University of Zurich[UZH]

Department of Informatics

Active Robot Vision: from State Estimation to Motion Planning

Dissertation submitted to the Faculty of Business,
Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor der Wissenschaften, Dr. sc.
(corresponds to Doctor of Science, PhD)

presented by
Zichao Zhang
from Anhui, China

approved in September 2020

at the request of
Prof. Dr. Davide Scaramuzza, advisor
Prof. Dr. Margarita Chli, examiner
Prof. Dr. Timothy Barfoot, examiner
Prof. Dr. Frank Dellaert, examiner
Prof. Dr. Michael Kaess, examiner

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, 16.09.2020

The Chairman of the Doctoral Board: Prof. Dr. Thomas Fritz

To my parents, friends and Yuanyuan.

Acknowledgements

First of all, I would like to thank my supervisor Prof. Davide Scaramuzza for accepting me as a PhD student in the lab. Davide's guidance and advice helped shape the research in this thesis, and I am grateful for the freedom and resources to pursue my own ideas.

This thesis would not have been possible without the help, fruitful discussions, and fun distractions from my colleagues. I therefore wish to express my gratitude to all the current and past members, visitors, and students. I would particularly like to thank Henri Rebecq, Davide Falanga, Titus Cieslewski, Christian Forster, Elias Mueggler, Matthias Faessler, Antonio Loquercio, Elia Kaufmann, Philipp Foehn, Daniel Gehrig, Mathias Gehrig, Manasi Muglikar, Yunlong Song, Guillermo Gallego, Jeff Delmerico, Suseong Kim, Manuel Werlberger, Reza Sabsevari, Dario Brescianini, Javier Hidalgo Carrio, Dimche Kostadinov, Christian Pfeiffer, Junjie Zhang, Giovanni Cioffi, Julien Kohler, Alessandro Simovic, Raphael Meyer, Thomas Längle, Manuel Sutter, Ruben Gomez Ojeda, Roberto Tazzari, Yuto Suebe, Sihao Sun, Cedric Scheerlinck, Timo Stoffregen, Kosta Derpanis, Francisco Javier Perez Grau, Yi Zhou, Bianca Sangiovanni, Ana Maqueda, Naveen Kuppuswamy, Stefano Ghidoni, Gabriele Costante, Antonio Toma and Tamar Tolcachier. I also had the pleasure to work with great students, namely Kunal Shrivastava, Juichung Kuo, Jonathan Huber, Guillem Torrente, Francois Elvinger and Patric Widmer.

I am very grateful to Prof. Torsten Sattler for hosting me at Chalmers University of Technology for a fruitful and enjoyable research stay.

I would like to thank the agencies funding my research, namely the National Centre of Competence in Research (NCCR) Robotics, the Swiss National Science Foundation, the DARPA FLA Program and the China Scholarship Council.

I would like to thank Prof. Margarita Chli, Prof. Timothy Barfoot, Prof. Frank Dellaert and Prof. Michael Kaess for accepting to review my thesis and for their valuable feedback. Last but not least, I am very grateful to my family, Yunayuan and my friends who supported me at all times.

Zurich, September 2020 Z. Z.

Abstract

Cameras are appealing choices for mobile robot perception, since they provide rich information for various tasks and are compact, low-cost and ubiquitous at the same time. The last decade witnesses rapid progress in camera-based localization and mapping for mobile robots. Visual odometry, or more general visual simultaneous localization and mapping, has reached the maturity that allows commercial applications. Combined with complementary sensors, such as inertial measurement units, visual SLAM becomes the enabling technology for various emerging applications, such as autonomous cars, virtual and augmented reality.

In the transition from research to real-world applications, the major challenge for vision-based algorithms is the robustness to difficult environments. In contrast to well-controlled lab environments, real-world scenarios pose various challenges for cameras, such as high dynamic range and visual ambiguity. Among different methods that can be adopted for better robustness, one particularly intriguing aspect is that the robot can control the data acquisition process actively from different levels. For example, the robot can control its sensor parameters to adapt to illumination change and also has the ability to choose its motion to avoid visually degraded areas.

The thesis mainly presents algorithms and studies about different aspects of active robot vision. At the sensor level, the thesis presents contributions in adaptive camera configuration, *i.e.* choosing the optimal sensor parameters for certain tasks and environments. The camera parameters that are considered include the camera field-of-views and the exposure time/gain setting. At the motion planning level, the thesis presents different methods of considering perception quality in motion planning to improve the robustness in visually degraded environments, from incorporating Fisher information in existing planners to designing a map representation for better performance. Finally, as a complementary research topic, part of the thesis is dedicated to develop rigorous evaluation and benchmarking methods for visual(-inertial) odometry and localization. The following is a list of the contributions of the thesis, in chronological order:

- An experimental study of the impact of different field-of-view cameras on visual odometry accuracy in typical application scenarios.
- An active camera control algorithm that is optimized for vision-based state estimation algorithms, which attempts to improve the robustness of several visual odometry algorithms in difficult high dynamic range environments.
- A perception-aware receding horizon planner for micro aerial vehicles that allows

the robot to reach a given destination, avoid obstacles and avoid visually degraded areas at the same time.

- The first dedicated map representation for perception-aware planning that is at least one order of magnitude faster than the standard practice of using point clouds.
- A study of existing trajectory evaluation metrics and an open-source trajectory evaluation toolbox for visual(-inertial) odometry.
- The first continuous-time, probabilistic trajectory evaluation framework that is able to take into consideration the inaccuracy in temporal association and groundtruth uncertainty.
- A novel method to verify and refine the reference poses in long-term visual localization datasets, which significantly improves the quality of the popular Aachen Day-Night dataset.

List of Contributions

Journal Publications

- **Zichao Zhang**, and Davide Scaramuzza. "Fisher Information Field: an Efficient and Differentiable Map for Perception-aware Planning". Under review in: *IEEE Transactions on Robotics* (2020).
  Links: Appendix D, Code, Video

- **Zichao Zhang**, Torsten Sattler, and Davide Scaramuzza. "Reference Pose Generation for Visual Localization via Learned Features and View Synthesis". Under review in: *International Journal of Computer Vision: Special Issue on Performance Evaluation in Computer Vision* (2020).
  Links: Appendix F, PDF

- **Zichao Zhang**, Guillermo Gallego, and Davide Scaramuzza. "On the Comparison of Gauge Freedom Handling in Optimization-based Visual-Inertial State Estimation". In *IEEE Robotics and Automation Letters* (2018). DOI: 10.1109/LRA.2018.2833152
  Links: PDF, Code

- Christian Forster, **Zichao Zhang**, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. "SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems". In *IEEE Transactions on Robotics* (2017). DOI: 10.1109/TRO.2016.2623335
  Links: PDF, Video, Software,

Book Chapter

- Davide Scaramuzza, and **Zichao Zhang**. "Visual-Inertial Odometry of Aerial Robots". In *Springer Encyclopedia of Robotics* (2020). DOI: 10.1007/978-3-642-41610-1
  Links: PDF

Peer-Reviewed Conference Papers

- Juichung Kuo, Manasi Muglikar, **Zichao Zhang**, and Davide Scaramuzza. "Redesigning SLAM for Arbitrary Multi-Camera Systems". In *IEEE International Conference on Robotics and Automation (ICRA)* (2020). DOI: 10.1109/ICRA40945.2020.9197553
  Links: PDF, Video

- Manasi Muglikar, **Zichao Zhang**, and Davide Scaramuzza. "Voxel Map for Visual SLAM". In *IEEE International Conference on Robotics and Automation (ICRA)* (2020). DOI: 10.1109/ICRA40945.2020.9197357
  Links: PDF, Video

- **Zichao Zhang**, and Davide Scaramuzza. "Beyond Point Cloud: Fisher Information Field for Active Visual Localization". In *IEEE International Conference on Robotics and*

Automation (ICRA) (2019). DOI: 10.1109/ICRA.2019.8793680
Links: Code, PDF, Video

- **Zichao Zhang**, and Davide Scaramuzza. "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry". In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018). DOI: 10.1109/IROS.2018.8593941
Links: Appendix E, PDF, Open-source toolbox

- **Zichao Zhang**, and Davide Scaramuzza. "Perception-aware Receding Horizon Navigation for MAVs". In *IEEE International Conference on Robotics and Automation (ICRA)* (2018). DOI: 10.1109/ICRA.2018.8461133
Links: Appendix C, PDF, Video

- Ruben Gomez-Ojeda, **Zichao Zhang**, Javier Gonzalez-Jimenez, and Davide Scaramuzza. "Learning-based Image Enhancement for Visual Odometry in Challenging HDR Environments". In *IEEE International Conference on Robotics and Automation (ICRA)* (2018). DOI: 10.1109/ICRA.2018.8462876
Links: PDF, Video

- **Zichao Zhang**, Christian Forster, and Davide Scaramuzza. "Active Exposure Control for Robust Visual Odometry in HDR Environments". In *IEEE International Conference on Robotics and Automation (ICRA)* (2017). DOI: 10.1109/ICRA.2017.7989449
Links: Appendix B, PDF, Video

- **Zichao Zhang**, Henri Rebecq, Christian Forster, and Davide Scaramuzza. "Benefit of Large Field-of-View Cameras for Visual Odometry". In *IEEE International Conference on Robotics and Automation (ICRA)* (2016). DOI: 10.1109/ICRA.2016.7487210
Links: Appendix A, PDF, Video, Code, Dataset

Workshop Paper

- **Zichao Zhang**, and Davide Scaramuzza. "Rethinking Trajectory Evaluation for SLAM: a Probabilistic, Continuous-Time Approach". In *Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR, ICRA* (2019).
Links: PDF

Open-source Code and Software

- Toolbox for Quantitative Trajectory Evaluation of VO/VIO
- Fisher Information Field for Efficient Perception-aware Planning
- Covariance Transformation for Visual-inertial Systems
- Open-source Interface for SVO 2.0 and Omnidirectional Camera Model

Award

- Best Paper Award at ICRA2019 Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR (Organizers: Sajad Saeedi, Bruno Bodin, Wenbin Li, Luigi Nardi)

Contents

1 Introduction

This thesis presents algorithms for vision-based estimation and navigation of mobile robots. Compared with pure computer vision problems, robot vision has the unique aspect of being active. Specifically, to better achieve a certain task, the robot has the capability to actively interfere with the data acquisition process from various aspects. The focus of the thesis is thus to explore different problems in active robot vision, including adaptive sensor configurations and perception-aware motion planning. Additionally, as a complementary topic, the thesis presents several contributions in the evaluation and benchmarking of vision algorithms.

This thesis is split into three parts. First, it focuses on the problem of adaptive sensor configurations, including: 1) how to choose from different Field-of-View (FoV) cameras for a given task/environment; 2) how to control the camera settings (*i.e.* exposure and gain) for better robustness in high dynamic range (HDR) environments. Second, it explores the incorporation of Fisher information in motion planning algorithms, including the application of Fisher information in a receding-horizon planner along with other planning objectives and a dedicated map representation for perception-aware planning. Third, it provides different methods to facilitate algorithm benchmarking and evaluation, including quantitative trajectory evaluation algorithms for visual(-inertial) odometry and reference pose generation for visual localization in challenging conditions.

This thesis is structured in the form of a collection of papers. An introductory section that highlights the concepts and ideas behind the thesis is followed by self-contained publications in the appendix. Since most of the presented work is tightly related to the context of visual odometry, Section 1.1 provides a brief overview about this topic and highlights the robotics-specific challenges. Then Section 1.2 specifies the research objectives that the thesis tries to achieve. Section 1.3 reviews the related research and puts the contributions of the thesis in context. New research work after the publication of the corresponding contribution is also included to provide an up-to-date perspective. Chapter 2 presents a summary of the papers in the appendix, and Chapter 3 suggests possible future directions.

1.1 Visual Odometry in Robotics

Visual odometry (VO) is the process of motion estimation from images alone, as defined in the seminal work [194]. It is first applied to the planetary exploration tasks [165] [170] and experiences rapid research progress since then. Nowadays, it has reached the maturity that allows commercial applications. Visual odometry and its variants, such as Visual-Inertial Odometry (VIO), are the enabling technology behind commercial drones, autonomous driving, Virtual Reality (VR) and Augmented Reality (AR).

Another tightly related terminology is visual Simultaneous Localization and Mapping (SLAM), which refers to the process of constructing a representation of the environment and inferring the sensor location at the same time. This thesis follows the definition in [242] that a SLAM system should have the ability to maintain a globally consistent map (*e.g.* via place recognition and loop closure). In practice, VO is usually part of a complete SLAM system and is mainly responsible for local motion estimation.

In the following, an overview of the main research findings in the field of visual odometry is presented. Though the main focus is on visual odometry, many of the research findings are applicable to VIO and visual(-inertial) SLAM as well, since these algorithms share similar underlying principles.

Additionally, for the history of visual odometry and SLAM, I refer the reader to [242, 93] and [35]. For visual-inertial odometry, [247, 114] provide an up-to-date overview.

1.1.1 An Overview of Visual Odometry

On a high level, the basic working principle of visual odometry is simple: given an initial map, the pose where an image is taken can be determined with respect to the map; using the estimated poses, the initial map can be extended to facilitate further motion estimation. Repeating this process gives continuous motion estimation from a video stream. Detailed methods, however, for the above process can be drastically different, and the existing literature is vast. Specifically, there are two fundamental design questions for any visual odometry pipeline:

- how should the data association be established on the image plane?
- how should the motion and structure be estimated in the 3D space?

Next, I will summarize the main research findings in this field from these two aspects.

Data Association and Measurement Model

Data association essentially establishes the correspondences of image coordinates for two or multiple images. Such correspondences provides constraints for motion and structure

estimation at a later stage.

Feature-based Methods Feature-based methods rely on repeatable feature detectors and invariant feature descriptors. Specifically, salient features are extracted from images, and feature descriptors are computed at the detected feature locations. Then matching features are found by comparing the descriptors and removing outliers using, for example, RANdom SAmple and Consensus (RANSAC).

Obviously, the types of features play an important role in feature-based methods. The most dominant feature type is point features, such as SIFT [159]. While SIFT descriptors is robust to illumination and view point change to a certain extent, it is relatively expensive to match. Therefore, in robotics, a common practice is to use more efficient binary features, such as [224] [228] [36] [150]. In addition to point features, lines [136] [101] [92] and planes are also being used in visual odometry and are shown to improve robustness in man-made environments.

Since the results from feature matching inevitably include outliers, RANSAC, first proposed in [86], is an essential step for feature-based methods. Due to its iterative nature, RANSAC is often the performance bottleneck in practice. Therefore, one research focus is to improve the RANSAC sampling efficiency. This can be done either by drawing better samples [57] [27] or reducing the dimension of the minimal models using specific knowledge about the platform [94] [241].

Direct methods Direct methods use the pixel intensities directly in the estimation process [119]. The basic idea is estimating desired parameters by minimizing the pixel intensity difference, *i.e.* the photometric error. This can be used to establish 2D correspondences [13] or estimating the motion and structure parameters directly [192] [132] [304].

Using direct methods to estimate motion and structure parameters requires calculating the intensity differences for all the pixels that are of interest. Using all the pixels in the image gives the best performance, but is often too expensive for robotic applications. Therefore a common practice is to apply direct methods only for the pixels with high gradients [81] [80] [91] [82].

Direct methods are able to utilize intensity information from areas with weak texture, even when few features can be extracted and matched. Therefore it is more robust to poorly textured environments compared with feature-based methods. The main disadvantages of direct methods is the small convergence basin and the sensitivity to illumination change. First, since image intensities are highly non-convex functions of the pixel coordinates, most direct methods rely on a good initial value for the underlying optimization to converge. Second, directly minimizing the photometric errors relies on

the constant brightness assumption across different images, which often does not hold in practice. This can be mitigated by explicit compensation [82] [21] or using alternative image representations [5]. Paper B in this thesis is also related to this topic.

Motion and Structure Estimation

Filter-based method Early research in this field focuses on using filters to estimate motion and 3D structure. The basic idea of using filters for visual odometry is to marginalize out all poses but the latest one and meanwhile keep 3D landmarks in the filter, since they will be still observed in future frames. During the marginalization, the information related to the marginalized parameters (i.e., past poses) is summarized into a probabilistic distribution of the parameters maintained in the filter. One common type of filters is Extended Kalman Filter (EKF), where the marginalization is performed around the linearization point of the process and measurement model. One representative visual odometry system using EKF is [66].

Using filters for visual odometry does not increase the state dimension when a new image arrives, and therefore limits the growth of computational complexity. However, there are several limitations. First, the complexity still depends on the number of landmarks maintained in the filter, which limits its application at large scale. Second, since the marginalization is performed around a linearization point, the accumulation of linearization error will result in inaccurate results and also possibly inconsistency of the estimator. Also see [268] for a detailed comparison between filtering and keyframe optimization methods.

Although most modern systems exploit keyframe optimization instead of filters for motion and structure estimation, filters are still used in visual odometry for specific tasks due to its memory efficiency and constant update time for a fixed state dimension. For example, in [299] [279] [91] [81], different types of filters are used to efficiently update the depth information as new images arrive. Moreover, in visual-inertial odometry, which shares similar principle as visual odometry, researchers proposed to also marginalize landmarks to bound the complexity [182], which in principle can also be applied to visual odometry.

Keyframe-based method Keyframe methods, as the predominant paradigm now, uses bundle adjustment (BA) [290] to optimize the motion and structure parameters. The key characteristic of keyframe methods is to sparsify the complete problem involving all parameters by selecting a subset of keyframes. Utilizing all frames will give the best accuracy, but is computationally prohibitive. Therefore, keyframe methods select, usually by heuristics, a subset of poses, namely *keyframes*, and landmarks and perform BA over the subset. The rest poses and landmarks are either discarded or marginalized. Both direct methods and feature-based methods can be used in this paradigm.

PTAM [137] proposed to run keyframe optimization in a separate thread from tracking, which is widely adopted in modern visual odometry systems. Their key observation is that the bundle adjustment does not have to happen at frame-rate for successful tracking. In the parallel structure, tracking operates at frame-rate, and BA continuously refines keyframes and landmarks at a lower rate, which both permits real-time performance and maintains high accuracy.

One common practice in keyframe methods is to discard non-keyframe poses as well as old poses and non-visible landmarks to limit the number of variables in the optimization. To retain the information of these variables for better accuracy, they can instead be marginalized, and the information can be summarized as a linearized error term in future optimization, often referred as *sliding window optimization*. However, care needs to be taken to maintain the sparsity (*e.g.* some measurements are selectively discarded) as well as consistency of the optimization problem. This approach has been demonstrated for both feature-based methods [259] and direct methods [82].

Representative Systems

In the last decade or so, there are many successful systems that influence the current paradigms in visual odometry and related fields. Next I will describe several representative systems and highlight their contributions.

MonoSLAM [66] is one of the pioneer works in real-time visual odometry. It uses an EKF for the estimation, where the estimates and the covariance of the 3D features and poses are maintained. A constant velocity model is used as the process model of the EKF, and image patches are used to search for correspondences in new images. One of the key features of the pipeline is that it uses the covariance information from the EKF to predict the search range for feature matching, which greatly reduces the computational burden. The system maintains ~ 100 features, aiming at applications in restricted space.

PTAM [137] is a keyframe-based visual odometry pipeline. The key feature of PTAM is using two parallel threads for tracking and mapping, which is de facto standard in modern visual odomety/SLAM applications. In the parallel structure, the tracking thread operates at frame rate, and the mapping is only performed when a new keyframe is inserted, which both avoids processing redundant information from nearby frames and allows to use expensive but accurate estimation methods.

DTAM [192] minimizes photometric error, instead of geometric error, to estimate motion and structure parameters. The system estimates a dense textured depth map at each keyframe. The depth map estimation is done by minimizing a cost function consisting of the sum of photometric error of overlapping frames and a regularization terms efficiently. Using the dense map, the motion of the camera is estimated by minimizing the photometric error of projecting the dense map into the image. Using a dense textured depth map,

DTAM has advantages such as increased robustness to motion blur and usefulness in real-world interaction, but requires more computational power.

ORB-SLAM [186, 187] is a keyframe-based visual SLAM system, using ORB features, for tracking, mapping, re-localization and loop detection. The visual odometry part (without loop detection and closing) follows a similar structure as PTAM. The main difference is that the guided search of ORB features, instead of image patches, is used to establish correspondences. The system has a well-designed map (keyframes and 3D points) management mechanism considering the requirements of both local tracking (e.g., insert keyframes often for better tracking) and global optimization (e.g., culling unnecessary points/keyframes for efficiency). The robustness of the system is further improved by the feature-based re-localization module.

LSD-SLAM [80] is a keyframe-based direct monocular SLAM system. Different from DTAM, every keyframe is associated with a *semi-dense* textured depth map, which only includes pixels with high gradient. The semi-dense depth map, along with per-pixel variance, is estimated using a probabilistic filter (originally proposed in [81]) from multiple small-baseline stereo pairs. The tracking of the current frame is done by minimizing the photometric error with respect to the last keyframe. The system further aligns different keyframes by a similarity transformation and performs pose graph optimization over all keyframes to maintain a globally consistent map.

DSO [82] is a direct visual odometry that maintains a sparse map. The system can be divided into a frontend and a backend. The frontend uses a similar image-to-image alignment as LSD-SLAM to track the camera motion and provides the backend with initial values for optimization. The backend exploits a sliding window optimizer to optimize over multiple keyframes and sparse points, minimizing the photometric error of projecting the sparse points into the keyframes. The system incorporates the photometric model of the camera to account for the exposure change, which greatly increases the robustness in terms of difficult illuminations.

SVO [91], [92] is a hybrid system combining the strengths of direct methods and feature-based methods. To track the camera motion, it first aligns the image with respect to the previous one to get an initial pose estimate, minimizing the photometric error over a sparse set of image patches. Then using the initial estimate, it searches for the correspondences of the image patches from overlapping keyframes along the epipolar line, and the pose estimate (and optionally the keyframes and points in the map) is further refined by minimizing the reprojection error. The system is extremely efficient and can achieve a high frame rate even on embedded platforms.

1.1.2 Robotics-specific Challenges and Research Problems

Visual odometry and SLAM sit at the intersection of computer vision and robotics. Following are several challenges and research problems that are of particular interest for robotics, which motivate the presented work in this thesis.

Robustness in harsh environments

As recognized by [35], one of the major challenges for SLAM currently is the robustness in difficult environments. This is especially true for visual odometry/SLAM, since standard cameras, which are not designed for motion estimation, have several limitations. These challenges include but are not limited to:

Difficult lighting conditions In contrast to well-lit lab environments, real-world scenarios usually have drastically different illuminations, which is a big problem for the limited dynamic range of the camera. For example, when the camera transitions between sunlit areas and shadows, the images are likely to get overexposed or underexposed. In addition, the illumination level of the same scene can vary, making it difficult for algorithms designed for constant illumination.

Fast motion In many real-world applications, the camera can undergo very aggressive motion. For instance, the user of a virtual reality headset can rotate his/her head very fast, and an autonomous drone can fly closely over a surface at a high speed. In these situations, the images will be severely blurred due to the fast motion.

Difficult texture Vision-based algorithms rely on the visual cues in images. If there are not sufficient texture in the environment, the algorithm can degrade quickly and fail. Unfortunately, many real-world scenes, such as the sky, texture-less walls and ground, have little information that can be used for visual odometry or SLAM.

Hardware and sensor configuration

The performance of vision algorithms highly depends on the data provided by the sensors. For robotics platforms, obtaining optimal data for certain tasks is challenging from both research and engineering perspectives.

Sensor Synchronization For better robustness or other necessary information such as scale, visual odometry and SLAM are usually performed over a set of heterogeneous sensors. For example, visual-inertial odometry is a popular choice for Micro Aerial Vehicles (MAV) due to its efficiency and low-cost. However, to optimally fuse the information from different sensors, accurate time synchronization is necessary, which is often a challenging engineering problem. Moreover, designing algorithms that work with imperfect time synchronization is also an active research topic.

Sensor Selection and Placement Ideally, the more sensors a platform is equipped with, the better performance we could possibly achieve. In practice, however, due to the limitation of power, computational resource and cost, only limited amount of sensors can be used. This brings the question of choosing the most appropriate sensor combination for a given task and environment. For example, which optics should be chosen for a given sensor? How to place a fixed number of cameras on a car for more accurate estimation?

Sensor Configuration Even for a fixed combination, a sensor itself has its own parameters that should be adapted for the tasks at hand. For cameras, such parameters include camera exposure time and gain. Ideally, the adaptation to specific environment should be done automatically without human intervention.

The coupling of perception and planning

Different from most computer vision algorithms, robot vision algorithms are tightly coupled with motion planning. In particular, different robot movement will result in different images and thus has an impact on the final performance. For example, an energy efficient motion could possibly point the camera to look at textureless regions, which will cause visual odometry to fail. Therefore, the limitation and requirement of the vision algorithms should be taken into consideration at the planning stage. This brings additional complexity to the overall robotic system design.

Task-depedent map representations

Most literature about visual odometry and SLAM focuses on the accuracy of the estimated motion and ignores the constructed map (partially due to the lack of the groundtruth). For most robotic applications, the robot needs to be aware of the surroundings in addition to its own location, and thus the map is of equal importance. Obviously, different tasks have different requirements, and the sparse point cloud (with associated frames) from a typical visual odometry pipeline is not necessarily the optimal map representation. Therefore, how to create a suitable map representation for a given task from the output of visual odometry and SLAM is an important research topic in robotics (*e.g.* [198] for collision-free trajectory optimization, [184] for visible landmark query and Paper D in this thesis for perception-aware motion planning).

1.2 Research Objectives

Despite the rapid advance in the field of mobile robots, deploying robots in a real-world scenario is still challenging, due to the open problems in various aspects of robotics. From the perspective of robot vision, tackling the challenges discussed in Section 1.1.2 will both bring robots one step closer to real-world applications and improve our understanding of existing algorithms. Therefore, this thesis focuses on part of those challenges: adaptive sensor configuration and the coupling between perception and planning. Moreover, for better evaluation of the existing methods, part of the thesis is dedicated to evaluation and benchmarking techniques. Next, specific research goals are discussed.

1.2.1 Adaptive Camera Configuration

In this thesis, we limit the scope of sensor configuration to a single camera. Moreover, out of the many properties of a camera (*e.g.* resolution, frame-rate, global shutter/rolling-shutter), this part of the thesis focuses on two aspects: optics and exposure/gain settings.

Choosing Optics

It is well known that increasing the FoV helps improve the robustness of VO, especially for fast motion: the apparent motion is smaller, therefore features are easier to track. However, it is not clear what is the influence of large FoV cameras on the state estimation accuracy. In fact, if the image sensor is of the same resolution, a large FoV will lead to a coarser angular resolution, which potentially has a negative impact on the accuracy. This raises a practical question: given a sensor, what is the optimal optics? Intuitively, the answer should be environment dependent. Due to the difficulty of accurately modeling a realistic 3D world analytically, we resort to experimental study in typical environments. Experiments regarding both separate VO building blocks and the complete pipeline are performed. Both synthetic and real-world datasets are used in the experiments.

Choosing Camera Exposure/Gain

Adaptively changing the exposure time and gain can to an extent mitigate the limited dynamic range of standard cameras and improve the robustness of vision algorithms in HDR environments. Specifically, different from previous work, we would like to optimize the camera exposure and gain for visual odometry. To this end, the goal of this work is twofold. First, we need to establish an image quality metric that is tightly related to vision algorithms. This metric will then be the objective for the exposure and gain control algorithms. Second, we need to find an effective method to optimize the metric mentioned above. Finally, the overall algorithm needs to be implemented and tested on actual hardware.

1.2.2 Perception-aware Planning

To consider the perception in motion planning, the first question is how to quantify the perception quality. While there are many heuristics (*e.g.* number of tracked features), in this thesis, we use the Fisher Information Matrix (FIM) as a central tool. This is due to the fact that FIM is a pivotal concept in estimation problems in general and has rich theoretical connections. It also has already found many related applications in robotics (*e.g.* [42]).

Planning with Fisher Information Matrix

The planning problem, without considering perception quality, already has many objectives. Most notably, a planned motion has to be collision free and respect the dynamics of the platform. Moreover, it is also common to minimizes the energy cost of the planned motion. Therefore, our first goal is to incorporate FIM in existing/standard planning algorithms and study the behavior. We choose to use a quadrotor as the platform due to the expertise in our lab and a receding horizon planner for simplicity.

Designing a Map Representation

Essentially, motion planning aims to find feasible and/or efficient motion, given certain information/constraints regarding the surrounding environment. Therefore, the map representation is a core concept in planning (as well as in SLAM) and has a tremendous impact on the performance of motion planning algorithms. For example, using Euclidean Signed Distance Field (ESDF) [198] proves to be more efficient than using point clouds and allows optimization-based motion planning algorithms. Unfortunately, little work has been done in this aspect for perception-aware planning. This part of the thesis aims to fill this gap: developing an efficient map representation for perception-aware planning. We further would like the map representation to be generally applicable to common motion planning algorithms.

1.2.3 Algorithm Benchmarking and Evaluation

Although it is not directly related to the main topic of the thesis, we find rigorous evaluation framework to be a fundamental requirement for the above research. Unfortunately, compared with the amount of literature in visual odometry and SLAM, there is relatively little study about the evaluation. In fact, during the research work of the PhD, we found several limitations of existing evaluation methods and datasets. Therefore, we also perform several studies regarding the limitations we identified. Specifically, this part of the thesis includes a systematic overview of existing trajectory evaluation methods in visual(-inertial) odometry and a novel method to improve existing visual localization datasets for long-term visual localization.

1.3 Related Work

Large FoV Visual Odometry and Optimal Optics

Due to its superior robustness, there is a continuous research interest in using large FoV cameras for visual odometry and SLAM. Most VO algorithms for omnidirectional cameras [218, 246, 278, 242] rely on robust feature descriptors (*e.g.* SIFT [159]) to establish feature correspondence. To cope with the significant distortion of large FoV images, special descriptors were developed that model the distortion effects to improve feature matching [105, 9, 207, 158]. Direct methods are also used. [61] and [243] used Lucas-Kanade feature tracking [13] to estimate the motion of landmark observations between frames of omnidirectional images. Moreover, most of the recent systems mentioned in Section 1.1.1 have evolved to support large FoV cameras, such as [44, 168] and our work [318]. More recently, deep-learning based depth estimation on large FoV stereo images shows promising results and is applied to visual odometry and SLAM as well [306, 307].

In contrast, the study of the impact of different FoVs is relatively rare. A comparison of the performance of a catadioptric and a perspective camera in a visual SLAM system was presented in [217]. But the catadioptric camera that was used for the experiments had a higher pixel resolution than the perspective camera, which does not allow for a direct comparison. On the other hand, the comparison presented in [270] experimentally confirmed that a larger FoV camera has a higher motion estimation accuracy than a smaller FoV perspective camera, even in the case of a fixed pixel resolution. Unfortunately, the experiments were limited to synthetic data and an indoor environment. In contrast, our study is performed on both synthetic and real-world datasets in different environments.

Automatic Camera Control

Most existing auto-exposure/gain algorithms are designed for image photography using heuristic metrics. A system for configuring the camera parameters was presented in [190] The exposure time was selected according to the intensity histogram of the image. [117] modeled the pixel intensity distribution empirically and directly calculated the exposure time that minimizes the rendering error. [289] used a set of indicators from the intensity histogram and the cumulative histogram to characterize the image quality and desgined a camera exposure control method based on these indicators.

By contrast, less work has been done to optimize the camera settings for vision-based state estimation. [256] used the gradient information within an image to select the proper exposure time. The authors defined an information metric based on the gradient magnitude at each pixel. The exposure change was simulated by applying different gamma corrections to the original image to find the gamma value that maximizes the gradient information. Then, the exposure time was adjusted based on the gamma value. In comparison, our work [316] utilizes the photometric response function of the camera to

predict intensity and gradient change, which in principle models the images at different exposure times more accurately.

Since our work [316] was published, there are several research works exploiting the same direction, aiming to maximize the gradient information in the image [133, 255, 171] for different tasks. Related to the research of automatic camera control, deep learning techniques are also used for image enhancement or learning invariant representations for better robustness in HDR environments [102, 60, 49, 124].

Perception-aware Planning

Considering perception performance in planning has been extensively studied in different contexts. Early works include maximizing the Fisher information about the robot state and the map in navigation tasks [85, 166], minimizing the entropy of the robot state in known environments [33, 226], and actively searching features in SLAM systems [65]. Recently, with the advance of drones, several works have been done to couple perception, planning and control on agile aerial platforms [1, 229, 181, 202, 323, 303, 83].

Despite the extreme diversity of the research in this topic, related work can be categorized based on the method to generate motion profiles. One paradigm used sampling-based methods, which discretize the space of possible motions and find the optimal one in a discrete set. [226] used Dijkstra's algorithm to find the path on a grid that minimizes a combined cost of collision and localization. [202] and [62] adapted the rapidly-exploring random tree (RRT) algorithms to incorporate the perception cost, and the latter additionally considered the photometric property of the environment. Instead of a combined cost, as in most of previous works, [116] used multi-objective search for perception-aware planning. Alternatively, researchers have explored to plan in the continuous motion space. [118] considered optimizing the motion within a finite horizon to minimize a joint cost including the final pose covariance, which was later extended to visual-inertial sensing and self-calibration in [79]. [303] studied the general problem of trajectory optimization on manifolds and applied their method to planning under the FoV constraint of the camera. [83] tackled the problem at the controller level by incorporating related costs in model predictive control (MPC). Our work [323] falls in the first category and proposes to evaluate motion primitives against multiple costs, including the localization uncertainty, in a receding horizon fashion.

In the above methods, calculating the perception related cost/metric is a crucial part and often the computational bottleneck (*e.g.* [166]). Unfortunately, little work has been done in developing dedicated representations for efficient computation. [226] pre-computed and stored the information in a 2D grid, but their method was limited to 360° FoV sensors. [116] trained a neural network to predict the state estimation error and generated a map of perception cost using the network prediction. However, their map only contains the averaged cost of different orientations and, therefore, cannot be used to evaluate the

cost of an arbitrary 6 Degrees-of-Freedom (DoF) pose. In contrast, our method [321] explicitly models the FoV constraint and can represent the information of 6 DoF poses efficiently. More recently, after our paper [321] was published, [95] used a similar idea as ours to avoid iterating over all the landmarks in the environment in observability-aware trajectory optimization.

Algorithm Benchmarking

For trajectory evaluation of VO/VIO, most existing quantitative trajectory evaluation approaches were introduced together with a specific algorithm or a dataset. [271] provided a benchmark for RBG-D simultaneous localization and mapping (SLAM) systems, and proposed to use both the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE). ATE is also widely used to evaluate visual odometry/SLAM algorithms, for example, in [82, 186, 92]. Compared with ATE, relative error, as analyzed in [32] and [142], is less sensitive to the specific time the estimation error occurs. [98] further extended the relative error as a function of sub-trajectory length and velocity to provide more informative results. Despite the rich literature in this field, there is very little work dedicated to the exact problem of quantitative trajectory evaluation for VO/VIO, which leaves many open issues. It is not clear, for example, to what extent the current approaches are applicable: is the method for one sensing modality also suitable for another (*e.g.* can the same evaluation method be used for both VO and VIO)? Our work [320] provides a unified perspective of the current available trajectory accuracy metrics, demonstrating how the evaluation method should be chosen based on sensing modalities. Moreover, our work provides an easy-to-use toolbox for related research.

One common approach to obtain reference poses for visual localization datasets is to use Structure-from-Motion (SfM) on a large set of images. Query images are then obtained by removing some images and 3D points from the SfM model, *i.e.* a smaller model to test visual localization algorithms. This method is used in, for example, [154, 153, 238]. For long-term localization datasets, where the query images are typically taken under different conditions w.r.t. the reference model, the SfM method tends to fail. In this case, additional sensors, such as multi-camera systems [14, 11, 237] and Lidar [164, 237] could be used to get sufficiently good pose estimates. If only images are available, manually annotated correspondences can be used [239, 237], which is not scalable and potentially inaccurate. Our method [319] makes use of recently developed learning-based local features and view synthesis and is able to semi-automatically generate accurate reference poses using only images without manual labelling. Moreover, it can be used as a tool to verify the accuracy of existing reference poses.

1.4 Summary

In this section, I first discussed the field of visual odometry to provide a general context for the thesis. Moreover, I identified several robotics-specific problems and challenges. Motivated by these challenges, I outlined the research objectives of this thesis. Lastly, I discussed the related work to show how this thesis relates to or builds upon related research.

2 Contributions

This chapter summarizes the key contributions of the papers that are reprinted in the appendix. It further highlights the connections between the individual results and refers to related video and open-source code contributions. In total, this research has been published in five peer-reviewed conference papers. One journal paper is currently under review at the *International Journal of Computer Vision (IJCV)*. One further journal paper D that extends our previous work [321] is under review in the *IEEE Transaction on Robotics (TRO)*. These works led to several open-source software and a best paper award at an international workshop.

2.1 Adaptive Camera Configuration

Deploying vision-based state estimation algorithms (*i.e.* visual/inertial odometry and SLAM) on real hardware requires specific knowledge about the sensors. Different sensor configurations often have a large impact on the actual performance. In this part of my thesis, we explored the effect of camera optics and exposure/gain settings. First, we studied the trade-off between field-of-view and angular resolution for different tasks and concluded empirically how the optimal optics should be chosen. Second, we designed a novel camera control algorithm that uses the photometric model of the camera to maximize the gradient information in the image, which proved to improve the robustness of several visual SLAM algorithms in HDR environments.

2.1.1 Paper A: Large FoV Cameras for Visual Odometry

(P1) Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. "Benefit of Large Field-of-View Cameras for Visual Odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2016. DOI: 10.1109/ICRA.2016.7487210

It is well-known that using large FoV cameras improve the robustness of visual odometry algorithms. However, for a fixed sensor (*i.e.* fixed resolution), using a wide angle lens sacrifices the angular resolution, which potentially decreases the estimation accuracy. In this paper, we studied the problem of choosing the optimal optics for specific tasks and

environments. We first performed simulation and analysis regarding different components of visual odometry. We then adapted a state-of-the-art visual odometry algorithm SVO [92] to work with large FoV cameras and performed extensive experiments on both synthetic and real-world datasets. Empirically, we concluded that a large FoV lens is beneficial in a confined environments, whereas in an open environment, a standard pinhole camera results in better estimation accuracy. The implementation of large FoV camera model is available open source, and the adaptation of SVO is available as binaries. We also released the datasets used in this work to the public. This work was included as part of a journal publication (Paper R1).

Related Publication

(R1) C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Trans. Robot.* 33.2 (2017), pp. 249–265. DOI: 10.1109/TRO.2016.2623335

Related Software

(S1) C++ Omnidirectional camera model

(S2) SVO 2.0 binaries

Related Dataset

(D1) The Multi-FOV dataset

Related Video

(V1) https://youtu.be/6KXBoprGaR0

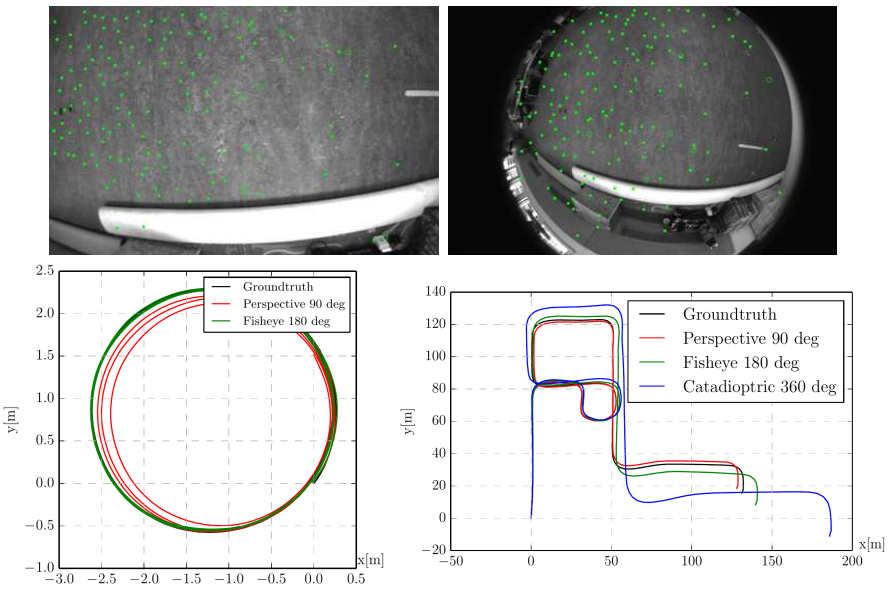


Figure 2.1 – **Top**: tracked features using pinhole and fisheye cameras; **Bottom**: Estimated trajectories in a confined room and an urban canyon respectively. Note that large FoV cameras perform better in the confined room and worse in the urban canyon.

2.1.2 Paper B: Active Camera Control for Robust Visual Odometry

One of the major limitations of standard cameras is the limited dynamic range, which often causes under-exposed or over-exposed images in practice. Instead of a fixed exposure time and gain setting, the camera built-in auto-exposure/gain algorithm can to an extent adapt to different lighting conditions but is not designed for vision-based state estimation algorithms. Motivated by the fact that many vision algorithms utilize the image gradient, we proposed to maximize the gradient information in the captured images. To this end, we first designed a robust image quality metric, which is based on the percentiles of the pixel gradients. We then designed a camera control algorithm to maximize this metric using the information from the camera's photometric response function. Experimental results show that our camera control method improves the performance of different visual odometry algorithms in HDR environments.

(P2) Z. Zhang, C. Forster, and D. Scaramuzza. "Active exposure control for robust visual odometry in HDR environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017, pp. 3894–3901. DOI: 10.1109/ICRA.2017.7989449

Related Video

(V2) https://youtu.be/TKJ8vknIXbM

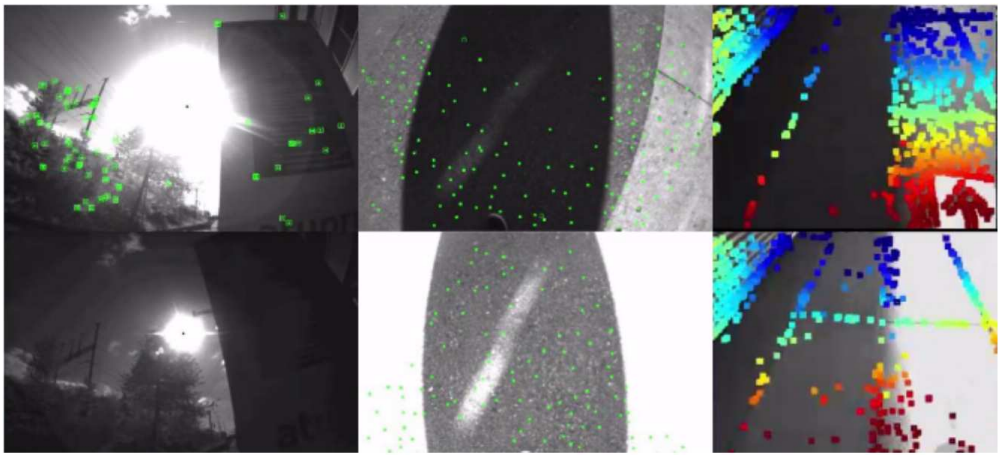


Figure 2.2 – Comparison of our exposure control method with the camera built-in auto-exposure for different VO algorithms (left to right: ORB-SLAM2, SVO 2.0, DSO). The first row shows the tracking results of our method and the second row the built-in auto-exposure. It can be seen that with our exposure control method, more features can be tracked.

2.2 Perception-aware Motion Planning

Apart from adapting the sensor configurations, another unique aspect of robot vision is the possibility to actively plan the sensor motion for better perception quality. In this part, we explored different methods to incorporate Fisher Information, a pivotal concept in SLAM/estimation, in motion planning. We first demonstrated the incorporation of FIM in a receding horizon planner for MAVs, along with other possibly conflicting planning objectives. To overcome the limitation observed in the previous work, we further designed the first dedicated map representation for 6 DoF perception-aware planning.

2.2.1 Paper C: Incorporating Fisher Information in Visual Navigation

(P3) Z. Zhang and D. Scaramuzza. "Perception-aware Receding Horizon Navigation for MAVs". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 2534–2541. DOI: 10.1109/ICRA.2018.8461133

In this paper, we proposed a vision-based navigation algorithm for MAVs. The algorithm steers the MAV to fly to a given destination, avoid obstacles and prefer texture-rich areas at the same time. We first generated a library of candidate trajectories in a fixed horizon in front of the MAV. Then we evaluated sampled poses from them in terms of perception quality, collision probability, and distance to the goal and chose the best one to execute. Specifically, the perception quality is derived from the FIM of localizing w.r.t. the current map (*i.e.* 3D landmarks) from the SLAM algorithm online. We showed that our method improves the navigation accuracy and success rate in both environments with texture-less regions and the ones without. An interesting observation is that obstacles act as both repellers, due to the collision risk, and attractors, due to the visual features on them.

Related Video

(V3) https://youtu.be/FK6S_CRXiuI

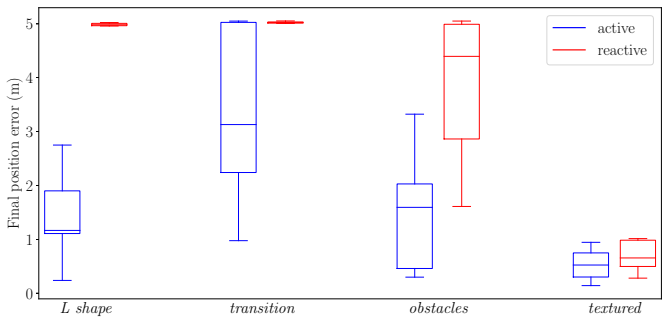


Figure 2.3 – Navigation accuracy in different environments, where *L shape*, *transition* and *obstacles* are environments with "perception traps" (textureless regions) and *textured* is an environment with rich texture everywhere.

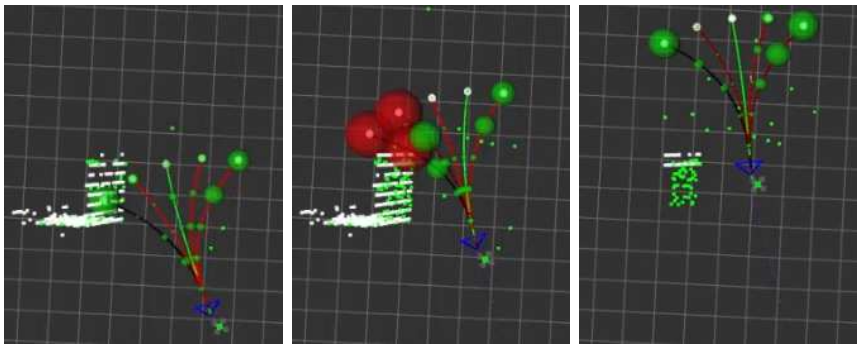


Figure 2.4 – An example of the double role of obstacles. White points denote obstacles, green points the landmarks from VIO, curves the candidate trajectories to choose from, and the spheres are the sampled poses from the candidate trajectories. The time evolves from left to right. With our algorithm, the MAV first tried to get closer to the obstacle due to the visual features on it and then moved away to reduce collision risk.

2.2.2 Paper D: Fisher Information Field for Perception-aware Planning

(P4) Z. Zhang and D. Scaramuzza. "Beyond Point Clouds: Fisher Information Field for Active Visual Localization". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2019, pp. 5986–5992. DOI: 10.1109/ICRA.2019.8793680

(P5) Z. Zhang and D. Scaramuzza. "Fisher Information Field: an Efficient and Differentiable Map for Perception-aware Planning". In: *Under review in IEEE Trans. Robot.* (2020). URL: https://arxiv.org/abs/2008.03324

As seen in the previous work, FIM is an effective tool for perception-aware planning. Since most SLAM/localization algorithms use point clouds as the map representation, calculating FIM directly from the point clouds seems a convenient choice. This process, however, requires iterating over all the 3D points, and thus quickly becomes inefficient as the number of landmarks increases. To overcome this drawback, we proposed a dedicated map representation for 6 DoF perception-aware planning.

The key idea is to pre-compute the rotation-independent component from FIM and store it in a voxel grid. This is achieved by formulating the FIM in a specific manner and approximating the non-trivial camera visibility function with different simplified models. Then at the planning stage, the full (approximated) FIM at arbitrary 6 DoF poses can be recovered from the precomputed components in *constant time*, regardless of the number of existing landmarks. The constant query time is also due to the use of the voxel hashing technique [193], inspired by [198].

In the conference version [321], we used a quadratic function for visibility approximation. We showed that computing FIM from our map representation is at least an order of magnitude faster than using point clouds directly in several scenarios, at the cost of additional memory and construction time beforehand. However, due to the limited

expressive power of the quadratic function, the approximation error (with respect the exact FIM calculated from the point clouds) is non-negligible in certain cases.

We further have extended the previous work in several aspects in [322]. Specifically, we design a non-parametric visibility approximation that is more accurate and scalable than the quadratic model. We further demonstrate the Fisher information field in different active settings - with both sampling and optimization based motion planning algorithms. We show that the proposed map representation is advantageous in terms of efficiency and being differentiable, which is important for optimization-based motion planning.

Related Video

(V4) https://youtu.be/q3YqIyaFUVE, http://rpg.ifi.uzh.ch/fif.html

Related Software

(S3) https://github.com/uzh-rpg/rpg_information_field

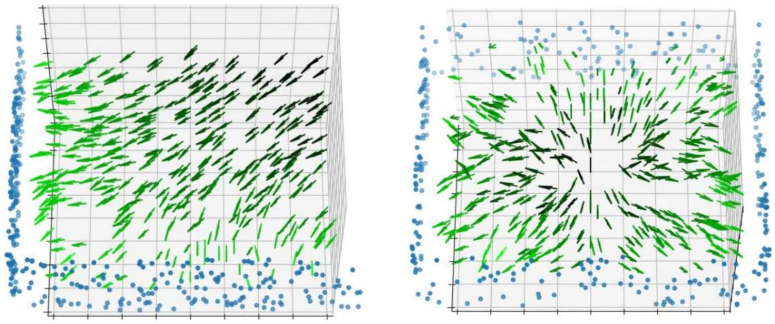


Figure 2.5 – Visualization of the Fisher information field in simulation. Blue circles are 3D landmarks. Each arrow stands for one optimal view direction, determined from the map representation, at the corresponding position. Brighter color means better localization quality.

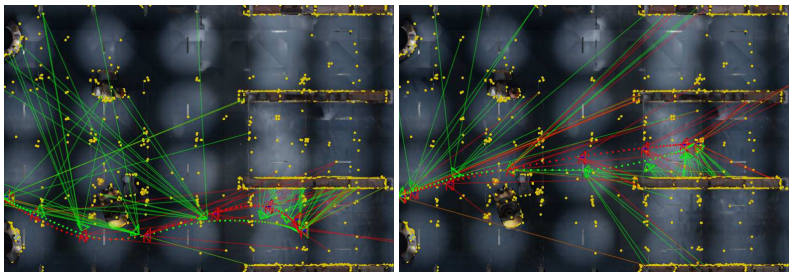


Figure 2.6 – The comparison of the optimized trajectories using the proposed Fisher Information Field (green) and without considering the Fisher information (red). The poses sampled at a constant time interval are visualized as points of the corresponding color. The yellow points are the landmarks for localization, and the lines denote the potentially matchable landmarks considered in the trajectory optimization. Intuitively, including the Fisher information in the trajectory optimization force the camera to orient towards and move closer to areas with more landmarks (*e.g.* the shelves).

2.3 Algorithm Benchmarking and Evaluation

During the previous research, evaluating vision-based estimation algorithms has been a recurring topic, where both rigorous evaluation methods and high quality datasets are required. For this part of the thesis, we first summarized the previous trajectory evaluation methods in a unified perspective and provided an open source toolbox. We also investigated the quality of existing visual localization datasets and proposed a method for verifying, refining and generating reference poses for long-term visual localization.

2.3.1 Paper E: Quantitative Trajectory Evaluation for VO/VIO

(P6) Z. Zhang and D. Scaramuzza. "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018. DOI: 10.1109/IROS.2018.8593941

In this tutorial, we provided a systematic overview about existing trajectory metrics: Absolute Trajectory Error (ATE) and the Relative Error (RE). Their motivation, advantages and disadvantages were discussed. Moreover, we hightlighted that the trajectory alignment step, which is to eliminate the ambiguity in the estimator output, should be done by considering the unobservable DoFs in the estimator. For example, to evaluate a trajectory from VIO, a 4 DoF alignment (position plus the rotation around gravity) should be used instead of a rigid body transformation, since the gravity direction is observable for a visual-inertial system. We further released an open source toolbox to facilitate the evaluation in related research.

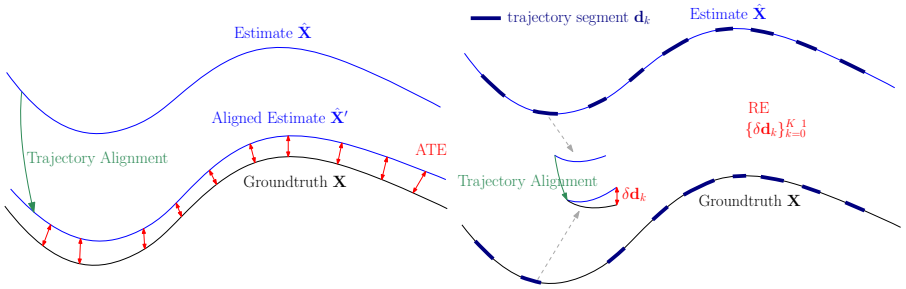


Figure 2.7 – Commonly used error metrics in trajectory evaluation: absolute trajectory error and relative error.

A follow-up of this work tried to formulate the trajectory evaluation problem in a continuous-time and probabilistic framework using Gaussian Process. We showed some promising properties of our method, such as the ability to consider the time offset and inaccuracy in groundtruth and the theoretical connection between existing trajectory error metrics, both of which are the first in literature. This work was presented at the *ICRA19 workshop on Dataset Generation and Benchmarking* (Paper R2) and awarded the best paper award.

Related Software

(S4) Trajectory evaluation toolbox

Related Publication

(R2) Z. Zhang and D. Scaramuzza. "Rethinking Trajectory Evaluation for SLAM: a Probabilistic, Continuous-Time Approach". In: *ICRA19 Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for VR/AR*. 2019

2.3.2 Paper F: Reference Pose Generation for Visual Localization

(P7) Z. Zhang, T. Sattler, and D. Scaramuzza. "Reference Pose Generation for Visual Localization via Learned Features and View Synthesis". In: *Under review in Int. J. Comput. Vis.* (2020). URL: https://arxiv.org/abs/2005.05179

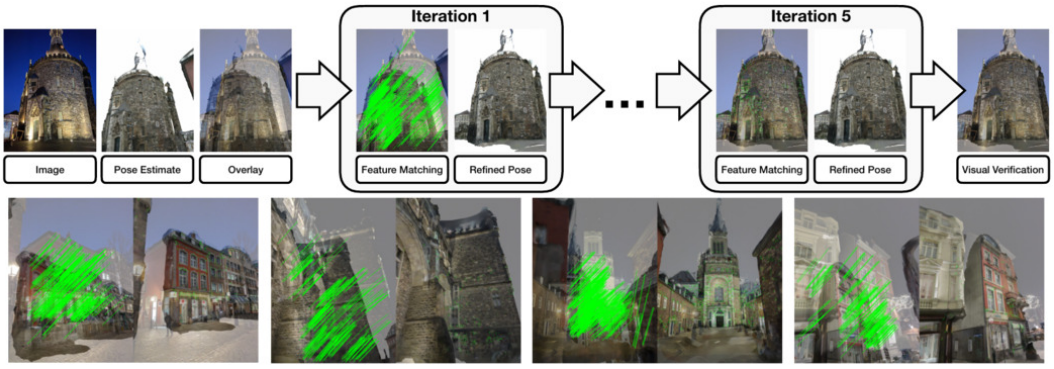


Figure 2.8 – Top: Overview of the proposed method for reference pose generation. We overlay the synthesized and the actual images, and the green lines stand for the feature matches between the two images. Better overlay and closer feature locations (*i.e.* shorter green lines) indicate better pose accuracy. **Bottom**: Several examples where the night time reference poses in the Aachen Day Night dataset show large error (left), and the refined poses (right) are more accurate.

Different from SLAM, visual localization is the problem of computing a 6 DoF pose from a single image, given a known scene model. Therefore, high quality datasets containing query images and the corresponding 6 DoF reference poses are the foundation for benchmarking and improving existing visual localization methods. Unfortunately, relatively little work has been done in assessing and improving the reference pose quality compared with the literature in visual localization algorithms. This work proposed a semi-automated approach to verify and refine (potentially) inaccurate reference poses in visual localization dataset. The method is motivated by the recent advance of learned features, which show better robustness to different conditions than conventional features (*e.g.* between real images and the renderings from an imperfect 3D model).

Specifically, given an reference pose, our approach matches the renderings of a 3D model and real images via learned features. The feature matches indicate the accuracy of

the reference pose and can be used to calculate a more accurate pose if needed. We significantly improved the nighttime reference poses of the popular Aachen Day-Night dataset, showing that state-of-the-art visual localization methods perform better (up to 47%) than predicted by the original reference poses. Moreover, we generated reference poses for new nighttime images using the same method, effectively doubling the size of the original dataset. The code and the extended dataset will be released upon publication.

2.4 Unrelated Contributions

During the PhD, five papers were co-authored that are not part of this thesis. In general, these papers studied the topic of visual(-inertial) SLAM as well, and some papers are, to an extent, inspired by the research problems explored in the previous work. For example, the idea of Paper U4 originated from investigating the FIM in visual-inertial optimization, which was the first step for my research in perception-aware planning. The voxel hashing technique used in the Fisher information field (Section 2.2.2) is one of the core components in both Paper U1 and U2. FIM is also applied to keyframe selection in Paper U1, which simplifies the standard design paradigm (*i.e.* using heuristics) and generalizes to various camera configurations automatically.

(U1) J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza. "Redesigning SLAM for Arbitrary Multi-Camera Systems". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020

(U2) M. Muglikar, Z. Zhang, and D. Scaramuzza. "Voxel Map for Visual SLAM". in: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020

(U3) D. Scaramuzza and Z. Zhang. "Aerial Robots, Visual-Inertial Odometry of". In: *Encyclopedia of Robotics*. Springer Berlin Heidelberg, 2020, pp. 1–9. DOI: 10.1007/978-3-642-41610-1_71-1

(U4) Z. Zhang, G. Gallego, and D. Scaramuzza. "On the Comparison of Gauge Freedom Handling in Optimization-Based Visual-Inertial State Estimation". In: *IEEE Robot. Autom. Lett.* 3.3 (July 2018), pp. 2710–2717. DOI: 10.1109/lra.2018.2833152

(U5) R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza. "Learning-Based Image Enhancement for Visual Odometry in Challenging HDR Environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 805–811. DOI: 10.1109/ICRA.2018.8462876

3 Future Directions

While the fundamental principles of visual odometry and SLAM are in general well understood, developing and deploying such systems in real-world environments is still challenging. On the one hand, it requires a tremendous amount of engineering effort to tackle various corner cases. On the other hand, I believe that it also indicates the current design paradigms and techniques used in visual SLAM systems can be further improved, which opens up several interesting research directions.

Moreover, taking into consideration the active nature of mobile robots adds considerable complexity to the overall design of perception and planning systems. The coupling of the perception and planning, however, has the potential to greatly improve the adaptivity of mobile robots. In contrast to SLAM, the problem of active SLAM, or more general active perception, is far from solved itself, and many fundamental research questions are still open.

Next, I will describe several interesting research directions regarding both visual SLAM in general and possible continuation of the work on active robot vision presented in this thesis.

Less Engineering A typical SLAM system nowadays involves many components that are designed heuristically, such as keyframe selection and map management. While there are several common practices that work well in general, these heuristics bring additional complexity and problems for applying SLAM algorithms. First, these heuristics are often designed with specific sensor configuration or environment in mind and do not necessarily work for other situations. Second, these heuristics usually involve tunable parameters (or magic numbers) that have a significant impact on the actual performance, and thus tuning is often necessary. Third, it is difficult to judge whether a set of parameters is optimal. Instead, it would be of both research and practical interest to replace the heuristically designed modules with methods that are theoretically grounded. Several examples in this direction are feature selection [42], information sparsification [113, 293] and our work on keyframe selection and map management [143, 184]. Efforts in this direction could both simplify the current design paradigms and improve performance.

Incorporating Learned Components With the advent of deep learning, it is a natural question to ask how learning methods could help build better visual SLAM systems. It is a very active research field currently, and I think there are several promising directions.

For visual odometry and SLAM, one of the most important tasks is data association. Recently learned features [74, 215] show better robustness to condition changes, and their advantages in 6 DoF pose estimation are demonstrated in the task of long-term visual localization. It would be interesting to see whether these features could be used to improve the robustness in HDR environments or in the presence of motion blur.

Another direction is to use learning-based method to provide priors for SLAM systems. This could be useful in case where standard methods fails, such as the scale ambiguity for monocular systems. There are already several works in this direction [280, 310, 104, 22, 326]. Moreover, with the recent advance in neural rendering and scene representation [262, 263, 177], it is also an interesting direction to explore the possibilities and advantages of using them as alternative map representations for SLAM.

Understanding the Sensor Limitation Although the camera control method proposed in Paper B improves the robustness of several visual odometry algorithms, it is limited by the inherent properties of the cameras. Specifically, there are only two parameters (exposure and gain) that can be used to control all the pixels in the image, and thus the ability to deal with HDR environments is inherently limited. For example, in extreme cases where both bright and dark areas exist in the FoV, there could be simply no camera setting that can achieve good image quality. In comparison, event cameras, where each pixel is triggered individually, can naturally overcome this limit. With this comparison in mind, an interesting research question is that: can we establish a model that connects the different sensor properties (*e.g.* triggering mode, pixel clock frequency) to its sensing ability (*e.g.* in HDR environment)? Such model would be useful to better understand existing sensors and design new sensors and can be potentially applied to recently developed programmable sensors [51].

Rich Map Representation Motion planning in realistic environments often needs to consider multiple objectives, which requires various information about the environment. The inclusion of perception quality brings additional requirements. For example, whether a feature is visible or matchable from a certain viewpoint is useful to quantify the estimation uncertainty accurately. Another useful information is whether the visual cues belong to persistent or movable objects and should be used for localization. Such information, however, requires a combination of semantic, geometric, and texture information of the environment, which nowadays no known map representation can provide. The lack of such information is also one major limitation of the proposed Fisher information

field. Therefore, an interesting direction is to develop a rich map representation for perception-aware planning. Such map could be also useful for motion planning and SLAM in general, since the information required by these problems share many similar properties.

Beyond Fisher Information In the work on perception-aware planning in this thesis, Fisher information is used to quantify the perception/estimation quality. However, since most estimation problems in practice are nonlinear, FIM is, at its best, the first order approximation of the inverse of the estimator variance. Although it is useful to reflect the quality of the estimate, FIM usually does not correspond to the actual uncertainty and does not capture the information of estimator bias. Unfortunately, properly characterizing the bias and variance of a nonlinear estimator is difficult (*e.g.* [84]). Monte-Carlo simulation could be used instead, but may be too inefficient to be executed online. Therefore, theoretical tools that can better capture the estimator properties or efficient online simulation would be very useful.

Assessing Dataset Quality and Evaluation Metrics High quality datasets are the foundation for benchmarking and advancing state-of-the-art SLAM or localization methods. However, inherently, the groundtruth (or reference poses in visual localization) for any dataset exhibits some uncertainties. Assessing and quantifying the quality of the groundtruth would be very useful to put the evaluation in context and better understand the performance of the algorithms (as we show in [319]). However, little work has been done in this aspect. Moreover, it would be interesting to establish the connection between the inaccuracy in the groundtruth and the commonly used error metrics. For example, if the groundtruth is inaccurate to a certain degree, what would be the best performance an algorithm could achieve on this dataset?

Following a similar idea, metrics that can take into consideration the uncertainty in the groundtruth are also important. In the evaluation of visual localization methods, a common practice is to use certain error thresholds to account for the uncertainties in the reference poses. For example, if two pose estimates are both below the uncertainty of the reference pose, they are considered equally accurate, regardless of their absolute errors w.r.t. the reference pose. Unfortunately, there is no similar treatment for trajectory evaluation in visual(-inertial) odometry or SLAM. Our work in [324] is the first step in this direction but requires further development and evaluation.

A Large Field-of-View Cameras For Visual Odometry

Reprinted, with permission, from:

Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. "Benefit of Large Field-of-View Cameras for Visual Odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2016. DOI: 10.1109/ICRA.2016.7487210 [318]

Benefit of Large Field-of-View Cameras for Visual Odometry

Zichao Zhang, Henri Rebecq, Christian Forster and Davide Scaramuzza

Abstract — The transition of visual-odometry technology from research demonstrators to commercial applications naturally raises the question: "*what is the optimal camera for vision-based motion estimation?*" This question is crucial as the choice of camera has a tremendous impact on the robustness and accuracy of the employed visual odometry algorithm. While many properties of a camera (e.g. resolution, frame-rate, global-shutter/rolling-shutter) could be considered, in this work we focus on evaluating the impact of the camera field-of-view (FoV) and optics (i.e., fisheye or catadioptric) on the quality of the motion estimate. Since the motion-estimation performance depends highly on the geometry of the scene and the motion of the camera, we analyze two common operational environments in mobile robotics: an urban environment and an indoor scene. To confirm the theoretical observations, we implement a state-of-the-art VO pipeline that works with large FoV fisheye and catadioptric cameras. We evaluate the proposed VO pipeline in both synthetic and real experiments. The experiments point out that it is advantageous to use a large FoV camera (e.g., fisheye or catadioptric) for indoor scenes and a smaller FoV for urban canyon environments.

Multimedia Material

A video showing our omnidirectional visual odometry pipeline performing on real and synthetic data is available at: https://youtu.be/6KXBoprGaR0.

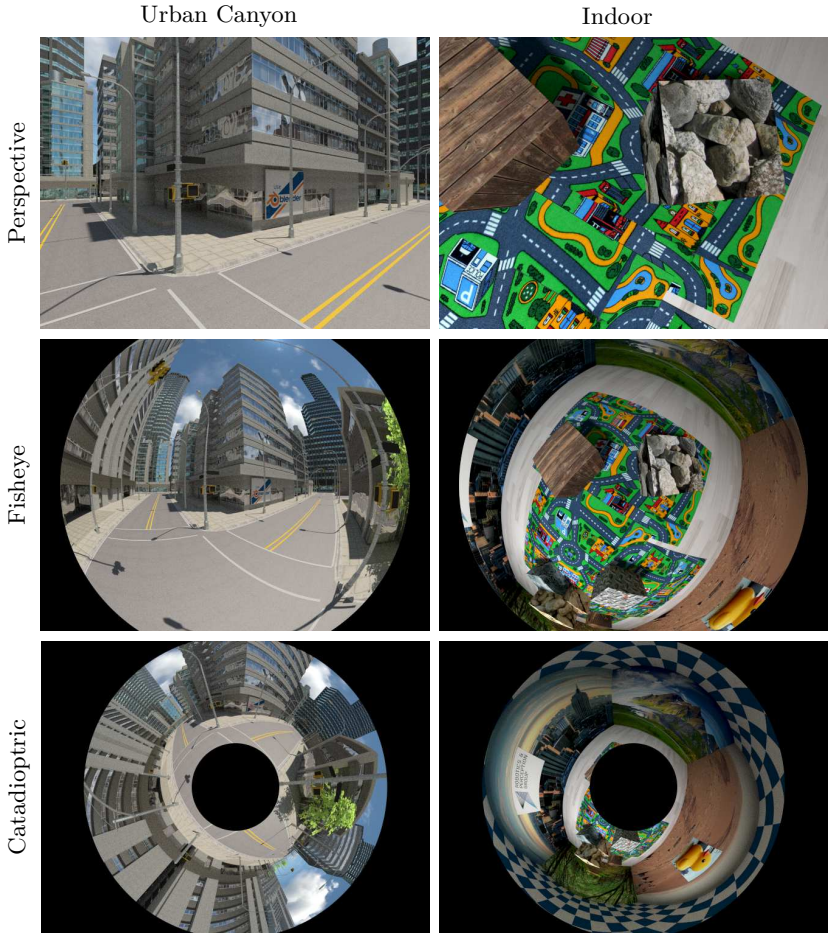


Figure A.1 – Images from our synthetic datasets, showing different FoV cameras.

A.1 Introduction

Estimating the six degrees-of-freedom motion of a camera simply from its stream of images has been an active field of research for several decades [291, 285, 55]. Today, state-of-the-art algorithms run in real-time on smartphone processors and achieve the accuracy and robustness that is required to enable various interesting applications. However, the remaining challenge to enable commercial applications in risky fields such as drone delivery or autonomous driving is *robustness*, especially during fast motions, illumination changes, and in environments with difficult texture. All three nuisances increase the difficulty to track visual cues, which is fundamental to enable vision-based motion estimation.

Our work is motivated by the question of whether the robustness of existing visual odometry (VO) algorithms can be significantly improved by selecting the best camera for the task at hand. In order to minimize the design space, we limit ourselves to the selection of the optimal optics. We are particularly interested in the performance of

omnidirectional cameras, which are fisheye and catadoptric cameras characterized by a large field of view (FoV). In theory, a larger FoV allows tracking visual landmarks over longer periods, which should increase the precision of pose estimation as more measurements are available and, at the same time, increase robustness since the visual overlap between subsequent images is larger. However, increasing the FoV while fixing the resolution means that the angular resolution of a pixel is reduced, hence, lowering the measurement accuracy of a single camera pixel.

The contribution of this work is threefold: after discussion of related work in Section A.1.1, we present in Section A.2 simulation experiments that show the impact of the FoV of a camera on the accuracy and robustness of a canonical VO pipeline. The analysis encompasses standard steps of a visual-odometry pipeline. After studying the theoretical advantages of large FoV cameras and to facilitate an analysis on real images, we describe in Section A.3 challenges and solutions to enable a state-of-the-art VO pipeline (in our case *SVO* [91]) to operate with such images. Therefore, we provide a detailed study of six error metrics on the pose estimation accuracy. Our analysis helps to select the proper error metrics as a function of the camera FoV. Finally, in Section A.4, we evaluate the performance of the proposed omnidirectional SVO algorithm in synthetic as well as real experiments for various camera optics. Since the impact of the camera FoV is a function of the application scenario, we perform the experiments in different environments that reflect typical applications of VO (*e.g.* automotive, drones, gaming). As a further contribution, we publicly release all our synthetic and real datasets that we recorded with different FoV cameras[1].

A.1.1 Related Work

The type of camera used for vision-based navigation methods has a significant impact on the accuracy and robustness of the motion estimation process. A comparison of the performance of a catadioptric and a perspective camera in a visual SLAM system was presented in [217]. A catadioptric camera has a shaped mirror mounted in the front that allows it to capture the full 360 degree view. Experimental results showed that the catadioptric camera outperforms the perspective camera in terms of motion estimation accuracy. However, the catadioptric camera that was used for the experiments had a higher pixel resolution than the perspective camera. Thereby, the lower angular resolution of the larger FoV catadioptric camera was compensated, which provided an unfair advantage to the catadioptric camera. Nevertheless, the comparison presented in [270] experimentally confirmed that a larger FoV camera has a higher motion estimation accuracy than a smaller FoV perspective camera even in the case of a fixed pixel resolution. Unfortunately, the experiments were limited to synthetic data and an indoor environment. In our experiments we confirm these results in an indoor scenario, but we show, both on synthetic and real data, that large FoV cameras perform worse than standard perspective

[1] Available at http://rpg.ifi.uzh.ch/fov.html

cameras in outdoor environments.

Most VO algorithms for omnidirectional cameras [218, 246, 278, 242] rely on robust feature descriptors (*e.g.* SIFT [159]) to establish feature correspondence. To cope with the significant distortion of large FoV images, special descriptors were developed that model the distortion effects to improve feature matching [105, 9, 207, 158]. Other works, such as [61] and [243], used Lucas-Kanade feature tracking [13] to estimate the motion of landmark observations between frames of omnidirectional images.

In this work, we develop a VO pipeline for omnidirectional cameras based on the state-of-the-art *Semi-direct Visual Odometry* (SVO) algorithm [91]. SVO is a very fast odometry algorithm because it does not extract salient features in every frame. Instead, it uses a direct method to estimate the camera motion by mimizing the photometric error of corresponding pixels in subsequent views, similar to LSD [81] and DTAM [192]. However, in contrast to LSD and DTAM, the so called *sparse image alignment* step in SVO works only with sparse pixels and, thus, the convergence radius of the alignment is small and can only be applied on a frame-to-frame basis. Therefore, given the frame-to-frame pixel correspondence, which is found by means of sparse image alignment, the SVO pipeline uses a classic feature-based nonlinear refinement step to minimize the drift. In Section A.3 we describe the required modifications to the standard SVO[2] to enable motion estimation with cameras that have a FoV larger than 120 degrees.

In the next section, we will study the impact of a large FoV on the performance of VO.

A.2 Optimal FoV Studies for Canonical VO Pipeline

In this section, we study the impact of the camera FoV on a canonical VO pipeline by means of Monte Carlo simulations. First, we present a study of the influence of the FoV on the accuracy of three standard components of a VO pipeline: feature correspondence, pose optimization and combined map-pose estimation. By pose optimization we denote the nonlinear refinement of the camera pose, which minimizes the reprojection error of known 3D landmarks. Note that this step is typically applied in an odometry pipeline after finding a solution to the perspective-n-point (PnP) problem. The third experiment implements a canonical VO pipeline combining both depth estimation and pose optimization.

As we will see, the optimal FoV depends greatly on the structure of the environment. Therefore, we perform the study in two different simulated scenes: in the first scene the camera moves in an *urban canyon* that simulates an automotive setting, while, in the second environment, the camera moves in a *confined room* that simulates common indoor scenarios. We evaluate the second scene both with a forward- and downward-looking

[2] Available at http://github.com/uzh-rpg/rpg_svo

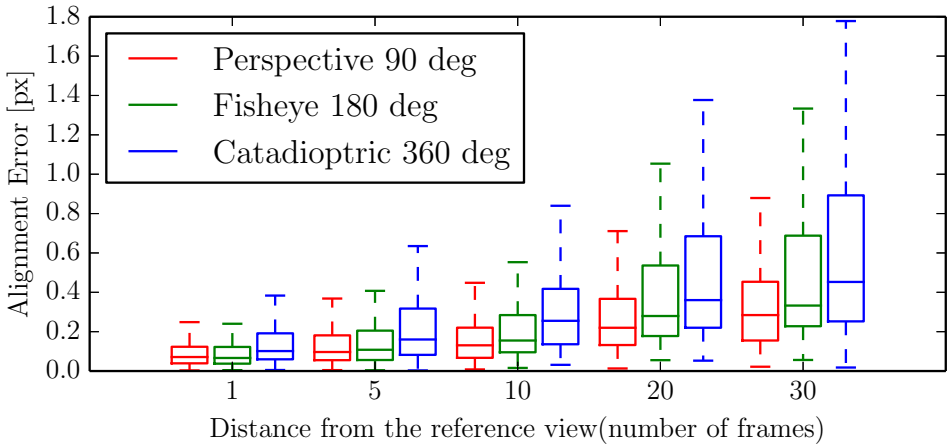


Figure A.2 – *Experiment 1*: Keypoint alignment accuracy for different optics as a function of the distance from the reference frame.

camera.

A.2.1 Experiment 1: Feature Correspondence

The foundation of all geometric vision problems is *feature correspondence*. Hence, the accuracy of 3D landmark measurements (*i.e.* keypoints) in the images directly affects the accuracy of the motion estimate. Therefore, our first experiment evaluates the accuracy of feature correspondence for three different cameras with a constant image resolution. The experiment is based on synthetic scenes rendered for different FoV cameras using Blender (Fig. A.1). Given a keypoint in a reference image, we search for the corresponding keypoint in a subsequent image of the same camera trajectory by means of Lucas-Kanade feature alignment [13]. The groundtruth of the keypoint alignment is calculated by first backprojecting the keypoint from the reference image to the 3D model of the scene to get the 3D landmark and then projecting the landmark to the subsequent frame.

Figure A.2 shows the alignment error as a function of the distance to the reference view. We observe that the accuracy of feature correspondence decreases as we select a frame in the camera trajectory that is farther from the reference frame. Also, the accuracy is slightly reduced when the cameras with larger FoVs are used. The reason for this is that for larger FoV cameras, the image patches used in the alignment suffer from more severe distortions between the reference frame and the selected frame. Given these considerations, in the following experiments we corrupt all feature correspondences with zero-mean additive white Gaussian noise with $\sigma = 0.25$ pixels, which reflects the average uncertainty of our measurements.

A.2.2 Experiment 2: Pose Optimization

The pose optimization step refines the pose $\mathtt{T}_{\mathrm{CW}} \in \mathrm{SE}(3)$ of the camera C with respect to a world frame W by minimizing the reprojection error of the visible landmarks. Hence, we are solving the following nonlinear least-squares problem:

$$\mathtt{T}_{\mathrm{CW}} = \arg\min_{\mathtt{T}} \frac{1}{2} \sum_{i=1}^{N} \parallel \mathbf{r}(\tilde{\mathbf{u}}_i, \pi(\mathtt{T}\ {}_{\mathrm{W}}\mathbf{p}_i)) \parallel^2, \tag{A.1}$$

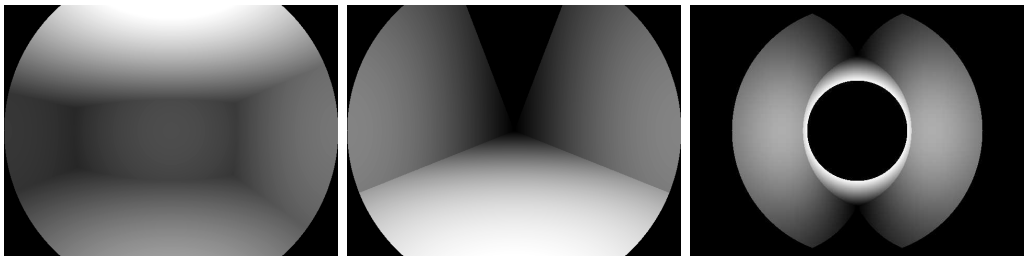
where ${}_{\mathrm{W}}\mathbf{p}_i \in \mathbb{R}^3$ are the landmark positions expressed in the world frame. The metric we use for the reprojection residual $\mathbf{r}(\tilde{\mathbf{u}}_i, \hat{\mathbf{u}}_i)$ between the measured feature position $\tilde{\mathbf{u}}_i \in \mathbb{R}^2$ and the predicted feature position $\hat{\mathbf{u}}_i = \pi(\mathtt{T}\ {}_{\mathrm{W}}\mathbf{p}_i) \in \mathbb{R}^2$ is discussed in more detail in Section A.3.2. By $\pi : \mathbb{R}^3 \to \mathbb{R}^2 : \mathbf{u} = \pi(\mathbf{p})$ we denote the camera projection function.

In this section, we assume that a perfectly known 3D map of the environment is available, whereas in the next section the map is computed using triangulation.

For this experiment, we simulate cameras with varying FoVs using the equidistant fisheye model [125]. The image resolution is fixed, thus the angular resolution decreases as the FoV increases. A forward-looking camera is placed in the center of the scene (Fig. A.3). For each feature in the image plane, the corresponding visible 3D point is found using raytracing on the synthetic scenes. We sample 150 features uniformly in the image plane and compute their corresponding 3D landmarks. Features are corrupted as described in Section A.2.1. With these inputs (2D-3D correspondences), we solve the absolute pose estimation problem. The experiment is repeated 1000 times for each FoV.

Fig. A.4 shows the pose estimation accuracy as a function of the FoV, for the confined room and canyon scenes. It can be observed, that larger FoV cameras perform better in the room scene, despite the loss of angular resolution. Indeed, increasing the FoV yields more evenly distributed landmarks in space (as a larger FoV allows to capture points with a greater angular distance to the optical axis), which stabilizes the pose optimizer (this was also reported in [270]). By contrast, in Fig. A.4b, the translation error reaches a minimum for a FoV of about 215 degrees. This can be interpreted as the result of two competing effects. On the one hand a larger FoV provides a better conditioning for the PnP problem, which raises the pose estimation accuracy. On the other hand, as the FoV grows, the angular resolution decreases (since the image resolution is fixed), leading to larger angular errors on the landmark measurements, thus degrading the pose estimation accuracy. As shown in Fig. A.4b, for the canyon scene, the first effect prevails for small and moderate FoVs while the second eventually becomes predominent for very large FoVs.

Note that this experiment was conducted using a synthetic camera, allowing for arbitrarily large FoVs. While, in reality, fisheye lenses typically reach a maximum FoV of approximately 215° (*e.g.* the KodakSP360 camera), this experiment still provides some



(a) (b) (c)

Figure A.3 – Rendered images showing what the camera sees in different setups: front-looking camera in box environment, front-looking camera in canyon environment, up-looking catadioptric camera in canyon environment. Note that the texture is not given because the groundtruth depth is available.

valuable insight on the trade-off involved when selecting an optics for a given sensor. The vertical line in Fig. A.4 marks the frontier between existing and purely synthetic cameras.

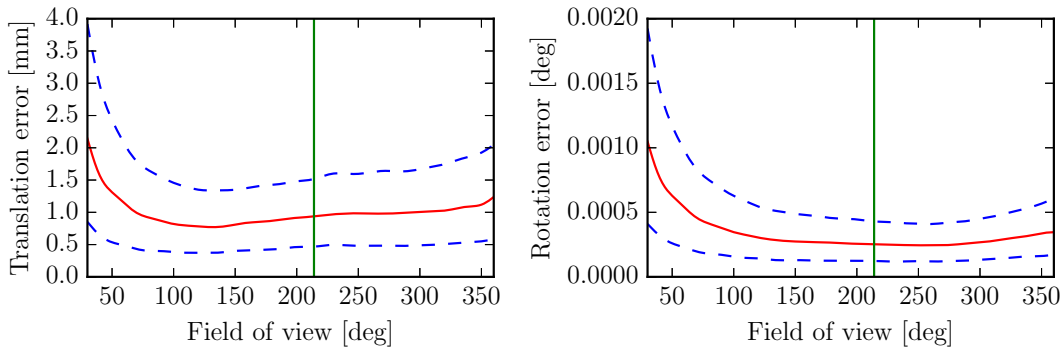
A.2.3 Experiment 3: Canonical Visual Odometry Pipeline

This section assumes no prior knowledge of the map, therefore in the following experiment we simulate a full VO pipeline: from noisy observations we triangulate 3D landmarks that are used to estimate the camera pose of subsequent images (see Fig. A.5). This is a standard approach for incremental camera motion estimation [242].

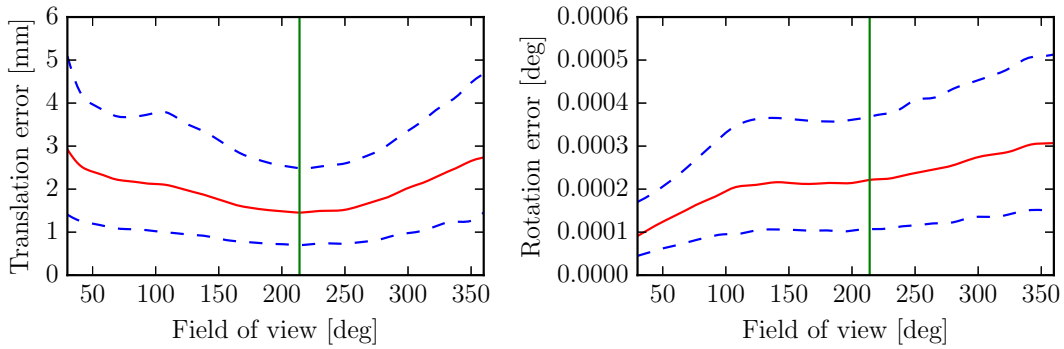
We simulate a camera trajectory (Fig. A.6) in the desired environment and select a reference *keyframe* (red in Fig. A.5) among the trajectory frames. As in the previous experiment, we sample features uniformly in the reference keyframe image plane. Corresponding landmarks (red dots) are triangulated using a set of previous frames (shown in grey), projected and corrupted in the image plane as before. Then, the poses of the following frames (green) are estimated based on the triangulated landmarks. This experiment is conducted for various camera FoVs on both synthetic scenes, with 1000 runs for each configuration. Additionally, in two cases, an up-looking catadioptric camera with a horizontal FoV of 360° and vertical FoV from -50° to +50° above the horizon is simulated.

The results of our experiment are shown in Fig. A.7. The pose estimation accuracy is evaluated as a function of the distance to the keyframe. This provides a measure of robustness and drift: Robustness is increased if we can move farther away from the last keyframe without loosing much pose accuracy, whereas drift is reduced if we can track features over longer time intervals.

The main conclusion from these experiments is that, for visual localization, large FoV



(a) Confined room environment



(b) Urban canyon environment

Figure A.4 – *Experiment 2*: Pose estimation accuracy with respect to FoV for two synthetic scenes. Solid line is the median; dashed lines bound the confidence interval.

cameras should be preferred in confined environments (*e.g.* indoor flight for a drone), whereas smaller FoV cameras will perform better for forward-looking cameras in canyon-like environments (typically a camera mounted on a car in the city). Specifically, the analysis of the plots in Fig. A.7 follows.

Room environment Regardless of the camera orientation, the motion estimation accuracy grows with the FoV (Figs. A.7a and A.7b). The superiority of wide angle optics in this setup stems from two different beneficial effects: first, the better angular distribution of features, as demonstrated in Section A.2.2; and second, the ability of large FoV cameras to track features longer greatly increases the robustness of visual localization in this environment (see Fig. A.7b: almost all features remain visible as the down-looking camera moves). Interestingly, the catadioptric camera performs slightly worse than the large FoV fisheye cameras. This is consistent with the results from the previous section: the localization accuracy stops increasing when the FoV reaches a threshold of around 210 degrees, and the catadioptric camera's self-occlusion zone furthermore reduces the available image area compared to the fisheye cameras.

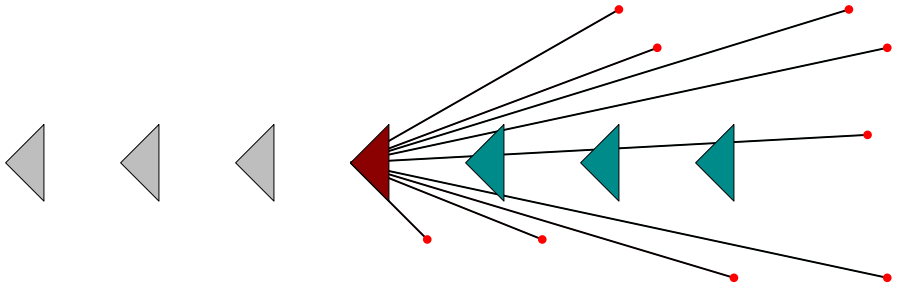
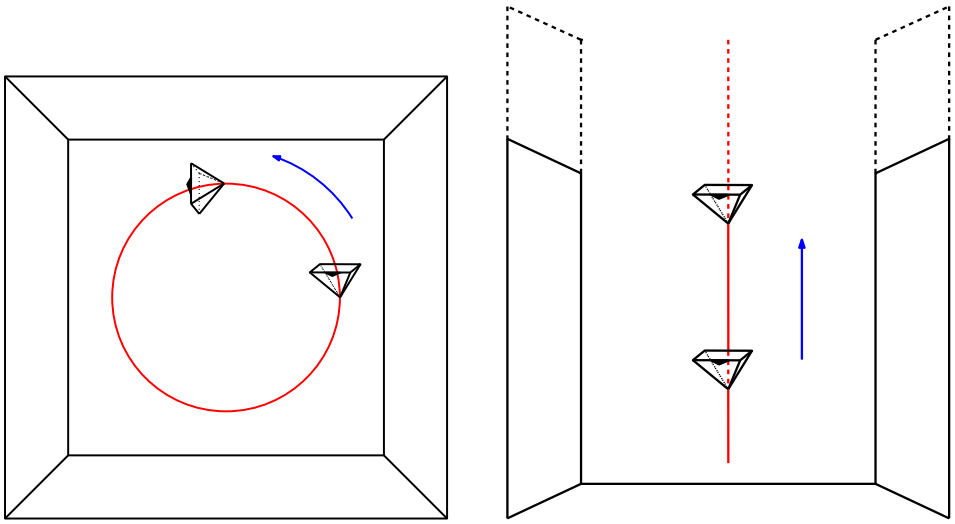


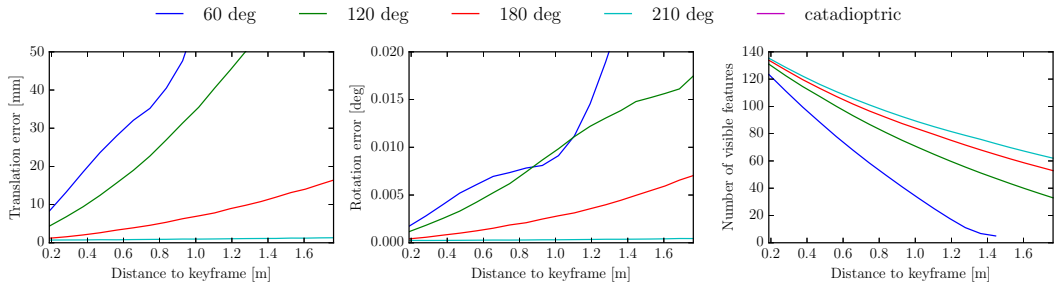
Figure A.5 – *Experiment 3*: Camera moving along the trajectory, keyframes and triangulated landmarks.



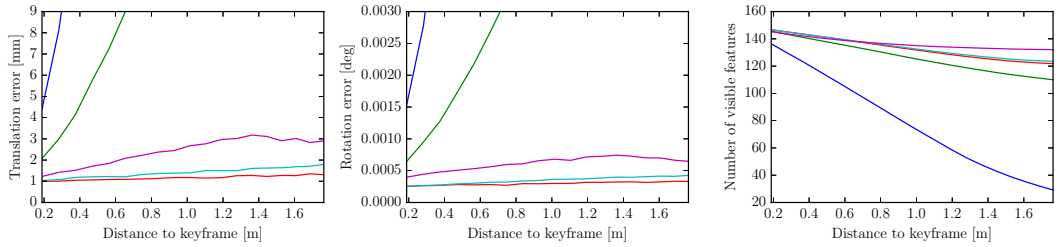
(a) Box environment **(b)** Canyon environment

Figure A.6 – *Experiment 3*: Top views of the different setups. For the box scene, the experiment is conducted with both downward-looking and forward-looking camera but only the latter is shown in this figure.

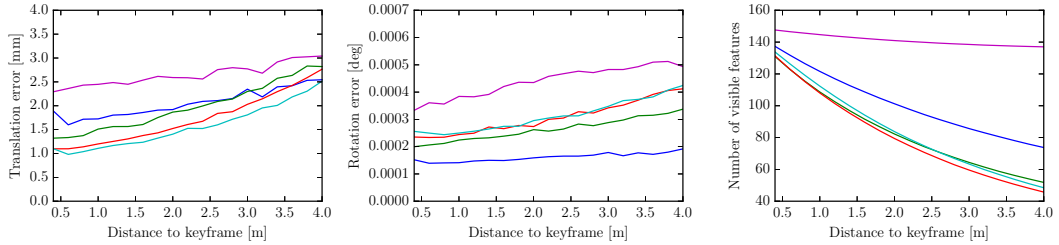
Front-looking camera in canyon environment This experiment (Fig. A.7c) shows that a smaller FoV should be preferred in an urban canyon scenario. The reason why large FoV optics perform worse in this setup is twofold. Firstly, because the depth range of the scene is much higher than the room scene. Whereas the triangulation error introduced by the loss of angular resolution remains small when the depth range of the landmarks is limited, it eventually becomes predominant when the depth range is very high (in the canyon environment, the farthest point is 250m away from the camera). Secondly, because of the uniform sampling of the features in the image plane, the landmarks corresponding to the features extracted in the reference frame tend to be farther away for smaller FoV cameras, thus having a slower apparent motion with respect to the camera. These features can therefore be tracked more reliably (because of the reduced optical flow between two successive frames), and longer. Our experiment confirmed this somewhat surprising fact (third column of Fig. A.7c): the camera with the smallest FoV observes



(a) Forward-looking camera in the box environment



(b) Downward-looking camera in the box environment



(c) Forward-looking camera in the canyon environment

Figure A.7 – *Experiment 3*: Pose error and number of visible features for different FoVs in the canonical VO pipeline.

features longer on average.

A.3 Implementation of a Semi-Direct Omnidirectional Visual Odometry

In this section, we describe the challenges and, accordingly, our solutions, to enable a state-of-the-art VO pipeline to work with wide field-of-view cameras. In particular, we develop a unified VO system that works with fisheye as well as catadioptric cameras.

We base our developments on the state-of-the-art SVO [91] pipeline. The standard SVO algorithm does not scale to large FoV cameras, which required us to perform three main modifications: (1) implementation of polynomial and equidistant camera models that adequately model large FoV cameras; (2) use of reprojection-error metrics based on bearing vectors in the pose optimization (bundle adjustment) step; (3) sampling of the curved epipolar line based on the unit sphere for better correspondence search and

triangulation.

In the following, we discuss the implementation of these modifications in more detail.

A.3.1 Omnidirectional Camera Model

The omnidirectional camera model from [245] is used in our work. In this model, a Taylor series expansion is used to describe the image projection function. We choose this camera model largely due to its advantage of being able to describe catadioptric and fisheye cameras within one unified framework compared to other omnidirectional models such as the unified projection model [100] and the equidistant model [125].

A.3.2 Error Metrics for Pose Optimization

The SVO algorithm finds 2D-3D landmark correspondence using direct methods, specifically *sparse image alignment* and *feature alignment* [91]. In the subsequent pose optimization step, the six degree of freedom (DoF) pose of a frame is refined by minimizing the reprojection error. This problem is formalized in (A.1) and can be solved by standard least squares optimization techniques such as the Gauss-Newton method.

In a standard implementation, one would minimize the image error (see Fig. A.8):

$$\mathbf{r}_u = \tilde{\mathbf{u}} - \pi(\mathbf{p}), \tag{A.2}$$

where $\mathbf{p} = [p_x, p_y, p_z]^\top$ is the 3D landmark (in the camera frame). However, this requires to compute the projection function and its Jacobian at each iteration, which can be expensive when complicated camera models are used. Therefore, SVO minimized the reprojection error on the *unit plane*:

$$\mathbf{r}_m = \tilde{\mathbf{m}} - \left[\frac{p_x}{p_z}, \frac{p_y}{p_z}\right]^\top, \tag{A.3}$$

where $\tilde{\mathbf{m}}$ is the corresponding position of observation $\tilde{\mathbf{u}}$ on the unit plane. Unfortunately, this approach does not scale when the FoV is large as p_z approaches zero for landmarks observed at the border of the image. Hence, implementations of omnidirectional vision systems such as [151, 138] use the angular error $\Delta\theta$ between the unit bearing vectors $\tilde{\mathbf{f}}$ and $\mathbf{f}$ corresponding to $\tilde{\mathbf{u}}$ and $\mathbf{p}$, respectively:

$$\mathbf{r}_{a1} = 1 - \tilde{\mathbf{f}}^\top \mathbf{f} \quad \Longrightarrow \quad \|\mathbf{r}_{a1}\|^2 = 4\sin^4(\Delta\theta/2), \tag{A.4}$$

$$\mathbf{r}_{a2} = \arccos(\tilde{\mathbf{f}}^\top \mathbf{f}) \quad \Longrightarrow \quad \|\mathbf{r}_{a2}\|^2 = (\Delta\theta)^2. \tag{A.5}$$

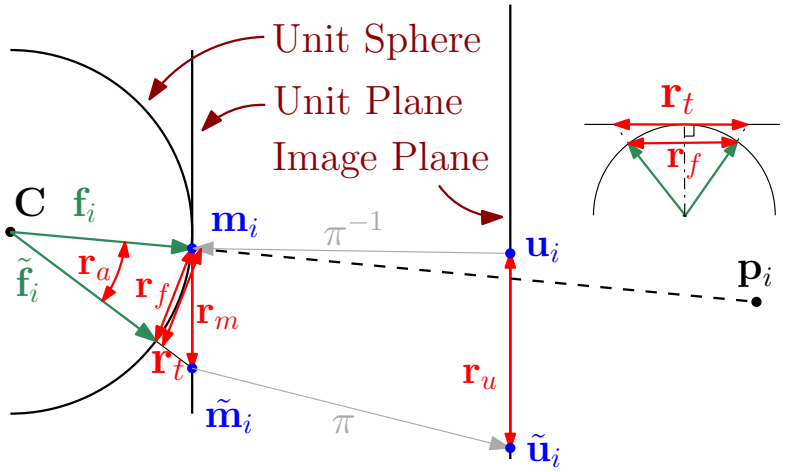


Figure A.8 – Different error metrics that we evaluated for pose optimization. The landmark $\mathbf{p}_i \in \mathbb{R}^3$ is measured at pixel location $\tilde{\mathbf{u}}_i$. After applying the inverse camera projection $\tilde{\mathbf{f}}_i = \pi^{-1}(\tilde{\mathbf{u}}_i)$, which also models the distortion, we find the corresponding bearing vector $\tilde{\mathbf{f}}_i$ and unit plane coordinates $\tilde{\mathbf{m}}_i$. Given an estimate of the pose of the camera center $\mathbf{C}$, we can predict the feature position $\mathbf{u}_i = \pi(\mathbf{p}_i)$ or use intermediate results (before applying the camera distortion) to find the predicted bearing vector $\mathbf{f}_i$ or unit plane coordinates $\mathbf{m}_i$. We evaluate the efficiency and accuracy of various residual metrics $\{\mathbf{r}_{a1}, \mathbf{r}_{a2}, \mathbf{r}_t, \mathbf{r}_f, \mathbf{r}_m, \mathbf{r}_u\}$.

Instead, the difference between the bearing vectors gives:

$$\mathbf{r}_f = \tilde{\mathbf{f}} - \mathbf{f} \quad \Longrightarrow \quad \|\mathbf{r}_f\|^2 = 4\sin^2(\Delta\theta/2). \tag{A.6}$$

The authors of [201] studied different error metrics for the omnidirectional SfM problem and showed experimentally that the following tangential error was the best error metric for the pose estimation problem:

$$\mathbf{r}_t = \sqrt{\frac{2}{1+\tilde{\mathbf{f}}^\top\mathbf{f}}}(\tilde{\mathbf{f}} - \mathbf{f}) \implies \|\mathbf{r}_t\|^2 = 4\tan^2(\Delta\theta/2). \tag{A.7}$$

To answer the question of which error metric to use, the same Monte Carlo experiment as in Section A.3.2 is performed using different error metrics. The average position errors after the optimization are shown in Fig. A.9. It can be observed that the image error $\mathbf{r}_u$, the tangential error $\mathbf{r}_t$ and the bearing vector difference error $\mathbf{r}_f$ have comparable performances for all the FoVs. In comparison, the unit plane error $\mathbf{r}_m$ results in equal accuracy for small FoVs, but exhibits large errors for large FoVs. When using the angular error metrics $\mathbf{r}_{a1}$ and $\mathbf{r}_{a2}$, the pose estimations oscillate around the true values instead of converging after 4-6 iterations as the other error metrics.

The time cost for each error metric is summarized in Table A.1. The angular error $\mathbf{r}_{a1}$ and $\mathbf{r}_{a2}$, which are not listed in the table, have a much worse time performance because of the convergence problem.

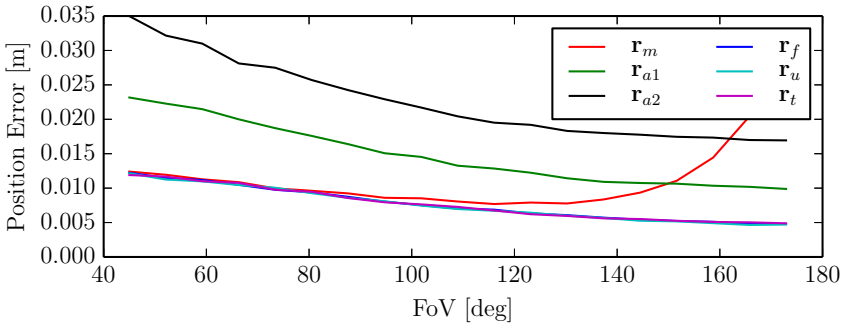


Figure A.9 – Pose optimization errors of the error metrics in Fig. A.8 under different FoVs. Only the position errors are given here for briefness, since the rotation and reprojection errors show a similar trend.

Table A.1 – Average Convergence Time

	$\mathbf{r}_u$	$\mathbf{r}_m$	$\mathbf{r}_f$	$\mathbf{r}_t$
Time(ms)	0.4	0.2-0.25*	0.28	0.31

* increases with the field of view

Therefore, it can be concluded that for pose optimization, the unit plane error $\mathbf{r}_m$ should be used for small FoVs (e.g. perspective cameras with less than 100° FoVs) due to its efficiency and for large FoVs, the bearing vector difference error $\mathbf{r}_f$ should be used. In the experiments of this work, the bearing vector difference error $\mathbf{r}_f$ is used for omnidirectional cameras and the unit plane error $\mathbf{r}_m$ for perspective cameras.

A.3.3 Feature Correspondence along Curved Epipolar Lines

SVO triangulates new landmarks from known camera poses by means of a *depth filter* [91]: In a selected reference image $\mathtt{I}_r$ salient corners are selected for which the depth is estimated using measurements from older and newer frames $\mathtt{I}_k$. A measurement is obtained by sampling the epipolar line in a neighbouring image $\mathtt{I}_k$ pixel by pixel and computing the correlation of an 8×8 pixel patch with the reference patch in $\mathtt{I}_r$. The pixel on the epipolar line with highest correlation is used to update the depth of the reference pixel through triangulation (see Fig. A.10).

For omnidirectional cameras, the epipolar line in $\mathtt{I}_k$ is not straight but forms a curve. To sample pixels on the curved epipolar line, we compute the bearing vectors $\{\mathbf{f}^{\min}, \mathbf{f}^{\max}\}$ that correspond to the confidence interval of the current depth estimate $d \pm 2\sigma_d = \{d^{\min}, d^{\max}\}$ in the reference image. Subsequently, we rotate a bearing vector $\mathbf{f}'$ in small angular steps from $\mathbf{f}^{\min}$ to $\mathbf{f}^{\max}$ around the axis $\mathbf{f}^{\min} \times \mathbf{f}^{\max}$ and project it on the image $\mathbf{u}' = \pi(\mathbf{f}')$,

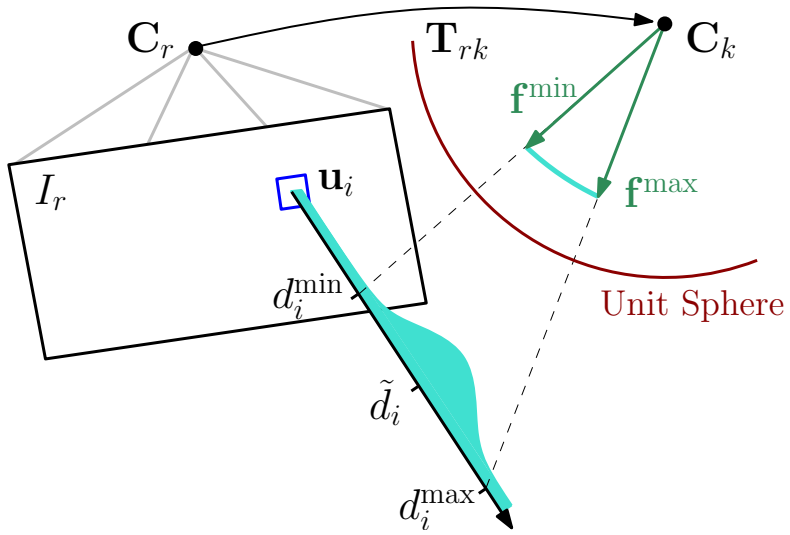
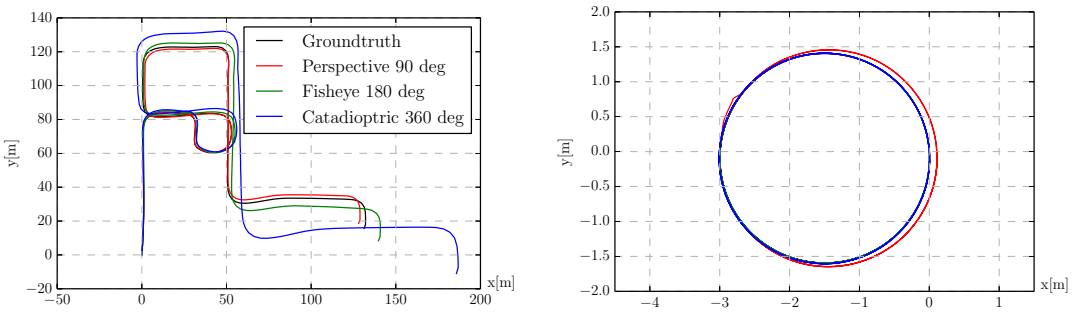


Figure A.10 – Epipolar search on unit sphere for depth filter update.



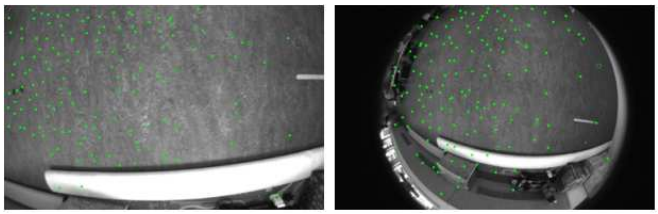
(a) Urban Canyon **(b)** Indoor

Figure A.11 – *Synthetic Datasets*: Top views of the estimated trajectories.

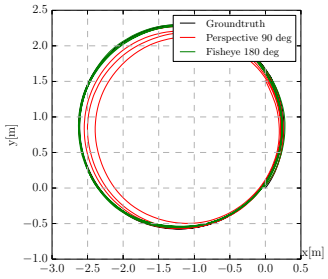
which results in a pixel location $\mathbf{u}'$ that lies on the curved epipolar line.

A.4 Experiments

The modified SVO algorithm described in the previous section allows us to verify our FoV studies in Section A.2 on real and synthetic images. In the following, we first discuss the synthetic experiments and subsequently the real experiments performed with a micro aerial vehicle (MAV) and an automobile.



(a) Feature tracking (left: perspective, right: fisheye)



(b) Trajectory top view

Figure A.12 – *Real Datasets*: Results on the *Flyroom* sequence.

A.4.1 Synthetic Datasets

To generate photorealistic synthetic images, we used the Cycles raytracing engine[3] implemented in Blender. In addition to the already built-in perspective and equidistant fisheye camera models, we implemented a catadioptric camera model based on [245], which we release as an open-source patch for Blender[4].

We first ran our algorithm on two synthetic datasets: *Urban Canyon* and *Indoor* (Fig. A.1). The *Urban Canyon* dataset simulates a forward-looking camera mounted on a car driving in a city environment and the *Indoor* dataset contains views from a downward-looking camera moving along a circle in an indoor environment. We rendered these two datasets with three different camera models respectively: perspective (90° FoV), fisheye (180° FoV) and catadioptric (360° FoV). Note that for the catadioptric camera, the same trajectories were used for the rendering but the camera was set up to be upward-looking (facing the mirror).

The top view of the trajectories estimated is shown in Fig. A.11. It can be observed that the perspective camera exhibits the smallest drift in the *Urban Canyon* dataset, followed by the fisheye camera and the catadioptric camera. However, in the *Indoor* dataset, while the trajectories estimated by the omnidirectional cameras are almost identical to the groundtruth, the perspective camera exhibits significant drift.

A.4.2 Real Datasets

To further verify our FoV studies with real world scenarios, we first recorded a *Flyroom* dataset with a downward-looking camera mounted on a MAV. The camera was 1 m above the ground and moved along a circle of about 1.5 m radius at a speed of 1.3 m/s. The datasets were recorded with a perspective camera (90° FoV) and a fisheye one (180° FoV),

[3] http://www.blender.org/manual/render/cycles/
[4] https://github.com/uzh-rpg/rpg_blender_omni_camera

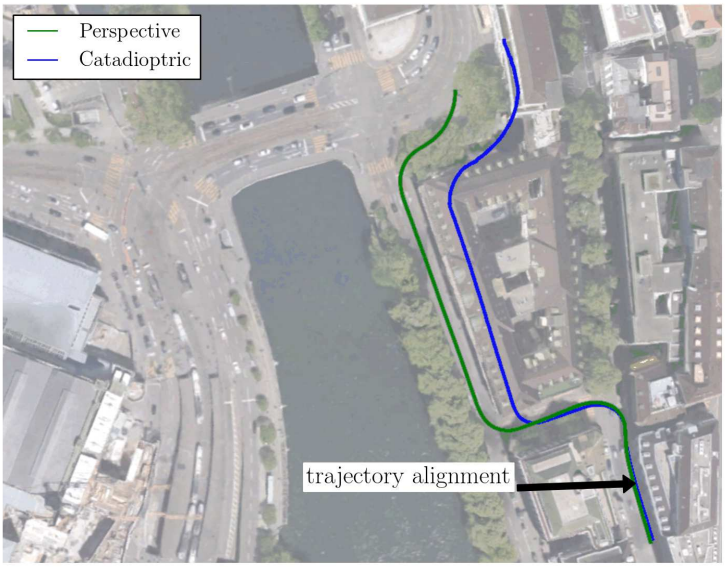


Figure A.13 – *Real Datasets*: Results on the *Zurich* sequence. The first straight segment of each estimated trajectory is aligned with the corresponding part of the streets that the car drove along.

respectively. The groundtruth was acquired via a motion capture system. Fig. A.12 shows the performance comparison between the two cameras. It can be observed from Fig. A.12b that the trajectory estimated by the fisheye camera follows the circle precisely, while the trajectory estimated by the perspective one drifts away as it repeats the circle. It can be seen from Fig. A.12a that while the perspective camera can only track features that are very close, the fisheye one can keep track of features from a much larger area.

We also ran our algorithm on the *Zurich* dataset from [244]. The *Zurich* dataset contains two sequences: a forward-looking perspective camera (45° FoV) and an upward-looking catadioptric camera (360° FoV).The two sequences were recorded on the same car simultaneously while the car drove through Zurich downtown. Since no groundtruth is available for this dataset, the estimated trajectories were aligned with a satellite map for evaluation. As is shown in Fig. A.13, the trajectory estimated with the perspective camera is more consistent with the streets on the map.

A.4.3 Discussion

The results from the above experiments are consistent with our simulations and analysis presented in Section A.2.

- For indoor scenarios, such as the *Indoor* and *Flyroom* datasets, large FoV omnidirectional cameras outperform the perspective ones. The reason for this is twofold: first,

features are more evenly distributed in space, which stabilizes the pose estimation, and, second, the camera can track features for a longer time.

- For outdoor environments such as the *Urban Canyon* and *Zurich* datasets, the trajectories can be estimated more accurately using perspective cameras, mainly because the loss of angular resolution for higher FoVs is drastically amplified by the higher depth range.

A.5 Conclusions

It is well known that VO can benefit from large FoVs. Indeed, a larger FoV theoretically allows for tracking visual landmarks over longer periods, which should increase the precision of pose estimation (since more measurements are available) and increase robustness since the visual overlap between successive images is larger. However, at the same time, increasing the FoV while fixing the resolution decreases the angular resolution of the image, thus, lowering the measurement accuracy of a single camera pixel.

In this work, we showed that for a constant image resolution, the best choice of FoV and optics is not as straightforward as it seems. We first performed extensive simulations to study the impact of different FoVs on the standard VO modules as well as the complete pipeline, which point out that large FoV cameras (*e.g.* omnidirectional cameras) are preferable in indoor environments, while smaller FoV cameras perform better in urban canyon scenarios. We also performed experiments using both synthetic and real world datasets and these are in accordance with the simulation results. Moreover, we provided an in-depth analysis of the challenges arising when adapting VO algorithms for large FoV cameras, and adapted the state-of-the-art algorithm SVO to work with omnidirectional cameras.

Based on the simulations and experiments, it can be concluded that for small, confined environments, large FoV cameras should be used and for larger scale scenarios, small FoV cameras should be preferred.

B Active Exposure Control for Robust Visual Odometry

Reprinted, with permission, from:

Z. Zhang, C. Forster, and D. Scaramuzza. "Active exposure control for robust visual odometry in HDR environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017, pp. 3894–3901. DOI: 10.1109/ICRA.2017.7989449 [316]

Active Exposure Control for Robust Visual Odometry in HDR Environments

Zichao Zhang, Christian Forster and Davide Scaramuzza

Abstract — We propose an active exposure control method to improve the robustness of visual odometry in HDR (high dynamic range) environments. Our method evaluates the proper exposure time by maximizing a robust gradient-based image quality metric. The optimization is achieved by exploiting the photometric response function of the camera. Our exposure control method is evaluated in different real world environments and outperforms the built-in auto-exposure function of the camera. To validate the benefit of our approach, we adapt a state-of-the-art visual odometry pipeline (SVO) to work with varying exposure time and demonstrate improved performance using our exposure control method in challenging HDR environments.

Multimedia Material

A video demonstrating the improvement on different visual odometry algorithms is available at https://youtu.be/TKJ8vknIXbM.

B.1 Introduction

Recently, VO (visual odometry) algorithms have reached a high maturity and there is an increasing number of applications in various fields, such as VR/AR. Although many impressive results have been presented, one of the remaining challenges is robustness in HDR environments. The difficulty in such environments comes from the limitations of both the sensor and the algorithm. For conventional cameras, the dynamic range is narrow compared to real world environments. Without proper exposure control, images can be easily overexposed or underexposed, and very little information can be

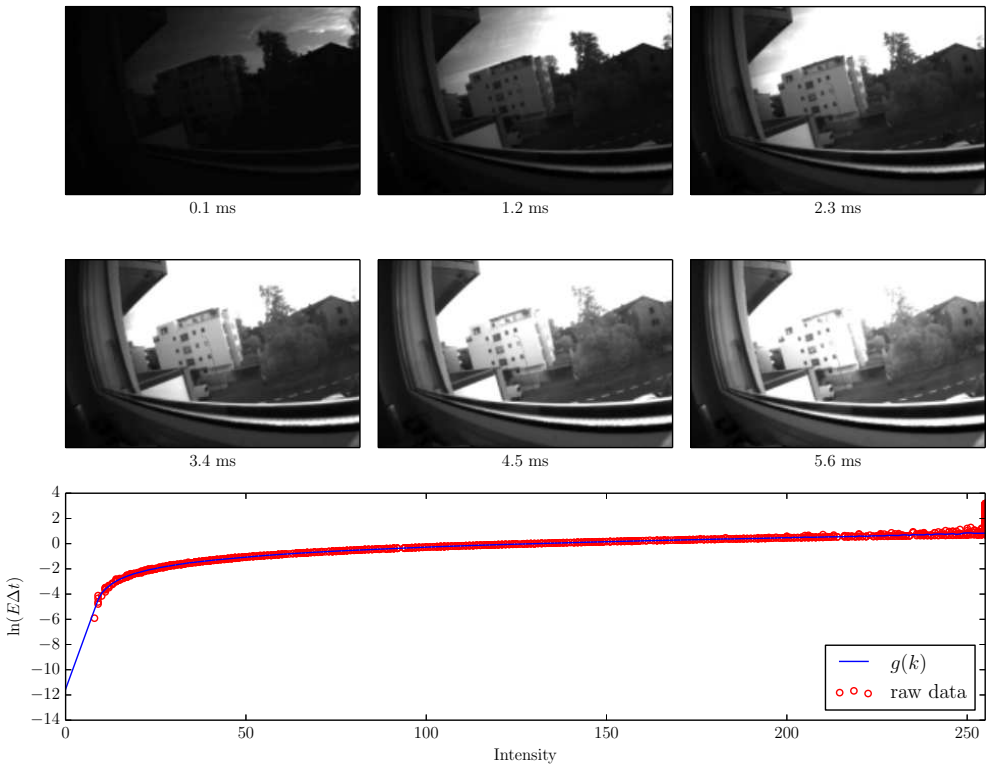


Figure B.1 – The top two rows are images captured under different exposure times, used as the input to the calibration. The third row shows the recovered inverse response function.

extracted from such images. In order to overcome the problem of the narrow dynamic range, many cameras automatically adjust the exposure time. The change of exposure time, however, breaks the brightness constancy assumption across consecutive frames, which is the underlying assumption of many VO algorithms. Therefore, to work in HDR environments, a VO algorithm should be *active*, instead of passive. An active VO algorithnm, on the one hand, must actively adjust the exposure time of the camera to maximize the information for VO; on the other hand, the effect of the varying exposure time needs to be explicitly compensated.

While the topic of exposure control has been studied extensively, little work has been done to optimize the exposure time for VO applications. Moreover, most exposure control methods rely on heavily engineered parameters, because of the lack of a quantitative knowledge on how the change of the exposure time affects the image. Regarding exposure compensation, a widely used technique is to model the brightness change with an affine transformation. Alternatively, researchers have recently exploited the photometric response function of the camera for exposure compensation [152, 82]. While both methods are shown to work, to the best of our knowledge, there is no comparison study of them yet in the existing literature. It would be interesting to know, from a practical perspective, which compensation method should be used when building VO applications.

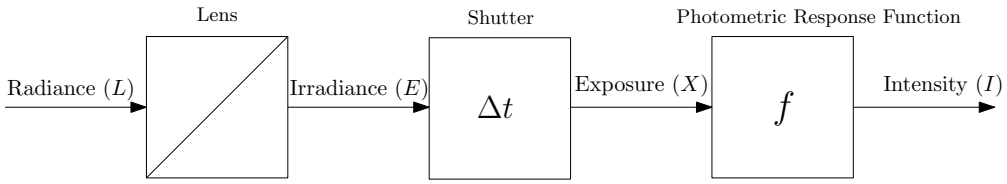


Figure B.2 – Image Acquisition Process

In this paper, we first propose an active exposure control method to maximize the gradient information in the image. This is inspired by the observation that most vision algorithms, including VO, actually extract information from gradient-rich areas. For instance, corners are essentially points where the gradient is large in two orthogonal directions [107]; direct VO algorithms also make use of the pixels with high gradients [82, 80]. Therefore, we propose a gradient-based image quality metric and show that it is robust in HDR environments by an extensive evaluation in different scenarios. Moreover, we use the photometric response function of the camera to design our exposure control scheme. By exploiting the photometric response function, we are able to evaluate the derivative of our metric with respect to the exposure time. Such information enables us to apply mathematically grounded methods, such as gradient descent, in exposure control. Second, we introduce our adaptations of exposure compensation to a state-of-the-art VO algorithm, namely SVO (Semi-direct Visual Odometry [91]). We formulate these adaptations in an algorithm-agnostic manner, so that they can be easily generalized to other VO algorithms. In addition, an experimental comparison of the aforementioned exposure compensation methods is presented. Finally, we demonstrate in several real-world experiments that, with the proposed exposure control method, our VO algorithm is able to operate in HDR environments.

B.1.1 Related Work

Many existing exposure control approaches use heuristics based on image statistics, such as the mean intensity value and the intensity histogram. A system for configuring the camera parameters was presented in [190]. The exposure time was selected according to the intensity histogram of the image. Their method was successfully used in practice during the RoboCup competitions [135]. More recently, Torres *et al.* [289] used a set of indicators from the intensity histogram and the cumulative histogram to capture the different aspects of the image quality, and a camera exposure control method was designed based on these indicators.

By contrast, other works explicitly explore the information in the image. Lu *et al.* [160] characterized the image quality using Shannon's entropy. They showed experimentally that the entropy of the image was related to the performance of the object recognition algorithm. Therefore, the exposure control was achieved by searching for the highest

entropy in the parameter space of the camera. Closely related to our work is [256], which used the gradient information within an image to select the proper exposure time. The authors defined an information metric based on the gradient magnitude at each pixel. The exposure change was simulated by applying different gamma corrections to the original image to find the gamma value that maximizes the gradient information. Then, the exposure time was adjusted based on the gamma value. Our work differs from [256] in two aspects. First, we use a different gradient-based metric, which we demonstrate to be more robust. Second, our control scheme also exploits the photometric response function of the camera.

Different methods have been proposed for exposure compensation. Jin *et al.* [122] used an affine transformation to model the illumination change in the feature tracking problem and showed success tracking under significant illumination changes. Kim *et al.* [134] jointly estimated the feature displacements and the camera response function and used the estimated response function to improve the performance of feature tracking. More recently, Engel *et al.* [82] used an affine brightness transfer function to compensate for the variation of the exposure time and applied it to VO. In addition, they also proposed to use the photometric response function of the camera for exposure compensation if the exposure time of the camera is known. Similarly, Li *et al.* [152] exploited the camera response function to account for the brightness change caused by the auto-exposure of the camera and applied it to a tracking and mapping system.

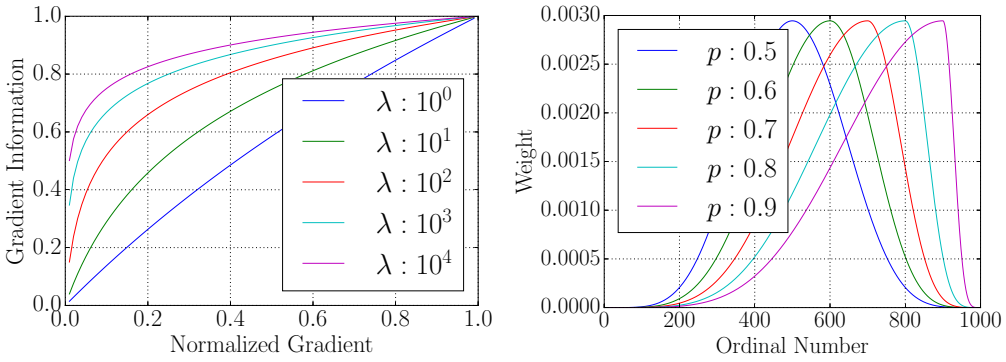
After introducing the photometric response function in Section B.2, we propose our gradient-based image quality metric in Section B.3. Based on the photometric response function and the image quality metric, our exposure control method is described in Section B.4. Then, in Section B.5, we describe our adaptations of exposure compensation to a VO algorithm and compare the two commonly used exposure compensation techniques mentioned above experimentally. Finally, we validate our exposure control algorithm and demonstrate robust VO in HDR environments in Section B.6.

B.2 Photometric Response Function

In this work, we use the photometric response function proposed in [67]. For completeness, we briefly introduce the function in the following.

The image formation process is illustrated in Fig. B.2. For each pixel, the *irradiance* E describes the amount of energy that hits the pixel per time unit, and the *exposure* X is the total amount of energy received by the pixel during the exposure time Δt. The *photometric response function* f maps the exposure X to the intensity I in the image:

$$I = f(X) = f(E\Delta t). \tag{B.1}$$



(a) Information in M_{shim} **(b)** M_{perc} weights ($k = 5$).

Figure B.3 – The mapping function of M_{shim} and the weights in M_{softperc}

Note that $f(\cdot)$ is invertible because the intensity should increase monotonically with the exposure. Then, for convenience, we can define the *inverse response function*

$$g = \ln f^{-1}, \tag{B.2}$$

and (B.1) can be written as

$$\tilde{g}(I) \quad = \quad \ln E + \ln \Delta t. \tag{B.3}$$

Obviously, for a digital image, where the possible intensities are a range of discrete values $\{0, 1, \ldots, Z_{\max}\}$, $\tilde{g}$ can only take values $g(k), k = 0, 1, \ldots, Z_{\max}$. These values can be determined by analyzing the images of a static scene captured under different exposure times. For the details of the photometric calibration process, we refer the reader to [67]. A sample calibration sequence and the recovered inverse response function g are illustrated in Fig. B.1. After recovering g, we estimate a tenth order polynomial to fit the discrete values in (B.3) and use the polynomial to calculate the derivative g'.

In the next section, the image quality metric used in our exposure control method is introduced.

B.3 Image Quality Metrics

The metrics for image quality are highly application-dependent. Regarding VO applications, the gradient information is of great importance for both feature-based and direct methods. In this section, we first introduce several gradient-based metrics and then compare them on real world data.

B.3.1 Gradient-Based Metrics

Given an image, denoted as $\mathtt{I}$, captured with an exposure time Δt, the magnitude of the gradient at a pixel $\mathbf{u}$ is

$$G(\mathtt{I}, \mathbf{u}, \Delta t) \quad = \quad \|\nabla \mathtt{I}(\mathbf{u}, \Delta t)\|^2, \tag{B.4}$$

where $\nabla \mathtt{I}(\cdot) = [\frac{\partial \mathtt{I}}{\partial x}, \frac{\partial \mathtt{I}}{\partial y}]^\top$. In the rest of this section, we drop the notation of $\mathtt{I}$ in (B.4) for simplicity.

A straightforward metric is the sum of (B.4) on all the pixels in the image:

$$M_{\text{sum}} \quad = \quad \sum_{\mathbf{u}_i \in \mathtt{I}} G(\mathbf{u}_i). \tag{B.5}$$

Alternatively, Shim *et al.* [256] defined the *gradient information* of a pixel $\mathbf{u}_i$ as

$$m_{\mathbf{u}_i} = \begin{cases} \frac{1}{N} \log(\lambda(\tilde{G}(\mathbf{u}_i) - \sigma) + 1), & G(\mathbf{u}_i) \geq \sigma \\ 0, & G(\mathbf{u}_i) < \sigma \end{cases}, \tag{B.6}$$

where $\tilde{G}$ is the gradient magnitude normalized to the range of $[0, 1]$, $N = \log(\lambda(1 - \sigma) + 1)$ is a normalization factor to bound the gradient information to the range of $[0, 1]$, σ is an activation threshold, and λ determines whether strong or weak intensity variations are emphasized. Then the total gradient information in an image is

$$M_{\text{shim}} \quad = \quad \sum_{\mathbf{u}_i \in \mathtt{I}} m_{\mathbf{u}_i}. \tag{B.7}$$

M_{shim} can be interpreted as a weighted sum of the gradient magnitudes from all the pixels. The mapping from the normalized gradient magnitude $\tilde{G}$ to the gradient information (B.6) is plotted in Fig. B.3a for different λs. For both M_{sum} and M_{shim}, the main problem is that the squared sum is not a robust estimator of the scale of the gradient magnitudes (see Section B.3.2).

Instead, we consider using a certain percentile of all the gradient magnitudes as a robust estimator:

$$M_{\text{perc}}(p) \quad = \quad \text{percentile}(\{G(\mathbf{u}_i)\}_{\mathbf{u}_i \in \mathtt{I}}, p), \tag{B.8}$$

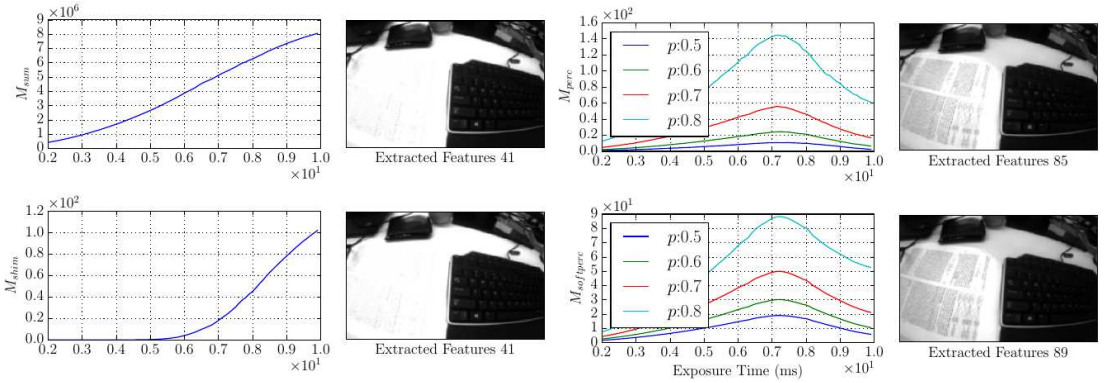


Figure B.4 – A HDR scene. The left column illustrates how different metrics change with the exposure time. The right column shows the best image in terms of each metric, respectively.

where p indicates the percentage of the pixels whose gradient magnitudes are smaller than M_{perc}. For example, M_{perc} is the median of all the gradient magnitudes when $p = 0.5$.

Lastly, we define another gradient-based metric, which is called *soft percentile* in this paper. We first sort the gradient magnitudes of all the pixels $\{G(\mathbf{u}_i)\}_{\mathbf{u}_i \in \mathtt{I}}$ in an ascending order. The sorted gradient magnitudes are denoted as $\{G_{\text{ith}}\}_{i \in [0,S]}$, where S is the total number of pixels in the image. Then we calculate the soft percentile metric as a weighted sum of the sorted gradient magnitudes:

$$M_{\text{softperc}}(p) \quad = \quad \sum_{i \in [0,S]} W_{\text{ith}}(p) \cdot G_{\text{ith}}. \tag{B.9}$$

The weights $\{W_{\text{ith}}(p)\}_{i \in [0,S]}$ are

$$W_{\text{ith}} = \begin{cases} \frac{1}{N} \sin(\frac{\pi}{2\lfloor p \cdot S \rfloor} i)^k, & i \leq \lfloor p \cdot S \rfloor \\ \frac{1}{N} \sin(\frac{\pi}{2} - \frac{\pi}{2} \frac{i - \lfloor p \cdot S \rfloor}{S - \lfloor p \cdot S \rfloor})^k, & i > \lfloor p \cdot S \rfloor \end{cases}, \tag{B.10}$$

where $\lfloor \cdot \rfloor$ rounds a number down to the closest integer, and N normalizes the sum of $\{W_{\text{ith}}(p)\}_{i \in [0,S]}$ to 1.

The weight function (B.10) is plotted in Fig. B.3b for different values of p. Intuitively, the soft percentile approximates a certain percentile with a weighted sum of the gradient magnitudes. The larger the k, the closer M_{softperc} is to M_{perc}. The advantage of the soft percentile metric over the percentile metric is that it changes smoothly with the exposure time, which we will see next.

Table B.1 – Number of FAST features in the best image. The percentile-based metrics performs better in 13 out of 18 datasets.

Dataset	M_{sum}	M_{shim}	M_{perc}	M_{softperc}
office window1	272	272	**288**	**288**
office window2	23	23	**33**	**33**
building	**66**	61	62	62
office desk	336	336	**403**	391
office ceiling	**490**	429	456	456
keyboard	60	59	**84**	**84**
light	32	7	**34**	**34**
office door	55	53	**59**	**59**
home window1	67	**69**	67	67
home window2	**49**	46	45	46
corridor	**26**	**26**	**26**	**26**
clutter	**51**	50	**51**	49
shelf	78	**81**	80	78
lounge	81	50	**82**	**82**
garage	**66**	**66**	**66**	**66**
shady building	**100**	74	**100**	**100**
sunny building	**71**	**71**	**71**	**71**
grass	**81**	**81**	**81**	**81**

B.3.2 Evaluation

In order to understand the difference of the aforementioned metrics, we evaluate them on 18 real world datasets. Each of the datasets consists of a sequence of images of the same scene, captured with different exposure time settings. In the evaluation, we compute the gradient-based metrics for all the images and observe how different metrics change with the exposure time. For M_{softperc}, $k = 5$ is used. For both M_{perc} and M_{softperc}, the best image is chosen as the one that corresponds to the maximum value when $p = 0.5$ (as we will see, the best image does not change much with different values of p). To quantitatively measure the image quality, we compute the number of FAST features [223] that can be extracted from the best images.

The results are listed in Table. B.1. It can be observed that in 13 out of 18 datasets, the best images in terms of M_{perc} have the most features. In addition, in the datasets where another metric performs better, the numbers of the features in the best images of M_{perc} are actually very close to those of the best metric (*e.g.* in *home window1*, 67 features compared to 69 features of M_{shim}). In contrast, in some datasets, much less features can be extracted from the best images of other metrics (*e.g. keyboard* and *lounge* datasets). The performance of M_{softperc} is quite close to M_{perc}.

To give an intuition of the difference among the metrics, we show the results of the *keyboard* dataset in Fig. B.4. The scene mostly consists of two areas with very different

brightness, a black keyboard and a piece of white paper with text. It can be observed that both M_{sum} and M_{shim} increase with the exposure, and the best images according to these metrics are obviously overexposed in the bright area (*i.e.* the piece of paper). In contrast, the best images in terms of the percentile based metrics, M_{perc} and M_{softperc}, preserve the details in the bright area well.

There are two observations worthnoting regarding the percentile based metrics. While M_{perc} and M_{softperc} have quite similar performance in our evaluation, if the plots of M_{perc} and M_{softperc} in Fig. B.4 are closely compared, it can be seen that the curves corresponding to M_{softperc} are smoother. In addition, while the curves of different p values have similar maxima, the one corresponding to a higher p usually has larger derivative with respect to the exposure time. This is because, in an image, there are usually a large number of pixels with low gradient magnitudes under all exposure times (*e.g.* smooth area), which will make the percentiles with small p values change less significant. Both the smoothness and the derivatives are important for our optimization-based exposure control algorithms, which will be discussed in more details in Section B.4. Based on the above observations, we will use M_{softperc} and $p = 0.7 \sim 0.8$ in the rest of the work.

To summarize, in our evaluation, the percentile based metrics M_{perc} and M_{softperc} are more robust than M_{sum} and M_{shim}, and M_{softperc} with a large p has a more desirable behavior. In the next section, we will describe our exposure control method.

B.4 Exposure Control

With the photometric response function in Section B.2, we are able to predict how the image changes with the exposure time and, furthermore, we know how the metrics in Section B.3.1 change accordingly. Such information allows us to use standard optimization methods, such as gradient descent, for exposure control. Following this idea, in this section, we first derive the derivative of the soft percentile metric (B.9) with respect to the exposure time and then describe our exposure control method.

B.4.1 Derivative of the Gradient Magnitude

Because our metric is based on the image gradient magnitude, the first step is to calculate the derivative of the squared gradient magnitude $G(\cdot)$ with respect to the exposure time Δt. Taking the derivative of the right-hand side of (B.4), $\frac{\partial G(\cdot)}{\partial \Delta t}$ becomes

$$2\nabla \mathtt{I}(\mathbf{u}, \Delta t)^\top \frac{\partial}{\partial \Delta t}[\nabla \mathtt{I}(\mathbf{u}, \Delta t)]. \tag{B.11}$$

The first term of (B.11) is simply the gradient of the image, and the second term can be transformed by applying the Schwarz's theorem:

$$\frac{\partial}{\partial \Delta t}[\nabla \mathtt{I}(\mathbf{u}, \Delta t)] \quad = \quad \nabla[\frac{\partial}{\partial \Delta t}\mathtt{I}(\mathbf{u}, \Delta t)]. \tag{B.12}$$

Note that the derivative inside the right-hand side of (B.12) is actually the derivative of the photometric response function (B.1). Thus, for a pixel with the intensity I, the derivative can be calculated as

$$\frac{\partial I}{\partial \Delta t} \overset{(B.1)}{=} f'[f^{-1}(I)]E(\mathbf{u}) = \frac{E(\mathbf{u})}{[f^{-1}]'(I)} \overset{(B.2)}{=} \frac{1}{g'(I)\Delta t}, \tag{B.13}$$

where $E(\mathbf{u})$ is the exposure corresponding to the pixel. Finally, inserting (B.13) into (B.12) and then (B.12) into (B.11), the derivative of the gradient magnitude becomes

$$\frac{\partial G(\cdot)}{\partial \Delta t} \quad = \quad 2[\nabla \mathtt{I}(\cdot)]^\top \nabla \left[\frac{1}{g'(\mathtt{I}(\cdot))\Delta t}\right]. \tag{B.14}$$

Note that $g'(\mathtt{I}(\cdot))$ means applying g' to all pixels of $\mathtt{I}$.

B.4.2 Derivative of the Soft Percentile Metric

Because M_{softperc} is simply a weighted sum of all the gradient magnitudes in the image, its derivative is straightforward:

$$\frac{\partial M_{\text{softperc}}}{\partial \Delta t} = \sum_{i \in [0,S]} W_{\text{ith}} \frac{\partial G_{\text{ith}}}{\partial \Delta t} \tag{B.15}$$

Before proceeding to our exposure control method, we first validate our derivative formulation on a sequence recorded in an office environment. The sequence consists of images of different exposure settings of the same static scene. For each image, we calculate M_{softperc} (*i.e.* (B.10)) and then $\frac{\partial M_{\text{softperc}}}{\partial \Delta t}$ based on M_{softperc} of two consecutive images. The measured derivatives are then compared with the derivatives calculated from (B.15). For comparison, M_{perc} and its derivatives are also computed. For both M_{perc} and M_{softperc}, $p = 0.8$ is used.

The results are shown in Fig. B.5. It can be seen that the measured and predicted derivatives of M_{softperc} are close to each other. By contrast, the predicted derivatives of the M_{perc} show larger errors with respect to the measured one. This is another reason why M_{softperc} is preferred over M_{perc} in our method: the derivative of a percentile is difficult to estimate accurately. Instead of merely using the derivative from a single pixel

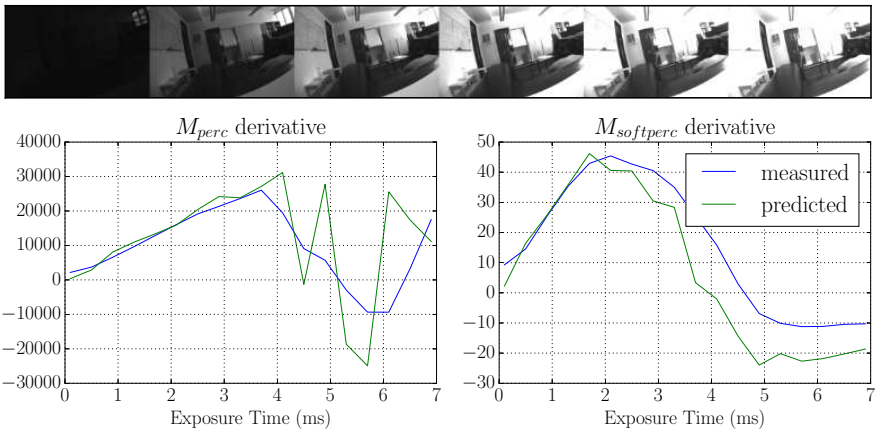


Figure B.5 – Validation of the metrics derivatives. The first row shows sample images from the office sequence; the second row shows the measured and predicted derivatives of M_{perc} and M_{softperc}.

(*i.e.* the pixel that has the percentile gradient magnitude), using the derivatives of all the pixels will result in a smoother and more accurate estimation.

B.4.3 Exposure Control Scheme

In Section B.3, we have shown that the soft percentile metric M_{softperc} is a robust indicator of the image quality. Therefore, the goal of our exposure control is to maximize M_{softperc} for future images. To achieve this goal, the exposure time is updated based on the latest image from the camera driver in a gradient ascent manner. In particular, given an image `I` and the corresponding exposure time Δt, the desired exposure time for the next image is calculated as:

$$\Delta t_{\text{next}} = \Delta t + \gamma \frac{\partial M_{\text{softperc}}}{\partial \Delta t}, \tag{B.16}$$

where the derivative of M_{softperc} is calculated by (B.15), and r is a design parameter to control the size of the update step. Then the new desired exposure time is sent to the camera driver and the update (B.16) is performed on the next image.

B.5 Exposure Compensation

Many VO algorithms, especially direct methods, assume that the brightness of the same part of the scene is constant over different frames. However, the change of exposure time breaks this assumption. In this section, we introduce the adaptations of two commonly used VO module—direct image alignment and direct feature matching—using both affine compensation [122, 82] and photometric compensation [152, 134, 82] and compare their performance experimentally.

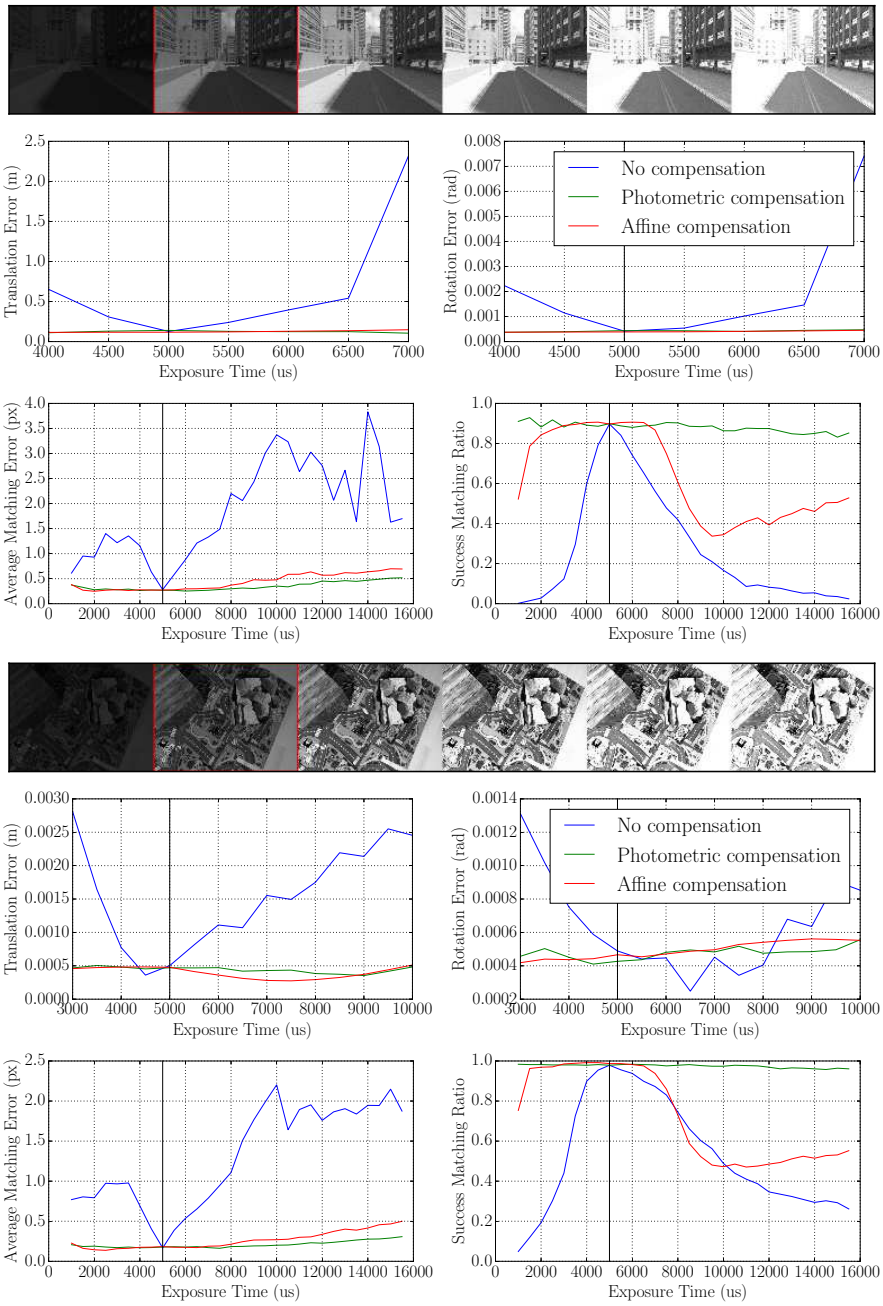


Figure B.6 – Evaluation of different exposure compensation methods on synthetic datasets. Top: urban canyon dataset. Bottom: room dataset. The first row shows the samples of the augmented dataset, where the red square indicates the original image. The second row shows the estimation error of the direct image alignment and, the third, the success matching ratio and matching errors of the direct feature matching.

B.5.1 Direct Image Alignment

Given a reference image $\mathtt{I}_r$ and a current image $\mathtt{I}_c$, the goal of the direct image alignment is to estimate the 6 DoF motion $\mathtt{T}_{rc}$ (*i.e.* the pose of $\mathtt{I}_c$ in the frame of $\mathtt{I}_r$). In $\mathtt{I}_r$,

there is a subset of pixels $S = \{\mathbf{u}_i\}$ with known depths $D = \{d_i\}$. Assuming brightness constancy, the direct image alignment estimates $\mathtt{T}_{rc}$ by minimizing the photometric error:

$$\mathtt{T}_{rc} = \arg\min_{\mathtt{T}} \sum_{\mathbf{u}_i \in S} [\mathtt{I}_r(\mathbf{u}_i) - \mathtt{I}_c(\mathbf{u}_i^c)]^2 \tag{B.17}$$

$$\mathbf{u}_i^c = \pi(\mathtt{T}^{-1}\pi^{-1}(\mathbf{u}_i, d_i)) \tag{B.18}$$

where $\mathbf{u}_i^c$ is the corresponding pixel of $\mathbf{u}_i$ in the current image, $\pi(\cdot)$ is the projection function that projects a 3D point into the image, and $\pi^{-1}(\mathbf{u}, d)$ backprojects a pixel $\mathbf{u}$ in the image to the corresponding 3D point, given the depth d.

When the brightness of the scene is not constant between $\mathtt{I}_r$ and $\mathtt{I}_c$, one can use an affine transformation to model the brightness change. In this case, the direct image alignment solves the optimization problem

$$\{\mathtt{T}_{rc}, \alpha_{rc}, \beta_{rc}\} = \arg\min_{\mathtt{T},\alpha,\beta} \sum_{\mathbf{u}_i \in S} [\alpha \mathtt{I}_r(\mathbf{u}_i) + \beta - \mathtt{I}_c(\mathbf{u}_i^c)]^2. \tag{B.19}$$

Alternatively, if the brightness change is caused by the variation of the exposure time, we can also incorporate the photometric response function (B.1) into the optimization problem:

$$\mathtt{T}_{rc} = \arg\min_{\mathtt{T}} \sum_{\mathbf{u}_i \in S} [f(\frac{\Delta t_c}{\Delta t_r} f^{-1}(\mathtt{I}_r(\mathbf{u}_i))) - \mathtt{I}_c(\mathbf{u}_i^c)]^2, \tag{B.20}$$

where Δt_r and Δt_c are the exposure times of $\mathtt{I}_r$ and $\mathtt{I}_c$ respectively. The optimization problems (B.17), (B.19) and (B.20) can be solved by nonlinear least-square optimization methods such as Gauss-Newton.

B.5.2 Direct Feature Matching

Direct feature matching aims to estimate the 2D position of a feature in an image $\mathtt{I}$, given an initial feature position $\mathbf{u}'$ and a reference template P of the feature. The estimation can be done by minimizing the photometric error:

$$\arg\min_{\delta\mathbf{u}} \sum_{\Delta\mathbf{u} \in P} [P(\Delta\mathbf{u}) - \mathtt{I}(\mathbf{u}' + \delta\mathbf{u} + \Delta\mathbf{u})]^2. \tag{B.21}$$

where $\Delta\mathbf{u}$ iterates inside the template P. The final estimation of the feature position is $\mathbf{u}' + \delta\mathbf{u}$. If an affine transformation is used to model the brightness change, (B.21)

becomes

$$\arg\min_{\delta\mathbf{u},\alpha,\beta} \sum_{\Delta\mathbf{u}\in P} [\alpha P(\Delta\mathbf{u}) + \beta - \mathtt{I}(\mathbf{u}' + \delta\mathbf{u} + \Delta\mathbf{u})]^2. \tag{B.22}$$

Similar to (B.20), we can also use the photometric response function in the direct feature matching:

$$\arg\min_{\delta\mathbf{u}} \sum_{\Delta\mathbf{u}\in P} [f(\frac{\Delta t_c}{\Delta t_r} f^{-1}(P(\Delta\mathbf{u})) - \mathtt{I}(\mathbf{u}' + \delta\mathbf{u} + \Delta\mathbf{u})]^2 \tag{B.23}$$

where Δt_r is the exposure time with which the reference template is captured. Direct feature matching (B.21), (B.22) and (B.23) can be solved using the Lucas-Kanade algorithm [13].

B.5.3 Evaluation

In the following, we evaluate the performance of the direct image alignment and the direct feature matching with both the exposure compensation methods (*i.e.* the affine compensation Eq. (B.19), (B.22) and the photometric compensation Eq. (B.20), (B.23)) on synthetic and real world datasets.

For synthetic evaluation, we use the Multi-FoV dataset [318], which contains images from two virtual scenes (urban canyon and room) with groundtruth poses and depth maps of the images. Because the dataset is rendered with a constant exposure time, we first augment the dataset using the photometric response function from a real camera (*e.g.* Fig. B.1). In particular, we assume the exposure time of the original dataset to be a certain value, then calculate the irradiance of the scene and use the irradiance to generate images of different exposure times. Therefore, in the augmented dataset, for each *frame*, we have several *images* of different exposure times. Note that the same photometric response function is used in the photometric compensation afterwards (*i.e.* (B.20) and (B.23)).

To evaluate the direct image alignment, we randomly select two consecutive frames from the augmented dataset and estimate the relative transformation between the two frames. We fix one image from the first frame and use several images of different exposure times from the second frame. In our evaluation, the pixels in the small patches around the features extracted from the first image are used. The depth values of the pixels are from the ground truth depth map, and the initial pose is generated by adding a small disturbance to the ground truth pose. We measure the performance of the alignment by calculating the translation and rotation error compared to the ground truth.

The results of the direct image alignment experiment are shown in the second row of Fig. B.6. It can be observed that the estimation errors of both exposure compensation

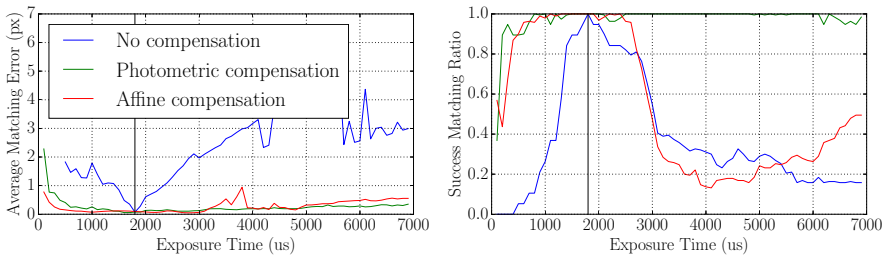


Figure B.7 – Evaluation of the direct feature matching in an office environment. See Fig. B.5 for image samples.

methods are smaller than the situation where no compensation is applied. Interestingly, the performance of the affine compensation is similar to the photometric compensation. Note that in this experiment the response function used in the photometric compensation is perfect, in that the dataset is generated using the same function. It can be expected that on real datasets, the affine compensation will perform at least as good as the photometric compensation.

For the evaluation of the direct feature matching, we first select a random frame from the dataset and extract several FAST features from one arbitrary reference image of the frame. Then we try to match these features in all the images of the same frame. The reference templates of the features are taken from the reference image, and we add noise to the positions where the features are extracted to get the initial positions for the direct feature alignment. The success matching ratio and the final matching errors are used as performance metrics.

The results of the direct feature alignment experiment are shown in the last row of Fig. B.6. Obviously, both exposure compensation methods improve the performance of the direct feature matching. Differently from the results of the direct image alignment, the photometric compensation has better performance than the affine compensation. In order to take into consideration the inaccuracy of the response function, we further evaluate the direct feature matching on the real sequence we used in Section B.4.2. The results are similar, as shown in Fig. B.7.

In summary, both exposure compensation methods improve the performance of the direct image alignment and the direct feature matching. Regarding the comparison between these two methods, the affine compensation performs as good as the photometric compensation in the direct image alignment, even if the latter uses a perfect photometric response function; in the direct feature matching, however, using the photometric compensation can achieve more success matches and a better matching accuracy than the affine compensation for both synthetic and real world datasets.

B.6 Experiments

In the following, we first validate our exposure method in indoor and outdoor environments. Then, we show the performance of an active VO with exposure control and compensation in real-world HDR environments.

B.6.1 Implementation Details

The Selection of γ

The only parameter of our algorithm is the gradient ascent rate γ. Intuitively, a large γ will make the exposure control more responsive but tend to overshoot, and a small one will have a smoother but slower behavior. By thorough outdoor and indoor experiments, we found that in general a small γ should be used for high irradiance (*e.g.* sunlit outdoor environment) and a large value for low irradiance (*e.g.* indoor environment). Therefore, we use a lookup table that maps the irradiance to γ and adjust γ at every frame. The values of the lookup table are determined experimentally.

Automatic Gain

In addition to the exposure time, we find it also necessary to adjust the gain of the camera. First, in extreme bright or dark scenes, it may happen that even when the camera is at its maximum/minimum exposure time, the image is still not well exposed. In such situations, the gain also needs to be adjusted. Second, the exposure time also puts a limit on the frame rate. For example, if the exposure time is too high, we can only have a low frame rate, which means that the frequency at which we can adjust the exposure time is also limited.

With the gain, denoted as g, the photometric response function (B.1) becomes

$$I = f(X) = f(gE\Delta t). \tag{B.24}$$

Obviously, to keep the image intensities constant, the exposure time should decrease/increase with same change ratio as the gain increases/descreases. In practice, we use a heuristic policy: if the exposure time is above a certain threshold, we increase the gain and decrease the exposure time accordingly, and vice versa.

Handling Overexposed/Underexposed Pixels

One major limitation of our method is that it exploits the gradient in the image; therefore, overexposed/underexposed pixels actually provide no information for our algorithm (*i.e.* the gradient and its derivative is in fact zero). If, for example, the image is totally

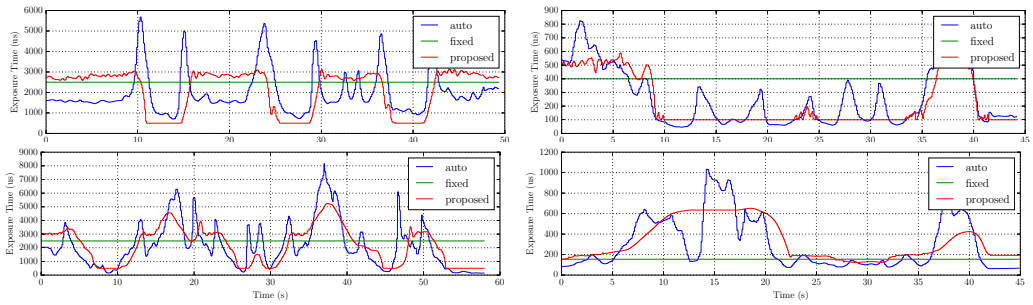


Figure B.8 – Comparison of our exposure control method with the built-in auto-exposure of the camera and a fixed exposure time in both indoor and outdoor environments.

overexposed, there is no gradient information that can be used, and then the algorithm will not adjust the exposure time at all, which is obviously not the desired behavior. We mitigate this drawback with a simple heuristic: we assign small negative derivatives (*e.g.*-2.0) to overexposed pixels and positive derivatives (*e.g.* 2.0) to underexposed ones, which forces the algorithm to react correctly to both overexposed and underexposed pixels.

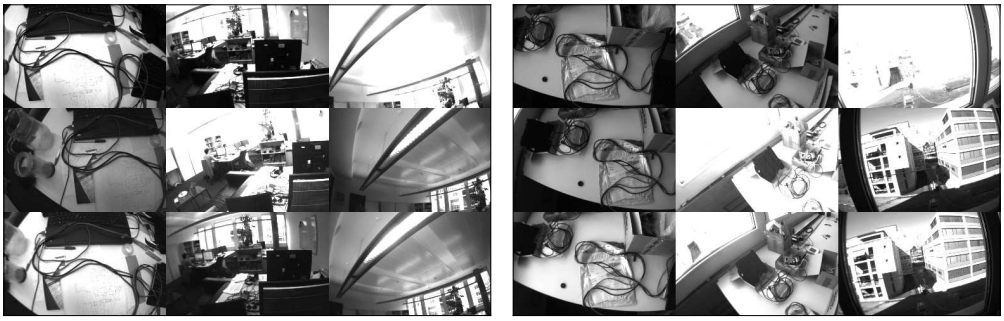
B.6.2 Exposure Control

To compare our method with different camera settings, we mounted three MatrixVision Bluefox monochrome cameras in parallel on a camera rig. Each of the camera has a resolution of 752×480 pixels. The three cameras used the built-in auto-exposure algorithm, a fixed exposure time, and our exposure control algorithm, respectively. We then moved the rig in different environments and recorded the exposure time history for all the cameras.

We ran tests in 12 indoor and outdoor HDR environments (*e.g.* buildings under direct sunshine and shadowed areas). The fixed exposure time was hand-tuned at the start point of each test. In most of the tests, our exposure control method was able to adjust the exposure time successfully without obvious overshooting. The exposure time history in several sample sequences is shown in Fig. B.8. It can observed that the exposure time variation of the built-in auto-exposure and that of our method have a similar trend. However, our method is more stable. During the test, we often observed that the exposure time of the auto-exposure could change significantly when the position of the camera changed very little (*e.g.* the peaks in the top-left plot of Fig. B.8).

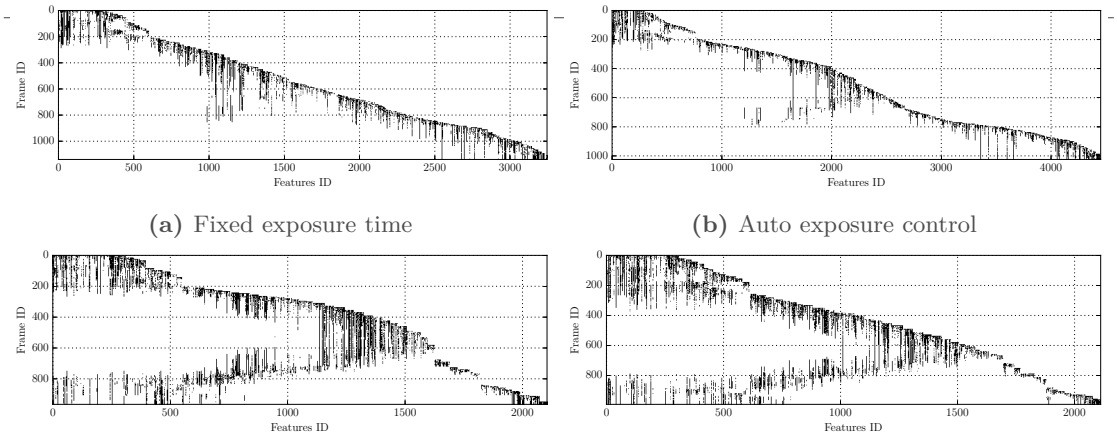
B.6.3 Active Visual Odometry

To show that combining exposure control and exposure compensation can improve the performance of VO algorithms in HDR environments, we implemented the exposure compensation methods of Section B.5 into SVO [91]. Then we tested the adapted SVO in 10 scenes. The sequences were collected using the same three-camera setup as the



(a) Office light sequence. **(b)** Window sequence.

Figure B.9 – Real sequences in HDR environments to test VO. First row: fixed exposure time; Second row: auto-exposure; third row: our method.



(a) Fixed exposure time **(b)** Auto exposure control

(c) Auto exposure control with compensation **(d)** Our exposure control with compensation

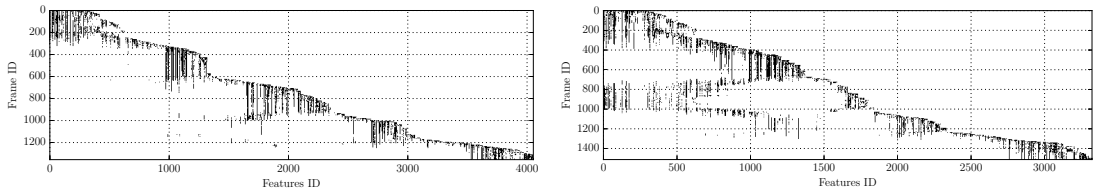
Figure B.10 – Feature tracks in the office light sequence.

previous experiment. To better show the influence of exposure control and exposure compensation separately, we tested the following configurations:

- fixed exposure time + no exposure compensation
- auto-exposure + no exposure compensation
- auto-exposure + exposure compensation
- our exposure control + exposure compensation

Based on our results on several sequences, with exposure control and compensation, the robustness and accuracy of our VO algorithm is improved. Moreover, our exposure control algorithm performs better than the auto-exposure. In the following, the results from two representative sequences are discussed in detail.

First, we show the result from a sequence in an office environment. In the sequence, the camera was first pointed toward the desk, then moved to look at the office light and



(a) Auto exposure control with compensation **(b)** Our exposure control with compensation

Figure B.11 – Feature tracks in the window sequence.

lastly moved back to the initial position. Samples of the sequence are shown in Fig. B.9a. To analyze the behavior of our VO algorithm in detail, we recorded the features that were tracked in each frame of the sequence. The feature tracks (frame ID vs. feature ID) are shown in Fig. B.10. A dot of coordinates (x, y) in each of these plots means that feature x was tracked in frame y. A continuous vertical line indicates a feature that was persistently tracked, while a non continuous line means that the same feature was lost and then re-detected and tracked again.

In this sequence, the adapted SVO correctly tracked the pose without loosing tracking with all the four test configurations. Comparing the configurations with exposure compensation (Fig. B.10c and Fig. B.10d) against the ones without (Fig. B.10a and Fig. B.10b), we can observe that the first two configurations present increased tracking robustness during viewpoint changes; indeed, features can be tracked longer and get more frequently re-detected. On the one hand, with a fixed exposure time, the image was badly overexposed when switching from the desk to the office light; on the other hand, when using auto-exposure without exposure compensation, the VO could not track the features well with the changing brightness.

The bad tracking in the middle also have an impact when the camera moved back to its initial position. In Fig. B.10c and Fig. B.10d, the VO was able to track some old features at last (*i.e.* the top-left area and the bottom-left area indicate the same features were tracked by both the first frames and the last frames). Obviously, this is not the case in Fig. B.10a and Fig. B.10b. The reason is that the aforementioned bad tracking resulted in too much drift in the last frames to correctly project the old feaures into these frames.

The result of a second test sequence is shown in Fig. B.11. In this sequence, the camera was first pointed toward a desk near a window, then moved to look at the building outside the window and lastly moved to the initial position. Samples of the sequence are shown in Fig. B.9b. Note that because the building was under direct sunlight at the time of recording, this sequence is more difficult than the first one. Only the configurations with exposure control were able to finish the whole sequence. Similar to our analysis of the first sequence, it can be observed that the tracking quality with our exposure control method is better than the auto-exposure.

B.7 Conclusions and Future Work

In this work, we proposed an active exposure control method to tackle this problem. We first proposed a gradient-based image quality metric and showed its robustness on various real world datasets. Then we designed a novel exposure control method, by exploiting the photometric response function of the camera, to maximize our image quality metric. We showed that our exposure control method outperforms the built-in auto-exposure of the camera in both indoor and outdoor environments. To integrate our exposure control method with VO, we introduced the adaptations for exposure compensation to a state-of-the-art algorithm. We also experimentally compared two different exposure compensation methods and demonstrated that we can improve the robustness of VO by combining active exposure control and compensation in challenging real-world environments.

Future work would include modeling the effect of motion blur by exploiting the information from VO. Also we would like to explore the possibility to analyze the impact of the exposure time on the accuracy of VO directly.

C Incorporating Fisher Information in Visual Navigation

Reprinted, with permission, from:

Z. Zhang and D. Scaramuzza. "Perception-aware Receding Horizon Navigation for MAVs". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 2534–2541. DOI: 10.1109/ICRA.2018.8461133 [323]

Perception-aware Receding Horizon Navigation for MAVs

Zichao Zhang, Davide Scaramuzza

Abstract — To reach a given destination safely and accurately, a micro aerial vehicle needs to be able to avoid obstacles and minimize its state estimation uncertainty at the same time. To achieve this goal, we propose a perception-aware receding horizon approach. In our method, a single forward-looking camera is used for state estimation and mapping. Using the information from the monocular state estimation and mapping system, we generate a library of candidate trajectories and evaluate them in terms of perception quality, collision probability, and distance to the goal. The best trajectory to execute is then selected as the one that maximizes a reward function based on these three metrics. To the best of our knowledge, this is the first work that integrates active vision within a receding horizon navigation framework for a goal reaching task. We demonstrate by simulation and real-world experiments on an actual quadrotor that our *active* approach leads to improved state estimation accuracy in a goal-reaching task when compared to a purely-reactive navigation system, especially in difficult scenes (*e.g.* weak texture).

Multimedia Material

A video explaining the idea and showing the results is available at https://youtu.be/FK6S_CRXiuI.

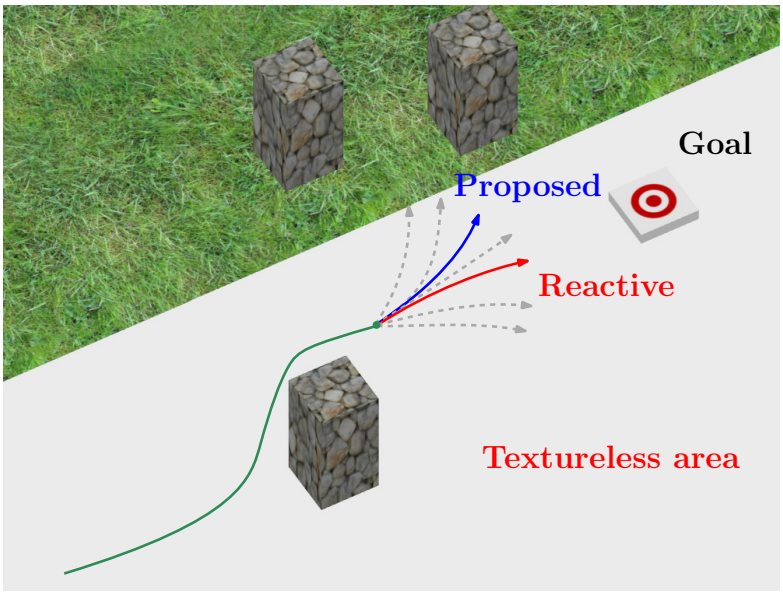


Figure C.1 – Illustration of the proposed perception-aware receding horizon navigation system. Our method is able to select a suitable motion (blue) that can simultaneously avoid obstacles, reach a given destination and minimize state estimation uncertainty. By contrast, a purely-reactive navigation scheme (red) can enter textureless area, resulting in large state estimation error and the failure to reach the given destination.

C.1 Introduction

Being both agile and versatile, micro aerial vehicles (MAVs) are suitable for various tasks such as industrial inspection, agriculture, and goods delivery. To enable MAVs to operate autonomously in an unknown environment, reliable on-board state estimation is necessary. Among different sensors for state estimation, cameras are lightweight and power efficient and, therefore, ideal for MAVs due to their limited payload and battery life.

For vision-based state estimation, it is well-known that the motion of a camera has a significant impact on the estimation accuracy [65]. Therefore, the motion of MAVs should be planned considering both the task at hand and the perception quality.

In this work, we focus on the task of reaching a given goal with the highest accuracy while avoiding obstacles in the environment. When it comes to planning a trajectory in partially-unknown environments, the common approach is to couple a global planner with a receding-horizon method: while the global trajectory is being executed, a local planner is used to generate and search feasible, collision-free trajectories in a local robot-centric map of the environment. While this approach has been successfully implemented in several recent works [52, 169, 144, 157, 180], these do not take perception constraints into account (e.g., favour texture rich areas to minimize state estimation uncertainty).

Very little work has been done to consider the *quality of perception* and its influence on the accuracy of visual odometry in a receding horizon fashion. The problem of choosing the motion that maximizes the accuracy of state estimation is known as *active SLAM* [65]. This problem is often solved by optimizing the motion trajectory in a global map (see, *e.g.* [225]), which is usually expensive to compute. Instead, we propose to solve the active SLAM problem in a receding horizon fashion. In particular, we limit the scope of the problem to a local map and a short horizon and continuously recompute a suitable local trajectory for state estimation. This way, the active SLAM problem can be solved efficiently and can be integrated naturally into a receding horizon setting.

To demonstrate the usefulness of the idea, we implement a *perception-aware receding horizon navigation* system. Specifically, we use a monocular odometry and mapping system for state estimation and mapping. Instead of directly optimizing the motion parameters, we use an efficient trajectory generation method (minimum jerk [183]) to generate a library of candidate trajectories within a short horizon. For each of the trajectories, we evaluate its perception quality, the probability of collision, and its distance to the goal using the information from the monocular odometry and mapping system. The trajectory to execute is then selected as the one that maximizes a reward function based on these three metrics. The trajectory generation and evaluation process is repeated online.

C.1.1 Related Work

Receding Horizon Planning for MAVs

Receding horizon planning has been widely used to generate collision-free trajectories online. Liu *et al.* [157] developed a framework of receding horizon planning, which continuously plans trajectories within a safe flight corridor. Chen *et al.* [52] also optimized trajectories by using a similar corridor representation for the free space. Differently, Landry *et al.* [144] represented the free space as convex volumes, and enforced each trajectory segment to stay in the volumes to avoid collision. Mohta *et al.* [180] first planned a safe path consisting of straight line segments and then optimized safe trajectories within a finite horizon by enforcing the trajectories to be close to the safe path.

Instead of resorting to optimization, sampling-based methods have also been studied. Dey *et al.* [70] first showed an implementation of receding horizon control with trajectory libraries using monocular sensing on MAVs. Florence *et al.* [87] generated a library of candidate trajectories in a limited time horizon and selected the one to execute based on a cost function including the collision risk and the distance to a given destination. Matthies *et al.* [169] exploited a rapid-exploring random tree (RRT) planner and performed collision checking by projecting the sampled trajectories into the disparity space.

The aforementioned methods are mainly designed for collision free motion planning

in a receding horizon fashion. None of them, however, took perception quality into consideration, which, as we later show, is extremely important in environments containing visually degraded areas (i.e., poor texture).

Active SLAM for MAVs

Different active SLAM approaches for MAVs have been proposed. Mostegel *et al.* [181] designed a set of heuristic metrics for evaluating localization quality and map generation likelihood. Depending the current state estimation quality, their control scheme decides whether to maximize the localization quality or map generation likelihood. Sabdat *et al.* [229] proposed a perception quality metric that combines the number of visible features and the viewing angle with respect to visible surfaces and incorporated this metric into Rapidly-Exploring Random Tree (RRT*) [126] for planning. Alzugaray *et al.* [6] first sampled positions near obstacles based on the intuition that pose estimation error is small when the camera is close to the features on obstacles. Then path planning was carried out based on the sampled positions. Different from previous methods, which focus on the geometric information, Costante *et al.* [**Costante2016arxiv**] additionally incorporated the photometric information of the scene to calculate the localization uncertainty of the camera. Some recent work also considered different sensor configurations, such as the global positioning system (GPS) [109] and inertial measurements units (IMU) [206]. As a complementary problem, researchers have also proposed methods to minimize reconstruction uncertainty [90] [121].

While the aforementioned methods maintain a global map for planning, some recent work focuses on planning based on short term accuracy. Rong *et al.* [221] evaluated short-term perception quality using empirical observability gramian. They demonstrated their metric can successfully reflect the change of perception quality, *e.g.* in visually degraded areas. Papachristos *et al.* [202] proposed a two-step planning strategy for perception-aware exploration. They first planned a view that maximizes the information gain in terms of explored space, then a second planner sampled views locally and selected the one with the least state estimation uncertainty. The proposed method is similar to [202] in that we also evaluate the perception quality of candidate motion within a limited time horizon. The differences are twofold. First, we focus on the task of reaching a given destination instead of exploration. Second, in their method, the obstacle avoidance is done by planning in free space, while in our method it is considered in a unified receding horizon navigation framework.

C.1.2 Contributions and Outline

To the best of our knowledge, this is the first work that integrates active vision in a receding horizon navigation framework for a goal reaching task. We demonstrate by simulation and real-world experiments that our *active* approach leads to improved state

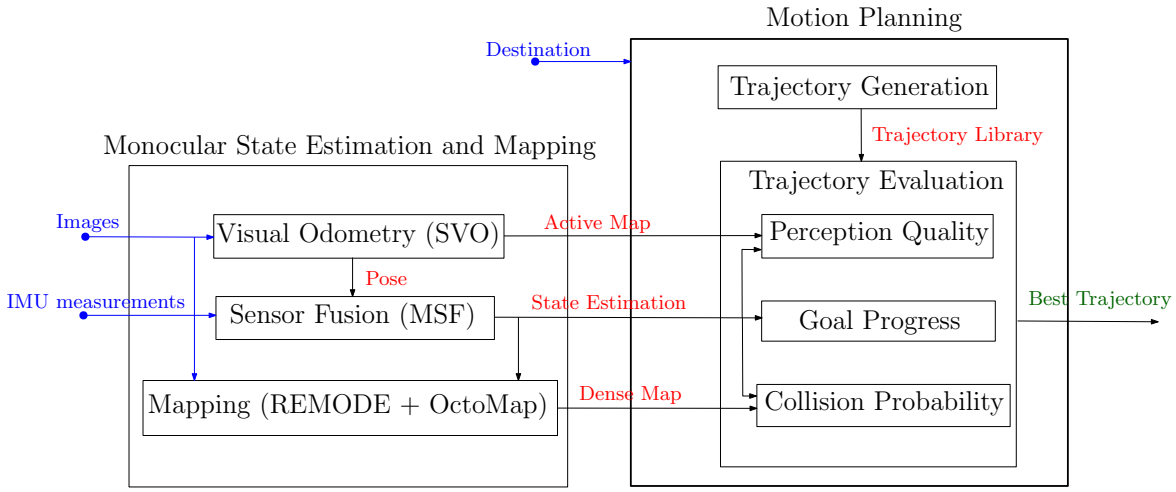


Figure C.2 – Overview of the proposed system. The input of the system is marked as blue, communication among different modules red and the output green.

estimation accuracy when compared to a purely-reactive navigation system, especially in difficult scenes (*e.g.* weak texture).

The rest of the paper is structured as follows. In Section C.2, we give an overview of our perception-aware receding horizon navigation system. In Section C.3, we describe our monocular state estimation and mapping system. Then in Section C.4, we detail how to plan the motion of a quadrotor using the information from the system in Section C.3. To demonstrate the effectiveness of the proposed method, we show simulation and real-world experiments in Section C.5.

C.2 System Overview

Fig. C.2 gives an overview of the proposed system, which consists of a monocular state estimation and mapping system and a motion planning system.

The monocular state estimation and mapping system is responsible to provide the state estimation of the MAV and the map for different purposes. We first use SVO [92] to estimate the 6-Degrees-of-Freedom (DOF) pose of the camera. The pose estimation is further fused with the IMU measurement using the Multi-Sensor Fusion (MSF) software [162] to get the correct scale and the extra velocity estimation, which is necessary for trajectory generation and control. Then, the state estimate and the images are fed into a variant of REMODE [204] to get a dense 3D map of the frontal view. Before using the dense map for motion planning, we utilize OctoMap [112] to reduce the noise in the map. In addition to the state estimation and the dense map, we also get an *active map* from SVO, which will be detailed in Section C.3.

The motion planning system consists of a trajectory generation module and a trajectory evaluation module. We use an efficient trajectory generation method [183] to generate a library of candidate trajectories based on the current state estimate. We then evaluate each trajectory based on three metrics:

- the *collision probability*, based on dense 3D map from REMODE and Octomap (Section C.4.3),
- the *perception quality*, based on the *active map* from SVO (Section C.4.4),
- and the *goal progress* (which is a function of the distance to the goal), based on the current state estimate and given goal (Section C.4.5).

Based on the evaluation, we select the best trajectory to execute and send the desired state to the controller.

C.3 Monocular State Estimation and Mapping

Our monocular state estimation and mapping is similar to the one proposed in [89]. We use SVO plus MSF for state estimation, and REMODE to generate a dense pointcloud for obstacle avoidance. SVO is an extremely efficient VO algorithm that is suitable for resource-constrained systems, such as MAVs. REMODE, on the other hand, is originally designed to run on a Graphic Processing Unit (GPU). In [89], the authors proposed several modifications to enable REMODE to execute on a smartphone processor. We refer the reader to [89] for more details.

Compared to [89], our system is different in the following aspects. First, to evaluate the perception quality, we also extract an *active map* from SVO. Internally SVO maintains a set of sparse points, which can be divided into two categories: *landmarks* and *seeds*. Landmarks are 3D points that have been observed multiple times from different frames, and their positions are already well estimated. In contrast, seeds are 3D points, whose positions are not accurately estimated yet.

Intuitively, it is the landmarks that contribute the most to the accuracy of the pose estimate of a frame. Therefore, we extract the visible landmarks from keyframes that overlap with the current frame. We denote these landmarks as the *active map* (cf. Fig. C.3). Later, we will show how to use the active map to evaluate the *perception quality* (Section C.4.4). Second, we use Octomap to further reduce the noise in the output of REMODE. If the dense pointcloud contains too many outliers, the trajectory evaluation module will wrongly estimate the collision probability, resulting in unnecessary collision avoidance maneuvers. Fig. C.4 shows an example of the active map and dense map, which will be denoted as M_A and M_D, respectively, in the rest of the paper.

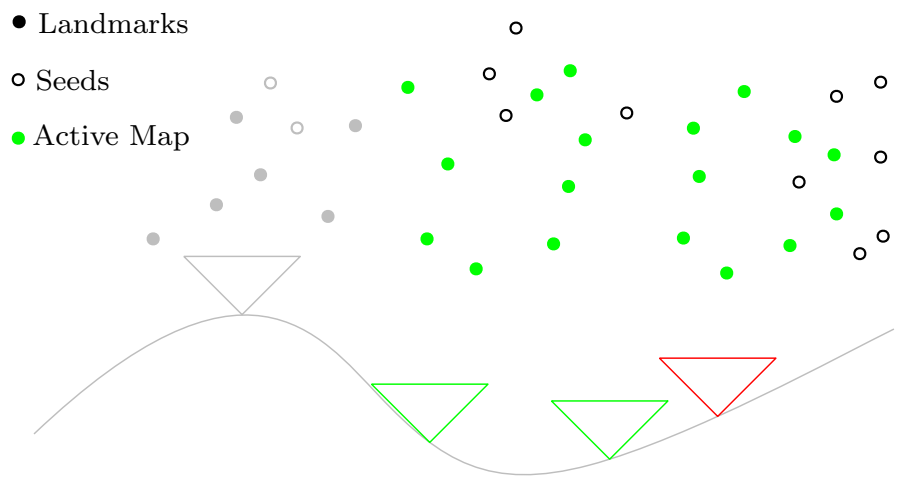


Figure C.3 – The extraction of the *active map*. For the current frame (red triangle), we find the keyframes that have overlap with it (green triangles). Then the landmarks that are visible in these keyframes are extracted as the active map (green solid dots). Keyframes that have no overlap (gray triangles) and seeds (circles) are not considered in the active map.

C.4 Trajectory Generation and Evaluation

In this section, we describe how we generate a library of candidate trajectories and select the best one to execute.

C.4.1 Notations

A pre-subscript denotes the frame where the quantity is expressed. We use $\mathtt{T}_{ab} \in SE(3)$ to represent the rigid body transformation of frame b in frame a, which transforms a point in frame b to frame a as ${}_a\mathbf{p} = \mathtt{T}_{ab} \cdot {}_b\mathbf{p}$. If a quantity/transformation is expressed in the world frame, we omit the (pre-)subscript for simplicity.

C.4.2 Trajectory Generation

We use [183] to generate our candidate trajectories. Conceptually, the trajectory generation process, denoted as $g(\cdot)$, is

$$f(t) \leftarrow g(\mathbf{p}_0, \mathbf{v}_0, \mathbf{p}_f, t_f). \tag{C.1}$$

$\mathbf{p}_0$ and $\mathbf{v}_0$ are the initial position and velocity of the trajectory in the world frame, which come from the current state estimation. $\mathbf{p}_f$ is the desired end point of the trajectory, and t_f is the time it takes for the quadrotor to follow the trajectory and reach $\mathbf{p}_f$. The output is a function $f(t)$. For $t \in [0, t_f]$, $f(t)$ gives the state (position, orientation, velocity) on the trajectory at time t.

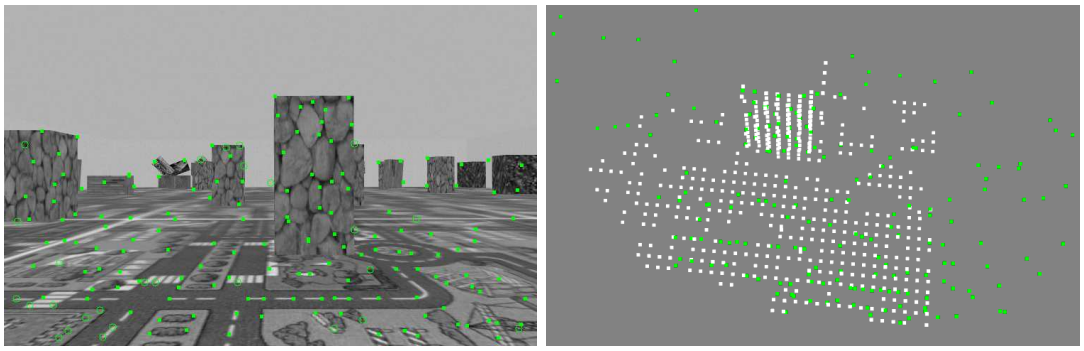


Figure C.4 – An example of the *active map* and *dense map*. The left one is the image from the camera, where the solid green dots are *landmarks* and the circles are *seeds*. The right one shows the corresponding active map (green) and the dense map (white).

Now we detail how to select $\mathbf{p}_f$ and t_f for generating a library of trajectories. We plan the trajectories by selecting the end points on an arc in front of the quadrotor, since a forward-looking camera is used and we want move in visible directions. To this end, we need to know the radius l and angle θ of the arc, as illustrated in Fig. C.5. While l is simply a design parameter, θ is calculated as

$$\theta = \max(k_\theta \|\mathbf{v}_0\|, \theta_{\max}), \tag{C.2}$$

where k_θ is a constant. Intuitively, θ increases with the velocity until a maximum value $\theta_{\max}$. Using θ and l, we uniformly sample the end points on the arc.

As for t_f, we use the following formula:

$$t_f = \min(\frac{l}{\|\mathbf{v}_0\| + \Delta v}, \frac{l}{v_{\max}}), \tag{C.3}$$

where Δv in a constant value, and $v_{\max}$ is the maximum velocity allowed. (C.3) means that we want to increase the velocity until the maximum value is reached.

After generating N candidate trajectories, we sample J poses from each trajectory by a constant time interval. Finally, we have a set of sampled poses for each candidate trajectory, which can be formulated as

$$C = \{c_i\}_{i=1}^{N}, \tag{C.4}$$

$$c_i = \{\mathtt{T}_j\}_{j=1}^{J}, \tag{C.5}$$

where $\mathtt{T}_j \in SE(3)$ is the jth pose on the trajectory expressed in the world frame. Since the yaw of the quadrotor is not constrained, we simply set the yaw to be the same as the

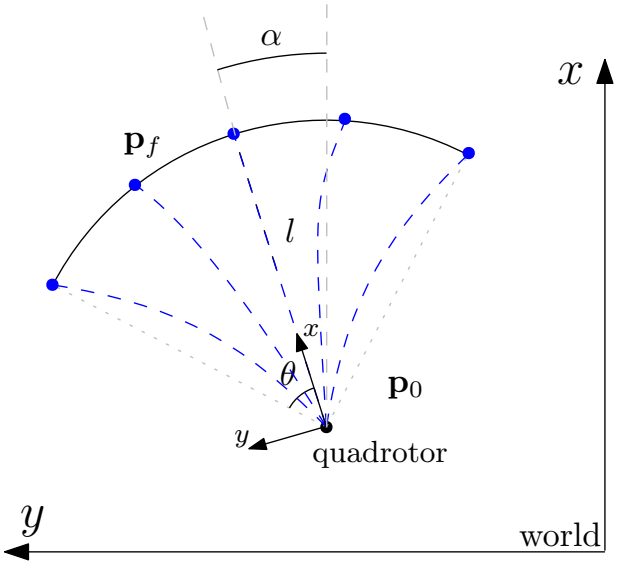


Figure C.5 – Trajectory generation. We uniformly sample N points on an arc, defined by θ and l, in front of the quadrotor. The candidate trajectories are shown in blue. α is the yaw angle of the quadrotor.

direction of velocity to facilitate obstacle avoidance (*i.e.* the frontlooking camera will always look in the moving direction).

Next, we need to select the best trajectory c_{best} for the given task. We choose the following criterion:

$$c_{best} = \arg\max_{c \in C} \ (1 - p_{col})(R_{\text{perc}} + R_{\text{goal}}) + p_{\text{col}} R_{\text{col}}, \tag{C.6}$$

where p_{col} is the probability to collide with obstacles (*i.e.* the dense map). R_{col}, R_{perc} and R_{goal} are the rewards related to collision risk, perception quality and goal progress respectively, which we will describe now. For simplicity, we will drop the subscript in (C.5) and refer to the trajectory to be evaluated as c.

C.4.3 Collision Probability

We use a similar method to calculate the probability of collision as [87], which is illustrated in the Fig. C.6. In particular, for each sampled position $\mathbf{p}_j$, we find the nearest point in the dense map M_D:

$$\mathbf{d}_j = \arg\min_{\mathbf{d} \in M_D} \|\mathbf{d} - \mathbf{p}_j\| \tag{C.7}$$

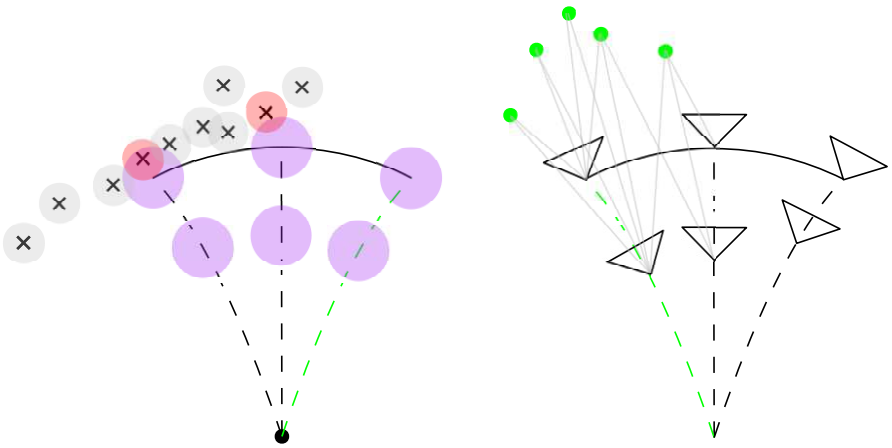


Figure C.6 – The evaluation of collision probability (left) and perception quality (right). Left: Crosses are points from the dense map, and dashed curves are candidate trajectories. To calculate the collision probability of a certain position on a trajectory, we find the nearest point (crosses with red circles) in the dense map and calculate the collision probability using a multivariate Gaussian distribution. Right: Solid green dots are the landmarks from the active map. For all the sampled poses along a trajectory, we collect the visible landmarks and construct the information matrix to evaluate the perception quality. The best trajectories in both cases are colored as green.

Then we calculate the collision probability of $\mathbf{p}_j$ using a multivariate Guassian distribution

$$p_j = V \times \frac{1}{\sqrt{2\pi\Sigma}} \exp[-\frac{1}{2}(\mathbf{d}_j - \mathbf{p}_j)^\top \Sigma^{-1}(\mathbf{d}_j - \mathbf{p}_j)], \tag{C.8}$$

where V is the volume of the safety sphere we want to keep around the quadrotor. In (C.8), $\Sigma = (\sigma_d + \sigma_p)\mathrm{I}_{3\times3}$, where σ_d and σ_p are the uncertainty of the map point and the position on the trajectory respectively. Then the collision probability for the trajectory c is

$$p_{\text{col}} = 1 - \prod_{j=1}^{K}(1 - p_j), \tag{C.9}$$

and the collision reward is

$$R_{\text{col}} = k_{\text{col}}, \tag{C.10}$$

where k_{col} is a negative constant indicating how much we need to penalize collision risk.

When the above method is used to calculate the collision probability, one drawback, for example, is that when the trajectory reaches beyond a planar obstacle, the collision probability of the positions behind the obstacle will be small, which is not realistic. Strictly speaking, we have no information about the space behind the obstacle. However, it is often true that the space behind is also occupied. Therefore, different from (C.8)

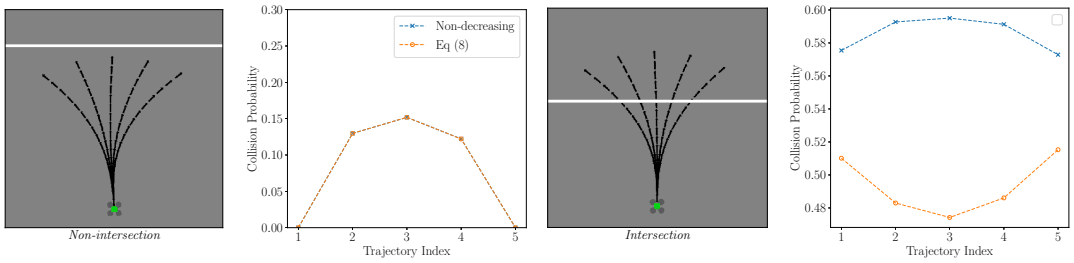


Figure C.7 – The calculation of collision probability. The left column shows the simulated trajectories (black arrow) and obstacles (horizontal white band), and the right shows the collision probabilities calculated using different methods. The trajectory index increases from left to right.

used by [87], we calculate the collision probability of $\mathbf{p}_j$ as

$$p_j = \begin{cases} \text{eq (C.8)} & p_j \geq p_{j-1} \\ p_{j-1} & p_j < p_{j-1} \end{cases} \tag{C.11}$$

Basically speaking, we assume the collision probabilities for positions along the trajectory is non-decreasing. To demonstrate the effect of (C.11), we compare the collision probabilities using both methods when a quadrotor approaches a planar obstacle, as shown in Fig. C.7. In the first row, we can see that when the candidate trajectories do not intersect with the obstacle, (C.8) and (C.11) have the same result. When there is intersection, as shown in the second row, the result from (C.8) indicates that the central trajectory is less likely to collide with the obstacle. By contrast, our method assigns a higher collision probability to the central trajectories. One may argue that both methods are not correct anyway because no information is known about the space behind the obstacle. However, when the calculated collision probabilities are used to select the best trajectory, our method is still advantageous by preferring the trajectories away from the center.

C.4.4 Perception Quality

Given a trajectory c and the active map $M_A = \{\mathbf{l}_k\}_{k=1}^K$, we need to quantify the pose estimation error if the quadrotor follows the trajectory. The smaller the pose estimation error is, the larger the R_{perc} in (C.6) will be. To this end, we first simulate the observations for the sampled poses $\{\mathtt{T}_j\}_{j=1}^J$ and then construct a least squares problem and evaluate the estimation error from the information matrix of the least squares problem.

For each pose $\mathtt{T}_j$, we can find the visible landmarks in M_A and denote their indexes collectively as O_j, as illustrated in Fig. C.6. Then if the quadrotor moves to $\mathtt{T}_j$, its pose

is usually estimated by solving the following least squares problem

$$\mathtt{T}_j^* = \arg\min_{\mathtt{T}} \sum_{k \in O_j} \|\text{proj}(\mathtt{T}_{cb}\mathtt{T}^{-1}\mathbf{l}_k) - \tilde{\mathbf{u}}_{jk}\|^2, \tag{C.12}$$

where $\mathtt{T}_{cb}$ is the relative transformation from the camera frame c to the body frame b, $\text{proj}(\cdot)$ is the projection function of the camera, and $\tilde{\mathbf{u}}_{jk}$ is the noisy observation of the kth landmark. (C.12) is usually solved using iterative optimization methods such as the Gauss-Newton algorithm. To put it formally, the following problem is considered instead

$$\xi^* = \arg\min_{\xi} \sum_{k \in O_j} \|\text{proj}(\mathtt{T}_{cb}(\mathtt{T}_j\text{Exp}(\xi))^{-1}\mathbf{l}_k) - \tilde{\mathbf{u}}_{jk}\|^2 \tag{C.13}$$

and $\mathtt{T}_j$ is updated each iteration as $\mathtt{T}_j \leftarrow \mathtt{T}_j\text{Exp}(\xi)$. ξ is the element in $\mathfrak{se}(3)$, and $\text{Exp}(\cdot)$ maps the element in $\mathfrak{se}(3)$ to $SE(3)$. (C.13) is solved by linearizing around the current estimate of $\mathtt{T}_j$:

$$\xi^* = \arg\min_{\xi} \sum_{k \in O_j} \|\mathbf{u}_{jk} - \tilde{\mathbf{u}}_{jk} + \mathrm{J}_{\xi}^{jk}\xi\|^2, \tag{C.14}$$

where $\mathbf{u}_{jk} = \text{proj}(\mathtt{T}_{cb}\mathtt{T}_j^{-1}\mathbf{l}_k)$ and $\mathrm{J}_{\xi}^{jk} = \frac{\partial \mathbf{u}_{jk}}{\partial \xi}$. (C.14) can be solved in a closed form. The covariance of the estimated parameter is [108]

$$\Sigma_j = (\mathrm{J}_j^\top \Sigma_{\mathbf{u}}^{-1} \mathrm{J}_j)^{-1} = (\mathrm{H}_j)^{-1}, \tag{C.15}$$

where J_j is stacked up by J_{ξ}^{jk}, and $\Sigma_{\mathbf{u}}$ is the covariance matrix of $\{\mathbf{u}_{jk}\}_{k \in O_j}$, which is usually a diagonal matrix with the same value $\sigma_{\mathbf{u}}$ on the diagonal when the observations are independent and have the same noise level. H_j is the *information matrix*.

We do not only try to minimize the pose estimation error of one pose on the trajectory but all the sampled poses. The intuition is that accurate pose estimation along the trajectory can help better triangulate new landmarks, which is beneficial for the pose estimation afterwards. Because the active map M_A contains only well estimated points and is fixed, the poses $\{\mathtt{T}_j\}_{j=1}^J$ are actually independent (*i.e.* conditionally independent, conditioned on the landmark positions). Therefore we can write the whole information matrix as

$$\mathrm{H} = \begin{bmatrix} \mathrm{H}_1 & 0 & \dots & 0 \\ 0 & \mathrm{H}_2 & \dots & 0 \\ 0 & \dots & \dots & 0 \\ 0 & \dots & 0 & \mathrm{H}_J \end{bmatrix}, \tag{C.16}$$

and the full covariance is $\Sigma = \mathrm{H}^{-1}$, which is also a diagonal block matrix.

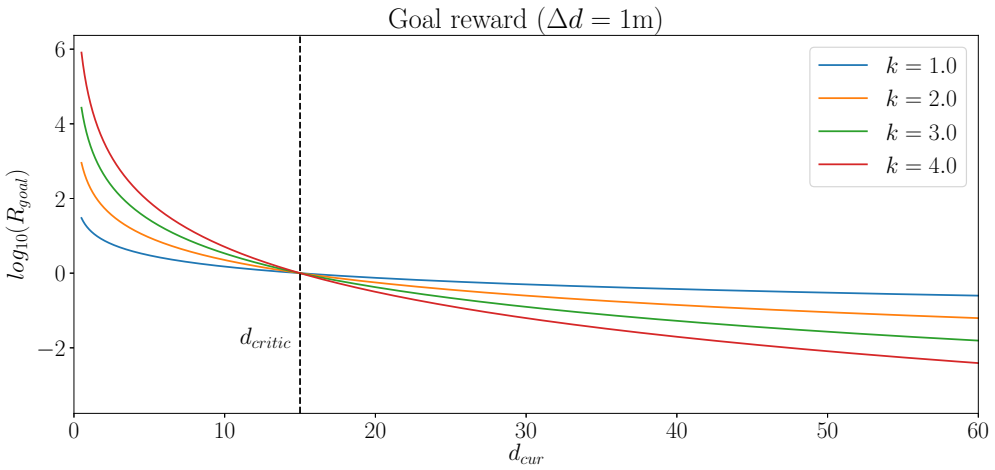


Figure C.8 – Goal reward function when $d_{\text{critic}} = 15\text{m}, \Delta d = 1\text{m}$. For the same Δd, the goal reward is higher when the quadrotor is close to the goal. When the distance to goal is smaller than d_{critic}, the goal reward increases more rapidly.

Using (C.16), we calculate the reward using an analog form of the *D-opt* criterion [43]

$$R_{\text{perc}} = k_{\text{perc}} \exp(\log([\det(\text{H})]^{\frac{1}{6J}})), \tag{C.17}$$

where k_{perc} is a parameter determining the weight for the perception quality.

C.4.5 Goal Progress

One straightforward way to evaluate how much a trajectory approaches a destination is to use the distance decrease Δd from the start point of the trajectory to the end point. However, in practice we find this approach does not generalize well to different situations. One reason is that the evaluation of the goal progress should be related to the current distance d_{cur} to the goal. For example, reducing the distance from 2m to 1m should be better rewarded than from 100m to 99m. Therefore, we define the goal reward as

$$R_{\text{goal}} = k_{\text{goal}} \Delta d \times (\frac{d_{\text{critic}}}{d_{\text{cur}}})^k, \tag{C.18}$$

where d_{critic} is a parameter that controls the size of strong attraction area near the goal, and k controls the strength of the distance-dependent property of the weight. An example of the goal reward at different distances is shown in Fig. C.8.

To summarize, we combine the collision probability (C.10), perception quality (C.17) and goal progress (C.18) and select the best trajectory defined by (C.6). In the next section, we will demonstrate the effectiveness of the proposed system in both simulation and real-world experiments.

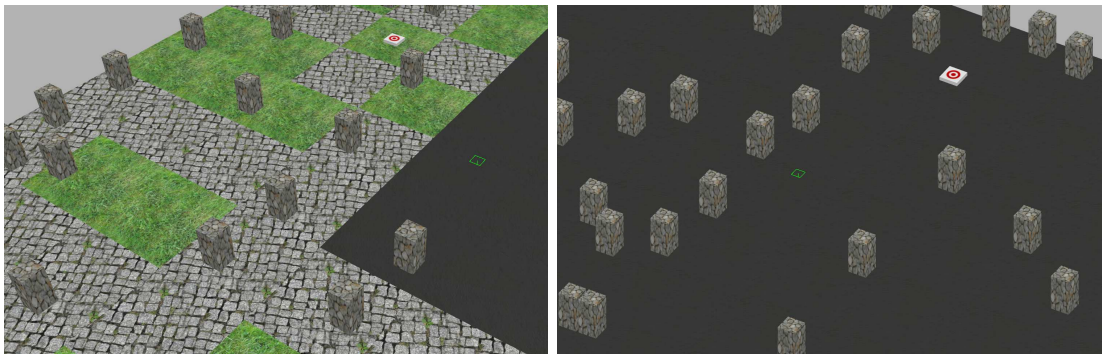


Figure C.9 – A close look of the two scenes (*L shape* and *obstacles*) used for simulation.

C.5 Experiments

To prove the effectiveness of the proposed method, we performed experiments in both simulation and real-world environments. The parameters related to calculating the total reward (C.6) are shown in Table. C.1.

Table C.1 – Parameters for simulation and real-world experiments. They share the same parameters except d_{critic} and l, which depend on the scene dimensions.

	k_{col}	k_{perc}	k_{goal}	d_{critic}	k	l (meter)
Simulation	-10000	1.5	10	15	3	5.0
Real-world	-	-	-	1	-	2.5

C.5.1 Simulation

We tested in different scenarios in simulation to show that our system does not overfit a particular environment. Statistical results are summarized from multiple runs.

We utilized the RotorS simulator [96]. Examples of the scenes used are shown in Fig. C.9 and their dimensions reported in Table. C.2. The simulated MAV is a AscTec Hummingbird quadrotor with a forward-looking camera. In each scenario, we started the quadrotor at slightly different beginning positions and commanded it to fly to a given destination for 10 times. In our simulation, the quadrotor rarely crashed into obstacles, even with large state estimation error, which is the advantage of using only local information for motion planning. Therefore we define the criteria for success using the state estimation error. During each run, once the state estimation diverged from the groundtruth over 5 meters, we terminated the execution and reported the trial as a failure. Once the distance of the state estimation to the given destination is smaller

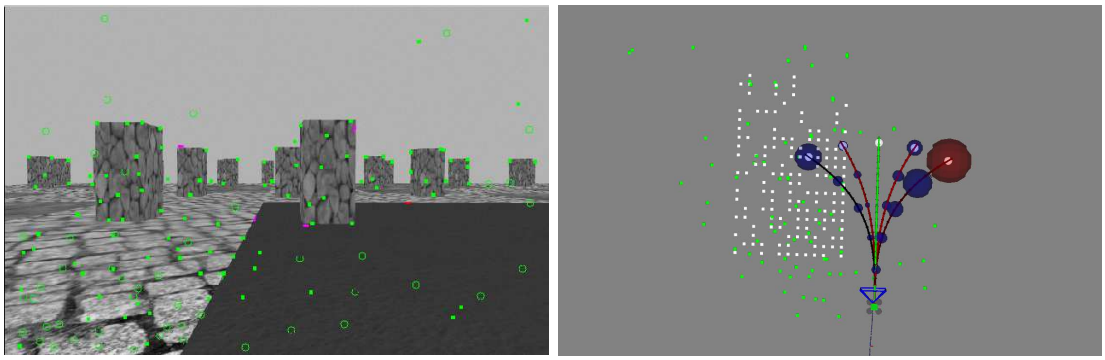


Figure C.10 – An instant of the perception aware receding horizon navigation in execution. On the left is the image from the forward-looking camera, where the green solid dots and circles are landmarks and seeds in SVO respectively. On the right is the visualization of the trajectory generation and evaluation. Five trajectories are generated in this case. The blue/red spheres on each trajectory denotes the covariance at the sampled poses, where red means the corresponding pose is not constrained (the information matrix is singular). Our method correctly identifies the rightmost trajectory, which steers the quadrotor towards textureless region, is the worst in terms of perception quality.

than 3 meters, we reported the trial as a success. To eliminate the inaccuracy induced by the initialization of visual odometry, we initialized SVO using the groundtruth from the simulator.

We compared the performance of our method and a purely-reactive navigation method (*i.e.* without R_{perc} in (C.6)). Fig. C.12 shows the trajectories overlayed on the simulated scenes and the state estimation error over the traveled distance. Fig. C.11 shows the final state estimation error when the simulation was terminated. Table. C.2 reported the number of successful trials for each scene.

The first scene *L shape* consists of areas with strong texture (grass and stone) and weak texture (dark area in the bottom left part). We can observe that the purely-reactive navigation method commanded the quadrotor to fly directly to the goal but the trajectory passed visually degraded part, resulting in large state estimation error. By contrast, our method prevented the quadrotor from entering the less textured area and was able to maintain a reasonable state estimation. Similarly, the second scene *transition* contains a visually degraded area in the middle, which lies between the start point and the destination. While our method was able to steer the quadrotor away from that area and reached the destination, reactive navigation was not able to finish the task successfully. An example of our system in execution is shown in Fig. C.10.

Our method outperforms the reactive navigation in the first two scenes by a large margin. The main reason for the drastic difference is that these two scenes contain two obviously separated areas with good and poor texture (bottom left and the rest for *L shape*, middle and surround for *transition*), and the visually degraded area lies between the start position and the goal. Therefore, reactive navigation will inevitably enter the area with

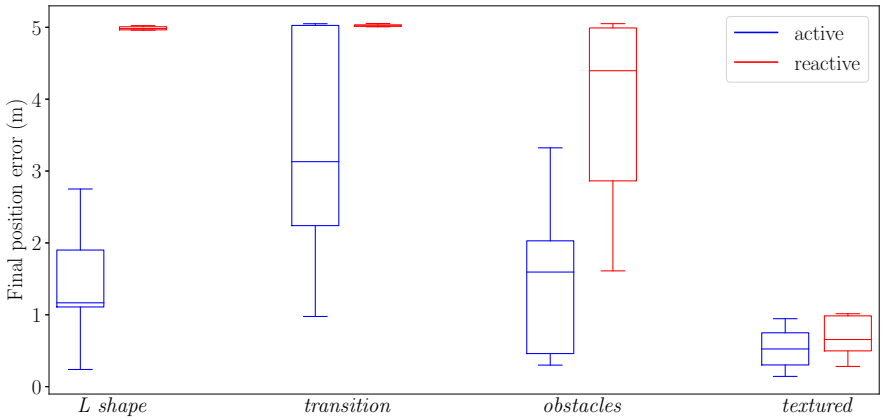


Figure C.11 – Position error when the each run is terminated (either the goal is reached or the state estimation is 5 meters away from the groundtruth.)

Table C.2 – Successful runs out of 10 trials for different scenes.

Scene	Dimension (m)	Reactive	Ours
L shape	100×100	1	**5**
transition	100×100	0	**5**
obstacles	80×80	6	**9**
textured	60×60	9	**10**

poor texture, resulting in poor performance. To illustrate the usefulness of our method in a more general setup, we tested on two more scenes. In the third scene *obstacles*, the ground has little texture and most of the visual information comes from the obstacles. In this scene, we can see that the trajectories of our method and reactive navigation are less different than the first two scenes, but the state estimation error of our method is still obviously smaller, as shown in Fig. C.12 and Fig. C.11. If the trajectories are inspected closely, we can see that our method steered the quadrotor to move closer to obstacles compared to the reactive one. This implies that obstacles with visual features are both repellers and attractors in our method: getting close to such obstacles will decrease state estimation uncertainty but also increase the collision risk. The fourth scene *textured* is fully textured without obviously visually degraded part. Both reactive navigation and our method performed well in this scene, but we can still observe slightly better performance from our method.

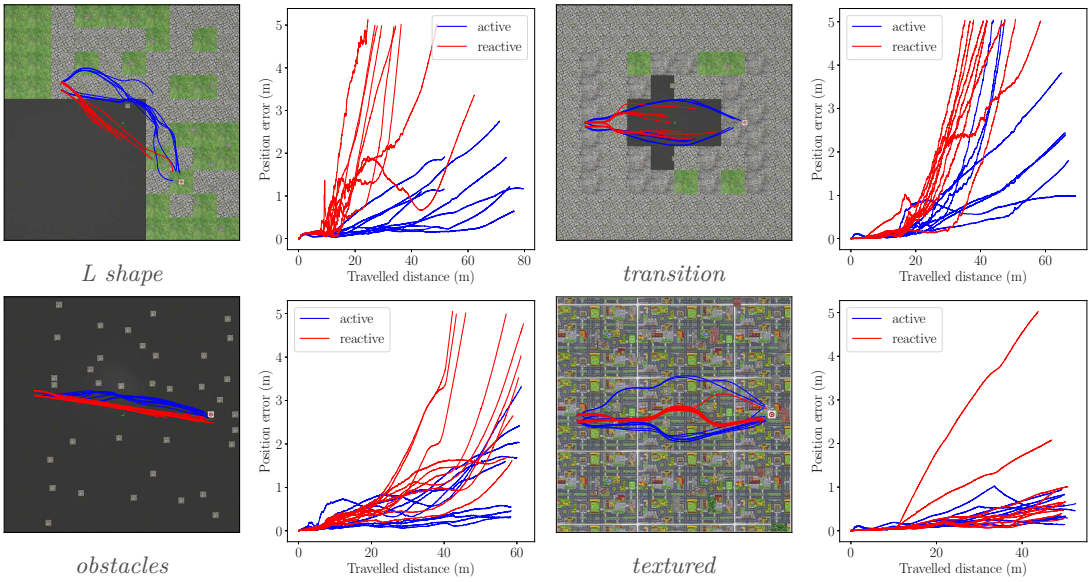


Figure C.12 – Trajectory overlayed on the simulation environment (left column) and position error with respect to the traveled distance (right column) for different scenes. Each navigation strategy is executed 10 times on the same scene with slightly different start positions.

C.5.2 Real-world Experiments

The quadrotor used in real-world experiments is equipped with a forward looking MatrixVision mvBlueFOX-MLC200w monochrome global shutter camera. The onboard computer is ODROID-XU4. It also carries an PX4FMU autopilot board from Pixhawk that includes an IMU. In our experiments, the monocular state estimation and mapping system was done on-board, while the trajectory generation and evaluation was computed on a laptop at 50Hz.

We tested our method in a scene that contains both texture-rich and textureless areas. A photo of the scene and several example trajectories from our method are shown in Fig. C.13. Similar to the results in simulation, our system was able to command the quadrotor to follow a more informative path to reach the goal.

C.6 Conclusions and Future Work

In this work, we proposed to integrate active perception in a receding horizon setting for a goal reaching task. In particular, we designed a perception-aware receding horizon navigation system using a single forward looking camera for MAVs. We used a monocular visual odometry SVO and a dense reconstruction algorithm REMODE to provide the essential information for navigation. Using the information, we generated a library of trajectories and evaluated them in terms of collision probability, perception quality and

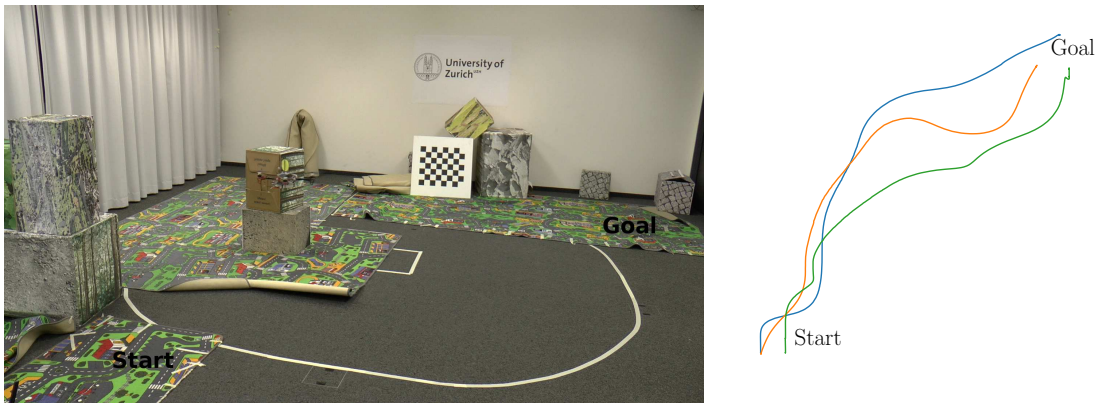


Figure C.13 – Real world scene used to test our method. The start point is at the bottom left and the goal top right. The plot on the right shows the topviews of the trajectories of the quadrotor when it was controlled by our method. The trajectories are "attracted" by the textured area instead of going directly from the start position to the goal.

goal progress to select the next motion for MAVs, which naturally combines different performance metrics. We demonstrated the effectiveness of our system by extensive simulation and real-world experiments: in addition of the capability of avoiding obstacles, our perception-aware receding horizon navigation system is able to select motion to favor the state estimation accuracy, which is especially advantageous in environments with visually degraded regions.

Future work would include further real-world validation in different environments to better understand the completeness and failure cases of the method. Generating informative motion primitives is also of interest.

D Fisher Information Field for Perception-aware Planning

Reprinted, with permission, from:

Z. Zhang and D. Scaramuzza. "Fisher Information Field: an Efficient and Differentiable Map for Perception-aware Planning". In: *Under review in IEEE Trans. Robot.* (2020). URL: https://arxiv.org/abs/2008.03324 [322]

Fisher Information Field: an Efficient and Differentiable Map for Perception-aware Planning

Zichao Zhang and Davide Scaramuzza

Abstract — For mobile robots to localize robustly, actively considering the perception requirement at the planning stage is essential. Traditionally, the map is represented as a point cloud. The localization quality metric at the pose of interest is computed from the point cloud and integrated in motion planning algorithms. This approach scales linearly with the number of landmarks in the environment and does not allow the reuse of the computed localization quality metrics. To overcome these drawbacks, we propose the first dedicated map representation for evaluating the localization quality of 6 degree-of-freedom poses for perception-aware motion planning. By formulating the Fisher information and sensor visibility carefully, we are able to separate the rotational invariant component from the localization information and store it in a voxel grid, namely the *Fisher information field*. This step only needs to be performed once for a known environment. The Fisher information for arbitrary poses can then be computed from the field in *constant* time, eliminating the need of costly iterating all the 3D landmarks at the planning time. Experimental results show that the proposed Fisher information field can be applied to different motion planning algorithms and is at least one order-of-magnitude faster than using the point cloud directly. Moreover, the proposed map representation is differentiable, resulting in better performance than the point cloud when used in trajectory optimization algorithms.

Multimedia Material

A video demonstrating the proposed Fisher Information Field can be viewed at http://rpg.ifi.uzh.ch/fif.html.

Open Source Code

An implementation of the proposed method is available at https://github.com/uzh-rpg/rpg_information_field.

D.1 Introduction

On-board visual sensing and computing permits robots to operate autonomously without relying on external infrastructures. This ability is essential in a wide-range of real-world scenarios, such as indoor environments and city canyons, where Global Positioning System (GPS) is inaccessible or unreliable. However, relying on on-board sensing brings additional constraints to motion planning algorithms. Specifically, the robot motion impacts the information that will be captured by the cameras and thus influences the performance of perception algorithms. Therefore, the requirement of visual perception has to be taken into consideration in motion planning to better accomplish a task - this is known as *active perception* [12]. In other words, the motion planning process has to be *perception-aware*. In this paper, we are particularly interested in the task of *visual localization*, which estimates the 6 Degree-of-Freedom (DoF) camera pose from which a given image was taken relative to a reference scene representation. We refer to the process of considering the quality of visual localization in motion planning as *active visual localization*. Active visual localization, or more generally active visual simultaneous localization and mapping (SLAM), is still an open problem [35].

One major paradigm for active visual localization is to plan the sensor motion based on the Fisher information/covariance in the corresponding estimation problem. Specifically, the *localizability* of a 6 DoF pose (*i.e.* how well/uncertain the pose can be estimated) is quantified by Fisher information and considered in motion planning algorithms. For a landmark-based map, which is arguably the most common scene representation, the 6 DoF pose is usually estimated by solving a Perspective-n-Points (PnP) problem, and the corresponding Fisher information needs to be computed by iterating over all the points in the map (Section D.6). This method, in spite of the convenience of using the same scene representation as visual localization (*i.e.* point clouds), exhibits several limitations. **First**, to evaluate the localizability of a single pose, one needs to evaluate the information for *all* the 3D points, the complexity of which increases *linearly* with the number of landmarks. **Second**, this process has to be repeated many times in both sampling-based (*i.e.* evaluating motion samples) and optimization-based methods

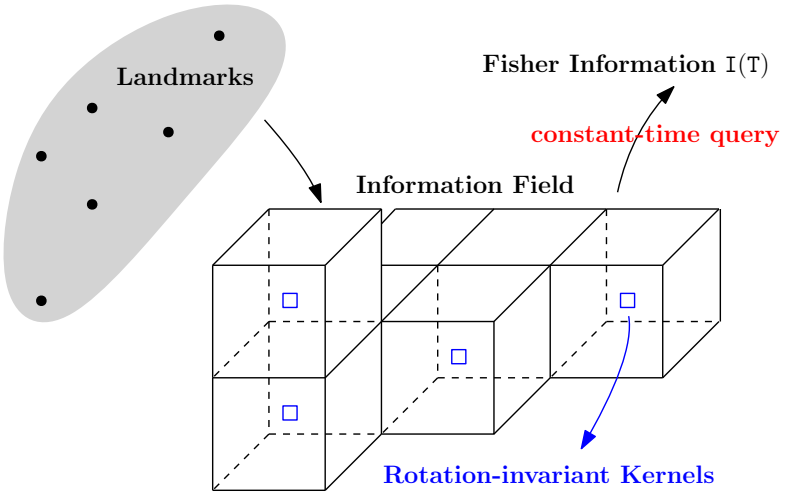


Figure D.1 – Illustration of the proposed Fisher information field. The gray cloud denotes the 3D landmarks in the environment. For each voxel (black cubes), the building process summarizes the rotation-independent information kernels (D.21) or (D.31) (blue squares). Then the information of an arbitrary pose T can be computed in *constant time* without accessing the original 3D landmarks.

(*i.e.* optimization iterations), which introduces redundant computation, especially when the planning is performed multiple times in the same environment. **Third**, due to the discontinuity of the actual visibility (see Fig. D.2), the Fisher information and related metrics are not differentiable with respect to the 6 DoF pose, which is not ideal for optimization-based motion planning algorithms. These limitations indicate that point clouds, as a natural representation in SLAM/localization, is not ideal for the task of active visual localization. Unfortunately, there is little work in designing dedicated scene representations for computing the Fisher information of 6 DoF localization.

In view of the aforementioned limitations of point clouds, we propose a dedicated scene representation, namely *Fisher Information Field* (FIF), for querying the Fisher information of arbitrary 6 DoF poses. Specifically, the scene is represented as a voxel grid. For each voxel, we summarize a *rotation-independent* component of the Fisher information from all the 3D landmarks and store it in the voxel, which is applicable for all the poses that fall in this voxel, regardless of the orientation. At query (*e.g.* motion planning) time, given a 6 DoF pose, we first get the corresponding voxel via voxel hashing [193], and then the full Fisher information (under some approximation) of this pose can be recovered by applying a linear transformation to the stored rotation-independent component. The computing of the Fisher information for a pose is thus of *constant* time complexity instead of linear. Moreover, since the proposed FIF is precomputed in a voxel grid, it can be used for multiple planning sessions and easily updated when landmarks are added to or deleted from the environment.

The idea of using a voxel grid is similar to Euclidean Signed Distance Field (ESDF)

[198] for collision avoidance, which stores in each voxel the distance to the closest point. However, the key difference/difficulty is that the Fisher information additionally depends on the camera orientation due to the fact that the visibility of landmarks can vary drastically with orientations. We therefore propose a novel formulation of the Fisher information (Section D.5) that allows the aforementioned separation and pre-computation, which is key to the efficient query process. The formulation is also differentiable, making our map representation suitable for optimization-based motion planning algorithms.

In summary, the proposed FIF overcomes the aforementioned limitations of using point clouds to compute the Fisher information of 6 DoF poses. To the best of our knowledge, this is the first dedicated map representation that is capable of computing the Fisher information of 6 DoF localization efficiently. Our map representation is general and can be integrated with different motion planning algorithms. Experiments with both sampling-based and optimization-based methods demonstrate that FIF is up to two order of magnitude faster than point clouds in a typical motion planning scenario. The performance, in terms of the localization success rate and accuracy of the planned motion, is comparable to point clouds and even better in trajectory optimization, where the differentiability of FIF becomes crucial.

This paper is an extension of our previous paper [321]. The novelty of the present work includes:

- A non-parametric visibility approximation that is more accurate and scalable than the quadratic function in the previous work.
- Demonstration of the FIF in both sample-based and optimization-based motion planning algorithms.
- Extension of the open source code to include the novel contributions.

The rest of the paper is structured as follows. After reviewing the related work in Section D.2, we briefly introduce Fisher information matrix and Gaussian process as preliminaries for our approach in Section D.3. In Section D.4, we describe how Fisher information is typically used in a perception-aware planning setup and hightlight the limitations of computing the Fisher information from point clouds directly. Then we introduce our formulation of the Fisher information in Section D.5 and how it can be used to design a dedicated map representation for motion planning Section D.6. In Section D.7, we present detailed experimental results regarding the properties of the proposed Fisher information field and its application to different motion planning algorithms in photorealistic simulation. Finally, we conclude the paper with some discussion about our method and possible future directions in Section D.8.

D.2 Related Work

D.2.1 Perception-aware Motion Planning

Considering perception performance in planning has been extensively studied in different contexts. Early works include maximizing the Fisher information (or equally minimizing the covariance) about the robot state and the map in navigation tasks [85, 166], minimizing the entropy of the robot state in known environments [33, 226], and actively searching features in SLAM systems [65]. Recently, with the advance of drones, several works have been done to couple perception, planning and control on agile aerial platforms [1, 229, 181, 202, 6, 323, 303, 83].

Despite the extreme diversity of the research in this topic, related work can be categorized based on the method to generate motion profiles. One paradigm used sampling-based methods, which discretize the space of possible motions and find the optimal one in a discrete set. Roy *et al.* [226] used the Dijkstra's algorithm to find the path on a grid that minimizes a combined cost of collision and localization. Papachristos *et al.* [202] and Costante *et al.* [62] adapted the rapidly-exploring random tree (RRT) algorithms to incorporate the perception cost, and the latter additionally considered the photometric property of the environment. Alzugaray *et al.* [6] sampled positions near obstacles based on the intuition that pose estimation error is small when the camera is close to the features on obstacles. Then path planning was carried out based on the sampled positions. Zhang *et al.* [323] proposed to evaluate motion primitives against multiple costs, including the localization uncertainty, in a receding horizon fashion. Instead of a combined cost, as in most of previous works, Ichter *et al.* [116] used multi-objective search for perception-aware planning.

Alternatively, researchers have explored to plan in the continuous motion space. Indelman *et al.* [118] considered optimizing the motion within a finite horizon to minimize a joint cost including the final pose covariance, which was later extended to visual-inertial sensing and self-calibration in [79]. Watterson *et al.* [303] studied the general problem of trajectory optimization on manifolds and applied their method to planning under the field-of-view (FoV) constraint of the camera. The perception constraint can also be used at the controller level. Falanga *et al.* [83] integrated the objective of maximizing the visibility of a point of interest and minimizing its velocity in the image plane as the costs in model predictive control (MPC). Lee *et al.* [149] trained a neural network to predict the dynamics of the pixels on the objects of interest (*e.g.* gates in drone racing) and incorporated such information in a MPC framework. Greeff *et al.* [103] considered the perception task of visual localization in a teach-and-repeat setup. They modeled the probability of whether a landmark can be matched considering the perspective change and FoV constraint, and used the model in a MPC controller. In the context of drone racing, there is also work that considers the time optimality in trajectory generation or optimal control, in addition to perception constraints. Murali *et al.* [188] generated the

position trajectory by considering collision constraints, and optimized the yaw considering the co-visibility of certain landmarks and the execution time of the trajectory. Spasojevi *et al.* [265] further proposed a trajectory parameterization algorithm that considers the FoV constraints and optimizes the traverse time at the same time.

In the aforementioned work, the perception related cost/metric were always calculated from a sparse set of 3D points in the environment. As noted in Alexei *et al.* [166], calculating the metric (*e.g.* "localizability") is an expensive operation, which we believe is due to the lack of proper map representations. Unfortunately, little work has been done in developing dedicated representations for the efficient computation of related metrics, which is the primary contribution of this work. Next, we further review some related work in map representations for perception quality and other related tasks.

D.2.2 Related Map Representations

Roy *et al.* [226] pre-computed and stored the information in a 2D grid, but their method was limited to 360° FoV sensors. Specifically, the visual information (*e.g.* visibility) are invariant regardless of the camera orientation for omnidirectional sensors, and thus their map did not need to consider the impact of orientations, which is not true for cameras with limited FoVs. More recently, Ichter *et al.* [116] trained a neural network to predict the state estimation error and generated a map of perception cost using the network prediction. However, their map only contains the averaged cost of different orientations and, therefore, cannot be used to evaluate the cost of an arbitrary 6 DoF pose. In contrast, our method explicitly models the FoV constraint and can represent the information of 6 DoF poses efficiently. As a concurrent work, Fey *et al.* [95] proposed the similar idea of combining the information from many landmarks for efficient online inference in the context of trajectory optimization. In contrast, our work focuses on a general map representation that is applicable to different motion planning algorithms.

Our approach is also connected to a couple of map representations for other tasks. It is partially inspired by the approach of using ESDF for collision-free motion planning [197]. Conceptually, both ESDF and our method summarizes the information from many 3D points/landmarks into a compact field (in the form of a voxel grid) for efficient query. In the context of computer graphics, a common technique to speedup the rendering process is *precomputed visibility volume* [205]. Basically, the scene is first divided in to cells, and for each cell, the visibility states of the static objects from this cell are precomputed and stored before the rendering process. Then at rendering time, whether to render a specific object can be efficiently determined from the precomputed values. The precomputed visibility volume reduces the rendering time at the cost of increasing runtime memory. It is conceptually similar to our approach, where we achieve efficient query of the FIM at the cost of more memory usage.

D.3 Preliminaries

D.3.1 Fisher Information

For a general parameter estimation problem, the Fisher information matrix (FIM) summarizes the information that the observations carry about the parameters to be estimated. To put it formally, if the measurement process can be described as a conditional probability density function $\mathrm{p}(\mathbf{z}|\mathbf{x})$, where $\mathbf{z}$ is the measurement and $\mathbf{x}$ the parameters, one definition[1] of the Fisher information is

$$\mathtt{I}_{\mathbf{x}}(\mathbf{z}) = (\frac{\partial}{\partial \mathbf{x}} \log \mathrm{p}(\mathbf{z}|\mathbf{x}))^{\top} (\frac{\partial}{\partial \mathbf{x}} \log \mathrm{p}(\mathbf{z}|\mathbf{x})). \tag{D.1}$$

With identical and independent zero-mean Gaussian noise $\mathcal{N}(0, \sigma^2)$ on the measurement, (D.1) can be written as

$$\mathtt{I}_{\mathbf{x}}(\mathbf{z}) = \frac{1}{\sigma^2} (\mathtt{J}_{\mathbf{x}})^{\top} \mathtt{J}_{\mathbf{x}}, \quad \text{where } \mathtt{J}_{\mathbf{x}} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}}. \tag{D.2}$$

Note that in practice (D.1) and (D.2) are usually evaluated at the estimate $\mathbf{x}^*$ instead of the unknown true value $\mathbf{x}$.

The Fisher information is a pivotal concept in parameter estimation problems. Most notably, the inverse of the FIM defines the Cramér-Rao lower bound, which is the smallest covariance (in terms of Loewner order) that can be achieved by an unbiased estimator [108, App. 3.2] [17, p. 14]. Note that the widely used nonlinear maximum likelihood estimator (MLE) is in general biased, but the bias also tends to decrease when the Fisher information increases [23]. Due to its rich theoretical implications, the FIM is widely used in different applications, such as optimal design of experiments [208], active SLAM [118] and information selection [42, 113, 293].

D.3.2 Gaussian Process Regression

A Gaussian process is a collection of random variables, and any subset of them has a joint Gaussian distribution [214]. In the context of a regression task, suppose we know the samples at $\mathbf{z} = \{z_i\}_{i=1}^{P}$ with the output $\mathbf{y} = \{y_i\}_{i=1}^{P}$, and we would like to know the output value y^* at z^*. Under the assumption of Gaussian process, we have

$$\begin{bmatrix} \mathbf{y} \\ y^* \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} K_{\mathbf{zz}} & K_{\mathbf{z}z^*} \\ K_{z^*\mathbf{z}} & k(z^*, z^*) \end{bmatrix}), \tag{D.3}$$

[1] The presented definition is the *observed Fisher information*. See [78] for the comparison of different concepts.

where $K^{i,j}_{\mathbf{zz}} = k(z_i, z_j)$, $K^{i}_{z^*\mathbf{z}} = k(z^*, z_i)$ and $K^{i}_{\mathbf{z}z^*} = k(z_i, z^*)$. Then the GP regression simply takes the conditional distribution

$$y^* \sim \mathcal{N}(K_{z^*\mathbf{z}}K^{-1}_{\mathbf{zz}}\mathbf{y}, k(z^*, z^*) - K_{z^*\mathbf{z}}K^{-1}_{\mathbf{zz}}K_{\mathbf{z}z^*}), \tag{D.4}$$

which gives both the regressed value and variance.

Obviously, the properties of the prior (D.3) and the regressed result (D.4) depends on the function $k(\cdot)$. $k(a, b)$ is called the *kernel function*, and intuitively encodes the correlation of the outputs at a, b. Often $k(\cdot)$ is a parameterized functions, whose parameters are the *hyperparameters* of a GP. Perhaps the most used kernel function is the *Squared Exponential kernel*:

$$k_{\mathrm{SE}}(a, b) = \sigma^2 \exp(-\frac{(a-b)^2}{2l^2}), \tag{D.5}$$

where σ and l are the hyperparameters and can be calculated by maximizing the likelihood of the training data $\mathbf{z}$ and $\mathbf{y}$.

GP is a flexible model that finds many applications in robotics (*e.g.* motion planning [72], state estimation [18]). For simplicity, the above introduction is limited to the case where both the output and input are scalars. However, GP can also be generalized to vector input and output. For a thorough description of GP (*e.g.* optimization of hyperparameters), we refer the reader to [214].

D.4 Planning with FIM: Standard approach

To take localization quality into consideration, a common practice is to incorporate FIM in the motion planning algorithm. Without the loss of generality, we denote the motion as a continuous time function $f(t; \mathbf{m})$, parameterized with $\mathbf{m}$. The output of the function is the 6 DoF pose of the camera at a given time. We can then formulate a perception-aware motion planning algorithm as:

$$\mathbf{m}^* = \arg\min_{\mathbf{m}} \mu_{\mathrm{v}} C_{\mathrm{v}}(f(t; \mathbf{m})) + \mu_{\mathrm{o}} C_{\mathrm{o}}(f(t; \mathbf{m})), \tag{D.6}$$

where C_{v} is the cost related to visual localization, C_{o} denotes the other cost terms collectively (*e.g.* collision and execution time) and $\mu_{\mathrm{v}}/\mu_{\mathrm{o}}$ are the corresponding weights. Since localization can be viewed as the estimation of the poses of interest, FIM can be used to quantify the estimation error and, thus, the localization quality. Evaluating the

cost using M discrete samples, we have

$$C_{\mathrm{v}} = -s(\begin{bmatrix} \mathtt{I}_{\mathtt{T}_1} & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \mathtt{I}_{\mathtt{T}_M} \end{bmatrix}), \quad \mathtt{I}_{\mathtt{T}_i} = \sum_{k}^{k \in V_i} \mathtt{I}_{\mathtt{T}_i}(\mathbf{u}_{ik}) \tag{D.7}$$

where $\mathtt{T}_i$ is the ith sample, V_i is the index set of visible landmarks in $\mathtt{T}_i$, and $\mathbf{u}_{ik}$ is the projection of the kth landmark in $\mathtt{T}_i$. $s(\cdot)$ is a metric function that converts the information matrix into a scalar (*e.g.* determinant).

(D.6) can be solved using sampling-based methods, such as RRT [202] and motion primitives [323], or optimization-based methods [118]. Either way, the FIMs for individual poses in (D.7) need to be computed multiple times for different motion samples or the iterations in optimization, which is the computational bottleneck for solving (D.6). Specifically, the calculation of $\mathtt{I}_{\mathtt{T}_i}$ requires iterating all the landmarks in the environment and evaluating the individual FIM for all the visible ones (the sum in (D.7)), which scales linearly with the number of landmarks. Moreover, $\mathtt{I}_{\mathtt{T}_i}$ needs to be recomputed from scratch once the pose $\mathtt{T}_i$ changes (both the visibility and the Jacobian in (D.2) change), which introduces redundant computation across the iterations in the planning algorithm as well as multiple planning sessions. This motivates us to look for an alternative formulation of (D.7) to mitigate the bottleneck.

It is worth mentioning that, compared with complete probabilistic treatment as in [33, 226], we make the simplification in the problem formulation (D.6) (D.7) that the localization process purely depends on the measurements (*i.e.* no prior from the past). However, this is not a limitation of our work. The computational bottleneck exists as long as the Fisher information is used to characterize the visual estimation process. The essence of this work is a compact representation of the information to allow efficient computation, which is widely applicable.

D.5 Approximating FIM: Factoring out the Rotation

In this section, we focus on the formulation of the Fisher information for a single pose, since the FIMs of different poses are calculated independently in the same way. Let $\mathtt{T}_{\mathtt{wc}} = \{\mathtt{R}_{\mathtt{wc}}, \mathbf{t}_{\mathtt{wc}}\}$ stands for the pose of the camera in the world frame, $\{\mathbf{p}_i^{\mathtt{w}}\}_{i=1}^N$ the 3D landmarks in the world frame and $\mathtt{I}_i$ the information matrix corresponding to the observation of the ith landmark. The FIM for the pose can be written as

$$\mathtt{I}_{\mathtt{T}_{\mathtt{wc}}} = \sum_{i=1}^{N} v(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}}) \mathtt{I}_i, \tag{D.8}$$

where $v(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}})$ is a binary valued function indicating the visibility of the ith landmark. Conceptually, our goal is to find an approximation $\mathtt{S}(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}}) \approx v(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}})\mathtt{I}_i$ that can be written as $\mathtt{S}(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}}) = \mathtt{S}(\mathtt{H}(\mathbf{t}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}}), \mathtt{R}_{\mathtt{wc}})$ and satisfies

$$\mathtt{I}_{\mathtt{T}_{\mathtt{wc}}} \approx \sum_{i=1}^{N} \mathtt{S}(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}}) = \mathtt{S}(\sum_{i=1}^{N} \mathtt{H}(\mathbf{t}_{\mathtt{wc}}, \mathbf{p}_i^{\mathtt{w}}), \mathtt{R}_{\mathtt{wc}}). \tag{D.9}$$

In words, we would like to find an approximation that can be factored into two components, one of which does not depend on rotation (*i.e.* $\mathtt{H}(\cdot)$ in (D.9)), and the approximation is linear in terms of the rotation-independent part.

The linear form lead to two favorable properties. First, for one position $\mathbf{t}_{\mathtt{wc}}$, the sum of the rotation-independent $\mathtt{H}(\cdot)$ of all the landmarks need to be computed *only once*, and the sum can be used to calculate the approximated information at this position for arbitrary rotations in constant time; second, we can easily update the sum when new landmarks are added or old ones deleted. This form naturally leads to a volumetric representation that allows online update, as described in Section D.6.1.

The approximation (D.9) is achieved by first carefully parameterizing the information matrix $\mathtt{I}_i$ to be rotation-invariant (Section D.5.1) and replacing the binary valued function $v(\cdot)$ with a smooth alternative (Section D.5.2).

D.5.1 Rotation Invariant FIM

The observation of a 3D landmark $\mathbf{p}^{\mathtt{w}}$ can be represented in different forms, such as (normalized) pixel coordinates and bearing vectors. In this work, we choose to use the bearing vector $\mathbf{f}$ because of its ability to model arbitrary FoVs. Then the noise-free measurement model of a landmark $\mathbf{p}_i^{\mathtt{w}}$ is

$$\mathbf{f}_i = \frac{\mathbf{p}_i^{\mathtt{c}}}{\|\mathbf{p}_i^{\mathtt{c}}\|_2} = \frac{1}{n_i}\mathbf{p}_i^{\mathtt{c}}, \quad \mathbf{p}_i^{\mathtt{c}} = \mathtt{T}_{\mathtt{cw}}\mathbf{p}_i^{\mathtt{w}}, \tag{D.10}$$

and the Jacobian of interest is

$$\mathtt{J}_i = \frac{\partial \mathbf{f}_i}{\partial \mathbf{p}_i^{\mathtt{c}}} \frac{\partial \mathbf{p}_i^{\mathtt{c}}}{\partial \mathtt{T}_{\mathtt{wc}}}. \tag{D.11}$$

While the first part in (D.11) is trivially

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{p}_i^{\mathtt{c}}} = \frac{1}{n_i}\mathcal{I}_3 - \frac{1}{n_i^3}\mathbf{p}_i^{\mathtt{c}}(\mathbf{p}_i^{\mathtt{c}})^\top, \tag{D.12}$$

the derivative $\frac{\partial \mathbf{p}_i^{\mathtt{c}}}{\partial \mathtt{T}_{\mathtt{wc}}}$ is more involved. To handle the derivatives related to 6 DoF poses without over-parametrization, the element in $\mathfrak{se}(3)$ (denoted as $\boldsymbol{\xi}$) is often used. In our case, $\frac{\partial \mathbf{p}_i^{\mathtt{c}}}{\partial \mathtt{T}_{\mathtt{wc}}}$ is replaced by

$$\frac{\partial \mathbf{p}_i^{\mathtt{c}}}{\partial \mathtt{T}_{\mathtt{wc}}} \to \frac{\partial (\exp(\boldsymbol{\xi}^\wedge)\mathtt{T}_{\mathtt{wc}})^{-1}\mathbf{p}_i^{\mathtt{w}}}{\partial \boldsymbol{\xi}} \quad \text{or} \quad \frac{\partial (\mathtt{T}_{\mathtt{wc}} \exp(\boldsymbol{\xi}^\wedge))^{-1}\mathbf{p}_i^{\mathtt{w}}}{\partial \boldsymbol{\xi}}, \tag{D.13}$$

where $\exp(\cdot)$ is the exponential map of the Special Euclidean group SE(3). The two forms corresponds to expressing the perturbation $\delta\boldsymbol{\xi}$ globally in the world frame or locally in the camera frame respectively. Using the first form, we have the Jacobian in (D.11) as

$$\mathtt{J}_i = \frac{\partial \mathbf{f}_i}{\partial \mathbf{p}_i^{\mathtt{c}}} \mathtt{R}_{\mathtt{cw}} [-\mathcal{I}_3, \quad [\mathbf{p}_i^{\mathtt{w}}]_\times]. \tag{D.14}$$

With the global perturbation formulation, for two poses that differ by a relative rotation $\mathtt{T}_{\mathtt{wc}}$ and $\mathtt{T}_{\mathtt{wc}'} = \{\mathtt{R}_{\mathtt{wc}}\mathtt{R}_{\mathtt{cc}'}, \mathbf{t}_{\mathtt{wc}}\}$, their Jacobians (D.14) have a simple relation $\mathtt{J}_i' = \mathtt{R}_{\mathtt{c}'\mathtt{c}}\mathtt{J}_i$, from which the corresponding FIMs turn out to be the same

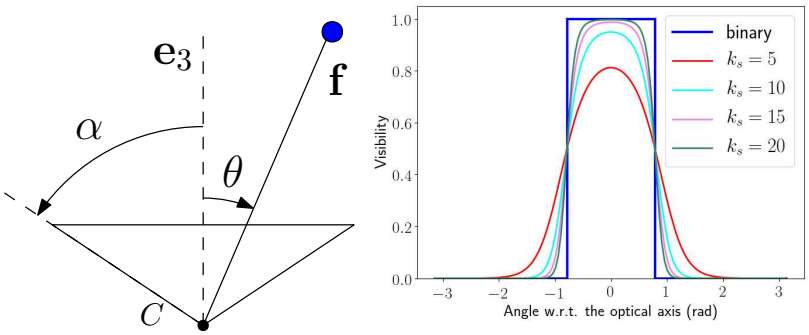
$$\mathtt{I}_i' \overset{\text{(D.2)}}{=} \frac{1}{\sigma^2}\mathtt{J}_i^\top \mathtt{R}_{\mathtt{cc}'}\mathtt{R}_{\mathtt{c}'\mathtt{c}}\mathtt{J}_i = \mathtt{I}_i. \tag{D.15}$$

The rotation-invariance is not surprising. Intuitively, since we are considering only part of (D.8) (without visibility constraint) and modeling the camera as a general bearing sensor, the camera should receive the same information regardless of its rotation. Moreover, the choice of global frame expresses the constant information in a fixed frame, resulting in the invariance (D.15). If the local perturbation in (D.13) is chosen, such invariance is not possible, and the information matrix will be related by an adjoint map of SE(3) [269, Ch. 2].

To summarize, by choosing the bearing vector as the observation and parameterizing the pose perturbation in the global frame, the information matrix, without the visibility constratint, is rotation-invariant. Next, we will see how to handle the visibility function $v(\cdot)$ in (D.8).

D.5.2 Visibility Approximation

The exact visibility $v(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}^{\mathtt{w}})$ is a non-trivial function, as the horizontal/vertical/diagonal FoVs are not the same. In practice, to check whether a point is visible at a pose, one needs to project the 3D point to the image plane $\mathbf{u} = \text{proj}(\mathbf{p}^{\mathtt{w}}, \mathtt{T}_{\mathtt{wc}})$ and check whether



(a) Visibility as a function of θ **(b)** Visibility approximated by sigmoid functions with different k_s

Figure D.2 – Visibility modeling. α is half of the FoV, $\mathbf{f}$ is the bearing vector observation, $\mathbf{e}_3$ is the optical axis of the camera, and C is the projection center.

the projected pixel coordinates $\mathbf{u}$ is within the image boundary I:

$$v(\mathtt{T}_{\mathtt{wc}}, \mathbf{p}^{\mathrm{w}}) = \begin{cases} 1, & \mathbf{u} \in I \\ 0, & \mathbf{u} \notin I \end{cases}, \tag{D.16}$$

For simplicity, we assume that the visibility $v(\cdot)$ is a function of the angle θ between the bearing vector $\mathbf{f}$ of the landmark and the optical axis $\mathbf{e}_3 = [0, 0, 1]^\top$. This essentially assume that the FoVs along different directions of the image plane are the same. We further use a sigmoid function to have a smooth visibility function

$$v(\theta) = \frac{1}{1 + e^{-k_s(\cos\theta - \cos\alpha)}}, \tag{D.17}$$

which is important if the visibility model is involved in optimization. k_s controls the steepness of the visibility function, and α is half of the FoV. The visibility modeling and the effect of different k_s are illustrated in Fig. D.2.

Since our goal is to arrive at the form (D.9), we further assume that the simplified $v(\theta)$ can be written, by certain approximation, as a dot product of two vectors:

$$v(\theta) \approx (\mathbf{v}^{\mathrm{r}}(\mathtt{R}_{\mathtt{wc}}))^\top \mathbf{v}^{\mathrm{p}}(\mathbf{p}^{\mathrm{w}}, \mathbf{t}_{\mathtt{wc}}), \tag{D.18}$$

where $\mathbf{v}^{\mathrm{r}}$ and $\mathbf{v}^{\mathrm{p}}$ only depend on the rotation and position respectively. The motivation of this form is for the easy separation of the rotation-dependent and translation-dependent components. Once we have an approximation that satisfies (D.18) with $\mathbf{v}^{\mathrm{r}}$ and $\mathbf{v}^{\mathrm{p}}$ of

length N_v, the full FIM from N landmarks (D.8) can be written in the form of (D.9):

$$\begin{aligned}\mathtt{I}_{\mathtt{T}_{\mathtt{wc}}} &\overset{(D.18)}{\approx} \sum_{i=1}^{N} (\mathbf{v}^{\mathrm{r}})^{\top} \mathbf{v}_i^{\mathrm{p}} \mathtt{I}_i = \sum_{i=1}^{N} \mathrm{diag}_6((\mathbf{v}^{\mathrm{r}})^{\top} \mathbf{v}_i^{\mathrm{p}}) \mathtt{I}_i \\ &= \mathrm{diag}_6((\mathbf{v}^{\mathrm{r}})^{\top}) \sum_{i=1}^{N} \mathrm{diag}_6(\mathbf{v}_i^{\mathrm{p}}) \mathtt{I}_i \\ &= \mathtt{V}_{\mathtt{I}}(\mathtt{R}_{\mathtt{wc}}) \mathtt{C}_{\mathtt{I}}(\mathbf{t}_{\mathtt{wc}}),\end{aligned} \tag{D.19}$$

where

$$\mathtt{V}_{\mathtt{I}}(\mathtt{R}_{\mathtt{wc}}) \triangleq \mathrm{diag}_6((\mathbf{v}^{\mathrm{r}})^{\top}), \tag{D.20}$$

$$\mathtt{C}_{\mathtt{I}}(\mathbf{t}_{\mathtt{wc}}) \triangleq \sum_{i=1}^{N} \mathrm{diag}_6(\mathbf{v}_i^{\mathrm{p}}) \mathtt{I}_i. \tag{D.21}$$

$\mathrm{diag}_n(\mathtt{A})$ denotes a diagonal matrix by repeating $\mathtt{A}$ by n times on the diagonal, $\mathtt{V}_{\mathtt{I}}(\mathtt{R}_{\mathtt{wc}})$ is a $6 \times 6N_v$ matrix and only depends on the rotational component of the pose, and $\mathtt{C}_{\mathtt{I}}(\mathbf{t}_{\mathtt{wc}})$ is of size $6N_v \times 6$ and only depends on the positions of the camera and the landmarks. We refer to $\mathtt{C}_{\mathtt{I}}(\mathbf{t}_{\mathtt{wc}})$ as the *information kernel* at $\mathbf{t}_{\mathtt{wc}}$ in the rest of the paper.

There are possibly different ways of approximating the visibility function in the form of a vector dot product (D.18). Next, we present two approaches that we explore in this paper.

Polynomial

In the previous conference version [321], we used a quadratic function of $\cos\theta$ to approximate the simplified visibility function. Similarly, we can use a dth-order polynomial:

$$v(\theta) \approx k_d \cos^d\theta + k_{d-1}\cos^{d-1}\theta + \ldots + k_0, \tag{D.22}$$

$$\cos\theta = (\mathtt{R}_{\mathtt{wc}}\mathbf{e}_3)^{\top}\left(\frac{\mathbf{p}^0}{\|\mathbf{p}^0\|_2}\right), \tag{D.23}$$

where $\mathbf{p}^0 = \mathbf{p}^{\mathtt{w}} - \mathbf{t}_{\mathtt{wc}}$. Finding out the polynomial coefficients can be done by fitting the polynomial model into the simplified visibility function (D.17). Then, with $\mathtt{R}_{\mathtt{wc}}\mathbf{e}_3 = [z_1, z_2, z_3]$ and $\frac{\mathbf{p}^0}{\|\mathbf{p}^0\|_2} = [p_1, p_2, p_3]$, (D.22) can be written as:

$$\begin{aligned}v(\theta_i) &\approx \sum_{p=0}^{p=d} k_p (z_1 p_1 + z_2 p_2 + z_3 p_3)^p \\ &= (\mathbf{v}^{\mathrm{r}}(z_1, z_2, z_3))^{\top} \mathbf{v}^{\mathrm{p}}(p_1, p_2, p_3),\end{aligned} \tag{D.24}$$

by separating $\{z_i\}_{i=1}^3$ and $\{p_i\}_{i=1}^3$ in each term of the polynomial expansion. (D.24) satisfies the form (D.18), and setting $p = 2$ results in the quadratic approximation in our previous work [321]. The length of $\mathbf{v}^{\mathrm{r}}$ and $\mathbf{v}^{\mathrm{p}}$, which determines the time and space complexities of the map representation (see Section D.6.1), will be the sum of the number of the terms in a series of trinomial expansions

$$N_v = \sum_{p=0}^{p=d} (p+2)(p+1)/2. \tag{D.25}$$

GP Regression

Given a landmark $\mathbf{p}^{\mathtt{w}}$, we can also use a GP to regress the visibility of it in a camera pose $\mathtt{T_{wc}}$. Since we simplify the visibility as (D.17), the visibility of a landmark does not depend on the full camera rotation $\mathtt{R_{wc}}$ but the direction of the optical axis $\mathbf{z} = \mathtt{R_{wc}}\mathbf{e}_3$, which we use as the input for the GP. The motivation of using GP is that the regressed mean value in (D.4) is in the form of a matrix product and thus linear in terms of its components.

In particular, we sample N_s poses with the same position as $\mathtt{T_{wc}}$ but different rotations $R^s = \{\mathtt{R}_{wc,g}\}_{g=1}^{N_s}$. The corresponding orientations of the optical axis are denoted as $Z^s = \{\mathbf{z}_g^s\}_{g=1}^{N_s}$. We then compute the visibility of the landmark at these poses according to (D.17), denoted as $\mathbf{v}^s = [v_1, v_2, \ldots, v_{N_s}]^\top$. Given a kernel function defined for two unit 3D vectors $k(\mathbf{z}_i, \mathbf{z}_j)$, the visibility of the landmark at $\mathtt{T_{wc}}$ is

$$v(\mathtt{T_{wc}}, \mathbf{p}_i^w) = \mathtt{K_z} \mathtt{K}_{\mathbf{z}}^s \mathbf{v}^s, \tag{D.26}$$

where $\mathtt{K_z}$ and $\mathtt{K}_{\mathbf{z}}^s$ are of $1 \times N_s$ and $N_s \times N_s$, and

$$\mathtt{K_z}(1, g) = k(\mathbf{z}, \mathbf{z}_g^s),\ \mathtt{K}_{\mathbf{z}}^s(g_i, g_j) = k(\mathbf{z}_{g_i}^s, \mathbf{z}_{g_j}^s), \tag{D.27}$$

for $g, g_i, g_j \in [1, 2, \ldots, N_s]$. It can be seen that (D.26) satisfies the form (D.18) by observing $\mathbf{v}^{\mathrm{r}} = \mathtt{K_z}$ and $\mathbf{v}^{\mathrm{p}} = \mathtt{K}_{\mathbf{z}}^s \mathbf{v}^s$, where $\mathtt{K_z}$ only depends on the $\mathtt{R_{wc}}$ and the sampled rotations [2], and $\mathtt{K}_{\mathbf{z}}^s$ only depends on the sampled rotations, $\mathbf{t}_{\mathtt{wc}}$ and $\mathbf{p}^{\mathtt{w}}$. The length of $\mathbf{v}^{\mathrm{r}}$ and $\mathbf{v}^{\mathrm{p}}$ is the same as the number of GP samples $N_v = N_s$.

Since our goal is to have a collective term for different landmarks, the same sampled rotations are used for each landmark to have the same $\mathtt{K_z}$ and thus the same $\mathtt{V_I}(\mathtt{R_{wc}})$ in (D.18). As for the kernel function, we use the squared exponential function, adapted for 3D vectors:

$$k_{\mathrm{SE}}(\mathbf{z}_1, \mathbf{z}_2) = \sigma^2 \exp(-\frac{\|\mathbf{z}_1 - \mathbf{z}_2\|_2^2}{2l^2}), \tag{D.28}$$

[2] Technically, $\mathtt{K_z}$ does not depend on the position only when the kernel function $k(\mathbf{z}_i, \mathbf{z}_j)$ is stationary, *i.e.* only related to the difference between $\mathbf{z}_1$ and $\mathbf{z}_2$, which is the case in our method.

Regarding the hyperparameters, the noise parameter is fixed to $\sigma^2 = 1e-10$, and the length scale l is optimized following the standard approach of maximizing the marginal likelihood [214, p. 112]. The training data is generated by calculating the simplified visibility (D.17) the sampled rotations for a set of random landmarks.

D.5.3 Discussion

Visibility Approximation We arrive at the convenient form (D.9) at the cost of introducing discrepancy between the exact visibility model and the visibility approximations. The discrepancy consists of two parts: 1) the difference between the sigmoid visibility function (D.17) and the exact one (D.16) 2) the difference between the (linearly) separable visibility models (D.22) (D.26) and the sigmoid function. We will study in details about the impact of the discrepancy in the experimental part Section D.7.1.

GP and Spherical Interpolation The GP visibility approximation in (D.26) is a model that regressed the scalar visibility over an sphere (*i.e.* unit vectors for optical axis orientations). In terms of GP regression, it is essentially a weighted sum (weights determined by the kernel function) of the values at different samples (*e.g.* $\mathbf{v}^s$ in (D.26)) and thus is often seem as an interpolation method (also known as *kriging*). Therefore, the aforementioned GP visibility approximation is tightly related to spherical interpolation (see *e.g.* [220, 40]).

Polynomial vs. GP Both the polynomial and GP approximations offer scalability: increasing the polynomial order and the number of GP samples should in principle improve the approximation accuracy but increase the memory footprint and calculation time (see Section D.7.1). In comparison with the polynomial model, GP offers finer granularity for controlling the model complexity, *i.e.* the length N_v of the vector $\mathbf{v}^{\mathrm{r}}$ and $\mathbf{v}^{\mathrm{p}}$ in (D.19). For the GP model, N_v is equal to the number of GP samples and thus can be any positive integer. For the polynomial model, it can only be of specific values determined by (D.25) (*e.g.* $10, 20, 35, 56$ for $d = 2, 3, 4, 5$ respectively). In terms of implementation, for the polynomial approximation, it is alway possible to collect all the terms from the trinomial expansions and write out (D.24) for any order d, but the process is increasingly complicated for high order polynomials. In contrast, increasing the number samples in GP is trivial.

D.6 Building a Map for Perception-aware Planning

D.6.1 The Fisher Information Field

Representation, Query and Update

Using the formulation (D.19), (D.20) and (D.21), we propose a volumetric representation, namely the *Fisher Information Field* (FIF) for perception-aware planning. In particular, after discretizing the space of interest into voxels, we compute the information kernels $\mathtt{C}_\mathtt{I}(\cdot)$ at the centers of the voxels (from all the 3D landmark), and store each kernel in the corresponding voxel. Then, when the information of a certain pose is queried, the related kernels (by nearest neighbor or interpolation) are retrieved, and (D.19) is used to recover the information in constant time. The method is illustrated in Fig. D.1. Once the field is built, the query of the information for an arbitrary pose only requires the linear operation of the related kernels instead of checking all the points in the point cloud, which is the key advantage of the proposed method.

Field Update In practice, especially during the exploration of an unknown environment, new landmarks may be added and existing ones deleted over time, and our representation needs to adapt to such changes. Fortunately, since the kernel (D.21) is in the form of the summation of components calculated from each landmark independently, adding/deleting the contribution of a landmark can be done trivially by adding/subtracting the corresponding components from existing kernels. However, in a non-trivial scenario, determining whether a landmark is matchable from a certain pose is not a trivial tasks and might require more information of the environment. In this case, an offline mapping process, as shown in Section D.7.2, is preferable.

Complexity and Trace Kernel

The constant query time comes at the cost of extra memory. In particular, the information kernel $\mathtt{C}_\mathtt{I}(\cdot)$ at each location consists of $36N_v$ floats, where N_v is the length of vector in the dot product approximation (D.18). Admittedly, the size of storage needed is non-negligible (*e.g.* 360 float numbers for quadratic visibility model and 1800 for GP with 50 samples), and it increases linearly with the number of voxels in the field. But the memory footprint is still acceptable in practice, as we will show in Section D.7.2 in a realistic setup.

Note that the aforementioned information representation can be used to recover the full approximated information matrices (6×6). However, in the cost (D.7), only one scalar metric $s(\cdot)$ is needed in the overall cost for planning. This brings the possibility of reducing the memory usage by directly expressing one specific metric instead of the full information matrix. Out of different metrics often used with the Fisher information [208,

Ch. 6&9], the T-optimality criterion, which is the matrix trace, is especially suitable (*i.e.* a linear function) for this purpose. In particular, taking the trace of the approximated information (D.19), we can arrive at the following form

$$\mathrm{Tr}(\mathtt{I}_{\mathtt{T_{wc}}}) \approx \sum_{i=1}^{N} (\mathbf{v}^{\mathrm{r}})^{\top} \mathbf{v}_i^{\mathrm{p}} \,\mathrm{Tr}(\mathtt{I}_i) = \mathtt{V}_{\mathrm{Tr}}(\mathtt{R_{wc}}) \mathtt{C}_{\mathrm{Tr}}(\mathbf{t}_{\mathtt{wc}}) \tag{D.29}$$

where

$$\mathtt{V}_{\mathrm{Tr}}(\mathtt{R_{wc}}) \triangleq (\mathbf{v}^{\mathrm{r}})^{\top}, \tag{D.30}$$

$$\mathtt{C}_{\mathrm{Tr}}(\mathbf{t}_{\mathtt{wc}}) \triangleq \sum_{i=1}^{N} \mathbf{v}_i^{\mathrm{p}} \,\mathrm{Tr}(\mathtt{I}_i). \tag{D.31}$$

We call (D.31) the *trace kernel*. Notably, $\mathtt{V}_{\mathrm{Tr}}(\cdot)$ and $\mathtt{C}_{\mathrm{Tr}}(\cdot)$ are both of size $N_v \times 1$, reducing the memory usage by a factor of 36 compared with the information kernel (D.21). The trace kernel can be used in the same volumetric mapping framework mentioned above, but only requires N_v float numbers for one voxel (at the cost of losing certain information contained in the full FIMs).

D.6.2 Integration in Motion Planning

Conceptually, integrating FIM in motion planning is straightforward. For instance, we can define a certain value as the threshold to determine whether a pose can be localized, or we can adding the determinant/trace of the FIM as an additional term in the cost function. However, there are several problems with the naive approach. First, the thresholds are less intuitive to choose (*e.g.* than the distance threshold for collision avoidance). Second, since different information metrics for the same FIM have very different values, the thresholds or weights for these information metrics have to be chosen separately, which makes the parameter tuning complicated. More importantly, this also makes a fair comparison difficult. For example, if the thresholds/weights using two metrics are chosen differently, how can we tell a worse performance is due to the metric or the lack of tuning? The same problem exists for using different types of FIFs as well, since the information metrics from them are also different due to approximation.

To overcome the above problems, we propose a unified approach of defining thresholds or costs for the metrics from FIM. Instead of defining the thresholds for the information metrics directly, we compute the thresholds from certain specifications of the landmarks:

Information Threshold For a certain pose, we assume that if there are M landmarks in the camera FoV that are within $d_{\min}$ to $d_{\max}$ distance to the camera, the pose is considered to be able to be localized. We first randomly generate several sets of landmarks

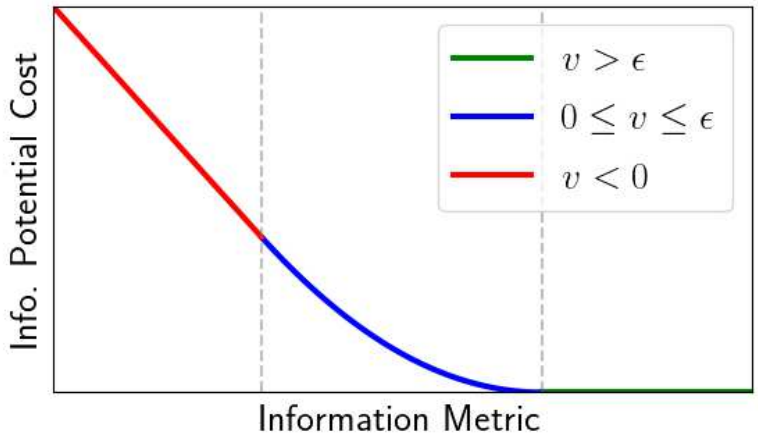


Figure D.3 – Illustration of the information potential cost in (D.32).

that satisfy the criteria. Then, for a certain information metric from a certain map representation (FIF or the point cloud), we first construct the map representation if necessary (*i.e.* FIF), calculate the information metric from the map representation and use the average value as the threshold for this combination of information metric and map representation. Therefore, given the *same* landmark specifications (M, $d_{\min}$ and $d_{\max}$), the thresholds for different combinations of information metrics and map representations are calculated automatically.

Information Potential Cost For optimization-based methods, we further define an information potential cost similar to the collision potential cost in [329]. The intuition is that we no longer care about the improvement of the localization quality after it has reached a certain level. Specifically, assuming that the pose can be estimated well enough with M landmarks in the FoV from $d_{\min}$ to $d_{\max}$ distance to the camera, we first calculate the threshold ϵ_{FIM} as mentioned before. We then define the information potential cost c_{FIM} as

$$c_{\text{FIM}}(v_{\text{FIM}}) = \begin{cases} 0, & v_{\text{FIM}} > \epsilon_{\text{FIM}} \\ k_q(v_{\text{FIM}} - c_{\text{FIM}})^2, & 0 \leq v_{\text{FIM}} \leq \epsilon_{\text{FIM}} \\ k_l \cdot v_{\text{FIM}} + b_l, & v_{\text{FIM}} <= 0 \end{cases} \tag{D.32}$$

where v_{FIM} is the information metric. $k_q > 0$ is chosen empirically. k_l and b_l are calculated to guarantee the continuity in both c_{FIM} and its derivative at $c_{\text{FIM}} = 0$ after k_q has been chosen. An illustration of the information potential cost is shown in Fig. D.3.

With our approach, we can specify the thresholds for different metrics/map representations in the same way and, in optimization-based motion planning, use the same weight for the cost related to FIM. This greatly simplifies the experiments in Section D.7.2 and makes

		PC	Q-I	Q-T	G-I-30	G-I-70	G-I-150	G-T-30	G-T-70	G-T-150
Build	t_{build} (sec)	-	7.34	2.96	17.84	41.10	124.74	10.93	26.48	82.72
	Mem. (MB)	0.02	58.50	1.62	135.00	315.00	675.00	3.75	8.75	18.75
Query (us)	FIM	97.3	0.4	-	0.9	2.7	4.7	-	-	-
	det	98.2	3.0	-	5.7	12.1	24.9	-	-	-
	λ_{min}	102.9	6.3	-	8.6	14.7	27.6	-	-	-
	Trace	97.7	1.6	0.6	3.9	10.0	23.3	1.1	1.9	3.4

Table D.1 – Time and memory required for building different types of FIF, and the time per query for full FIM, determinant (det), smallest eigen value (λ_{min}) and the trace of FIM (Trace). By pre-computation, different types of FIF are significantly faster than using point clouds (PC) at query time, which is important for online applications such as motion planning.

.

	Q-I	G-I-30	G-I-50	G-I-70	G-I-100	G-I-120	G-I-150
Diff. (%)	66.44	9.97	8.76	8.86	8.54	8.63	8.18

Table D.2 – Relative difference with respect to the exact FIM (computed from the point cloud and the exact pinhole camera model) in terms of the Frobenius norm (see (D.33)).

the results using different information metrics and map representations comparable.

D.7 Experiments

We implemented the proposed Fisher Information Field in C++. We used the library from [198] for the voxel hashing functionality. The voxels are organized in a hierarchical manner, where N^3 voxels forms a *block* as a mid-level data structure.

Next, we present both quantitative and qualitative results using our implementation. Specifically, we aim to answer the following questions:

- How do different visibility approximations affect the efficiency and accuracy of FIF?
- How can FIF be used with different motion planning algorithms? What are the benefits?
- How do FIF compare with point clouds in terms of the computation of FIM and perception-aware planning?

Moreover, we also present qualitative results of building FIF from the output a visual-inertial odometry pipeline incrementally, showing the possibility of using our method in previous unknown environments.

D.7.1 Simulation

Ideally, constructing FIF requires the ability to determine which landmarks can be matched from a certain viewpoint. It is, however, a difficult task in non-trivial setups: it is coupled with the detailed scene structure (*e.g.* occlusion) and the method for establishing correspondences (*e.g.* feature-based or direct methods, types of features). Since the focus of the paper is a representation for FIM that allow efficient query, we first performed evaluation in a simplified simulated environment, where we assumed the correspondence with respect to a certain landmark can be established as long as the landmark is in the FoV. Experiments in a more realistic environment are shown in Section D.7.2.

In the following experiment, we generated 1000 random landmarks in a 10m $\times$ 10m $\times$ 5m area. We further built the proposed FIF within a smaller 9m $\times$ 9m $\times$ 4m region with the size of a voxel set to 0.5m, resulting in $\sim$ 16000 voxels. A pinhole camera model with 90deg horizontal FoV was used. With this setup, we tested the proposed FIFs using both the information kernel (D.21) and the trace kernel (D.31), with different visibility approximations, namely

- Quadratic approximation that satisfies: 1) the visibility at the center of the FoV is 1.0; 2) and visibility at the boundary of the FoV to 0.5.
- GP with $N_s = 30, 50, 70, 100, 120, 150$ trained to approximate the sigmoid visibility function (D.17) with $k_s = 15$.

We use the notation "V-M(-Ns)" to denote a specific map representation, where V (**Q**uadratic or **G**P) stands for the visibility approximation, M (**I**nformation or **T**race) the field type and Ns the number of samples for GP. The results (FIM and different metrics) computed from the landmarks using the exact pinhole camera model was used as the groundtruth (denoted as "PC").

Complexity and FIM Accuracy

To evaluate the query time and accuracy, we randomly sampled 200 poses within the area where FIF had been constructed. From these poses, the following were tested:

- Compute the full FIM from the nearest voxel (only for FIFs using the information kernels). This is to study the validity of the visibility model, excluding the impact of the voxel size.
- Compute different FIM metrics using the interpolation from the nearby voxels. This is to simulate the practical use cases where the discretization resolution is limited.

Complexity The query times for different settings, along with the time and memory required to build the FIFs are reported in Table D.1. For GP, the cases where $N_s = 50, 100, 120$ are omitted for brevity, since they follow the same trend. In terms of query time, which is the motivation of the proposed method, all types of FIFs took much shorter time per query compared with the point cloud method. On the other hand, the speedup at the query time comes at the cost of additional building time and memory. In terms of different types of FIFs, quadratic approximations are faster and require less memory than GP, and trace kernels are more efficient than information kernels the full FIM in terms of both memory and query time (*c.f.* last row of Table D.1). In addition, increasing the number of samples in GP increases the memory footprint and the query time. Note that the query of det, $\lambda_{\min}$ and Trace are several times more expensive than FIM in our experiment, mainly due to the interpolation mentioned above (*i.e.* accessing up to 8 surrounding voxels).

FIM Accuracy To evaluate the accuracy of the computed FIM $\hat{\mathtt{H}}$ from the proposed information fields, we computed the relative difference with respect to the groundtruth FIM $\mathtt{H}$ calculated using the exact pinhole camera model from the landmarks:

$$e_{\text{FIM}} = \|\hat{\mathtt{H}} - \mathtt{H}\|_F / \|\mathtt{H}\|_F, \tag{D.33}$$

where $\|\|_F$ denotes the Frobenius norm. The results are reported in Table D.2. It can be seen that, even for GP with only 30 samples, the recovered FIM is much more accurate than quadratic approximations. This is likely due to the elimination of the long tail in the quadratic approximation. Moreover, increasing the number of samples in GP in general decreases the difference with respect to the exact FIM. Note that increasing the number of GP samples infinitely will *not* reduce the error to zero, because our GP visibility models are designed to approximate a simplified camera model instead of the exact one.

Relative Measures

In terms of motion planning, it is also of interest to check the relative values of the queried information metrics, in addition to the FIM errors mentioned above. To this end, we performed a series of experiments about how the information metrics change with different poses. Note that we report only the results for the information kernels, since the trace kernels will yield the same results as querying the trace of FIM from the information kernels.

Optimal Views In this experiment, we computed the optimal views at 200 sampled positions. Specifically, for each position, we densely sampled the rotations, calculated different information metrics at the poses consisting of the position and the sampled rotations, and determined the optimal view as the one that maximizes certain information

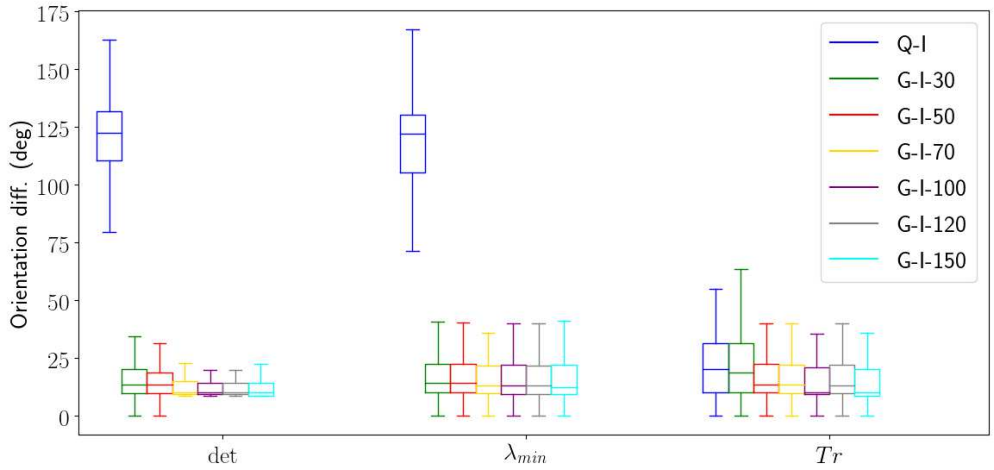


Figure D.4 – The differences of optimal views determined by different types of FIFs with respect to the ones determined by the point cloud. Results using determinant, the smallest eigenvalue and the trace of the FIM are shown.

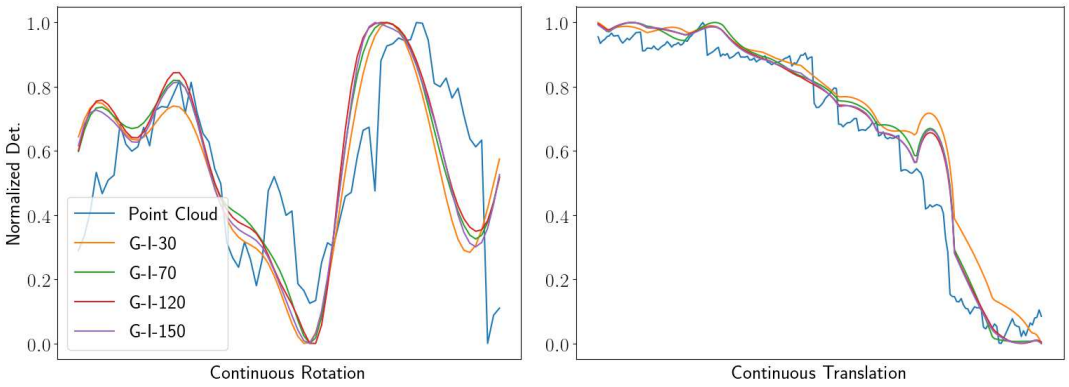
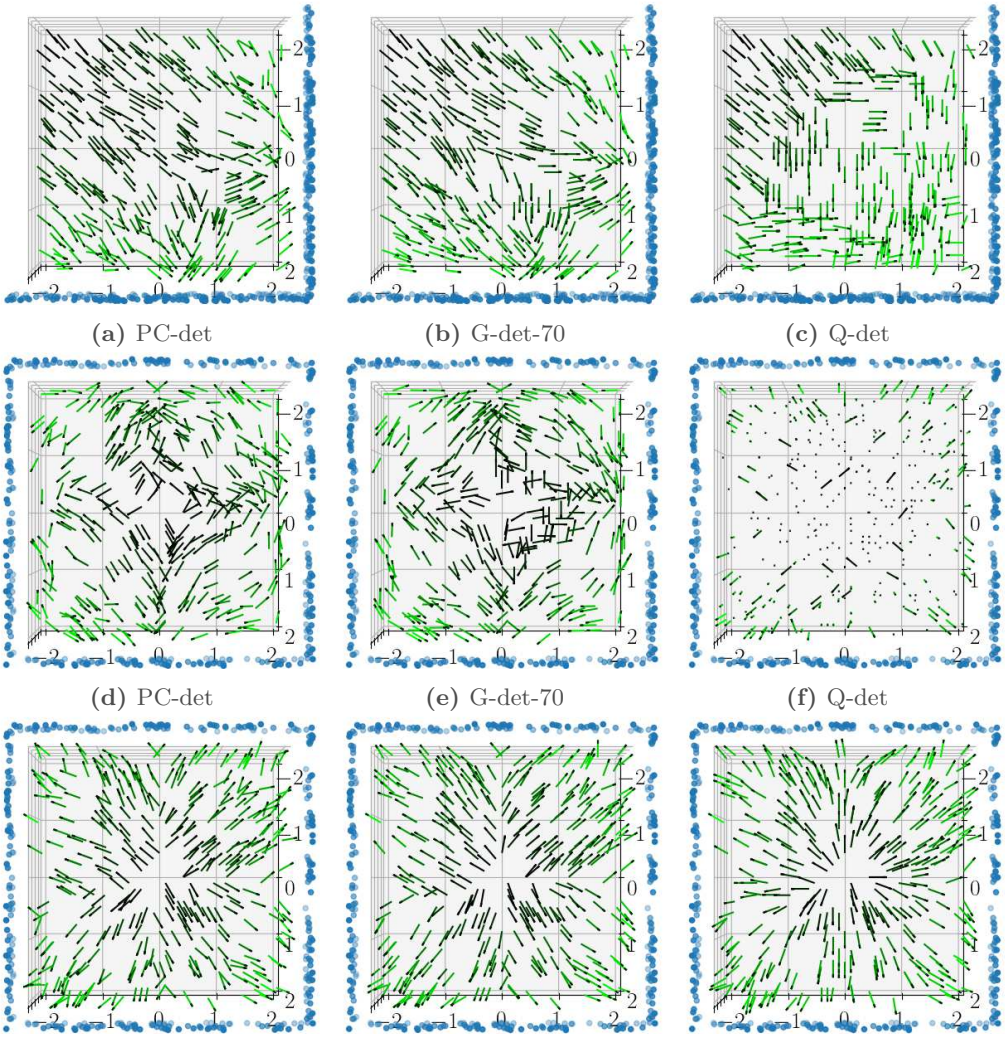


Figure D.5 – The evolution of information metric for continuous pose change.

metric. We calculated the difference of the optimal views determined from the FIFs with respect to the ones determined by computing information metrics from the point cloud.

The differences between the optimal views calculated from FIFs and the point cloud, using different information metrics, are shown in Fig. D.4. The GP approximations are in general more accurate in determining the optimal views than the quadratic approximation. Increasing the number of samples in GP leads to better results, but seems to saturate at around $N_s = 70$. Interestingly, for quadratic approximations, the optimal views determined using the trace are much more accurate than the ones using the determinant and the smallest eigenvalue. We further computed the optimal views in simulation environments with specific landmarks layout, and several examples are shown in Fig. D.6. The results are consistent with the aforementioned experiment (using randomly generated



(a) PC-det (b) G-det-70 (c) Q-det

(d) PC-det (e) G-det-70 (f) Q-det

(g) PC-Trace (h) G-Trace-70 (i) Q-Trace

Figure D.6 – Visualization of the information field in simulated scenes for the trace and determinant metrics. Blue circles are 3D landmarks, and each line segment stands for one optimal view direction. Brighter color means better localization quality. **Left**: point cloud with the exact camera model; **Middle**: GP approximation ($N_s = 70$); **Right**: quadratic approximation. Note the obvious failure case (f) for the combination of the quadratic model and the determinant, where the optimal views are vertical to the xy plane.

landmarks). Intuitively, the optimal view at a position should point to the area where the landmarks are concentrated and close to the position, which is the case for the results from the point cloud and GP. Using quadratic approximation with the determinant shows larger discrepancy with respect to the point cloud, and even counter-intuitive results (Fig. D.6 (f)).

Smoothness In this experiment, we selected two continuous trajectories inside the

FIF					ESDF		# landmarks	
Voxel size	Q-I mem.	Q-T mem.	GP-I mem.	GP-T mem.	Voxel size	Mem.	r2-a20	r1-a30
1.0m	108M	3M	578M	17M	0.1m	485M	3470	1445

Table D.3 – Specifications of different maps for the photorealistic simulation environment ($\sim 50\text{m} \times 30\text{m} \times 9\text{m}$). The memory of the FIF does not change with the number of landmarks, and thus only one number is listed.

FIF: 1) pure rotation around a fixed axis; 2) pure translation along a straight line. We then calculate the information metrics along the two trajectories. The evolution of the determinant (normalized to $0-1$ for visualization) for several FIFs and the point cloud is plotted in Fig. D.10. Other information metrics also exhibited similar behaviors, and thus the results are omitted. It can be seen that, while the overall trend from the FIFs are similar to the point cloud, the results from the FIFs are obviously smoother. This property is especially important for optimization-based motion planning, as the optimization is less likely to be stuck in local minimums. This is due to fact that the proposed visibility approximations are differentiable, whereas the actual visibility model is not.

Summary and Discussion

In the above experiments, we thoroughly tested different visibility approximations with both information and trace kernels, which were proved to have much shorter query time than using the point cloud directly. In general, the accuracy of the proposed FIF increases with more expensive visibility approximations (quadratic vs. GP, increasing number of GP samples). This indicates the scalability of the proposed method: one can choose different types of visibility approximations considering the required performance and the computational resource at hand. In addition, the trace kernels proves to be significantly more efficient than the information kernels, which might be of interest for computationally constrained platforms.

D.7.2 Motion Planning

We further applied the proposed FIF to different motion planning algorithms in photorealistic simulation. The experimental setup is described below.

Photorealistic Simulation We used the Nvidia Issac simulator [3] for photorealistic simulation. We only used the rendering capability of the simulator, which is done by

[3] https://developer.nvidia.com/isaac-sim

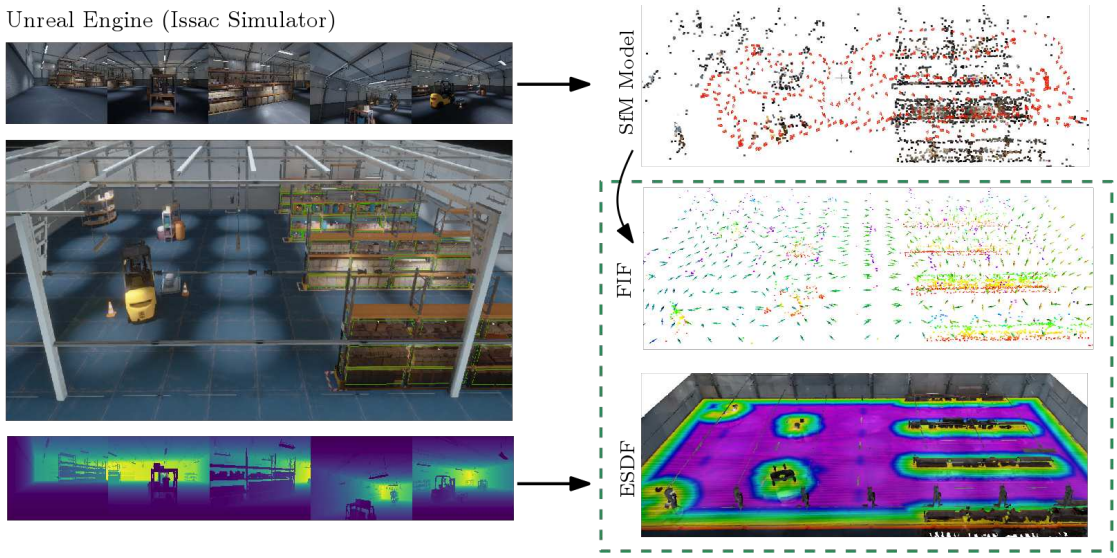


Figure D.7 – Creating different maps from the photorealistic simulation for the motion planning experiments. Images and depth maps were rendered from the Unreal Engine and were used, together with the camera poses, to build a SfM model (via COLMAP) and an ESDF map (via Voxblox) respectively. Then we built the proposed Fisher Information Field from the SfM model. The FIF and ESDF were then used in different perception-aware motion planning experiments.

Unreal Engine[4]. Rendering images at desired camera poses was achieved via integrating UnrealCV [210] with the simulator. The built-in map *warehouse* (see Fig. D.7) was adapted and used in our experiments. The environment is of approximately 50m $\times$ 30m $\times$ 9m.

Planning Algorithms We chose two representative motion planning algorithms: RRT* [126] (implemented in [272]) and trajectory generation for quadrotors [5]. RRT* is a sampling-based method, whereas the trajectory generation for quadrotors relies on nonlinear optimization. We adapted these algorithms to incorporate the information from the proposed FIF, which are described in the following sections.

Prerequisite: Mapping the Environment We first mapped the environment to get different maps. In particular:

- For collision avoidance, we chose to use Euclidean Signed Distance Field (ESDF) implemented in Voxblox [198]. We densely sampled camera poses from the environment and fed the poses and depth to Voxblox to build the ESDF. The dense sampling is not necessary though: a more realistic exploration trajectory could also yield an ESDF that is sufficient for planning.

[4] https://www.unrealengine.com/
[5] https://github.com/ethz-asl/mav_trajectory_generation

	cfg.	No Info.	PC-det	PC-Trace	GP-det	GP-Trace	Quadratic-det	Quadratic-Trace
r2-a20	Bottom	73%	0%X	31%	0%	17%	0%	0%
	Diagonal	0%	0%X	0%	0%	0%	0%	11%
	Top	44%	0%	29%	0%	57%	13%	15%
r1-a30	Bottom	79%	0%X	25%	0%X	15%	0%X	38%
	Diagonal	20%	0%X	75%	0%X	27%	16%X	52%
	Top	89%	0%X	33%	0%	53%	15%	29%

Table D.4 – Failure rates of localizing the rendered images on the shortest path from RRT* using different types of FIFs. The results using two set of 3D points "r2-a20" and "r1-a30" are listed, where the former contains more points than the latter. "No Info." denotes the case where the Fisher information was not considered in RRT*. An "X" denotes that RRT* failed to find a path (*e.g.* due to the lack of landmarks in the environment).

- For building the FIFs, we need sparse landmarks that can be used for localization. For this purpose, we manually control the camera to move around the environment to collect a series of images. We then fed the images and the corresponding poses to COLMAP [250] to build a Structure from Motion (SfM) model. The 3D landmarks were then used to build different types of FIFs. To determine which landmarks are visible from a certain pose, we filtered the landmarks by the difference with respect to the average view direction in the SfM model (similar to the perspective change in [103]) and the depth map described below.

- To determine the visibility of the landmarks more accurately, we densely rendered the depth maps at the camera poses from a regular 3D grid. The depth maps were used to identify the occluded landmarks.

The mapping process and the visualization of different maps are shown in Fig. D.7. In addition, to study the impact of the number of landmarks, we further generated two SfM models and the corresponding FIFs. The first one only contains the landmarks that has less than 2px average re-projection errors and has at least two views with larger than 20deg parallax. The thresholds for the second one were set to 1px and 30deg. The second SfM model contains less but, in principle, more accurate landmarks. The two setups are denoted as "r2-a20" and "r1-a30" respectively. The detailed specifications of the different maps used in our planning experiments are listed in Table D.3.

Workflow and Evaluation To test different motion planning algorithms, we followed the same workflow:

1. run the motion planning algorithm

2. sample poses from the planned motion

3. render images at the sampled poses

4. localize the rendered images in the SfM model using the image registration pipeline from COLMAP

Whether the rendered images can be successfully registered and the localization accuracy are used as the evaluation metric about how much the motion planning algorithm respect the perception quality.

Tested FIFs Both GP and quadratic visibility approximations were tested, using both the information and trace kernels. For GP visibility approximation, we chose $N_s = 70$ based on the simulation results. Using $N_s = 50$, however, produced similar results in our experiment. As for the information metrics, we experimented with the determinant (calculated from the information kernel) and the trace (calculated from the trace kernel). The information metrics calculated from the point cloud were used as the baseline. The same notation as in the Section D.7.1 is used.

RRT*

The state space of RRT* was set to 4 DoF: position and yaw. It spanned the horizontally range of the *warehouse* and was set to 2m in height. The path lengths in the state space, in terms of position and yaw, were used as the objective to minimize, and the weights of the two costs were chosen experimentally. Both ESDF and FIF were used to check the state validity. In particular, the minimal distance to obstacles (from ESDF) was set to 2m, and the information threshold (see Section D.6.2) was calculated by assuming that at least 10 landmarks in 1m to 3m meter range in the FoV of the camera are needed to have a valid localization. The planner was set to run for 500s, regardless of whether a valid path was found. Three planning settings (*i.e.* start and end states) were tested, denoted as *bottom*, *diagonal* and *top* (see Fig. D.9).

Localization Failure Rate We rendered images from the poses of the vertices on the shortest path in the final tree spanned by RRT* and registered the images in the SfM model. Since we used the FIF as a state validity checker, we computed the percentages of the images that failed to be localized, show in Table D.4. First, the failure rates are higher in "r1-a30", which contains less landmarks for localization, and there are more cases where RRT* failed to find a valid path as well, due to the stricter perception constraint. Second, in general, considering Fisher information in RRT* helps to reduce the failure rates, which can be seen by comparing "No Info." with the other columns. Third, "PC-det" and "PC-Trace" both use the exact camera model, but the later shows worse performance (in some cases even worse then "No. Info"). This indicates that the trace of FIM, despite of its efficiency, may be a weaker indicator of the localization/pose estimation quality than the determinant. This is also validated by the worse performance of the trace with both GP and quadratic visibility approximations. Fourth, in terms of the determinant, "PC-det" should give the best performance (due to the use of the exact

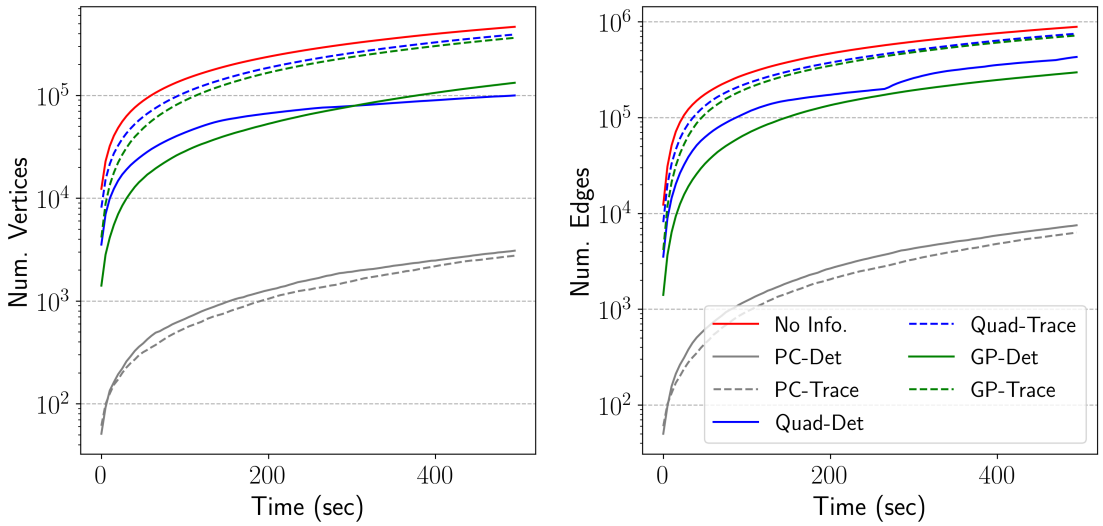


Figure D.8 – The number of vertices and edges in the RRT* tree with respect the planning time. The plot is generated for the *bottom* planning configuration in "r2-a20". Note that the y-axis is in log_{10} scale.

camera model), which is validated by the 0 failure cases. "'PC-det' also has highest of number of experiments where RRT* reported no solution, which indicates using "PC-det" puts a stricter criterion about whether the image from a pose can be localized. Finally, in terms of different visibility approximations, GP outperforms the quadratic model when the determinant of the FIM was used. Notably, GP with determinant is the only FIF that has 0 failure cases. On the other hand, the comparison of "GP-Trace" and "Q-Trace" is inconclusive. In Fig. D.9 we plot the final paths of "GP-det" and "No Info." as a qualitative example. Intuitively, with the information from the FIF, RRT* prefers view directions towards area with more landmarks.

Efficiency As for the efficiency of different map representations, we plot the number of vertices and edges in the tree spanned by RRT* with respect to the time spent. Since the results of different planning configurations are similar, we show one example in Fig. D.8. All types of FIFs tested are at least one order of magnitude faster in terms of the number of vertices that can be explored for the same time. In addition, the quadratic model is more efficient than GP, and computing the trace from the FIFs is faster than computing the determinant. Qualitatively, in Fig. D.8, we plot the vertices for the first 10 iterations for the *bottom* planning configuration in both "r2-a20" and "r1-a30" together with the landmarks. Comparing the left column ("GP-det") and the right column ("PC-det"), the vertices explored using the proposed FIF cover a larger area with a higher sampling density than using the point cloud. Comparing the first row with the second, we can see that decreasing the number of landmarks effectively reduce the region where the poses are considered to be able to be localized. This is also potentially useful to identify the "perception traps" in a given environment.

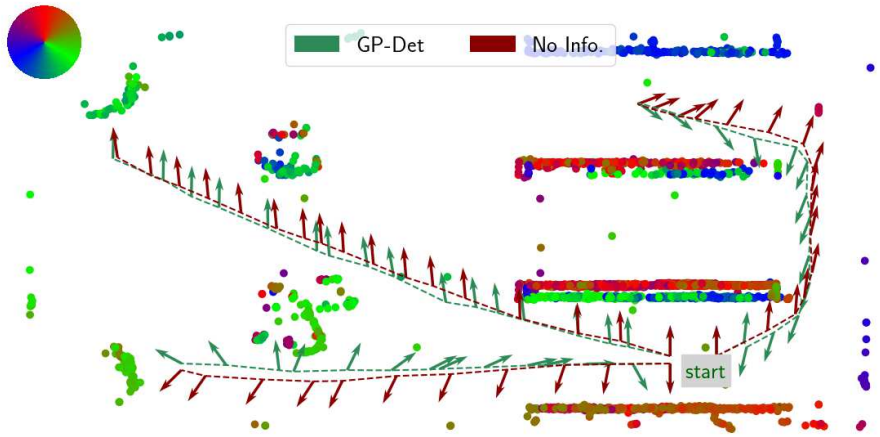
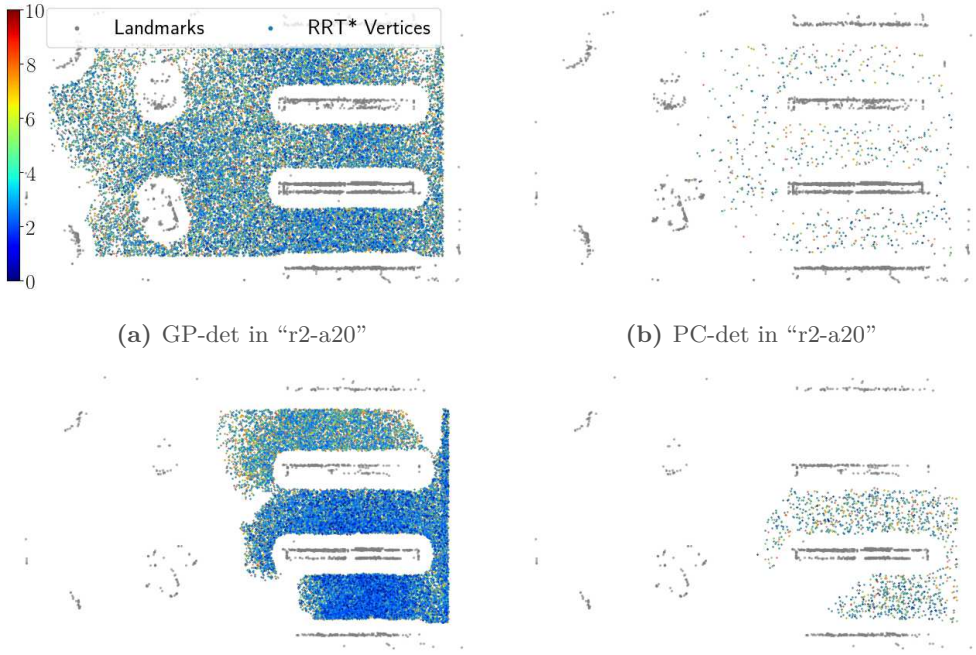


Figure D.9 – Example RRT* paths in "r2-a3". The colored points are the landmarks, where the average observing directions (from the images in the SfM model) are color coded according to the color wheel on the top left. The arrows on each path indicate the camera view direction.



(a) GP-det in "r2-a20"

(b) PC-det in "r2-a20"

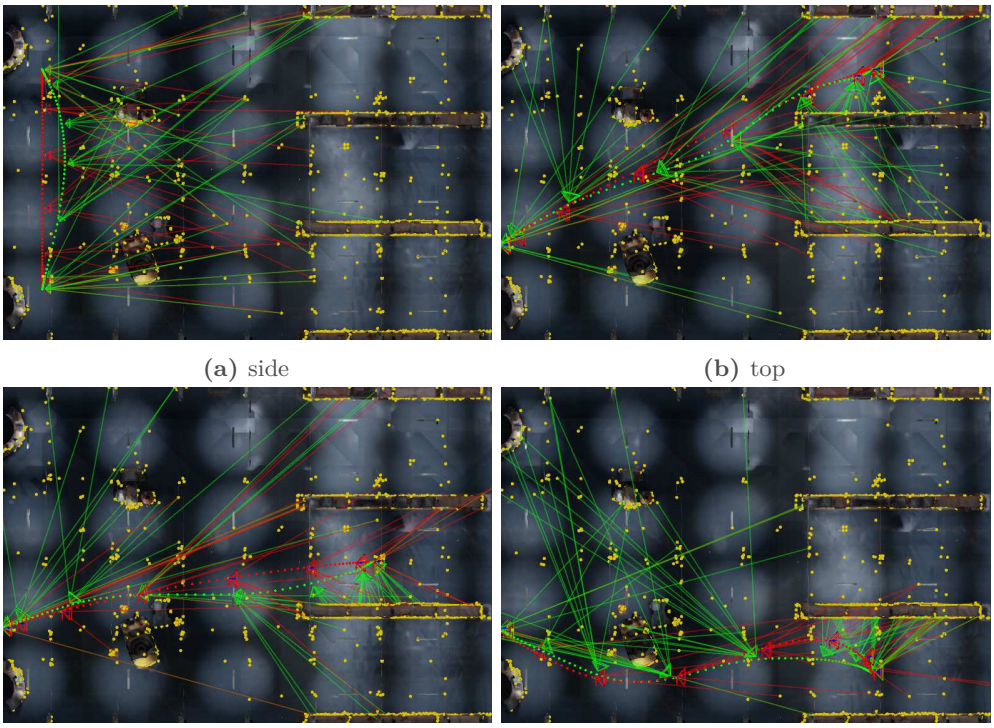
(c) GP-det in "r1-a30"

(d) PC-det in "r1-a30"

Figure D.10 – The RRT* vertices that were explored for the first 10 iterations. The color of a RRT* vertices indicate at which iteration the vertice was added to the tree (shown in the colorbar). The gray points denote the landmarks in the environment.

Trajectory Optimization

Following the standard practice [173], we used a piecewise 4 DoF polynomial (5 segments) to represent a continuous-time trajectory for quadrotors. Given start and end states,



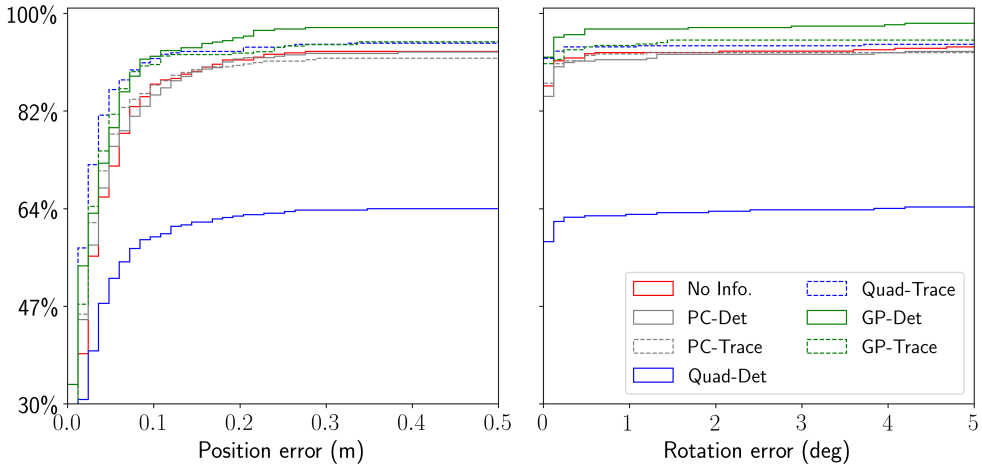
(a) side **(b)** top

(c) middle **(d)** bottom

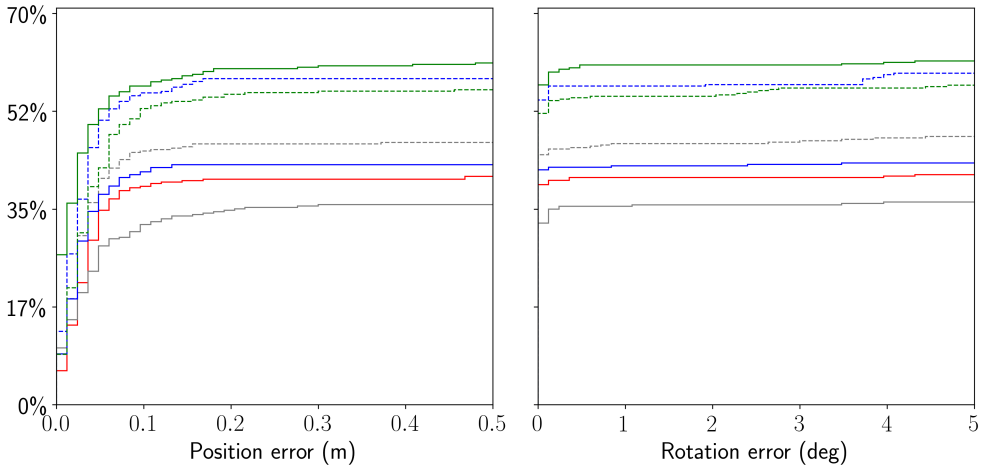
Figure D.11 – The comparison of the optimized trajectories using the proposed Fisher Information Field (green) and without considering the Fisher information (red). The poses sampled at a constant time interval are visualized as points of the corresponding color. The yellow points are the landmarks for localization, and the lines denote the potentially matchable landmarks considered in the trajectory optimization. GP visibility approximation and the determinant of the FIM are used. All trajectories start from the left part of the scenario. Note that only the top views are shown, but the trajectories are optimized in the 3D space. For example, in *side*, the green trajectory is higher than the red one, favoring the landmarks located on the ceiling.

we first initialized the trajectory using [216] and used it as an initial value for further nonlinear optimization. In the nonlinear optimization, we considered the position and yaw dynamic cost of the quadrotors, the collision potential cost as in [196, 199], and the information potential cost from the FIFs. Specifically, the dynamic costs were calculated in closed-form, derived from the polynomial coefficients, and the collision potential cost and the information potential cost were calculated as the integral along the trajectory, with a sampling interval of 0.1sec. For the information potential cost (D.32), 200 landmarks within 0.3m to 1.0m were considered sufficient and used to calculated ϵ_{FIM} for different information metics and map representations. The weights among these costs were chosen experimentally but kept fixed for all the experiments. The optimization was modeled as a general unconstrained optimization problem using Ceres[6], and the default optimizer parameters were used. For each trajectory optimization, we let the optimizer run for maximum 100 iterations. Similar to the RRT* experiment, we chose four sets of

[6] http://ceres-solver.org/



(a) r2-a20



(b) r1-a30

Figure D.12 – Cumulative histograms of the localization error of the images rendered from the optimized trajectories using different map representations in "r2-a20" and "r1-a30". Each point (X, Y) on the curve denotes there are $X\%$ images that were able to be localized below Ym (or Ydeg) error.

start and end states, namely *top*, *middle*, *bottom* and *side* (see Fig. D.11). The duration of the trajectory was set to 10sec.

Localization Accuracy After the optimizer converged or reached the maximum number of iterations, we sampled poses from the trajectory by 0.1sec time interval and rendered images from these poses. Since the FIM was used as an optimization objective, we calculated the localization error with respect to the true poses for evaluation. The cumulative histograms of the position and rotation errors, aggregated over all planning settings, are shown in Fig. D.12. Similar to the RRT* experiment, we observed that decreasing the number of landmarks reduced the localization accuracy, and the benefit

	r2-a20		r1-a30	
	$\overline{\text{iter.}}$	$\overline{\text{time}}$ (s)	$\overline{\text{iter.}}$	$\overline{\text{time}}$ (s)
No Info.	59.0	0.057	59.0	0.054
PC-det	12.8	44.49	9.0	12.68
PC-Trace	17.5	65.74	16.3	31.10
GP-det	100.0	1.35	62.5	0.99
GP-Trace	99.0	0.39	93.8	0.40
Quadratic-det	85.8	0.35	61.8	0.26
Quadratic-Trace	90.8	0.13	79.0	0.12

Table D.5 – Average number of iterations and optimization time over all planning settings in the trajectory generation experiment. The maximum number of iterations was set to 100.

of considering the Fisher information becomes more significant, as shown by the larger margin. Compared with "No Info.", considering the Fisher information in general improves the localization accuracy, with a few exceptions. In particular, the quadratic approximation with the determinant of FIM performed significantly worse in "r2-a20" and similar to "No Info." in "r1-a30". Since the relative values matter in optimization, this is consistent with the results in Fig. D.4. Computing information metrics from the point cloud overall shows no obvious improvement with respect to "No Info.". Notably, "PC-det", which was the best performing representation in the RRT* experiments, achieved lower accuracy than not considering the Fisher information at all. We further observed that the optimization using the point cloud tended to terminate prematurely (see Table D.5), which is possibly due to the discontinuity shown in Fig. D.10 caused by the exact camera model. Finally, "GP-det", "GP-Trace" and "Q-Trace" are the best performing map representations, where the GP approximation with the determinant of FIM consistently shows the highest localization accuracy.

Examples of the optimized trajectories (from "GP-det") are shown in Fig. D.11. Intuitively, including the Fisher information in the trajectory optimization force the camera to orient towards and move closer to areas with more landmarks (*e.g.* the shelves), resulting in more matchable landmarks in the camera FoV and higher localization accuracy.

Efficiency The average number of iterations and the optimization time are listed in Table D.5. Among the methods that consider the Fisher information, calculating information metrics from the point cloud takes the most time with the lowest number of iterations, which indicates that the evaluation of the information metrics using the point cloud is far more expensive than the proposed map representations. Moreover, we suspect that the lower number of iterations indicates the the optimization terminated prematurely, considering the lower localization accuracy shown in Fig. D.12. Similar to the results in the RRT* experiment, we observe that using GP is more time-consuming than the quadratic model. The trace kernels, despite the slightly worse localization

accuracy, is very efficient: with the quadratic model, it is only around two times slower than the case where no Fisher information was considered.

Summary and discussion

Different from the simplified simulation in Section D.7.1, we applied the proposed FIF to different motion planning algorithms in a realistic simulation. The images were rendered using a photorealistic game engine, and the localization accuracy was evaluated using mainstream visual localization techniques. It can be seen that, in general, integrating FIF helps improve the localization quality, in terms of the successful rate and localization accuracy. Compared with the standard practice of using the point cloud, our method is at least one order-of-magnitude faster, and the differentiable/smooth visibility approximations additionally bring better performance in the trajectory optimization experiment. Next, we further discuss several aspects of our method.

Generalizability *First*, we would like to highlight that the proposed FIF is not specific to certain motion planning algorithms. In particular, we intentionally chose two representative motion planning algorithms (*i.e.* sampling-based and optimization-based). Moreover, off-the-shelf open source implementations of these algorithms were used through their existing interface, without specific customization for our map representation. *Second*, it is relatively easy to build a specific map with "perception traps" to show the benefit of perception-aware motion planning algorithms compared with standard ones (as in our previous work [323]). However, in the above experiment, we tried to avoid artificial corner cases to evaluate our method in a relative realistic setup. The improved performance indicates the proposed method is a useful tool in general.

Offline Mapping The process of constructing the FIFs in this section is a relatively complicated process, since it requires the knowledge of the scene depth as well as the average view direction for each landmark to determine accurately whether a landmark can be matched from a given pose. While it certainly constitutes a barrier for building the FIFs incrementally online, this also justifies our proposal of having a dedicated map for localization/perception quality: since quantifying the localization quality from the point cloud is an expensive process, a dedicated map that can be built offline, where the efficiency is less important, and used for efficient planning online would be useful. Besides, in many practical applications, the robot operates in a known environment, and building a map offline is thus a reasonable choice.

D.7.3 Incremental Update

As mentioned in Section D.6.1, due to the additive natural of FIM, the information and trace kernels can be potentially updated as new landmarks are added/deleted from the

environment. It is, however, limited by the fact that it is difficult to accurately determine whether the correspondence with respect to a landmark can be established from a certain pose in an online fashion. Nevertheless, for relatively simple environments, incrementally building the FIF can still give reasonable result. We refer the reader to the accompanying video for such examples.

D.8 Conclusion and Future Work

In this work, we proposed the first dedicated map representation, the Fisher information field, for considering localization accuracy in perception-aware motion planning. For a known environment, the proposed map representation pre-computes the rotation-invariant component of the Fisher information and stores it in a voxel grid. At planning time, the Fisher information matrix (and related metrics) can be computed in constant time, regardless of the number landmarks in the environment. We validated the effectiveness and advantages of the FIF by applying it to different motion planning algorithms, namely RRT* and trajectory optimization. Integrating the proposed map in motion planning algorithms was shown to increase the localization success rate and accuracy. All the variants of the proposed map showed $1 \sim 2$ order-of-magnitude shorter planning time than the point cloud. In trajectory optimization, the proposed map representation, in addition of being far more efficient, achieved better localization accuracy than the point cloud, thanks to the fact that our map is differentiable.

The pre-computation, which is the key for the efficiency of the proposed map, is possible due to the special form of the visibility model (D.18) we enforced. Following this form, polynomial and GP approximations were explored in the paper. In particular, the quadratic polynomial model (D.22) and GP model (D.26) with different number of samples were implemented and tested. Different information metrics from the FIM were also tested. While using the combination of GP ($50 \sim 70$ samples) with the FIM determinant showed overall the best performance among all the variants, the efficiency (in both memory and query time) of using the FIM trace and/or the quadratic model is still appealing. For example, in the trajectory optimization experiment, the combination of the quadratic model and the FIM trace was only slightly worse than GP with the FIM determinant but ~ 10 times more efficient (Table D.5).

The assumption of building the proposed map from the point cloud is that whether a landmark can be matched from a certain viewpoint can be determined accurately. This is, however, not an easy task in any non-trivial environment, which requires much information in addition to the positions of the landmarks. To overcome this limitation, there are several interesting research directions. Increasing the accuracy of the process would help improve the consistency between the prediction of the proposed map and the actual localization result. Moreover, being able to determine the matchability of landmarks without knowing the full information of the environment would greatly

extend the application scenario of the proposed method. For example, it would allow to incrementally build an accurate FIF during the exploration of an unknown environment using the output of visual-inertial odometry as an input, as shown in Section D.7.3 for a simple scene layout.

E Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry

Reprinted, with permission, from:

Z. Zhang and D. Scaramuzza. "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018. DOI: 10.1109/IROS.2018.8593941 [320]

A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry

Zichao Zhang and Davide Scaramuzza

Abstract — In this tutorial, we provide principled methods to quantitatively evaluate the quality of an estimated trajectory from visual(-inertial) odometry (VO/VIO), which is the foundation of benchmarking the accuracy of different algorithms. First, we show how to determine the transformation type to use in trajectory alignment based on the specific sensing modality (*i.e.* monocular, stereo and visual-inertial). Second, we describe commonly used error metrics (*i.e.* the absolute trajectory error and the relative error) and their strengths and weaknesses. To make the methodology presented for VO/VIO applicable to other setups, we also generalize our formulation to any given sensing modality. To facilitate the reproducibility of related research, we publicly release our implementation of the methods described in this tutorial.

Open Source Code

A trajectory evaluation toolbox that implements the methods in this tutorial is available at https://github.com/uzh-rpg/rpg_trajectory_evaluation.

E.1 Introduction

Visual(-inertial) odometry (VO/VIO) uses cameras and inertial measurement units (IMUs), which are complementary sensors, to estimate the state (position, orientation and velocity) of the robot. VO/VIO is able to provide robust state estimate for other

tasks, such as control and planning, and therefore is widely used in robotic applications. The accuracy of a VO/VIO algorithm is quantified by evaluating the estimated trajectory (*i.e.* the time history of the state) with respect to the groundtruth, which is necessary to understanding and benchmarking different algorithms.

Quantitatively comparing the estimated trajectory with the groundtruth, however, is not an easy task. There are two major difficulties. First, the estimated trajectory and the groundtruth are usually expressed in different reference frames, and, therefore, cannot be compared directly. Second, a trajectory consists of the states at many different times and, therefore, is high-dimensional data. Thus, how to summarize the information of the whole trajectory into concise accuracy metrics is not trivial. To address the first problem, the estimated trajectory requires to be properly transformed into the same reference frame as the groundtruth, which is often called *trajectory alignment*. To address the second problem, meaningful error metrics need to be used and their properties well understood.

To tackle the above difficulties, this tutorial provides principled methods for trajectory alignment with the focus on VO/VIO and discusses different error metrics, as illustrated in Fig. E.1. We first detail the trajectory alignment methods for different visual-inertial systems (monocular, stereo and visual-inertial) and discuss the strengths and weaknesses of commonly used error metrics. We then further formulate the trajectory estimation and quantitative evaluation problem in a sensor-agnostic manner, from which we can generalize the methods presented in this tutorial to trajectory evaluation for other sensing modalities. Note that in this tutorial, we assume that the temporal correspondence of the estimate and the groundtruth has already been established.

E.1.1 Related Work

Most existing quantitative trajectory evaluation approaches were introduced together with a specific algorithm or a dataset. Sturm *et al.* [271] provided a benchmark for RBG-D simultaneous localization and mapping (SLAM) systems, and proposed to use both the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE). ATE is also widely used to evaluate visual odometry/SLAM algorithms, for example, in [82, 186, 92]. Compared with ATE, relative error, as analyzed in Burgard *et al.* [32] and Kümmerle *et al.* [142], is less sensitive to the specific time the estimation error occurs. Geiger *et al.* [98] further extended the relative error as a function of sub-trajectory length and velocity to provide more informative results.

Despite the rich literature in this field, there is very little work dedicated to the exact problem of quantitative trajectory evaluation for VO/VIO, which leaves many open issues. It is not clear, for example, to what extent the current approaches are applicable: is the method for one sensing modality also suitable for another (*e.g.* can the same

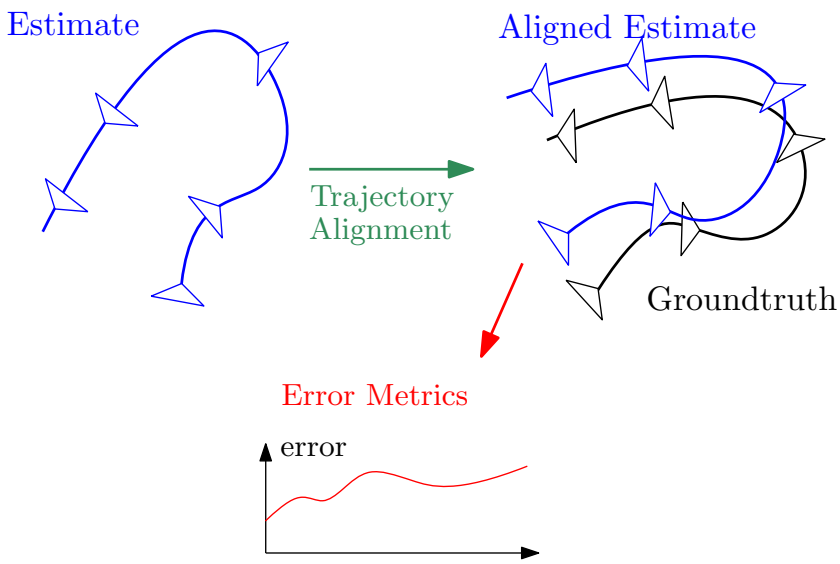


Figure E.1 – The process of quantitative trajectory evaluation. First, the estimated trajectory (blue) needs to be aligned with the groundtruth (black), Then, the trajectory estimation error can be calculated from the aligned estimate and the groundtruth using certain error metrics.

evaluation method be used for both VO and VIO)? More importantly, quantitatively evaluating an estimated trajectory involves many details, which are often described vaguely in the literature but have a big impact on the final result. This severely hinders the reproducibility of related research.

E.1.2 Contributions and Outline

The contributions of this tutorial are:

- We derive and describe in details the methods to evaluate an estimated trajectory from VO/VIO, including trajectory alignment (based on the specific sensing modality) and commonly used error metrics.
- We provide a general formulation for quantitative trajectory evaluation, which can be used to generalize the presented methods to other setups.
- We release our implementation of the evaluation methods to the public.

The rest of the tutorial is structured as follows. The formulation of visual(-inertial) odometry as a least squares problem is introduced in Section E.2. The ambiguity of visual-inertial systems and the trajectory alignment method, which is tightly related to the ambiguity, are detailed in Section E.3. Commonly used error metrics (absolute and relative errors) are then described in Section E.4. In Section E.5, the presented trajectory

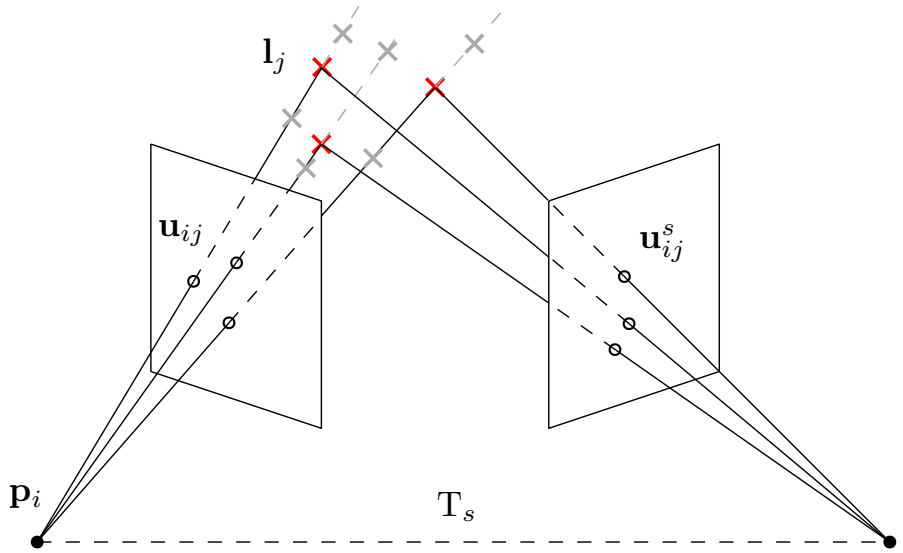


Figure E.2 – Camera measurement model and scale ambiguity for a single camera. The camera projects 3D points (red crosses) to 2D points (black circles) on the image plane. For a single camera, 3D points that are in the same direction but at different distances (gray crosses) are projected to the same 2D point, which leads to the scale ambiguity in (E.9). When a second camera with a constant transformation T_s relative to the first one is added, the scale ambiguity is eliminated.

evaluation methods are generalized to other setups than VO/VIO. Finally, example VIO evaluation on real data is demonstrated in Section E.6.

E.2 Visual(-inertial) Odometry Formulation

In this section, we first define the states and the noise-free measurement model for a visual-inertial system and then formulate VO/VIO as a least squares problem.

E.2.1 States and Measurement Models

States: For a general visual-inertial system, the variables of interest (called *state*) at t_i is

$$\mathbf{x}_i = \{\mathbf{p}_i,\ \mathtt{R}_i,\ \mathbf{v}_i,\ \mathbf{b}^a_i,\ \mathbf{b}^g_i\}, \tag{E.1}$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the position of the system, $\mathtt{R}_i \in \mathrm{SO}(3)$ the rotation matrix, $\mathbf{v}_i \in \mathbb{R}^3$ the velocity, and $\mathbf{b}^g_i, \mathbf{b}^a_i \in \mathbb{R}^3$ the gyroscope and accelerometer biases. $\mathbf{x}_i$ is expressed in the world frame, except that the biases in the body frame (the IMU frame is assumed to be the same as the body frame for simplicity). It is also common to maintain a map of 3D points (landmarks) as auxiliary states $L = \{\mathbf{l}_j\}_{j=0}^{J}$.

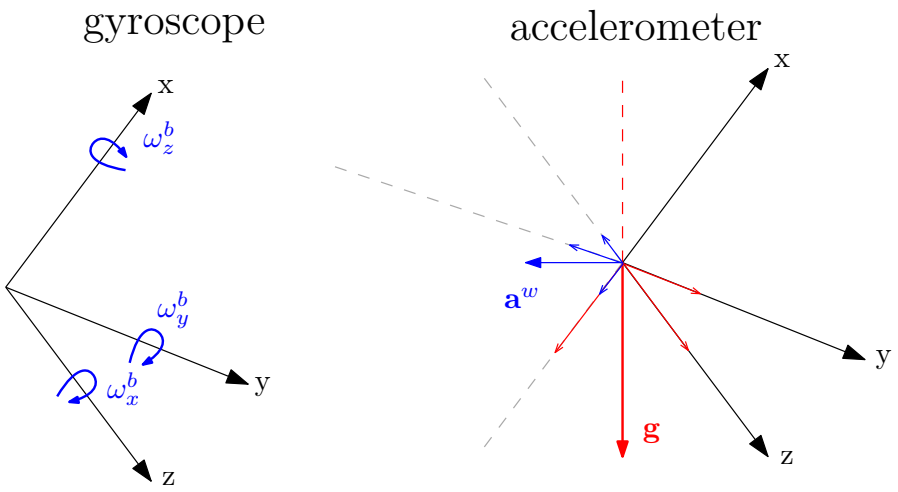


Figure E.3 – IMU measurement model (E.4). The biases are not visualized. In the illustration of the accelerometer, if the body frame (black) is rotated around the gravity direction (red), the gravity components on the axes of the body frame remain unchanged (invariant). The invariance does not hold for rotations around any other axis.

A trajectory can be parameterized either discretely or using continuous-time representations (*e.g.* [18]), and the former is dominant in VO/VIO. When a discrete parameterization is used, a trajectory can be represented using the states at a set of discrete times $t_s = \{t_i\}_{i=0}^{N-1}$, namely $\mathbf{X} = \{\mathbf{x}_i\}_{i=0}^{N-1}$.

Measurement Models: The measurements of a visual-inertial system come from the cameras and the IMUs. The camera project 3D points to 2D points on the image plane. The pixel coordinates of the tracked features $\tilde{\mathbf{u}}_{ij}$ are usually used as the measurements, and the noise-free measurement model is

$$\mathbf{u}_{ij} = \text{proj}(\mathtt{R}_i^\top \mathbf{l}_j - \mathtt{R}_i^\top \mathbf{p}_i), \tag{E.2}$$

where $\text{proj}(\cdot)$ projects a 3D point in the camera frame to the pixel coordinates. In a stereo configuration, for the same 3D landmark, we also have another measurement $\tilde{\mathbf{u}}_{ij}^s$ with the noise-free measurement model

$$\mathbf{u}_{ij}^s = \text{proj}(\mathtt{R}_i^\top \mathbf{l}_j - \mathtt{R}_i^\top \mathbf{p}_i - \mathbf{t}_{bs}), \tag{E.3}$$

where $\mathbf{t}_{bs}$ is the baseline between the stereo pair. Note that we made a few simplifications in the above formulations: the camera frame in (E.2) is assumed to be the same as the body frame, and the stereo cameras in (E.3) is assumed to be only different by a translation. For a more general setup, it can be shown that the conclusions in this section still hold. The camera measurement model is illustrated in Fig. E.2.

The IMU outputs the angular velocity $\tilde{\boldsymbol{\omega}}_i$ and the specific force (acceleration together

Algorithm 1: Closed-form solution to (E.13)

Data: estimation $\{\hat{\mathbf{p}}_i\}_{i=0}^{N-1}$, groundtruth $\{\mathbf{p}_i\}_{i=0}^{N-1}$
Result: $s, \mathtt{R}, \mathbf{t}$ that minimize $\sum_{i=0}^{N-1}\|\mathbf{p}_i - s\mathtt{R}\hat{\mathbf{p}}_i - \mathbf{t}\|^2$

Calculate: $\boldsymbol{\mu}_{\mathbf{p}} = \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{p}_i \quad \boldsymbol{\mu}_{\hat{\mathbf{p}}} = \frac{1}{N}\sum_{i=0}^{N-1}\hat{\mathbf{p}}_i \quad \boldsymbol{\sigma}^2_{\mathbf{p}} = \frac{1}{N}\sum_{i=0}^{N-1}\|\mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}}\|^2$
$\boldsymbol{\sigma}^2_{\hat{\mathbf{p}}} = \frac{1}{N}\sum_{i=0}^{N-1}\|\hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}}\|^2 \; \Sigma = \frac{1}{N}\sum_{i=0}^{N-1}(\mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}})(\hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}})^\top$
Singular value decomposition: $\Sigma = UDV^\top$
if $det(U)det(V) < 0$ **then**
  $W = \mathrm{diag}(1, 1, -1)$
else
  $W = \mathrm{I}_{3\times 3}$
end
$\mathtt{R} = UWV^\top$
$s = \frac{1}{\boldsymbol{\sigma}^2_{\hat{\mathbf{p}}}}\mathrm{trace}(DW)$ or $s = 1$ if the scale is known
$\mathbf{t} = \boldsymbol{\mu}_{\mathbf{p}} - s\mathtt{R}\boldsymbol{\mu}_{\hat{\mathbf{p}}}$

with gravity) $\tilde{\mathbf{a}}_i$ in the body frame. The measurement model is

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_i^{\mathrm{b}} + \mathbf{b}_i^g, \quad \mathbf{a}_i = \mathtt{R}_i^\top(\mathbf{a}_i^{\mathrm{w}} - \mathbf{g}) + \mathbf{b}_i^a, \tag{E.4}$$

where $\boldsymbol{\omega}_i^{\mathrm{b}}$ is the angular velocity in the body frame, $\mathbf{a}_i^{\mathrm{w}}$ the acceleration in the world frame, $\mathbf{g}$ the gravity vector in the world frame. The IMU measurement model (E.4) is illustrated in Fig. E.3. The outputs of the gyroscope and the accelerometer (E.4) are usually at a high frequency and do not directly relate to our states (E.1). Therefore, a common practice in (keyframe-based) VIO algorithms is to use the integration of (E.4). In this paper, we use the preintegrated IMU measurements proposed in [161, 88]. Roughly speaking, we integrate the raw IMU measurements to get the relative rotation $\Delta\tilde{\mathtt{R}}_{ik}$, velocity $\Delta\tilde{\mathbf{v}}_{ik}$ and position $\Delta\tilde{\mathbf{p}}_{ik}$ between two states $\mathbf{x}_i$ and $\mathbf{x}_k$, and the integration is formulated to be independent of the states (except for the biases) so that re-integration is not needed when the states change (*e.g.* during optimization iterations). The corresponding measurement model is

$$\begin{aligned}
\Delta\mathtt{R}_{ik} &= \mathtt{R}_i^\top\mathtt{R}_k, \\
\Delta\mathbf{v}_{ik} &= \mathtt{R}_i^\top(\mathbf{v}_k - \mathbf{v}_i - \mathbf{g}\Delta t_{ik}), \\
\Delta\mathbf{p}_{ik} &= \mathtt{R}_i^\top(\mathbf{p}_k - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ik} - \frac{1}{2}\mathbf{g}\Delta t_{ik}^2),
\end{aligned} \tag{E.5}$$

where $\Delta t_{ik} = t_k - t_i$.

Table E.1 – Transformations in trajectory alignment for different visual and inertial configurations.

Configuration	Monocular	Stereo	Inertial(+visual)
Type	Similarity	Rigid body	Yaw-only rigid body
Align-Multi	Alg. 1	Alg. 1	Alg. 1 with rotation (E.18)
Align-Single	×*	(E.19)	(E.19) with rotation (E.20)

* Scale cannot be estimated from a single state.

E.2.2 VO/VIO as a Least Squares Problem

By collecting the visual measurements $\tilde{\mathbf{z}}_{\mathrm{V}}$ (pixel coordinates of the observed landmarks) and inertial measurements $\tilde{\mathbf{z}}_{\mathrm{I}}$ (preintegrated IMU measurements *e.g.* [88]), VO/VIO can be formulated as a nonlinear least squares (NLLS) problem

$$\hat{\mathbf{X}}^* = \arg\min_{\mathbf{X}} J(\mathbf{X}), \tag{E.6}$$

where

$$J(\mathbf{X}) = \arg\min_{\mathbf{X}} \ \|\mathbf{f}_{\mathrm{V}}(\mathbf{X}) \boxminus \tilde{\mathbf{z}}_{\mathrm{V}}\|^2_{\Sigma_{\mathrm{V}}} + \|\mathbf{f}_{\mathrm{I}}(\mathbf{X}) \boxminus \tilde{\mathbf{z}}_{\mathrm{I}}\|^2_{\Sigma_{\mathrm{I}}} \tag{E.7}$$

where $\mathbf{f}_{\mathrm{V}}(\cdot)$ and $\mathbf{f}_{\mathrm{I}}(\cdot)$ denote the aforementioned noise-free visual and inertial measurement models respectively, Σ is the measurement covariance and $\|\mathbf{r}\|^2_{\Sigma} \triangleq \mathbf{r}^\top \Sigma^{-1} \mathbf{r}$ is the squared Mahalanobis distance [1]. In words, (E.6) aims to find the $\mathbf{X}$ that minimizes the sum of covariance weighted visual and inertial residuals. Note that $\boxminus$ is used because the inertial residual involves rotation. For the complete formulation of the residuals, we refer the reader to [88].

Next, we will show the inherent ambiguity of the NLLS problem (E.6) and how the trajectory alignment should be performed accordingly.

E.3 Visual(-inertial) Ambiguity and Trajectory Alignment

In this section, we first discuss the ambiguities in different visual(-inertial) setups and the complication of quantitative trajectory evaluation due to the ambiguities. We then show how to perform trajectory alignment for specific visual(-inertial) setups.

[1] Strictly speaking, directly solving (E.6) and (E.7) results in a batch optimization approach. Other methods such as filters and sliding window estimators aim to solve the same problem but in a recursive manner.

E.3.1 Ambiguities and Equivalent Parameters

(E.6) has infinite solutions that have the same minimum cost. The reason is that the predicted measurements $\mathbf{f}(\mathbf{X})$ are invariant to certain transformations $g(\cdot)$ of the parameter, namely $\mathbf{f}(\mathbf{X}) = \mathbf{f}(\mathbf{X}')$ with $\mathbf{X}' = g(\mathbf{X})$. Since the measurements $\tilde{\mathbf{z}}$ are constant, the cost function (E.7) is also invariant to such transformations. Therefore, the NLLS problem (E.6) has certain ambiguities related to $g(\cdot)$, and parameters that are different by such transformations are equivalent. Note that in practice, a unique solution can be obtained by enforcing additional constraints [317].

Obviously, the transformations $g(\cdot)$ depend on the specific sensors used. To see this, we now derive the transformations that will not change the predicted measurements (E.2), (E.3) and (E.5). Consider a similarity transformation parameterized by $\mathtt{S} = \{s, \mathtt{R}, \mathbf{t}\}$ as a starting point, where s is a scalar, $\mathtt{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$. $\mathtt{S}$ transforms the state $\mathbf{x}_i$ and $\mathbf{l}_j$ as

$$\mathbf{p}'_i = s\mathtt{R}\mathbf{p}_i + \mathbf{t},\ \mathtt{R}'_i = \mathtt{R}\mathtt{R}_i,\ \mathbf{v}'_i = s\mathtt{R}\mathbf{v}_i,\ \mathbf{l}'_j = s\mathtt{R}\mathbf{l}_j + \mathbf{t}, \tag{E.8}$$

and the biases are expressed in the body frame and, thus are not changed by $\mathtt{S}$.

Substituting (E.8) into the monocular measurement model (E.2), and it is obvious that

$$\mathbf{u}'_{ij} = \mathrm{proj}(s\mathtt{R}_i^\top \mathbf{l}_j - s\mathtt{R}_i^\top \mathbf{p}_i) = \mathbf{u}_{ij} \tag{E.9}$$

for any $\mathtt{S}$. For a stereo setup (E.3), the predicted measurement using the transformed states is

$$\mathbf{u}_{ij}^{s'} = \mathrm{proj}(s\mathtt{R}_i^\top \mathbf{l}_j - s\mathtt{R}_i^\top \mathbf{p}_i - \mathbf{t}_{bs}), \tag{E.10}$$

and $\mathbf{u}_{ij}^{s'} = \mathbf{u}_{ij}^{s}$ holds only when $s = 1$, and $\mathtt{S}$ becomes a rigid body transformation. The difference of a monocular and a stereo setup is illustrated in Fig. E.2.

From the inertial measurement model (E.5), we have

$$\begin{aligned} \Delta\mathtt{R}'_{ik} &= \mathtt{R}_i^\top \mathtt{R}_k, \\ \Delta\mathbf{v}'_{ik} &= \mathtt{R}_i^\top (s\mathbf{v}_k - s\mathbf{v}_i - \mathtt{R}^\top \mathbf{g}\Delta t_{ik}), \\ \Delta\mathbf{p}'_{ik} &= \mathtt{R}_i^\top (s\mathbf{p}_k - s\mathbf{p}_i - s\mathbf{v}_i\Delta t_{ik} - s\mathtt{R}^\top \frac{1}{2}\mathbf{g}\Delta t_{ik}^2). \end{aligned} \tag{E.11}$$

Comparing (E.11) with (E.5), we can see that the predicted measurements remain unchanged only when $s = 1$ and $\mathtt{R}^\top \mathbf{g} = \mathbf{g}$, which means $\mathtt{R}$ can only be a rotation around

z-axis and is parameterized by only one parameter θ:

$$\mathtt{R}_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{E.12}$$

This yaw-only rigid body transformation (one DoF rotation plus a translation) corresponds to the four unobservable DoFs for visual-inertial systems [129]. Note that although the above derivation is based on the preintegration measurement model (E.5), the conclusion is generally applicable for inertial sensors. Intuitively, as illustrated in Fig. E.3, the gyroscope and the accelerometer measure the angular velocity and acceleration in the body frame, which are not affected by rigid body transformations. However, the accelerometer additionally measures the gravity, whose projections on the axes of the body frame only remain unchanged when the rotation is around the gravity (*i.e.* in the form of (E.12)).

To summarize, for a monocular setup, parameters that are different by a similarity transformation are equivalent. Such transformations for a stereo setup and inertial sensors are rigid body transformations and 4 DoF yaw-only rigid body transformations (*i.e.* a rotation around the gravity plus a translation) respectively.

E.3.2 Trajectory Evaluation with Ambiguities

The aforementioned ambiguities complicate the trajectory evaluation process: we cannot directly take the difference (*e.g.* Euclidean distance of the positions) between an estimate $\hat{\mathbf{X}}$ and the groundtruth $\mathbf{X}_{\text{gt}}$ as the estimation error. To see this, consider the subspaces (in the parameter space) of the equivalent parameters of $\hat{\mathbf{X}}$ and $\mathbf{X}_{\text{gt}}$, denoted as E_{est} and E_{gt} respectively, each of which contains an infinite number of equivalent parameters. For arbitrary $\hat{\mathbf{X}}_a, \hat{\mathbf{X}}_b \in E_{\text{est}}$, the estimation error computed from $\hat{\mathbf{X}}_a$ and $\mathbf{X}_{\text{gt}}$ (or any element in E_{gt}) should be exactly the same as the error with $\hat{\mathbf{X}}_b$ due to the equivalence. This is obviously not the case if we use the difference as an error metric directly.

Therefore, instead of the difference between the estimate and the groundtruth, it is the "distance" between the two corresponding equivalent parameter subspaces that should be used to quantify the estimation error. A common practice is to first find an *equivalent* estimate $\hat{\mathbf{X}}' \in E_{\text{est}}$ that is, by some metric, closest to the groundtruth $\mathbf{X}_{\text{gt}}$ and then calculate the difference from $\hat{\mathbf{X}}'$ and $\mathbf{X}_{\text{gt}}$ (see Section E.4). The process of finding $\hat{\mathbf{X}}'$ is referred to as *trajectory alignment*, which we will see next for different sensor combinations.

E.3.3 Trajectory Alignment in Visual(-inertial) Systems

To find the equivalent estimate $\hat{\mathbf{X}}'$, we essentially need to find a transformation $g'(\cdot)$, which can be of different types as described in Section E.3.1, and then calculate $\hat{\mathbf{X}}' = g'(\mathbf{X})$. For both similarity and rigid body transformations, the method proposed in Umeyama *et al.* [292] has become the de-facto standard. In this section, we first present Umeyama's method and then show how it can be adapted to calculate the 4 DoF transformation for visual-inertial systems.

One remaining open choice is which states should be used to calculate the transformation. While there is no "gold standard", two common ways are usually used in practice: 1) using all the estimated states; 2) using only the first one or several initial states. The former tends to give a lower error if later an error metric for the whole trajectory (*e.g.* ATE) is used, and the latter gives an intuitive error distribution that the estimation error increases over time. We will see the examples about this point on real data in Section E.6.2. In terms of computing the alignment transformation, Umeyama's method is only suitable for calculating the transformation using multiple estimated states, and, therefore, we will in addition show how to calculate rigid body and 4 DoF transformations from the first state, which will also be used for calculating the relative error metric in Section E.4.

Alignment Using Multiple States

As discussed in Salas *et al.* [231], it is usually sufficient to calculate the trajectory alignment transformation using only the translational components of the estimation and the groundtruth. To put it formally, given the estimated positions $\{\hat{\mathbf{p}}_i\}_{i=0}^{N-1}$ and the groundtruth positions $\{\mathbf{p}_i\}_{i=0}^{N-1}$, we want to find a similarity transformation $\mathtt{S}' = \{s', \mathtt{R}', \mathbf{t}'\}$ that satisfies:

$$\mathtt{S}' = \underset{\mathtt{S}=\{s,\mathtt{R},\mathbf{t}\}}{\arg\min} \sum_{i=0}^{N-1} \|\mathbf{p}_i - s\mathtt{R}\hat{\mathbf{p}}_i - \mathbf{t}\|^2 \tag{E.13}$$

To solve the least squares problem (E.13), the method in Umeyama *et al.* [292] is often used, as summarized in Alg. 1. Note that if the scale is known (stereo and inertial setup in Section E.3.1), we directly set $s = 1$ in line 9 of Alg. 1 After calculating the transformation $\mathtt{S}'$, the aligned estimation is then:

$$\hat{\mathbf{p}}_i' = s'\mathtt{R}'\hat{\mathbf{p}}_i + \mathbf{t}', \ \hat{\mathtt{R}}_i' = \mathtt{R}'\hat{\mathtt{R}}_i, \ \hat{\mathbf{v}}_i' = s'\mathtt{R}'\hat{\mathbf{v}}_i \tag{E.14}$$

If a yaw-only rigid body transformation is desired, we need to adapt the rotation calculation in Umeyama's method. As proved in [292], the rotation calculated in line 8

of Alg. 1 is the closed-form solution of

$$\mathtt{R}' = \underset{\mathtt{R}\in\mathrm{SO}(3)}{\arg\min} \|\mathrm{P} - \mathtt{R}\hat{\mathrm{P}}\|_F^2, \tag{E.15}$$

where $\mathrm{P} = [\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{N-1}], \hat{\mathrm{P}} = [\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{N-1}], \mathbf{r}_i = \mathbf{p}_i - \boldsymbol{\mu}_\mathbf{p}, \hat{\mathbf{r}}_i = \hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}}$, and $\|\cdot\|_F$ is the Frobenius norm. The cost in (E.15) can be further written as

$$\|\mathrm{P} - \mathtt{R}\hat{\mathrm{P}}\|_F^2 \quad = \quad \mathrm{trace}(\mathrm{P}\mathrm{P}^\top + \hat{\mathrm{P}}\hat{\mathrm{P}}^\top - 2\mathtt{R}\hat{\mathrm{P}}\mathrm{P}^\top), \tag{E.16}$$

and therefore (E.15) is equivalent to

$$\mathtt{R}' = \underset{\mathtt{R}\in\mathrm{SO}(3)}{\arg\max}\ \mathrm{trace}(\mathtt{R}\hat{\mathrm{P}}\mathrm{P}^\top). \tag{E.17}$$

If the rotation is of the form (E.12), we only need to find the following maximum with respect to θ:

$$\theta' = \underset{\theta}{\arg\max}\ (p_{12} - p_{21})\sin\theta + (p_{11} + p_{22})\cos\theta \tag{E.18}$$

where p_{ij} is the coefficient of $\hat{\mathrm{P}}\mathrm{P}^\top$. With the solution θ' to (E.18), we can calculate the desired rotation $\mathtt{R}'_z$ using (E.12) and the translation with line 10 in Alg. 1 (with $s = 1$). The aligned estimation is calculated the same as (E.14).

It is worth noting that, in this section, the alignment is based on a least squares solution, which is valid only when all the states are of the same uncertainty. If we have the knowledge about the quality of the state estimate, for example, covariance from VO/VIO, more sophisticated methods can be used to account for this (*e.g.* optimization as in [231]).

Alignment Using A Single State

It is possible to calculate a rigid body transformation or a yaw-only transformation with only the first state. Calculating a rigid body transformation is trivially

$$\mathtt{R}' = \mathtt{R}_0\hat{\mathtt{R}}_0^\top, \quad \mathbf{t}' = \mathbf{p}_0 - \mathtt{R}'\hat{\mathbf{p}}_0. \tag{E.19}$$

Similar to the previous case, computing a yaw-only transformation needs a different treatment. Specifically, the rotation $\hat{\mathtt{R}}'_0 = \mathtt{R}'_z\hat{\mathtt{R}}_0$ should be as close to $\mathtt{R}_0$ as possible:

$$\mathtt{R}'_z = \underset{\mathtt{R}_z}{\arg\min} \|\mathtt{R}_0 - \mathtt{R}_z\hat{\mathtt{R}}_0\|_F^2 \quad \Rightarrow \quad \mathtt{R}'_z = \underset{\mathtt{R}_z}{\arg\max}\ \mathrm{trace}(\mathtt{R}_z\hat{\mathtt{R}}_0\mathtt{R}_0^\top), \tag{E.20}$$

which is of the same form as (E.17) and can be solved similarly. Once we have $\mathtt{R}'_z$, the translational component $\mathbf{t}'$ is calculated the same as (E.19).

E.3.4 Summary

To summarize, different combinations of visual and inertial sensors result in different ambiguities in VO/VIO. Due to the ambiguities, certain types of transformations should be used to align the estimation with the groundtruth before calculating the estimation error. For various combinations of visual and inertial sensors, we summarize the types of trajectory alignment transformations and the methods to calculate them in Table E.1. Using the aligned trajectory estimate, we can now calculate different error metrics to quantify the accuracy of VO/VIO.

E.4 Trajectory Error Metrics

To calculate the estimation error from the groundtruth $\mathbf{X}_{\text{gt}}$ and the aligned estimation $\hat{\mathbf{X}}'$, two commonly used error metrics are the absolute trajectory error (ATE) and the relative error (RE). In this section, we will describe them in details and discuss their advantages and disadvantages.

E.4.1 Absolute Trajectory Error

For a single state, the error between $\hat{\mathbf{x}}'_i$ and the groundtruth $\mathbf{x}_i$ can be parameterized as

$$\varDelta\mathbf{x}_i = \{\varDelta\mathtt{R}_i, \varDelta\mathbf{p}_i, \varDelta\mathbf{v}_i\} \tag{E.21}$$

and satisfies

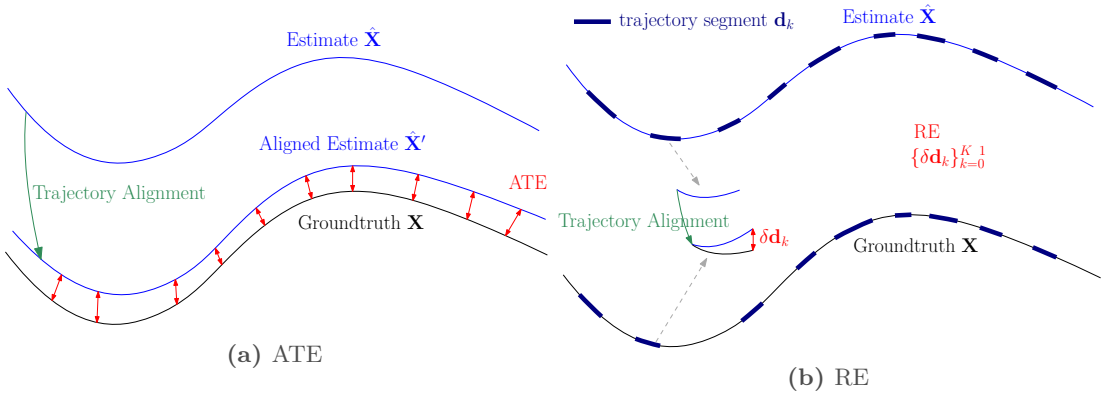
$$\mathtt{R}_i = \varDelta\mathtt{R}_i\hat{\mathtt{R}}'_i,\ \ \mathbf{p}_i = \varDelta\mathtt{R}_i\hat{\mathbf{p}}'_i + \varDelta\mathbf{p}_i,\ \ \mathbf{v}_i = \varDelta\mathtt{R}_i\hat{\mathbf{v}}'_i + \varDelta\mathbf{v}_i \tag{E.22}$$

Note that the parameterization of the error (E.21) and (E.22) is not unique. For example, $\varDelta\mathtt{R}_i$ can also appear on the right side of $\hat{\mathtt{R}}'_i$ in (E.22). While there is no standard for error parameterization, one must be consistent during the trajectory evaluation. In addition, since the biases are always expressed in the body frame, the biases error is trivially the Euclidean distance of the estimate and the groundtruth.

With (E.22), we can easily calculate the error $\varDelta\mathbf{x}_i$

$$\varDelta\mathtt{R}_i = \mathtt{R}_i(\hat{\mathtt{R}}'_i)^\top, \quad \varDelta\mathbf{p}_i = \mathbf{p}_i - \varDelta\mathtt{R}_i\hat{\mathbf{p}}'_i, \quad \varDelta\mathbf{v}_i = \mathbf{v}_i - \varDelta\mathtt{R}_i\hat{\mathbf{v}}'_i. \tag{E.23}$$

To quantify the quality of the whole trajectory, the root mean square error (RMSE) is



(a) ATE **(b)** RE

Figure E.4 – Illustrations of absolute trajectory error and relative error. The error after alignment is exaggerated for visualization. For relative error, the trajectory segments should be all possible pairs of states that satisfy certain criteria, and they are un-overlapped in (b) for the ease of visualization.

usually used

$$\begin{aligned} \text{ATE}_{\text{rot}} &= (\frac{1}{N}\sum_{i=0}^{N-1}\|\angle(\Delta\mathtt{R}_i)\|^2)^{\frac{1}{2}}, \\ \text{ATE}_{\text{pos}} &= (\frac{1}{N}\sum_{i=0}^{N-1}\|\Delta\mathbf{p}_i\|^2)^{\frac{1}{2}}, \end{aligned} \tag{E.24}$$

where $\angle(\cdot)$ means converting the rotation matrix to angle-axis representation and using the rotation angle as the error. Alternatively, one can also convert $\Delta\mathtt{R}_i$ to other representations (*e.g.* Euler angles) and get the corresponding rotation errors. The velocity error is defined similarly and omitted here. The calculation of ATE is illustrated in Fig. E.4a.

The advantage of ATE is that it gives a single number metric for the position/rotation/velocity estimation, which is easy to compare. However, as recognized by several researchers [32, 142, 98], ATE is sensitive to the time when the error occurs. For example, a rotation estimation error tends to give a larger ATE when it happens at the beginning of the trajectory than the situation when it occurs at the end. Therefore, in addition to ATE, the relative error is also widely used to provide more informative evaluation.

E.4.2 Relative Error

The basic idea of relative error is that, since VO/VIO systems do not have a global reference (global position and yaw), the estimation quality can be evaluated by measuring the relative relations between the states at different times.

To put it formally, first a set of K pairs of states is selected by some criteria (*e.g.* distance

Table E.2 – Comparison of absolute trajectory error and relative error.

	absolute trajectory error	relative error
Compute	**1**. Align the estimated trajectory. **2**. Calculate the RMSE using the aligned estimation and the groundtruth (E.24)	**1**. Select all sub-trajectory of length d. **2**. Align each sub-trajectory using the first state. **3**. Calculate the error of the end state of each sub-trajectory (E.26). **4**. Collect the errors for all the sub-trajectories (E.27). **5**. For different lengths d, repeat step 1-4.
Pros	• Single number metric, easy for comparison.	• Informative statistics can be computed from the errors of all sub-trajectories. • By changing the length d, the relative error can reflect both short and long term accuracy.
Cons	• Sensitive to the time when the estimation error occurs.	• Relatively complicated to compute. • Less straightforward for ranking the estimation accuracy.

along the trajectory) from $\hat{\mathbf{X}}$:

$$\mathfrak{F} = \{\mathbf{d}_k\}_{k=0}^{K-1}, \quad \mathbf{d}_k = \{\hat{\mathbf{x}}_s, \hat{\mathbf{x}}_e\}, \tag{E.25}$$

where $e > s$, and each pair defines a sub-trajectory. For each $\mathbf{d}_k$, a relative error $\delta\mathbf{d}_k$ is calculated in a similar way as the absolute error. Specifically, an alignment transformation, depending on the sensor configuration as in Table E.1, is computed from the first state $\hat{\mathbf{x}}_s$ and the corresponding groundtruth $\mathbf{x}_s$, and the aligned second state $\hat{\mathbf{x}}'_e$ computed using (E.14). Then the error $\delta\mathbf{d}_k$ for the state pair $\mathbf{d}_k$ is

$$\delta\boldsymbol{\phi}_k = \angle\, \delta\mathtt{R}_k = \angle\, \mathtt{R}_e(\hat{\mathtt{R}}'_e)^\top, \quad \delta\mathbf{p}_k = \|\mathbf{p}_e - \delta\mathtt{R}_k\hat{\mathbf{p}}'_e\|_2, \quad \delta\mathbf{v}_k = \|\mathbf{v}_e - \delta\mathtt{R}_k\hat{\mathbf{v}}'_e\|_2, \tag{E.26}$$

which are all scalars. Collecting the error (E.26) for all the pairs of states (sub-trajectories) in $\mathfrak{F}$ gives

$$\mathrm{RE}_{\mathrm{rot}} = \{\delta\boldsymbol{\phi}_k\}_{k=0}^{K-1}, \quad \mathrm{RE}_{\mathrm{pos}} = \{\delta\mathbf{p}_k\}_{k=0}^{K-1}, \quad \mathrm{RE}_{\mathrm{vel}} = \{\delta\mathbf{v}_k\}_{k=0}^{K-1}. \tag{E.27}$$

The calculation of RE is illustrated in Fig. E.4b.

Since the relative error (E.27) does not generate a single number but a collection of errors

for all the sub-trajectories that satisfy certain criteria, statistics such as the median, average and percentiles can be calculated, which gives more information than ATE. Another advantage is that by selecting the states according to different criteria, RE can have different meanings. For example, a common practice is to select pairs of states that are spaced by a certain distance along the trajectory. The RE from the states pairs that are spatially close reflects the local consistency, while the error for a larger distance reflects more the long-term accuracy. The disadvantage of RE is that it is relatively complicated to calculate, and it is less obvious to rank the estimation quality than using a single number metric as ATE.

E.4.3 Discussion and Summary

As discussed above, both ATE and the RE have their own advantages and disadvantages. It is probably not possible to say that a metric should be preferred in all situations over the other one. However, as pointed by [271], the two error metrics are actually highly correlated. In practice, providing both error metrics, if possible, will give a better understanding of the actual estimation quality from different aspects. We summarize the computation and properties of ATE and RE in Table E.2.

Together with the trajectory alignment described in Section E.3, we can quantify the accuracy of a trajectory estimate from VO/VIO. Before demonstrating the evaluation procedures on real data in Section E.6, we first show that the aforementioned methods for VO/VIO can be generalized to arbitrary sensing modalities.

E.5 General Trajectory Evaluation Problem

E.5.1 Trajectory Estimation Problem

Similar to VO/VIO in Section E.2, we define the estimation problem by specifying the parametrization of the trajectory, the measurements, and the cost function to minimize.

Parameterization: Using discrete parameterization, a trajectory can be represented using the states $\mathbf{X} = \{\mathbf{x}_i\}_{i=0}^{N-1}$ at a set of discrete times $t_s = \{t_i\}_{i=0}^{N-1}$.

Measurements: The measurements are collected at t_s, denoted as $\tilde{\mathbf{M}} = \{\tilde{\mathbf{z}}_i\}_{i=0}^{N-1}$, Note that $\tilde{\mathbf{z}}_i$ can be either the raw readings from the sensors or the output of processing the raw data (*e.g.* keypoint coordinates (E.2), preintegrated IMU measurements (E.5)). The corresponding noise-free measurement model is denoted as $\mathbf{f}(\mathbf{x})$.

Cost: For an estimate of the system parameters $\mathbf{X}$, a commonly used cost is the sum of

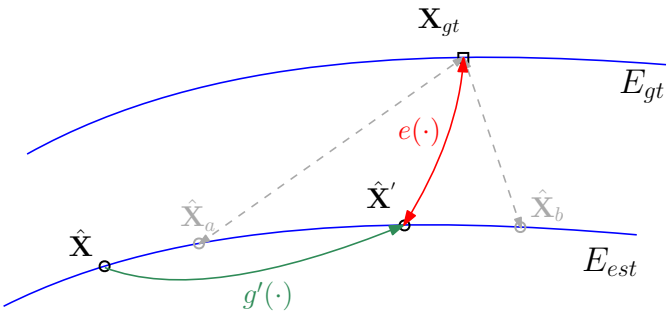


Figure E.5 – Illustration of the equivalent subspaces (blue) and the trajectory evaluation process in the parameter space. Directly using the difference (dashed gray line) between an estimation and the groundtruth does not give the same estimation error for equivalent parameters. Instead, the distance between the equivalent subspaces should be used. The first step (green) is to find a unique equivalent estimation $\hat{\mathbf{X}}'$ that is closest to $\mathbf{X}_{gt}$ by a distance metric $d_g(\cdot)$. The second step (red) is to calculate the distance between $\hat{\mathbf{X}}'$ and $\mathbf{X}_{gt}$ using an error metric $e(\cdot)$.

the squared Mahalanobis distance between the actual and the predicted measurements:

$$c(\mathbf{X}, \tilde{\mathbf{M}}) = \sum_{i=0}^{N-1} \| f(\mathbf{x}_i) - \tilde{\mathbf{z}}_i \|^2_{\Sigma_i}, \tag{E.28}$$

where Σ_i is the measurement covariance. The trajectory estimation is then the process of determining a set of trajectory parameters that minimize the cost (E.28):

$$\hat{\mathbf{X}}^* = \arg\min_{\mathbf{X}} c(\mathbf{X}, \tilde{\mathbf{M}}). \tag{E.29}$$

E.5.2 Ambiguities and Equivalent Parameters

With only the sensor measurements, the estimation problem (E.29) usually does not have a unique solution. For example, the absolute position cannot be determined for a visual(-inertial) odometry system. To put it formally, there exist a set of transformations $G = \{g(\cdot)\}$ that satisfy

$$c(g(\mathbf{X}), \tilde{\mathbf{M}}) = c(\mathbf{X}, \tilde{\mathbf{M}}) \quad \forall \mathbf{X},\ \forall g(\cdot) \in G, \tag{E.30}$$

where G is determined by the sensor combinations. In other words, for any $\mathbf{X}$, there is a subspace $E_{\mathbf{X}}$ (in the parameter space) where each element has the same cost (E.28) as $\mathbf{X}$.

Due to this ambiguity, we cannot directly take the difference (*e.g.* Euclidean distance if the states are vectors) between the estimation $\hat{\mathbf{X}}$ and the groundtruth $\mathbf{X}_{\text{gt}}$ as the estimation error, as illustrated in Fig. E.5.

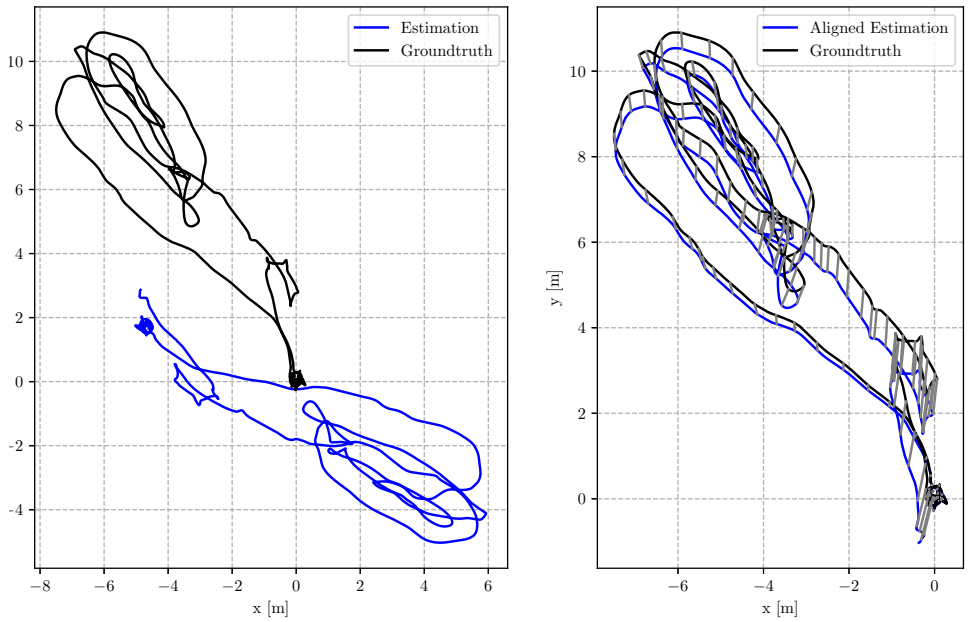


Figure E.6 – Trajectory alignment for the estimate from VINS-Mono on *Machine Hall 01*. The left is the top view of the unaligned estimation and the groundtruth, and the right is the aligned trajectory. The states correspondences are shown as gray lines (every 10th is drawn for clear visualization).

E.5.3 Quantitative Trajectory Evaluation

To uniquely define the estimation error of an estimate $\hat{\mathbf{X}}$, the first step is to find an equivalent estimation $\hat{\mathbf{X}}'$ that is closest to $\mathbf{X}_{gt}$ according to a certain distance metric $d_g(\cdot)$:

$$g'(\cdot) = \underset{g(\cdot)\in G}{\arg\min}\ d_g(g(\hat{\mathbf{X}}),\ \mathbf{X}_{gt}), \quad \hat{\mathbf{X}}' = g'(\hat{\mathbf{X}}), \tag{E.31}$$

which is the trajectory alignment process. Then we can quantify the difference between the estimation and the groundtruth by calculating the error between $\hat{\mathbf{X}}'$ and $\mathbf{X}_{gt}$ using a certain error metric $e(\cdot)$ as $e(\hat{\mathbf{X}}', \mathbf{X}_{gt})$. The above process in the parameter space is illustrated in Fig. E.5. We denote the distance metric $d_g(\cdot)$ and error metric $e(\cdot)$ only conceptually, because there is no standard way for defining them, as described in Section E.3 and Section E.4.

Therefore, for any sensor combination with ambiguities, to calculate the estimation error, we need to follow similar procedures as VO/VIO : 1) align the estimate with the groundtruth; 2) calculate the estimation error using certain metrics. Importantly, the transformation used for trajectory alignment needs to be computed by considering the properties of the sensors used, as we already see for visual(-inertial) systems.

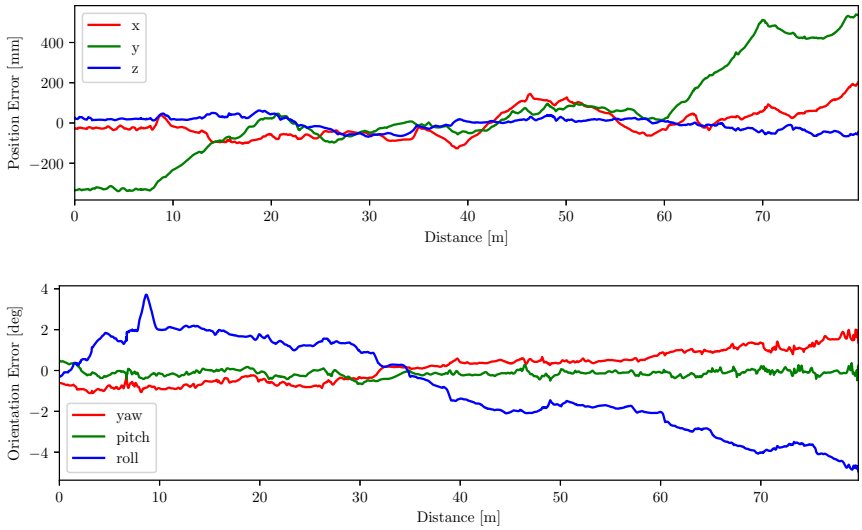


Figure E.7 – Absolute position and orientation error (E.23) with respect to the traveled distance, computed from the aligned trajectory and the groundtruth in Fig. E.6. The ATE (E.24) is 0.2795 m for translation and 2.4935 deg for rotation.

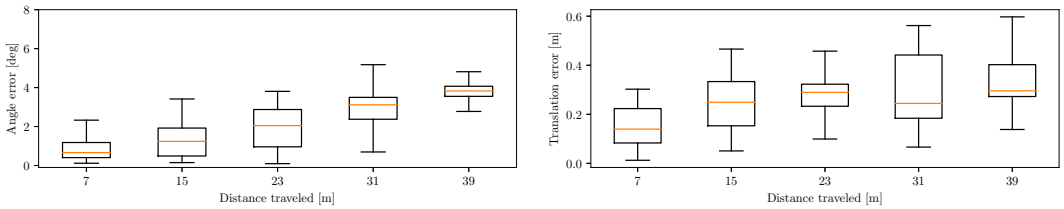


Figure E.8 – Relative translation and rotation errors (E.27) for different sub-trajectory lengths shown as a series of boxplots. The box in the middle indicates the two quartiles of all the estimation errors, the line through the box the median, and the whiskers the upper and lower quartiles.

E.6 Example Quantitative Evaluation

To illustrate the methods described in Section E.3 and Section E.4 with concrete examples, we first demonstrate the complete process of computing ATE and RE from an unaligned estimation and the groundtruth. Then we show the impact of the number of frames used for trajectory alignment, which seems to be a trivial detail but turns out rather crucial.

E.6.1 ATE and RE: a Complete Example

We ran VINS-Mono [209], which is a visual-inertial odometry algorithm, on the *Machine Hall 01* sequence from the EuRoC dataset [34] and evaluated the estimated trajectory.

Table E.3 – ATE using different states for trajectory alignment. When more states are used in the alignment, the translation ATE tends to be smaller.

States used	ATE_{pos} (m)	ATE_{rot} (deg)
1	0.4383	2.4919
1 - 452	0.4134	2.7427
1 - 904	0.3515	2.7902
1 - 1355	0.3180	2.8365
1 - 1807 (all)	0.2795	2.4935

As discussed above, the first step is to align the estimation with the groundtruth. We used all the states to calculate a yaw-only rigid-body transformation to align the trajectory as described in Section E.3. The process is illustrated in Fig. E.6. We can see the "raw" estimation from VINS-Mono is in a different reference frame as the groundtruth, and therefore cannot be directly compared. We then computed the estimation error using the aligned trajectory and the groundtruth. The absolute error for each state (E.23) is plotted in Fig. E.7, and the ATE (E.24) described in the caption.

The relative position and rotation errors (E.27) are plotted in Fig. E.8. We calculated the relative errors for sub-trajectories of different lengths. It is clear from Fig. E.8 that the estimation error (both translation and rotation) increases with the length of the sub-trajectories.

E.6.2 ATE: How Many Frames to Align?

As discussed in Section E.3, there is no standard for selecting the number of states to be used for trajectory alignment. However, it is of interest to understand how this choice affects the computed estimation error. To this end, we performed the same evaluation as the previous section, but used different states for trajectory alignment: the first Q states are used, where Q varies from 1 to the number of all the states in the trajectory.

We show the ATE of the whole trajectory for five different alignments in Table E.3. We can see that the position ATE decreases when more states are used in the alignment, while the rotation ATE does not show a obvious tendency. Intuitively, since the trajectory alignment aims to minimize the least squares position error (E.13), the more states that are used, the smaller the position ATE is likely to be. The rotation components are not used in computing the alignment transformation and thus are less correlated.

Note that in Table E.3, the difference of $\text{ATE}_{\text{trans}}$ between using the first state and all the states for alignment is quite large ($\sim 150\%$). Therefore, in practice, when comparing different algorithms, one needs to be consistent in which states are used for trajectory alignment across different algorithms for a fair comparison. Moreover, this information

is quite crucial to reproduce quantitative accuracy evaluations for VO/VIO and should always be presented together with the evaluation results.

E.7 Conclusion

In this tutorial, we presented principled approaches for quantitative trajectory evaluation for VO/VIO algorithms. We discussed the ambiguities in visual(-inertial) systems, which is the main source of the complication in trajectory evaluation. Then we detailed the quantitative evaluation methods for VO/VIO, including the trajectory alignment and error metrics. We further showed that similar approaches can be adopted for other sensing modalities that has ambiguities. To benefit the reproducibility of related research, we release our implementation of the methods in this tutorial to the public.

F Reference Pose Generation for Visual Localization

Reprinted, with permission, from:

Z. Zhang, T. Sattler, and D. Scaramuzza. "Reference Pose Generation for Visual Localization via Learned Features and View Synthesis". In: *Under review in Int. J. Comput. Vis.* (2020). URL: https://arxiv.org/abs/2005.05179 [319]

Reference Pose Generation for Visual Localization via Learned Features and View Synthesis

Zichao Zhang, Torsten Sattler and Davide Scaramuzza

Abstract — Visual Localization is one of the key enabling technologies for autonomous driving and augmented reality. High quality datasets with accurate 6 Degree-of-Freedom (DoF) reference poses are the foundation for benchmarking and improving existing methods. Traditionally, reference poses have been obtained via Structure-from-Motion (SfM). However, SfM itself relies on local features which are prone to fail when images were taken under different conditions, e.g., day/ night changes. At the same time, manually annotating feature correspondences is not scalable and potentially inaccurate. In this work, we propose a semi-automated approach to generate reference poses based on feature matching between renderings of a 3D model and real images via learned features. Given an initial pose estimate, our approach iteratively refines the pose based on feature matches against a rendering of the model from the current pose estimate. We significantly improve the nighttime reference poses of the popular Aachen Day-Night dataset, showing that state-of-the-art visual localization methods perform better (up to 47%) than predicted by the original reference poses. We extend the dataset with new nighttime test images, provide uncertainty estimates for our new reference poses, and introduce a new evaluation criterion. We will make our reference poses and our framework publicly available upon publication.

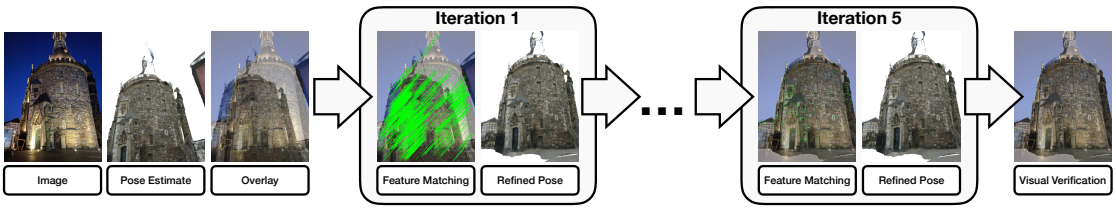


Figure F.1 – Overview of our approach: Given an image, we render a synthesized view of a 3D model from the given initial pose estimate of the image. Superimposing the rendered image over the original image provides a visual cue on the accuracy of the pose estimate. We match features extracted from the actual image and the rendering (shown as green lines connecting the corresponding positions in the overlay of the two images). This provides 2D-3D correspondences between the image and the underlying scene model. These 2D-3D matches are then used to obtain a refined estimate. Iterating this approach leads to subsequently more accurate poses (as evident from the smaller lines caused by a more accurate overlay). The final pose estimate can also be verified visually.

F.1 Introduction

Visual localization is the problem of estimating the camera pose, *i.e.* the position and orientation from which an image was taken, with respect to a known scene. Visual localization is a core component of many interesting applications such as self-driving cars [110] and other autonomous robots such as drones [155], as well as for augmented and virtual reality systems [45, 163].

Similar to other areas in computer vision, the availability of benchmark datasets such as [257, 295, 131, 239, 237, 11, 164] has served as a main driving force for research. Yet, there is a fundamental difference between visual localization and areas such as semantic segmentation and object detection in the way ground truth is obtained. For the latter, ground truth is provided by human annotations. However, humans are not able to directly predict highly accurate camera poses. Instead, ground truth is typically computed through a reference algorithm, *e.g.* Structure-from-Motion (SfM). Thus, localization benchmarks do not measure absolute pose accuracy. Rather, they measure to what degree methods are able to replicate the results of the reference algorithm. Given that the reference approach itself will produce inaccuracies and errors in the pose estimates, we use the term “reference poses" instead of “ground truth poses".

It is crucial that the reference algorithm generates poses with a higher accuracy than the actual localization methods evaluated on a benchmark. It is thus common to provide more data to the reference algorithm compared to what is made available to the localization approaches. For example, data from other sensors such as depth [257, 295], Lidar [164], an external motion capture system such as Vicon [250], or additional images not available to the localization methods [250] can be used if available. This paper considers the case where only images are available. In this case, SfM is typically used as the reference algorithm, *i.e.* the reference poses are obtained jointly from all test images whereas

localization approaches typically localize a single image at a time. This should lead to more accurate reference poses compared to what can be obtained from a single image.

In particular, we are interested in reference pose generation in the context of long-term localization, which is the problem of localizing images taken under different conditions, *e.g.* day-night or seasonal changes, against a scene captured under a reference condition. Given that scenes change over time, long-term localization is an important problem in practice. The main challenge in this setting is data association, *i.e.* establishing feature matches between images taken under different conditions. Naturally, this causes problems for generating reference poses using SfM algorithms, which themselves rely on local features such as SIFT [159] for data association. In previous work, we thus relied on human annotations to obtain feature matches between images taken under different conditions [237]. However, this approach is not scalable. Furthermore, human annotations of feature positions in images might not be too accurate, as they can easily be off by 5-10 pixels or more.

This paper is motivated by the observation that the reference poses for the nighttime test images of the Aachen Day-Night dataset [237, 239], obtained from human annotations, are not accurate enough to benchmark state-of-the-art localization methods. This paper thus proposes a semi-automated approach to reference pose generation. Our method is inspired by previous work on pose verification via view synthesis [276, 277, 288] and the observation that modern learned local features [74, 215] capture higher-level shape information. The latter allows feature matching between real images and 3D models, *e.g.* obtained via multi-view stereo [251]. As shown in Fig. F.1, our approach starts with a given initial pose estimate. It renders the 3D scene model from the current pose estimate. Feature matches between the actual and the re-rendered image are then used to refine the pose estimate. This procedure is repeated for a fixed number of iterations. Detailed experiments, for multiple ways to obtain initial poses, show that our approach yields more accurate pose estimates.

Re-rendering the image from its estimate pose enables visual inspection of the accuracy of the estimate. Using this aid, we observe that even larger differences in pose of 20cm or more can have little impact on the rendered image. This is not particularly surprising as the uncertainty of a pose estimate depends on the distance to the scene. However, it also implies that using fixed thresholds on the pose error to measure localization accuracy [257, 237] is not appropriate if there are significant changes in scene depth between test images. As a second contribution, we thus discuss and evaluate multiple evaluation measures that (explicitly or implicitly) use per-image uncertainty measures rather than global thresholds on pose errors.

In detail, this paper makes the following contributions: (**1**) we propose an approach based on view synthesis and learned features that can be used to generate reference pose for long-term visual localization benchmarks. (**2**) we provide a detailed experimental analysis

Figure F.2 – Multi-View Stereo reconstructions obtained from SfM models of the Aachen dataset using SIFT (left) and D2-Net (right) features (top-down view). D2-Net features are more robust to changes in conditions, *e.g.* day-night and seasonal changes, than classic SIFT features, but also produce more false positive matches. This leads to connecting unrelated scene parts during the SfM process and ultimately in an incorrect 3D model. In contrast, SIFT correctly reconstructs the scene. Some wrong placements are illustrated through colored ellipses.

of our approach, including studying different initialization approaches, different strategies for rendering and different features. (**3**) we show that the existing nighttime reference poses of the Aachen Day-Night dataset are not accurate enough to evaluate state-of-the-art long-term localization approaches. We further use our approach to obtain refined reference poses and show that current localization approaches achieve a much higher (up to 47%) pose accuracy than indicated by the original reference poses. (**4**) we extend the Aachen Day-Night dataset by additional nighttime test images, effectively doubling the number of available test images. We evaluate state-of-the-art localization approaches on the extended dataset and will provide a benchmark at visuallocalization.net. (**5**) we discuss and experimentally study additional evaluation measures. (**6**) we will make source code for our approach and our evaluation measures publicly available to facilitate the creation of new benchmarks. (**7**) we provide a concise review of current trends in the area of visual localization.

F.2 Related work

Besides discussing related work on benchmark creation for visual localization and the use of view synthesis for pose estimation and verification, this section also aims at giving an interested reader a concise overview over main trends in the area of visual localization.

Visual localization. Traditionally, most visual localization algorithms have been based on a combination of local features and a 3D scene model [267, 219, 154, 153, 56, 120, 235, 123, 305]. In most cases, the underlying 3D model is a sparse 3D point cloud constructed using SfM [250, 264] or SLAM [66, 187]. Each point in this model has been triangulated from two or more local image features such as SIFT [159] or ORB [227]. Thus, each 3D point can be associated with one or more local image descriptors. 2D-3D correspondences between local features in a query image and 3D model points can be found using nearest

neighbor search in the descriptor space. In turn, these 2D-3D matches can be used to estimate the camera pose of the query image by applying an n-point pose solver [106, 140, 141, 146, 3, 139, 86] inside a hypothesize-and-verify framework such as RANSAC [86] and its variants [58, 148, 213]. Research on such 3D *structure-based* methods has mostly focused on scalability, *e.g.* by accelerating the 2D-3D matching stage [154, 153, 56, 234, 73, 155, 123, 54] and the use of image retrieval [120, 239, 232, 276, 156, 38], by reducing memory requirements through model compression [154, 39, 37, 163, 76], or by making the pose estimation stage more robust to the ambiguities encountered at scale [153, 313, 275, 298, 4, 2].

Such approaches are computationally too complex for mobile devices with limited resources, *e.g.* robots and smart phones. In order to achieve real-time localization on such devices, non-real-time global localization against a pre-built map is combined with real-time local camera pose tracking [185, 176, 163, 248, 127, 75, 123, 297]. To this end, results from the localization process (most often 2D-3D inliers) are integrated into visual(-inertial) odometry or SLAM to prevent drift in the local pose estimates.

Structure-based approaches rely on underlying 3D models, which are expensive to build at scale and costly to maintain [238]. Alternatives to using 3D models are to triangulate the absolute pose of a query image from relative poses to multiple database images with known poses [315, 328], to estimate the absolute pose from 2D-2D matches with multiple database images [325], or by computing local SfM models on the fly [238].

Instead of explicitly using an underlying 3D model, *absolute pose regression* train a CNN to directly regress the camera pose from an input image [29, 59, 115, 131, 130, 172, 189, 212, 294, 300, 308]. However, they are not consistently more accurate than simple image retrieval baselines [7, 288, 287] that approximate the pose of a query image by the poses of the top-retrieved database images [240]. Furthermore, these approaches need to be trained specifically per scene. The latter problem can be overcome by *relative pose regression* techniques [15, 71, 147, 328, 230], which train CNNs to predict relative poses. In combination with image retrieval against a database of images with known poses, these relative poses can be used for visual localization. While recent work shows promising results [71, 230, 328], relative pose regression techniques do not yet achieve the same level of pose accuracy as methods explicitly based on 2D-3D matches.

Rather than learning the full localization pipeline, *scene coordinate regression* algorithms only replace the 2D-3D matching stage through a machine learning algorithm, typically either a random forest [257, 48, 46, 47, 174, 175, 167, 296, 295] or a CNN [24, 26, 25, 28, 167, 309, 327]. For a given patch from an image, these methods predict the corresponding 3D point in the scene. The resulting in a set of 2D-3D matches can then be used for camera pose estimation. Scene coordinate regression techniques constitute the state-of-the-art in terms of pose accuracy in small scenes. However, they currently do not scale well to larger scenes. For example, ESAC [25], a state-of-the-art scene

coordinate regression technique, localizes 42.6% of all daytime query images of the Aachen Day-Night dataset [239, 237] within errors of 25cm and 5°. In contrast, SIFT-based Active Search [234], a classical structure-based method, localizes 85.3% within the same error thresholds.

Learned local features. State-of-the-art approaches for long-term localization [74, 232, 99, 145, 266, 311, 20, 276, 277] are based on local features and explicit 3D scene models.[1] Classical handcrafted features such as ORB [227], SIFT [159], and SURF [19] struggle to match features between images taken under strongly differing viewing conditions, *e.g.* day and night or seasonal changes. Thus, long-term localization approaches typically use machine learning, both for image retrieval [7, 195, 211] and for local features [**N**, 200, 69, 20, 311, 74].

Traditionally, local feature learning has focused on learning feature descriptors [16, 30, 77, 179, 261, 260, 281, 282]. However, it has been shown that the local feature detector often is the limiting factor [276, 288, 237, 99]. Thus, recent work trains feature detectors and descriptors jointly [20, 69, 200, 311, 302, 195], leading to state-of-the-art feature matching performance for images taken under strongly differing conditions. Interestingly, using deeper layers of neural networks pre-trained on ImageNet [68] to define both feature detector and descriptor leads to very competitive performance [20, 74]. Equally important, such features are very robust to changes in different conditions, even though this might come at a price of more false positives (cf. Fig. F.2). We use this robustness to establish correspondences between real images and renderings of 3D models and the resulting 2D-3D matches to compute reference poses for benchmarking long-term visual localization. In addition, we benchmark state-of-the-art long-term localization approaches [232, 99, 215, 74] based on local features our reference poses.

Semantic visual localization. Besides using learned features that are more robust to changes in viewing conditions, long-term localization approaches also use semantic image segmentation [31, 97, 145, 266, 249, 252, 254, 277, 283, 284, 301, 312]. These methods are based on the observation that the semantic meaning of scene elements, in contrast to their appearance, is invariant to changes. Semantic image segmentations are thus used as an invariant representation for image retrieval [8, 283, 312], to verify 2D-3D matches [31, 145, 266, 284] and camera pose estimates [254, 266, 277, 284], for learning local features [97, 249], and as an additional input to learning-based localization approaches [31, 252, 301].

View synthesis. As shown in Fig. F.1, our approach iteratively renders a 3D model from a camera pose estimate and uses matches between the rendering and the actual image to refine the pose. Our approach takes inspiration from previous work on using view synthesis for pose estimation and verification. [258, 253] render detailed laser

[1] See also visuallocalization.net/benchmark/.

scans [258] respectively dense Multi-View Stereo point clouds [253] from new perspectives. They show that SIFT feature matching between the renderings and actual images is possible if both were taken from very similar poses. [288] show that view synthesis from very similar viewpoints (obtained from depth maps) improves SIFT feature matching between day and night images. [10] learn features that can be matched between paintings and renderings of a 3D model. In these works, view synthesis is used to create novel viewpoints in a given scene in order to enable camera pose estimation at all. In contrast, this paper focuses on using view synthesis to refine an initial pose estimate and to use it for generating reference poses for a long-term localization benchmark. Thus, the contributions of this paper center around a detailed experimental evaluation of the use of view synthesis to improve pose accuracy rather than on proposing a new method.

[276, 277] use view synthesis for automated pose verification. To this end, they render a dense laser scan point cloud from a set of given poses. They densely extract descriptors from each rendering and compare each descriptor against a descriptor extracted at the same pixel in the original image to compute an image-level similarity score. This score is then used to select the pose that best explains the input image. In contrast, this paper uses view synthesis to refine the camera pose estimates. While [276, 277] automate pose estimation, their approach still has room for improvement, even if additional information such as semantics is used [277]. Thus, we use the rendering for visual inspection of the poses rather than automating the verification process.

Visual localization benchmarks. This paper considers the visual localization problem, *i.e.* the task of computing the full camera pose for a given image. Closely related is the visual place recognition problem of determining which place is visible in a given image, without necessarily estimating its camera pose. However, we will not discuss pure place recognition datasets that do not provide full 6DoF camera poses such as [53, 274, 288, 286, 178].

Early localization benchmarks used SfM to reconstruct scenes from internet photo community collections such as Flickr. Query images were then obtained by removing some images from the reconstruction, together with all 3D points visible in only one of the remaining images [154]. Examples for this approach to benchmark creation are the Dubrovnik, depicting the old city of Dubrovnik (Croatia), Rome [154] and Landmarks 1k [153] datasets. The latter two datasets consists of individual landmarks in Rome respectively around the world. The same approach was later also used for images taken under more controlled conditions, *e.g.* the crowd-sourced Arts Quad [63, 153] dataset, the scenes from the Cambridge Landmarks [131] benchmark, and the San Francisco SF-0 [50, 153, 238] dataset. Similarly, RGB-D SLAM algorithms [191, 64] were used to obtain reference poses for the 7Scenes [257] and 12Scenes [295] datasets. Both depict small indoor scenes captured with RGB-D sensors.

Long-term localization benchmarks [237, 41, 14] typically use images captured under a

reference condition to represent the scene while images taken under different conditions are used as query. SLAM and SfM algorithms depend on data association between images. Thus, they tend to fail if images were taken under too dissimilar conditions. Using image sequences and / or multi-camera systems can allow using SLAM and SfM algorithms under stronger viewing condition changes. The former exploits the fact that it is not necessary to find matches between each query image and a reference image. Rather, finding enough matches for some query images is sufficient to register an entire sequence. The latter exploit the fact that a larger field-of-view typically leads to more matches. Both the SILDa [14] and (extended) CMU Seasons [11, 237] use sequences and multi-camera systems. SILDa depicts a single building block in London, UK under diferent conditions. The (extended) CMU Seasons dataset was constructed from images collected in and around Pittsburgh, US over the span of a year. For the (extended) CMU Seasons, additional humanly annotated matches were used in areas where cross-seasonal matching failed [237]. Human annotations were also used for the Mall [273] dataset to obtain initial pose estimates of test images with respect to a laser scan.

Manually annotated matches are often not very precise [238]. If available, additional sensors such as Lidar can be used to avoid the need for human annotations. The RobotCar Seasons [164, 237], depicting the city of Oxford, UK under various seasonal conditions, and the University of Michigan North Campus Long-Term Vision and LIDAR [41] datasets use Lidar data to obtain reference poses. However, human intervention might still be necessary if the scene geometry changes [237].

The Aachen Day-Night [239, 237] depicts the old inner city of Aachen, Germany. The 3D model of the scene was reconstructed from daytime images using SfM. Similarly, reference poses for daytime query images were also obtained using SfM. Since additional sensor data is not available and since SfM failed to provide reference poses [237], manual annotations were used for a set of nighttime query images. To this end, a daytime image taken from a similar viewpoint was selected for each nighttime query. The pixel positions corresponding to 10 to 30 3D points visible in the daytime image were then annotated manually. [237] estimated that the median mean position accuracy for the nighttime images is between 30cm and 40cm. However, in this paper, we show that the pose estimates are actually often worse. This observation motivates our approach for refining the original reference poses. We show that the refined poses are more accurate and are thus more suitable to measure the performance of state-of-the-art localization techniques. While this paper focuses on the Aachen Day-Night dataset, our approach is not specific to it and can be applied on other datasets as well.

F.3 Reference Pose Generation

Typically, a visual localization dataset provides a set of images $\mathcal{I} : \{\mathtt{I}_i\}_{i=1}^N$ and the corresponding reference poses $\mathcal{T} : \{\mathtt{T}_i\}_{i=1}^N$ in a 3D model $\mathcal{M}$. Our goal is to know

whether the poses $\mathcal{T}$ are accurate (verification) and get more accurate reference poses if necessary (refinement). Since each image in a visual localization dataset is usually treated individually, we consider a single image $\mathtt{I}$ and its (potentially inaccurate) pose $\mathtt{T}$ in this section. $\mathtt{T}$ represents the camera pose with respect to the model $\mathcal{M}$. More specifically, $\mathtt{T}$ is a 4×4 transformation matrix:

$$\mathtt{T} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathtt{0}_{1\times 3} & 0 \end{bmatrix}, \tag{F.1}$$

and $\mathbf{p} = \mathtt{R} \cdot {}_c\mathbf{p} + \mathbf{t}$ converts point coordinates in the camera frame ${}_c\mathbf{p}$ to the coordinates in the model.

Given the 3D model $\mathcal{M}$, we first render a synthesized view $\mathtt{I}^r$ (or multiple rendered images) at pose $\mathtt{T}$ (Section F.3.1). Then learned features are extracted and matched between the actual image $\mathtt{I}$ and the synthesized image $\mathtt{I}^r$. By analyzing the matched features, denoted as $\{\mathbf{u}_l\}_{l=1}^{N_f}$ and $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ for the actual and rendered images respectively, we can determine whether the pose $\mathtt{T}$ is accurate (Section F.3.2). Finally, we can back-project the 2D features from the rendered view $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ to the 3D model $\mathcal{M}$ to get a set of 3D points $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$. From the 2D-3D correspondences $\{\mathbf{u}_l\}_{l=1}^{N_f}$ and $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$, we can calculate a more accurate pose $\mathtt{T}^r$ for the actual image (Section F.3.3). The aforementioned process is repeated several times to get more accurate poses (cf. Fig. F.1). We also provide a method to quantify the uncertainties of the resulting poses (Section F.3.4).

For simplicity of presentation, we assume that all the 2D features $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ have a valid back-projection in $\mathcal{M}$ and all the 3D points $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$ are inliers in the refinement process. In practice, we remove 2D features with invalid depth (*e.g.* due to an incomplete the model $\mathcal{M}$) and reject outliers using LO-RANSAC [148]. For simplicity, we assume that the features are ordered based on matches: for a feature $\mathbf{u}_l$ in the real image, the corresponding matching feature in a rendering is $\mathbf{u}_l^r$.

F.3.1 Rendering Synthesized Views

There are different methods to render synthesized views from a pose $\mathtt{T}$ with respect to a scene model $\mathcal{M}$. In this work, we investigate view synthesis from two different scene models: a 3D point cloud with SIFT descriptors and a 3D mesh. In the process of generating reference poses using SfM, the scene is typically reconstructed as a 3D point cloud, where each point is associated with a descriptor, *e.g.* SIFT. A 3D mesh can be further generated using Multi-View Stereo. Therefore, these two models are readily available from the standard process for generating reference poses.

To render images from a 3D mesh, there are various off-the-shelf renderers that can be used. As for a point cloud with descriptors, we follow [203] and train a CNN to reconstruct the images from such a scene representation. The network uses a U-Net

architecture [222]. The input to the network is a 3D tensor of size $h \times w \times 129$, where h and w are the height and width of the image to be synthesized. The 129 channels consists of a depth channel and one channel per byte in the SIFT descriptor (128 bytes). The input is constructed by finding and projecting the visible points in the point cloud to the pose to render and then filling the input tensor at the pixel coordinates where there is a projected 3D point. The output of the network is the synthesized image at a given pose. For details of the method (*e.g.* training and evaluation), we refer the reader to [203]. While each rendering technique alone is sufficient in certain cases, combining the two rendering methods utilizes the information from different scene models and results in the best performance in our experiment (cf. Section F.5.4).

F.3.2 Matching Features with Synthesized views

To extract and match features between the real images $\mathtt{I}$ and the rendered images $\mathtt{I}_r$, we choose to use learning-based local features. This is due to the fact that the rendered images usually have large appearance change compared with the real night images. Traditional features, such as SIFT, rely on low level image statistics and are not robust to day-night condition change and rendering artifacts. In particular, we choose to use the D2-Net feature [74] in our pipeline, which uses a single CNN for joint feature detection and description and achieves state-of-the-art matching performance in challenging conditions.

For the images rendered using the two rendering techniques, we extract and match features between each rendered image and the real image individually. We then directly aggregate the feature matches obtained from both rendered images for the next step. Note that after obtaining the 2D feature matches, we can already verify whether there exists pose errors in the reference poses by checking the matching locations in the rendered and real images (cf. Fig. F.3 and Fig. F.8 for large and small pose errors respectively): if the real and rendered images are taken from the same pose, the two features $\mathbf{u}_l$ and $\mathbf{u}_l^r$ should be found at identical 2D positions (up to noise in the feature detection stage). Similarly, a large 2D distance $||\mathbf{u}_l - \mathbf{u}_l^r||_2$ is indicative for a significant difference in pose.

F.3.3 Refining Reference Poses

Given N_f matched features $\{\mathbf{u}_l\}_{l=1}^{N_f}$ and $\{\mathbf{u}_l^r\}_{l=1}^{N_f}$ between the real and rendered images, we first back-project the features in the rendered images to $\mathcal{M}$ to get the corresponding 3D points as $\{\mathbf{p}_l^r\}_{l=1}^{N_f}$

$$\mathbf{p}_l^r = \pi^{-1}(\mathbf{u}_l, \mathtt{T}, K, D, \mathcal{M}), \tag{F.2}$$

where $\pi : \mathbf{p} \rightarrow \mathbf{u}$ is the camera projection function and π^{-1} the inverse. K and D are the intrinsics and distortion parameters respectively. In practice, we get the depth map

at $\mathtt{T}$ in the process of rendering images from the 3D mesh, and the depth at $\mathbf{u}_l$ can be directly read from the depth map. After finding the 3D points, the refined reference pose $\mathtt{T}^r$ can be computed by solving a nonlinear least-squares problem

$$\mathtt{T}^r = \arg\min_{\mathtt{T}^r} \sum_{l=1}^{N_f} \|\pi(\mathbf{p}_l, \mathtt{T}^r, K, D) - \mathbf{u}_l\|^2. \tag{F.3}$$

We minimize (F.3) over the inliers of a pose obtained by LO-RANSAC.

F.3.4 Uncertainty Quantification

In addition to the refined pose $\mathtt{T}^r$, it is also important to have a quantitative measure about the uncertainty of the refined pose. Instead of the absolute pose uncertainty between the refined pose and the unknown ground truth, we propose to use an sampling strategy to compute the *sampling uncertainties*.

In particular, for a sampling ratio k (*e.g.* 50%), we first randomly sample from all the 2D-3D matches (*i.e.* all the inliers used in (F.3)). We then apply LO-RANSAC to the sampled subset and solve the nonlinear optimization problem (F.3) using the inliers returned by LO-RANSAC to get a pose $\mathtt{T}^s$. The sampling and solving process is repeated multiple times (typically $N_s = 50$ times in our experiment), resulting in multiple pose estimates $\{\mathtt{T}^s_n\}_{n=1}^{N_s}$. The sampling uncertainty for the camera position s^c_k and rotation s^r_k are calculated as

$$s^c_k = \text{median}(\{\epsilon^c_n\}_{n=1}^{N_s}) \quad s^r_k = \text{median}(\{\epsilon^r_n\}_{n=1}^{N_s}), \tag{F.4}$$

where $\epsilon^c_n, \epsilon^r_n = \mathtt{T}^r \boxminus \mathtt{T}^s_n$ (cf. (F.5)). We calculate the sampling uncertainties for different sampling ratios.

We would like to highlight the difference between the sampling uncertainties and the absolute uncertainties. The absolute uncertainties reflect the differences between the refined poses and the unknown ground truth, which cannot be calculated directly. The proposed sampling uncertainties, on the other hand, evaluate the variance with respect to the refined poses, which are essentially the local minima in the optimization problem (F.3). Therefore, the sampling uncertainties tend to be smaller than the actual uncertainties, since the local minima can hardly be the actual ground truth poses. Nevertheless, small sampling uncertainties still indicate that the refined poses are stable solutions for the given set of 2D-3D matches.

F.3.5 Discussion

The method proposed in this section essentially estimates more accurate poses from some potentially inaccurate initial estimates. Yet, it can not only be used to verify and refine existing reference poses, but also to easily extend existing visual localization datasets. For example, to add more images to an existing localization dataset, one only needs to provide coarse initial poses for these images, which can be obtained by, for example, manually selecting the most similar images. This is useful especially for images with large appearance difference compared with the localization database (e.g., adding nighttime images to a localization database constructed from daytime images), where accurate poses cannot be reliably estimated using SfM directly.

F.4 Metrics for Localization Accuracy

The reference poses generated using SfM or our method are inherently subject to inaccuracies, which complicates the evaluation process. For example, the difference between the reference pose and a pose to evaluate is no longer a meaningful metric if the actual error (*i.e.* the difference between the pose to evaluate and the unknown ground truth) is comparable to the uncertainty in the reference pose. Therefore, it is a common practice to set certain thresholds for the reference poses based on their uncertainties, and measure whether the poses to evaluate lie within those thresholds. Unfortunately, quantifying the uncertainties in the reference poses is a highly non-trivial task in itself. The actual uncertainties depend on various factors, such as the depth of the scene and the accuracy of the local features. In this section, we first discuss several performance metrics based on directly considering the uncertainties in pose space. We then discuss a performance metric based on the re-projection of the scene points, which removes the necessity of directly quantifying the pose uncertainty.

F.4.1 Direct Pose Uncertainty-Based Measures

Direct pose uncertainty-based measures analyze the position and rotation error between the reference and estimated poses. Typically, given a reference pose $\mathtt{T}$ and a pose to evaluate $\hat{\mathtt{T}}$, the position and orientation error $\epsilon^t, \epsilon^r = \mathtt{T} \boxminus \hat{\mathtt{T}}$ are computed as [237]:

$$\epsilon^t = \|\mathbf{t} - \hat{\mathbf{t}}\|_2, \quad \epsilon^r = \arccos(\frac{1}{2}(\mathrm{trace}(\mathtt{R}^{-1}\hat{\mathtt{R}}) - 1)). \tag{F.5}$$

To account for the uncertainties in the reference poses, we can either use a set of fixed thresholds for all the images in a dataset or define thresholds for each image individually.

Fixed error thresholds. We can define a set of N_e increasing error thresholds $E^{\text{fixed}} = \{\mathbf{e}_j^{\text{pose}}\}_{j=1}^{N_e}$, where $\mathbf{e}_j = (t_j, r_j)$ contains both position and orientation thresholds.

These thresholds apply to all the images in a dataset. A pose is said to be below a threshold $\mathbf{e}_j$ if $\epsilon^t < t_j$ and $\epsilon^r < r_j$. The overall localization accuracy is the percentages of images that are localized within these thresholds $O = \{o_j\}_{j=1}^{N_e}$, and higher values indicate better performance. For example, the error thresholds for Aachen night time images on visuallocalization.net are 0.5/1.0/5.0 m and 2.0/5.0/10.0 deg, and the localization accuracy is reported as three percentages corresponding to the these categories.

Sampling uncertainties as error thresholds. Using the same thresholds for all the images in a dataset, however, has limitations. The uncertainties are image-dependent if, as in our case, the poses are calculated by minimizing the reprojection errors of 2D-3D correspondences. The position uncertainty is lower for images observing landmarks that are closer to the camera. Ideally, these uncertainties should be taken into consideration to choose the error thresholds *per image*. Following the same idea in as Section F.3.4, we can use a set of sampling uncertainties $E_i^{\text{sample}} = \{\mathbf{s}_k\}_{k=k_1,k_2,\ldots}$, where $\mathbf{s}_k = \{s_k^t, s_k^r\}$ is the sampling uncertainty with sampling ratio k. For example, in our experiment, we use a set of thresholds calculated from sampling ratios of 50%, 30% and 10% respectively. Note that, as discussed in Section F.3.4, the sampling uncertainties tend to be lower than the (unknown) absolute uncertainties. Therefore using this metric tends to under-estimate the accuracy of localization algorithms.

F.4.2 Indirect Pose Uncertainty-Based Measures

To avoid the need to consider the uncertainties in 6 DoF poses (which is non-trivial as seen before), we follow the literature on object pose estimation and measure pose accuracy based on reprojections [111]. More precisely, we measure the difference between the reprojection of a set of 3D points in the reference and estimated poses. Intuitively, perturbations to the camera pose will result in the changes of the reprojected 2D locations of 3D points. Therefore, we can define certain thresholds around the reprojection of the 3D points as an *indirect* measure of the pose uncertainty. A key advantage of this approach is that the error thresholds can be defined is the image plane. While we use the same thresholds for all the images, this actually results in per-image uncertainty thresholds in pose space: the same change in reprojection error will typically result in a position error that increases with increasing distance of the camera to the scene. Formally, we define the following metric:

Maximum reprojection difference. The maximum distance between the projected points in the reference pose $\mathtt{T}_i^r$ and the estimated pose $\hat{\mathtt{T}}_i$ is used to measure the localization error:

$$r_i^{\infty} = \max_{l \in [1, N_f^i]} \|\pi(\mathbf{p}_l^r, \mathtt{T}_i^r) - \pi(\mathbf{p}_l^r, \hat{\mathtt{T}}_i)\|_2, \tag{F.6}$$

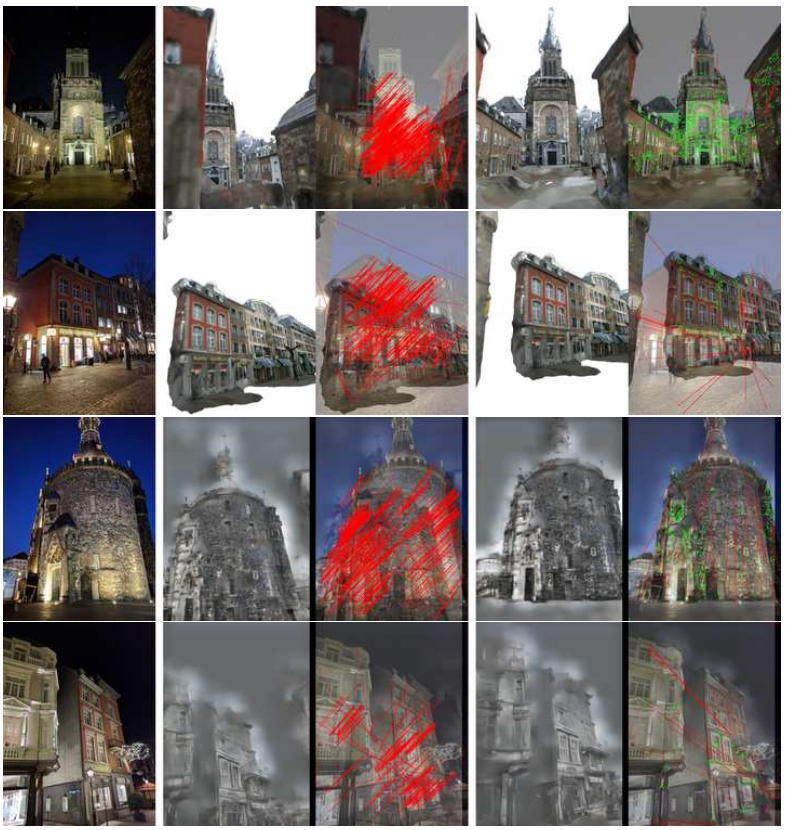


Figure F.3 – Comparison of images rendered from the original and refined (ours) reference poses of the nighttime images in Aachen Day-Night dataset. **First column**: nighttime images; **Second column**: images rendered from the existing reference poses, overlay of the rendering and the image together with D2-Net matches between the two; **Third column**: images rendered from our refined poses and the corresponding overlays with D2-Net matches. The top two rows render a Multi-View Stereo (MVS) mesh and the bottom two use Structure-from-Motion inversion [203] (invSfM). The colored lines visualize D2-Net feature matches. Green is used to indicate that the 2D location difference between a feature in the real image and its match in the rendered image is below 20 pixel.

where the intrinsics and distortion parameters are omitted for simplicity. Similar to the pose error, a set of reprojection thresholds $E^{\text{rep}} = \{e_j^{\text{rep}}\}_{j=1}^{N_e}$ are selected, and the percentages of the images with r_i^{∞} lower than these thresholds are used to indicate the overall accuracy on the dataset. We slightly abuse N_e here to denote the number of error thresholds in general.

F.5 Experimental Evaluation

To demonstrate the value of the proposed method, we first use our method to analyze the reference poses of the nighttime query images in the Aachen Day-Night dataset (Section F.5.2). Then, we extend the dataset with new nighttime query images and

generate the corresponding reference poses using our method (Section F.5.3). We also compare our method against baseline methods of directly matching features (SIFT and D2-Net) and computing poses via SfM models. To understand the impact of the different parameters in our method, we perform an extensive ablation study regarding different learned features, different rendering techniques, and the stability of our reference poses (Section F.5.4). Finally, we evaluate state-of-the-art localization methods on both the original and the extended Aachen Day-Night datasets based on the performance metrics discussed in Section F.4 (Section F.5.5).

In this paper, we focus on the Aachen Day-Night dataset [237, 239]. This is motivated by our observation that the reference poses for the nighttime images are the least accurate reference poses among the three datasets from [237]. At the same time, the dataset is becoming increasingly popular in the community, *e.g.* [232, 311, 302, 20, 74, 25, 179, 254, 54, 215, 99, 233, 314] have already been evaluated on the dataset. However, our approach is generally applicable and can be applied to other datasets as well. Note that we only consider the nighttime query images in this paper as SfM already provides accurate reference poses for the daytime queries of the Aachen Day-Night dataset.

F.5.1 Experimental Setup and Data Acquisition

Additional data capture. To extend the Aachen Day-Night dataset, we captured another 119 nighttime images and 119 daytime images with the camera of a Nexus 5X smart phone in July 2017. The nighttime and daytime images form pairs of photos taken from very similar poses. Registering the daytime images against the reference SfM model provided by the Aachen Day-Night dataset then yields initial pose estimates for the new nighttime queries.

Scene model generation. Our approach to refine camera poses requires an underlying 3D scene model. The Aachen Day-Night dataset provides a *reference SfM model* consisting of 4,328 database images and 1.65M 3D points triangulated from 10.55M SIFT features [237]. This publicly available reference model is a sub-model of a larger *base SfM model* that was reconstructed using COLMAP [250]. This base model also contains images from a set of videos as well as the daytime queries, resulting in a SfM model with 7,604 images and 2.43M 3D points triangulated from 17.75M features. This model was registered against the original Aachen SfM model from [239] to recover the scale of the scene. The reference model was obtained by removing the sequences and query images from the base model.

We started from the base model and created an *extended SfM model*. We registered the additional daytime images and an additional image sequence[2] against the base model while keeping the poses of the base model images fixed. The resulting model contains

[2]Using one of the original videos and extracting images at a higher frame rate.

12,916 images and 3,90M 3D points triangulated from 32.19M SIFT features. We used this extended base model when creating our new reference poses.

We removed all query images and the newly added sequence images from the extended base model to create an *extended reference SfM model* consisting of 6,697 images and 2.32M points triangulated from 15.93M SIFT features. This model will be used to benchmark localization algorithms on our extended Aachen Day-Night dataset. We will make this new reference model publicly available, but will withhold the base models and the reference poses for the query images. Instead, we will provide an evaluation service on visuallocalization.net. The motivation behind publishing this smaller dataset is to make sure that the reference poses were computed from additional data not available to localization algorithms. The inclusion of the original sequences is necessary as some of the newly added nighttime queries depict places not covered in the original reference model.

In addition to the extended models, we also created a colored 3D mesh of the scene. We used COLMAP's Multi-View Stereo pipeline [251] to obtain a dense point cloud. Screened Poisson surface reconstruction [128] of the point cloud then yields a colored mesh.

Rendering. Our method requires rendering the scene from estimated poses. For each pose, we generate two renderings: (1) we render the MVS *mesh*, (2) we use the SfM inversion approach (*invSfM*) from [203] to recover an image directly from a rendering of the extended base model. We use our own implementation of invSfM. Note that we only use the CoarseNet stage and skip the VisibNet and RefineNet. We use the MVS mesh to determine which points are visible instead of VisibNet. While skipping RefineNet reduces image quality, we found the results to be of sufficient quality.

Fig. F.3 shows example renderings obtained from the mesh and invSfM.

Implementation details. If not mentioned otherwise, we extract D2-Net features [74] from both rendered images. The refinement process is repeated for 5 iterations. We use single scale features since the initial pose estimates are accurate enough such that multi-scale processing is not required. To determine whether our refinement succeeded, we only accept the refined pose when there are more than 10 effective inliers[3] found by LO-RANSAC [148, 236] from the input 2D-3D matches, using the P3P solver from [139]. More precisely, we subdivide each image into a 50×50 grid and count at most one inlier per cell. The cell size and the inlier threshold are determined experimentally.

[3]The effective inlier count takes the spatial distribution of the matches in the image into account. It has been shown to be a better measure than the raw inlier count [120].

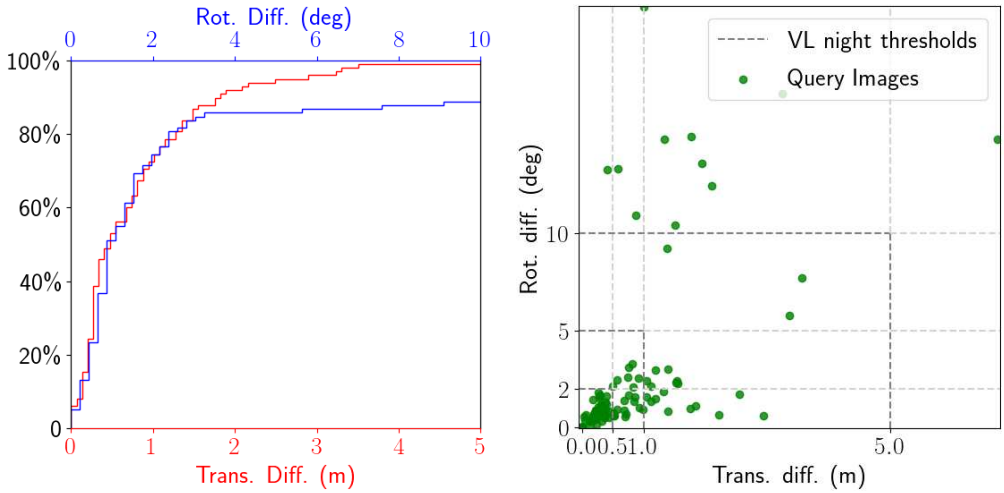


Figure F.4 – Differences between the original reference poses and the refined reference poses (our method). **Left**: Cumulative distribution of position and rotation differences. **Right**: Distribution of the position and rotational differences. The position and rotation thresholds (0.5/1.0/5.0 m, 2/5/10 deg) used in [237] and visuallocalization.net (VL) are also shown for reference.

Table F.1 – Evaluation of state-of-the-art localization methods in the original Aachen nighttime images. We evaluate results submitted by the authors to visuallocalization.net on both the original and our refined poses. We compare the methods based on the Pose Error, *i.e.* the percentage of queries localized within fixed error thresholds of the reference poses. As can be seen, our more accurate reference poses yield a better measure of pose accuracy. For our poses, we also report results for two additional metric: the percentage of queries localized within sampling-based thresholds (Sampling) of the reference poses (cf. Section F.4.1) and the percentage of queries with maximum reprojection errors within given error thresholds in pixels (Reprojection Diff.) (cf. Section F.4.2).

	Original Poses	**Refined Poses**		
	Pose Error	Pose Error	Sampling	Reprojection Diff.
	0.5m,2°/1m,5°/5m,10°	0.5m,2°/1m,5°/5m,10°	(50%/30%/10%)	(10/20/50/100 px)
Active Search v1.1 [237]	27.6/38.8/56.1	48.0/57.1/64.3	2.0/4.1/11.2	28.6/39.8/52.0/62.2
D2-Net [74]	45.9/68.4/88.8	86.7/96.9/100.0	7.1/13.3/35.7	46.9/68.4/89.8/98.0
DELF [195]	39.8/61.2/85.7	75.5/89.8/96.9	4.1/5.1/14.3	28.6/56.1/78.6/88.8
DenseVLAD [288] + D2-Net [74]	39.8/55.1/74.5	75.5/81.6/84.7	7.1/8.2/24.5	45.9/65.3/77.6/82.7
Hierarchical Localization [232]	42.9/62.2/76.5	77.6/87.8/88.8	7.1/9.2/24.5	41.8/65.3/78.6/85.7
NetVLAD [7] + D2-Net [74]	43.9/66.3/85.7	90.8/96.9/96.9	8.2/11.2/40.8	51.0/75.5/92.9/95.9
R2D2 V2 20K [215]	46.9/66.3/88.8	90.8/99.0/100.0	8.2/14.3/36.7	51.0/70.4/92.9/95.9

F.5.2 Refining the Original Aachen Nighttime Poses

In a first experiment, we analyze the accuracy of the reference poses for the 98 original nighttime queries of the Aachen Day-Night dataset. We show that the original reference poses are inaccurate and that our refinement approach considerably improves the pose

accuracy.

Our approach used the original poses for initialization. For 3 out of the 98 images, our method failed to find sufficiently many 2D-3D matches, mostly due to an incomplete mesh (see Section F.5.4). For the failure cases, we simply kept the existing reference poses.

Qualitative evaluation. Fig. F.3 visually compares the original reference poses with our refined poses. As can be seen, the existing reference poses, obtained from manual annotated 2D-3D matches, can be rather inaccurate. In contrast, our method generates reference poses such that the rendering from the refined pose is visually consistent with the actual image. Thus, features matching between the real and rendered images are found at the same positions (up to noise), as can be see from the (short) green lines. Fig. F.3 shows selected examples where the original reference poses were rather inaccurate. Visual comparison between the original and our refined poses showed that our approach consistently produced more accurate poses for all nighttime queries.

We would like to highlight that our method is robust to large initial pose error. This can be seen from the top row in Fig. F.3, where the initial rendered image exhibits a significant viewpoint change with respect to the actual image.

It is also worth noting that D2-Net features can provide robust matches even though the rendered images (using a model reconstructed from daytime imagery) are visually very different from the actual images and contain non-trivial rendering artifacts.

Quantitative evaluation. To quantify the differences between the original and our reference poses, we computed the differences in camera position and orientation (see (F.5)). Fig. F.4 shows the results of this comparison. It can be seen that there exists a non-trivial discrepancy between the original and refined reference poses.

[237] measures localization accuracy by the percentage of nighttime query poses estimated within (0.5 m, 2 deg), (1 m, 5 deg), and (5 m, 10 deg) of the reference poses. These thresholds are also shown in Fig. F.4. As can be seen, the differences between the original and refined poses fall outside of the largest error threshold for 11 images ($\sim 11.2\%$ of all the nighttime queries). Interestingly, the best results reported on visuallocalization.net register 88.8% of the nighttime queries within 5 m and 10 deg. Thus, state-of-the-art methods might actually be more accurate than the reference poses.

Finally, Table F.1 evaluates several state-of-the-art localization methods using the existing and refined reference poses. As can be seen, the accuracy of the localization methods is indeed (significantly) under-estimated by the existing reference poses. In contrast, our reference poses allow us to measure localization performance more accurately. Note that Table F.1 also provides results for additional evaluation measures for our new reference

Figure F.5 – Pairs of day-night images taken from similar poses. We obtain reference poses for the daytime images via SfM. The resulting poses are used to initialize our approach for generating poses for the nighttime images. For the SIFT and D2-Net registration baselines, an additional 20 daytime images that overlap with these images are selected from the base model for the daytime image in each pair.

poses. These results will be discussed in Section F.5.5.

Summary. Our results clearly show that our new reference poses are more accurate than the original poses. We will integrate our new poses in the visuallocalization.net online benchmark, allowing us to easily update all results on the website.

F.5.3 Extending the Aachen Day-Night Dataset

Our approach is capable of estimating an accurate pose from a coarse initialization. Besides verifying and refining existing reference poses, our approach can also be used for generating reference poses for new images. In the next experiment, we thus extend the Aachen Day-Night dataset by additional nighttime queries. We compare our reference poses with two registration baselines using SIFT and D2-Net features, respectively.

Reference pose generation. As shown in Fig. F.5, we captured a daytime photo from a similar pose for each the 119 new nighttime images. The poses of these daytime images in the extended base model, obtained via SfM, then provide initial pose estimates for the nighttime queries that are subsequently refined by our approach. We excluded images for which our method resulted in less than 10 effective inliers to avoid unreliable reference poses. This results in reference poses for 93 out of the 119 images.

We compare our method with two baselines using SIFT and D2-Net features, respectively. Both baselines match features between the 93 new nighttime queries and a small set of images in the extended base SfM model. For a nighttime query, this set includes the corresponding daytime image $\mathcal{I}_D$ as well as the 20 images in the extended base model

that share the most 3D points with $\mathcal{I}_D$. 2D-2D matches between the nighttime image and the daytime photos in the set then yield a set of 2D-3D correspondences based on the 3D points visible in the latter. COLMAP's image registration pipeline was then used to obtain the camera pose based on these matches. Note that for D2-Net features, we re-triangulated the extended base 3D model before day-night feature matching.

(a) Night-day matches for which SIFT registration failed.

(b) Refinement process of our method (1st iteration).

Figure F.6 – Typical failure cases of the SIFT registration baseline. **Top**: nighttime images where SIFT registration failed and the corresponding daytime images; **Bottom**: Visualization of the first iteration of our method (left: initial pose; right: refined pose). The differences between D2-Net features and the projection of the matching 3D points are color coded according to the direction in the image plane (cf. legend in the top-right).

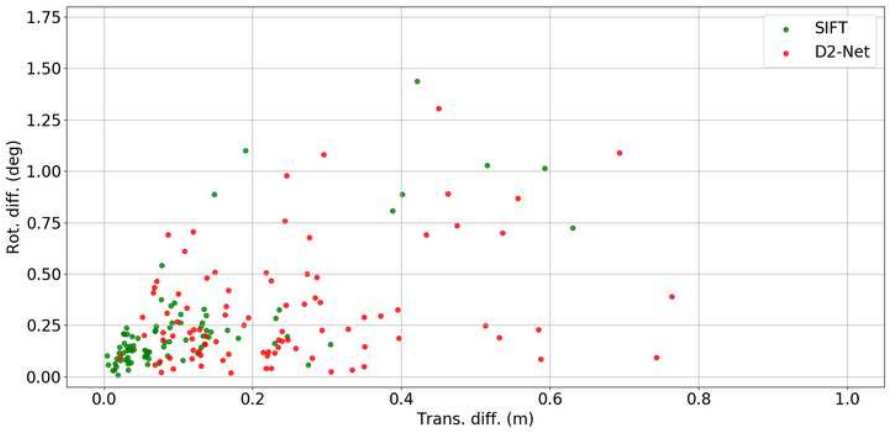


Figure F.7 – Distribution of the pose difference between our method and the two registration baselines.

Robustness. Both the D2-Net baseline and our method are able to consistently estimate poses for challenging images for which the SIFT baseline fails. Fig. F.6 shows such failure cases of SIFT. In each of the shown cases, there is a strong light source in the scene, causing significant appearance differences between the day and nighttime images. SIFT is not able to deal with these strong changes. In contrast, our method, as well as the D2-Net baseline, which relies on high level learned features, are able to handle these cases (cf. Fig. F.6b).

[237] reported that the reference poses obtained via SfM and SIFT were unreliable. Interestingly, we observe the opposite for many images in our experiments. We attribute this to the inclusion of the corresponding daytime images: as shown in [288], SIFT features better handle day-night changes under small viewpoint changes. Note that daytime images taken from very similar poses are not available for the original nighttime queries.

Quantitative evaluation. Excluding the failure cases, we computed the pose differences between our method and two baselines. The results of this comparison are shown in Fig. F.7. Interestingly, the poses from our method and the SIFT registration are very consistent. For the majority of the images, the pose difference is below 0.2 m and 0.5 deg. In contrast, we observe much larger difference between our poses and the D2-Net registration baseline. As there is no external reference poses that can be used to calculate the absolute pose accuracy, we resort to visual inspection based on the renderings.

Visual inspection. Fig. F.8 analyses example poses obtained by the D2-Net baseline. Besides overlaying the real and rendered images, we also show D2-Net features matches between the two. For each match, we compute the 2D offset between the feature positions in the real and the rendered view. Following [250], we color-code the features based on

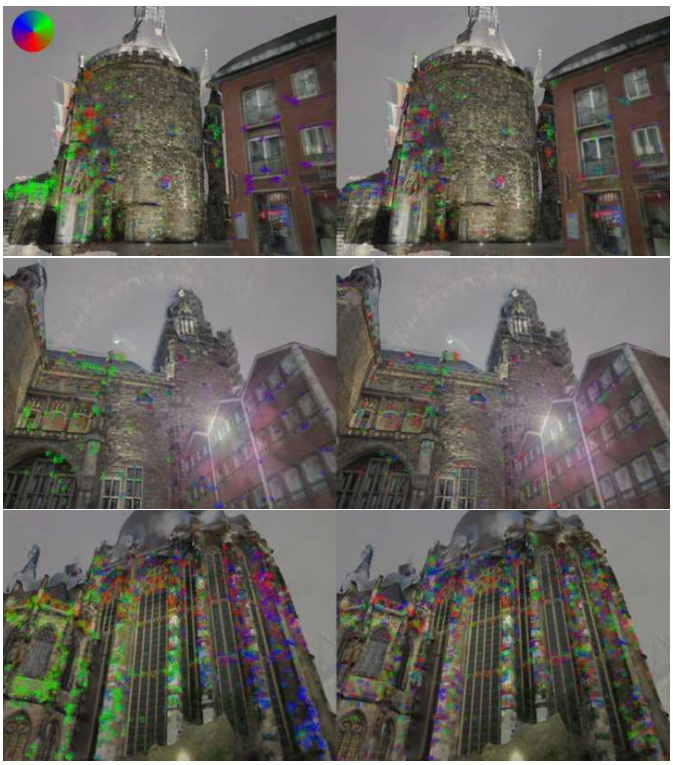
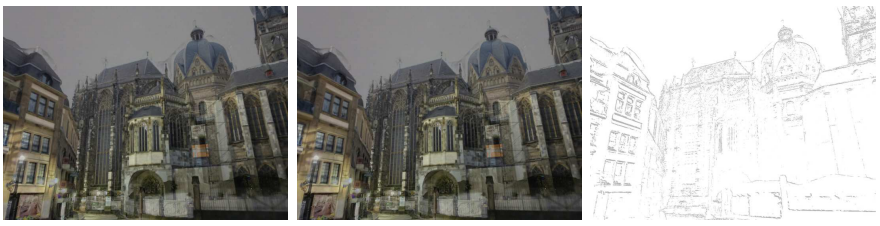


Figure F.8 – Comparing the D2-Net baseline against our refinement. **Left**: overlay of real photos and images rendered with D2-Net poses. D2-Net features in the rendered images are connected to the matching locations in the real images (circles), and the color indicates the direction of the feature location differences in the two images (see legend in the top-left). **Right**: corresponding visualization using poses obtained by one iteration of our method (initialized with D2-Net poses). The patterns of the feature directions in the left images indicate the inaccuracy in the poses from D2-Net registration, which are corrected with our method (right images).

the directions of these 2D offsets. As argued in [250], these directions should be randomly distributed for accurate pose estimates. Patterns of similar direction in the same region of an image indicate a shift between the two images and thus pose errors.

The D2-Net poses in Fig. F.8 are visually more accurate than those in Fig. F.3 and Fig. F.6b. Still, we observe clear patterns in the distribution of the directions (*e.g.* the concentration of green color on one side and purple on the other), which indicates inaccuracies in the poses of the D2-Net baseline. We further used one iteration of our method to refine the D2-Net poses. As can be seen in Fig. F.8, the refinement improves the distribution of directions. We conclude that our approach is able to provide more accurate than the D2-Net baseline.

As can be seen from Fig. F.7, the pose differences between our approach and the SIFT baseline are significantly smaller than the differences between our approach and D2-Net. Unlike for D2-Net poses, we did not see strong feature direction patterns for the SIFT



(a) Typical visual difference between poses the SIFT baseline and our method. Left and Middle: overlay of the rendered and real images for SIFT respectively our poses. Right: the intensity difference of the rendered images. The images are converted to 8-bit gray-scale images, and the pixels with intensity difference larger than 10 are shown in gray.

(b) Example where the pose from our method (right) is more accurate than the SIFT pose (left), as can be seen from the sign in the middle of the cutouts.

(c) Example where the SIFT pose (left) is more accurate than our method (right), as can be seen from the windows and the edge of the roof in the cutouts.

Figure F.9 – Visual comparison of images rendered from the poses obtained by our method and the SIFT baseline.

poses. We therefore omit the corresponding visualizations. We observe that if the SIFT baseline is able to estimate a pose it is usually visually similar to the pose obtained with our approach (cf. Fig. F.9a). There are images where the poses from our method seem to be visually more accurate than the SIFT registration and vice versa (shown in Fig. F.9b and F.9c, respectively). Yet, overall there are only 7 out of the 93 new nighttime queries for which we consider the SIFT poses to be visually more accurate than the poses provided by our method. For these images, we use the SIFT poses as reference poses. At the same time, SIFT failed to provide poses for 5 of the nighttime images due to a lack of sufficient matches.

Discussion and summary. It is interesting to see that SIFT poses are not necessarily more accurate than our poses. SIFT features are much more accurately localized in images than D2-Net features [74]. Thus, one might have expected that a few accurately localized SIFT matches are better than many less accurately localized D2-Net matches. Yet, finding more matches with D2-Net between the renderings and the real images seems to compensate for the inaccuracy of the D2-Net feature detections.

For the newly acquired nighttime images, we observe that our approach performs similar

to SIFT in terms of accuracy. In this case, SIFT benefits from daytime images taken from similar viewpoints. As evident from the failure cases of SIFT on both the original and new queries, our approach is more robust than the SIFT baseline. As a result, our approach is better suited to for reference pose generation for datasets that benchmark long-term visual localization algorithms.

Compared with the D2-Net baseline, the poses resulting from our method are more accurate. The main difference between the D2-Net baseline and our approach is the use of rendered images. The results thus validate our choice to iteratively render the scene from the current pose estimate and match features against the rendering. Moreover, as seen from the analysis of the D2-Net baseline, the ability of our method to verify and refine existing poses is also valuable when it is combined with other approaches.

F.5.4 Ablation Study

Next, we present ablation studies to analyze our proposed approach. We first obtain an estimate for the stability of our reference poses. Next, we determine the impact of using different features and rendering techniques, which are the two key ingredients in our method. Finally, we show failure cases of our method.

Pose stability. To provide a quantitative measure of the uncertainties/stability of the reference poses obtained with our method, we compute the sampling uncertainties as described in Section F.3.4 for both the original and additional nighttime images: we randomly sample a percentage of 2D-3D matches from the inliers used to estimate the reference poses. This sample is then used to obtain another pose estimate. The differences between these new and our poses provide a measure for the stability of the minima found by our approach.

We used three sampling rates that use 90%, 50%, and 10% of the inliers, respectively. For each rate, we drew 50 random samples and report the median position and orientation differences. In addition, since our method uses different rendering techniques and is an iterative process, we also computed the following for comparison:

- *Compare-InvSfM*: the differences between the refined poses using both types of rendered images and using InvSfM only;
- *Compare-Mesh*: the differences between the refined poses using both types of rendered images and mesh rendering only;
- *Compare-Prev-Iter*: the pose differences between the two last iterations of our refinement process.

The results of our comparisons are shown in Fig. F.10. For the original images, more

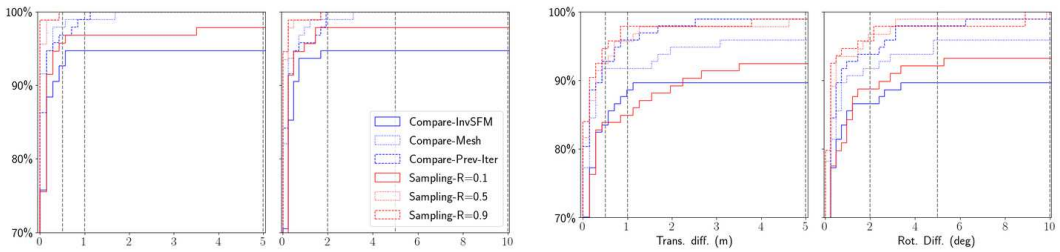


Figure F.10 – Different uncertainties for the original (**top**) and additional (**bottom**) Aachen Night images. The vertical dash lines corresponds to the error thresholds proposed in [237] and used by the online benchmark.

than 90% of the images are below the finest error threshold (0.5 m, 2 deg) of the visual localization benchmark, independently of which sampling rate and rendering is used. For the additional images, the uncertainties are higher. Still, more than 80% of the images fall in that threshold as well. The fact that the uncertainties of the additional images are overall higher than the original images indicates that the newly added images might be more challenging. Regarding the different rendering techniques, images rendered using the MVS mesh seem to provide more information for the final refined poses, as *Compare-Mesh* shows less uncertainty than *Compare-InvSfM*.

While it is difficult to quantify the absolute uncertainties, the uncertainties shown in Fig. F.10 indicate that the reference poses generated using our method are at least stable solutions considering the available 2D-3D matches. This can be seen from the fact that even using as little as 10% of the available inlier matches leads to very similar pose estimates for nearly all images.

Different features. Instead of using D2-Net features, we also used SIFT and R2D2 [215] features to obtain matches between the rendered and real images.

Fig. F.11 compares the results obtained with different types of features. As can be seen, SIFT failed to find enough matches in most cases for both the original and additional night images. This is not surprising considering SIFT relies on low-level image statistics, which are strongly impacted by imperfections in the MVS model and the invSfM rendering process. In contrast, both D2-Net and R2D2 features were able to find enough matches for most of the original Aachen night images. The success rate for both features drops on the additional Aachen night images, where the D2-Net feature performed better. Plotting the reprojection error (after nonlinear optimization) against the number of effective inliers, we observe a clear trend across different features: D2-Net recovers the most matches, followed by R2D2 and SIFT; while SIFT features were most accurately localized in the images, D2-Net has the largest reprojection errors.

To see how the number of effective inliers and reprojection error translates to the quality

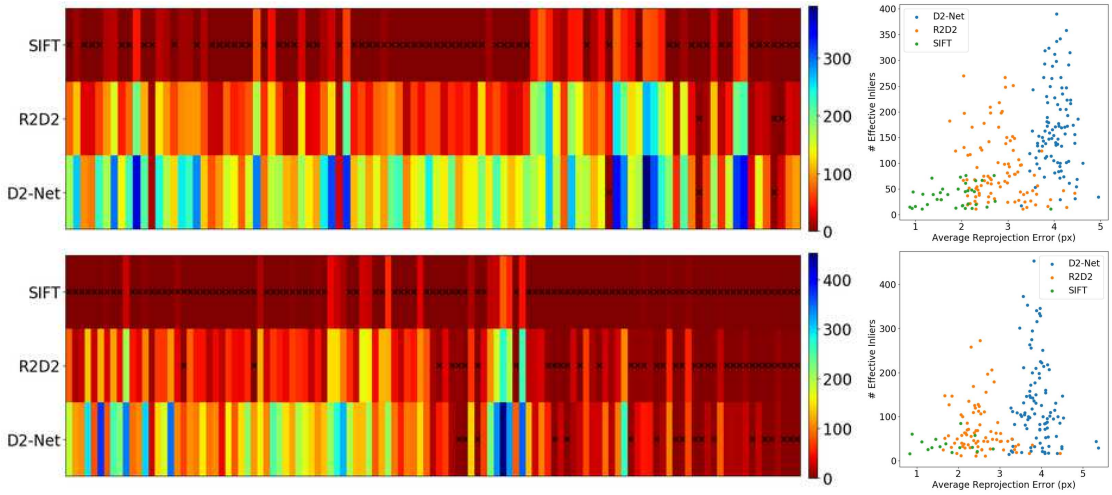


Figure F.11 – Effect of using different features in our method. **Left**: the number of effective inliers for each image. Each block along the horizontal axis corresponds to one image. A black cross indicates there are less than 10 effective inliers, *i.e.* the pose is likely not reliable. **Right**: the number of effective inliers and the mean reprojection error (after nonlinear optimization) for different features. Failure cases (*i.e.* the black crosses) are excluded. The top row shows the result for the original Aachen nighttime images, and the bottom for additional images.

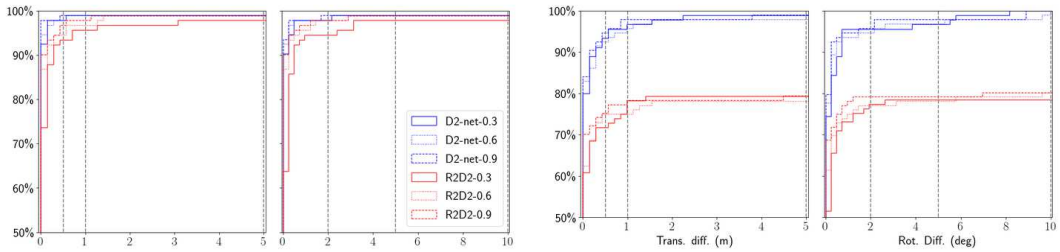
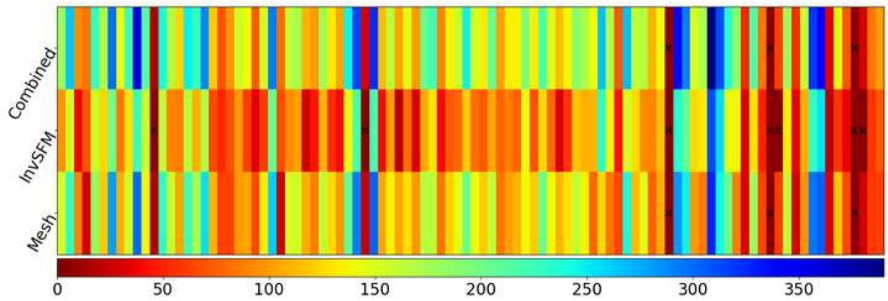


Figure F.12 – Sampling uncertainties of the D2-Net and R2D2 poses for the original (**top**) and additional (**bottom**) Aachen night images. Median position and orientation errors over 50 random samples are shown.

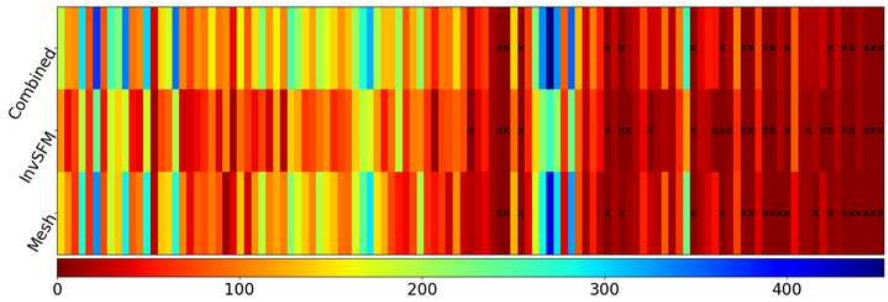
of the refined poses, we further computed the sampling uncertainties for D2-Net and R2D2, shown in Fig. F.12. We excluded SIFT since it failed for most of the images. It can be seen that the refined poses from D2-Net features are more stable than the R2D2 poses for both the original and additional images.

The results validate our choice of using D2-Net features to match between real and rendered images as they better handle imperfections in the renderings.

Different rendering techniques. The experiments presented so far used both rendering types (using MVS mesh and the invSfM process). Next, we compare using both types against using only one of the two using the number of effective inliers.



(a) Original Aachen night images.



(b) Additional Aachen night images.

Figure F.13 – The number of effective inliers for D2-Net features when different rendering techniques are used. The visualization is the same as Fig. F.11.

As can be seen in Fig. F.13, using renderings based on the MVS mesh in general resulted in more effective inliers compared to using invSfM for rendering. Accordingly, there are more images where our method could find sufficient effective inliers in the images rendered from mesh. This is also consistent with our results in Fig. F.10, that show that the poses based only on mesh rendering are more accurate than those obtained using only invSfM. Yet, there are a few cases where mesh rendering fails while invSfm rendering succeeds. The corresponding nighttime images show parts of the model that are only sparsely covered by images and where the MVS reconstruction is thus incomplete. The invSfM process seems to be more stable for such cases.

Combining the 2D-3D matches obtained from both types of renderings increases the number of effective inliers. Note that the effective inlier count selects at most one inlier for each 50 pixels by 50 pixels region in an image. A higher effective inlier count thus indicates that the matches found by the two rendering types are somewhat complimentary as matches are found in diferrent image regions. Moreover, there are a few cases (right part of Fig. F.13b) for which using both rendering types is necessary to obtain sufficiently many inliers.

The results validate our choice of using both rendering techniques as they are (partially) complimentary.

(a) Failure mode 1: low quality mesh. **(b)** Failure mode 2: scene with little texture.

Figure F.14 – Typical failure cases of our method. Left: real nighttime images; Right: MVS mesh renderings from the initial pose.

Table F.2 – Localization accuracy using different metrics on the extended Aachen Day-Night dataset. We compare the methods based on the Pose Error, *i.e.* the percentage of queries localized within fixed error thresholds of the reference poses. As can be seen, our more accurate reference poses yield a better measure of pose accuracy. For our poses, we also report results for two additional metric: the percentage of queries localized within sampling-based thresholds (Sampling) of the reference poses (cf. Section F.4.1) and the percentage of queries with maximum reprojection errors within given error thresholds in pixels (Reprojection Diff.) (cf. Section F.4.2). The same error thresholds as Table F.1 are used.

	Original Night Images			**All Night Images**		
	Pose Error	Sampling	Reprojection Diff.	Pose Error	Sampling	Reprojection Diff.
D2-Net	90.8/98.0/98.0	11.2/19.4/43.9	56.1/80.6/92.9/95.9	90.6/97.4/97.9	6.3/11.0/30.9	36.1/73.8/91.1/96.9
R2D2-20k	90.8/95.9/95.9	7.1/11.2/38.8	54.1/76.5/89.8/93.9	88.5/94.8/96.3	5.2/7.9/29.8	40.8/72.8/91.6/94.8
R2D2-40k	91.8/98.0/98.0	7.1/13.3/44.9	56.1/76.5/92.9/95.9	88.5/95.3/97.9	5.8/8.9/33.0	41.9/73.3/91.6/95.8

Failure cases. Fig. F.14 shows examples of two typical failure cases of our method. The first failure mode is when the nighttime image was taken in a part of the scene where the MVS mesh is of low quality, *e.g.* parts of the surface have not been reconstructed (cf. Fig. F.14a). This could be overcome by using a more complete/higher quality mesh of the scene, but might require additional data capture. The second failure mode is caused by weakly textured scenes (cf. Fig. F.14b). In the shown example, the rendered image is of reasonable quality visually. However, due to the lack of texture, our method failed to find enough matches between the rendered image and the real night image. Using contour edges as an additional feature type could help avoid this failure mode. However, edges are also typically harder to match than local features. Furthermore, care would need to be taken to handle protruding regions in the MVS model.

F.5.5 Evaluation of State-of-the-Art Methods

Table F.1 evaluates published state-of-the-art localization methods using our new reference poses for the original nighttime images. The results were obtained by re-evaluating poses submitted to visuallocalization.net.[4] In the following, we present results for state-of-the-

[4]There results available at visuallocalization.net for methods that outperform the approaches used in Table F.1. For our experiments, we limited ourselves to methods that have been published in peer reviewed conferences and journals. Updated results for the other methods will be available on the

art methods on our new extended Aachen Day-Night dataset. Note that the extended dataset uses a larger reference SfM model than the original one and we thus cannot use results from the benchmark website.

Given that D2-Net and R2D2 features achieve state-of-the-art results in Table F.1, we use two image retrieval-based approaches based on these features in our evaluation. Both approaches first re-triangulate the reference SfM model with feature matches between the reference images found by D2-Net respectively R2D2. Next, NetVLAD [7] is used to retrieve the 20 most similar reference image for each nighttime query. Feature matches between each query and its retrieved image yield a set of 2D-3D matches via the 3D points visible in the reference images. These 2D-3D matches are used for pose estimation against the reference model inside COLMAP.[5] For R2D2, we provide results for two variants that use at most 20k (R2D2-20k) respectively 40k (R2D2-40k) features per image.

Table F.2 shows the results of our experiments using the evaluation measures discussed in Section F.4. Similarly, Table F.1 also shows results for all metrics for our new reference poses. Overall, the accuracy is lower when considering all nighttime queries compared to only focusing on the original night images, independent of the metric used. This indicates the newly added images might be more challenging. In the following, we discuss the results per evaluation metric.

Pose error with fixed thresholds. We consider the three fixed error thresholds used in [237] and on the benchmark website, *i.e.* (0.5 m, 2 deg), (1 m, 5 deg), and (5 m, 10 deg). Based on the this metric, the performance on the original and extended Aachen dataset seems saturated for certain algorithms (*e.g.* D2-Net and R2D2). However, these thresholds were originally chosen to take the uncertainties in the original nighttime reference poses into account. As shown in our previous experiments, our new reference poses are significantly more accurate. As such, using rather loose thresholds could lead to an overestimate in the localization accuracy. Furthermore, as discussed in Section F.4, using the same thresholds for all images does not take into account that the uncertainty in the pose depends on the distance of the camera to the scene.

Sampling uncertainties as error thresholds. The second metric aims at computing error thresholds on the camera pose per image. For each reference pose, we randomly sampled set containing 10%, 30% and 50% of the inliers of our method. For each sampling percentage, we drew 50 samples and computed the median position and orientation difference between the poses obtained from the samples and the reference poses. These median differences were then used as the error thresholds.

As can be seen from Table F.1 and Table F.2, the sampling uncertainties tend to

benchmark website once we update the reference poses.

[5] Based on code available at https://github.com/tsattler/visuallocalizationbenchmark.

underestimate the localization performance of the different methods. This is due to the fact that our reference poses are rather stable under using a subset of the inlier matches (cf. Fig. F.12). The sampling uncertainties reflect the stability of the local minimum reached in the refinement process, rather than the absolute uncertainties. Thus, this metric should not be used to evaluate localization performance.

Maximum reprojection difference. Our reference poses are obtained by minimizing a reprojection error in image space, rather than an error in camera pose space. Thus, evaluating localization algorithms based on the quality of their reprojections seems a natural metric, especially if these algorithms compute poses by minimizing an image space error.

For each 3D point in the inlier 2D-3D matches of the reference poses, we compute a reprojection difference between the reference and an estimate pose. For each image, we report the maximum difference and we compute the percentages of images that have a maximum reprojection difference below 10, 20, 50 and 100 pixels. Since all nighttime images have a resolution of 1600×1200 pixels, these thresholds correspond to 0.5%, 1%, 2.5%, and 5% of the image diagonal.

Comparing the results with the pose error metric using fixed thresholds, we can see that although the top performing algorithms achieve approximately 90% in the finest pose error category, they only have $70-80\%$ of all the images that were localized within 20 pixel according to the maximum reprojection difference. Even less images are localized within 10 pixels. Since the accuracy of local features are typically below 5 pixel (cf. Fig. F.11(right)), this indicates that there is still much room for improvement on our extended version of the Aachen Day-Night dataset. As such, we believe that the maximum reprojection error metric should be the metric of choice for this dataset.

F.6 Conclusion

In this paper, we have considered the problem of creating reference camera poses for long-term visual localization benchmark datasets. In this setting, classical features often struggle to obtain matches between images taken under strongly differing conditions. At the same time, human annotations are both time-consuming to generate and not necessarily highly accurate. Thus, we have presented an approach for refining reference poses based view synthesis and learned features that allow robust feature matching between real and rendered images. In addition, we have discussed multiple metrics for evaluating localization performance.

The main contribution of this paper is an extensive set of experiments. We have shown that the original nighttime reference poses of the Aachen Day-Night dataset are rather inaccurate. As a result, the localization accuracy of state-of-the-art methods is currently

drastically underestimated. Using our approach, we have created a more accurate set of reference poses. We will integrate these poses into the online evaluation service provided at visuallocalization.net as to provide better evaluations to the community. We also used our approach to create an extended version of the Aachen Day-Night dataset and showed that this dataset offers room for improvement. We will make the dataset available on the benchmark website. Furthermore, we will release the code for our approach as to allow other researchers to more easily build localization benchmarks.

One disadvantage of our approach is its rather slow run-time, taking about 10-20 seconds per iteration for a single image, where most of the time is spend for rendering and especially for the SfM inversion process. This is not an issue when creating reference poses for a benchmark, as these calculations only need to be done once and can be done offline. At the same time, our approach can be used as a post-processing step for any visual localization algorithm. An interesting research question is whether more efficient rendering techniques can be used to improve its run-time to a degree that enables online operation.

Acknowledgements. This work was supported by the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots), the Chalmers AI Research Centre (CHAIR) (VisLocLearn), the National Centre of Competence in Research (NCCR) Robotics, through the Swiss National Science Foundation and the SNSF-ERC Starting Grant. We thank Mihai Dusmanu and Martin Humenberger for contributing the D2-Net and R2D2 results, respectively.

Bibliography

[1] M. Achtelik, S. Weiss, M. Chli, and R. Siegwart. "Path Planning for Motion Dependent State Estimation on Micro Aerial Vehicles". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2013.

[2] D. Aiger, H. Kaplan, E. Kokiopoulou, M. Sharir, and B. Zeisl. "General Techniques for Approximate Incidences and Their Application to the Camera Posing Problem". In: *International Symposium on Computational Geometry (SoCG)*. 2019.

[3] C. Albl, Z. Kukelova, and T. Pajdla. "Rolling Shutter Absolute Pose Problem With Known Vertical Direction". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2016.

[4] P. F. Alcantarilla, K. Ni, L. M. Bergasa, and F. Dellaert. "Visibility learning in large-scale urban environment". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2011.

[5] H. Alismail, M. Kaess, B. Browning, and S. Lucey. "Direct Visual Odometry in Low Light Using Binary Descriptors". In: *IEEE Robot. Autom. Lett.* 2.2 (Apr. 2017), pp. 444–451.

[6] I. Alzugaray, L. Teixeira, and M. Chli. "Short-term UAV path-planning with monocular-inertial SLAM in the loop". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2017.

[7] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. "NetVLAD: CNN architecture for weakly supervised place recognition". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2016.

[8] R. Arandjelović and A. Zisserman. "Visual Vocabulary with a Semantic Twist". In: *Asian Conf. Comput. Vis. (ACCV)*. 2014.

[9] Z. Arican and P. Frossard. "OmniSIFT: Scale invariant features in omnidirectional images". In: *IEEE Int. Conf. Image Process. (ICIP)*. IEEE. 2010, pp. 3505–3508.

[10] M. Aubry, B. C. Russell, and J. Sivic. "Painting-to-3D model alignment via discriminative visual elements". In: *ACM Trans. Graph.* 33.2 (2014), p. 14.

[11] H. Badino, D. Huber, and T. Kanade. "Visual Topometric Localization". In: *IEEE Intell. Vehicles Symp.* 2011.

[12] R. Bajcsy. "Active perception". In: *Proc. IEEE* 76.8 (1988), pp. 966–1005.

[13] S. Baker and I. Matthews. "Lucas-Kanade 20 Years On: A Unifying Framework". In: *Int. J. Comput. Vis.* 56.3 (2004), pp. 221–255.

[14] V. Balntas, D. Frost, R. Kouskouridas, A. Barroso-Laguna, A. Talattof, H. Heijnen, and K. Mikolajczyk. *(SILDa): Scape Imperial Localisation Dataset*. 2019. URL: https://image-matching-workshop.github.io/challenge/.

[15] V. Balntas, S. Li, and V. Prisacariu. "RelocNet: Continuous Metric Learning Relocalisation using Neural Nets". In: *Eur. Conf. Comput. Vis. (ECCV)*. Sept. 2018.

[16] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. "Learning local feature descriptors with triplets and shallow convolutional neural networks". In: *British Mach. Vis. Conf. (BMVC)*. 2016.

[17] T. D. Barfoot. *State Estimation for Robotics*. 1st. New York, NY, USA: Cambridge University Press, 2017.

[18] T. D. Barfoot, C. H. Tong, and S. Sarkka. "Batch Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression". In: *Robotics: Science and Systems (RSS)*. 2014.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-Up Robust Features (SURF)". In: *Comput. Vis. Image. Und.* 110.3 (2008), pp. 346–359.

[20] A. Benbihi, M. Geist, and C. Pradalier. "ELF: Embedded Localisation of Features in Pre-Trained CNN". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[21] P. Bergmann, R. Wang, and D. Cremers. "Online Photometric Calibration of Auto Exposure Video for Realtime Visual Odometry and SLAM". In: *IEEE Robot. Autom. Lett.* 3.2 (Apr. 2018), pp. 627–634.

[22] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. "CodeSLAM - Learning a Compact, Optimisable Representation for Dense Visual SLAM". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[23] M. J. Box. "Bias in Nonlinear Estimation". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 33.2 (1971), pp. 171–201.

[24] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. "DSAC - Differentiable RANSAC for Camera Localization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[25] E. Brachmann and C. Rother. "Expert Sample Consensus Applied to Camera Re-Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[26] E. Brachmann and C. Rother. "Learning Less is More - 6D Camera Localization via 3D Surface Regression". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[27] E. Brachmann and C. Rother. "Neural- Guided RANSAC: Learning Where to Sample Model Hypotheses". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[28] E. Brachmann and C. Rother. "Visual Camera Re-Localization from RGB and RGB-D Images Using DSAC". In: *arXiv e-prints* (2020).

[29] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. "Geometry-Aware Learning of Maps for Camera Localization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[30] M. Brown, G. Hua, and S. Winder. "Discriminative Learning of Local Image Descriptors". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2011).

[31] I. Budvytis, M. Teichmann, T. Vojir, and R. Cipolla. "Large Scale Joint Semantic Re-Localisation and Scene Understanding via Globally Unique Instance Coordinate Regression". In: *British Mach. Vis. Conf. (BMVC)*. 2019.

[32] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós. "A comparison of SLAM algorithms based on a graph of relations". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. Oct. 2009, pp. 2089–2095. DOI: 10.1109/IROS.2009.5354691.

[33] W. Burgard, D. Fox, and S. Thrun. "Active Mobile Robot Localization". In: *Int. Joint Conf. Artificial Intell. (IJCAI)*. Aug. 1997.

[34] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. "The EuRoC micro aerial vehicle datasets". In: *Int. J. Robot. Research* 35.10 (2015), pp. 1157–1163. DOI: 10.1177/0278364915620033.

[35] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Trans. Robot.* 32.6 (2016), pp. 1309–1332.

[36] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. "BRIEF: Computing a Local Binary Descriptor Very Fast". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34.7 (2012), pp. 1281–1298.

[37] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler. "Hybrid Scene Compression for Visual Localization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[38] S. Cao and N. Snavely. "Graph-Based Discriminative Learning for Location Recognition". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2013.

[39] S. Cao and N. Snavely. "Minimal Scene Descriptions from Structure from Motion Models". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2014.

[40] M. F. Carfora. "Interpolation on spherical geodesic grids: A comparative study". In: *Journal of Computational and Applied Mathematics* (2007).

[41] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. "University of Michigan North Campus long-term vision and lidar dataset". In: *Int. J. Robot. Research* 35.9 (2016), pp. 1023–1035.

[42] L. Carlone and S. Karaman. "Attention and anticipation in fast visual-inertial navigation". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2017.

[43] H. Carrillo, I. Reid, and J. A. Castellanos. "On the comparison of uncertainty criteria for active SLAM". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2012, pp. 2080–2087. DOI: 10.1109/ICRA.2012.6224890.

[44] D. Caruso, J. Engel, and D. Cremers. "Large-scale direct SLAM for omnidirectional cameras". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2015, pp. 141–148.

[45] R. O. Castle, G. Klein, and D. W. Murray. "Video-rate Localization in Multiple Maps for Wearable Augmented Reality". In: *ISWC*. 2008.

[46] T. Cavallari, L. Bertinetto, J. Mukhoti, P. Torr, and S. Golodetz. "Let's Take This Online: Adapting Scene Coordinate Regression Network Predictions for Online RGB-D Camera Relocalisation". In: *3D Vision (3DV)*. 2019.

[47] T. Cavallari, S. Golodetz, N. Lord, J. Valentin, V. Prisacariu, L. Di Stefano, and P. H. S. Torr. "Real-Time RGB-D Camera Pose Estimation in Novel Scenes using a Relocalisation Cascade". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

[48] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. Di Stefano, and P. H. S. Torr. "On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[49] C. Chen, Q. Chen, J. Xu, and V. Koltun. "Learning to See in the Dark." In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018, pp. 3291–3300.

[50] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. "City-Scale Landmark Identification on Mobile Devices". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2011.

[51] J. Chen, S. J. Carey, and P. Dudek. "Scamp5d vision system and development framework". In: *Int. Conf. on Distributed Smart Cameras (ICDSC)*. 2018.

[52] J. Chen, T. Liu, and S. Shen. "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2016, pp. 1476–1483. DOI: 10.1109/ICRA.2016.7487283.

[53] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford. "Deep Learning Features at Scale for Visual Place Recognition". In: *IEEE Int. Conf. Robot. Autom. (ICRA)* (2017).

[54] W. Cheng, W. Lin, K. Chen, and X. Zhang. "Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[55] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. "Structure from motion causally integrated over time". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.4 (Apr. 2002), pp. 523–535.

[56] S. Choudhary and P. J. Narayanan. "Visibility Probability Structure from SfM Datasets and Applications". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2012.

[57] O. Chum and J. Matas. "Matching with PROSAC - progressive sample consensus". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. Vol. 1. 2005, 220–226 vol. 1.

[58] O. Chum and J. Matas. "Optimal Randomized RANSAC". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.8 (2008), pp. 1472–1482.

[59] 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. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[60] L. Clement and J. Kelly. "How to Train a CAT: Learning Canonical Appearance Transformations for Direct Visual Localization Under Illumination Change". In: *IEEE Robot. Autom. Lett.* 3.3 (2018), pp. 2447–2454.

[61] P. Corke, D. Strelow, and S. Singh. "Omnidirectional visual odometry for a planetary rover". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. Vol. 4. IEEE. 2004, pp. 4007–4012.

[62] G. Costante, C. Forster, J. A. Delmerico, P. Valigi, and D. Scaramuzza. "Perception-aware Path Planning". In: *arXiv e-prints* (2016).

[63] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. "Discrete-Continuous Optimization for Large-Scale Structure from Motion". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2011.

[64] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt. "BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration". In: *ACM Trans. Graph.* (2017).

[65] A. J. Davison and R. M. Murray. "Simultaneous Localization and Map-Building Using Active Vision". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.7 (2002).

[66] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. "MonoSLAM: Real-time single camera SLAM". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.6 (June 2007), pp. 1052–1067.

[67] P. E. Debevec and J. Malik. "Recovering high dynamic range radiance maps from photographs". In: *ACM SIGGRAPH*. ACM. 2008, p. 31.

[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2009.

[69] D. DeTone, T. Malisiewicz, and A. Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2018.

[70] D. Dey, K. S. Shankar, S. Zeng, R. Mehta, M. T. Agcayazi, C. Eriksen, S. Daftry, M. Hebert, and J. A. Bagnell. "Vision and learning for deliberative monocular cluttered flight". In: *Field and Service Robot.* 2015, pp. 391–409.

[71] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo. "CamNet: Coarse-to-Fine Retrieval for Camera Re-Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[72] J. Dong, M. Mukadam, F. Dellaert, and B. Boots. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." In: *Robotics: Science and Systems (RSS)*. 2016.

[73] M. Donoser and D. Schmalstieg. "Discriminative Feature-to-Point Matching in Image-Based Locallization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2014.

[74] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[75] R. C. DuToit, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis. "Consistent map-based 3D localization on mobile devices". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017.

[76] 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. Robot. Autom. (ICRA)*. 2015.

[77] P. Ebel, A. Mishchuk, K. M. Yi, P. Fua, and E. Trulls. "Beyond Cartesian Representations for Local Descriptors". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[78] B. Efron and D. V. Hinkley. "Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information". In: *Biometrika* 65.3 (1978), pp. 457–483.

[79] Y. B. Elisha and V. Indelman. "Active online visual-inertial navigation and sensor calibration via belief space planning and factor graph based incremental smoothing". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. Sept. 2017.

[80] J. Engel, J. Schöps, and D. Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2014.

[81] J. Engel, J. Sturm, and D. Cremers. "Semi-Dense Visual Odometry for a Monocular Camera". In: *Int. Conf. Comput. Vis. (ICCV)*. 2013.

[82] J. Engel, V. Koltun, and D. Cremers. "Direct Sparse Odometry". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.3 (Mar. 2018), pp. 611–625. DOI: 10.1109/TPAMI.2017.2658577.

[83] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza. "PAMPC: Perception-aware model predictive control for quadrotors". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018, pp. 1–8.

[84] S. Farboud-Sheshdeh, T. D. Barfoot, and R. H. Kwong. "Towards Estimating Bias in Stereo Visual Odometry". In: *2014 Canadian Conference on Computer and Robot Vision*. 2014, pp. 8–15.

[85] H. S. S. Feder, J. J. Leonard, and C. M. Smith. "Adaptive Mobile Robot Navigation and Mapping". In: *Int. J. Robot. Research* 18.7 (1999), pp. 650–558.

[86] M. A. Fischler and R. C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Commun. ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692.

[87] P. Florence, C. John, and R. Tedrake. "Integrated perception and control at high speed". In: *WAFR: Workshop on the Algorithmic Foundations of Robotics*. 2016.

[88] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry". In: *IEEE Trans. Robot.* 33.1 (2017), pp. 1–21. DOI: 10.1109/TRO.2016.2597321.

[89] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza. "Continuous On-Board Monocular-Vision–based Aerial Elevation Mapping for Quadrotor Landing". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2015, pp. 111–118. DOI: 10.1109/ICRA.2015.7138988.

[90] C. Forster, M. Pizzoli, and D. Scaramuzza. "Appearance-based Active, Monocular, Dense Depth Estimation for Micro Aerial Vehicles". In: *Robotics: Science and Systems (RSS)*. 2014. DOI: 10.15607/RSS.2014.X.029.

[91] C. Forster, M. Pizzoli, and D. Scaramuzza. "SVO: Fast Semi-Direct Monocular Visual Odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2014, pp. 15–22. DOI: 10.1109/ICRA.2014.6906584.

[92] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Trans. Robot.* 33.2 (2017), pp. 249–265. DOI: 10.1109/TRO.2016.2623335.

[93] F. Fraundorfer and D. Scaramuzza. "Visual Odometry. Part II: Matching, Robustness, Optimization, and Applications". In: *IEEE Robot. Autom. Mag.* (2012). DOI: 10.1109/MRA.2012.2182810.

[94] F. Fraundorfer, P. Tanskanen, and M. Pollefeys. "A Minimal Case Solution to the Calibrated Relative Pose Problem for the Case of Two Known Orientation Angles". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2010, pp. 269–282.

[95] K. M. Frey, T. J. Steiner, and J. P. How. *Towards Online Observability-Aware Trajectory Optimization for Landmark-based Estimators*. 2019. arXiv: 1908.03790.

[96] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart. "Robot Operating System (ROS): The Complete Reference (Volume 1)". In: ed. by A. Koubaa. Cham: Springer International Publishing, 2016. Chap. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. ISBN: 978-3-319-26054-9. DOI: 10.1007/978-3-319-26054-9_23. URL: http://dx.doi.org/10.1007/978-3-319-26054-9_23.

[97] S. Garg, N. Suenderhauf, and M. Milford. "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views". In: *Int. J. Robot. Research* 0.0 (2019).

[98] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2012.

[99] H. Germain, G. Bourmaud, and V. Lepetit. "Sparse-to-Dense Hypercolumn Matching for Long-Term Visual Localization". In: *3D Vision (3DV)*. 2019.

[100] C. Geyer and K. Daniilidis. "A unifying theory for central panoramic systems and practical implications". In: *Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2000, pp. 445–461.

[101] R. Gomez-Ojeda, F. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez. "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments". In: *IEEE Trans. Robot.* 35.3 (2019), pp. 734–746.

[102] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza. "Learning-Based Image Enhancement for Visual Odometry in Challenging HDR Environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 805–811. DOI: 10.1109/ICRA.2018.8462876.

[103] M. Greeff, T. D. Barfoot, and A. P. Schoellig. "A Perception-Aware Flatness-Based Model Predictive Controller for Fast Vision-Based Multirotor Flight". In: *IFAC World Congress*. 2020.

[104] W. N. Greene and N. Roy. "Metrically-Scaled Monocular SLAM using Learned Scale Factors". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020.

[105] P. Hansen, P. Corke, and W. Boles. "Wide-angle visual feature matching for outdoor localization". In: *Int. J. Robot. Research* (2009).

[106] R. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. "Review and analysis of solutions of the three point perspective pose estimation problem". In: *Int. J. Comput. Vis.* 13.3 (1994), pp. 331–356.

[107] C. Harris and M. Stephens. "A combined corner and edge detector". In: *Proc. Fourth Alvey Vision Conf.* Vol. 15. 1988, pp. 147–151. DOI: 10.5244/C.2.23.

[108] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd Edition. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511811685.

[109] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss. "Observability-Aware Trajectory Optimization for Self-Calibration With Application to UAVs". In: *IEEE Robot. Autom. Lett.* 2.3 (July 2017), pp. 1770–1777. DOI: 10.1109/LRA.2017.2647799.

[110] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler. "Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2019.

[111] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. "Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes". In: *Asian Conf. Comput. Vis. (ACCV)*. Ed. by K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu. 2012.

[112] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. "OctoMap: an efficient probabilistic 3D mapping framework based on octrees". In: *Auton. Robots* 34.3 (Apr. 2013), pp. 189–206. ISSN: 1573-7527. DOI: 10.1007/s10514-012-9321-0. URL: https://doi.org/10.1007/s10514-012-9321-0.

[113] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess. "Information Sparsification in Visual-Inertial Odometry". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018, pp. 1146–1153.

[114] G. Huang. "Visual-Inertial Navigation: A Concise Review". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2019.

[115] Z. Huang, Y. Xu, J. Shi, X. Zhou, H. Bao, and G. Zhang. "Prior Guided Dropout for Robust Visual Localization in Dynamic Environments". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[116] B. Ichter, B. L. and Edward Schmerling, and M. Pavone. "Robust Motion Planning via Perception-Aware Multiobjective Search on GPUs". In: *Proc. Int. Symp. Robot. Research (ISRR)*. Dec. 2017.

[117] D. Ilstrup and R. Manduchi. "One-shot optimal exposure control". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2010, pp. 200–213.

[118] V. Indelman, L. Carlone, and F. Dellaert. "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments". In: *Int. J. Robot. Research* 34.7 (2015), pp. 849–882.

[119] M. Irani and P. Anandan. "All About Direct Methods". In: *Proc. Workshop Vis. Algorithms: Theory Pract.* 1999, pp. 267–277.

[120] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. "From Structure-from-Motion Point Clouds to Fast Location Recognition". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2009.

[121] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza. "An information gain formulation for active volumetric 3D reconstruction". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2016. DOI: 10.1109/icra.2016.7487527.

[122] H. Jin, P. Favaro, and S. Soatto. "Real-time feature tracking and outlier rejection with changes in illumination". In: *Int. Conf. Comput. Vis. (ICCV)* (2001). DOI: 10.1109/ICCV.2001.937588.

[123] E. S. Jones and S. Soatto. "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach". In: *Int. J. Robot. Research* 30.4 (Apr. 2011).

[124] E. Jung, N. Yang, and D. Cremers. "Multi-Frame GAN: Image Enhancement for Stereo Visual Odometry in Low Light". In: *Conf. on Robotics Learning (CoRL)*. 2019.

[125] J. Kannala and S. S. Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28.8 (2006), pp. 1335–1340.

[126] S. Karaman and E. Frazzoli. "Sampling-based Algorithms for Optimal Motion Planning". In: *Int. J. Robot. Research* 30.7 (June 2011), pp. 846–894. ISSN: 0278-3649. DOI: 10.1177/0278364911406761. URL: http://dx.doi.org/10.1177/0278364911406761.

[127] A. Kasyanov, F. Engelmann, J. Stückler, and B. Leibe. "Keyframe-based visual-inertial online SLAM with relocalization". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2017.

[128] M. Kazhdan and H. Hoppe. "Screened Poisson Surface Reconstruction". In: *ACM Trans. Graph.* 32.3 (July 2013).

[129] J. Kelly and G. S. Sukhatme. "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration". In: *Int. J. Robot. Research* 30.1 (2011), pp. 56–79. DOI: 10.1177/0278364910382802.

[130] A. Kendall and R. Cipolla. "Geometric loss functions for camera pose regression with deep learning". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[131] A. Kendall, M. Grimes, and R. Cipolla. "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2015, pp. 2938–2946. DOI: 10.1109/ICCV.2015.336.

[132] C. Kerl, J. Sturm, and D. Cremers. "Robust Odometry Estimation for RGB-D Cameras". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2013.

[133] J. Kim, Y. Cho, and A. Kim. "Exposure Control Using Bayesian Optimization Based on Entropy Weighted Image Gradient". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 857–864.

[134] S. J. Kim, J.-M. Frahm, and M. Pollefeys. "Joint Feature Tracking and Radiometric Calibration from Auto-Exposure Video". In: *Int. Conf. Comput. Vis. (ICCV)*. 2007.

[135] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. "Robocup: The robot world cup initiative". In: *IAA*. ACM. 1997, pp. 340–347.

[136] G. Klein and D. Murray. "Improving the Agility of Keyframe-Based SLAM". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2008, pp. 802–815.

[137] G. Klein and D. Murray. "Parallel tracking and mapping for small AR workspaces". In: *IEEE ACM Int. Sym. Mixed and Augmented Reality (ISMAR)*. Nara, Japan, Nov. 2007, pp. 225–234.

[138] L. Kneip and P. Furgale. "OpenGV: A unified and generalized approach to real-time calibrated geometric vision". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2014, pp. 1–8.

[139] L. Kneip, D. Scaramuzza, and R. Siegwart. "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2011, pp. 2969–2976. DOI: 10.1109/CVPR.2011.5995464.

[140] Z. Kukelova, M. Bujnak, and T. Pajdla. "Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length". In: *Int. Conf. Comput. Vis. (ICCV)*. 2013.

[141] Z. Kukelova, M. Bujnak, and T. Pajdla. "Closed-Form Solutions to Minimal Absolute Pose Problems with Known Vertical Direction". In: *Asian Conf. Comput. Vis. (ACCV)*. 2010.

[142] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. "On measuring the accuracy of SLAM algorithms". In: *Auton. Robots* 27.4 (Sept. 2009), p. 387. ISSN: 1573-7527. DOI: 10.1007/s10514-009-9155-6. URL: https://doi.org/10.1007/s10514-009-9155-6.

[143] J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza. "Redesigning SLAM for Arbitrary Multi-Camera Systems". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020.

[144] B. Landry, R. Deits, P. R. Florence, and R. Tedrake. "Aggressive quadrotor flight through cluttered environments using mixed integer programming". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2016. DOI: 10.1109/icra.2016.7487282.

[145] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler, and F. Kahl. "Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[146] V. Larsson, Z. Kukelova, and Y. Zheng. "Making Minimal Solvers for Absolute Pose Estimation Compact and Robust". In: *Int. Conf. Comput. Vis. (ICCV)*. 2017.

[147] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. "Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network". In: *ICCV Workshops*. 2017.

[148] K. Lebeda, J. E. S. Matas, and O. Chum. "Fixing the Locally Optimized RANSAC". In: *British Mach. Vis. Conf. (BMVC)*. 2012.

[149] K. Lee, J. Gibson, and E. A. Theodorou. "Aggressive Perception-Aware Navigation Using Deep Optical Flow Dynamics and PixelMPC". In: *IEEE Robot. Autom. Lett.* (2020).

[150] S. Leutenegger, M. Chli, and R. Siegwart. "BRISK: Binary Robust invariant scalable keypoints". In: *Int. Conf. Comput. Vis. (ICCV)*. 2011, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542.

[151] M. Lhuillier. "Automatic structure and motion using a catadioptric camera". In: *Proceedings of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*. 2005.

[152] S. Li, A. Handa, Y. Zhang, and A. Calway. "HDRFusion: HDR SLAM using a low-cost auto-exposure RGB-D sensor". In: *arXiv e-prints* (2016).

[153] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. "Worldwide Pose Estimation Using 3D Point Clouds". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2012.

[154] Y. Li, N. Snavely, and D. P. Huttenlocher. "Location Recognition using Prioritized Feature Matching". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2010.

[155] 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. Comput. Vis. Pattern Recog. (CVPR)*. 2012.

[156] L. Liu, H. Li, and Y. Dai. "Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map". In: *Int. Conf. Comput. Vis. (ICCV)*. 2017.

[157] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar. "Planning Dynamically Feasible Trajectories for Quadrotors Using Safe Flight Corridors in 3-D Complex Environments". In: *IEEE Robot. Autom. Lett.* 2.3 (July 2017), pp. 1688–1695. DOI: 10.1109/LRA.2017.2663526.

[158] M. Lourenço, J. P. Barreto, and F. Vasconcelos. "SRD-SIFT: Keypoint detection and matching in images with radial distortion". In: *IEEE Trans. Robot.* 28.3 (2012), pp. 752–760.

[159] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vis.* 60.2 (Nov. 2004), pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

[160] H. Lu, H. Zhang, S. Yang, and Z. Zheng. "Camera parameters auto-adjusting technique for robust robot vision". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010.

[161] T. Lupton and S. Sukkarieh. "Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions". In: *IEEE Trans. Robot.* 28.1 (Feb. 2012), pp. 61–76.

[162] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart. "A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2013.

[163] 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 (RSS)*. 2015.

[164] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. "1 Year, 1000 km: The Oxford RobotCar dataset." In: *Int. J. Robot. Research* 36.1 (2017), pp. 3–15.

[165] M. Maimone, Y. Cheng, and L. Matthies. "Two years of Visual Odometry on the Mars Exploration Rovers". In: *J. Field Robot.* 24.3 (2007), pp. 169–186. DOI: 10.1002/rob.20184.

[166] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte. "An experiment in integrated exploration". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. Sept. 2002.

[167] D. Massiceti, A. Krull, E. Brachmann, C. Rother, and P. H. Torr. "Random Forests versus Neural Networks - What's Best for Camera Relocalization?" In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017.

[168] H. Matsuki, L. von Stumberg, V. Usenko, J. Stückler, and D. Cremers. "Omnidirectional DSO: Direct Sparse Odometry With Fisheye Cameras". In: *IEEE Robot. Autom. Lett.* 3.4 (2018), pp. 3693–3700.

[169] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2014, pp. 3242–3249. DOI: 10.1109/ICRA.2014.6907325.

[170] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova. "Computer vision on Mars". In: *Int. J. Comput. Vis.* 75.1 (2007), pp. 67–92.

[171] I. Mehta, M. Tang, and T. D. Barfoot. "Gradient-Based Auto-Exposure Control Applied to a Self-Driving Car". In: *Conf. Comput. Robot Vis. (CRV)*. 2020.

[172] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. "Image-based Localization using Hourglass Networks". In: *ICCV Workshops*. 2017.

[173] D. Mellinger and V. Kumar. "Minimum snap trajectory generation and control for quadrotors". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2011, pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.

[174] L. Meng, J. Chen, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva. "Backtracking regression forests for accurate camera relocalization". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2017.

[175] L. Meng, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva. "Exploiting Points and Lines in Regression Forests for RGB-D Camera Relocalization". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018.

[176] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. "Scalable 6-DOF Localization on Mobile Devices". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2014.

[177] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020. arXiv: 2003.08934 [cs.CV].

[178] M. J. Milford and G. F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2012.

[179] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. "Working hard to know your neighbor's margins: Local descriptor learning loss". In: *Advances in Neural Information Processing Systems*. 2017.

[180] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico, et al. "Fast, autonomous flight in GPS-denied and cluttered environments". In: *J. Field Robot.* 35.1 (2018), pp. 101–120.

[181] C. Mostegel, A. Wendel, and H. Bischof. "Active monocular localization: Towards autonomous monocular exploration for multirotor MAVs". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2014, pp. 3848–3855. DOI: 10.1109/ICRA.2014.6907417.

[182] A. I. Mourikis and S. I. Roumeliotis. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. Apr. 2007, pp. 3565–3572.

[183] M. W. Mueller, M. Hehn, and R. D'Andrea. "A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2013, pp. 3480–3486.

[184] M. Muglikar, Z. Zhang, and D. Scaramuzza. "Voxel Map for Visual SLAM". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020.

[185] R. Mur-Artal and J. D. Tardós. "Visual-Inertial Monocular SLAM With Map Reuse". In: *IEEE Robot. Autom. Lett.* 2.2 (2017), pp. 796–803.

[186] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. "ORB-SLAM: a Versatile and Accurate Monocular SLAM System". In: *IEEE Trans. Robot.* 31.5 (2015), pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671.

[187] R. Mur-Artal and J. D. Tardós. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Trans. Robot.* 33.5 (Oct. 2017), pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103.

[188] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman. "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness". In: *IEEE Am. Control Conf. (ACC)*. 2019.

[189] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard. "Semantics-aware Visual Localization under Challenging Perceptual Conditions". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017.

[190] A. J. Neves, B. Cunha, A. J. Pinho, and I. Pinheiro. "Autonomous configuration of parameters in robotic digital cameras". In: *Pattern Recognition and Image Analysis*. Springer, 2009, pp. 80–87.

[191] R. A. Newcombe, S. Izadi, O. Hilliges, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-Time Dense Surface Mapping and Tracking". In: *IEEE ACM Int. Sym. Mixed and Augmented Reality (ISMAR)*. 2011.

[192] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. "DTAM: Dense Tracking and Mapping in Real-Time". In: *Int. Conf. Comput. Vis. (ICCV)*. 2011, pp. 2320–2327. DOI: 10.1109/ICCV.2011.6126513.

[193] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. "Real-time 3D reconstruction at scale using voxel hashing". In: *ACM Trans. Graph.* 32.6 (Nov. 2013), p. 169.

[194] D. Nister, O. Naroditsky, and J. Bergen. "Visual odometry". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2004, pp. 652–659.

[195] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. "Large-Scale Image Retrieval with Attentive Deep Local Features". In: *Int. Conf. Comput. Vis. (ICCV)*. 2017, pp. 3476–3485.

[196] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran. "Continuous-Time Trajectory Optimization for Online UAV Replanning". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2016.

[197] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart. "Signed distance fields: A natural representation for both mapping and planning". In: *RSS Workshop: Geometry and Beyond - Representations, Physics, and Scene Understanding for Robotics*. 2016.

[198] 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. Intell. Robot. Syst. (IROS)*. 2017.

[199] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto. "Safe Local Exploration for Replanning in Cluttered Unknown Environments for Micro-Aerial Vehicles". In: *IEEE Robot. Autom. Lett.* (2018).

[200] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. "LF-Net: Learning Local Features from Images". In: *Conf. Neural Inf. Process. Syst. (NIPS)*. 2018.

[201] A. Pagani and D. Stricker. "Structure from Motion using full spherical panoramic cameras". In: *Int. Conf. Comput. Vis. Workshops (ICCVW)*. IEEE. 2011, pp. 375–382.

[202] C. Papachristos, S. Khattak, and K. Alexis. "Uncertainty-aware receding horizon exploration and mapping using aerial robots". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017, pp. 4568–4575. DOI: 10.1109/ICRA.2017.7989531.

[203] F. Pittaluga, S. J. Koppal, S. B. Kang, and S. N. Sinha. "Revealing Scenes by Inverting Structure From Motion Reconstructions". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. June 2019.

[204] M. Pizzoli, C. Forster, and D. Scaramuzza. "REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2014, pp. 2609–2616. DOI: 10.1109/ICRA.2014.6907233.

[205] *Precomputed Visibility Volume*. https://docs.unrealengine.com/en-US.

[206] J. A. Preiss, K. Hausman, G. S. Sukhatme, and S. Weiss. "Trajectory Optimization for Self-Calibration and Navigation". In: *Robotics: Science and Systems (RSS)* (2017).

[207] L. Puig and J. Guerrero. "Scale space for central catadioptric systems: Towards a generic camera feature extractor". In: *Int. Conf. Comput. Vis. (ICCV)*. Nov. 2011, pp. 1599–1606.

[208] F. Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006. DOI: 10.1137/1.9780898719109.

[209] T. Qin, P. Li, and S. Shen. "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Trans. Robot.* 34.4 (2018), pp. 1004–1020. DOI: 10.1109/TRO.2018.2853729.

[210] W. Qiu, F. Zhong, Y. Zhang, S. Qiao, Z. Xiao, T. S. Kim, and Y. Wang. "UnrealCV: Virtual Worlds for Computer Vision". In: *Proc. ACM Int. Conf. Mult.* 2017, pp. 1221–1224. DOI: 10.1145/3123266.3129396.

[211] F. Radenović, G. Tolias, and O. Chum. "Fine-Tuning CNN Image Retrieval with No Human Annotation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41.7 (2019), pp. 1655–1668.

[212] N. Radwan, A. Valada, and W. Burgard. "VLocNet++: Deep Multitask Learning For Semantic Visual Localization And Odometry". In: *IEEE Robot. Autom. Lett.* 3.4 (2018), pp. 4407–4414.

[213] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. "USAC: A Universal Framework for Random Sample Consensus". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (2013), pp. 2022–2038.

[214] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[215] J. Revaud, C. D. Souza, M. Humenberger, and P. Weinzaepfel. "R2D2: Reliable and Repeatable Detector and Descriptor". In: *Conf. Neural Inf. Process. Syst. (NIPS)*. 2019, pp. 12405–12415.

[216] C. Richter, A. Bry, and N. Roy. "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments". In: *Proc. Int. Symp. Robot. Research (ISRR)*. 2013, pp. 1–8.

[217] A. Rituerto, L. Puig, and J. Guerrero. "Comparison of omnidirectional and conventional monocular systems for visual SLAM". In: *10th OMNIVIS with RSS* (2010).

[218] A. Rituerto, L. Puig, and J. J. Guerrero. "Visual SLAM with an omnidirectional camera". In: *IEEE Int. Conf. Pattern Recog. (ICPR)*. 2010, pp. 348–351.

[219] D. Robertson and R. Cipolla. "An image-based system for urban navigation". In: 2004.

[220] S. M. Robeson. "Spherical methods for spatial interpolation: Review and evaluation". In: *Cartography and Geographic Information Systems* (1997).

[221] Z. Rong and N. Michael. "Detection and prediction of near-term state estimation degradation via online nonlinear observability analysis". In: *IEEE Int. Symp. Safety, Security, and Rescue Robot. (SSRR)*. Oct. 2016, pp. 28–33. DOI: 10.1109/SSRR.2016.7784273.

[222] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2015.

[223] E. Rosten and T. Drummond. "Machine learning for high-speed corner detection". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2006, pp. 430–443. DOI: 10.1007/11744023_34.

[224] E. Rosten, R. Porter, and T. Drummond. "Faster and Better: A Machine Learning Approach to Corner Detection". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 32.1 (Jan. 2010), pp. 105–119. DOI: 10.1109/TPAMI.2008.275.

[225] N. Roy and S. Thrun. "Coastal navigation with mobile robots". In: *Conf. Neural Inf. Process. Syst. (NIPS)*. 2000.

[226] N. Roy, W. Burgard, D. Fox, and S. Thrun. "Coastal navigation-mobile robot navigation with uncertainty in dynamic environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 1999.

[227] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *Int. Conf. Comput. Vis. (ICCV)*. 2011.

[228] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An Efficient Alternative to SIFT or SURF". In: *Int. Conf. Comput. Vis. (ICCV)*. 2011, pp. 2564–2571.

[229] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan. "Feature-rich path planning for robust navigation of MAVs with Mono-SLAM". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. May 2014, pp. 3870–3875. DOI: 10.1109/ICRA.2014.6907420.

[230] S. Saha, G. Varma, and C. V. Jawahar. "Improved Visual Relocalization by Discovering Anchor Points". In: *British Mach. Vis. Conf. (BMVC)*. 2018.

[231] M. Salas, Y. Latif, I. D. Reid, and J. Montiel. "Trajectory Alignment and Evaluation in SLAM: Horn's Method vs Alignment on the Manifold". In: Robotics: Science and Systems Workshop. 2015.

[232] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. "From Coarse to Fine: Robust Hierarchical Localization at Large Scale". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[233] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. "SuperGlue: Learning Feature Matching with Graph Neural Networks". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020.

[234] T. Sattler, B. Leibe, and L. Kobbelt. "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.9 (2017), pp. 1744–1756.

[235] T. Sattler, B. Leibe, and L. Kobbelt. "Fast Image-Based Localization using Direct 2D-to-3D Matching". In: *Int. Conf. Comput. Vis. (ICCV)*. 2011.

[236] T. Sattler et al. *RansacLib - A Template-based *SAC Implementation*. 2019. URL: https://github.com/tsattler/RansacLib.

[237] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018, pp. 8601–8610. DOI: 10.1109/CVPR.2018.00897.

[238] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. "Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?" In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[239] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. "Image Retrieval for Image-Based Localization Revisited". In: *British Mach. Vis. Conf. (BMVC)*. 2012.

[240] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. "Understanding the Limitations of CNN-Based Absolute Camera Pose Regression". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[241] D. Scaramuzza. "1-Point-RANSAC Structure from Motion for Vehicle-Mounted Cameras by Exploiting Non-Holonomic Constraints". In: *Int. J. Comput. Vis.* 96.1 (2011).

[242] D. Scaramuzza and F. Fraundorfer. "Visual Odometry [Tutorial]. Part I: The First 30 Years and Fundamentals". In: *IEEE Robot. Autom. Mag.* 18.4 (Dec. 2011), pp. 80–92. DOI: 10.1109/MRA.2011.943233.

[243] D. Scaramuzza. "1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints". In: *Int. J. Comput. Vis.* 95.1 (2011), pp. 74–85.

[244] D. Scaramuzza. "Performance evaluation of 1-point-RANSAC visual odometry". In: *J. Field Robot.* 28.5 (2011), pp. 792–811.

[245] D. Scaramuzza, A. Martinelli, and R. Siegwart. "A flexible technique for accurate omnidirectional camera calibration and structure from motion". In: *Int. Conf. Comput. Vis. Syst. (ICVS)*. IEEE. 2006, pp. 45–45.

[246] D. Scaramuzza and R. Siegwart. "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles". In: *IEEE Trans. Robot.* 24.5 (2008), pp. 1015–1026.

[247] D. Scaramuzza and Z. Zhang. "Aerial Robots, Visual-Inertial Odometry of". In: *Encyclopedia of Robotics*. Springer Berlin Heidelberg, 2020, pp. 1–9. DOI: 10.1007/978-3-642-41610-1_71-1.

[248] 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". In: *IEEE Robot. Autom. Lett.* 3.3 (2018), pp. 1418–1425.

[249] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. "Semantic Visual Localization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[250] J. L. Schönberger and J.-M. Frahm. "Structure-from-Motion Revisited". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2016.

[251] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2016.

[252] Z. Seymour, K. Sikka, H.-P. Chiu, S. Samarasekera, and R. Kumar. "Semantically-Aware Attentive Neural Embeddings for Image-based Visual Localization". In: *British Mach. Vis. Conf. (BMVC)*. 2019.

[253] Q. Shan, C. Wu, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. "Accurate Geo-registration by Ground-to-Aerial Image Matching". In: *3D Vision (3DV)*. 2014.

[254] T. Shi, S. Shen, X. Gao, and L. Zhu. "Visual Localization Using Sparse Semantic 3D Map". In: *IEEE Int. Conf. Image Process. (ICIP)*. 2019.

[255] I. Shim, T. Oh, J. Lee, J. Choi, D. Choi, and I. S. Kweon. "Gradient-Based Camera Exposure Control for Outdoor Mobile Platforms". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.6 (2019), pp. 1569–1583.

[256] I. Shim, J.-Y. Lee, and I. S. Kweon. "Auto-adjusting camera exposure for outdoor robotics using gradient information". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2014.

[257] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon. "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (2013), pp. 2930–2937.

[258] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt. "SIFT-Realistic Rendering". In: *3D Vision (3DV)*. 2013.

[259] G. Sibley, L. Matthies, and G. Sukhatme. "Sliding window filter with application to planetary landing". In: *J. Field Robot.* 27.5 (2010), pp. 587–608.

[260] 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. Comput. Vis. (ICCV)*. 2015.

[261] K. Simonyan, A. Vedaldi, and A. Zisserman. "Learning Local Feature Descriptors Using Convex Optimisation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 36.8 (2014), pp. 1573–1585.

[262] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. "DeepVoxels: Learning Persistent 3D Feature Embeddings". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[263] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations". In: *Conf. Neural Inf. Process. Syst. (NIPS)*. 2019.

[264] N. Snavely, S. Seitz, and R. Szeliski. "Modeling the World from Internet Photo Collections". In: *Int. J. Comput. Vis.* 80.2 (2008), pp. 189–210.

[265] I. Spasojevic, V. Murali, and S. Karaman. "Perception-aware time optimal path parameterization for quadrotors". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020.

[266] E. Stenborg, C. Toft, and L. Hammarstrand. "Long-Term Visual Localization Using Semantically Segmented Images". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018.

[267] Stephen Se, D. Lowe, and J. Little. "Global localization using distinctive visual features". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2002.

[268] H. Strasdat, J. Montiel, and A. Davison. "Real-time Monocular SLAM: Why Filter?" In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010.

[269] H. Strasdat. "Local accuracy and global consistency for efficient SLAM". PhD thesis. Imperial College London, UK, 2012.

[270] B. Streckel and R. Koch. "Lens model selection for visual tracking". In: *Pattern Recognition*. Springer, 2005, pp. 41–48.

[271] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. Oct. 2012.

[272] I. A. Şucan, M. Moll, and L. E. Kavraki. "The Open Motion Planning Library". In: *IEEE Robot. Autom. Mag.* (2012).

[273] X. Sun, Y. Xie, P. Luo, and L. Wang. "A Dataset for Benchmarking Image-Based Localization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[274] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free". In: *Robotics: Science and Systems (RSS)*. 2015.

[275] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. "City-Scale Localization for Cameras with Known Vertical Direction". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.7 (2017), pp. 1455–1461.

[276] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2018.

[277] H. Taira, I. Rocco, J. Sedlar, M. Okutomi, J. Sivic, T. Pajdla, T. Sattler, and A. Torii. "Is This the Right Place? Geometric-Semantic Pose Verification for Indoor Visual Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[278] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. "Monocular visual odometry in urban environments using an omnidirectional camera". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2008, pp. 2531–2538.

[279] J. J. Tarrio and S. Pedre. "Realtime Edge-Based Visual Odometry for a Monocular Camera". In: *Int. Conf. Comput. Vis. (ICCV)*. Dec. 2015, pp. 702–710.

[280] K. Tateno, F. Tombari, I. Laina, and N. Navab. "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. July 2017.

[281] Y. Tian, B. Fan, and F. Wu. "L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2017.

[282] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. "SOSNet: Second Order Similarity Regularization for Local Descriptor Learning". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2019.

[283] C. Toft, C. Olsson, and F. Kahl. "Long-term 3D Localization and Pose from Semantic Labellings". In: *ICCV Workshops*. 2017.

[284] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. "Semantic Match Consistency for Long-Term Visual Localization". In: *Eur. Conf. Comput. Vis. (ECCV)*. 2018.

[285] C. Tomasi and T. Kanade. "Shape and Motion from Image Streams: a Factorization Method". In: *Int. J. Comput. Vis.* 7597 (1992), pp. 137–154.

[286] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. "Visual Place Recognition with Repetitive Structures". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2015).

[287] A. Torii, J. Sivic, and T. Pajdla. "Visual localization by linear combination of image descriptors". In: *Proceedings of the 2nd IEEE Workshop on Mobile Vision, with ICCV*. 2011.

[288] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. "24/7 Place Recognition by View Synthesis". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.2 (Feb. 2018), pp. 257–271. DOI: 10.1109/TPAMI.2017.2667665.

[289] J. Torres and J. M. Menéndez. "Optimal camera exposure for video surveillance systems by predictive control of shutter speed, aperture, and gain". In: *IS&T/SPIE Electronic Imaging*. 2015.

[290] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. "Bundle Adjustment – A Modern Synthesis". In: *Vision Algorithms: Theory and Practice*. Ed. by W. Triggs, A. Zisserman, and R. Szeliski. Vol. 1883. LNCS. Springer Verlag, 2000, pp. 298–372.

[291] S. Ullman. *The Interpretation of Visual Motion*. MIT Press: Cambridge, MA, 1979.

[292] S. Umeyama. "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 13.4 (1991).

[293] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers. "Visual-Inertial Mapping With Non-Linear Factor Recovery". In: *IEEE Robot. Autom. Lett.* 5.2 (2020), pp. 422–429.

[294] A. Valada, N. Radwan, and W. Burgard. "Deep Auxiliary Learning For Visual Localization And Odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018.

[295] J. Valentin, A. Dai, M. Niessner, P. Kohli, P. Torr, S. Izadi, and C. Keskin. "Learning to Navigate the Energy Landscape". In: *3D Vision (3DV)*. 2016.

[296] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr. "Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2015.

[297] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg. "Global Localization from Monocular SLAM on a Mobile Phone". In: *IEEE Transactions on Visualization and Computer Graphics* 20.4 (2014), pp. 531–539.

[298] C. T. Viktor Larsson Johan Fredriksson and F. Kahl. "Outlier Rejection for Absolute Pose Estimation with Known Orientation". In: *British Mach. Vis. Conf. (BMVC)*. 2016.

[299] G. Vogiatzis and C. Hernández. "Video-based, Real-Time Multi View Stereo". In: *Image Vis. Comput.* 29.7 (2011), pp. 434–441. DOI: 10.1016/j.imavis.2011.01.006.

[300] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. "Image-Based Localization Using LSTMs for Structured Feature Correlation". In: *Int. Conf. Comput. Vis. (ICCV)*. 2017.

[301] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. "The ApolloScape Open Dataset for Autonomous Driving and its Application". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

[302] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely. "Learning Feature Descriptors using Camera Pose Supervision". In: *arXiv e-prints* (2020).

[303] M. Watterson, S. Liu, K. Sun, T. Smith, and V. Kumar. "Trajectory Optimization On Manifolds with Applications to SO(3) and R3XS2". In: *Robotics: Science and Systems (RSS)*. June 2018.

[304] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. "ElasticFusion: Real-time dense SLAM and light source estimation". In: *Int. J. Robot. Research* 35.14 (2016), pp. 1697–1716.

[305] B. Williams, G. Klein, and I. Reid. "Real-Time SLAM Relocalisation". In: *Int. Conf. Comput. Vis. (ICCV)*. 2007.

[306] C. Won, J. Ryu, and J. Lim. "OmniMVS: End-to-End Learning for Omnidirectional Stereo Matching". In: *Int. Conf. Comput. Vis. (ICCV)*. Oct. 2019.

[307] C. Won, H. Seok, Z. Cui, M. Pollefeys, and J. Lim. "OmniSLAM: Omnidirectional Localization and Dense Mapping for Wide-baseline Multi-camera Systems". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020.

[308] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, and H. Zha. "Local Supports Global: Deep Camera Relocalization With Sequence Enhancement". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[309] L. Yang, Z. Bai, C. Tang, H. Li, Y. Furukawa, and P. Tan. "SANet: Scene Agnostic Network for Camera Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[310] N. Yang, R. Wang, J. Stueckler, and D. Cremers. "Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry". In: *Eur. Conf. Comput. Vis. (ECCV)*. Sept. 2018.

[311] T.-Y. Yang, D.-K. Nguyen, H. Heijnen, and V. Balntas. "UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description without Local Correspondence Supervision". In: *arXiv e-prints* (2020).

[312] X. Yu, S. Chaturvedi, C. Feng, Y. Taguchi, T. Lee, C. Fernandes, and S. Ramalingam. "VLASE: Vehicle Localization by Aggregating Semantic Edges". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018.

[313] B. Zeisl, T. Sattler, and M. Pollefeys. "Camera Pose Voting for Large-Scale Image-Based Localization". In: *Int. Conf. Comput. Vis. (ICCV)*. 2015.

[314] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao. "Learning Two-View Correspondences and Geometry Using Order-Aware Network". In: *Int. Conf. Comput. Vis. (ICCV)*. 2019.

[315] W. Zhang and J. Kosecka. "Image based Localization in Urban Environments". In: *International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*. 2006.

[316] Z. Zhang, C. Forster, and D. Scaramuzza. "Active exposure control for robust visual odometry in HDR environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017, pp. 3894–3901. DOI: 10.1109/ICRA.2017.7989449.

[317] Z. Zhang, G. Gallego, and D. Scaramuzza. "On the Comparison of Gauge Freedom Handling in Optimization-Based Visual-Inertial State Estimation". In: *IEEE Robot. Autom. Lett.* 3.3 (July 2018), pp. 2710–2717. DOI: 10.1109/lra.2018.2833152.

[318] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. "Benefit of Large Field-of-View Cameras for Visual Odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2016. DOI: 10.1109/ICRA.2016.7487210.

[319] Z. Zhang, T. Sattler, and D. Scaramuzza. "Reference Pose Generation for Visual Localization via Learned Features and View Synthesis". In: *Under review in Int. J. Comput. Vis.* (2020). URL: https://arxiv.org/abs/2005.05179.

[320] Z. Zhang and D. Scaramuzza. "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018. DOI: 10.1109/IROS.2018.8593941.

[321] Z. Zhang and D. Scaramuzza. "Beyond Point Clouds: Fisher Information Field for Active Visual Localization". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2019, pp. 5986–5992. DOI: 10.1109/ICRA.2019.8793680.

[322] Z. Zhang and D. Scaramuzza. "Fisher Information Field: an Efficient and Differentiable Map for Perception-aware Planning". In: *Under review in IEEE Trans. Robot.* (2020). URL: https://arxiv.org/abs/2008.03324.

[323] Z. Zhang and D. Scaramuzza. "Perception-aware Receding Horizon Navigation for MAVs". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 2534–2541. DOI: 10.1109/ICRA.2018.8461133.

[324] Z. Zhang and D. Scaramuzza. "Rethinking Trajectory Evaluation for SLAM: a Probabilistic, Continuous-Time Approach". In: *ICRA19 Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for VR/AR*. 2019.

[325] E. Zheng and C. Wu. "Structure From Motion Using Structure-Less Resection". In: *Int. Conf. Comput. Vis. (ICCV)*. 2015.

[326] S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison. "SceneCode: Monocular Dense Semantic Reconstruction Using Learned Encoded Scene Representations". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[327] L. Zhou, Z. Luo, T. Shen, J. Zhang, M. Zhen, Y. Yao, T. Fang, and L. Quan. "KFNet: Learning Temporal Camera Relocalization using Kalman Filtering". In: *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*. 2020.

[328] Q. Zhou, T. Sattler, M. Pollefeys, and L. Leal-Taixe. "To Learn or Not to Learn: Visual Localization from Essential Matrices". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2019.

[329] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa. "CHOMP: Covariant Hamiltonian optimization for motion planning". In: *Int. J. Robot. Research* 32 (2013).

Zichao Zhang

Date of Birth: 14.05.1989

PhD Student in Robotics
Robotics and Perception Group
University of Zurich
Andreasstrasse 15, 8050 Zurich
Email: zzhang@ifi.uzh.ch zhangzichao17@gmail.com

EDUCATION

09/2015 - 09/2020	**PhD Program in Computer Science** Robotics and Perception Group, Department of Informatics/Institute of Neuroinformatics University of Zurich/ETH Advisor: Prof. Dr. Davide Scaramuzza
08/2019 - 10/2019	**Visiting Researcher** Chalmers University of Technology Topic: Learning-based view synthesis for visual localization Advisor: Prof. Dr. Torsten Sattler
09/2015 - 09/2016	**M.S. in Multi-Mode Cognitive System** University of Zurich Overall grade: 5.7 (*summa cum laude*)
09/2014 - 09/2015	**Visiting Researcher** Robotics and Perception Group, Department of Informatics/Institute of Neuroinformatics University of Zurich/ETH Advisor: Prof. Dr. Davide Scaramuzza
09/2011 - 09/2014	**Joint Master/PhD Program in Precision Instruments and Mechanology** School of Instrumental Science and Opto-electronics Engineering, Beihang University Advisor: Prof. Dr. Jianhui Zhao GPA: 3.80/4.00
09/2007 - 07/2011	**B.S. in Detection, Guidance and Control** School of Instrumental Science and Opto-electronics Engineering, Beihang University Overall GPA: 3.73/4.00 Major GPA: 3.79/4.00 Rank: 3/43

PROJECTS

11/2017 - 06/2019	**Low-latency Tracking and Robust Visual-Inertial SLAM Techniques for AR/VR (Leader)** *Industrial project*
09/2015 - 09/2016	**Fast Lightweight Autonomy Program (Member)** *Supported by DARPA*
01/2013 - 09/2014	**High-precision Autonomous Navigation Technology for Deep Space Explorer (Member)** *Key project supported by Chinese National Science Foundation*
01/2012 - 01/2013	**Study on the Mechanism, Model and Control Scheme of Controllable Pendulum (Member)** *Supported by Chinese Aeronautical Foundation*

RESEARCH INTERESTS

Computer Vision, Visual-Inertial SLAM, Active Vision, Mobile Robots Navigation

SKILLS

Language: Chinese (native), English (proficient)

Programming: C++, Python, MATLAB; ROS, OpenCV, PyTorch; Ceres, GTSAM

2020/10/15

Software and open source code:

1. Trajectory Evaluation Toolbox: https://github.com/uzh-rpg/rpg_trajectory_evaluation
2. Fisher Information Field: https://github.com/uzh-rpg/rpg_information_field
3. Visual-inertial Covariance Transformation: https://github.com/uzh-rpg/rpg_vi_cov_transformation
4. Open Source Interface for SVO 2.0: https://github.com/uzh-rpg/rpg_svo_example
5. Omnidirectional Camera Model: https://github.com/zhangzichao/omni_cam

TEACHING

Teaching Assistant

09/2015 - 01/2016	TA for course Fundamentals of Image Processing and Computer Vision, University of Zurich
03/2013 - 06/2013	TA for course Analog Circuit, Beihang University

Student Supervision

02/2019 – 05/2019	Semester Project: Improve Inertial Odometry using Machine Learning
10/2018 – 03/2019	Internship: Integrating Loop Closure in Visual-Inertial Odometry
10/2018 – 04/2019	Lab Project: Robust and Adaptive Multi-Camera Visual Odometry
04/2018 – 08/2018	Semester Project: Smart Feature Selection for Visual Odometry
11/2017 – 04/2018	Internship: Fast Visual-Inertial Odometry
09/2017 – 12/2017	Semester Project: Online Place Recognition for Visual Odometry
03/2017 - 08/2017	Bachelor Thesis: Learning Motion from Blur

ACADEMIC SERVICE AND TALKS

Conference Reviewers

ICRA 2017-2020, IROS 2016-2020, ECCV 2016, ICCV 2017/2019, CVPR 2017/2019, RSS 2020

Journal Reviewers

IEEE Transaction on Robotics, IEEE Robotics and Automation Letters, Journal of Field Robotics

Talks at International Workshops

ICRA 2017 Workshop on Reproducible Research in Robotics: Current Status and Road Ahead
ICRA 2018 Workshop on Informative Path Planning and Adaptive Sampling
ICRA 2019 Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR

PUBLICATIONS

Preprints

1. **Zichao Zhang**, Torsten Sattler, Davide Scaramuzza. Reference Pose Generation for Visual Localization via Learned Features and View Synthesis, arXiv, 2020.
2. **Zichao Zhang**, Davide Scaramuzza. Fisher Information Field: an Efficient and Differentiable Map for Perception-aware Planning, arXiv, 2020.

Journal Papers

1. Christian Forster, **Zichao Zhang**, Michael Gassner, Manuel Werlberger, Davide Scaramuzza, SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems, *IEEE Transactions on Robotics (TRO),* 2017.
2. **Zichao Zhang**, Guillermo Gallego, Davide Scaramuzza, On the Comparison of Gauge Freedom Handling in Optimization-based Visual-Inertial State Estimation, *IEEE Robotics and Automation Letters (RA-L)*, 2018.

Book Chapters

1. Davide Scaramuzza, **Zichao Zhang**, Visual-Inertial Odometry of Aerial Robots, *Springer Encyclopedia of Robotics*, 2019.

Conference Papers

1. Juichung Kuo, Manasi Muglikar, **Zichao Zhang**, Davide Scaramuzza, Redesigning SLAM for Arbitrary Multi-Camera Systems, *to appear in IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
2. Manasi Muglikar, **Zichao Zhang**, Davide Scaramuzza, Voxel Map for Visual SLAM, *to appear in IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
3. **Zichao Zhang**, Davide Scaramuzza, Beyond Point Clouds: Fisher Information Field for Active Visual Localization, *IEEE International Conference on Robotics and Automation (ICRA), 2019.*
4. **Zichao Zhang**, Davide Scaramuzza, A Tutorial on Quantitative Trajectory Evaluation for Visual(-inertial) Odometry, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
5. **Zichao Zhang**, Davide Scaramuzza, Perception-aware Receding Horizon Navigation for MAVs, *IEEE International Conference on Robotics and Automation (ICRA), 2018.*
6. Ruben Gomez-Ojeda, **Zichao Zhang**, Javier Gonzalez-Jimenez, Davide Scaramuzza, Learning-based Image Enhancement for Visual Odometry in Challenging HDR Environments, *IEEE International Conference on Robotics and Automation (ICRA), 2018*.
7. **Zichao Zhang**, Christian Forster, Davide Scaramuzza, Active Exposure Control for Robust Visual Odometry in HDR Environments*, IEEE International Conference on Robotics and Automation (ICRA), 2017.*
8. **Zichao Zhang**, Henri Rebecq, Christian Forster, Davide Scaramuzza, Benefit of Large Field-of-View Cameras for Visual Odometry, *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

Workshop Papers

1. **Zichao Zhang**, Davide Scaramuzza, Rethinking Trajectory Evaluation for SLAM: a Probabilistic, Continuous-Time Approach, *ICRA 2019 Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR*. **Best Paper Award**