see Figure 7.1).

This paper introduces a learning-based approach for the problem of life-long feature selection. Our method is based on a classifier of raw image patches and depth maps and tries to grasp not only the *matchability*, but also a semantic meaning of a particular 3D landmark. Specifically, our contributions are:

- We propose a classifier framework to select appearance-based map features for long-term localization purposes using raw image and depth information.

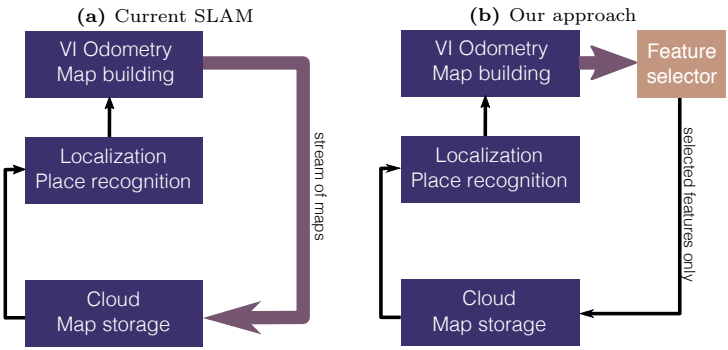


Figure 7.2: *Left:* The diagram presents the current flow in a typical mapping and localization system [147]. An agent streams mapping data (either raw image stream or visual-inertial odometry output) to the backend. The backend processes the maps and merges them into a single consistent global map which is then provided back to the agent(s) for localization. *Right:* The proposed data flow. Useful features can be selected on the agent's side which significantly reduces the amount of the data transmitted back to the backend.

- Resulting feature selection is shown to implicitly handle semantics, such as vegetation vs. buildings outdoors, or stable structures vs. clutter indoors.
- Maps obtained with our strategy allow for superior localization performance compared to state-of-the-art methods.
- We can significantly reduce the number of features in our map (*e.g.* by 80%) and still correctly register most of the query frames (75% of frames, about 20% better than state-of-the-art).
- We show that our method generalizes well, by deploying it in environments with significantly different appearances than the training environment and on a public long-term vision dataset.

The remainder of this paper is structured as follows. In Section 2, we discuss related work for appearance-based SLAM and feature selection approaches. In Section 3, we provide a problem definition. Our methodology is explained in Section 4 by describing the architecture of our classifier and discussing the general concept of the underlying ground truth training data. Setup of the experiments is discussed and their results are presented in Section 5. The work is concluded and discussed in Section 6.

2 Related Work

When extracting keypoints from images, SLAM systems typically rely on well-known algorithms for detecting distinctive points, such as the Harris corner detector [84] or a Difference of Gaussians (DoG) [131] based approach. These algorithms are crucial as they ensure that a potentially interesting point may be redetected in another image. One of the main goals of keypoint descriptors, such as SIFT [131], BRISK [121], or FREAK [3] is to obtain a description of the neighborhood of a keypoint that is viewpoint, scale and lighting invariant. Unfortunately, these algorithms do not consider suitability of a detected keypoint for use in long-term matching, and thus, too many partially irrelevant features are typically stored.

There are several approaches addressing this problem in current SLAM literature. In [35], it is proposed to store several maps of the same area covering different appearances. The advantage of this approach is that it contains very rich information about the environment. This comes at the cost of increased data quantity (including partially redundant information) and the need for introducing additional logic for handling different map variants [128]. A method for learning place-specific classifiers for identifying distinctive landmarks was proposed in [144] and subsequently improved in [129]. While these approaches improve localization quality between different appearance conditions, they require training location-specific classifiers and cannot be applied to previously unseen environments for reducing the amount of stored data at the time of initial recording.

Early consideration of life-long mapping can be found in [112], [113] where an incremental update scheme for an existing map is proposed. There, relocalization under different viewing conditions is achieved by storing redundant information and relying on particular types of map representation. Reducing map size by removing rarely observed features from the mapping backend was proposed in [167] and [49]. Even in a system that involves this type of map reduction, the present work serves as a useful additional component because it reduces the amount of data transferred to the backend and does not require multiple visits to the same environment for quantifying feature usefulness.

The work presented in [217] compares SIFT and SURF features, proposes the use of higher resolution images, and suggests introducing additional geometric constraints for localization. *Rosen et al.* [179] derive a stochastic filter that uses *Survival Analysis* [93] for modeling the lifetime of a feature. While that work is useful for map maintenance, it does not aim at providing an a priori feature quality criterion. In the work by *Hartmann et al.* [85], a random forest is trained to predict the matchability of hand-crafted SIFT descriptors. That is, the classifier predicts feature distinctiveness for better matching between consecutive frames but does not detect whether a feature (with a potentially high *matchability* score) might belong to a dynamic object or have an unstable appearance over time. In contrast, our approach captures these aspects as the training procedure learns directly on image patches and the groundtruth contains multiple visits to the same environment over an extended period of time.

Furthermore, there are several approaches that assume existence of additional

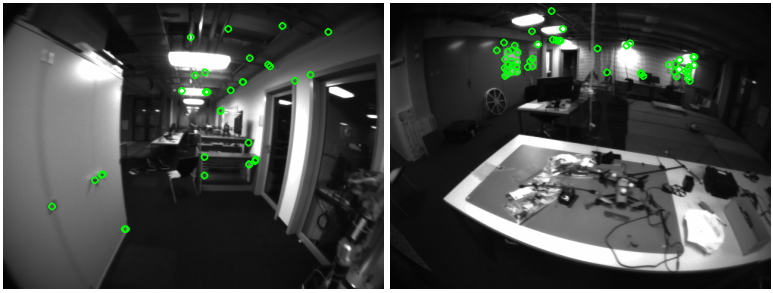


Figure 7.3: Two sample frames from our indoors dataset. The landmarks that were reliably redetected every time are marked using green circles. Those landmarks generally belong to constant elements of the environment, such as ceiling, door frames or heavy furniture. Our aim is to train a classifier that will be able to pick these useful features.

geo-location information for performing a feature quality assessment and a potential reduction of the number of features. Similarly to [85], the approach presented by *Kim et al.* [108] aims at detecting features that will have high matching score for obtaining good geo-localization. *Turcot and Lowe* [216] propose an approach that requires several different views of a scenery to be present in order to asses the quality of the features therein. *Knopp et al.* [111] propose the elimination of confusing features by counting the number of false matches within an image retrieval framework. The two latter approaches are thus not applicable to choosing stable features when there is no prior information about the current scenery.

3 Predicting feature reliability

Our goal is to obtain a subset of 3D landmarks of an appearance-based map that will provide as much information as possible for reliable localization and place recognition in a life-long scenario. The task of selecting those reliable, persistent localization features is not simple even for humans. We could claim that static objects in the environment can generally serve as useful landmarks, but it is not intuitively clear which of these will be redetected or matched correctly in the future, as illustrated in Figure 7.3.

The decision whether a feature is suitable for long time-span localization depends on three criteria:

- First, how well can this feature be matched in different images?

- Second, is this feature robust against changes in the environment?
- Third, will this feature remain in the environment for a longer period of time?

These criteria can be formulated as a classification problem in which detected features have to be classified as "stable" or "unstable" for life-long relocalization.

The classification task we have formulated above is extremely challenging to model due to the number of variables that influence changes in objects. Therefore we have opted for a data-driven approach by proposing a method that is based on recent advances in deep learning techniques. Particularly, convolutional neural networks (CNNs) have proven to be highly-successful in solving various computer vision challenges, such as object recognition [182] or video classification [105].

Our approach to select reliable localization features is based directly on the raw image information instead of hand-crafted detectors or descriptors that were previosuly used. This makes it possible to detect whether a feature belongs to a potentially dynamic object. Additionally, we do not have an off-the-shelf model and would like to learn what constitutes a stable landmark from the past experiences. We propose the use of a CNN-based classifier that uses a local keypoint neighborhood and optionally depth information as inputs and outputs a probability that the feature is "stable". We expect the depth maps to provide additional information about the 3D geometry of the scene. This might help to avoid *e.g.* occlusion corners, which yield unstable descriptors over viewpoint changes. Ground truth labeling of features in training data is based upon the redetection statistics that were collected over an extended period of time. This ensures potentially good matchable descriptors to be classified as unsuitable if the underlying feature belongs to an object that may move or disappear over time.

4 Methodology

The proposed CNN is trained using a set of labeled data pairs $\{image, label\}$ or triplets $\{image, depth\ map, label\}$. We consider the image patch around the keypoint as a main source of information, providing cues of how the keypoint looks like and what type of object it potentially belongs to.

The details of the entire process are presented in the subsections below, that cover the training set formulation 4.1, network architecture 4.2 and the training process 4.3.

4.1 Obtaining Training Datasets

Raw image patches: In this paper, we assume we are dealing with a visual-landmark-based mapping and localization system. In such a system, sparse point features are tracked over consecutive frames and used to triangulate 3D landmarks. We can therefore extract a patch with the local neighborhood of a feature and use it as an input to the network. The actual neighborhood size should depend on

the camera resolution, focal length of the lens, feature scale and the environment properties.

Depth maps: The depth maps are extracted from the clusters of camera frames using block matching and planar [71] or polar [172] rectification, depending on the type of motion. An additional bilateral filtering step [211] with the similarity function defined on the raw image is used to densify the depth maps. This method fills the gaps in the depth map while preserving the edges. Similarly as for image patches, a local neighborhood of the feature is extracted from the depth map.

Ground-truth labeling: We have decided to use binary labeling ("stable" and "unstable") of the features based on the empirical probability of reobserving them. The probability can be estimated from multiple co-registered datasets with the same trajectory, similarly to [101]. In general, we can say that the features that get merged across most of the datasets:

- are most likely stable objects as they are consistently re-detected over a long period of time,
- maintain a stable appearance under lighting and viewpoint changes as the datasets are recorded during different times of the day and features are observed from multiple angles,
- have beneficial descriptor pattern that gets easily and uniquely matched.

For each landmark, we compute a score

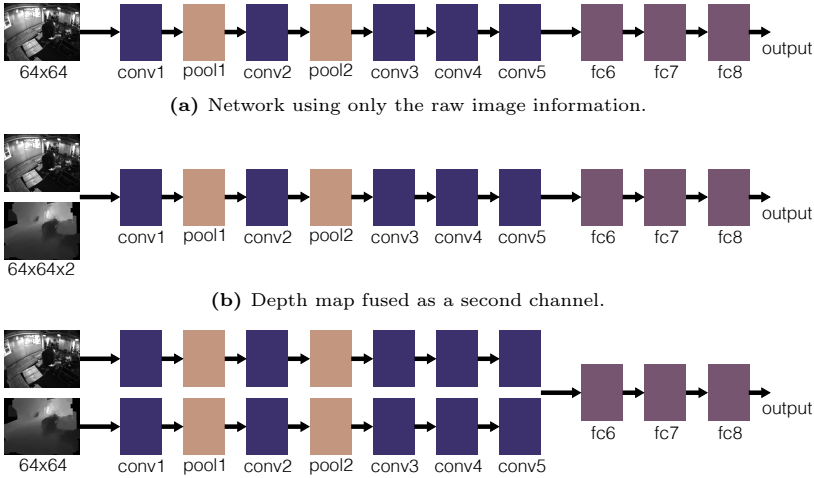
$$s = \frac{\text{number of datasets the landmark was observed in}}{\text{total number of datasets in the databse}}$$

and then label the landmark as "stable" if $s \geq 0.5$ (and as "unstable" otherwise).

4.2 Convolutional Neural Network Architecture

The proposed network architecture is based on the popular AlexNet network [116], which demonstrated very good classification performance while being computationally tractable on a desktop PC with a modern GPU (both in terms of training time and GPU memory requirements). There, we chose the last fully-connected layer to have 2 outputs, where each denotes a probability that the input belongs to either the "stable" or "unstable" class. This network is using a Softmax loss function, suitable for classification tasks. The additional depth information can be fused either as a second image channel or in a separate convolutional layer pipeline. The proposed network architectures are presented in Figure 7.4.

To classify a 3D landmark in the map that should be compressed, we calculate an output of the network for each of its observations (so for all the image patches observing this landmark) and then calculate a median probability of belonging to the "stable" class. If we need to select n best landmarks, we can simply use the "stable" class probability as the scoring function.



(a) Network using only the raw image information.

(b) Depth map fused as a second channel.

(c) Depth map with separate convolutional layers.

Figure 7.4: Three CNN configurations that were evaluated. In b) and c), the depth map information in included.

4.3 Training process

As our training set labeling might contain errors (e.g. a potentially "stable" landmark could have been classified as "unstable" one due to temporary occlusions) and the appearance-based landmark quality in general cannot be rigorously defined, we decided to use fine-tuning [46]. This technique suggests to pre-train the network on a well-labeled, large dataset (e.g. ImageNet) and only later, after reducing the network learning rate, try to fine-tune the network to the actual task. *Karayev et al.* [103] state that pre-training a network allows us to reuse the mid-level features learned from the object classification datasets and is generally superior to hand-tuned features.

While pre-trained models for ImageNet object classification are publicly available (e.g. Caffe's Model Zoo [98]), there seems to be no existing pre-trained model corresponding to our input parameters (small input image size, grayscale). Our approach is to train a network on the ImageNet dataset adapted to our needs, by scaling down the original ImageNet images and converting them to grayscale. The training performed on this small ImageNet dataset then provides us with a pre-trained network. In the final step, we fine-tune this network by learning on the labeled patch data from the mapping system.

5 Experiments

Our experimental setup and the datasets used for evaluations are presented in 5.1 and 5.2, respectively. In 5.3, we evaluate the network's output on the test set of "stable" and "unstable" labeled patches. We also verify the consistency of network predictions for patches of the same landmark (*i.e.* for multiple observations along a track of the same real-world feature). The second part of the evaluation is aimed at verifying the influence of our proposed method of selecting a subset of map features on the place recognition results (see 5.4). We compare our place recognition precision-recall results with the results of using other appearance-based methods for feature selection and map compression. Additionally, in Subsection 5.5, we measured the network's forward propagation time on a CPU and a GPU.

5.1 Experimental Setup

The **Lab** and **Hall** datasets were recorded with a visual-inertial sensor incorporating a low-cost IMU and a monocular, grayscale camera with a resolution of 720x480 pixels and a wide field-of-view. Visual-inertial odometry with a translational error in the range of 1% of the distance traveled (similar to [152]) was used to construct the maps. Keypoints were detected using a Difference of Gaussians (DoG) keypoint detector. These keypoints were characterized using FREAK binary descriptors [3] to co-register the maps. The keypoint locations were used to extract the raw image patches and depth map patches. Random cropping of patches (the network takes 64×64 patches cropped out of 68×68 keypoint patches) and mirroring them helped to augment the training set and improve the robustness towards viewpoint and orientation changes.

The Caffe [98] deep learning framework was used for CNN training and image patch classification. Both learning and classification was performed on a desktop PC with an Intel i7-920 (@ 2.66 GHz) CPU and an NVIDIA GeForce GTX980 GPU.

5.2 The Datasets

For evaluation we use 3 different sets of datasets with different scene characteristics, but all containing highly dynamic parts of the scene.

Lab datasets consist of 27 trajectories along the same path, of about 150m, in an indoor research lab environment with rather confined spaces (see Figure 7.3). The datasets were recorded over a period of 3 months. A total of 25 trajectories were used to construct the training set while 2 (recorded 2 months apart) were used for evaluation.

Hall datasets consist of 14 trajectories walking the same path, of about 100m, in a large university hall. The space is usually crowded, it also contains student working places with many dynamic objects. The datasets were recorded over a period of 2 weeks. A total of 12 trajectories were used for training and the remaining 2 (recorded 10 days apart) for evaluation.

We have also used the public University of Michigan **NCLT** datasets [32]. We have selected an area that was covered by 19 datasets in total (from $(x, y) = (-301, -448)$ to $(-183, -432)$ according to the ground-truth annotation), collected over 15 months. We have used only the front-facing camera of the Ladybug sensor. A total of 17 of the trajectories were used for training and the remaining 2 for evaluation (2012-08-20 and 2012-03-17).

The training datasets were optimized by applying a visual-inertial weighted least-squares optimization to each run individually, minimizing the reprojection errors and inertial residuals. Positions of the landmarks were included in the optimization procedure. Afterwards loop-closure was performed using a place recognition algorithm similar to [134][95]. As a result of this procedure, we obtained a map with co-registered trajectories. Based on the output of the loop-closure engine, we also partially merged features. The final map contains information on the number of runs and the number of frames per run in which a feature was visible.

The two evaluation datasets were used to verify the place recognition precision and recall (see Section 5.4). From the first dataset, a reference map was created and reduced (in terms of number of features) to a desired degree using the proposed method. The second dataset was used as a query dataset, *i.e.* we attempted to localize each of its keyframes against the reduced map dataset.

The localization evaluation requires accurate ground-truth poses so we needed to precisely align the pairs of datasets in a common global frame. We used our loop-closure system to find candidate 2D-3D matches between the two evaluation datasets. The matches were then filtered using a geometric-verification method (P3P pose solver [74]). Using the feature correspondences, the maps were co-registered. To further refine the map quality, a joint visual-inertial weighted least squared optimization was performed.

5.3 Evaluating Network Output

As a first step of evaluating our method, we have verified the binary classification accuracy of the proposed method and the three network architectures introduced in Section 4.2. The results are presented in Table 7.1. For both the Lab and NCLT datasets, depth information provides significant improvement over using

Table 7.1: Comparison of validation set binary classification accuracy for Lab and NCLT datasets and three network configurations: 1) using only a raw image patch 2) raw image patch stacked with depth as a two-channel image 3) raw image and depth propagated through convolutional layers separately and combined in fully-connected layers.

	Lab dataset	NCLT dataset
raw image patch	65.2%	69.5%
+ depth as a 2nd channel	67.4%	71.7%
+ depth in a FC-layer	71.2%	73.1%

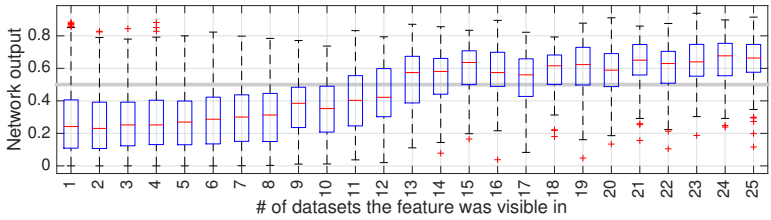


Figure 7.5: Illustration of the relationship between the statistics of the classifier output and the number of datasets a particular feature was observed in (Lab). There exists a clear positive correlation of the network output with number of datasets observing a landmark. The median output crosses the 0.5 level between 12 and 13 datasets which matches the labels used for learning.

solely the raw image patches. Additionally, fusing the depth map information in a fully-connected layer level yields the highest accuracy, which might be contributed to the fact that it is easier to optimize the convolutional filters separately. All the subsequent evaluations will assume this network architecture.

We have also verified the relationship of the network predictions with the ground-truth labeling of the test set. The results are visualized in Figure 7.5. It shows the relationship between the ground truth labeling of our training set (given by the number of runs in which a landmark has been viewed) and the resulting classification scores on the evaluation set. All landmarks that have a score above 0.5 are classified as suitable for life-long localization. From this plot, it can be seen how, *on average*, image patches that contain "stable" landmarks (*i.e.* , landmarks visible in more than 12 runs) are classified as good and vice versa.

We have furthermore evaluated how consistent the network output is for multiple observations over different runs of a single real-world landmark. That is, we want to know how the scores of our classifier differ for different observations of the same feature. Ideally, all patches of a single landmark should yield scores which are close to each other. Figure 7.6 shows the distribution of zero-mean landmark scores for a map with 100,000 features. The results show that the output of the network is consistent for a single landmark, most often associating it with the same class.

A further evaluation provides a more intuitive understanding of the network output. Figure 7.7 shows an example of a frame with classified keypoints from the Lab dataset. Figure 7.8 presents the best and the worst features in NCLT and Lab datasets according to our proposed classifier. We can state that the network implicitly learns the semantic meaning of the patches, corresponding to which objects tend to be "stable" and "unstable". This matches our initial proposal to use the semantic information and also means that our training set actually conveyed the notion of stability.

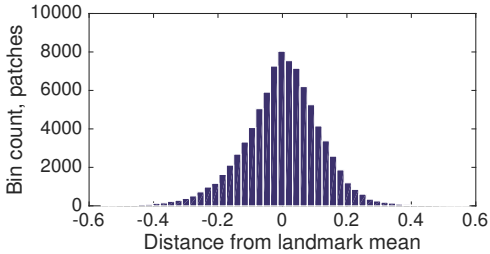


Figure 7.6: Distribution of classifier outputs after subtracting the feature-dependent mean from the classifier output. Most of the values are in the range of $[-0.2, 0.2]$ which suggests that our classifier yields reproducible results under different viewing conditions of a landmark.

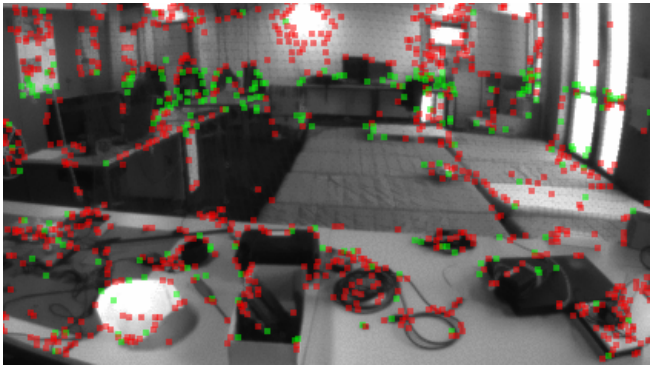


Figure 7.7: A single frame and a list of detected keypoints processed by our method. Features in the foreground belong mostly to tools and hardware pieces that were displaced almost every day and do not provide reliable place recognition clues. Features classified as "stable" are most often parts of stable objects (windows, furniture, poster on the left). It is worth noting that features belonging to the saturated region on the wall opposite (a reflection of a ceiling lamp) are also rejected.

5.4 Evaluating Localization Quality

In the previous section, we demonstrated that the network output is correlated with our ground-truth labeling and that the patch scores correspond to our expectations. This section evaluates how the localization quality depends on the selected subset of features and its size.

The key idea is to use one of the evaluation datasets (*map dataset*) for creating a reduced map using the proposed methodology. Then, we try to register each keyframe of the second evaluation set (*query dataset*) against the map dataset. We compare the resulting 6 DoF pose with ground-truth information. If the pose error is within the tolerance limits (0.07 m for position, 1.0 deg for rotation), then the localization is considered as successful. This is used to verify how the results of the place recognition engine change when we reduce the number of features in the map (while keeping features with higher classifier score).

We compare the results of our method to three other approaches that assign a score to a keypoint only based on the visual information of its neighborhood. These methods are briefly described below.

Random selection: Landmarks are assigned a score that is obtained by random sampling from a uniform distribution in range [0, 1]. Random selection provides a very good baseline to evaluate feature selection methods. If a certain method performs better than random, then we can claim that the criteria it uses are indeed correlated with the phenomenon we try to model.

DoG score: Our mapping system uses Difference of Gaussians (DoG) as a keypoint detector. Therefore it makes sense to use DoG response as a very basic feature selection method. Intuitively, the more salient the feature, the more probable it is that we will redetect it in the next run over the same area. We need to consider the limitations of this method: it cannot grasp any life-long feature behavior or actual feature matchability.

Predicting Matchability score: We also used a method that was proposed in [85]. It is expected to select well-matchable and uniquely-matchable features. As already mentioned when discussing related work, this method is expected to perform worse in comparison to our approach as it does not consider long-term feature visibility.

The results of the comparison on the Lab dataset are visualized in Figure 7.9. In this visualization, we used F_1 score, a popular data retrieval metric. It combines the values of precision and recall by computing a harmonic mean of the two:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Our approach outperforms the comparison methods and still yields a good trade-off between precision and recall even after removing 80% of the worst (in terms of classifier score) features (F_1 score of 0.75 for our method, 0.55 for state-of-the-art [85]). A top-down view of the localization results is presented in Figure 7.10.

So far, we have proven that we are able to learn the environment characteristics and then successfully predict the stability and matchability of visual features for

new maps of the area. While this result seems to be already useful, proving a certain level of generalization would be advantageous. We have therefore evaluated the performance of the suggested approach on the other datasets, Hall and NCLT. We have compared the classifier trained on the Lab dataset with the classifiers trained on the evaluated environments.

Figure 7.11 and Figure 7.12 present the localization F_1 scores of this evaluation. Even when trained in another environment, the proposed method still works better than the second-best Predicting Matchability [85]. In Figure 7.11, our approach shows only a small retrieval loss when trained on another indoor dataset, which means it can generalize withing similar environments. In Figure 7.12, one can notice that the gap between Predicting Matchability and our classifier trained on the Lab dataset is relatively small. This is in line with intuition as, due to the change of the environment from indoors to outdoors, our method can no longer predict long-term stable features as well as in the area it was trained on.

5.5 Runtime evaluation

The computation time for evaluating the quality of a single landmark patch is dependent on whether the network is evaluated on a CPU or a GPU. On our CPU, the mean duration of classification was 17.2 ms (with a standard deviation of 4.7 ms). When using a GPU this time goes down to 2.7 ms (with a standard deviation of 1.7 ms). Our results show that the approach is applicable for real-time feature classification at interactive frame rates.

6 Conclusion

This paper presents a method for assessing the suitability of features for life-long localization. Our proposed approach uses a CNN classifier that is trained on local image patches and depth images extracted from image features over multiple runs through the same environement, at different times, over a period of several months. Thus, the classifier evaluates the quality of a feature by considering both its distinctiveness and its long-term visibility, discarding seemingly good features that stem from non-persistent objects.

By using the proposed method, we were able to reduce the map size by over 70% while still being able to register around 80% of the relevant keyframes. Therefore, our classifier can significantly limit the map size, reducing the storage and data transfer requirements in typical mapping scenarios.

Our method provides better localization results than other feature selection methods, even for a completely new environment that was not present in the training set. An inspection of the classifier output suggests that our training data has captured a notion of feature stability, intuitively understood by object semantics. In future work, we find it would be interesting to exploit environment semantics in more detail, evaluate different network architectures or leverage the 3D context for better classification.

Acknowledgments

The research leading to these results has received funding from Google's project Tango.

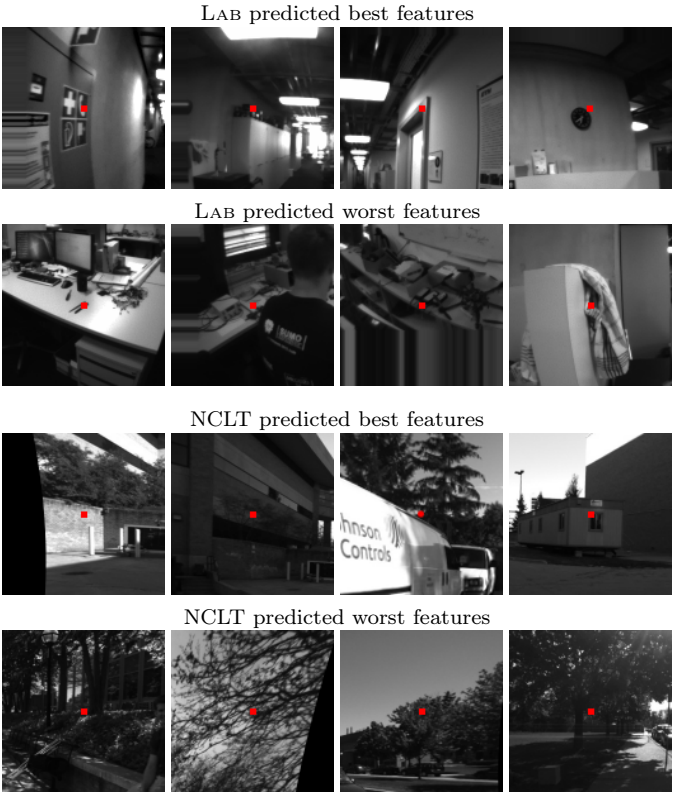


Figure 7.8: Image patches from the LAB and NCLT datasets that were classified as the best/worst by our method. The selection corresponds with human intuition: permanent structures with strong corners provide reliable localization cues while the clutter of dynamic objects (indoors) or vegetation (outdoors) should be rather rejected.

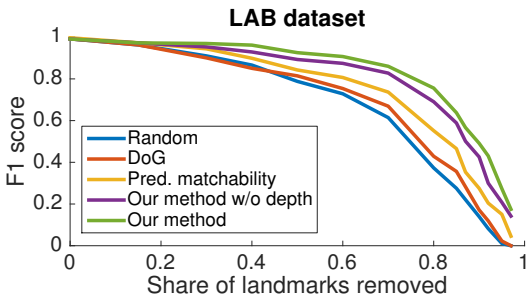


Figure 7.9: F_1 score curves comparing our proposed method with other baseline methods described in 5.4. The higher the F_1 score value for a given feature reduction number, the more keyframes were successfully localized (with fixed tolerance values). We can see that using the DoG response brings only a slight advantage over random feature selection, as it does not include descriptor uniqueness and long term feature stability. The plot also shows that our method performs best, especially at high map reduction values. We can reduce map by about 80% and still be able to register about 75% of query keyframes against it. Including the depth information brings a benefit over using the raw image only.

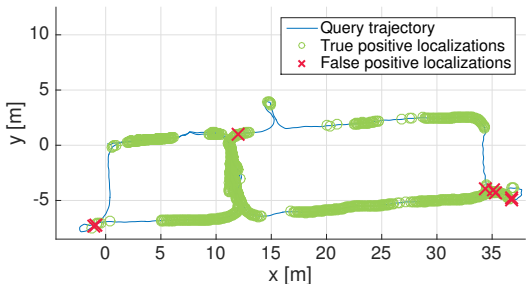


Figure 7.10: Top down plot of the LAB dataset presenting the localization results using a map reduced by the proposed method. The 3D features of the map were reduced by 80%, leaving about 35,000 features out of 180,000. The localization system returned 792 correct localizations and only 8 false positives.

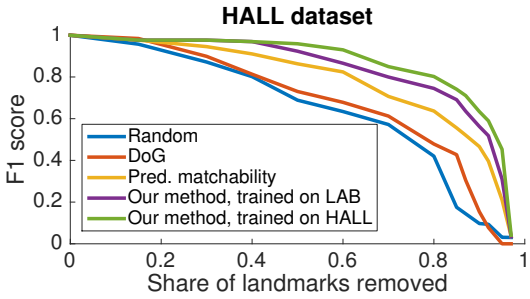


Figure 7.11: F_1 score curves comparing our proposed method with other baseline methods on an indoor HALL dataset. The results show that our approach generalizes well to other (but still indoor) environments – the classifier trained on the LAB dataset still works better than other state-of-the-art methods on the HALL data.

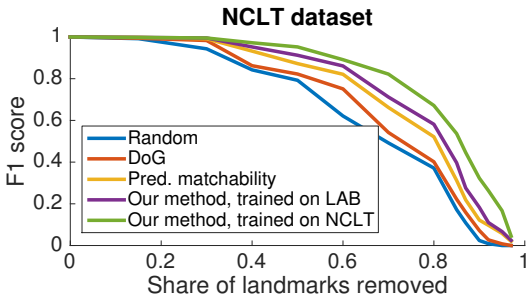


Figure 7.12: F_1 score curves comparing our proposed method with other baseline methods on an outdoor dataset from NCLT. Our method trained on the same dataset yields best results. Using the suggested classifier trained on the indoor LAB datasets still outperforms the other baseline methods.

Paper V

maplab: An open framework for research in visual-inertial mapping and localization

Thomas Schneider*, Marcin Dymczyk*, Marius Fehr*, Kevin Egger, Simon Lynen, Igor Gilitschenski, Roland Siegwart (*contributed equally)

Abstract

Robust and accurate visual-inertial estimation is crucial to many of today's challenges in robotics. Being able to localize against a prior map and obtain accurate and drift-free pose estimates can push the applicability of such systems even further. Most of the currently available solutions, however, either focus on a single session use case, lack localization capabilities, or do not provide an end-to-end pipeline. We believe that only a complete system, combining state-of-the-art algorithms, scalable multisession mapping tools, and a flexible user interface, can become an efficient research platform.

We, therefore, present maplab, an open, research-oriented visual-inertial mapping framework for processing and manipulating multisession maps, written in C++. On the one hand, maplab can be seen as a ready-to-use visual-inertial mapping and localization system. On the other hand, maplab provides the research community with a collection of multisession mapping tools that include map merging, visual-inertial batch optimization, and loop closure. Furthermore, it includes an online frontend that can create visual-inertial maps and also track a global drift-free pose within a localization map. In this letter, we present the system architecture, five use cases, and evaluations of the system on public datasets. The source code of maplab is freely available for the benefit of the robotics research community.

Published in:
IEEE Robotics and Automation Letters 3.3 (2018): 1418-1425, 2018
DOI: 10.1109/LRA.2018.2800113

1 Introduction

The ever growing deployment of simultaneous localization and mapping (SLAM) systems poses novel challenges for the robotics community. Availability of precise, drift-free pose estimates both outdoors and indoors has become a vital requirement of numerous robotics applications, such as navigation or manipulation. The increasing popularity of visual-inertial estimation systems created a strong incentive to improve their robustness to viewpoint and appearance changes (daylight, weather, seasons, etc.) or rapid motion. Current research efforts aim to collect data using heterogeneous agents, build maps of larger scale, cover various visual appearance conditions and maintain maps over a long time horizon. Investigating these and many related challenges requires a multi-session end-to-end mapping system that can be easily deployed on various robotic platforms and provides ready-to-use algorithms with state-of-the-art performance. At the same time it needs to offer high flexibility necessary for conducting research.

Most openly available frameworks for visual and visual-inertial SLAM either focus on a single-session case [122] or only provide large-scale batch optimization without an online frontend [199]. Usually, they are crafted for a very specific pipeline without a separation between the map structure and algorithms. They often lack completeness and will not offer a full workflow such that a map can be created, manipulated, merged with previous sessions and reused in the frontend within a single framework. This impairs the flexibility of such systems, a key for rapid development and research.

This work addresses this problem by introducing maplab[1], an open visual-inertial mapping framework, written in C++. In contrast to existing visual-inertial SLAM systems, maplab does not only provide tools to create and localize from visual-inertial maps but also provides map maintenance and processing capabilities. These capabilities are offered as a set of tools accessible in a convenient console that can easily be extended through a plugin system. These tools involve multi-session merging, sparsification, loop closing, dense reconstruction and visualization of maps. Additionally, maplab includes ROVIOLI (ROVIO with Localization Integration), a mapping and localization frontend based on ROVIO [20], a patch-based visual-inertial odometry system.

Maplab has been extensively field tested and has been deployed on a variety of robotic platforms including micro aerial vehicles [27], autonomous planes [89, 90], autonomous cars [25], autonomous underwater vehicles [214], and walking robots [59]. It has also served as a research platform for map summarization [49–52], map quality evaluation [146], multi-session 3d reconstruction [60], topological mapping [19], visual localization [77, 134, 163], and decentralized mapping [37].

To the best of our knowledge, maplab is the first visual-inertial mapping framework that integrates a wide variety of use cases within a single system. Maplab is free, open-source, and has already proved to be of great use for various research and industry projects. We strongly believe that the robotics community will harness

[1] Maplab is available at: www.github.com/ethz-asl/maplab

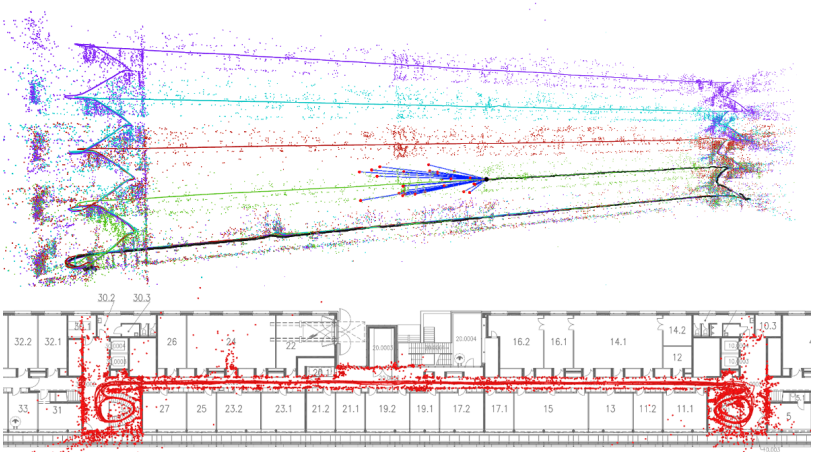


Figure 8.1: The maplab framework can build consistent visual-inertial maps from multiple mapping sessions. Here, 4 separate sessions are merged and jointly refined. The global map can then be used by odometry and localization frontend to correct for any drift when revisiting the area. The floorplan is overlayed with the landmarks of all floors demonstrating the accuracy and consistency of the map alignment.

it both as an off-the-shelf mapping and localization solution, as well as a mapping research testbed. The contributions of this work can be summarized as follows:

- it introduces a general purpose visual-inertial mapping framework using feature-based maps with multi-session support;
- it introduces ROVIOLI, a robust visual-inertial estimator tightly coupled with a localization system;
- it presents examples of algorithms and data structures for modifying and maintaining maps including map merging, sparsification, place recognition, and visualization;
- it highlights the extensibility of the system that makes it well suited for research;
- it provides evaluation of selected components of the framework.

2 Related work

There are several openly available visual and visual-inertial SLAM systems. One of the earliest examples is PTAM [109], a lightweight approach for mapping and tracking a local map in parallel. It was originally developed for augmented reality applications so it offers neither large-scale localization nor any offline processing tools. More recent examples include OKVIS [122], a visual-inertial keyframe-based estimator. This approach tracks a local map built from recently acquired keyframes, which minimizes the drift locally. Similarly, semi-dense [64] and dense [57] odometry frameworks achieve high-quality pose estimates by using photometric error formulations instead of feature-based matching. None of these methods, however, supports global localization against a previously recorded map.

ORB-SLAM [156] and ORB-SLAM2 [155] are vision-based frameworks that offer the possibility to create a map of the environment and then reuse it in a consecutive session, which closely relates to the workflow we propose here. In contrast to these systems, maplab offers an offline processing toolkit centered around a console user interface, which guarantees high flexibility and permits users to add their own extensions or modify the processing pipelines. We consider the ability to merge multiple mapping sessions into a single, consistent map and to refine it using a visual-inertial least-squares optimization a core capability of maplab that differentiates it from ORB-SLAM. Another difference worth emphasizing is the online frontend of maplab, ROVIOLI. Using image intensity within patches instead of point features guarantees a high level of robustness, even in the presence of motion blur [20].

Incorporating the capability to process multiple maps has received considerable attention in the SLAM research community with [22] being one of the earliest works incorporating multiple maps in a hybrid metric-topological approach to multi-session mapping. Use of anchor nodes to stitch together posegraphs from multiple mapping sessions is proposed in [142]. Trying to establish topological associations between maps is also proposed in [35], where maps are stored as a set of experiences. In contrast, maplab stores a unified localization map allowing to use a carefully selected subset of features, e.g. based on the current appearance conditions [25].

Systems that aim to reconstruct the 3d structure from large collections of unordered images [151, 199, 207] also contain functionalities similar to maplab. They typically offer efficient implementations of large-scale bundle adjustment optimization and advanced image and feature matching techniques. They lack, however, algorithms that process inertial data and cannot be run directly on a robotic platform in order to provide pose estimates online.

3 The maplab framework

From the user perspective, the framework consists of two major parts:

1. The online **VIO (Visual Inertial Odometry) and localization frontend**, ROVIOLI, that takes raw visual-inertial sensor data. It outputs

(global) pose estimates and can be used to build visual-inertial maps.

2. The (offline) **maplab-console** that lets the user apply various algorithms on maps in an offline batch fashion. It does also serve as a research testbed for new algorithms that operate on visual-inertial data.

The maplab framework follows an extensible and modular design. All software components are organized in packages, which are built using catkin, the official build system of ROS [175]. The C++11 standard is used throughout the framework and third-party dependencies are limited to popular and well-maintained libraries, among others Eigen [80] for linear algebra and Ceres [1] for non-linear optimization. Additionally, the framework provides ROS interfaces to conveniently input raw sensor data and output the results, such as pose estimates for an easy deployment on a robotic systems. The framework uses RViz as a 3d visualization tool to both visualize the state of the online mapping algorithms and the results of the offline processing from the maplab console.

3.1 Notation

Throughout this document and the source-code, we use the notation as defined in this section. A transformation matrix $\mathbf{T}_{AB} \in \mathrm{SE}(3)$ takes a vector ${}_B\mathbf{p} \in \mathbb{R}^3$ from the frame of reference $\mathcal{F}_B$ to the frame of reference $\mathcal{F}_A$. It can be partitioned into a rotation matrix $\mathbf{R}_{AB} \in \mathrm{SO}(3)$ and a translation vector ${}_A\mathbf{p}_{AB} \in \mathbb{R}^3$ as:

$$\begin{bmatrix} {}_A\mathbf{p} \\ 1 \end{bmatrix} = \mathbf{T}_{AB} \cdot \begin{bmatrix} {}_B\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{AB} & {}_A\mathbf{p}_{AB} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}_B\mathbf{p} \\ 1 \end{bmatrix} \tag{8.1}$$

The operator $\mathbf{T}_{AB}(\cdot)$ is defined to transform a vector in $\mathbb{R}^3$ from $\mathcal{F}_B$ to the frame of reference $\mathcal{F}_A$ as ${}_A\mathbf{p} = \mathbf{T}_{AB}\left({}_B\mathbf{p}\right)$ according to 8.1.

3.2 Workflow for multi-session mapping and localization

The typical workflow for a mapping and localization session within the maplab system is illustrated in Fig. 8.2. Often, it is beneficial to build a single localization map from multiple mapping sessions to ensure a good spatial and temporal (i.e. different appearances) coverage of the area. An initial, open loop map is built in each session using ROVIOLI in VIO mode and stored to disk. The maps can then be refined using various (offline) tools such as loop closure detection, visual-inertial optimization or co-registration of multiple sessions (map merging). Detailed inspection of the maps is possible using a large set of different visualizations, statistics and queries. More advanced modules allow, e.g., to create a dense representation (TSDF, occupancy, etc.) of the environment using data from a depth sensor or from stereo.

The resulting (multi-session) map can then be exported as a compact localization map and used by ROVIOLI (in LOC mode) for online localization during a second visit to the same place. Continuous online localization enables accurate tracking

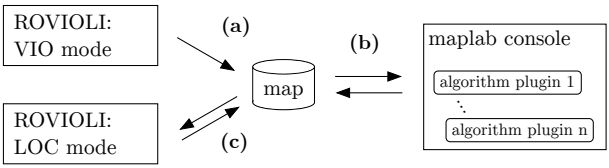


Figure 8.2: Typical workflow in maplab: (a) In VIO mode, ROVIOLI estimates the pose of an agent w.r.t. a (drifting) local frame; additionally a map is built based on these estimates. (b) Resulting maps can be loaded in the maplab-console where all of the available algorithms can be applied, e.g. map alignment and merging, VI optimization, loop closure. (c) In LOC mode, ROVIOLI can load the updated map to track a global (drift-free) pose online.

of a global pose w.r.t. a known 3d structure and thus compensates for drift in the visual-inertial state estimation.

3.3 maplab console: the offline user interface

The maplab framework uses a console user interface to manipulate maps offline. Multiple maps can be loaded into the console simultaneously, facilitating multi-session mapping experiments. All algorithms are available through console commands and can be applied to the loaded maps. Parameters specific to each algorithm are set by console flags or a flag file and can be modified at runtime. Combined with the real-time visualization of the map in RViz, this greatly facilitates algorithm prototyping and parameter tuning. It is possible to combine multiple algorithms and experiment with entire processing pipelines. Changes can be easily reverted by saving and reloading intermediate states of a map from disk.

The console uses a plugin architecture[1] and automatically detects all available plugins within the build workspace at run time. Therefore, the integration of a new algorithm or functionality is possible without any changes to the core packages. For algorithms that operate on the standard visual-inertial map datatype (see Section 3.4), no interfacing work will be necessary.

3.4 Map structure

The framework uses a data structure, called VI-map, for visual-inertial mapping data. The VI-map contains the raw measurements of all sensors and a sparse reconstruction of the covered environment. Each map may contain multiple *missions* where each is based on a single recording session. The core structure of a mission is a graph consisting of *vertices* and *edges*. A *vertex* corresponds to a state captured at a certain point in time. It contains a state estimate (pose $\mathbf{T}_{MI_k}$, IMU

[1] For more details, tutorials and documentation, please visit our wiki page: www.github.com/ethz-asl/maplab/wiki

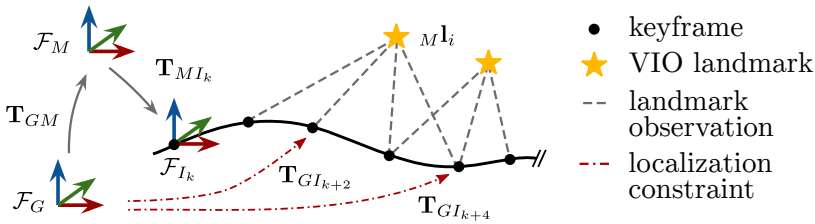


Figure 8.3: Coordinate frames used in maplab and ROVIOLI: $\mathcal{F}_G$: global, gravity-aligned map frame; all missions are anchored in this frame. $\mathcal{F}_{M_k}$: gravity-aligned frame that represents the origin of a mission k equivalent to the origin of the VIO. $\mathcal{F}_{I_k}$: IMU frame at time stamp k (body frame).

biases, velocity) and visual information from the (multi-) camera system including keypoints, descriptors (BRISK [121] or FREAK [3]), tracking information and images. An *edge* connects two neighboring vertices. While there are a few different types of edges in maplab, the most common type is the IMU edge. It contains the inertial measurements recorded between the vertices that the edge connects. Visual observations tracked by multiple vertices are triangulated as 3d landmarks. The landmark itself is stored within the vertex that first observed it. Loop closures might link observations of one mission to a landmark stored in another mission.

Fig. 8.3 illustrates the map structure and introduces the relevant coordinate frames. Each mission is anchored in the global coordinate frame $\mathcal{F}_G$ using a transformation $\mathbf{T}_{GM_i}$. The poses $\mathbf{T}_{M_iI_j}$ of mission i are expressed w.r.t. the mission frame $\mathcal{F}_{M_i}$. Therefore, it suffices to manipulate the transformation $\mathbf{T}_{GM_i}$ to anchor multiple missions in a single global coordinate system without the need for updating any vertex poses or landmark positions.

The map structure can be serialized to the Google Protobuf format, enabling portable file serialization and network transmission. Furthermore, data-intensive objects (such as images, dense reconstructions, etc.) can be attached to the maps using a resource management system. Resources are linked to either a *vertex* or a set of *missions* or simply a timestamp, and are stored on the file system separate from the main mapping data. This architecture allows for (cached) loading such (potentially large) objects on demand, effectively reducing the peak memory usage. This facilitates research in areas such as dense reconstruction and image-based/enhanced localization on large-scale maps that might otherwise exhaust the available memory on certain platforms.

3.5 Core packages of maplab

The maplab framework incorporates implementations of several state-of-the-art algorithms. All of them are conveniently accessible from the maplab console. We

only briefly highlight the ones that, in our opinion, bring a particular value to the robotics community:

VIWLS: visual-inertial weighted least-squares optimization with cost terms similar to [122]. The main batch optimization algorithm of the framework is used to refine maps e.g. after initialization with ROVIOLI or after loop closures have been established. By default, the optimization problem is constructed using visual and inertial data, but optionally it can include wheel odometry, GPS measurements or other types of pose priors.

Loop closure/localization: a complete loop closure and localization system based on binary descriptors. The search backend uses an inverted multi-index for efficient nearest neighbor retrieval on projected binary descriptors. The algorithm is a (partial) implementation of [135].

ROVIOLI: online visual-inertial mapping and localization frontend, see Section 3.6 for details.

Posegraph relaxation: posegraph optimization using edges introduced by the loop closure system. The algorithm is similar to [203]. Optionally, a Cauchy loss might be used to increase the robustness against false loop closures.

aslam_cv2: a collection of computer vision data structures and algorithms. It includes various camera and distortion models as well as algorithms for feature detection, extraction, tracking and geometric vision.

Map sparsification: algorithms to select the best landmarks for localization [49, 50] and keyframe selection to sparsify the pose graph. Useful for processing large-scale maps or for lifelong mapping.

Dense reconstruction: a collection of dense reconstruction, depth fusion and surface reconstruction [164] algorithms. Also includes an interface to CMVS/P-MVS2 [70]. See Section 4.5 for details.

3.6 ROVIOLI: online VIO and localization frontend

ROVIOLI (ROVIO with Localization Integration) is maplab's mapping and localization frontend which is used to build maps from raw visual and inertial data and also localize w.r.t. existing maps online. It is built around the visual-inertial odometry framework ROVIO [20] and extends it with localization and mapping capabilities. The following two modes of operation are available: (i) *VIO mode* in which a map is built based on the VIO estimates and (ii) *LOC mode* where additionally localization constraints are processed to track a (drift-free) global pose estimate w.r.t. a given map. The localization maps are either created directly in a previous (single-session) of ROVIOLI or are exported from the maplab-console. The preparation of a localization map within the console allows for building complex processing pipelines (e.g. multi-session maps, data selection and compression).

An overview of the (main) data flows and modules within ROVIOLI are shown in Fig. 8.4. The *Feature Tracking* module detects and tracks BRISK [121] or FREAK [3] keypoints. Feature correspondences between frames are established by matching descriptors from frame to frame. The expected matching window is predicted based on integrated gyroscope measurements to increase the efficiency

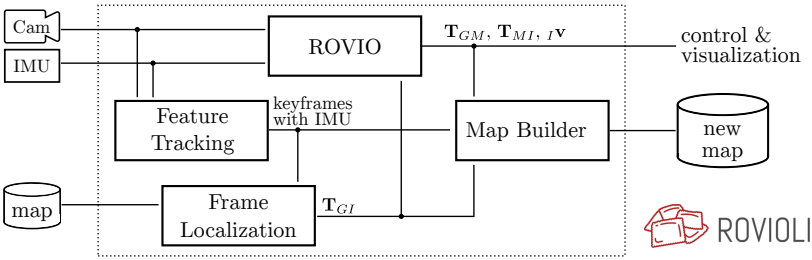


Figure 8.4: Modules and data flows within ROVIOLI (ROVIO [20] with Localization Integration).

and robustness. In *LOC mode*, keyframes containing feature points and descriptors are processed by the *Frame Localization* module to establish 2d-3d matches against the provided localization map. These 2d-3d matches are used to obtain a global pose estimate $\mathbf{T}_{GI_k}$ w.r.t. the map's frame of reference (see Fig. 8.3) using a P3P algorithm within a RANSAC scheme. The raw global pose estimates are fed to ROVIO where they are fused with the odometry constraints to estimate a transformation $\mathbf{T}_{GM}$ in addition to the local odometry pose $\mathbf{T}_{MI}$. The outputs of all modules are synchronized within the *Map Builder* to construct a visual-inertial map (VI-map). The resulting map can serve as a localization map in subsequent sessions or can be loaded into the maplab console for further processing.

A process-internal publisher-subscriber data exchange layer manages the data flows between all modules within ROVIOLI. This architecture makes it easy to extend the current online pipeline with new algorithms, e.g. for online multiagent mapping, semantic SLAM, or localization research.

4 Use-cases

This section gives an overview of five common use cases of maplab: online mapping and localization, multi-session mapping, map maintenance, large-scale mapping and dense reconstruction. While maplab offers much more than that, we believe these examples highlight the capabilities of the system, the expected performance and its scalability.

Furthermore, we provide the related console commands to reproduce every example. The intention is to show that the following results can be obtained by relying solely on the user interface, without any additional code development. For more documentation, updated commands, datasets and tutorials, please visit our wiki page: `www.github.com/ethz-asl/maplab/wiki`.

Table 8.1: Global position and orientation RMSEs on EuRoC datasets [28] for ROVIO (only VIO), ROVIOLI using one of the datasets as a localization map and ROVIO+VIWLS that corresponds to a full batch visual-inertial least-squares optimization (VIWLS). Additionally, the results of ORB-SLAM2 [155] (in batch and real-time) are compared. ROVIO and ROVIOLI use a single camera and IMU data whereas ORB-SLAM2 uses a stereo camera. The localization map for ORB-SLAM2 has been built in SLAM mode whereas the localization evaluation has been performed in localization mode. For V2-medium, we were unable to build a map with ORB-SLAM2's real-time mode as the estimator diverged (marked with X).

	MH1 *LOC: MH2		V2-easy *LOC: V2-medium	
	position	orientation	position	orientation
ROVIO	0.178 m	1.49 deg	0.064 m	0.90 deg
ROVIOLI*	0.082 m	1.43 deg	0.057 m	1.57 deg
ROVIO+ VIWLS	0.036 m	1.29 deg	0.027 m	1.06 deg
ORB-SLAM2* (batch mode)	0.084 m	0.78 deg	0.121 m	1.14 deg
ORB-SLAM2* (real-time)	0.464 m	13.34 deg	X	X

4.1 Online mapping and localization with ROVIOLI

For many robotic applications it is of high importance to have access to (drift-free) global pose estimates. Such capability enables, e.g., teach and repeat scenarios, robotic manipulation and precise navigation. Within maplab, as a first step, we use ROVIOLI to create an initial VI-map of the desired area of operation. The sensor data can be provided either offline in a Rosbag or online using ROS topics. Upon completion, the VI-map is automatically loop closed, optimized and optionally keyframed and summarized to obtain a compact localization map. In a second session the localization map can be passed to ROVIOLI to obtain drift-free global pose estimates in the mapped area.

We evaluated the ROVIOLI estimates against plain ROVIO [20] results and the estimates from a full-batch optimization on the EuRoC datasets [28]. To that end, in a first step, we created a localization map using one of the datasets. Then in a second step we processed a second EuRoC dataset using both ROVIOLI (using the previously built map) and ROVIO. The results are presented in Fig. 8.5 and Table 8.1, where we compare the groundtruth error of ROVIO, ROVIOLI, and the full-batch optimized trajectory. These experiments demonstrate the drift-free performance of the system and the improvements upon the regular VIO estimation. Additionally, Table 8.2 shows timing information of ROVIO and ROVIOLI compared to ORB-SLAM2 [155].

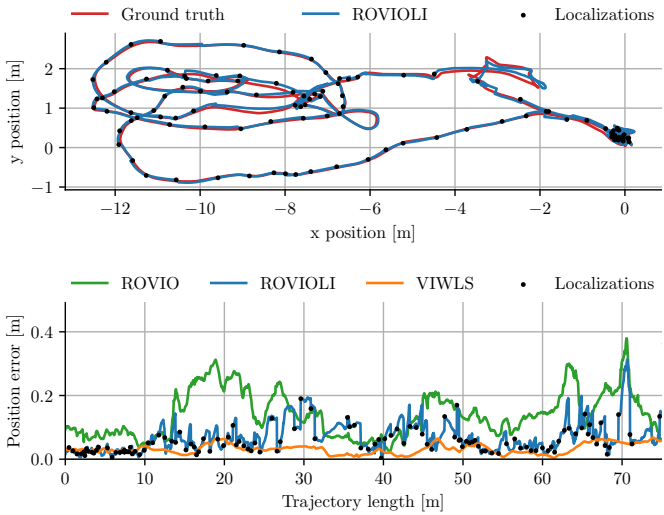


Figure 8.5: Evaluation of ROVIOLI on the EuRoC machine hall dataset [28]. *Top*: Ground-truth positions overlayed with the ROVIOLI position estimates. *Bottom*: Position error of the visual-inertial odometry pipeline ROVIO [20], ROVIOLI and the optimized VI-map (VIWLS) compared to the ground truth.

Table 8.2: *(a)* Timing and CPU load for ROVIO, ROVIOLI and ORB-SLAM2 on EuRoC MH1 dataset processed at 20 Hz. In case of ROVIOLI and ORB-SLAM2 (marked with *), the estimator was set to localize against a map built from EuRoC MH2. All reported values have been measured on an Intel Xeon E3-1505M@2.8Ghz. A CPU load of 800% corresponds to fully utilizing **all** 8 (logical) cores of the CPU. *(b)* Single frame processing times for the individual blocks of ROVIOLI. The total time does not correspond to the sum of the individual blocks as they run in parallel. Instead, it is the time it takes for a single frame to be fully processed.

(a)

	Frame update	CPU load
ROVIO	23 ms	56%±7.7%
ROVIOLI*	44 ms	105%±14.8%
ORB-SLAM2* (batch mode)	63 ms	162%±10.9%

(b)

ROVIOLI frame update	
ROVIO update	22.7 ms
Feature tracking	20.6 ms
Localization	20.4 ms
Map building	3.2 ms
Total	**44.2 ms**

4.2 Multi-session mapping

In many mapping applications, it is not possible to cover the entire environment within a single mapping session. Apart from that, it might be desirable to capture the environment in as many differing visual appearance conditions as possible [25]. Therefore, maplab offers tools to co-register maps from multiple sessions together and jointly refine them to obtain a single, consistent map.

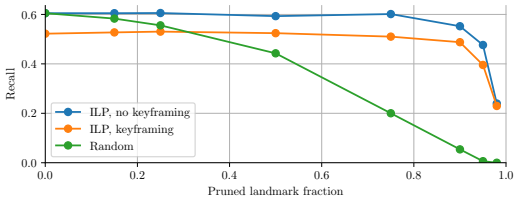
Hence this use case demonstrates the process of creating a map of a university building from four individual trajectories. Each trajectory passes through the ground floor, staircases and one other floor of a building. Combined, they cover over 1,000 meters and contain about 463,000 landmarks. On such large maps, many of the common operations such as optimization or loop closure quickly become intractable without a careful selection of the data. For this reason, we employ a keyframing scheme using heuristics based on vertex distance, orientation, and landmark covisibility. The loop closure algorithm of maplab correctly identifies the geometric transformations between all missions and the non-linear optimization refines the geometry. The result is a compact, geometrically-consistent localization map of 8.2 MB ready to be used by ROVIOLI for localization within the entire building as shown in Fig. 8.1.

This use case can be reproduced using the following commands in the maplab console:

```
# Load multiple single session maps from ROVIOLI.
load_merge_all_maps --maps_folder YOUR_MAPS_FOLDER
# Keyframing and initial optimization.
kfh
optvi
# Set one mission as base, anchor the others.
set_mission_baseframe_to_known
anchor_all_missions
# Pose-graph relaxation, loop-closure, optimization.
relax
lc
optvi
```

4.3 Map maintenance

Large feature-based models, potentially built in multiple sessions, easily comprise thousands of landmarks and reach considerable storage size. However, it is not really necessary to keep all of the landmarks to guarantee good localization quality with ROVIOLI. Maplab offers a map summarization functionality based on [49] that uses an integer-based optimization to perform the landmark selection. The algorithm attempts to remove the least commonly seen landmarks but at the same time maintain a balanced coverage of the environment. Maplab also includes a keyframing algorithm to remove redundant vertices and only keep the ones necessary for an efficient and accurate state estimation. By removing the vertices we also eliminate many vertex-landmark associations that contain descriptors of considerable size. Both summarization and keyframing permit to significantly reduce the model size without a large loss in pose estimation quality.



pruned fraction	0	0.5	0.75	0.9	0.95	0.98
# landmarks	18,316 (16,088)	8,824 (9,148)	4,259 (4,570)	1,818 (1,822)	899 (906)	349 (358)
map size [MB]	34.559 (3.740)	29.217 (3.209)	24.028 (2.602)	17.619 (1.837)	12.203 (1.214)	6.707 (0.712)

Figure 8.6: The localization performance and map size after ILP landmark summarization and keyframing+summarization (in brackets). Keyframing removes vertices including vertex-landmark associations, effectively making the map smaller. The original map had $6,258$ vertices whereas the keyframed map contains 760. Keyframing consistently reduces the recall by a few percent while summarization only affects the quality when the pruned landmark fraction exceeds 85%. For comparison, we provide a recall curve for a random selection of landmarks to be removed.

The map maintenance is demonstrated on a database map built from four mapping sessions recorded on the ground floor of the building introduced in Section 4.2. Each mapping session covers about 90 meters and contains about 20,000 landmarks, out of which about 5,000 are considered reliable. A fifth dataset is used as a query – we try to localize each vertex against the database, built from the four datasets, and verify if the position error is smaller than 50cm. We compare the recall of localization maps that were pre-processed in different ways, either summarized, keyframed or both.

Fig. 8.6 presents the influence of landmark summarization and keyframing on the localization map size and demonstrates how those approaches affect the localization. The results confirm that keyframing significantly reduces the localization map size with a rather marginal loss of localization quality. Similarly, summarization can reduce the total amount of landmarks by 90% without grave consequences. When these methods are combined we can reduce the map size 13 times and keep the recall level at 51%, compared to 60% for the full map.

```
# Keyframe the map and sparsify landmarks to 10,000.
kfh
landmark_sparsify --num_landmarks_to_keep=10000
```

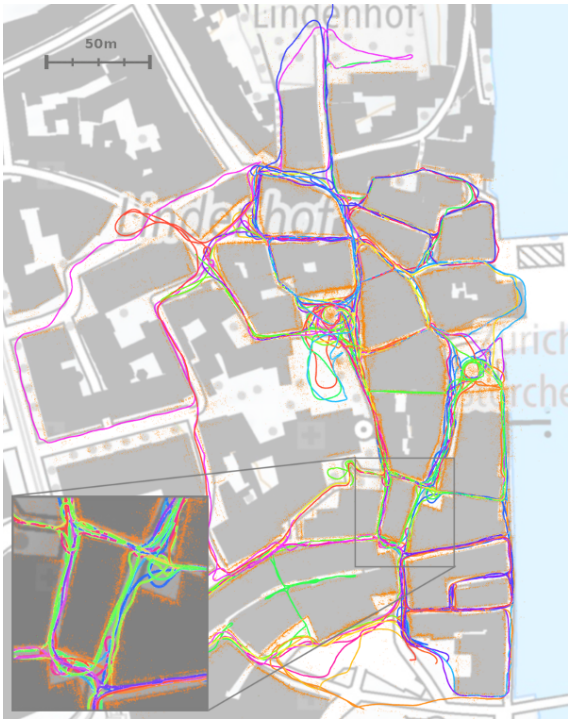


Figure 8.7: Large-scale, multi-session VI-map of Zurich's old town. Built from the raw visual-inertial data recorded in 45 sessions using Google Tango tablets on two different days (sunny and cloudy). The total duration of the recordings is 231 min. The final map contains trajectories with a total length of 16.48 km, 435k landmarks with 7.3M observations and has a size of 480 MB. The map is available on the maplab wiki page for download.

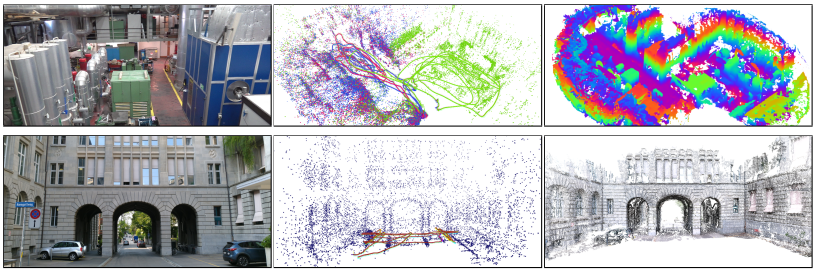


Figure 8.8: Two different dense reconstruction tools are available in maplab. *Top:* stereo dense reconstruction is used to compute depth maps based on grayscale images and optimized camera poses. They are then fused in voxblox [164] to create a surface mesh. 3 EuRoC datasets [27] (MH1-3) are combined to create an aligned and optimized VI-map. *Bottom:* CMVS/PMVS2 [70] reconstruction results based on a single recording session using a multi-camera system with a RGB camera.

4.4 Large-scale mapping

In this use case we would like to demonstrate the large-scale mapping capabilities of maplab and the applicability to a sensor other than the VI-sensor [159]. To that end we used the publicly available Google Tango tablets, and recorded a large-scale, multi-session map of the old town of Zurich. We exported the raw visual-inertial data and processed it with ROVIOLI to obtain the initial open loop maps. We then loaded these maps into the maplab console for alignment and optimization and applied the same tools as described in Section 4.2. The bundle adjustment and pose-graph relaxation was performed on a desktop computer with 32 GB RAM overnight. An orthographic projection of the optimized VI-map onto the map of Zurich, as well as further details about the map can be found in Fig. 8.7. The figure shows that the resulting map is consistent with the building and streets across most of the map with some minor inconsistencies in areas of low coverage.

4.5 Dense reconstruction

Many applications in robotics, such as path planning, inspection and object detection require a more dense 3d representation of the environment. Maplab offers several dense reconstruction tools, which use the optimized *vertex* poses of the sparse map to compute dense depth information based on camera images attached to the VI-map.

Stereo dense reconstruction

In order to compute depth maps from multi-camera systems, this tool first identifies stereo cameras that are suitable for planar rectification. It then utilizes a (semi-global) block matcher to compute depth maps for every stereo pair along the trajectory. The resulting depth maps (or point clouds) are attached to the VI-map and stored in the resource system. The following commands assume that the maps are already aligned, loop closed and optimized as described in Section 4.2.

```
stereo_dense_reconstruction
```

TSDF-based depth fusion

Once the VI-map contains depth information, e.g. obtained using the above described commands or an RGB-D sensor, the globally consistent camera poses of the VI-map can be utilized to create an equally consistent global 3d reconstruction. To that end, maplab employs voxblox [164], a volumetric mapping library, for TSDF-based depth fusion and surface reconstruction. The following commands will insert depth maps or point cloud data into a voxblox grid and store a surface mesh to the file system. The top row of Fig. 8.8 shows the reconstruction results of three combined EuRoC machine hall datasets [28].

```
create_tsdf_from_depth_resource
    --dense_tsdf_voxel_size_m 0.10
    --dense_tsdf_truncation_distance_m 0.30
export_tsdf
    --dense_result_mesh_output_file YOUR_FILE
```

Export to CMVS/PMVS2

For more accurate dense reconstructions maplab offers an export command to convert the sparse VI-map and images to the input data format for the open-source multi-view-stereo pipeline, CMVS/PMVS2 [70]. Even though the export of grayscale images is supported, the best results are obtained using RGB images. The VI-map and the resulting 3d reconstruction can be seen in the bottom row of Fig. 8.8.

```
export_for_pmvs
    --pmvs_reconstruction_folder EXPORT_FOLDER
```

5 Using maplab for research

All the algorithms and console commands required for the use cases in Section 4 are available in maplab and constitute most of the basic tools needed in visual-inertial mapping and localization. Furthermore, a rich set of helper functions, queries, and manipulation tools are provided to ease rapid prototyping of new

algorithms. The plugin architecture of the console allows for an easy integration of new algorithms into the system. Examples demonstrating how to extend the framework are provided in the project's wiki pages. We would like to invite the community to take advantage of this research-friendly design.

6 Conclusions

This work presents maplab, an open framework for visual-inertial mapping and localization with the goal of making research in this field more efficient by providing a collection of basic algorithms and letting researchers focus on actual tasks. All components in maplab are written in a flexible and extensible way such that novel algorithms that rely on visual-inertial state estimates or localization can be integrated and tested easily. For this reason, the framework provides an implementation of the most important tools required in mapping and localization related research such as visual-inertial optimization, a loop closure/localization backend, multi-session map merging, pose-graph relaxation and extensive introspection and visualization tools. All these algorithms are made accessible from a console-based user interface where they can be applied to single or multi-session maps. Such a workflow has proven to be very efficient when prototyping new algorithms or tuning parameters.

Secondly, the framework contains an online visual-inertial mapping and localization front-end, named ROVIOLI. It can build new maps from raw visual and inertial sensor data and additionally track a global (drift-free) pose in real-time if a localization map is provided. Previous work made use of this capability on different robotic platforms and demonstrated its ability of accurately tracking a global pose for a multitude of applications, including navigation and trajectory following.

7 Acknowledgement

We would like to acknowledge the many other contributors of maplab, most importantly: Titus Cieslewski, Mathias Bürki, Timo Hinzmann, Mathias Gehrig and Nicolas Degen. Furthermore we would also like to thank Michael Blösch the author of ROVIO. The research leading to these results has received funding from Google Tango and the EU H2020 project UP-Drive under grant no. 688652.

Part B

ROBUST PLACE RECOGNITION

Paper VI

Efficient Descriptor Learning for Large Scale Localization

Antonio Loquercio, Marcin Dymczyk, Bernhard Zeisl, Simon Lynen, Igor Gilitschenski, Roland Siegwart

Abstract

Many robotics and Augmented Reality (AR) systems that use sparse keypoint-based visual maps operate in large and highly repetitive environments, where pose tracking and localization are challenging tasks. Additionally, these systems usually face further challenges, such as limited computational power, or insufficient memory for storing large maps of the entire environment. Thus, developing compact map representations and improving retrieval is of considerable interest for enabling large-scale visual place recognition and loop-closure.

In this paper, we propose a novel approach to compress descriptors while increasing their discriminability and matchability, based on recent advances in neural networks. At the same time, we target resource-constrained robotics applications in our design choices. The main contributions of this work are twofold. First, we propose a linear projection from descriptor space to a lower-dimensional Euclidean space, based on a novel supervised learning strategy employing a triplet loss. Second, we show the importance of including contextual appearance information to the visual feature in order to improve matching under strong viewpoint, illumination and scene changes. Through detailed experiments on three challenging datasets, we demonstrate significant gains in performance over state-of-the-art methods.

Published in:
Robotics and Automation (ICRA), 2017 IEEE International Conference on, 2017
DOI: 10.1109/ICRA.2017.7989359

1 Introduction

Image-based localization aims to accurately estimate and track the pose of a mobile platform with respect to a global map. It is therefore a fundamental task for many robotics and Augmented Reality (AR) applications. The former need localization for path-planning, obstacle avoidance, or manipulation while the latter require high quality pose estimates to correctly project virtual objects into the camera view.

Traditionally, large-scale image-based localization has been treated as an image retrieval problem [42]. However, other approaches, such as [185], achieve higher

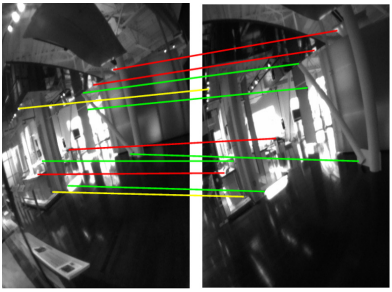


Figure 9.1: Retrieved matches. The descriptor projection and augmentation methodology herein proposed enables difficult descriptor matches under strong appearance changes. Then, some matches (green) may still be possible without making use of our projection technique, relying on search directly in binary descriptor space. However, the proposed linear projection enables additional, more challenging matches (yellow), that can be further improved by including broader context information (red).

accuracy using the full 3D model of the workspace to estimate the 6-DoF pose of the query camera employed by the mobile platform. The 3D model is often pre-built with a large set of database images using Structure-from-Motion (SfM) or Simultaneous Localization and Mapping (SLAM) techniques. Recent advances in research now make it possible to construct models on a huge scale consisting of millions of points in only a few hours [65], creating the need for methods that can handle such large datasets.

The key step for image-based localization using 3D models consists of finding correspondences between 2D local features in the query image and 3D points in the model. Commonly, the same type of local image descriptors, *e.g* [3, 132], used to build the 3D model are extracted from the query image. This allows to formulate the correspondence search as an instance of descriptor matching: We select the 3D landmark corresponding to a 2D feature by searching for the nearest neighbor of that feature's descriptor in the space containing 3D landmarks descriptors.

However, as the size of the environment increases, the large-scale search required to establish correspondences is particularly challenging. On the one hand, due to

the high dimensionality of image descriptors, not necessarily lying in an Euclidean space, search is computationally expensive [215]. On the other hand, strong illumination, viewpoint, or scene changes, as well as perceptual aliasing [42], can quickly debase descriptors repeatability and discriminability.

To make retrieval both tractable and efficient, current approaches rely on classical dimensionality reduction techniques [102], or on more elaborate optimization methods aimed to generate discriminative and compact descriptors [83, 171]. In designing such projection systems, two competing constraints need to be satisfied: Capturing the complexity involved in the search problem and introducing as low latency as possible, to enable real time implementations even on devices with limited computational capabilities. Trading off between these two requirements, we propose in this paper two ways to successfully map descriptors in a reduced dimensional Euclidean space while maximizing their matchability and discriminability: a fast linear projection and a more accurate, but slower, context-augmented mapping. The resulting descriptors, learning an invariance to strong changes in appearance, enable retrieval of hard matches as the ones shown in Fig. 9.1. Overall, this work makes the following contributions:

- We use a novel supervised learning strategy to train an efficient linear projection of descriptors.
- We show that including context information in the projection function consistently increases the retrieval performance, outperforming even completely learned descriptors.
- We demonstrate the effectiveness of our methods via an extensive experimental evaluation, and benchmark the performance gains against baseline methods.

2 Related Work

Great improvements in designing image-based localization systems have been achieved in the last few years [134, 135, 147, 185, 232]. All of them rely on some form of direct or indirect matching between database and query descriptors. In this paper, we concentrate on ways to reduce descriptors dimensionality in order to lower the memory requirements, speed up the matching process, and increase its efficiency.

Traditionally, classical approaches from statistics are used to encode descriptors while maximizing their information content. The most commonly used linear technique, principal component analysis [102], performs a mapping to the hyperplane maximizing data variance. Other techniques try to generalize this idea to non-linear mappings by either using the kernel trick [188] or with graph based methods [16, 180, 210]. However, such *unsupervised* projections are not tuned on the final matching task.

Instead, all relevant *supervised* learning based approaches fall into one or both of the following two categories: (i) learn a metric suited to the classification problem [225] or (ii) learn a projection function of the input data [83, 134, 171, 197]. Methods from both categories aim to reduce the distance between corresponding descriptor pairs and to increase the distance between non-corresponding ones. In this work, we set the projected space to be Euclidean, so that the resulting descriptors, avoiding all the complexities related to a learned metric, can be easily used within existing algorithms.

The spectrum of possibilities to learn a projection function ranges from employing statistical tests [134], convex-optimization [197], or minimizing a margin based loss [83, 171]. However, learning a mapping to a space where nearest neighbor search is efficient is not a trivial task. In fact, the notion of a nearest neighbor depends on the interaction between all other points. Therefore great care has to be taken in order to present the optimization with the right training data for learning these relationships. For this reason, *supervised* methods usually require large amount of training data to generalize. Moreover Philbin *et al.* proposed in [171] to directly include in the optimized loss function *hard* pairs $\{(d_i, d_j)\}$, where d_i and d_j lie very close in input space despite representing non matching 3D points. While this choice proves to boost the optimization performance, it still misses that hard cases in input space might not be hard cases in projected space, and vice-versa. This is particularly true when the learned projection function is not linear. To face this issue, we present a novel learning policy that, concentrating on the current hard cases in projected space, allows to drastically reduce the amount of training data required to generalize.

Nonetheless, low-level features as corners, edges, or blobs are known to have limited discriminability in large environments with high repetitive structures, where many detected features look alike. State-of-the-art approaches try to tackle this issue by learning more powerful features [11, 195, 197, 229, 230]. Many of these works using convolutional neural networks (CNNs) have reported promising results, yielding an edge over hand-crafted solutions as SIFT [132] or FREAK [3].

In this paper, we will generalize the techniques mentioned above to encode any given type of image patch descriptor in a compact Euclidean space while increasing discriminability. With our results, establishing the 2D-3D matches required by image-based localization algorithms is both reliable and efficient, even in challenging environments.

3 Methodology

Our aim is to learn a projection function f, parametrized by θ, mapping an image patch descriptor $d \in X$ to a lower dimensional Euclidean space $T \in \mathbb{R}^d$:

$$f_\theta \colon X \mapsto T \tag{9.1}$$

The small dimensionality of the projected descriptors $f_\theta(d)$ will both reduce the memory footprint and increase the efficiency of k-Nearest Neighbors (k-NN) search

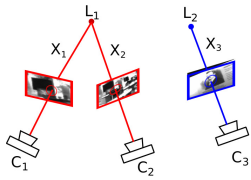


Figure 9.2: A descriptor triplet (X_1, X_2, X_3). The first two components of the triplet, X_1 and X_2, are matching, meaning they observe the same feature. X_3, on the other hand, is not matching, because it describes a different interest point.

enabling the use of fast algorithms such as KD-Tree [17]. Moreover, since the projected space is Euclidean, we can use the L_2 distance to compare descriptors and determine their similarity. In general, we would like f_θ to have as small overhead as possible, to enable real time implementations on devices with limited computational resources.

Building on the success of state-of-the-art supervised learning systems [83, 134, 171, 197], we model projection as an optimization problem. This is typically done by optimizing a loss L on the projection function f_θ. The main objective of the optimization is to decrease the distance in projected space between matching descriptors while increasing the distance between non-matching ones.

3.1 Linear descriptor projection

The case where f_θ is a real valued matrix W representing a low latency linear mapping is of crucial importance for all devices with limited computational resources. To learn the components of W, we optimize a triplet loss function L, proved in [190] to optimally model relative interactions between input samples. In addition, a novel learning policy is used in order to increase performance while decreasing training time.

Triplet loss. A triplet is defined as a collection of 3 descriptors (X_1, X_2, X_3) where X_1 and X_2 are matching, while X_3 is not, as depicted in Fig. 9.2. We define $d(.)$ as the L_2 distance between two mapped samples, that is:

$$d(X_i, X_j) = \|f_\theta(X_i) - f_\theta(X_j)\|_2 \tag{9.2}$$

Accordingly we optimize a non-convex problem with the cost function defined as:

$$L(f_\theta) = \sum_{triplets} \max(d(X_1, X_2) + margin - d(X_1, X_3), 0)^2 \tag{9.3}$$

The idea behind this cost function is that in projected space non matching samples are expected to be more distant than a margin while matching samples should be as close as possible.

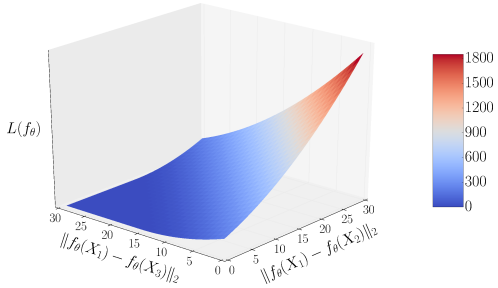


Figure 9.3: Margin based loss. Equation 9.3 is plotted in this illustration as function of the L_2 distance between matching and non matching samples. Even if convex in the latter parameters, $L(f_\theta)$ is not convex in the projection function parameters θ.

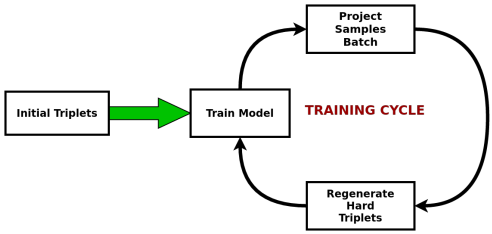


Figure 9.4: Training cycle of the proposed learning method. To initialize the training, some initial samples triplets should be provided. After these have been back-propagated, we project a batch of training samples, and use them to produce new *hard* triplets. These will be used to continue training the model.

Training cycle. Multiple works observed that a hard training policy is beneficial for the learning system [171, 195, 230]. To efficiently capture hard cases without pre-computing a large set of training data, we propose to produce difficult examples on the fly. In fact, we initialize the training with only a limited amount of input triplets and then, during learning, re-generate *hard* triplets, in which the non matching sample X_3 is one of the nearest neighbors of X_1 in projected space.

Our cyclic learning process, depicted in Fig. 9.4, drastically reduces training times while increasing optimization performance, since it concentrates the attention of the optimization on cases in which the projection performs worst.

Since we try to maximally separate non matching points, this is equivalent to learning a Mahalanobis matrix $M = W^T W$. However, by learning W we avoid all the complications related to adding the positive-semi definiteness constraints on M. The same learning problem was already tackled in [171, 197]. Even though our approach is conceptually similar to theirs, use of triplet loss and our training

policy allow us to increase performance, while controlling (in contrast [197]) the output dimensionality. In the experimental section we show that the resulting linear projection outperforms non-linear methods and successfully generalizes to unseen data.

3.2 Context augmented descriptors

A known problem in large scale localization is the degradation of performance that inevitably happens in large workspaces with highly repetitive structures, such as warehouses, offices, and museums. In these conditions low level features are not discriminative enough, and despite the improvement that can be obtained by learning an informative projection f_θ [171], the problem remains essentially unsolved.

However, this can be mitigated by including visual context from the feature neighborhood which additionally improves discriminability. Formally, this is equivalent to changing the projection function to $f_\theta(d_i, I_i)$, taking as input both the original descriptor d_i and its surrounding raw image context I_i. The intuition here is based on the fact that similar non-matching descriptors can be better distinguished if we include additional information coming from their neighborhood.

To achieve this, we need to first properly define context, then find a way to represent it and eventually propose a method to learn a combination between the feature and the context descriptors.

What is context? We define context for a visual feature by a square area around it in the image plane. Examples of context can be seen in Fig. 9.5. The neighborhood N is tied to the feature detected scale s by the formula:

$$N = C \times s \tag{9.4}$$

Where C is a learned constant, selected by cross-validation. This way, the context region automatically inherits the scale invariance of the feature detector.

Essentially, context is an higher resolution version of the image patch the original descriptor was extracted from. Therefore, the obvious question is why to use the original descriptor at all. Later in the experimental section, we show that a combination of context and the original feature outperforms an approach that learns directly on the context window, and provide a possible reasoning.

How to represent context? We want to find a function $g_\rho(I_i)$ that takes as input an image patch I_i and produces a discriminative descriptor of it. Traditionally, this has been solved by aggregating the image features in a single vector such as bag-of-visual words [170], VLAD [97], or Fisher vectors [183]. However, in the last few years convolutional neural networks (CNNs) emerged as a powerful image representation tool [8, 11, 87, 193, 195, 230].
We represent $g_\rho(.)$ through the simple architecture presented in [195], having three convolutional layers followed by hyperbolic tangent units, L_2 pooling [192], and local subtractive normalization. This choice was motivated by the efficient performance, and the small network size, that allows for real time evaluation. We

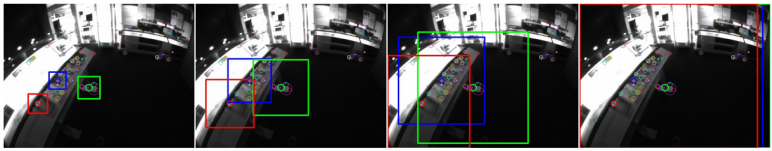


Figure 9.5: Example of local context. To achieve scale invariance, we couple the context with the feature scale through a constant multiplier. In the figure, the multipliers are respectively $20, 50, 100, 300$. Note that from multiplier 100 boundary effects starts being significant, while for 20 the context captures only few others interesting points. As we expect from a qualitative analysis, the multiplier 50 gives the best result in term of performance.

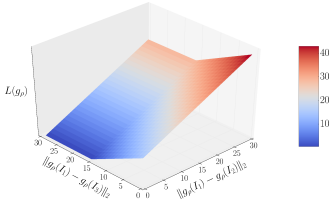


Figure 9.6: Context descriptor loss. Equation 9.5 is plotted in this illustration with respect to the L_2 distance between matching and non matching image patches. Differently from the loss in Fig. 9.3, we put here more emphasis on the distance between matching image patches, required to be as small as possible. Enforcing this concept, even if beneficial for learning an image patch feature, resulted in performance drops in training a linear projection.

train the context descriptor minimizing a triplet loss on (I_1, I_2, I_3), where the pair (I_1, I_2) is matching while (I_1, I_3) is not, defined as:

$$L(g) = \sum_{triplets} d(I_1, I_2) + \max(margin - d(I_1, I_3), 0) \tag{9.5}$$

where $d(.)$ represents L_2 distance between context descriptors:

$$d(I_i, I_j) = \|g_\rho(I_i) - g_\rho(I_j)\|_2 \tag{9.6}$$

This loss, slightly modified w.r.t. Equation 9.3, mimics the one used in [195], and is depicted in Fig. 9.6. Using descriptor to image correspondences, we can convert the training data we used before to learn a linear projection to image patch triplets, automatically generating training data for this task. Moreover, to again speed up the training process and increase its performance, we employ our hard triplet regeneration scheme described in Fig. 9.4.

Dataset	Notes	Descriptors	3D Landmarks	Keyframes
Museum	Train	2.7M	420K	11.4K
	Validation	30K	2.8K	0.4K
	Test	0.13M	19K	0.6K
Office1	Test only	0.23M	37K	3K
Office2	Test only	0.11M	15K	2.1K

Table 9.1: Train and evaluation datasets. Only the Museum dataset is used for training our models. To test generalization on visually different data, unseen during training, we evaluate on two other challenging indoor datasets.

How to combine feature and context descriptor? To reduce the computational overhead, we adopted a simple approach: We first stack the original descriptor and the context feature, then we learn a linear projection W_c for the combined feature as in 3.1. In the mathematical formalism we used before, this is equivalent to the final projection function:

$$f_\theta(d_i, I_i) = W_c \begin{bmatrix} d_i & g_\rho(I_i) \end{bmatrix} \tag{9.7}$$

Despite its simplicity, this method proved to learn efficiently a complex synergy between its two components.

Even if a joint training of W_c and $g_\rho(.)$ is possible, we experimentally noticed a slight improvement in performance by learning these two parts separately. The most likely reason for this is that, with separate learning, we can train each component on a more suited loss and present the respective optimization procedures, with our *hard* triplets regeneration policy, more suitable difficult cases.

Full frame context augmentation. To evaluate whether local context performs better than full image context, we used as image descriptor either VLAD [97] or the last hidden layer of some CNNs architecture pre-trained on the object recognition task [87, 116, 196, 208]. Then we learned a linear combination of the full image descriptor and feature descriptor as presented in the previous subsection. We did not use the model in [195] to generate full image descriptors, given its small input resolution (64×64) and reduced complexity. This intuition was in fact confirmed in our experiments with multiple context sizes (Fig. 9.5), where image-size neighborhoods resulted in considerable drops in performance.

4 Experimental Setup

To generate training data for our algorithms, we use the 3D model of a large museum built with SfM. Given the size and repetitiveness of this workspace, we aim to learn and later test invariance to strong appearance changes and perceptual aliasing. Since SfM reconstructs each 3D scene point from multiple 2D observations,

Test	FREAK (512b)	PCA (16d)	KPCA (16d)	Learnt linear (16d)	VLAD +FREAK (16d)	ResNet +FREAK (16d)	Ours w/context (16d)
Museum	0.488	0.495	0.512	0.531	0.573	0.612	**0.684**
Office1	0.178	0.188	0.210	0.237	0.310	0.379	**0.433**
Office2	0.228	0.244	0.277	0.301	0.348	0.394	**0.505**

Table 9.2: PR Area Under the Curve (AUC). Generalization results over 3 evaluation datasets. All projection methods are learned with the train split of the Museum dataset. The corresponding PR curves are depicted in Fig. 9.7

.

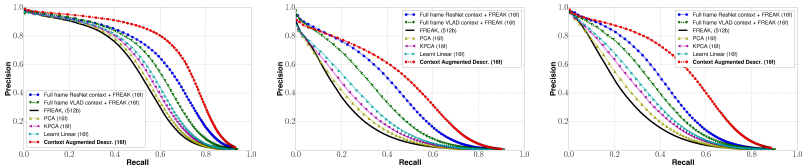


Figure 9.7: PR curves. Despite being learned on the train split of the Museum dataset, our methods generalize very well to the visually different data of the Office datasets. Indeed, it can be observed that both our linear and context augmented projection consistently outperform all baseline approaches. Interestingly, search with FREAK descriptors performs very poorly. We believe this is due to the strong changes in appearance present in our datasets. The corresponding PR AUC are presented in Table 9.2.

each 3D landmark can be associated with a set of descriptors. These associations can be used to produce matching and non matching descriptor samples.

Our framework employs 512-bit FREAK [3] descriptors. The train, validation, and test datasets are obtained by selecting different and not intersecting regions of the environment. As typical in any machine learning pipeline, the train and validation splits are used at learning time, the former to actually perform the optimization and the latter to check for over-fitting and pick the best model. The test set is solely used for evaluation. To further prove generalization on visually different data, we test our approaches on two other indoor 3D models recorded in a large office building. Table 9.1 presents more details about these datasets.

4.1 Optimization

We train our methods using the Google Tensorflow framework, and Adagrad for optimizing our loss. In all our training algorithms, we set the batch size to be 1K, and employ an exponential learning rate decay with coefficient 0.8 for each 10 processed batches. Our learning policy allows us to get good results already after as few as 5 epochs. The training times are around 30 min for linear projection and approximately 2 hrs for the CNN context projector.

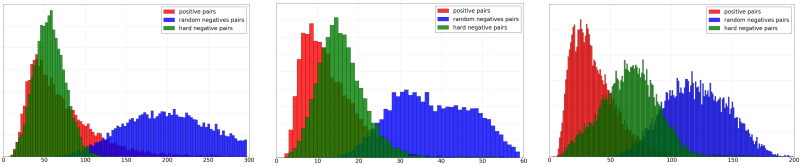


Figure 9.8: Pairwise distances distribution. In large and repetitive environments it is difficult to discriminate between matching (in red) and non-matching pairs (in green and blue). In fact, for each descriptor there are going to be many non-corresponding nearest neighbors (distances in green). Our linear and context projection alleviates this issue, improving descriptors discriminability.

4.2 Evaluation methodology

Since our data consists of descriptors associated to 3D landmarks, we can quantitatively evaluate our results by the Precision-Recall (PR) curve, which can be summarized by its Area Under the Curve (AUC). More specifically, we can compute for all descriptors in a test dataset the number of true positives, false positives and false negatives retrieved by search with a threshold on descriptor distance. Eventually, a set of thresholds can be used to compute a final PR curve.

We believe this setup can better evaluate the impact of our solutions compared to evaluating the performance of the full localization task, where higher-level algorithms can influence our results. In fact, we are primarily interested in testing the quality of our projection function, whose applications are not limited to localization.

5 Results

We benchmark our models only against traditional baseline methods such as PCA [102] or KPCA [188], because their performance was superior in our datasets than any other learned mapping [83, 134, 171]. Fig. 9.9 shows the results of the linear and a context augmented projection at different dimensions. As expected, higher dimensionality entails a gain in performance, at the cost of increased memory footprint and computational complexity. We believe 16 dimensions represented by singly-precision floating point numbers to offer optimal trade-off between performance and complexity, because it unlocks high quality and fast k-NN search, not raising the memory requirements w.r.t. 512-bit FREAK.

From Table 9.2 we can evince that our proposed linear projection not only outperforms any other considered context-free method, but also generalizes very well to visually different data of the Office datasets. In fact, in those workspaces it increases the PR AUC score of more than 30% w.r.t. FREAK descriptors and 13% w.r.t. our strongest baseline, *KPCA*.

Context augmented descriptors, described in sec. 3.2, provided impressive results: On the challenging Office1 datasets, their use resulted in a PR AUC score of 0.433, more than twice as high compared to using *KPCA* (0.210) or unprojected binary descriptors (0.178).

Results in Fig. 9.7 and Table 9.2 show that local context is more informative than full frame context, represented either by CNNs or VLAD. This can be justified by the observation that the same 3D landmark can appear in very different images due to large viewpoint changes. For the sake of brevity, we only show the performance obtained with Residual Net [87], because it outperformed all other CNN architectures.

In conclusions, we can surprisingly notice that FREAK descriptors are outperformed by any projection in all datasets. The intuition behind this results is that as our datasets are very challenging in terms of appearance changes and perceptual aliasing, they cause a lot of noise amongst the binary descriptors. Thus, methods such as *PCA* can already beat their performance by dropping low variance directions in data.

Our evaluation confirms two important hypothesis of this paper:

1. Our approach is able to learn a powerful projection function of input descriptors, which at the same time is highly efficient as it is a simple linear mapping.

2. Context information can substantially increase the discriminability power of low level features as edges, corners, or blobs (Fig. 9.8).

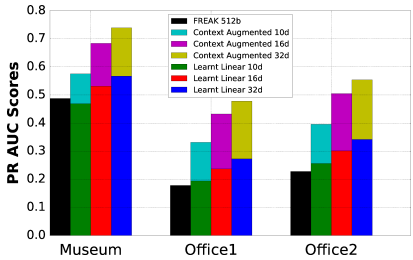


Figure 9.9: Influence of descriptor dimensionality. Comparison of learned linear projection and local context augmented projection at different projected dimensions.

5.1 Why not only learning?

Fig. 9.10 shows that the learned combination of the context descriptor and the binary descriptor actually outperforms any of the two components considered individually, despite lying in a much smaller-dimensional space. This proves that

	GPU (μs)	CPU (μs)
Linear Projection	4 ± 0.8	2.8 ± 0.9
Context augmented projection	595 ± 2	1144 ± 1.8

Table 9.3: Computational cost for one descriptor (in batch)

a complex synergy between the visual feature and its context is actually learned, but it also poses the question of why the CNN architecture does not learn automatically this interaction between the center of the image patch (from where the binary descriptor is extracted) and its neighborhood.
The intuition behind this result is that it is still not possible to learn everything from CNN architectures applied to raw image patches, and the use of handcrafted descriptors as SIFT, FREAK, or similar might still have some advantage over fully learned approaches.

Moreover, since the CNN architecture has access to an image patch that is larger than the one the original descriptor is extracted from, it can get more sensitive to large viewpoint changes. Surprisingly, and despite its simplicity, our proposed approach can learn which kind of information extracted from the CNN are actually beneficial for retrieval.

5.2 Computational cost

Table 9.3 provides the timing of our projection methods. Our GPU variant ran on a Nvidia GeForce GT 730M, while our CPU version ran on an Intel i7-core 2.50GHz processor. Our approach can still be optimized, particularly for dense computations.

Assuming to use a limited number of features for query image, *e.g* 100, we could already run a localization pipeline at around 10Hz.

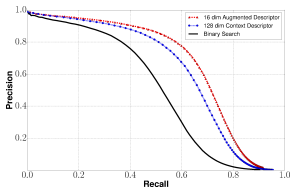


Figure 9.10: The combination between context and visual feature beats the performance of the two parts considered individually, despite its smaller dimensionality. This proves the suggested approach to yield better results than only using a single (handcrafted or learned) image patch descriptor.

6 Conclusions

In this paper, we have presented a novel approach for learning a linear and a context augmented feature projection. Even taken on its own, the linear projection beats in term of performance and computations more burdensome non-linear projection approaches such as Kernel PCA for nearest neighbor search in descriptor space. We further improved upon these results by additionally proposing a method to include context information to establish hard matches under strong appearance changes. Altogether, this resulted in a significant performance gain yielding over 50% higher (compared to binary search in original binary descriptor space) precision at the same recall and thus outperforming any other state-of-the-art linear or non-linear projection method.

While motivated by a scenario involving visual maps, the proposed methodology is of interest to a much wider audience as it can not only be applied to other map and descriptor types (e.g. LiDAR based maps) but can also be understood as a general technique for improving performance of numerous matching and retrieval tasks.

Acknowledgement

The research leading to these results has received funding from Google's project Tango.

Paper VII

LandmarkBoost: Efficient Visual Context Classifiers for Robust Localization

Marcin Dymczyk, Igor Gilitschenski, Juan Nieto, Simon Lynen, Bernhard Zeisl, Roland Siegwart

Abstract

The growing popularity of autonomous systems creates a need for reliable and efficient metric pose retrieval algorithms. Currently used approaches tend to rely on nearest neighbor search of binary descriptors to perform the 2D-3D matching and guarantee realtime capabilities on mobile platforms. These methods struggle, however, with the growing size of the map, changes in viewpoint or appearance, and visual aliasing present in the environment. The rigidly defined descriptor patterns only capture a limited neighborhood of the keypoint and completely ignore the overall visual context.

We propose LandmarkBoost-an approach that, in contrast to the conventional 2D-3D matching methods, casts the search problem as a landmark classification task. We use a boosted classifier to classify landmark observations and directly obtain correspondences as classifier scores. We also introduce a formulation of visual context that is flexible, efficient to compute, and can capture relationships in the entire image plane. The original binary descriptors are augmented with contextual information and informative features are selected by the boosting framework. Through detailed experiments, we evaluate the retrieval quality and performance of LandmarkBoost, demonstrating that it outperforms common state-of-the-art descriptor matching methods.

Published in:
Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on, 2018
In publishing

1 Introduction

Visual localization has become one of the core functionalities in robotics, being widely deployed on mobile platforms and various mobile devices. High-frequency metric 6-DoF pose estimates expressed in a global map frame enable or facilitate multiple tasks such as navigation, path-planning, obstacle avoidance, or multiagent collaboration. Frameworks that are accurate over long periods of time are also key for lifelong visual teach and repeat. These applications require reliable visual localization in the presence of illumination changes, varying weather conditions, or self-similarity of the environment.

Existing localization systems used in robotics usually rely on 2D-3D matching for precise metric pose estimation. This correspondence search is based on handcrafted features, often binary BRISK [121] or FREAK [3] descriptors, to enable realtime queries, even on platforms with limited computational resources. Their matching performance, however, suffers with the growing size of the database, visual aliasing [42], or changes in illumination and viewpoint. Intuitively, the small image patches covered by descriptor patterns are not expressive when reaching the city or even the building scale. The high dimensionality makes the search particularly challenging when covering large environments, and requires either approximate methods [161] or projection into different spaces [134].

More recent algorithms attempt to improve the performance of binary descriptors by checking for geometric consistency or by augmenting them with additional information, such as a broader neighborhood or semantic labels. Another approach is to use larger patches or the entire frame and learn the features with end-to-end methods. This leads, however, to much higher computational demands, basically requiring a costly, heavy and power-demanding GPU to be installed onboard. Additionally, most state-of-the-art systems merely consider place recognition, i.e. retrieving a topological location such as a map image, but do not provide an exact 6-DoF pose. This does not fulfill the requirements of robotic applications that rely on metric output.

We propose LandmarkBoost, a system that addresses the specific needs of mobile robotics for large-scale metric visual-based localization while keeping a limited computational cost of retrieval. Our method augments binary descriptors with visual context to build a comprehensive descriptor of the observations and improve the 2D-3D matching performance. The augmented descriptors are passed to a boosted classifier, effectively casting the matching problem as a classification. The algorithm returns 2D-3D correspondences that can be directly consumed by a 6-DoF pose estimation engine. The new features and a retrieval engine better handle the typical failure cases, such as visual aliasing, and improve the matching results, as in Fig. 10.1. The contributions of this work can be summarized as follows:

- We propose a visual context formulation that augments the binary descriptors and permits to improve the raw descriptor matching. The context is extracted from randomly generated regions that capture co-occurring statistics between landmarks within the entire image plane.

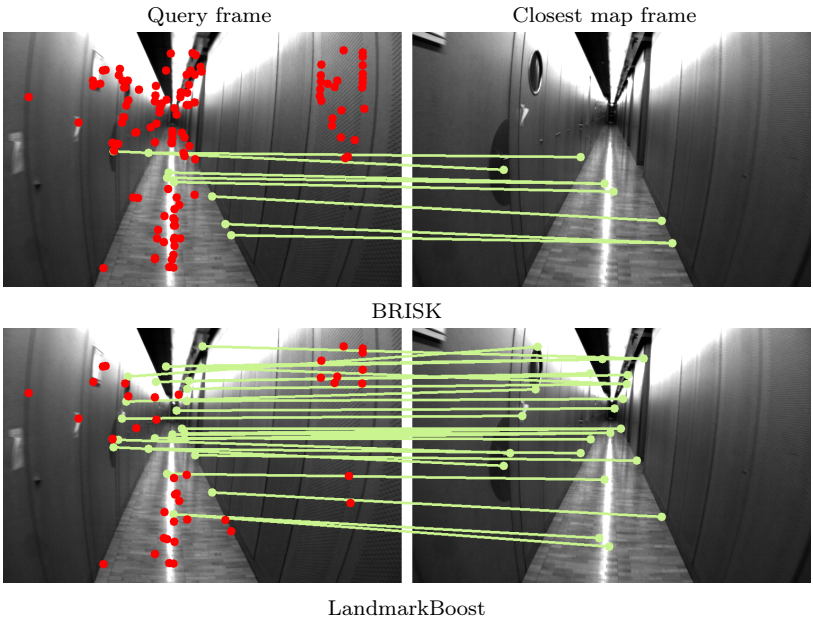


Figure 10.1: Matching result for BRISK descriptors and the proposed LandmarkBoost framework in a very self-similar environment. The left frame is a query image, the right one is the closest map frame. Red points in the query frame denote outliers, i.e. returned matches that failed geometric verification. Thanks to the inclusion of context information LandmarkBoost retrieves more geometrically consistent matches and fewer outliers.

- We introduce a boosted classifier that serves as an efficient observation-to-landmark matching engine, but also as a tool to discover useful context relationships between landmarks. The optimized implementation of the classifier enables support for large scale maps.

- We demonstrate that the runtime of the proposed approach is comparable to the conventional search methods for binary and projected descriptors. This confirms our boosted classifier can be used to perform online localization.

- We provide an extensive experimental validation that demonstrates our method outperforms the commonly-used binary descriptors both at 2D-3D correspondence search and the resulting pose estimation.

2 Related work

The principle of 2D-3D matching [185] followed by a n-point-pose (PnP) solver with RANSAC is a commonly used toolset for metric localization in robotics and mobile devices. The efficiency and scalability of such approaches [147] requires limited computational resources and permits to estimate a 6-DoF pose with high accuracy. Further optimizations to the pipeline [135] let us use such methods in real-time, making them particularly suitable for mobile robotics, enabling new applications such as collaboration between agents [59].

The quality of retrieved 3D landmark matches for a query 2D keypoint directly affects the precision and recall of pose estimation. Below, we present selected approaches that are used to improve the raw descriptor matching output.

Voting and filtering: The raw correspondence search result can be improved by filtering matches or reranking the list of candidates using some additional information. In [95], *Jégou et al.* introduce a Weak Geometric Consistency test while building a list of matches that checks if scale and angle of the retrieved matches are consistent with the query. A related approach is presented by *Zeisl et al.* [232], where spatial verification is formulated as a Hough voting problem that works in linear time. An alternative approach is proposed by *Lynen et al.* [134], where the raw descriptor matches are used to construct a voting space to only return the matches from high vote density regions. This method was further improved by a probabilistic approach to voting in [77]. All these approaches are complementary to LandmarkBoost and can be used to further refine the candidate list returned by our algorithm.

Image retrieval: Another group of approaches considers entire images instead of local features. Some of these methods depend on aggregating descriptors of local features, e.g. bag-of-words [41] or VLAD [96]. The image retrieval approaches can then be used to model places [42] and avoid potentially misleading features. They can also be used to obtain a localization prior to constrain the search space for 2D-3D correspondences [94]. Another family of methods describe the entire image without relying on keypoints, using handcrafted or learned methods, examples including GIST [165] and DIR [78] respectively. These algorithms, however, rely

on relatively expensive computations and large descriptor dimensionalities. An interesting approach is presented in [227], where a classifier is used to assign an image to one of over 26k geographic locations worldwide. It proves viability of classification methods for place recognition.

Temporal sequences: The localization queries can also be extended in the temporal dimension. Intuitively, using sequences of frames helps to distinguish places and avoid mistakes, e.g. caused by visual aliasing. Sequence-based methods can rely on traditional image comparison measures [149] or using long short-term memory (LSTM) networks [38]. Using ordered sets of frames, however, raises a question about robustness to trajectory changes and viewpoint variations, as small shifts of field-of-view can disrupt the performance [204].

Landmark covisibility: Yet another group of approaches tries to benefit from analyzing co-occurrence statistics of groups of landmarks. The assumption is that while it might be easy to confuse a single landmark observation, it is much less probable when dealing with a covisibility subgraph. *Mei et al.* [145] and later *Stumm et al.* [201, 202] present approaches that rely on landmark covisibility statistics to find reappearing patterns in the graph space. Similarly, in [133] orientation and scale of neighboring features are used.

Visual context and classifiers: Instead of landmark covisibility, we can focus just on a neighborhood of a point feature in the image plane, called visual context. In [130], *Loquercio et al.* suggest to combine point features with a local neighborhood of a fixed size. *Zhang et al.* [234] propose to build geometry-preserving visual phrases, that capture visual word co-occurrences and most importantly their spatial layout even including distant relationships in the image plane. It can be considered an extension to [231] where just local patterns were discovered. A depart from point matching is suggested in [143][129], where a bank of SVMs, operating on mid-level features, is used to obtain associations to the prior map. Unfortunately, while improving robustness, this approach reduces the metric accuracy of the pose estimates.

The method proposed in this paper aims to combine the advantages of some of the aforementioned approaches. We introduce a notion of visual context that is able to represent even long-range co-occurring feature patterns, implicitly capturing landmark covisibility instead of modelling it [202]. The context is described by sampling random region candidates and deciding during the training which of them contain useful information. This approach resembles the banks of SVMs in [143], but depends on region embeddings and simple decision stumps. Finally, we use a multi-class classifier to directly obtain 2D-3D matches which yields a precise metric localization.

3 Methodology

We present LandmarkBoost – a localization algorithm that uses a visual context and boosted classifiers to obtain 2D-3D matches between query keyframes and prior map landmarks. We assume the prior map to be a structure-from-motion

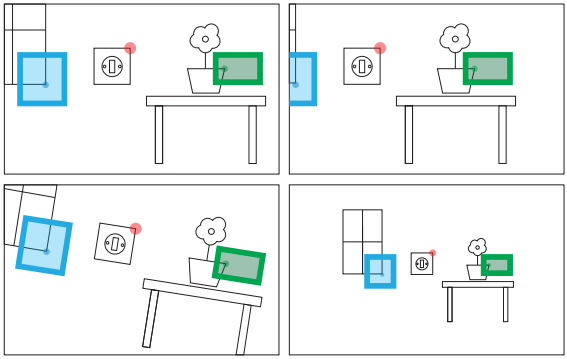


Figure 10.2: Context regions are anchored to a keypoint (red point) so that they move with the keypoint (top right). They model even distant relationships as they can be constructed anywhere in the image plane. They are rotated according to the gravity direction (bottom left). Finally, they are scaled using the keypoint detector scale to guarantee scale invariance (bottom right).

(SfM) model that contains covisibility information, detected keypoints, extracted features and 3D landmark positions. Additionally, we augment the SfM model with a concept of visual context that takes into account reoccurring feature patterns to improve the matching. In Section 3.1 we present our approach to model the visual context. Section 3.2 introduces the learning framework based on Jointboosting [212], with certain specific adaptations to the task of 2D-3D matching. The boosted classifiers are area-specific and need to be pretrained offline, using both the conventional features and visual context embeddings as input. The framework supports a large number of 3D landmarks and lets us perform visual context feature selection. Finally, in Section 3.3, we present an efficient approach to perform the classification online.

3.1 Modeling visual context

We propose to describe the visual context by means of context regions – areas in the image plane anchored to the point features. Those regions can be used to extract context embeddings that will be used to build feature vectors for context learning. The regions are common for all landmarks, i.e. the same set is applied to all observations. Initially, a large number of randomized regions is generated to guarantee a good coverage of the possible visual context locations. Then, we mine for the informative ones during subsequent training.

Context regions

Typically, feature-point descriptors are only extracted from the direct neighborhood of the keypoint location. For BRISK descriptors, the pattern covers a radius of 16 pixels. That is, however, often not sufficient to guarantee reliable matching as many corners look alike. Visual aliasing and identical objects appearing in many places within one environment result in false matches and frequent localization failures.

We propose the use of context regions anchored to the keypoint locations. Such regions can span much larger areas than the typical descriptor patterns. Their locations in the image are **relative to the keypoint location** and expressed in the normalized undistorted image plane to factor out the influence of lens effects. Additionally, the regions are scaled according to the keypoint scale to guarantee **scale invariance** and they are rotated using the gravity direction projected into the frame to ensure **rotation invariance**. The invariance properties of the regions are illustrated in Fig. 10.2.

Region generation

As we do not know *a priori* which parts of the keypoint neighborhood are useful for learning the context, our approach aims to generate a large number of regions and only later decide which are actually useful for each landmark. We use rectangular regions within the image plane, with varying areas, aspect ratios and locations.

While the area of regions is random, we guarantee that each point in the image has a uniform probability of being covered by a region of any admissible area. In other words, we want to guarantee the region area distribution to be:

$$Pr(\mathrm{area}) \propto \frac{1}{\mathrm{area}} \tag{10.1}$$

Using an inverse transform sampling method, we can sample from uniform distribution $U(m, n)$ and generate a desired region area using the inverse distribution function:

$$\mathrm{area} = F^{-1}(u) = \exp(u) \tag{10.2}$$

Region descriptors

Regions, when applied to the image, denote an area that contains visual context information. This information can be extracted as a feature vector and used for context learning. Our method describes regions using bag-of-words (BoW) embeddings [41] that are inexpensive to compute, compact and straightforward to interpret. The BoW vectors are computed per region, based on keypoint descriptors within the region, and then normalized. The visual vocabulary for BoW generation is trained using the k-means algorithm on an unrelated dataset using the same descriptor type.

3.2 Training shared classifiers

The proposed method attempts to train a classifier that takes a query feature vector v as an input (which may contain both, conventional descriptors and context embeddings) and outputs a ranked list of matching landmarks from the database. Our training algorithm is based on Jointboosting by *Torralba et al.* [212] which satisfies several of our requirements:

- the classifier is based on simple decision stumps, which makes evaluation even for tens of thousands of classes feasible and easy to parallelize,
- it relies on jointly training the decision stumps and sharing them among a set of classes so even a few observations are enough to build a reliable classifier for a landmark,
- the classifier outputs a ranked list of candidates, which can be further filtered,
- boosting algorithms inherently perform feature selection that can be used to mine for reliable visual context features.

Using a classifier instead of traditional nearest neighbor queries comes, however, at a cost. We need to train it for a specific set of landmarks, which means each area needs its own classifier. If a large environment is to be covered, the correct classifier can be preloaded based on a rough pose guess, e.g. GPS reading as proposed in [143]. Below, we always assume to deal with a single multi-class classifier.

Constructing training data

In the boosting framework, each database 3D landmark corresponds to a class. Each observation of a landmark c corresponds to a single training sample v_i with a label $z_c^i = 1$ and $\forall_{c' \neq c} z_{c'}^i = -1$. The feature vector for an observation is a concatenation of descriptors of all regions, each of length equal to the BoW vocabulary size. Additionally, we augment the feature vector with the original keypoint descriptor, see Fig. 10.3. By combining the context with the point descriptor, we let the learning algorithm decide which information reduces the misclassification cost more at a given training stage.

Training procedure

The boosted classifier H, as in [212], is a sum of M weak learners h_m:

$$H(v, c) = \sum_{m=1}^{M} h_m(v, c) \tag{10.3}$$

where c is the class index and v the feature vector. At every training round m, a new weak learner is added so the $H(v, c)$ is updated to $H'(v, c) = H(v, c) + h_m(v, c)$.

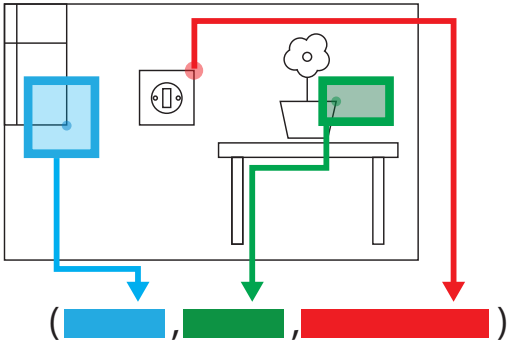


Figure 10.3: The proposed descriptor is a concatenation of descriptors of all context regions (blue and green) and the original keypoint descriptor (red). The descriptor of each region is a normalized BoW vector. In our experiments, we construct 1,000 regions and use 16 visual words to compute a BoW vector.

The new classifier is expected to reduce the overall weighted misclassification loss. The original Jointboost formulation follows the gentleboost [66] algorithm and minimizes the following weighted squared error J:

$$J = \sum_{c \in C} \sum_{i=1}^{N} w_i^c (z_i^c - h_m(v_i, c))^2 \tag{10.4}$$

The sample weight w_i^c is specific for a sample i and a class c and evolves during training. Note that the cost J is a sum over all classes C and all N training samples. The resulting weak learner function is a binary classifier that operates on a sharing set $S(n)$ and has the form as in [212]:

$$h_m(v, c) = \begin{cases} a \cdot \delta(v_i^f > \theta) + b, & \text{if } c \in S(n). \\ k^c, & \text{if } c \notin S(n). \end{cases} \tag{10.5}$$

with parameters (a, b, k^c, θ) fitted in the boosting round. For active classes $c \in S(n)$ the decision stump operates on a dimension f of the feature vector v_i and is a step function at the threshold value θ. Multiple feature dimensions $f \in F_m$ are tested in a single boosting round. For inactive classes the value depends on a class-specific constant k^c.

Efficient training of shared classifiers

The original Jointboost applications included mostly object classification, evaluated on 7- or 21-class datasets. We want to deploy the algorithm on thousands

of classes, one for each 3D landmark, and therefore we introduce a number of optimizations to the original pipeline.

Negative sample sets per class: In the original formulation, all samples of class c are used as a positive set $\mathcal{L}_c^+$ and all samples of classes other than c are used as negative samples for a class c. Instead, we introduce a negative $\mathcal{L}_c^-$ sample set per class, resulting in the following cost function:

$$\begin{aligned} J = \sum_{c \in C} (& \sum_{\mathcal{L}_c^+} w_i^c (1 - h_m(v_i, c))^2 \\ & + \sum_{\mathcal{L}_c^-} w_i^c (1 + h_m(v_i, c))^2) \end{aligned} \tag{10.6}$$

The limited size of the negative set addresses the heavy imbalance of the number of positive and negative samples per class compared to the original formulation. Additionally, this change significantly reduces the complexity of training, as we only evaluate $|\mathcal{L}_c^+| + |\mathcal{L}_c^-|$ samples per class.

Boot-strapping negative samples: After introducing a predefined negative sample set per class $\mathcal{L}_c^-$, care needs to be taken to guarantee this set includes hard negative samples. We use a landmark quantizer to find them, assuming they quantize to the same visual words as the positive samples of the class c. This approach follows the Nearest Neighbor Negatives concept introduced in [171]. Moreover, we do not add any negative samples that are observations of landmarks located nearby in the metric space – these might be duplicates of the same landmark that did not get associated correctly by the SfM pipeline.

Hard negative mining: We further augment the negative set $\mathcal{L}_c^-$ while training. We add samples of classes that were misclassified as positives of a class c. In this way, the initial negative sample set is getting extended by samples that refine the classifier [206].

Efficient sharing set initialization: In [212], *Torralba et al.* propose a greedy approach with a $\mathcal{O}(|C|^2)$ complexity to select the sharing set $S(n)$. Given the feature dimension f, we believe it is possible to initialize the sharing set based on v_i^f values. Classes that expose consistent behavior (e.g. majority of values are either zeros or very large) can be grouped together to form a candidate set $S(n)$ that would classify them against the rest. While initial sharing sets are further refined, fewer iterations of the selection loop are necessary.

Efficient sharing set update: The parameters a, b and k^c are refitted [213] at each sharing set estimation step. We propose to incrementally update them. E.g., when adding a class c' to the sharing set, we perform an update of $b_S = b_{num}/b_{den}$:

$$b'_S(f, \theta) = \frac{b_{num} + b_c(f, \theta) w_c'^+(f, \theta)}{b_{den} + w_c'^+(f, \theta)} \tag{10.7}$$

using precomputed values of $b_c(f, \theta)$ and $w_c'^+(f, \theta)$. While this change might not seem significant, we have to keep in mind that decision stumps are fitted $|F_m| \cdot |C|^2$

times per boosting round so this optimization has a high impact on the total runtime.

Background class: The classifiers in [194, 213] never return a "no match" answer. We introduce a background class that consists of samples that belong to untracked keypoints or landmarks not present in the database. A match to the background class means that the observation could not be matched to any of the known landmarks and should not be used for pose estimation. This increases the inlier ratio of the PnP+RANSAC algorithm.

3.3 Localization through Classification

The shared decision stumps h_m, after training as described in Section 4.1, can be used to establish 2D-3D matches, i.e. matches between the keypoints of the query frame and the 3D landmarks stored in the database (a prior map). The localization works as follows:

1. The context embedding is computed by adapting the regions to the keypoint location and scale. For each of the regions, a region descriptor is computed.

2. The context embeddings of each keypoint are combined with its descriptor to create a feature vector v.

3. The feature vector v is classified by M shared decision stumps:

$$c_{best} = \underset{c \in C}{\operatorname{argmax}} \sum_{m=1}^{M} h_m(v, c) \tag{10.8}$$

4. The keypoint-landmark pairs are translated into 2D-3D matches to estimate the pose using PnP and RANSAC algorithms [110].

The classification scheme presented above evaluates the classifier for all possible classes C. This might impair the performance when the classifier covers large areas with many landmarks. We propose to mitigate this issue by using an inverted file of database landmark quantizations. Given a query point descriptor, we can only test the candidates that quantize to the same visual words w, effectively reducing the set C to a subset $C_w \subset C$.

4 Experimental evaluation

The experimental evaluation verifies both the performance of the training procedure, as well as landmark and pose retrieval statistics. In Section 4.1 we describe the training procedure. Then, in Section 4.2 we evaluate the landmark retrieval, showing that the suggested approach returns more inlier matches than the baseline methods and that the correct matches are ranked higher. In Section 4.3 we provide a comprehensive evaluation of the pose retrieval. Finally, Section 4.4 presents a

Dataset	Notes	Distance	Descriptors	Landmarks	Frames
CLA F	Train	221m	405k	41.9k	1737
	Eval	199m	465k	39.5k	1849
Zurich Old Town	Train	2441m	583k	35.2k	2009
	Eval	438m	243k	15.2k	565

Table 10.1: Train and evaluation datasets. The datasets are public and available with the maplab mapping framework [187].

Figure 10.4: A non-matching pair of frames from the CLA F dataset. Even though they look alike, the places are more than 20m apart. Using just keypoint descriptors leads to a localization failure due to the extreme visual aliasing. Visual context helps to disambiguate the locations – small cues let us reject false positive landmark matches.

comparison of the runtime and inlier ratio between LandmarkBoost and baseline methods.

We use the implementation of Jointboosting by *Krähenbühl et al.* [114], with the improvements described in Section 3.2. The point descriptors are 384-bit BRISK [121], extracted by the maplab [187] mapping framework. The proposed algorithm is compared to baseline methods, namely exact binary descriptor (BF-Matcher) search implemented by OpenCV [23] and exact projected descriptor search using libnabo [56]. We directly feed the 2D-3D matches provided by the search methods to a localization engine based on PnP+RANSAC provided by opengv [110]. No match clustering or majority voting is used to avoid distorting the results of raw correspondence search.

The evaluation is performed on two sets of datasets: very self-similar indoor environment (see Fig. 10.4) and a large-scale outdoor dataset. The details of the datasets are given in Table 10.1. For each set of datasets, all but one are used to build a database (i.e. build a kd-tree or train a classifier) and the remaining one for evaluation. The goal is to retrieve relevant descriptors or localize each frame of the evaluation dataset.

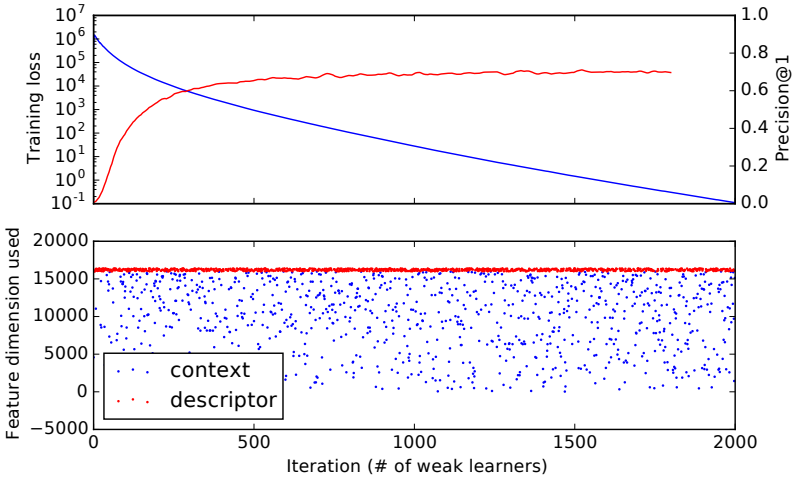


Figure 10.5: The training log of a 7,500-landmark classifier. The boosting algorithm successfully reduced the training cost (blue) at each iteration. The precision@1 (red) of the evaluation set classification steadily grows to reach the level of 72.8%. The feature vectors contain 16,000 context dimensions (1,000 regions with 16 visual words each) and 384 descriptor dimensions. Overall, 985 context features and 1,015 descriptor features were selected by the algorithm. Some of the feature dimensions were used by multiple decision stumps and can be combined to speed up the classification.

4.1 Classifier training

The classifier for the CLA F dataset was trained on 7,500 best landmarks, selected by landmark quality [49, 50]. The training fitted 2,000 decision stumps, used 1,000 regions and evaluated 500 features at each round. Further increasing the region count did not bring significant improvements to the matching performance. The training progress, that took about 8 hours, is depicted in Fig. 10.5. The fitted weak learners used both keypoint descriptors as well as context embeddings. This indicates the provided visual context contains useful information that helps to establish 2D-3D matches.

We have also evaluated the sharing set size over the boosting iterations. Intuitively, initial sharing sets should contain about 50% of classes as such splits permit to reduce the classification cost most efficiently. Later on, the sharing set size is getting smaller as more detailed splits are necessary to improve the accuracy. This expected behavior indeed takes place when training, as depicted in Fig. 10.6.

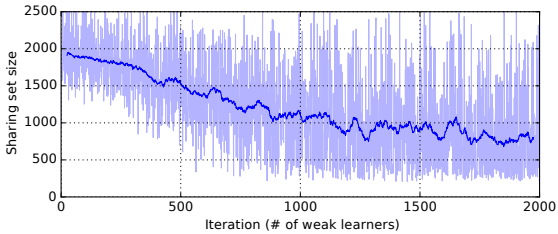


Figure 10.6: Evolution of the sharing set size for 5,000 landmarks. The size of the sharing set over the boosting rounds follows the intuition. Initially, the largest reduction of the misclassification cost is caused by large sharing set, approximately dividing the class set C in two. As the training progresses, more fine-grained splits are prevalent. The noisy nature of the sharing set size is a result of random candidate feature dimensions F_m, greedy sharing set updates and evolving sample weights w_i^c.

	4vw	8vw	16vw	32vw	64vw	128vw
Precision@1	0.339	0.419	**0.744**	0.703	0.644	0.567
MRR	0.361	0.598	**0.851**	0.825	0.786	0.7246

Table 10.2: Precision@1 and Mean Reciprocal Rank (MRR) of the Jointboost classifier for selected numbers of visual words (vw). The classifier was trained solely on the visual context embeddings, without the raw descriptor part. The quality of retrieval is affected by the BoW visual vocabulary size used to describe the regions. Too few visual words means low context embedding uniqueness, too many lead to quantization errors.

All of the above classifiers as well as all the subsequent evaluations use a BoW vocabulary that contains 16 visual words. This has proven to be an optimal value for descriptor retrieval as ilustrated in Table 10.2.

4.2 Landmark retrieval

In this section, the quality of the returned 2D-3D matches is evaluated. The precision and recall of the landmark retrieval has a direct influence on the quality and runtime of subsequent pose retrieval. A large fraction of false matches (outliers) might prevent the pose estimation algorithms to output a valid solution within a limited time. We are therefore interested in the precision of the first retrieved candidate, but also the rank of the true positive.

Following the benchmarking methodology of the Caltech Pedestrian Detection

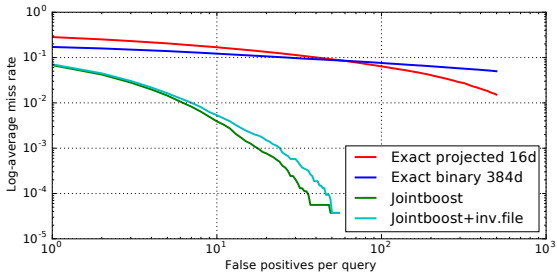


Figure 10.7: Average miss rate curve for landmark retrieval of CLA F. The proposed jointboost-based approaches deliver significantly reduced average miss rates when compared to the traditional binary and projected descriptors. The difference increases with the growing number of retrieved candidates. Slightly worse quality of Jointboosting with an inverted file results from quantization errors.

	Binary 384d	Projected 16d	Jboost	Jboost+inv.file
Precision@1	0.8081	0.6578	**0.8592**	0.8571
MRR	0.8306	0.7163	**0.9118**	0.9096

Table 10.3: Precision@1 and Mean Reciprocal Rank (MRR) of descriptor retrieval quality. The first statistic indicates that the Jointboosting-based methods have a higher probability of returning a correct first candidate. The second one shows that the true positive is on average ranked higher by the proposed methods.

Benchmark [45], we decided to use the average miss rate and the number of false positives per query to evaluate the retrieval results. This choice is motivated by the fact that in landmark retrieval only a single landmark class is a true positive. Fig. 10.7 illustrates the results of the proposed and baseline methods. It shows that significantly fewer candidates need to be retrieved using our approach compared to the baseline to obtain a relevant candidate. This is confirmed by statistics in Table 10.3, where both the precision of the first candidate as well as the Mean Reciprocal Rank are superior for LandmarkBoost.

Overall, our results indicate that LandmarkBoost outputs more accurate correspondences, measured both as the reliability of first reported match as well as the position of the true positive within a ranked candidate list.

4.3 Pose retrieval

The aim of this section is to evaluate how the superior landmark retrieval results reported in Section 4.2 translate into the pose retrieval, the ultimate goal of the localization systems. 2D-3D correspondences retrieved by the matching algorithms are passed to the pose estimation block performing RANSAC on PnP estimates. The inlier distance threshold and the maximum number of RANSAC iterations (500) are fixed over all evaluations.

The retrieved poses are then evaluated against the ground-truth values refined using a batch visual-inertial least squares optimization as in [135]. The thresholds of 20cm and 5deg are used as a threshold. All frames of the query datasets are used in this evaluation.

The results are presented in Fig. 10.8. LandmarkBoost methods bring a boost in performance, especially for the high recall part of the curve, where the precision of the conventional methods drops rapidly. This permits to correctly localize more frames, without sacrificing precision of the output.

4.4 Matching runtime evaluation

In this section, we present an evaluation of the classification (or search) runtime. We decided to assess the runtime with respect to the inlier ratio to illustrate the trade-off between the matching quality and computational requirements. As shown in Fig. 10.9, the proposed methods deliver the quality that is on par with the binary descriptor retrieval, but at a reduced query time, particularly for the classification using a set of candidates from the quantizer inverted file.

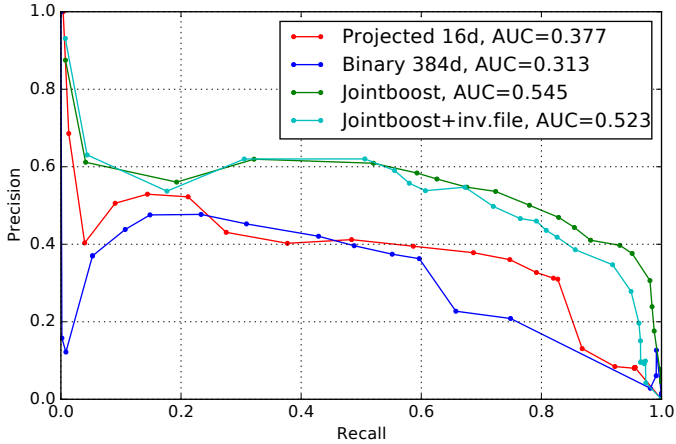
We believe that the runtime of Jointboost classification can be further reduced if necessary. The classifier evaluation consists of a series of binary tests and simple score arithmetic. These operations can be easily parallelized, further optimized using CPU vector instructions or even deployed on a GPU.

5 Conclusions

In this paper, we have presented LandmarkBoost, a novel approach for precise, metric localization. We have shown it is possible to construct a boosted classifier that operates on a large number of classes and delivers better performance than conventional, search based methods. The classifier incorporates not only the local point features, but also the visual context by mining for useful information that consistently reappears with a given landmark. This way, we achieve a higher level of robustness against visual aliasing or appearance changes compared to raw binary features. Finally, we have presented a comprehensive evaluation that demonstrates superior descriptor and pose retrieval quality and shows real-time capability for robotic applications.

While this paper focuses on binary descriptors and visual context based on neighboring keypoints, LandmarkBoost can easily accommodate different data types while remaining very efficient. The proposed formulation is ready to work with

(a) CLA F



(b) Zurich Old Town

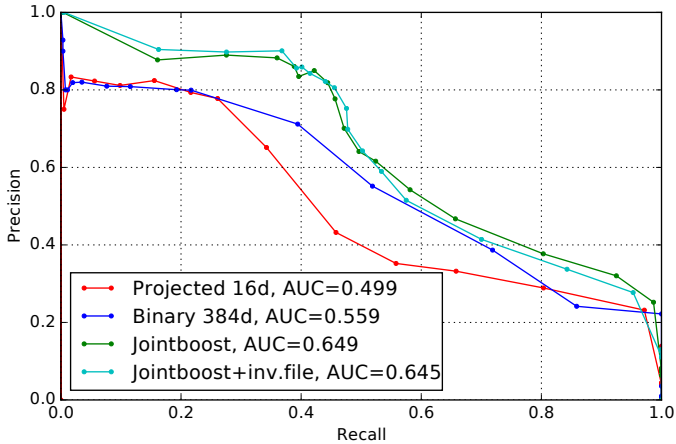


Figure 10.8: The precision-recall curves and the Area Under the Curve (AUC) for 6-DoF pose retrieval. Both proposed LandmarkBoost methods outperform the baseline binary and projected descriptor search approaches.

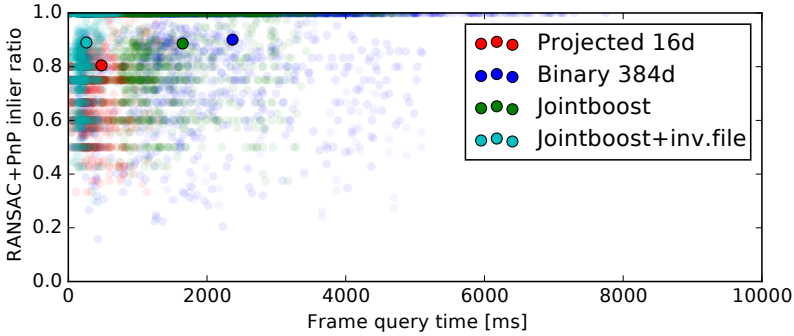


Figure 10.9: Runtime vs average inlier ratio. Mean runtime-inlier ratio pairs for each method are marked with full intensity markers. The figure presents the trade-off between the quality of landmark matches and the query time (single core of Intel i7-7820X@3.6GHz). The proposed Jointboost method delivers inlier ratios close to the baseline, but with about 30% shorter runtime. Using an inverted file for classification (see Section 3.3) further reduces the computational cost of LandmarkBoost.

other feature descriptors or region descriptors. We also believe that extending the framework to other kinds of data, such as semantics or object labels, might further improve the performance when compared to conventional descriptor search-based localization.

Bibliography

[1] S. Agarwal and K. Mierle. Ceres solver. http://ceres-solver.org.

[2] M. Agrawal and K. Konolige. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5), October 2008.

[3] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast Retina Keypoint. In *European Conf. on Computer Vision*, 2012.

[4] R. Allen, N. MacMillan, D. Marinakis, R. I. Nishat, R. Rahman, and S. Whitesides. The range beacon placement problem for robot navigation. In *Computer and Robot Vision (CRV), Canadian Conference on*. IEEE, 2014.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[6] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide Area Localization on Mobile Phones. In *IEEE Int. Symp. On Mixed and Augmented Reality*, 2009.

[7] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In *IEEE Int. Symp. On Mixed and Augmented Reality*, 2011.

[8] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *Int. Conf. on Computer Vision*, pages 1269–1277, 2015.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (12):2481–2495, 2017.

[10] T. Bailey and E. Nebot. Localisation in large-scale environments. *Robotics and Autonomous Systems*, 37(4):261–281, 2001.

[11] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016.

[12] D. Barnes, W. Maddern, G. Pascoe, and I. Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 1894–1900. IEEE, 2018.

[13] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conf. on Computer Vision*. 2006.

[14] M. Beinhofer, J. Müller, and W. Burgard. Near-optimal landmark selection for mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation*, pages 4744–4749. IEEE, 2011.

[15] M. Beinhofer, J. Müller, and W. Burgard. Effective landmark placement for accurate and reliable mobile robot navigation. *Robotics and Autonomous Systems*, 61(10):1060 – 1069, 2013.

[16] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[17] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[18] B. Bescós, J. M. Fácil, J. Civera, and J. Neira. Dynslam: Tracking, mapping and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.

[19] F. Blöchliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. In *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018.

[20] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[21] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1899–1906, 2003.

[22] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *The Int. Journal of Robotics Research*, 2004.

[23] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[24] S. Buoncompagni, D. Maio, D. Maltoni, and S. Papi. Saliency-based keypoint selection for fast object detection and matching. *Pattern Recognition Letters*, 62:32–40, 2015.

[25] M. Bürki, I. Gilitschenski, E. Stumm, R. Siegwart, and J. Nieto. Appearance-based landmark selection for efficient long-term visual localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4137–4143. IEEE, 2016.

[26] M. Bürki, M. Dymczyk, I. Gilitschenski, C. Cadena, R. Siegwart, and J. Nieto. Map management for efficient long-term visual localization in outdoor environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 682–688. IEEE, 2018.

[27] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart. Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1872–1878. IEEE, 2015.

[28] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. *The Int. Journal of Robotics Research*, 2016.

[29] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[30] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek. A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array. In *VLSI Circuits (VLSIC), 2013 Symposium on*, pages C182–C183. IEEE, 2013.

[31] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice. Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics*, 2014. In Press.

[32] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of michigan north campus long-term vision and lidar dataset. *The Int. Journal of Robotics Research*, page 0278364915614638, 2015.

[33] G. Carneiro and A. D. Jepson. The quantitative characterization of the distinctiveness and robustness of local image descriptors. *Image and Vision Computing*, 27(8):1143–1156, 2009.

[34] S. H. Cen and P. Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *IEEE Int. Conf. on Robotics and Automation*, pages 1–8. IEEE, 2018.

[35] W. Churchill and P. Newman. Experience-based Navigation for Long-term Localisation. *The Int. Journal of Robotics Research*, 2013.

[36] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Siegwart. Map API - scalable decentralized map building for robots. In *IEEE Int. Conf. on Robotics and Automation*, 2015.

[37] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Y. Siegwart. Map api-scalable decentralized map building for robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 6241–6247. IEEE, 2015.

[38] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[39] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3995–4001, 2017.

[40] L. Clement and J. Kelly. How to train a cat: Learning canonical appearance transformations for direct visual localization under illumination change. *IEEE Robotics and Automation Letters*, 3(3):2447–2454, 2018.

[41] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, 2004.

[42] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The Int. Journal of Robotics Research*, 2008.

[43] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (6):1052–1067, 2007.

[44] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. *arXiv preprint arXiv:1712.07629*, 2017.

[45] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4), 2012.

[46] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Int. Conf. on Machine Learning*, pages 647–655, 2014.

[47] Z. Dong, G. Zhang, J. Jia, and H. Bao. Keyframe-Based Real-Time Camera Tracking. In *Int. Conf. on Computer Vision*, 2009.

[48] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena. SegMap: 3d segment mapping using data-driven descriptors. In *Robotics: Science and Systems*, 2018.

[49] M. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart. Keep it brief: Scalable creation of compressed localization maps. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2536–2542. IEEE, 2015.

[50] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale. The gist of maps-summarizing experience for lifelong localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 2767–2773. IEEE, 2015.

[51] M. Dymczyk, I. Gilitschenski, R. Siegwart, and E. Stumm. Map summarization for tractable lifelong mapping. In *RSS Workshop on Geometry and Beyond*, 2016.

[52] M. Dymczyk, T. Schneider, I. Gilitschenski, R. Siegwart, and E. Stumm. Erasing bad memories: Agent-side summarization for long-term mapping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4572–4579. IEEE, 2016.

[53] M. Dymczyk, E. Stumm, J. Nieto, R. Siegwart, and I. Gilitschenski. Will it last? learning stable features for long-term visual localization. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 572–581. IEEE, 2016.

[54] M. Dymczyk, M. Fehr, T. Schneider, and R. Siegwart. Long-term large-scale mapping and localization using maplab. *arXiv preprint arXiv:1805.10994*, 2018.

[55] M. Dymczyk, I. Gilitschenski, J. Nieto, S. Lynen, B. Zeisl, and R. Siegwart. Landmarkboost: Efficient visual context classifiers for robust localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2018.

[56] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter. Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics*, 2012.

[57] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular slam. In *European Conf. on Computer Vision*, 2014.

[58] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical slam: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[59] P. Fankhauser, M. Bloesch, P. Krüsi, R. Diethelm, M. Wermelinger, T. Schneider, M. Dymczyk, M. Hutter, and R. Siegwart. Collaborative navigation for flying and walking robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2859–2866. IEEE, 2016.

[60] M. Fehr, M. Dymczyk, S. Lynen, and R. Siegwart. Reshaping our model of the world over time. In *IEEE Int. Conf. on Robotics and Automation*, pages 2449–2455. IEEE, 2016.

[61] M. Fehr, T. Schneider, M. Dymczyk, J. Sturm, and R. Siegwart. Visual-inertial teach and repeat for aerial inspection. *arXiv preprint arXiv:1803.09650*, 2018.

[62] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981.

[63] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative monocular slam with multiple micro aerial vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013.

[64] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation*, 2014.

[65] J.-M. Frahm and others (19 authors). Building rome on a cloudless day. In *European Conf. on Computer Vision*, pages 368–381. Springer, 2010.

[66] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2), 2000.

[67] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Mühlfellner, S. Wonneberger, J. Timpner, S. Rottmann, B. Li, et al. Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 809–816. IEEE, 2013.

[68] P. T. Furgale and T. D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010. doi: 10.1002/rob.20342.

[69] F. Furrer, T. Novkovic, M. Fehr, A. Gawel, M. Grinvald, T. Sattler, R. Siegwart, and J. Nieto. Incremental object database: Building 3d models from multiple partial observations. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018.

[70] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[71] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[72] D. Galvez-Lopez and J. D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics and Automation*, 2012.

[73] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5), 2012.

[74] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[75] S. Garg, N. Suenderhauf, and M. Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. In *Robotics: Science and Systems*, 2018.

[76] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3(3):1687–1694, 2018.

[77] M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart. Visual place recognition with probabilistic voting. In *IEEE Int. Conf. on Robotics and Automation*, pages 3192–3199. IEEE, 2017.

[78] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *European Conf. on Computer Vision*, 2016.

[79] J. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst Bemerkungen über seine Anwendungen in der Geodäisie. *Grunerts Archiv fiir Mathematik und Physik*, 1841.

[80] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[81] J. Guivant, E. Nebot, J. Nieto, and F. Masson. Navigation and mapping in large unstructured environments. *The Int. Journal of Robotics Research*, 23 (4-5):449–472, 2004.

[82] I. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL http://www.gurobi.com.

[83] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2006.

[84] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[85] W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[86] M. Havlena, W. Hartmann, and K. Schindler. Optimal reduction of large image databases for location recognition. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, 2013.

[87] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[88] K. He, Y. Lu, and S. Sclaroff. Local descriptors optimized for average precision. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2018.

[89] T. Hinzmann, T. Schneider, M. Dymczyk, A. Melzer, T. Mantel, R. Siegwart, and I. Gilitschenski. Robust map generation for fixed-wing uavs with low-cost highly-oblique monocular cameras. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3261–3268. IEEE, 2016.

[90] T. Hinzmann, T. Schneider, M. Dymczyk, A. Schaffner, S. Lynen, R. Siegwart, and I. Gilitschenski. Monocular visual-inertial slam for fixed-wing uavs using sliding window based nonlinear optimization. In *IEEE International Symposium On Visual Computing*, pages 569–581. Springer, 2016.

[91] G. Huang, M. Kaess, and J. J. Leonard. Consistent sparsification for graph optimization. In *European Conf. on Mobile Robots*. IEEE, 2013.

[92] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart. Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion. In *Adaptive Mobile Robotics*, pages 483–490. World Scientific, 2012.

[93] J. G. Ibrahim, M.-H. Chen, and D. Sinha. *Bayesian survival analysis*. Springer Science & Business Media, 2001.

[94] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[95] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conf. on Computer Vision*, pages 304–317. Springer, 2008.

[96] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[97] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012.

[98] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[99] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard. Temporally scalable visual slam using a reduced pose graph. In *IEEE Int. Conf. on Robotics and Automation*, pages 54–61. IEEE, 2013.

[100] E. Johns and G.-Z. Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *IEEE Int. Conf. on Robotics and Automation*, pages 3212–3218, May 2013. doi: 10.1109/ICRA.2013.6631024.

[101] E. Johns and G.-Z. Yang. Generative methods for long-term place recognition in dynamic scenes. *Int. Journal of Computer Vision*, 106(3):297–314, 2014.

[102] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[103] S. Karayev, A. Hertzmann, H. Winnemoeller, A. Agarwala, and T. Darrell. Recognizing image style. *CoRR*, abs/1311.3715, 2013. URL http://arxiv.org/abs/1311.3715.

[104] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

[105] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[106] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1): 359–392, 1998.

[107] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2938–2946, 2015.

[108] H. J. Kim, E. Dunn, and J.-M. Frahm. Predicting Good Features for Image Geo-Localization Using Per-Bundle VLAD. In *Int. Conf. on Computer Vision*, 2015.

[109] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE Int. Symp. On Mixed and Augmented Reality*, pages 225–234. IEEE, 2007.

[110] L. Kneip and P. Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *IEEE Int. Conf. on Robotics and Automation*, 2014.

[111] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *European Conf. on Computer Vision*. Springer, 2010.

[112] K. Konolige and J. Bowman. Towards lifelong visual maps. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1156–1163. IEEE, 2009.

[113] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *The Int. Journal of Robotics Research*, 29(8): 941–957, 2010.

[114] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Neural Information Processing Systems (NIPS)*, 2011.

[115] H. Kretzschmar and C. Stachniss. Information-theoretic compression of pose graphs for laser-based slam. *The Int. Journal of Robotics Research*, 2012. doi: 10.1177/0278364912455072.

[116] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Neural Information Processing Systems (NIPS)*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[117] Y. Latif, R. Garg, M. Milford, and I. Reid. Addressing challenging place recognition tasks using generative adversarial networks. In *IEEE Int. Conf. on Robotics and Automation*, pages 2349–2355, 2018.

[118] J. C. Lee, R. Dugan, et al. Google project tango. https://www.google.com/atap/projecttango/.

[119] J. Leonard and P. Newman. Consistent, convergent, and constant-time slam. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1143–1150, 2003.

[120] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Robotics Research*, pages 169–176. 2000.

[121] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Int. Conf. on Computer Vision*, 2011.

[122] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The Int. Journal of Robotics Research*, 34(3):314–334, 2015.

[123] F. Li and J. Kosecka. Probabilistic location recognition using reduced feature set. In *IEEE Int. Conf. on Robotics and Automation*, 2006.

[124] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *IEEE Int. Conf. on Robotics and Automation*, pages 7286–7291, 2018.

[125] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *European Conf. on Computer Vision*. 2010.

[126] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conf. on Computer Vision*, 2010.

[127] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time Image-based 6-DOF Localization in Large-Scale Environments. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[128] C. Linegar, W. Churchill, and P. Newman. Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation. In *IEEE Int. Conf. on Robotics and Automation*, Seattle, USA, 2015.

[129] C. Linegar, W. Churchill, and P. Newman. Made to Measure: Bespoke Landmarks for 24-Hour, All-Weather Localisation with a Camera. In *IEEE Int. Conf. on Robotics and Automation*, pages 787–794. IEEE, may 2016. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487208.

[130] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart. Efficient descriptor learning for large scale localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 3170–3177. IEEE, 2017.

[131] D. G. Lowe. Object recognition from local scale-invariant features. In *Int. Conf. on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

[132] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[133] S. Lowry and H. Andreasson. Logos: Local geometric support for high-outlier spatial verification. In *IEEE Int. Conf. on Robotics and Automation*, May 2018.

[134] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart. Placeless Place-Recognition. In *Proceedings of International Conference on 3D Vision (3DV)*, 2014.

[135] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, Rome, Italy, July 2015.

[136] W. Maddern, M. Milford, and G. Wyeth. Towards persistent indoor appearance-based localization, mapping and navigation using cat-graph. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4224–4230. IEEE, 2012.

[137] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *The Int. Journal of Robotics Research*, 36(1):3–15, 2017.

[138] W. P. Maddern, M. Milford, and G. Wyeth. Towards persistent localization and mapping with a continuous appearance-based topology. In *Robotics: Science and Systems*, 2012.

[139] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige. The office marathon: Robust navigation in an indoor office environment. In *IEEE Int. Conf. on Robotics and Automation*, 2010.

[140] M. Mauro, H. Riemenschneider, A. Signoroni, R. Leonardi, and L. V. Gool. An integer linear programming model for view selection on overlapping camera clusters. In *Proceedings of International Conference on 3D Vision (3DV)*, 2014.

[141] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric object-level slam. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 32–41. IEEE, 2018.

[142] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. Leonard. Real-time 6-DOF multi-session visual SLAM over large-scale environments. *Robotics and Autonomous Systems*, 2013. ISSN 09218890. doi: 10.1016/j.robot.2012.08.008.

[143] C. McManus, B. Upcroft, and P. Newmann. Scene signatures: Localised and point-less features for localisation. In *Robotics: Science and Systems*, 2014.

[144] C. McManus, B. Upcroft, and P. Newman. Learning place-dependant features for long-term vision-based localisation. *Autonomous Robots*, 39(3):363–387, jul 2015. ISSN 0929-5593. doi: 10.1007/s10514-015-9463-y.

[145] C. Mei, G. Sibley, and P. Newman. Closing loops without places. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[146] H. Merzić, E. Stumm, M. Dymczyk, R. Siegwart, and I. Gilitschenski. Map quality evaluation for visual localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 3200–3206. IEEE, 2017.

[147] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF Localization on Mobile Devices. In *European Conf. on Computer Vision*. 2014.

[148] M. Milford and G. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE Int. Conf. on Robotics and Automation*, pages 1643–1649, May 2012. doi: 10.1109/ICRA.2012.6224623.

[149] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE Int. Conf. on Robotics and Automation*, 2012.

[150] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena. Tsdf manifolds: A scalable and consistent dense mapping approach. *arXiv preprint arXiv:1710.07242*, 2017.

[151] P. Moulon, P. Monasse, R. Perrot, and R. Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[152] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint tkalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. on Robotics and Automation*, pages 3565–3572. IEEE, 2007.

[153] P. Muehlfellner, P. T. Furgale, W. Derendarz, and R. Philippsen. Evaluation of fisheye-camera based visual multi-session localization in a real-world scenario. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 56–62, Gold Coast, Australia, 23–26 June 2013. doi: 10.1109/IVWorkshops.2013.6615226.

[154] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale. Summary maps for lifelong visual localization. *Journal of Field Robotics*, 33(5):561–590, 2016.

[155] R. Mur-Artal and J. D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 2017.

[156] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 2015.

[157] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

[158] J. I. Nieto, J. E. Guivant, and E. M. Nebot. The hybrid metric maps (hymms): A novel map representation for denseslam. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 391–396.

[159] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart. A synchronized visual-inertial sensor system with fpga preprocessing for accurate real-time slam. In *IEEE Int. Conf. on Robotics and Automation*, 2014.

[160] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[161] M. Norouzi, A. Punjani, and D. J. Fleet. Fast search in hamming space with multi-index hashing. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[162] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart. Design of small hand-launched solar-powered uavs: From concept study to a multi-day world endurance record flight. *Journal of Field Robotics*, 34(7):1352–1377, 2017.

[163] H. Oleynikova, M. Burri, S. Lynen, and R. Siegwart. Real-time visual-inertial localization for aerial and ground robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[164] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.

[165] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. Journal of Computer Vision*, 2001.

[166] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *IEEE Int. Conf. on Robotics and Automation*, pages 3400–3407. IEEE, 2011.

[167] H. S. Park, Y. Wang, E. Nurvitadhi, J. Hoe, Y. Sheikh, and M. Chen. 3d point cloud reduction using mixed-integer quadratic programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2013.

[168] R. Paul and P. Newman. Self help: Seeking out perplexing images for ever improving navigation. In *IEEE Int. Conf. on Robotics and Automation*, pages 445–451. IEEE, 2011.

[169] E. Pepperell, P. I. Corke, and M. J. Milford. All-environment visual place recognition with smart. In *IEEE Int. Conf. on Robotics and Automation*, pages 1612–1618, May 2014. doi: 10.1109/ICRA.2014.6907067.

[170] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[171] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *European Conf. on Computer Vision*. Springer, 2010.

[172] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *Int. Conf. on Computer Vision*, volume 1, pages 496–501. IEEE, 1999.

[173] H. Porav, W. Maddern, and P. Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018.

[174] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[175] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *IEEE Int. Conf. on Robotics and Automation*, 2009.

[176] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017.

[177] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *IEEE Int. Conf. on Robotics and Automation*, pages 4304–4311. IEEE, 2016.

[178] L. Riazuelo, J. Civera, and J. Montiel. C2tam: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 2013.

[179] D. M. Rosen, J. Mason, and J. J. Leonard. Towards lifelong feature-based mapping in semi-static environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 1063–1070, may 2016. doi: 10.1109/ICRA.2016.7487237.

[180] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[181] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Int. Conf. on Computer Vision*, pages 2564–2571. IEEE, 2011.

[182] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Int. Journal of Computer Vision*, pages 1–42, April 2015. doi: 10.1007/s11263-015-0816-y.

[183] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *Int. Journal of Computer Vision*, 105(3):222–245, 2013.

[184] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, 2018.

[185] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Int. Conf. on Computer Vision*, pages 667–674. IEEE, 2011.

[186] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, 2018.

[187] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 3(3): 1418–1425, 2018.

[188] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer Berlin Heidelberg, 1997.

[189] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic visual localization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[190] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Neural Information Processing Systems (NIPS)*, pages 41–48, 2004.

[191] S. Se, D. G. Lowe, and J. J. Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[192] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2012.

[193] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 806–813, 2014.

[194] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conf. on Computer Vision*, 2006.

[195] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Int. Conf. on Computer Vision*, 2015.

[196] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[197] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014.

[198] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Int. Conf. on Computer Vision*, pages 1470–1477, 2003.

[199] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[200] E. Stenborg, C. Toft, and L. Hammarstrand. Long-term visual localization using semantically segmented images. *arXiv preprint arXiv:1801.05269*, 2018.

[201] E. Stumm, C. Mei, S. Lacroix, J. Nieto, M. Hutter, and R. Siegwart. Robust visual place recognition with graph kernels. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[202] E. S. Stumm, C. Mei, and S. Lacroix. Building location models for visual place recognition. *The Int. Journal of Robotics Research*, 35(4), 2016.

[203] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.

[204] N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[205] N. Sünderhauf, S. Shirazi, A. Jacobson, E. Pepperell, F. Dayoub, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, 2015.

[206] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1), 1998.

[207] C. Sweeney. Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org.

[208] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[209] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard. Robust mapping and localization in indoor environments using sonar data. *The Int. Journal of Robotics Research*, 21(4):311–330, 2002.

[210] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[211] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Int. Conf. on Computer Vision*, pages 839–846. IEEE, 1998.

[212] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[213] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(5), 2007.

[214] L. Traffelet, T. Eppenberger, A. Millane, T. Schneider, and R. Siegwart. Target-based calibration of underwater camera housing parameters. In *IEEE Int. Symposium on Safety, Security, and Rescue Robotics*, 2016.

[215] T. Trzcinski, V. Lepetit, and P. Fua. Thick boundaries in binary space and their influence on nearest-neighbor search. *Pattern Recognition Letters*, 33 (16):2173–2180, 2012.

[216] P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2109–2116. IEEE, 2009.

[217] C. Valgren and A. J. Lilienthal. Sift, surf & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010.

[218] D. Van Opdenbosch, T. Aykut, N. Alt, and E. Steinbach. Efficient map compression for collaborative visual slam. In *Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on*, pages 992–1000. IEEE, 2018.

[219] J. Ventura and T. Hollerer. Wide-Area Scene Mapping for Mobile Visual Tracking. In *IEEE Int. Symp. On Mixed and Augmented Reality*, 2012.

[220] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg. Global Localization from Monocular SLAM on a Mobile Phone. *IEEE Transactions on Visualization and Computer Graphics*, 2014.

[221] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. TILDE: A temporally invariant learned detector. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.

[222] M. P. Vitus and C. J. Tomlin. Sensor placement for improved robotic navigation. In *Robotics: Science and Systems*, 2010.

[223] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard. Dynamic pose graph slam: Long-term mapping in low dynamic environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.

[224] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *IEEE Int. Conf. on Robotics and Automation*, pages 2043–2050, 2017.

[225] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10 (Feb):207–244, 2009.

[226] A. Wendel, A. Irschara, and H. Bischof. Natural Landmark–based Monocular Localization for MAVs. In *IEEE Int. Conf. on Robotics and Automation*, 2011.

[227] T. Weyand, I. Kostrikov, and J. Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conf. on Computer Vision*, 2016.

[228] S. B. Williams. Efficient solutions to autonomous mapping and navigation problems. 2001.

[229] S. A. Winder and M. Brown. Learning local image descriptors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[230] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *European Conf. on Computer Vision*, pages 467–483. Springer, 2016.

[231] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[232] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *Int. Conf. on Computer Vision*, 2015.

[233] W. Zhang and J. Košecká. Hierarchical building recognition. *Image and Vision Computing*, 25(5):704–716, 2007.

[234] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[235] G. Zhou, B. Bescos, M. Dymczyk, M. Pfeiffer, J. Neira, and R. Siegwart. Dynamic objects segmentation for visual localization in urban environments. *arXiv preprint arXiv:1807.02996*, 2018.

Curriculum Vitae

Marcin Tomasz Dymczyk
born August 10, 1989
citizen of Poland

2018–	*Sevensense AG, Switzerland* Co-Founder and Robotics Engineer
2015	*Google, Zürich, Switzerland* Research Visit
2015–	*ETH Zürich, Switzerland* Doctoral Studies at the Autonomous Systems Lab, supervised by Prof. Dr. Roland Siegwart
2012–2014	*ETH Zürich, Switzerland* Master of Science in Electrical Engineering (with distinction), Systems and Control major
2013	*The Boston Consulting Group, Warsaw, Poland* Visiting Associate
2012	*Softexor, Wrocław, Poland* Robotics Engineer, Computer Vision for Multicopters
2011	*WABCO Vehicle Control Systems, Wrocław, Poland* Computer Vision Intern
2008–2012	*Wrocław University of Science and Technology, Poland* Bachelor of Science in Electrical Engineering
2008–2011	*Wrocław University of Economics, Poland* Bachelor of Science in Finance and Accounting
2005–2008	*14th High School, Wrocław, Poland* High School Exam in Mathematics and Physics