Methods for Large-scale Image-based Localization using Structure-from-Motion Point Clouds

Wentao Cheng

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

04-July-2019
.....
Date

Wentao Cheng
.....
Wentao Cheng

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

04-July-2019

.....

Date

.....

Prof. Weisi Lin

Authorship Attribution Statement

This thesis contains material from three papers published in the following peer-reviewed journals / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Wentao Cheng, Weisi Lin, Xinfeng Zhang, Michael Goesele, Ming-Ting Sun, A Data-Driven Point Cloud Simplification Framework for City-Scale Image-Based Localization. IEEE Transaction on Image Processing, 26(1): 262-275, 2016. The contributions of the co-authors are as follows:

- Prof. Weisi Lin provided the key idea of automatic parameter prediction and revised the whole manuscript.
- I prepared the manuscript and the response letter to reviewers.
- I prepared the code and ran all experiments for the manuscript.
- Dr. Xinfeng Zhang revised the introduction and related work sections of the manuscript.
- Dr. Michael Goesele revised the technical parts of the manuscript.
- Prof. Ming-ting Sun assisted in the design of experiments.

Chapter 4 is published as Wentao Cheng, Kan Chen, Weisi Lin, Michael Goesele, Xinfeng Zhang, Yabin Zhang, A Two- stage Outlier Filtering Framework for City-Scale Localization using 3D SfM Point Clouds. IEEE Transaction on Image Processing, 28(10): 4857-4869, 2019. The contributions of the co-authors are as follows:

- Dr. Kan Chen prepared the figure for illustrating the geometrical constraint, and assisted in designing the geometry-based outlier filter. He also revised the whole manuscript.
- Prof. Weisi Lin helped me design the visibility-based outlier filter, and revised the manuscript.
- I prepared the manuscript and the response letter to reviewers.
- I prepared the code and ran all experiments for the manuscript.
- Dr. Michael Goesele provided the idea of testing different distance thresholds for the comprehensiveness of experiments.
- Dr. Xinfeng Zhang revised the introduction and related work sections of the manuscript.
- Dr. Yabin Zhang assisted in collecting and visualizing the experiment data.

Chapter 5 is published as Wentao Cheng, Weisi Lin, Kan Chen, Xinfeng Zhang, Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization. International Conference on Computer Vision, 2019. The contributions of the co-authors are as follows:

- Prof. Weisi Lin proposed the key idea of using a cascade of match filters.
- I designed the framework proposed in this paper and implemented the framework using C++.
- I prepared the manuscript and the response letter.
- Dr. Kan Chen designed the experiments for the rebuttal period. He also revised the technical parts of the manuscript.
- Dr. Xinfeng Zhang gave useful suggestions for the experimental design.

04-July-2019 Wentao Cheng

....

Date Wentao Cheng

Acknowledgements

It is my great honour to collaborate with and gain knowledge from many great people during my doctoral studies at NTU and TU Darmstadt.

First of all, I would like to thank my supervisor, Prof. Weisi Lin, for giving me superb guidance and help throughout the whole study. His rigour and passion towards science lead me to successfully overcome research problems, and will benefit me for the rest of my life. For every paper I published and submitted, he is always by my side to give me fruitful discussion and careful revision. It is my great pleasure to be supervised by him to solve cutting-edge computer vision problems. I also would like to thank my former co-supervior, Dr. Michael Goesele, for his great guidance in both academic and life during the exchange semesters at TU Darmstadt. Besides, I would like to thank Prof Arjan Kuijper for his help to take over the co-supervisor position.

I also want to thank Fraunhofer Singapore Lab for providing me comfortable environment and powerful computer for research. Especially, I wish to express my thanks to Prof. Wolfgang Müller-Wittig, Prof. Alexei Sourin and Prof. Marius Erdt for their generous support. I also would like to thank Dr. Kan Chen who is my mentor of this lab. I am grateful to him for sparing no effort to help me improve my research and writing skills. Furthermore, I would like to thank Huan Yang, Qiaohong Li, Sharath Chandra Guntuku, Prakhya Sai Manoj, Yabin Zhang, Sheng Yang and other group members for their company and discussions. I want to especially thank my closest colleague, Dr. Xinfeng Zhang, for his insightful feedbacks about my research projects.

I would like to thank my fiancée Hui Zou, who has been waiting for me over the past six years. Without her empathy and unconditional support, it is never easy for me to keep my sanity and confidence. I sincerely cherish her love and sacrifice, and feel grateful to have her as my life partner. Last but not least, I would like to thank my parents for their long-lasting support.

Abstract

Image-based localization, *i.e.* estimating the camera pose of an image, is a fundamental task in many 3D computer vision applications. For instance, visual navigation for self-driving cars and robots, mixed reality and augmented reality all rely on this essential task. Due to easy availability and richness of information, large-scale 3D point clouds reconstructed from images via Structure-from-Motion (SfM) techniques have received broad attention in the area of image-based localization. Therein, the 6-DOF camera pose can be computed from 2D-3D matches established between a query image and an SfM point cloud.

During the last decade, to handle large-scale SfM point clouds, many image-based localization methods have been proposed, in which significant improvements have been achieved in many aspects. Yet, it remains difficult but meaningful to build a system, which (i) robustly handles the prohibitively expensive memory consumption brought by large-scale SfM point clouds, (ii) well resolves the match disambiguation problem, *i.e.* distinguishing correct matches from wrong ones, which is even more challenging in urban scenes or under binary feature representation and (iii) achieves high localization accuracy so that the system can be safely applied in low false tolerance applications such as autonomous driving. In this thesis, we propose three methods that tackle these challenging problems to make a further step towards such an ultimate system.

First of all, we aim to solve the memory consumption problem by means of simplifying a large-scale SfM point cloud to a small but highly informative subset. To this end, we propose a data-driven SfM point cloud simplification framework, which allows us to automatically predict a suitable parameter setting by exploiting the intrinsic visibility information. In addition, we introduce a weight function into the standard greedy SfM point cloud simplification algorithm, so that more essential 3D points can be well preserved. We experimentally evaluate the proposed framework on real-world large-scale datasets, and show the robustness of parameter prediction. The simplified SfM point clouds generated by our framework achieve better localization performance, which demonstrates the benefit of our framework for image-based localization in devices with limited memory resources.

Second, we investigate the match disambiguation problem in large-scale SfM point clouds depicting urban environments. Due to feature space density and massive repetitive structures, this problem becomes challenging if solely depending on feature appearances. As such, we present a two-stage outlier filtering framework that leverages both the visibility and geometry information of SfM point clouds. We first propose a visibility-based outlier filter, which is based on the bipartite visibility graph, to filter outliers on a coarse level. By deriving a data-driven geometrical constraint for urban environments, we present a geometry-based outlier filter to generate a set of fine-grained matches. The proposed framework only relies on the intrinsic information of an SfM point cloud. It is thus widely applicable to be embedded into existing image-based localization approaches. Our framework is able to handle matches of very large outlier ratio and outperforms state-of-the-art image-based localization methods in terms of effectiveness.

Last, we aim to build a general-purpose image-based localization system that simultaneously solves the memory consumption, match disambiguation and localization accuracy problems. We adopt a binary feature representation and propose a corresponding match disambiguation method by adequately utilizing the intrinsic feature, visibility and geometry information. The core idea is that we divide the challenging disambiguation task into two different tasks before deriving an auxiliary camera pose for final disambiguation. One task focuses on preserving potentially correct matches, while another focuses on obtaining high quality matches to facilitate subsequent more powerful disambiguation. Moreover, our system improves the localization accuracy by introducing a quality-aware spatial reconfiguration method and a principal focal length enhanced pose estimation method. Our experimental study confirms that the proposed system achieves superior localization accuracy using significantly smaller memory resources comparing with state-of-the-art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the past decades, the world has witnessed tremendous advancements of digital camera sensors, making them low cost and robust channels to capture visual reality. Consequently, the vast majority of personal devices, *e.g.* mobile phones, drones and autonomous vehicles, are equipped with camera sensors to interact with underlying environments. The widespread usage of camera sensors results in that the number of existing images grows with incredible speed. As a result, computer vision algorithms and methods, which can understand the image data and thereby provide practical applications, are urgently needed. In both academia and industry, computer vision researchers are still working consistently to develop powerful systems that make use of images to make our daily life more productive and convenient. For example, Google Street View enables users to virtually visit many cities by interactive panorama images. Digital preservation techniques turn historical monuments, *e.g.* Dunhuang Mogao Grottoes, into digitalized models that are more resistant to physical damage. Style transfer techniques are able to apply a new artistic style to an image while preserving its original content.

Among all computer vision tasks, 3D vision remains a long-term active topic, which involves 3D scene reconstruction from images, 3D scene understanding with images, *etc.* Given multiple images depicting a scene from different views, one can build a point cloud via Structure-from-Motion (SfM) techniques. For the past two decades, SfM techniques have made a great stride in both efficiency and robustness. Examples are Bundler [SSS06] and COLMAP [SF16], which make it possible to reconstruct a large-scale SfM point cloud from Internet image collections in a short amount of time.

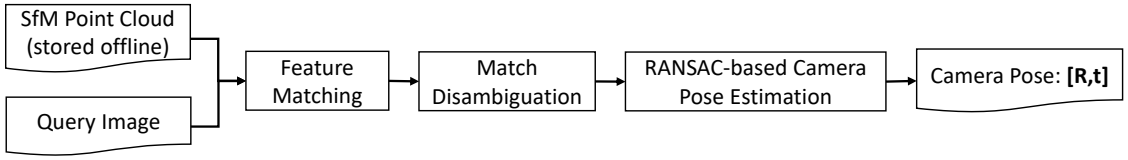


FIGURE 1.1: A general pipeline of image-based localization using SfM point clouds.

The easy availability of such large-scale SfM point clouds subsequently unleashes possibilities in various modern computer vision applications. For example, in the context of autonomous driving, cars can be localized in a 3D environment using mounted camera sensors to facilitate subsequent tasks such as path planning. For the task of augmented and virtual reality, mobile phone users are able to enjoy an advanced interaction with the aid of a pre-computed SfM point cloud.

At the core of aforementioned 3D vision applications is a fundamental problem: *how to determine the 6-DOF camera pose, a.k.a position and orientation, of an image in an SfM point cloud*? Such a problem is also referred as the *image-based localization*[1] problem. Pioneering work in localizing an image estimate a rough position by means of retrieving relevant database images, which encode location priors from the Global Positioning System (GPS). Obviously, this is insufficient for modern computer vision tasks such as visual navigation for autonomous vehicles, which require to compute an accurate camera pose to ensure safety. An SfM point cloud not only provides us the 3D scene geometry, but also it offers a better feature representation of the scene than database images. During SfM reconstruction, informative local feature descriptors that contribute to triangulate a point are well preserved. In the meantime, feature descriptors that appear in uninformative regions, *e.g.* grass, sky and water, usually are discarded. As a result, an SfM point cloud allows us to establish 2D-3D matches between feature descriptors in a query image and feature descriptors attached to points. Consequently, the camera pose of a query image can be computed from established 2D-3D matches by applying a perspective-n-point (pnp) pose solver with RANSAC [Fis81]. Fig. 1.1 illustrates a general pipeline of image-based localization.

In this thesis, we mainly focus on image-based localization in large-scale outdoor settings. Fig. 1.2 shows a typical example of localizing an image in an SfM point cloud depicting Dubrovnik (a historic city in Croatia) [LSH10]. As a large-scale SfM point cloud usually contains tens of millions of points and feature descriptors, it becomes

[1]In the rest of this thesis, unless specified, image-based localization uses an SfM point cloud as the default 3D scene representation.

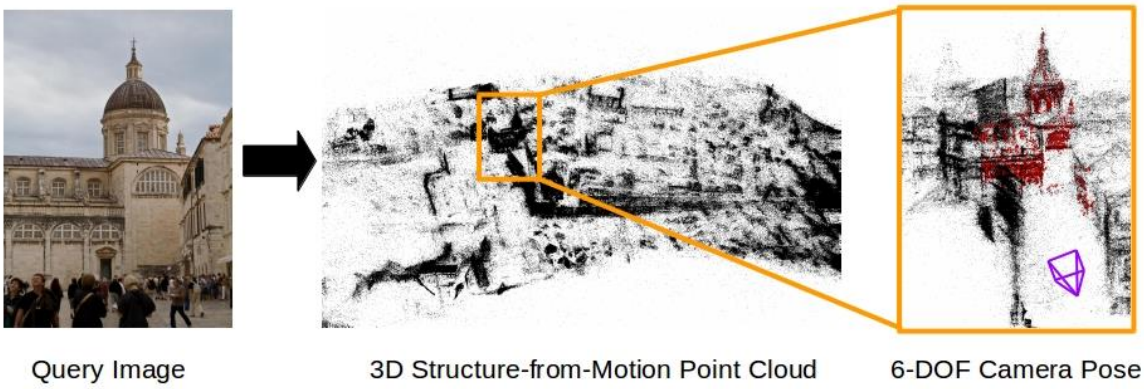


FIGURE 1.2: An illustration of image-based localization in a large-scale SfM point cloud [LSH10] depicting Dubrovnik, a historic city in Croatia.

more and more difficult to build an efficient, effective, robust and accurate image-based localization system. To this end, there are three main challenging problems that must be addressed.

Memory Consumption: A large-scale SfM point cloud brings in a memory consumption problem, since too many high-dimensional feature descriptors should be stored. As a side effect, the feature matching process against a large-scale SfM point cloud becomes prohibitively time consuming for daily usage. In order to successfully run image-based localization applications on devices with limited memory resources (*e.g.* mobile phones, drones, cars and robots), large-scale SfM point clouds should be "compressed" in a certain way to break the bottleneck of memory consumption and efficiency.

A straightforward way to tackle the memory consumption problem is the so-called *SfM point cloud simplification* methods, which select a subset of highly informative points and feature descriptors via the K-Cover theory [Fei98]. The resultant simplified SfM point cloud is able to maintain a satisfactory localization performance in the case that the number of points is extremely reduced, *e.g.* 1% of the original SfM point cloud. However, existing SfM point cloud simplification methods produce dramatically variations on different datasets using a fixed parameter setting. Hence, there is a strong need for a fully automatic SfM point cloud simplification method that can adjust the parameter by exploiting the point cloud data itself.

An alternative way to reduce the memory consumption is to replace high-dimensional feature descriptors, *e.g.* SIFT [Low04] stored in an SfM point cloud with compact binary feature representations. In addition, the fast computation in comparing the

distance between two binary feature descriptors significantly improves the efficiency of image-based localization. The dimension reduction of feature descriptors inevitably results in the loss of discriminative power, making it more and more difficult to distinguish correct matches simply from feature appearance. Thereby, an image-based localization method, which can leverage the advantage of binary feature representation and remedy the brought disadvantage, is needed.

Match Disambiguation: As stated, image-based localization methods typically require a hypothesis-and-verification strategy in RANSAC to handle wrong matches. In each RANSAC iteration, the probability of obtaining a valid camera pose hypothesis is approximately ε^n, where n represents the minimal number of matches used in perspective-n-point pose solvers, and ε represents the proportion of correct matches (a.k.a inlier ratio). In order to find a good solution from matches of very low inlier ratio, one may test all possible hypotheses, making the entire localization process slow. In practice, one can set a certain number of RANSAC iterations or let advanced RANSAC variations adjust themselves. As such, it is crucial to apply a match disambiguation method before RANSAC-based camera pose estimation to efficiently obtain a set of matches of high inlier ratio.

Typically, Lowe's ratio test [Low04] based on feature appearance is an appropriate method for match disambiguation between two images. Yet, the dense feature space corresponding to a large-scale SfM point cloud makes this ratio test lose its effectiveness, *i.e.* correct matches tend to be easily rejected. For SfM point clouds depicting urban environments, the match disambiguation problem becomes more challenging due to the vast existence of repetitive patterns. Elaborate match disambiguation or outlier filtering methods shift to exploit the visibility or geometry information, which is more reliable than Lowe's ratio test. However, these methods require either heavy additional geometrical assumptions or vast computation resources, making them unsuitable for practical image-based localization applications.

Accurate Camera Pose Estimation: Once 2D-3D matches have been well disambiguated, the accuracy of camera pose estimation remains a key issue for image-based localization. A major concern is how to prevent obtain a degenerate camera pose as the final RANSAC solution so that high localization accuracy can be achieved for tasks such as autonomous driving. Despite the quality of matches (*i.e.* number of correct matches), the spatial distribution of matches is another crucial factor for preventing a

degenerate solution. Surprisingly, researchers in the area of image-based localization pay little attention to the degenerate camera pose problem.

1.1 Problem Statement

Problem A—SfM Point Cloud Simplification: First, let us consider using SfM point cloud simplification methods to handle the prohibitively large memory consumption of a large-scale SfM point cloud. As stated, using a fixed parameter K value in existing K-Cover based methods to generate a simplified SfM point cloud leads to a large variation, *i.e.* inconsistent localization performance, among different datasets. We may naturally wonder:

Which kind of intrinsic characteristic of an SfM point cloud leads to the variation in applying K-Cover based SfM point cloud simplification methods?

Consequently, we aim to build a fully automatic SfM point cloud simplification system, which can predict a reasonable parameter K value for a specified dataset. Thereby, the following question must be addressed:

How to model the relationship between the parameter K in K-Cover based algorithms and the found intrinsic characteristic of an SfM point cloud?

The standard K-Cover algorithm treats the coverage of the original scene as the only criteria for the SfM point cloud simplification problem. Yet, existing methods show that other factors, *e.g.* feature distinctiveness and visibility probability, are also important for improving the quality of simplified point clouds. Inspired from existing methods, we aim to answer:

What is the scenario to consider visibility probability, and how to efficiently integrate it into the K-Cover algorithm?

Problem B—Match Disambiguation: By relaxing the criteria of traditional feature-wise match disambiguation methods, one has to fully utilize the visibility or geometry cues to handle the resultant matches with large outlier ratio. As a consequence, multiple questions arise concerning feature-, visibility-, and geometry-wise match disambiguation.

Existing methods typically leverage the same visibility cue that 3D points corresponding to correct 2D-3D matches are frequently co-visible in database images. Based

on this cue, image voting is performed to find relevant database images. Subsequently, the matches voting to relevant database images are regarded as potentially correct. The availability of both feature- and visibility-wise match disambiguation methods leads to an interesting question:

Are they mutual exclusive in match disambiguation?

In order to overcome the limitations of existing geometry-wise match disambiguation methods, we must answer:

Is there a geometrical proxy, which does not require other sensors except camera? If so, is it efficient and effective in match disambiguation?

Subproblem B.1—for Urban Environments: In addition, as large-scale SfM point clouds depicting urban environments are widely used in autonomous driving, tourist navigation, *etc*, a special question arises:

Can we derive a data-driven geometrical constraint from SfM point clouds depicting urban environments? How can this geometrical constraint be applied to disambiguate matches?

Subproblem B.2—for Binary Feature Representation: In the case that high-dimensional feature descriptors are replaced by compact binary feature descriptors for reducing memory consumption, relaxing the criteria in feature-wise match disambiguation makes the quality of resultant matches even worse. Thereby, we aim to answer:

Under binary feature representation, how can we better measure the feature distinctiveness of a 2D-3D match?

Problem C—Accurate Camera Pose Estimation: Finally, in RANSAC-based camera pose estimation, we aim to handle the degenerate camera pose problem efficiently. In contrast to existing method that tries to avoid selecting a degenerate camera pose in the verification stage of RANSAC as the best solution, we seek to make a step further by answering:

How can we avoid a degenerate camera pose in both the hypothesis and verification stage of RANSAC?

As aforementioned, the degenerate camera pose problem is closely related to both the quality and spatial distribution of matches. Concretely, we also aim to answer:

How to make the matches better distributed without severely degrading the quality?

1.2 Approaches and Contributions

In this thesis, we propose three works to address challenging problems in large-scale image-based localization using SfM point clouds.

Data-driven SfM Point Cloud Simplification: To answer **Problem A**, we propose a data-driven framework for efficient and effective SfM point cloud simplification [CLZ+16][2] . We derive a parameter prediction model to predict a reasonable parameter K used in K-Cover based point cloud simplification approaches to meet the requirement of the image-based localization problem according to the statistic characteristics of the original point cloud data. The key idea behind this model is to evaluate the potential of a point cloud for establishing sufficient 2D-3D feature matches, which is crucial for the image-based localization task. In addition, we propose a weighted K-Cover algorithm to simplify an SfM point cloud based on the visibility probability of each point which can be efficiently extracted from the original point cloud. Specifically, an adaptive exponential weight function is proposed and integrated into the basic greedy heuristic algorithm to solve the SfM point cloud simplification problem. The experiment results show that our prediction algorithm achieves a high reliability. Also, the proposed weighted K-Cover algorithm significantly outperforms the existing simplification methods in localization performance.

Two-stage Outlier Filtering for Urban Image-based Localization: Second, we aim to answer **Problem B** with **Subproblem B.1**, which is to handle the match disambiguation problem in large-scale SfM point clouds depicting urban environments. As such, we introduce a two-stage outlier filtering framework [CCL+19][3] that consists of an improved visibility-based outlier filter and a novel geometry-based outlier filter. The two-stage framework overcomes the limitations of both outlier filters with a coarse-to-fine design, and achieves both efficiency and accuracy in disambiguating

[2] **Wentao Cheng**, Weisi Lin, Xinfeng Zhang, Michael Goesele, Ming-Ting Sun, A Data-Driven Point Cloud Simplification Framework for City-Scale Image-Based Localization. IEEE Transaction on Image Processing, 26(1): 262-275, 2016.

[3] **Wentao Cheng**, Kan Chen, Weisi Lin, Michael Goesele, Xinfeng Zhang, Yabin Zhang, A Two-stage Outlier Filtering Framework for City-Scale Localization using 3D SfM Point Clouds. IEEE Transaction on Image Processing, 28(10): 4857-4869, 2019.

matches of very large outlier ratio.

The visibility-based outlier filter, which consists of database image voting, re-ranking and match augmentation operations, is conducted on the image-level to remove outliers in a coarse level. A database image voting method is proposed based on the widely known knowledge that correct matches exhibit a strong co-visibility relationship [LSHF12, SHR+15]. To further improve the filtering performance, we introduce a re-ranking scheme to eliminate falsely voted database images. Previous methods [ZSP15, SEKO17, CSC+17] assume that in the initialization step, the relaxed SIFT ratio test does not reject any correct matches. However, this assumption is untenable when dealing with the dense feature space of the city-scale SfM point cloud. To this end, we propose a match augmentation scheme to carefully recover rejected correct matches with the aid of selected database images. Although the proposed visibility-based outlier filter is efficient when dealing with extremely large outlier ratio scenarios, e.g. 99% outliers, the resultant matches may still contain a large number of outliers due to the limited accuracy of the database image voting procedure.

The second stage is a geometry-based outlier filter based on a novel data-driven geometrical constraint. Our key observation is that, in a city-scale SfM point cloud, there are many 3D points that can only be observed by nearby cameras due to strong view occlusions. We denote such 3D points as *locally visible points*. Based on this observation, we derive a geometrical constraint to restrict the position of camera that can observe the *locally visible points*. Previous geometry-based outlier filters either heavily rely on additional priors about the vertical direction and approximate height of a camera relative to an SfM point cloud [ZSP15, SEKO17], or require a set of high quality matches for statistically pruning outliers [CSC+17]. The derived geometrical constraint in our method does not require any prior knowledge about the camera model. In addition, this geometrical constraint enables us to efficiently handle potential low quality matches which are generated by the visibility-based outlier filter in the first stage. The effectiveness and efficiency of the proposed two-stage framework and its individual modules are comprehensively analyzed. Based on the extensive experimental results, the matches generated by our method show a high reliability for successful large-scale urban image-based localization.

Accurate Image-based Localization under Binary Feature Representation: Last, we aim to answer **Problem B** with **Subproblem B.2** and **Problem C**. We adopt a binary feature representation, which is memory-efficient, as the input to our

method. Our method [CLCZ19][4] therefore aims to well disambiguate matches in a cascaded manner by sequentially leveraging the intrinsic feature, visibility, and geometry information in an SfM point cloud. When engaging one type of information, we use a relaxed criteria to reject matches and retain a match pool that focuses on preserving correct matches. In the meantime, we use another strict criteria to obtain high confident matches, which facilitate the subsequent disambiguation.

In feature-wise match disambiguation, we reformulate a traditional match scoring function [JDS09] with a bilateral Hamming ratio test to better evaluate the distinctiveness of matches. In visibility-wise disambiguation, we explore the point-image relationship to disambiguate matches by retrieving relevant database images. Moreover, we propose a two-step match selection method by exploring the point-point relationship, which allows us to obtain substantial 2D-3D matches for computing an auxiliary camera pose. In geometry-wise disambiguation, we apply this auxiliary camera pose on the retained match pool to reject matches by means of re-projection error.

Our proposed framework can also improve the localization accuracy by handling degenerate camera pose. The first observation is that, correct matches that appear in sparse regions, are essential to establish a non-degenerate camera pose hypothesis. Due to the scarcity of such matches, they are usually neglected during camera pose estimation. Therefore, we propose a quality-aware spatial reconfiguration method to increase the possibility of sampling with such matches in RANSAC-based pose estimation. The second observation is that, several top ranked camera pose hypotheses that have similar and realistic focal length values, are more accurate than the camera pose hypothesis with the largest number of inliers. Based on this observation, we shift the focus to find a principal focal length value so that we can obtain a more accurate camera pose accordingly.

We evaluate our proposed image-based localization framework on several real-world datasets, a comprehensive comparison with the state-of-the-art methods demonstrates that the proposed framework can achieve very competitive localization accuracy. In the meantime, we require less than 20% memory consumption comparing with other state-of-the-art methods.

The proposed three projects have been published on peer-reviewed journals or conferences `https://scholar.google.com.sg/citations?user=c2oVlqUAAAAJ&hl=en`. The papers about data-driven SfM point cloud simplification [CLS15, CLZ+16] have been published in 2015 and 2016 respectively. Excluding the self-citations,

[4] **Wentao Cheng**, Weisi Lin, Kan Chen, Xinfeng Zhang, Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization. International Conference on Computer Vision, 2019.

these two papers have been cited nine times. In the articles that cite our papers, the proposed data-driven SfM point cloud simplification method usually is regarded as a representative work in point cloud simplification and compression. The papers [CCL+19, CLCZ19] of other two projects have been recently published on IEEE Transaction on Image Processing and International Conference on Computer Vision respectively in 2019. Due to the short publishing period, these two papers have been cited one time. However, the code of [CLCZ19] has been published on `https://github.com/wentaocheng-cv/cpf_localization`. From November 2019 to February 2020, this code repository has received three stars, which demonstrate the exposure of [CLCZ19] to the research community.

1.3 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 gives a survey about foundations and works that are related to image-based localization. Chapter 3 presents the data-driven SfM point cloud simplification framework. Chapter 4 presents the two-stage outlier filtering framework for urban image-based localization. Chapter 5 presents the accurate image-based localization framework under binary feature representation. Chapter 6 gives a conclusion to the aforementioned works and introduces potential research topics for future.

Chapter 2

Foundations and Literature Review

The foundations of image-based localization include several key concepts in computer vision. For instance, camera model, 3D reconstruction using Structure-from-Motion techniques, image feature description, feature matching and match disambiguation, camera pose solvers and RANSAC strategies all are involved in image-based localization. From Section 2.1 to Section 2.6, we provide a brief survey about the foundations of this thesis. Subsequently, we present a detailed review about recent image-based localization methods using image databases, SfM point clouds from Section 2.7 to Section 2.8.

2.1 Camera Model

In SfM point clouds, the positions of 3D points are measured in the world coordinate system. Meanwhile, each camera that takes query images has a local camera coordinate system. Let $\mathrm{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ be the rotation matrix and translation vector respectively. A world point $\mathbf{X}_W$ can be transformed into camera coordinates $\mathbf{X}_C$ as following:

$$\mathbf{X}_C = \mathrm{R} \cdot \mathbf{X}_W + \mathbf{t} = [\mathrm{R}|\mathbf{t}] \cdot \tilde{\mathbf{X}}_W, \tag{2.1}$$

where $\tilde{\mathbf{X}}_W = (\sigma\mathbf{X}_W, \sigma)^{\mathrm{T}}$ is the homogeneous representation of $\mathbf{X}_W$ with a non-zero scale factor σ.

To model the mathematical relationship between a 3D point and its projection on image plane, we adopt the standard pinhole camera model [HZ03]. Let $\mathbf{C} \in \mathbb{R}^3$ be

the center of projection. The projection of a 3D point $\tilde{\mathbf{X}}_C$ on an image plane can be computed as the intersection of the image plane and the line through $\mathbf{X}_C$ and $\mathbf{C}$. The internal camera calibration matrix $\mathbf{K}$ is defined as following:

$$\mathbf{K} = \begin{bmatrix} f & s & p_x \\ 0 & \alpha f & p_x \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

where f and s denotes the focal length and skew parameter respectively. The pixel is a square when the aspect ratio $\alpha = 1$. The offset from the origin of local camera coordinate system is represented by $(p_x, p_y)^{\mathrm{T}}$. Typically, we can adopt a reduced version of the internal camera calibration matrix as following:

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

Having obtained both the internal and external camera parameters, the image coordinates by projecting a 3D world point to an image plane can be computed as following:

$$\tilde{\mathbf{x}} = \mathbf{K} \cdot [\mathrm{R}|\mathbf{t}] \cdot \tilde{\mathbf{X}}_W = \mathrm{P} \cdot \tilde{\mathbf{X}}_W, \tag{2.4}$$

where $\mathrm{P} \in \mathbb{R}^{3\times4}$ denotes the projection matrix of the camera model. The position of camera center $\mathbf{C}$ in the world coordinate system can be defined as $\mathbf{C} = -\mathrm{R}^{\mathrm{T}}\mathbf{t}$.

2.2 Structure-from-Motion Reconstruction

In this thesis, we leverage the point cloud reconstructed via Structure-from-Motion (SfM) techniques as the 3D scene representation. Thereby, it is necessary to briefly describe the process of SfM reconstruction to better understand the data structure inside an SfM point cloud. Typical SfM techniques mainly involve three steps. First, pairwise image feature matching is performed so that the relative motion of database image pairs can be obtained. Second, the absolute camera poses of database images and positions of 3D points are recovered in a global coordinate system. Third, the estimated parameters are refined by non-linear bundle adjustment methods.

Incremental SfM techniques have been widely used in real-world applications due

to high efficiency. The entire reconstruction process starts with estimating the global camera poses from an image pair. Other images are subsequently added into reconstruction in an iterative manner. The basic algorithms in incremental SfM methods have been well studied in [Pol99]. Over recent years, several open source incremental SfM techniques [SSS06, Wu13, SF16] have been published. Snavely *et al.* improve the robustness of standard incremental SfM reconstruction pipeline by initializing with an image pair, which have sufficient number of matches and a wide baseline [SSS06]. Wu *et al.* introduce a preemptive feature matching strategy [Wu13] that significantly speedups the reconstruction process. Schonberger *et al.* propose the COLMAP method by refining the next view selection and other key steps in incremental SfM reconstruction [SF16]. In contrary, global SfM techniques simultaneously compute the absolute camera poses for all database images. A common strategy of global SfM is to estimate the global rotations first and then find a consistent embedding of translations [HTDL13]. The global rotation problem can be solved efficient in least square fashion [MP07]. Subsequently, recent works mainly focus on deriving robust algorithms for the optimization of translations [JCT13, WS14, CT15].

2.3 Image Features

Global features, *e.g.* Histogram Oriented Gradients (HOG) [DT05], describe an image by utilizing pixel-wise information. During the past decade, deep learning techniques have significantly improved the powerfulness of global features in image classification [KSH12], image recognition [HZRS16, SZ14] and semantic segmentation [LSD15]. Meanwhile, local features that describe an image in a sparse way unleash the possibility of many computer vision tasks. Extracting a local feature generally contains two key components: a detector that identifies interest regions with high repeatability, and a subsequent descriptor that characterizes the interest regions.

In modern SfM reconstruction methods [SSS06, SF16], a key step is to detect and describe the local appearance information of an image using sparse feature descriptors. Thereby the geometrical relationship between image pairs can be established with feature matches. In image-based localization task, a typical pipeline [SLK11] also usually starts with local feature extraction. In the following, we will briefly discuss recent works about local feature extraction.

As a pioneer in interest region detection, Harris *et al.* propose the Harris detector

that finds corners and infer features from an image [HS+88]. Based on the Harris detector, Shi *et al.* introduce a more robust corner detector by an eigenvalue-based score function [ST93]. To make feature invariant to scale and orientation, Lowe *et al.* leverage a difference-of-Gaussian function that identifies highly reliable interest regions or points [Low04]. The corresponding SIFT (Scale Invariant Feature Transform) descriptor by measuring local image gradients has proven to be insensitive to various geometrical and photometric transformations. Due to the robustness of SIFT feature descriptor, the majority of image-based localization approaches use it to describe the visual information of an image. In this thesis, we also choose to extract the SIFT feature descriptors from a given query image for image-based localization. Bay *et al.* propose a feature detector that is very fast while in the meantime exhibits high repeatability [BTVG06]. The proposed SURF (Speeded-Up Robust Features) feature detect blob-like structures at pixels that correspond to the largest determinant. Rosten *et al.* show that matching learning can significantly improve the efficiency of corner detectors [RD06], which work well in real-time tracking applications. They employ a machine learning algorithm to fast determine whether a pixel is a corner or not.

After finding interest regions using feature detectors, one should use an appropriate feature descriptor to summarize the information. Still, the most prevailing feature descriptor is SIFT [Low04]. Ke *et al.* introduce PCA-SIFT that is more compact than standard SIFT representation by applying Principal Components Analysis (PCA) to image patches [KS+04]. Based on the SIFT descriptor, Mikolajczyk *et al.* propose the gradient location and orientation histogram (GLOH) by changing to compute SIFT descriptors for a log-polar location [MS05]. To make dense feature matching efficient, Tola *et al.* propose DAISY descriptor that uses a circularly symmetrical weighting kernel [TLF10]. Alternatively, The BRIEF descriptor speeds up the construction and matching of descriptors by using a small number of intensity difference tests [CLO+12]. Rublee *et al.* propose ORB descriptor [RRKB11] that consists of an orientated FAST and rotated BRIEF descriptor. For a more detailed review about local feature detectors, please refer to [TM+08]. In addition, Mikolajczyk *et al.* evaluate the robustness of different local feature descriptors in [MS05].

2.4 Feature Matching

2.4.1 Nearest Neighbor Search

In order to compute the camera pose of a query image, 2D-3D matches should be established. As aforementioned, the matches can be found by matching between feature descriptors in a query image and an SfM point cloud. To perform exact nearest neighbor search in feature space, a simple way is to brute force search in either a query image or an SfM point cloud. However, the efficiency of exact nearest neighbor search significantly decreases when facing high-dimensional data such as SIFT. Based on the famous kd-tree structure [FBF76], Arya *et al.* propose the BBD-tree (balanced box-decomposition tree) that works well for approximate nearest neighbor search in high-dimensional feature space [AMN+94]. For very large image datasets, Nister *et al.* introduce an vocabulary tree that quantizes local visual descriptors using the hierarchical k-means algorithm [NS06]. Muja demonstrate that the randomized kd-tree algorithm is very effective for approximate nearest neighbor search in high-dimensional feature space [ML14]. In addition, they propose a priority search k-means algorithm that clusters data by leveraging the full distance across all dimensions. Finally, they release the open-source FLANN (fast library for approximate nearest neighbors) library to the computer vision community.

2.4.2 Match Disambiguation

In image-based localization tasks, the best match for a descriptor from a query image usually is found by searching its nearest neighbor among the descriptors in an SfM point cloud. Ideally, using at most six correct 2D-3D matches can compute the camera pose of a query image [HZ03]. However, there is no guarantee that every descriptor from a query image can establish a correct 2D-3D match, since it may appear in a region that is not reconstructed in the SfM point cloud. Though RANSAC [Fis81] in the later camera pose estimation step can handle wrong matches by a hypothesis-and-verification strategy, it will easily fail when given a set of highly contaminated matches. Thus, a simple and effective match disambiguation method, which preliminarily discard wrong matches before RANSAC-based camera pose estimation, is needed. The SIFT ratio test [Low04] has been widely used in many computer vision applications

for match disambiguation. It only preserves a match if the ratio between the distance to the nearest and second nearest neighbor is below a certain threshold. The motivation behind the SIFT ratio test is that the distance of a correct match needs to be significantly closer than the closest wrong match to achieve reliable matching. While the distance of a wrong match will likely be similar with other wrong matches due to the high dimensionality of the feature space. When performing feature matching between two images, the SIFT ratio test is effective by discarding most wrong matches while discarding very few correct matches. However, as the feature space of a large-scale SfM point cloud usually is much denser than an image, the ratio test becomes unreliable since it discards many correct matches. To effectively handle the intense feature ambiguity in image-based localization, recent works shift to exploit the visibility or geometry information in an SfM point cloud. For example, Li *et al.* propose a RANSAC sampling strategy [LSHF12] by prioritizing hypotheses in which points are frequently co-visible. In Section 2.8.2, we will introduce alternative match disambiguation methods in detail.

2.5 Camera Pose Estimation

As the final step of image-based localization, camera pose estimation aims to estimate the 6-DOF camera pose, a.k.a. $[\mathrm{R}|\mathbf{t}]$, from 2D-3D matches between 3D world points and corresponding pixel observations in a query image. In this section, we first discuss minimal pose solvers that require minimal number of correct matches. Robust methods that handle noisy input matches by RANSAC will also be discussed.

Assuming that the camera calibration is fully unknown, this requires camera pose estimation methods to compute both the internal and external camera parameters $\mathbf{K} \cdot [\mathrm{R}|\mathbf{t}]$ a.k.a. the projection matrix $\mathbf{P}$. In order to obtain the projection matrix, the Direct Linear Transform (DLT) algorithm rewrites an ordinary system of linear equations as a matrix equation [HZ03]. Since $\mathbf{P}$ has 11 degrees of freedom due to the scale uncertainty, using six 2D-3D matches is sufficient to compute the projection matrix.

In a practical scenario, a common assumption of the pinhole camera model is that the principal point is the image center and there is no skew effect. Consequently, the focal length remains the only unknown parameter of the camera calibration. Considering the unknown rotation and translation of camera, there are overall seven degrees

of freedom to be solved. Bujnak *et al.* propose a P4Pf method that requires four 2D-3D matches to obtain the focal length and external camera parameters [BKP08]. They propose to use either hidden variable or Gröbner basis based methods to solve the problem of multiple polynomial equations. For fisheye lenses, there exist a heavy radial distortion effect that influences the accuracy of camera pose estimation. Josephson *et al.* propose a minimal pose solver that requires only four 2D-3D matches to solve radial distortion, focal length and camera pose [JB09]. Similarly, Bujnak introduce a more efficient pose solver to solve the camera pose estimation problem under radial distortion [BKP11]. The speedup mainly comes from separately handling non-planar and planar scenes.

In the case that the internal camera calibration is fully known, using three 2D-3D matches (P3P) is sufficient to obtain the camera pose. Fischler *et al.* introduce the P3P problem, together with the RANSAC strategy, to modern computer vision research [Fis81]. Three correct 2D-3D matches are the minimal information to solve the P3P problem. Consequently, there maybe at most four solutions for world points in front of camera center. Additional information, *e.g.* additional matches, thus is required to disambiguate these four solutions. A detailed evaluation of different algebraic methods solving the P3P problem are discussed in [HLON94]. During the past two decades, many P3P methods with superior efficiency and robustness have be proposed. Gao *et al.* use both algebraic and geometrical approaches to solve the P3P problem [GHTC03]. They also provide a complete solution classification for the P3P equation system. In contrast to the above methods that derive intermediate point positions in the local camera frame, Kneip *et al.* propose a novel parameterization framework that directly computes the aligning transformation in a singe stage [KSS11].

2.6 RANSAC

In the previous section, we have discussed how to compute the camera pose assuming that the 2D-3D matches used are correct. In practice the 2D-3D matches established by approximate nearest neighbor search usually contain a significant number of wrong matches. Regarding robust camera pose estimation using noisy matches, the RANdom SAmple Consensus (RANSAC) algorithm [Fis81] is arguably the most prevalent strategy. RANSAC is a non-deterministic method that can efficiently provides a good solution with a certain probability. In each iteration, RANSAC randomly selects a

minimal subset from all 2D-3D matches to compute a camera pose hypothesis using the aforementioned pose solvers. Then each match is verified by evaluating the consistency with the camera pose hypothesis. Typically, a match can be regarded as an inlier if the re-projection error is below a defined threshold. The camera pose hypothesis supported by the largest number of inliers is finally selected as the best solution.

Let I be the number of correct matches (inliers) among all M 2D-3D matches. Suppose a minimal subset of s matches are required to obtain a camera pose hypothesis. Therefore, the probability that RANSAC samples s inliers and computes a correct camera pose hypothesis is

$$P_{\text{correct-sample}} = \prod_{j=0}^{s-1} \frac{I-j}{M-j} \leq \left(\frac{I}{M}\right)^s = \sigma^s, \tag{2.5}$$

where σ denotes the inlier ratio. Consequently, the probability that RANSAC fails to obtain at least one correct camera pose hypothesis in k iterations is

$$\eta = (1-\sigma^s)^k. \tag{2.6}$$

Therefore, RANSAC needs to run at least

$$k_{\max} = \left\lfloor \frac{\log\eta}{\log(1-\sigma^s)} \right\rfloor + 1 \tag{2.7}$$

to ensure that a correct camera pose is found with probability $1-\eta$. Based on Eq. 2.7, we can notice that the efficiency of RANSAC strongly relies on the inlier ratio and the minimal number of matches required by pose solvers.

During the past two decades, many RANSAC variants have been proposed. The standard RANSAC requires to evaluate all tentative matches in the hypothesis verification stage. To improve the efficiency of hypothesis evaluation, Chum *et al.* propose a Randomized RANSAC (R-RANSAC) algorithm that evaluate only a very small fraction of matches with high confidence [CM02]. Based on R-RANSAC, an optimal hypothesis verification method using sequential probability ratio test (SPRT) is proposed [CM08]. This method also does not require to know the fraction of outliers in advance, making it more robust to handle practical cases. Unlike the standard RANSAC that treats all matches equally, Chum *et al.* propose the Progressive Sample Consensus (PROSAC) algorithm [CM05], in which high confidence matches defined by a similarity metric function are prioritized in the model hypothesis stage. The computational

savings are mainly results from the observation that matches with high similarity are more likely to be inliers.

2.7 Localization using Image Databases

A large-scale scene is usually represented by a database of images before powerful Structure-from-Motion algorithms [SSS06, Wu13, SF16] were proposed. In this case, image-based localization is also referred as the location recognition problem. Given a database of street-view images with geotags, Zhang *et al.* propose a coarse-to-fine localization system [ZK06] that relies on the SIFT feature descriptor. Relevant database images are firstly retrieved by wide-baseline matching and voting. The final position of the query image is then refined by view interpolation techniques between top two relevant database images. However, when facing a large-scale database, this method becomes computationally expensive due to brute force retrieval. To guess where an image was taken on the earth, Hays *et al.* collect over six millions geotagged images and propose the *IM2GPS* method in which relevant database images are ranked based on an unified image descriptor [HE08]. Weyand *et al.* train a convolutional neural network (CNN) [SLJ+15] using geotagged images, which are quantized into small spatial cells [WKP16]. In addition, they extend the CNN model with a long short-term memory (LSTM) method to exploit the temporal coherence in a group of images.

In the past two decades, the computer vision community has witnessed tremendous advancement in image retrieval [SZ03, NS06, PCI+07] systems using local feature descriptors. Based on these image retrieval systems, many image-based localization methods have been proposed by leveraging the location priors such as GPS data. Schindler *et al.* show that training an visual vocabulary with informative features, which frequently appear in specific locations but rarely appear in other locations, significantly improves the localization performance [SBS07]. Similarly, Knoop *et al.* use geotags in an image database to detect and remove confusing features that depict uninformative objects such as trees and grass [KSP10]. They remove spatially clustered features that frequently match to the database images far away from the underlying database image. In order to facilitate city-scale localization, two image datasets [AKTS10, CBK+11] containing more than one million images are carefully constructed and released. Avrithis *et al.* collect nearly one million geo-tagged images depicting 22 European cities from Internet. A hierarchical clustering method [AKTS10], which

sequentially applies geographical and visual clustering, is proposed to construct a scene map for efficient indexing. The San Francisco dataset constructed by [CBK+11] contains 1.7 millions of street-view database images with ground truth labels and geotags. The facade-aligned and view-aligned scene representation are fused to increase the localization performance. In addition, they show that histogram equalized and upright feature descriptors are beneficial to the localization problem. Zamir *et al.* propose to first perform detailed feature matching so that each query descriptor can find its nearest neighbor. The established matched are then pruned by leveraging the geotags attached to database images [ZS10]. Sattler *et al.* handle the geometric burst problem [JDS09] in both image and place level [SHSP16], in which the place is defined by clustering database images from geotags.

2.8 Localization using SfM Point Clouds

Given an SfM point cloud, 2D-3D matches should first be established between the feature descriptors in a query image and SfM point cloud. The 6-DOF camera pose can then be computed using the found 2D-3D matches. In this section, we survey existing image-based localization approaches that rely on 3D scenes represented by SfM point clouds. Different schemes in feature matching, feature representations, and match disambiguations will be discussed.

2.8.1 SfM Point Cloud Simplification

In the case that a 3D point cloud is reconstructed by SfM techniques, the 6-DOF camera pose for a query image can be computed. In order to enable large-scale image-based localization on devices with limited memory storage, *e.g.* smartphones or drones, it is crucial to reduce the size of SfM point clouds. Havlena *et al.* [HHS13] propose to compute a minimum connected dominating set from database images and use the subset of images to reconstruct a compact SfM point cloud. A more straightforward way is to directly simplify an SfM point cloud. The SfM point cloud simplification problem can be treated as a K-Cover problem [LSH10]. Li *et al.* propose a greedy heuristic algorithm to iteratively select points that are visible for a maximum number of uncovered database images [LSH10]. Ensuring that each database image observes at least K points inherently ensures that a random query image from the underlying scene

is likely to obtain sufficient 2D-3D matches for camera pose estimation. Park *et al.* [PWN+13] solve the point cloud simplification problem using mixed-integer quadratic programming techniques. 3D points with high visibility are encouraged by a weight term, while the points that are frequently co-visible in the same image are penalized. Cao *et al.* [CS14] further combine the feature distinctiveness into a greedy heuristic algorithm to improve the accuracy of feature matching with the simplified point cloud. In addition, they introduce a probabilistic model which regards point-image relation as a random event.

2.8.2 Match Disambiguation Techniques

There can be more than tens of millions of feature descriptors in a large-scale SfM point cloud. In order to accelerate the feature matching step, Li *et al.* employ a prioritized 3D-to-2D matching strategy in which 3D points observed by large number of database images are considered first [LSH10]. Based on a compact visual vocabulary, Sattler *et al.* propose a fast feature matching algorithm by prioritizing visual words with few quantized descriptors [SLK11]. Choudhary *et al.* explore the bipartite visibility graph in an SfM point cloud to guide the feature matching process [CN12]. After finding a confident seed match, 3D points that are frequently co-visible with the 3D point of the seed match are prioritized due to high possibilities to be observed in the underlying query image. Instead of using the visibility graph, Sattler *et al.* prioritize points that lie close in 3D space to a point of highly confident match [SLK12]. Typically, the above methods terminate the feature matching step if sufficient number of matches are established. Though significantly improving the computational efficiency, these methods still rely on a strict SIFT ratio test to disambiguate matches, which may easily rejects correct matches in large-scale scenario.

Recent works [LSHF12, ZSP15, SHR+15, CSC+17] attempt to relax the SIFT ratio test in order to preserve more correct matches. To deal with the resultant matches of very large outlier ratio, Li *et al.* integrate the co-visibility priors into RANSAC to avoid computation on samples in which points are impossible to be observed in one image [LSHF12]. To deal with very large-scale image-based localization, *i.e.* the San Francisco dataset [LSHF12], Sattler *et al.* use an vocabulary containing 16 millions of visual words to perform implicit feature matching [SHR+15]. The visibility relationship among 3D points are utilized to find relevant database images, thereby disambiguating matches in image-level. The geometrical cues, either intrinsic or additional, are also

explored in match disambiguation [ZSP15, CSC+17, SEKO17]. Svärm *et al.* assume that the camera's vertical direction and approximate height relative to the SfM point cloud were known in advance. Based on this assumption, they propose an outlier filter by formulating a 2D registration problem. Concretely, matches are rejected based on the maximum number of geometrically consistent candidates [SEKO17]. Similarly, Zeisl *et al.* [ZSP15] use the same camera model assumption and derive a linear camera pose voting algorithm. To improve the efficiency of camera pose voting, they pre-filter obvious wrong matches using local feature constraints such as scale and orientation. Camposeco *et al.* [CSC+17] introduce a novel pose solver using intrinsic angle constraints of SfM point clouds. The camera position can be quickly estimated using two matches with this pose solver. However, all camera position hypotheses should be computed, *i.e.* 10^8 camera position hypotheses for 10^4 matches, to remove outliers. In addition, the final outlier filter requires a set of high quality matches to statistically remove outliers.

2.8.3 Localization under Binary Feature Representation

Comparing with SfM point cloud simplification methods, using a compact feature representation offers an orthogonal way to reduce the memory consumption. Feng *et al.* replace the high-dimensional SIFT feature descriptor with a binary feature descriptor (BRISK [LCS11]) to compress the SfM point cloud [FFW16]. In order to efficiently index binary feature descriptors, they propose a supervised random tree construction method by exploiting the visibility graph. Tran *et al.* [TLTD+19] map the SIFT feature descriptor to a short binary vector using a hash function of iterative quantization [GLGP13]. A corresponding cascade search pipeline is presented in which the dimension of searching vector is adjusted for speeding up the feature matching process. Even though both methods provide superior efficiency and compactness, the reported localization performances are worse than other methods using high-dimensional SIFT feature descriptors. Liu *et al.* [LLD17] convert the SIFT feature descriptor to a short binary signature via Hamming Embedding [JDS08] in visual words. Unlike previous methods that perform match disambiguation on individual match, they propose a ranking algorithm by globally exploiting the visibility information in a Markov random field. The top ranked matches are finally disambiguated by traditional SIFT ratio test to obtain one-to-one 2D-3D match for camera pose estimation. The experimental results show that the global ranking algorithm is beneficial to preserve correct matches

and to achieve satisfactory localization accuracy. However, high-dimensional SIFT feature descriptors still need to be stored, making the memory consumption problem unsolved.

2.8.4 Learning-based Localization

Recent advancement in deep learning techniques has made it possible to process general 3D point clouds for reconstruction [YFST18], semantic reasoning [SJS+18], and learning discriminative 3D descriptors [DBI18]. In the context of image-based localization, deep learning techniques have been leveraged either in a specific stage or to replace the full localization pipeline in a end-to-end manner.

Unlike traditional hand-crafted local feature descriptors (*e.g.* SIFT), learning-based feature representations are shown to be more robust under extreme illumination or geometry changes. Schönberger *et al.* present a generative descriptor learning method by understanding both 3D geometry and semantic information of a scene [SPGS18]. The learned 3D feature representation on 3D semantic voxel volumes is able to handle localization under large viewpoint changes. In traditional hand-crafted feature extraction methods, the feature detector only considers a small area around interest points. This makes feature extraction unreliable under extreme illumination changes. In contrary, Dusmanu *et al.* propose a trainable convolutional neural network (CNN) that simultaneously detect and describe in relatively large image regions [DRP+19]. These learning-based feature descriptors can easily be integrated into the image-based localization methods and frameworks proposed in this thesis.

Deep learning has also been explored for match disambiguation. Brachmann *et al.* propose DSAC which includes two differentiable schemes so that the hypothesis-and-verification strategy in RANSAC becomes trainable in deep learning architectures [BKN+17]. DSAC is designed specifically for 2D-3D matches, making it suitable as a counterpart of RANSAC in image-based localization tasks. Yet, its performance does not significantly outperform RANSAC. Yi *et al.* train an end-to-end neural network that disambiguates each match separately while embedding global information in the meantime [YTFO+18]. Since this neural network relies on a differentiable way to compute the essential matrix for image pairs, it is difficult to be generalized for camera pose estimation in image-based localization.

Deep learning can also be used to train an end-to-end image-based localization pipeline [KGC15, WHLT+17, KC+17]. In such methods, features in a query image

usually are densely extracted by very deep neural networks (*e.g.* VGG [SZ14] or ResNet [HZRS16]). Based on the dense image representation, existing methods use deep learning to directly regress a 6-DOF camera pose from an image. Kendall *et al.* propose a loss function that combines the position and orientation errors. The training data is automatically acquired from SfM point clouds. Walch *et al.* improve the performance of camera pose regression by integrating Long-Short Term Memory (LSTM) units with CNN [WHLT+17]. To better model the scene with CNN, Kendall *et al.* propose several novel loss functions based on re-projection error and scene geometry [KC+17]. However, these learning-based approaches are still less accurate than traditional image-based localization approaches.

2.8.5 Summary

My research mainly falls into the category of image-based localization using SfM point clouds. Though tremendous progress has been made by previous works, there still exist many problems to make image-based localization more effective, robust and accurate.

In the area of SfM point cloud simplification, previous works [LSH10, PWN+13, CS14] based on the K-Cover algorithm suffer from a common problem: due to inherent differences among datasets, it is difficult to use the same parameter setting to generate a simplified SfM point cloud with an acceptable localization performance. As such, to improve the robustness of SfM point cloud simplification methods, an approach that is able to predict the key parameter K for different datasets is needed. Regardless of the sources and algorithms for reconstructing the SfM point clouds, an invariant requirement for successful image-based localization is that sufficient number of correct 2D-3D matches should be established. The main idea of my research for SfM point cloud simplification is to build a model between the parameter K and the localization performance, in which the number of correct matches can serve as an intermediate proxy. Such a model not only can be used for parameter prediction, but also provides a data-driven description for the underlying SfM point cloud. In order to make image-based localization better deployed in devices with limited memory resources, my research also aims to use the built model to improve the effectiveness of SfM point cloud simplification.

One popular application of image-based localization is to navigate users in large-scale urban scenes. The major concern for this application is to disambiguate 2D-3D matches which may have a large proportion of outliers due to massive descriptors and

repetitive patterns. Though the state-of-the-art approaches [ZSP15, SEKO17] has been shown to effectively handle noisy matches, two problems prevent these approaches from being widely-used. The first problem is that they both heavily rely on an additional assumption that the vertical direction and approximate height of camera are known in advance. However, such an assumption requires additional sensors, *e.g.* Inertial Measurement Unit, which may not be available especially for historical images. The second problem is that the match disambiguation process using the additional assumption is very time consuming, making these approaches impractical. My motivation is based on a ubiquitous but unused intrinsic knowledge: in SfM point clouds depicting urban environments, the majority of 3D points can only be observed by nearby cameras due to strong occlusions. My research goal for urban image-based localization therefore is to efficiently and effectively leverage the intrinsic knowledge to disambiguate matches of large outlier ratios.

As an orthogonal strategy of simplification to address the memory consumption problem of large SfM point clouds, recent image-based localization approaches [TLTD+18, FFW16] using binary feature descriptors greatly reduce the localization accuracy and effectiveness. On top of a traditional binary feature representation [JDS08], my first research goal is to derive a method that can improve the feature-wise match disambiguation. Furthermore, the feature (*e.g.* visual similarity), visibility (*e.g.* point-image relationship), and geometry (*e.g.* camera pose) information in image-based localization leads to two interesting questions: How to unify them so that each can play its proper role, *i.e.* use its discriminative power to a tee? When is the appropriate phase to engage one specific information in a localization pipeline? Consequently, my second research goal is to design a solid and complete match disambiguation pipeline for matches under binary feature representation. The accuracy is also a key issue for image-based localization especially in autonomous driving applications. As such, my third research goal is to prevent degenerate pose hypotheses, which may cause localization drift, from being sampled or selected.

Chapter 3

Data-driven SfM Point Cloud Simplification

In this chapter, we consider using the SfM point cloud simplification method to address the memory consumption problem of large-scale SfM point clouds. We present a data-driven SfM point cloud simplification framework [CLZ+16][1] by taking it as a weighted K-Cover problem, which mainly includes two complementary parts. First, an utility-based parameter determination method is proposed to select a reasonable parameter K for K-Cover based simplification methods by evaluating the potential of a SfM point cloud for establishing sufficient 2D-3D feature correspondences. Second, we formulate the SfM point cloud simplification problem as a weighted K-Cover problem, and propose an adaptive exponential weight function based on the visibility probability of 3D points. The rest of this chapter is organized as following. Section 3.1 gives the preliminary knowledge that is related to our proposed framework. The parameter prediction approach is given in Section 3.2. Section 3.3 presents the proposed adaptive exponential weighted scheme. The experimental results are given in Section 3.4.

[1] **Wentao Cheng**, Weisi Lin, Xinfeng Zhang, Michael Goesele, Ming-Ting Sun, A Data-Driven Point Cloud Simplification Framework for City-Scale Image-Based Localization. IEEE Transaction on Image Processing, 26(1): 262-275, 2016.

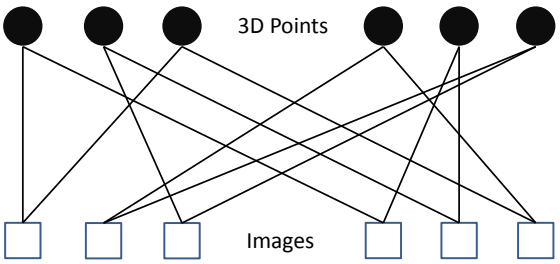


FIGURE 3.1: A bipartite graph representing the relations between 3D points and images used for reconstruction

3.1 Preliminaries

3.1.1 K-Cover Algorithm

We first give a brief description about the structure of SfM point clouds used in our work. The SfM point cloud can be reconstructed via a typical structure-from-motion pipeline [SSS06]. Such a SfM point cloud mainly contains two types of information: the information of database images and the information of 3D points. The 2D feature descriptors that are used for triangulating 3D points are also stored in a SfM point cloud. The relationship between database images and 3D points can be represented as a bipartite visibility graph $\mathcal{G} = (\mathcal{I}, \mathcal{P}, \mathcal{E})$, where nodes are $\mathcal{I}$ of size m and $\mathcal{P}$ of size n, each $I_i \in \mathcal{I}$ represents a database image and $P_j \in \mathcal{P}$ represents a 3D point. An edge $E_{i,j} \in \mathcal{E}$ exists if the 3D point P_j is visible in database image I_i. Fig. 3.1 illustrates an example of the bipartite visibility graph.

Based on the bipartite visibility graph, the K-Cover algorithm [LSH10] can be adopted for SfM point cloud simplification. The K-Cover algorithm query images and database images in a scene have a similar spatial distribution. Based on this assumption, Li *et al.* formulate the simplification problem as selecting a minimum subset of 3D points so that each database image can find at least K 2D-3D correspondences with the subset (each database image is covered at least K times) [LSH10]. Formally, the K-Cover algorithm can be represented as following:

$$\begin{aligned} &minimize \quad \Sigma_{j=1}^{n} S_j \\ &subject\ to \begin{cases} S_j \in \{0,1\} & j = 1, ..., n, \\ \Sigma_{j=1}^{n} E_{ij} S_j \geq K & i = 1, ..., m. \end{cases} \end{aligned} \tag{3.1}$$

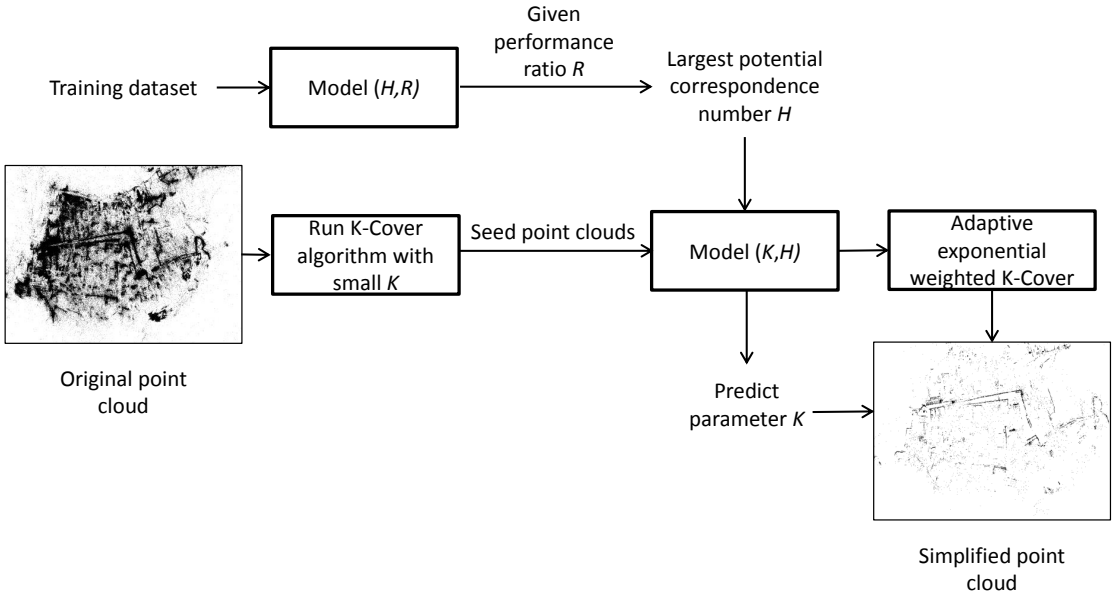


FIGURE 3.2: The proposed SfM point cloud simplification framework. H represents the largest number of correspondence that a query image from a random view in the underlying scene can robustly establish with respect to a SfM point cloud. K represents the parameter used in K-Cover based SfM point cloud simplification approaches. R represents the performance ratio provided by users.

where S_j is a binary variable which equals to 1 when a 3D point P_j is selected, otherwise 0. If point P_j is visible in image I_i, $E_{ij} = 1$, otherwise $E_{ij} = 0$.

The K-Cover problem is proven to be a combinatorial NP-hard problem [Fei98]. Alternatively, one can use a greedy heuristic method to find an approximate solution. In each iteration, the greedy heuristic method will select the 3D point that is visible in the largest number of uncovered database images. Here "uncovered" means that the corresponding database image has observed less than K 3D points with respect to the simplified SfM point cloud. Formally, the 3D point which has the largest visibility value $kc(P_j)$ will be selected in each iteration. $kc(P_j)$ can be computed as following:

$$kc(P_j) = \underset{j \in \mathcal{P} \setminus \mathcal{X}}{\operatorname{argmax}} \sum_{I_i \in \mathcal{I} \setminus \mathcal{Y}} E_{ij}. \tag{3.2}$$

where $\mathcal{X}$ represents the points that have already been selected, and $\mathcal{Y}$ is the set of database images which have been covered at least K times in the current iteration.

3.1.2 Overview

Fig. 3.2 illustrates the proposed SfM point cloud simplification framework. Given an original SfM point cloud, a set of seed SfM point clouds are generated by running the K-Cover algorithm using very small K values. In practice, this procedure requires a small amount of time. For each seed SfM point cloud, we use the *Poisson Binomial Distribution* to estimate its largest number of 2D-3D correspondences that a query image from random view can robustly establish, which is denoted as H. Based on a key observation that μ, which is used to compute H, increases linearly with K, we obtain a model of K and H. With such a model, a threshold can be estimated to determine the compactness of a SfM point cloud and when to invoke the adaptive exponential weighted K-Cover approach. We run an image-based localization approach on a training dataset to obtain another model between H and performance ratio R. By combining these two models, the parameter K can be fast predicted for a provided performance ratio R. In the following sections, we will describe the proposed SfM point cloud simplification framework in detail.

3.2 Predicting Parameter K

3.2.1 Motivation

Previous works based on the K-Cover algorithm all require users to manually choose a K value, either for obtaining a very compact SfM point cloud or for getting an acceptable localization performance using the simplified SfM point cloud. In practice, these two goals are hard to be achieved simultaneously with a manually chosen K. To get a simplified SfM point cloud with both acceptable localization performance and minimal number of points, the users have to test several simplified SfM point clouds generated by using multiple K values. However, due to different characteristics in various datasets, *e.g.* the density and spatial distribution of database images, using the same K value may result in large localization performance variations on different datasets. To evaluate the localization performance variations, we use the performance ratio R as the measurement, which can be represented as

$$R = \frac{SP}{OP}, \tag{3.3}$$

TABLE 3.1: An example to illustrate that using the same K achieves significantly different localization performance ratios on different datasets.

K	20	30	50
Dubrovnik	69.42%	85.14%	93.74%
Rome	88.72%	92.60%	95.70%
Aachen	9.46%	26.50%	53.62%

where SP represents the localization performance using simplified SfM point clouds, and OP stands for the localization performance using original SfM point clouds. To measure the localization performance, we adopt an evaluation protocol widely-used in previous image-based localization approaches [LSH10, LSHF12, SLK11, CSC+17, SEKO17, LLD17]. With the same query image set, the localization performance is measured as the number of successfully localized images whose estimated camera poses are supported by at least 12 inliers. Table 3.1 shows the performance ratio comparison on three datasets using the same localization method [SLK12]. For example, by setting $K = 30$, we obtain a 85.14% performance ratio on the Dubrovnik dataset, 92.60% on the Rome dataset and 26.50% on the Aachen dataset. A question naturally arises: Is it possible to predict the parameter K used in K-Cover based approaches for a given performance ratio?

In order to bridge the performance ratio R and the parameter K, we use an intermediate variable H which represents the largest number of correct 2D-3D correspondences that a query image from random view can robustly establish. If the value of H of a SfM point cloud increases, the possibility that a query image in the underlying scene can be successfully localized should also increase. Based on this fact, we propose to build the first model between R and H. The K-Cover based approaches rely on a common assumption that the spatial distribution of the query images is similar with the database images. However, ensuring that each database image covers at least K points does not guarantee that a random query image can also robustly establish at least K correct 2D-3D correspondences. Therefore, we propose to build the second model between K and H by taking the intrinsic characteristics of SfM point clouds into consideration.

3.2.2 Single Point Visibility Probability

3.2.2.1 Standard Definition

We first define $V(P_j)$ as the visibility probability of point P_j. The visibility probability measures the chance that a point is visible from a random view in the underlying scene. Assuming that 3D SfM point clouds are geometrically dense to describe the scene, a single point's visibility probability can be approximated by the following:

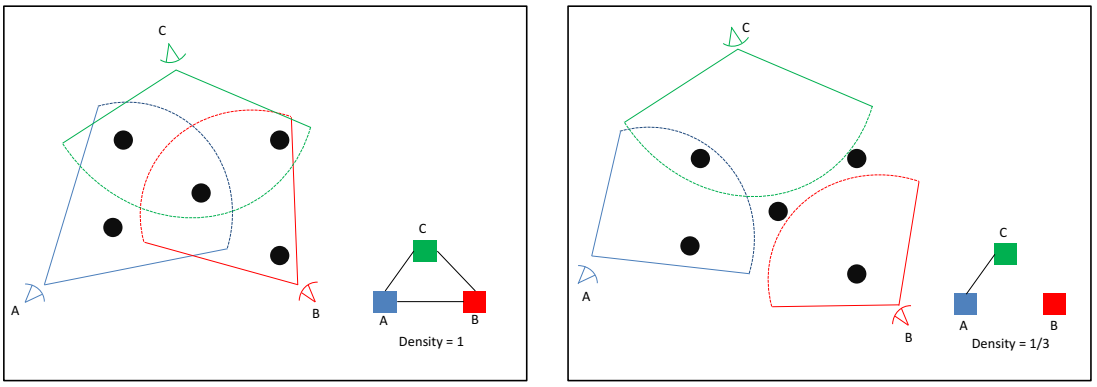
$$\phi(P_j) = \frac{d(P_j)}{m} \tag{3.4}$$

where $d(P_j)$ is the degree of point P_j in the bipartite graph $\mathcal{G}$, and m is the number of all database images used for reconstructing SfM point clouds.

3.2.2.2 Density Estimation Based on Graph

Assuming that the database images are densely distributed, $\phi(P_j)$ is suitable to approximate the true visibility probability of 3D points. In real cases it is, however, possible that the dataset does not meet this assumption. To accurately compute a point's true visibility probability, the density of a SfM point cloud should also be considered. A straightforward way to estimate the density of a SfM point cloud is to count how many database images or points fall into one cubic meter on average. But not all SfM point clouds are of meaningful geometrical setting, making it difficult to compare among different datasets. In this work, we propose to handle the density estimation problem using an image overlapping graph. Considering a certain number of database images, if they have a denser geometrical distribution as shown in Fig. 3.3A, the overlapping areas between them will be larger than database images have a sparser geometrical distribution as shown in Fig. 3.3B. If we construct a graph to measure the overlap between database images, such a graph will be denser when database images are denser geometrically.

We construct an image overlapping graph $\mathcal{O}$ based on the bipartite graph $\mathcal{G}$. Each node in $\mathcal{O}$ represents a database image. Therefore, the number of nodes in $\mathcal{O}$ is the same as m defined in Section 3.1. For two database images a and b, if they both observe the same 3D point, we link them in the graph $\mathcal{O}$. In addition, edges with less geometrical consistency should be removed from the image overlapping graph. To this end, we only preserve edges with either $\frac{N(a,b)}{N(a)} > 0.1$ or $\frac{N(a,b)}{N(b)} > 0.1$, where $N(a,b)$



(A) **Dense Case** (B) **Sparse Case**

FIGURE 3.3: An illustration of different geometrical densities using the same database images (a) when images have a dense geometrical distribution. The graph density is 1 in this example, (b) when images have a sparse geometrical distribution. The graph density is 1/3 in this example.

is the number of 3D points observed in both database image a and b, and $N(a)$ and $N(b)$ are the number of 3D points observed in database image a and b respectively. To measure the density of $\mathcal{O}$, we use the standard density estimation method in graph theory [Sco12]. The graph density can be computed as following:

$$D = \frac{2e}{m(m-1)}, \tag{3.5}$$

where e is the number of edges in graph $\mathcal{O}$, and m is the number of nodes in graph $\mathcal{O}$. The visibility probability $V(P_j)$ measures the chance that point P_j is visible from a random view in the underlying scene. $\phi(P_j)$ can be regarded as an appropriate interpretation of $V(P_j)$ only if the corresponding SfM point cloud is dense. Suppose a scene is sparsely reconstructed from a small amount of database images, the number of database images in which a single 3D point is visible will account for a large portion of the overall database images. As additional database images are added, it is possible that one added database image only contributes to triangulate a small amount of points. For other points i.e. P_j that are not visible in this added database image, $d(P_j)$ remains unchanged while the number of the overall database images m increases. $\phi(P_j)$ will become smaller on average with the increase in the density of a SfM point cloud. Since the graph density of $\mathcal{O}$ has a positive correlation with the density of the point cloud, we define a nonlinear weight function $f(D)$ to down-weight $\phi(P_j)$ in a low

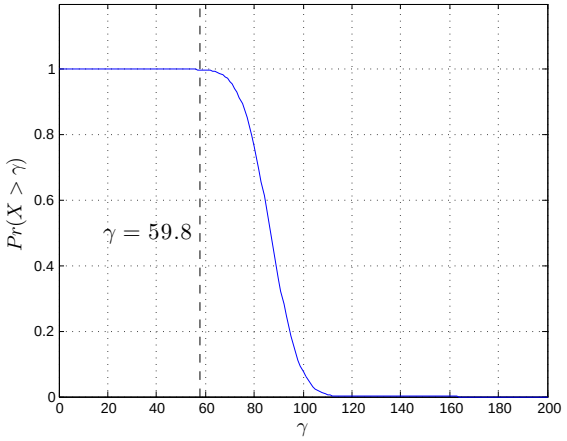


FIGURE 3.4: An example of evaluating a simplified SfM point cloud using the *Poisson Binomial* distribution. The simplified SfM point cloud is generated with $K = 30$ using the basic K-Cover algorithm on the Rome dataset. Based on the evaluation, a random query image in the underlying scene is able to establish $\gamma = 59.8$ 2D-3D correspondences with $Pr(X > \gamma) = 1$ with this point cloud.

graph density case in order to approximate $V(P_j)$ as following:

$$V(P_j) = f(D)\phi(P_j), \tag{3.6}$$

where $f(D) = (1-6.72e^{-213D})$. We use the original well-reconstructed SfM point cloud in the Dubrovnik dataset [LSH10], and compute the average visibility probability as the true visibility probability. Meanwhile, we use the basic K-Cover algorithm to generate several simplified point clouds, which have lower graph densities compared with the original SfM point cloud. The constants in the above equation therefore are obtained using nonlinear fitting. For a SfM point cloud merged from multiple individual models, the density of the whole SfM point cloud can be computed as a weighted average density of all individual point clouds based on the number of database images in each individual model.

3.2.3 Evaluate as Poisson Binomial Distribution

Let P be a SfM point cloud of size τ, and $V(P_j)$ be the visibility probability of each 3D point. The correlation between 3D points is usually adopted to guide the feature matching [SLK12, CN12]. For example, if a potentially correct 2D-3D match is established, the neighboring 3D points can be prioritized. However, the visibility probability is based on the relationship between 3D points and a random view. In addition, in

our work, we use the standard feature matching scheme in which each 3D point is independently matched with the query descriptors. Therefore, we assume that for each point the visibility probability is independent with the existence of other points, we define $X_j, j = 1, ..., \tau$ be a series of random Bernoulli trial as

$$X_j \sim Bernoulli(V(P_j)), j = 1, ..., \tau \tag{3.7}$$

where $V(P_j) = Pr(X_j = 1)$ is the success probability that 3D point P_j is visible in a random view. The *Poisson Binomial* random variable X is defined as the sum of independent and non-identical trials:

$$X = \sum_{j=1}^{\tau} X_j. \tag{3.8}$$

Thus the probability that γ points are visible in a random view is

$$Pr(X = \gamma) = \sum_{A \in \mathbb{F}_\gamma} \prod_{j \in A} V(P_j) \prod_{j \in A_c} (1 - V(P_j)) \tag{3.9}$$

where $\mathbb{F}_\gamma$ is the set of all subsets of γ integers which can be selected from $\{1, 2, 3, ..., \tau\}$. Suppose A is one set in $\mathbb{F}_\gamma$, and A_c is the complement of A. The cumulative distribution function (cdf) of the *Poisson Binomial* distribution is

$$Pr(X \leq \gamma) = \sum_{l=0}^{\gamma} (\sum_{A \in \mathbb{F}_l} \prod_{j \in A} V(P_j) \prod_{j \in A_c} (1 - V(P_j)). \tag{3.10}$$

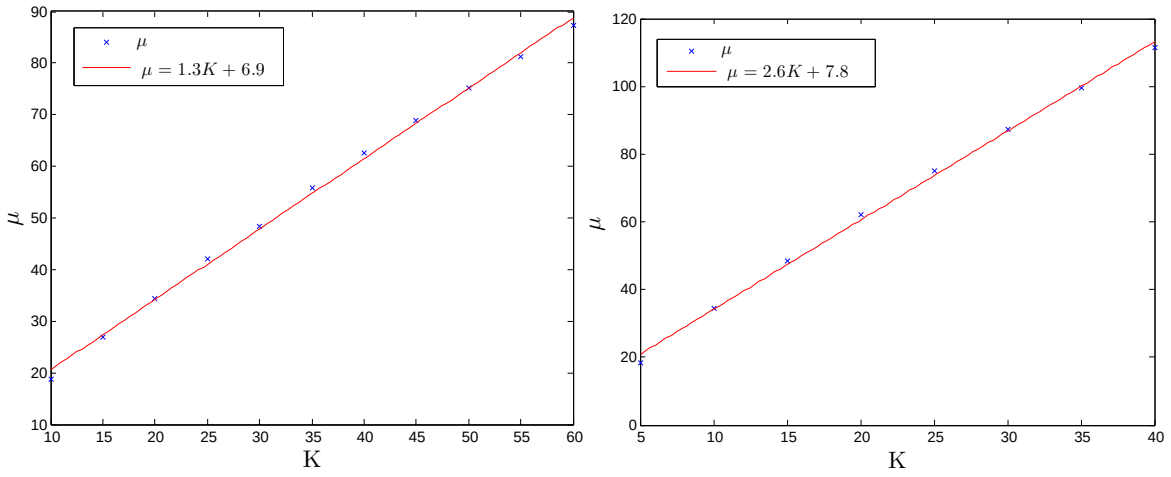
Thus the probability that more than γ points are visible in a random view is

$$Pr(X > \gamma) = 1 - \sum_{l=0}^{\gamma} (\sum_{A \in \mathbb{F}_l} \prod_{j \in A} V(P_j) \prod_{j \in A_c} (1 - V(P_j)). \tag{3.11}$$

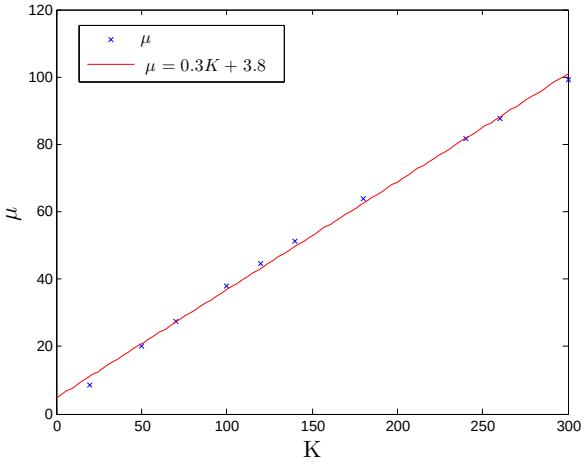
Based on the central limit theorem [AG80], the *Poisson Binomial* distribution can be regarded as the normal approximation (NA) [Hon11]. The NA method approximates the CDF of a *Poisson Binomial* distribution by

$$Pr(X \leq \gamma) \approx \Phi(\frac{\gamma + 0.5 - \mu}{\sigma}), \gamma = 1, ..., \tau \tag{3.12}$$

$$Pr(X > \gamma) \approx 1 - \Phi(\frac{\gamma + 0.5 - \mu}{\sigma}), \gamma = 1, ..., \tau \tag{3.13}$$



(A) **Dubrovnik** (B) **Rome**



(C) **Aachen**

FIGURE 3.5: The relation between K and μ

where the expectation can be represented as

$$\mu = E(X) = \sum_{j=1}^{\tau} V(P_j) \tag{3.14}$$

and the standard deviation is

$$\sigma = [Var(X)]^{1/2} = [\sum_{j=1}^{\tau} V(P_j)(1 - V(P_j))]^{1/2}. \tag{3.15}$$

Algorithm 1: Predicting Parameter K

Require: Original SfM point cloud P. Provided performance ratio R_e

1: run K-Cover on a training dataset, the results can be used as training data to obtain a model between H and R
2: predict H_e for R_e
3: generate seed point clouds using a set of very small K
4: build the model between K and H
5: predict K_e for H_e
6: **return** predicted parameter K_e

3.2.4 Efficient Prediction

Fig. 3.4 shows an example of evaluating a simplified SfM point cloud on the Rome dataset. $Pr(X > \gamma) = 1$ means that a query image from a random view in the underlying scene can robustly establish γ 2D-3D correspondences with respect to the simplified SfM point cloud. Concretely, H can be computed as following:

$$H = \underset{\gamma}{\operatorname{argmax}} \left(Pr(X > \gamma) = 1\right). \tag{3.16}$$

Traditionally, computing H for a simplified SfM point cloud required to run specific K-cover based simplification approaches. However, this could be time consuming when the used parameter K is large. We propose an efficient evaluation method based on a key observation that μ in Eq. 3.13 increases linearly with K as shown in Fig. 3.5. In addition, since $V(P_j) \ll 1$ in large-scale datasets, we find that using $\sigma = (\mu)^{1/2}$ to replace Eq. 3.15 does not affect the evaluation in practice. In order to derive the linear model parameters between K and μ, we first generate several seed SfM point clouds using a set of small K values and then fit them with a linear model. Thus given a parameter K, μ and H can be fast computed without running simplification approaches. It is difficult to model the relationship between potential correspondences number H and performance ratio R, since the final performance ratio may also depends on the reliability of feature matching approaches. In this work, we make a relaxation that most of possible 2D-3D correspondences can be found by powerful feature matching methods such as ANN [AMN+98] and FLANN [ML09], and the camera pose can be robustly estimated with found correspondences. Without losing generality, we use the localization results on the Dubrovnik dataset using the basic K-Cover algorithm as the training data to obtain a model between H and R.

3.3 Adaptive Exponential Weighted K-Cover

3.3.1 Analysis

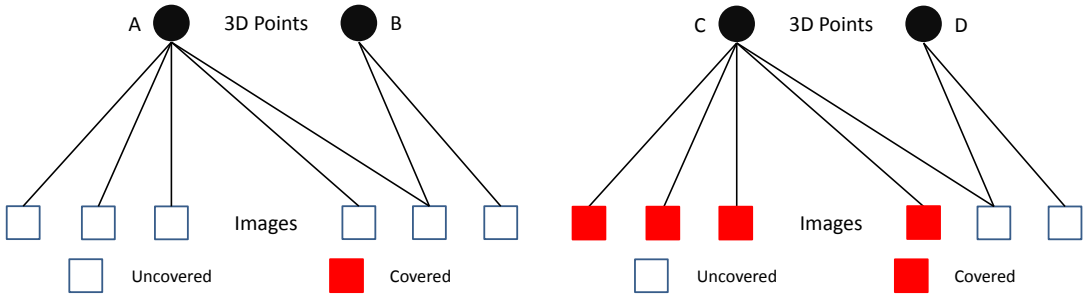
We begin by analyzing the greedy heuristic selection process of the K-Cover algorithm. For better illustration we generally divide the whole process into two phases as following:

–**Early phase**: when most database images have not yet been covered at least K times.

–**Late phase**: when most database images have already been covered at least K times.

A key observation is that K-Cover will adopt some "doubtful" selections in specific phase. As shown in Fig. 3.6A, point A is visible in five database images that have not yet been covered. Thus in the **Early Phase** K-Cover will select point A but not point B since point B is only visible in two uncovered database images. However, in Fig. 3.6B, even though point C is visible in more database images than point D, there is only one uncovered database image in the visible list of point C. In the **Late Phase** K-Cover will select point D which is visible in two uncovered database images and point C will be discarded. When experimenting on city-scale datasets, points that are visible in hundreds of database images may be discarded while points that are visible in only few database images may be selected instead. However, a 3D point's visibility probability is also an important factor that should be considered in image-based localization tasks since highly visible points will have high probability to be captured by a random view, or/and they may appear in a region that is more likely to be photographed, *e.g.* landmarks.

To recap, the goal of computing a subset is to ensure that the information of original SfM point cloud can be preserved as much as possible for image-based localization tasks. When K is large enough, the cardinality of the subset is also large enough to describe the scene. However, when facing devices with limited memory resources, a small K value is needed to locally store the simplified SfM point cloud. In such case, points with high visibility may not be fully selected to describe important regions, *e.g.* landmarks. If we simply select points with the highest visibility probability, the subset will exhibit an extremely uneven spatial distribution which will significantly decrease the image-based localization performance. In other words, this simple selection scheme cannot ensure the coverage of database images. As such, a selection scheme is needed



(A) **Early Phase** (B) **Late Phase**

FIGURE 3.6: An illustration of different phases in the greedy selection process of K-Cover algorithm. (a) In the **Early Phase** K-Cover selects point with high visibility probability. (b) In the **Late Phase** K-Cover selects point with low visibility probability.

to balance two key factors: the coverage of database images and the selection of points with high visibility probability.

3.3.2 Adaptive Exponential Weighted K-Cover

Based on the analysis of the K-Cover algorithm, we propose a weighted K-Cover algorithm as following:

$$wkc(P_j) = w(V(P_j), K)kc(P_j) \tag{3.17}$$

where $w(V(P_j), K)$ is a weighted term related to the 3D points' visibility probability and the parameter K. $kc(P_j)$ is the coverage term used in the K-Cover algorithm (see Eq. 3.2).

Defining a suitable $w(V(P_j), K)$ is very important for achieving our goal of balancing between the coverage of database images and the selection of points with high visibility probability. In general it should obey the following two rules:

- it should be monotonous with $V(P_j)$ which ensures that points with high visibility probability are more likely to be chosen.
- it should reflect more emphasis on the selection of points with high visibility probability in cases with extremely low memory resources.

For the first rule, we propose a weighted term with a constant exponent [CLS15] as following:

$$w(V(P_j), K) = (V(P_j))^{1.5}, \tag{3.18}$$

where the constant 1.5 is set empirically. In practice, we find that this setting works well in maintaining the balance. Other schemes, *e.g.* linear weighting, can be exploited in the future.

In order to obey the second rule, the weighted term should take the value of parameter K into consideration. In general, small K values should be adopt to deal with extremely low memory resources. However, due to different intrinsic characteristics of SfM point clouds, it is difficult to set a common threshold for K to determine when to put more emphasis on points of high visibility probability. To recap, the intermediate variable H proposed in Section. 3.2 can be used to fairly model among different SfM point clouds. For devices with limited memory resources, the SfM point cloud should be intensively simplified, and a relatively small number of potentially correct 2D-3D correspondences can be robustly established between a query image and the simplified SfM point cloud. Suppose the threshold of H is set as H^s, an adaptive K' threshold can be computed using the model between K and H

$$K' = \{K | H_K = H^s\}\,. \tag{3.19}$$

In this work, we empirically set H^s to 40. Thus the adaptive exponential weight term can be described as

$$w(V(P_j), K) = (V(P_j))^{1.5(\frac{K'}{K})^{1.5}}. \tag{3.20}$$

3.3.3 Efficient Implementation

The running time of K-Cover algorithm is approximately $O(nsv)$, where n is the number of points in the original SfM point cloud, s is the size of selected subset and v is the average number of database images that observe 3D points. Inspired by Cao *et al.* [CS14], we also maintain an upper bound for each point so that we can skip a large portion of points. In Eq. 3.17 the first term $w(V(P_j), K)$ is fixed for each point with Eq. 3.20, and the second term is always less than the number of visible database images of each point since $kc(P_j)$ counts the number of uncovered database images from the visible image list. Therefore, the upper bound of each point can be computed as $w(V(P_j), K) \cdot d(P_j)$. In each iteration, points whose upper bound are below the current largest coverage will be discarded directly, which significantly saves the computational cost. Algorithm 1 shows the whole pipeline of the adaptive exponential weighted K-Cover algorithm.

Algorithm 2: Adaptive Exponential Weighted K-Cover

Require: Bipartite graph G: nodes are database images $\mathcal{I} = \{I_1, I_2, ...I_m\}$ and 3D points $\mathcal{P} = \{P_1, P_2, ..., P_n\}$

$\mathcal{Y} \leftarrow \emptyset$, $\bar{S} \leftarrow \emptyset$
Visibility probability $V = \{V(P_1), V(P_2), ..., V(P_n)\}$
Set each point unselected $U = \{u_1, u_2, ..., u_n\} = \{FALSE\}$
Upper bound $\chi = \{\chi_1, \chi_2, ..., \chi_n\}$
while $\mathcal{Y} \neq \mathcal{I}$ **do**
  $MAX = 0$
  for $j = 1$ to n **do**
    if $u_j = TRUE$ **then**
      if $\chi_j \geq MAX$ **then**
        compute the value of Eq. 3.20
      end if
      update MAX
    end if
  end for
  if I_i are covered more than K times **then**
    $\mathcal{Y} = \mathcal{Y} \bigcup \{I_i\}$
  end if
  $\bar{S} = \bar{S} \bigcup \{P_j\}$
  $u_j = TRUE$
end while
return $\bar{S}$

3.4 Experimental Evaluation

In this section, we first introduce the datasets and evaluation criteria used in the experiments. The proposed parameter prediction method for K-Cover based methods is evaluated on three real-world datasets. Last, we validate the proposed adaptive exponential weighted K-Cover algorithm by comparing with state-of-the-art SfM point cloud simplification approaches.

3.4.1 Datasets

Table 4.1 shows the statistics of three widely-used datasets used in our experiments. The Dubrovnik and Rome datasets are released in [LSH10], in which the database images are downloaded from *Flickr.com*. The database images in the Aachen dataset [SWLK12] are captured by a single DSLR in a deliberate way. The Dubrovnik dataset

TABLE 3.2: Summarization of three city-scale datasets

Dataset	Database Images	3D Points	Query Images
Dubrovnik	6,044	1,886,884	800
Rome	15,179	4,067,119	1000
Aachen	3,047	1,540,786	369

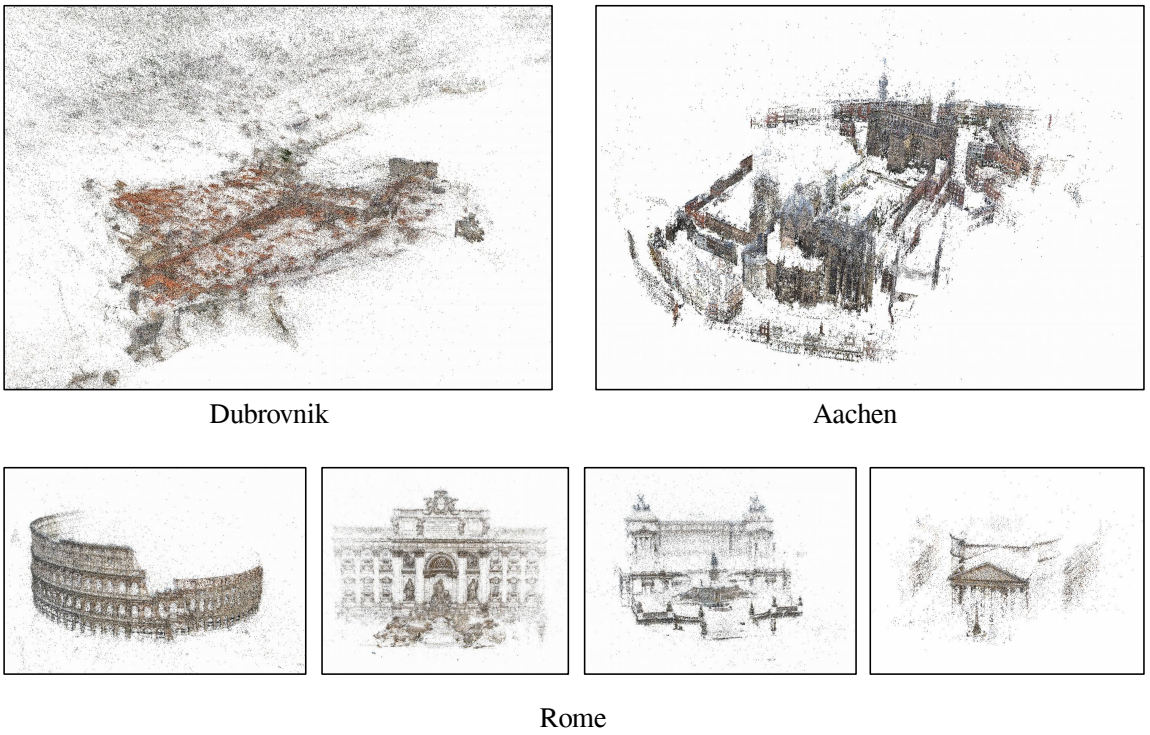


FIGURE 3.7: The visualization of SfM point clouds used in our experiments.

contains a full reconstruction that uses all database images and query images. Consequently, the camera poses of query images in the full reconstruction can be regarded as ground truth for comparison. As such, we mainly evaluate the localization accuracy on the Dubrovnik dataset. The SfM point cloud in the Rome dataset consists of several individual landmarks, *e.g.* the Trevi Fountain, the Colosseum, the Pantheon, etc, in Rome. The query images in the Aachen dataset are taken by a mobile phone over two years. This long-term capture makes image-based localization more challenging on this dataset. The corresponding SfM point clouds are visualized in Fig. 3.7.

3.4.2 Evaluation Criteria

3.4.2.1 Prediction Accuracy

Given a performance ratio, we first predict the parameter K and use this K to generate a simplified SfM point cloud. The prediction accuracy is measured by the error between provided performance ratio and real performance ratio using the resultant simplified SfM point cloud.

3.4.2.2 Simplification Performance

In order to evaluate the proposed adaptive exponential weighted K-Cover algorithm, we focus on the localization results using simplified SfM point clouds. The localization results can be evaluated using following measures:

–Localization Performance: we report the number of images that can be successfully registered by simplified SfM point clouds. We follow the evaluation protocol used in previous methods [LSH10, SLK11] that a query image can be regarded as successfully registered if the best camera pose estimated by perspective-n-point algorithms + RANSAC has at least 12 inliers. To evaluate the efficiency, we also report the average computational cost to register or reject one query image. In addition, the inlier ratios are also computed to measure the robustness of localization results.

–Localization Accuracy: The Dubrovnik dataset is the only dataset with meaningful geometry measured in meter in the experiments. To evaluate the localization accuracy, we compute the errors between the estimated camera poses and the ground truth in this dataset.

–Computational Cost: we report the average running time of each SfM point cloud simplification approach to compare their computational efficiency.

To the best of our knowledge, the state-of-the-art SfM point cloud simplification methods are based on the K-Cover algorithm. Therefore, we mainly compare our method with the following two K-Cover based approaches:

–KC. The basic K-Cover algorithm proposed in [LSH10].

–PKC. The probabilistic K-Cover proposed in [CS14]. The proposed weighted K-Cover algorithm using Eq. 3.18 is denoted as WKC, and the adaptive exponential weighted K-Cover using Eq. 3.20 is denoted as AEWKC.

3.4.3 Evaluation of Predicting K

Table 3.3 and Table 3.4 report the prediction results on existing K-Cover based SfM point cloud simplification approaches. For each approach, we pick ten K that can get a meaningful performance ratio, *e.g.* above 50%. The real performance ratio results are obtained by running image-based localization approaches on each simplified SfM point cloud for ten times. For each K, we report the potential correspondences number H, the predicted performance ratio, the real performance ratio and the error between the predicted and the real performance ratio. Note that the prediction experiment is not conducted on the Dubrovnik dataset with KC since it is used for training the model between H and R.

Table 3.3 shows the prediction results with the KC method. In general, our parameter prediction method is accurate. In particular, the prediction becomes more accurate when the predicted performance ratio is high, *e.g.* above 80%. For example, if users require a simplified SfM point cloud with 91.85% performance ratio of the Rome dataset, our prediction method can generate a point cloud that has 92.21% localization performance in practice. In the case that the error between the predicted performance ratio and the real performance ratio can be as high as more than 20%, the corresponding H is usually small which means that a query image may not get sufficient correspondences with respect to the simplified SfM point cloud for robust camera pose estimation. Insufficient correspondences increase the uncertainty of the camera pose estimation process. Insufficient correspondences also cause many localization results to lie near the inlier threshold. However, in practical scenarios, the common case is that users prefer to provide a high predicted performance ratio R to obtain an acceptable localization performance with the simplified SfM point cloud. In such cases, the proposed proposed prediction approach can achieve satisfactory results. Table 3.3 reports the prediction results with the AEWKC approach. The prediction achieves the highest accuracy on the Dubrovnik dataset. The accuracy on the other two datasets is generally below 5% when the predicted ratio is larger than 70%. Table 3.4 reports the prediction results on the PKC approach and the proposed WKC approach. The prediction is still accurate especially when users need a high performance ratio.

TABLE 3.3: The experimental results of predicting K with KC and our proposed AEWKC method (%). Note that the prediction experiment is not conducted on the Dubrovnik dataset with KC since it is used for training the model between H and R

KC					**AEWKC**				
Dubrovnik Dataset					*Dubrovnik Dataset*				
K	H	Predicted R	Real R	Error	K	H	Predicted R	Real R	Error
15	-	-	-	-	15	31.5	83.30	68.08	15.22
20	-	-	-	-	20	38.3	87.57	83.48	4.09
25	-	-	-	-	25	45.2	90.11	86.71	3.40
30	-	-	-	-	30	52.3	91.71	90.40	1.31
35	-	-	-	-	35	59.4	92.73	91.25	1.48
40	-	-	-	-	40	66.6	93.42	91.95	1.47
45	-	-	-	-	45	73.8	93.90	93.08	0.82
50	-	-	-	-	50	81.1	94.24	93.84	0.40
55	-	-	-	-	55	86.4	94.42	94.59	0.17
60	-	-	-	-	60	95.8	94.67	95.35	0.68
Rome Dataset					*Rome Dataset*				
K	H	Predicted R	Real R	Error	K	H	Predicted R	Real R	Error
10	15.6	57.37	72.40	15.03	10	35.7	86.21	79.85	6.36
12	19.5	66.61	78.32	11.71	12	39.9	88.28	83.17	5.11
14	23.5	74.01	82.15	8.14	14	44.2	89.82	86.03	3.79
15	25.6	77.07	84.27	7.20	15	46.4	90.44	87.21	3.23
18	31.8	83.55	87.27	3.72	18	52.9	91.82	88.69	3.13
20	35.9	86.32	88.72	2.40	20	57.2	92.46	89.70	2.76
22	40.2	88.40	89.48	1.08	22	61.6	93.04	90.50	2.54
25	46.6	90.49	91.10	0.61	25	68.3	93.55	91.78	1.77
28	53.1	91.85	92.21	0.36	28	75.0	93.96	92.45	1.51
30	57.5	92.50	92.60	0.10	30	79.5	94.17	93.17	1.00
Aachen Dataset					*Aachen Dataset*				
K	H	Predicted R	Real R	Error	K	H	Predicted R	Real R	Error
60	7.9	37.99	62.14	24.15	60	17.6	62.34	74.13	11.79
90	14.7	55.01	74.20	19.19	90	24.8	75.97	78.17	2.20
120	22.0	71.49	79.17	7.68	120	32.3	83.94	83.91	0.03
140	26.9	78.72	82.64	3.92	140	37.4	87.13	86.21	0.92
180	37.2	87.03	86.43	0.60	180	47.7	90.76	89.37	1.39
200	42.4	89.23	88.32	0.91	200	52.9	91.82	89.91	1.91
240	53.0	91.84	90.53	1.31	240	63.5	93.16	90.76	2.40
260	58.4	92.62	91.48	1.14	260	68.8	93.58	91.48	2.10
300	69.2	93.61	92.74	0.87	300	79.6	94.18	92.93	1.25
320	74.7	93.94	92.87	1.07	320	85.0	94.38	93.85	0.53

TABLE 3.4: The experimental results of predicting K with PKC and WKC method (%).

PKC					**WKC**				
Dubrovnik Dataset					*Dubrovnik Dataset*				
K	H	Predicted R	Real R	Error	K	H	Predicted R	Real R	Error
15	15.0	55.80	53.01	2.79	15	26.2	77.86	71.60	6.26
20	21.7	70.95	67.86	3.09	20	34.3	85.35	81.22	4.13
25	28.6	80.62	79.87	0.75	25	42.7	89.34	86.79	2.55
30	35.6	86.15	85.03	1.12	30	51.2	91.51	90.06	1.45
35	42.8	89.37	88.74	0.63	35	59.8	92.78	91.62	1.16
40	50.1	91.30	90.57	0.73	40	68.5	93.56	92.26	1.30
45	57.5	92.50	91.62	0.88	45	77.3	94.07	93.45	0.62
50	65.0	93.29	92.45	0.84	50	86.1	94.41	94.50	0.09
55	72.5	93.82	93.72	0.10	55	95.1	94.66	94.97	0.31
60	80.1	94.20	94.42	0.22	60	104.0	94.83	95.50	0.67
Rome Dataset					*Rome Dataset*				
K	H	Predicted R	Real R	Error	K	H	Predicted R	Real R	Error
10	15.6	57.37	73.36	15.99	10	28.4	80.41	78.66	1.75
12	19.7	67.04	79.06	12.02	12	33.9	85.08	82.64	2.44
14	23.8	74.48	81.88	7.40	14	39.4	88.07	84.02	4.05
15	25.9	77.47	84.32	6.85	15	42.2	89.16	84.40	4.76
18	32.3	83.94	87.10	3.16	18	50.7	91.42	88.10	3.32
20	36.6	86.71	88.69	1.98	20	56.5	92.37	89.70	2.67
22	41.0	88.72	89.37	0.65	22	62.3	93.04	90.80	2.24
25	47.6	90.74	90.75	0.01	25	71.0	93.73	91.61	2.12
28	54.3	92.05	91.96	0.09	28	79.9	94.19	92.28	1.91
30	58.8	92.66	92.76	0.10	30	85.8	94.40	92.91	1.49
Aachen Dataset					*Aachen Dataset*				
K	H	Predicted R	Real R	Error	K	H	Predicted R	Real R	Error
60	8.2	38.59	63.91	25.32	60	12.7	49.65	68.77	19.12
90	15.4	56.85	75.27	18.42	90	20.7	69.09	79.18	10.12
120	23.1	73.20	79.81	6.61	120	29.0	81.03	83.60	2.57
140	28.4	80.41	82.67	2.26	140	34.7	85.60	85.80	0.20
180	39.2	87.98	88.14	0.16	180	46.4	90.44	89.27	1.17
200	44.7	89.97	88.68	1.29	200	52.3	91.71	89.91	1.80
240	56.0	92.30	91.17	1.13	240	64.3	93.23	90.85	2.38
260	61.6	92.97	91.29	1.68	260	70.4	93.69	92.11	1.58
300	73.1	93.86	92.05	1.81	300	82.7	94.30	93.69	0.61
320	78.9	94.15	93.53	0.62	320	88.9	94.50	94.00	0.50

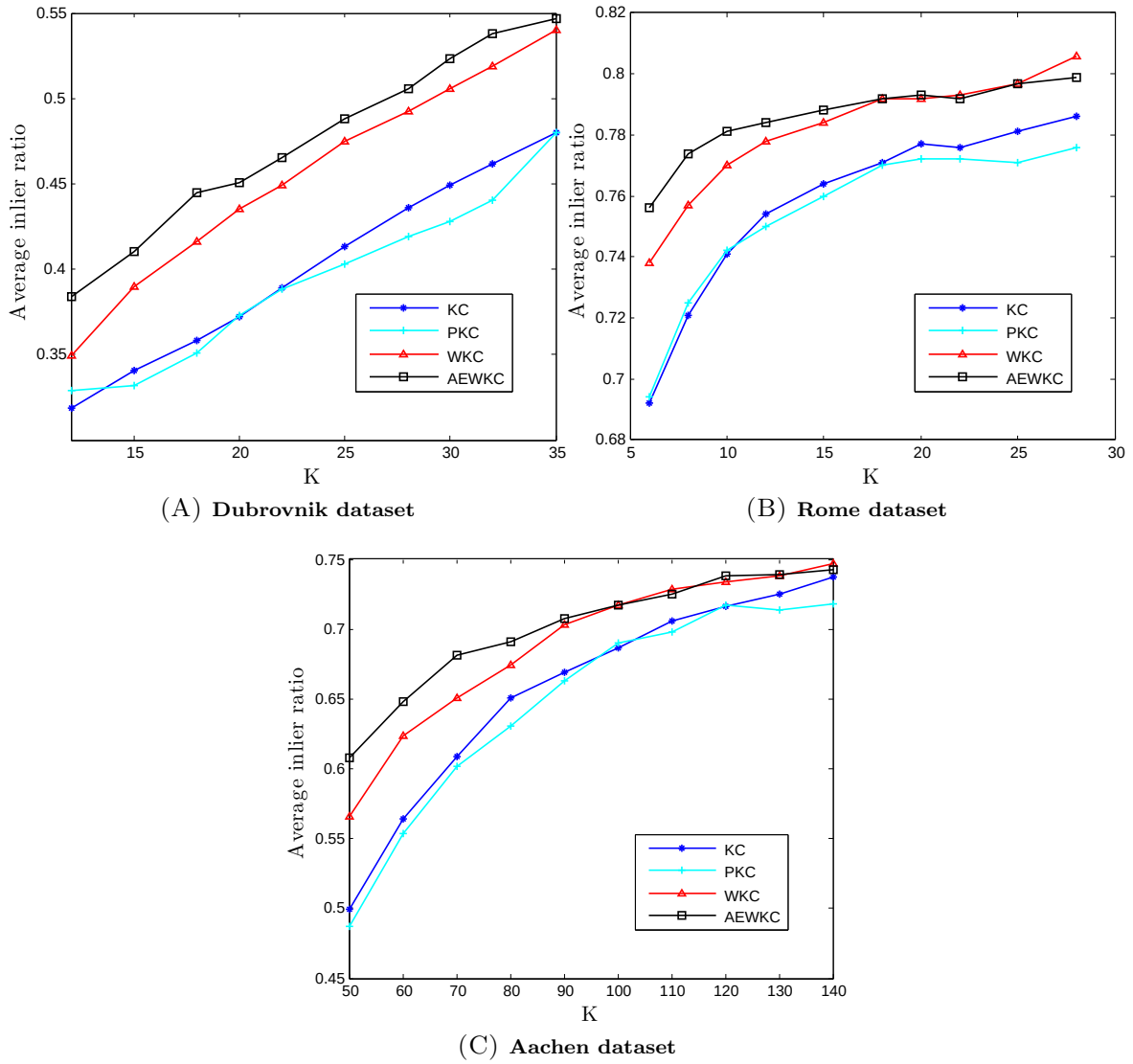
3.4.4 Evaluation of Adaptive Exponential Weighted K-Cover

In order to fairly compare with other SfM point cloud simplification methods, we keep the same number of points with other methods. For WKC and AEWKC, we use the

same parameter K with KC, and terminate WKC and AEWKC until the same number of points are selected as KC. However, PKC generally needs a smaller K to initialize the simplification, we follow the choice in [CS14] to use approximate $0.6K$ and terminate PKC until it selects the same number of points as KC. For each dataset, we start with a K value that provides a reasonable localization performance, *e.g.* $> 60\%$. The subsequent nine K values are set with different intervals for a complete comparison with a wide range of localization performance.

3.4.4.1 Localization Performance

Table 3.5 shows the localization results on all three datasets. Since we focus on improving the localization performance in limited environment, we report results when using small K. As can be seen from Table 3.5, WKC and AEWKC outperform other methods in most cases. For example, on the Dubrovnik dataset AEWKC achieves 82.96% localization performance with $K = 20$ compared to 68.99% using KC and 67.44% using PKC. The sizes of the simplified point clouds are extremely compact compared with original point cloud. Take the Dubrovnik dataset as example, AEWKC uses only 0.91% of the original point cloud to achieve 89.83% localization performance. Table 3.6 gives the comparison on the Rome dataset. WKC and AEWKC still outperform the other two methods in all cases. The advantage is lower compared to the results on the Dubrovnik dataset. On the Aachen dataset, a larger portion of the points are needed. 3.45% of points are needed to achieve 72.08% localization performance, which is 83.91% of the performance using the original point cloud. There are two reasons why the improvement of PKC over KC is not as significant as in [CS14]. First, PKC involves a trade-off which sacrifices the coverage of the database images and obtains a more discriminative SIFT feature space. The registration algorithm used in [CS14] utilizes a kd-tree structure and the SIFT features in a query image are directly matched with the SIFT features in the point cloud. However, the registration algorithm [SLK12] in our work adopts a bag-of-word model on the SIFT feature space, and the SIFT features in a query image are matched with the visual words first. The registration algorithm in our work thus has less workload which rely on computing the Euclidean distance between two SIFT features than the registration algorithm in [CS14]. The gain brought by the discrimination of the feature space with the registration algorithm in our work is thus less than that with the registration algorithm in [CS14]. Second, to generate a point cloud of the same number of points with the baseline KC approach,



(A) **Dubrovnik dataset** (B) **Rome dataset**

(C) **Aachen dataset**

FIGURE 3.8: The average inlier ratio comparison

PKC usually terminates earlier leaving some database images uncovered. In addition, PKC contains two stages in which the first stage generates an initial point subset with an extra parameter. In our work, we set this parameter to approximately $0.6K$ as recommended in [CS14]. This parameter setting based on empirical studies will aggravate the effect that PKC sometimes may terminate too early and reduces its effectiveness. Given a relatively large K, we notice that the WKC approach sometimes outperforms the proposed AEWKC approach. The advantage is generally below 0.5%. Concerning that in most cases AEWKC outperforms WKC with a large advantage, we believe that the AEWKC approach is the most effective one.

Fig. 3.8 shows the average inlier ratios of the point clouds using four simplification methods. In almost all the cases, AEWKC has the highest inlier ratio which indicates

TABLE 3.5: Localization performance comparison

(A) Dubrovnik dataset

K	#points	%points	KC(%)	PKC(%)	WKC(%)	AEWKC(%)
12	5808	0.31	40.43	40.53	59.82	**67.65**
15	7571	0.40	51.79	52.68	71.15	**76.10**
18	9391	0.50	61.98	62.56	78.06	**80.44**
20	10615	0.56	68.99	67.44	80.71	**82.96**
22	11894	0.63	72.25	73.75	83.22	**84.20**
25	13877	0.74	78.78	79.37	**86.25**	86.20
28	15895	0.84	82.50	83.25	88.43	**89.06**
30	17147	0.91	84.61	84.50	89.50	**89.83**
32	18700	1.00	86.54	86.74	**90.00**	89.75
35	20807	1.10	88.01	88.19	**91.05**	90.68

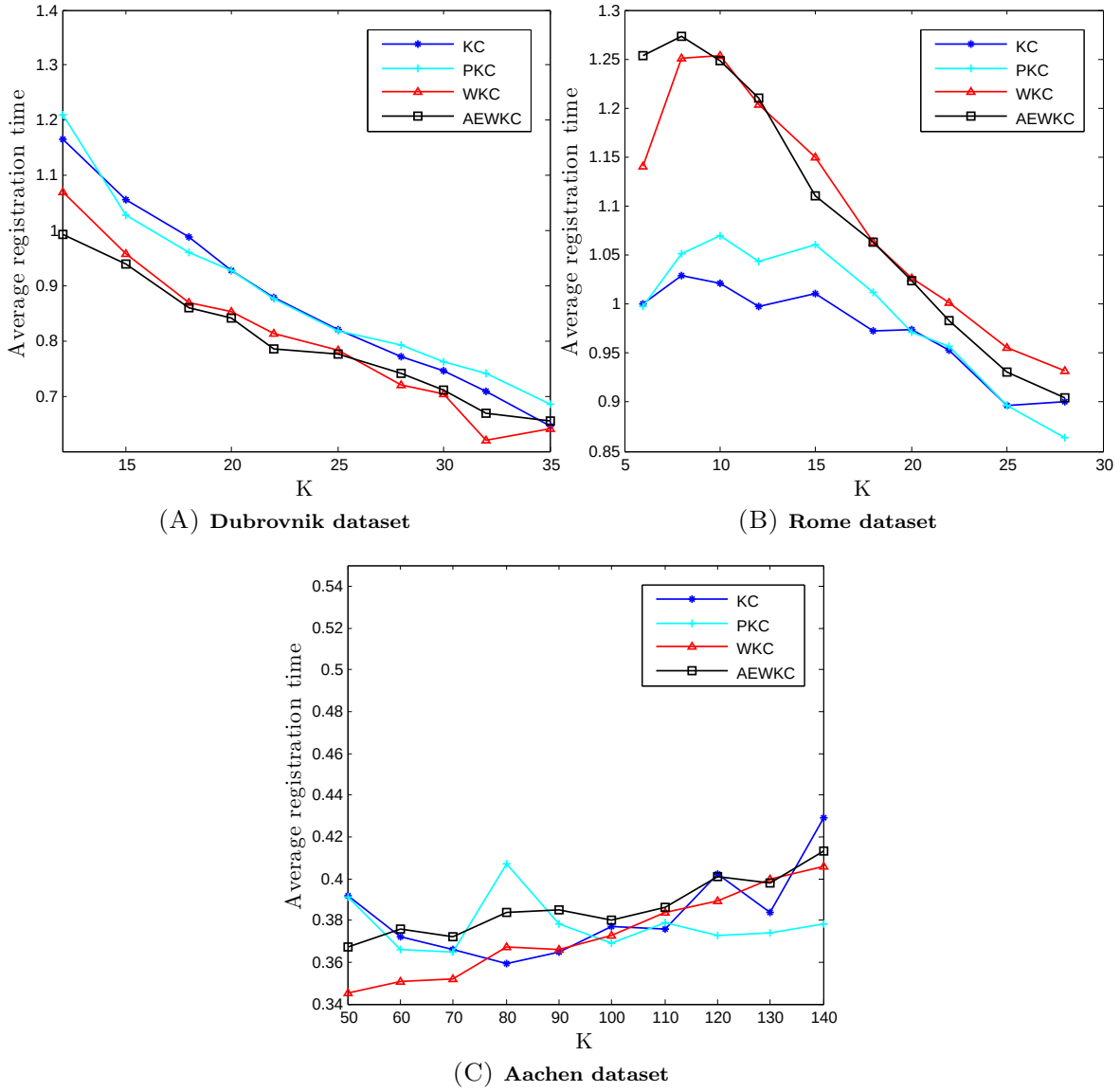
(B) Rome dataset

K	#points	%points	KC(%)	PKC(%)	WKC(%)	AEWKC(%)
6	5109	0.13	45.61	46.05	61.78	**67.66**
8	7053	0.17	63.22	64.43	72.82	**74.69**
10	9117	0.22	72.03	72.99	78.27	**79.45**
12	11216	0.28	77.93	78.66	82.23	**82.75**
15	14598	0.36	83.83	83.90	83.98	**86.78**
18	18071	0.44	86.84	86.66	87.66	**88.25**
20	20426	0.50	88.28	88.25	89.12	**89.24**
22	22828	0.56	89.04	88.92	**90.35**	90.05
25	26560	0.65	90.65	90.30	91.15	**91.33**
28	30367	0.75	91.75	91.52	91.82	**91.99**

(C) Aachen dataset

K	#points	%points	KC(%)	PKC(%)	WKC(%)	AEWKC(%)
50	19487	1.26	46.07	46.31	55.28	**60.43**
60	24036	1.56	53.38	54.90	59.62	**63.68**
70	28693	1.86	58.53	56.36	62.87	**65.58**
80	33445	2.17	63.14	63.68	65.85	**67.20**
90	38290	2.49	64.22	64.66	**68.83**	67.15
100	43186	2.80	67.20	64.55	68.29	**69.10**
110	48141	3.12	67.20	66.80	71.00	**71.17**
120	53220	3.45	68.02	68.56	**72.08**	**72.08**
130	58394	3.79	70.00	70.00	72.95	**73.14**
140	63613	4.13	71.05	71.07	73.71	**74.07**

the camera poses estimated with point clouds simplified by AEWKC are the most robust. Fig. 3.9 and Fig. 3.10 report the average registration time to localize a query image and the average rejection time to reject a query image. The registration and

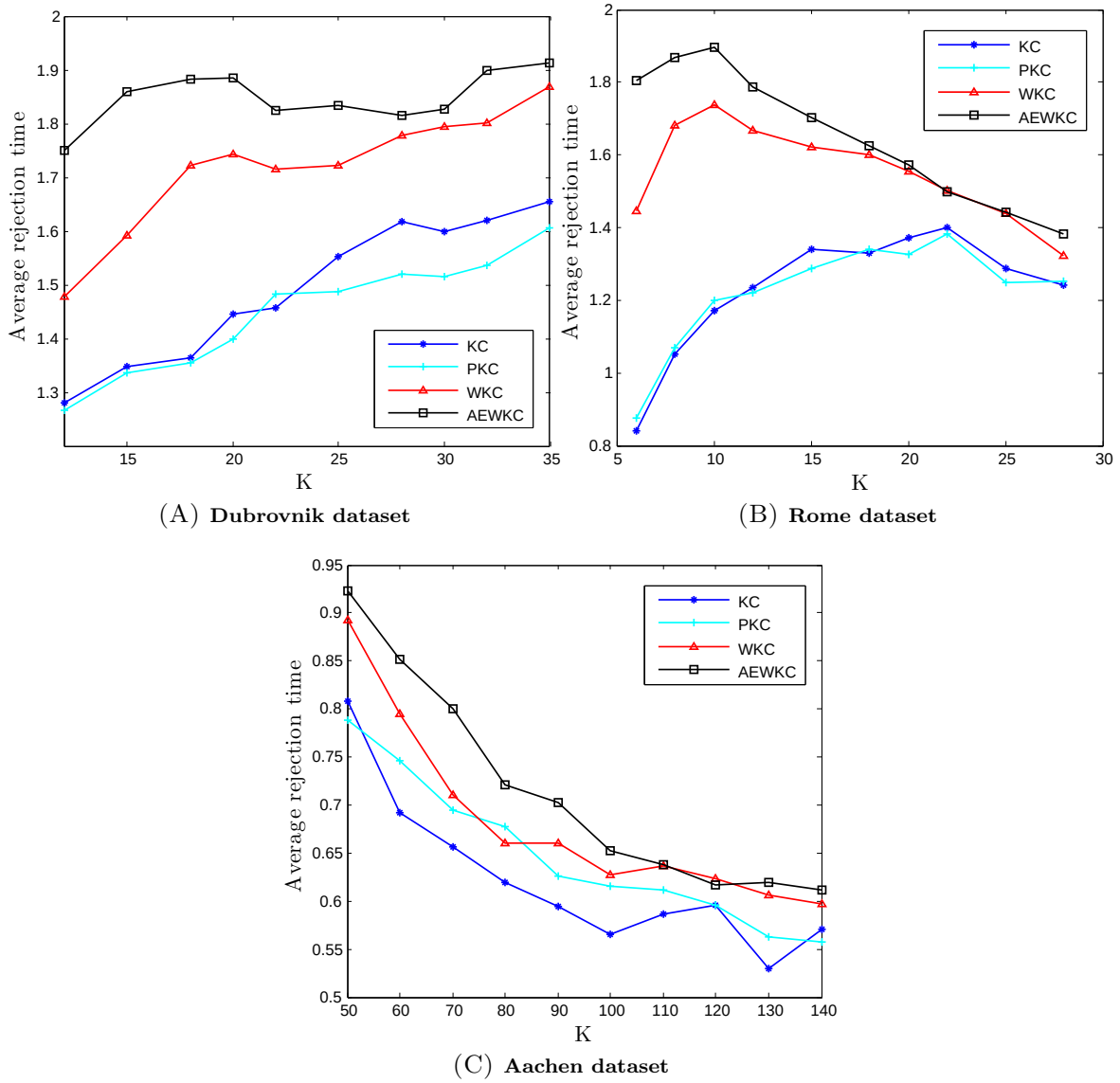


(A) **Dubrovnik dataset** (B) **Rome dataset**

(C) **Aachen dataset**

FIGURE 3.9: The average registration time comparison (in seconds)

rejection time mainly contain two parts: the time for establishing correspondences and the time for RANSAC. Fig. 3.9A shows that WKC and AEWKC need less time to register a query image on the Dubrovnik dataset. Even though WKC and AEWKC spend more time on establishing correspondences, the high quality correspondences significantly reduce the time for RANSAC. As shown in Fig. 3.9B, WKC and AEWKC have a higher registration time on the Rome dataset because the main computation budget is spent on establishing correspondences. Fig. 3.9C shows that four approaches' registration times are comparable on the Aachen dataset. We notice that AEWKC and WKC need slightly more time to reject an image. The reason is that point clouds simplified using AEWKC and WKC can establish more 2D-3D correspondences with



(A) **Dubrovnik dataset** (B) **Rome dataset**

(C) **Aachen dataset**

FIGURE 3.10: The average rejection time comparison

a query image, which make the feature matching process longer.

3.4.4.2 Localization Accuracy

Fig. 3.11 shows the average localization error on the Dubrovnik dataset. WKC and AEWKC not only register more query images than other two methods, but they also have smaller localization errors. For example, the average median localization error of AEWKC is 3.77 meter compared to the ground truth. As reported in [SLK12], the median localization error using the original point cloud is 1.40 meter. Considering the fact that we use less than 1% of points, the localization error is still acceptable.

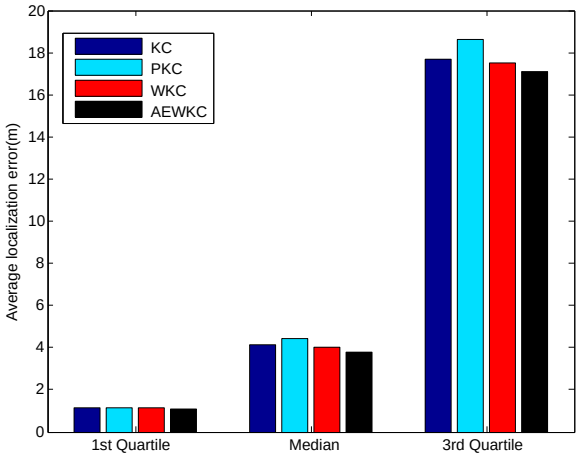


FIGURE 3.11: The comparison of average localization error on the Dubrovnik dataset.

TABLE 3.6: The computational cost comparison (unit:seconds)

(A) Dubrovnik dataset

K	12	15	18	20	22	25	28	30	32	35
KC	22.2	29.0	35.6	40.0	45.8	53.2	60.4	65.8	71.2	79.2
PKC	104.4	157.8	209.0	239.2	257.8	299.2	349.2	369.4	392.4	439.8
WKC	23.8	30.6	37.6	42.2	47.4	55.0	61.8	67.6	72.8	80.8
AEWKC	23.8	30.8	37.4	42.2	47.4	55.2	62.0	68.0	73.0	80.8

(B) Rome dataset

K	6	8	10	12	15	18	20	22	25	28
KC	40.8	56.0	72.2	90.2	117.0	144.6	163.2	182.6	212.2	244.0
PKC	227.0	284.4	402.0	517.2	672.8	834.0	961.8	1092.2	1312.4	1471.2
WKC	42.2	57.2	73.2	91.2	118.4	147.6	165.0	185.0	215.4	246.0
AEWKC	42.2	57.6	74.0	91.4	118.4	147.8	165.0	185.4	215.0	246.4

(C) Aachen dataset

K	50	60	70	80	90	100	110	120	130	140
KC	60.4	75.6	90.0	105.8	122.8	138.6	155.4	172.6	191.4	209.0
PKC	330.2	412.8	491.4	582.4	667.8	758.0	864.2	937.2	967.4	1061.2
WKC	62.2	77.6	91.2	107.6	124.6	140.4	157.0	175.0	193.6	212.0
AEWKC	62.0	77.6	91.8	108.2	124.6	141.2	157.4	175.0	194.0	211.8

3.4.4.3 Computational Cost

Our proposed method mainly contains two components: the parameter prediction method and the adaptive exponential weighted K-Cover algorithm (AEWKC). For the parameter prediction method, most of the computation are spent on the offline stage for building the model between R and H. In the online stage, our method is

time-efficient since very small K values are adopt to generate the seed point clouds.

Table 3.6 shows the comparison of the computational cost of the AEWKC algorithm and other state-of-the-art approaches. Every experiment was repeated 5 times and we reported the average running time. The proposed AEWKC and WKC approach utilize each 3D point's visibility which can be efficiently extracted from the original point cloud. Thus these two methods both have a negligible computational overhead against the KC approach. The PKC approach takes a point cloud's SIFT feature space into consideration and causes a large computational overhead to compute the Euclidean distance between 128 dimensional SIFT features. We use the PKC code provided by the authors. In general, the computational cost of PKC is five times higher than the cost of other four approaches.

3.5 Summary

In this chapter, we investigate in using simplification techniques to reduce the prohibitive memory consumption of large-scale SfM point clouds. Based on the K-Cover algorithm, we propose an SfM point cloud simplification framework that contains two key components. The first component is a prediction method to obtain an appropriate parameter setting in a data-driven manner. The second component is an adaptive weighted scheme which can be easily integrated into the greedy heuristic K-Cover algorithm. The experimental results on benchmark datasets demonstrate that: due to reliable parameter prediction, our proposed framework makes SfM point cloud simplification applicable to point clouds of different characteristics. In addition, the simplified SfM point cloud generated by our framework exhibits superior localization performance compared with state-of-the-art methods. The framework proposed in this chapter opens the door for large-scale image-based localization applications to be run on devices with limited memory resources.

Chapter 4

Two-stage Outlier Filtering for Urban Image-based Localization

In this chapter, we aim to handle the match disambiguation problem in urban image-based localization. The reasons causing this problem challenging are twofold. First, the dense feature space of a large-scale SfM point cloud makes correct matches difficult to be distinguished based on feature appearance. Second, urban environments typically contain massive repetitive structures. The existence of many nearly identical feature descriptors thereby severely reduce the distinctiveness of correct matches. To this end, we propose a two-stage outlier filtering framework [CCL+19][1] that has the following contributions:

- A two-stage outlier filtering framework is proposed that simultaneously leverages the merits of the visibility and the geometry intrinsics of an SfM point cloud for urban image-based localization. The proposed framework removes outliers in a coarse-to-fine manner by sequentially applying the designed visibility and geometry based outlier filters.
- We propose a visibility-based outlier filter, which utilizes the bipartite relationship between database images and 3D points in an SfM point cloud. Through database image re-ranking and match augmentation, the visibility-based outlier

[1] **Wentao Cheng**, Kan Chen, Weisi Lin, Michael Goesele, Xinfeng Zhang, Yabin Zhang, A Two-stage Outlier Filtering Framework for City-Scale Localization using 3D SfM Point Clouds. IEEE Transaction on Image Processing, 28(10): 4857-4869, 2019

filter is able to preserve more correct matches without severely degrading the filtering quality.

- We derive a novel data-driven geometrical constraint for *locally visible points*, which are widespread in SfM point clouds depicting urban environments. Based on this constraint, we propose a geometry-based outlier filter in which matches with *locally visible points* and *non-locally visible points* are separately evaluated with a hybrid scheme. Comparing with the classic re-projection error measurement, the derived geometrical constraint exhibits a superior efficiency in handling matches with large outlier ratio.

- The effectiveness and efficiency of the proposed two-stage framework and its individual modules are comprehensively analyzed. Based on the extensive experimental results, the matches generated by our method show a high reliability for successful large-scale urban image-based localization.

The rest of this chapter is organized as follows: Section 4.1 gives an overview of the proposed two-stage outlier filtering framework. Section 4.2 presents the proposed visibility-based outlier filter as the first stage. Section 4.3 presents the proposed geometry-based outlier filter as the second stage. Section 4.4 shows comprehensive experimental results on two read-world large-scale datasets.

4.1 Proposed Framework

Fig. 4.1 illustrates the complete localization pipeline with the proposed two-stage outlier filtering framework. The pipeline starts with a 1-to-N feature matching procedure [ZSP15] between a query image and a pre-computed SfM point cloud to obtain a set of 2D-3D matches $\mathcal{M}$. In the beginning of the visibility-based outlier filter, we use a relaxed SIFT ratio test as an initialization step to leverage its power of rejecting unreliable matches. By casting votes to database images using the initialized matches $\mathcal{M}_f$ and the bipartite visibility graph, the probability that a database image contains correct matches can be measured by its corresponding weighted votes. After database image re-ranking, wrong matches can be filtered using the top rank database images. Moreover, correct matches can also be augmented using the top rank database images. With the matches $\mathcal{M}_v$ obtained by the visibility-based outlier filter in the first stage, a

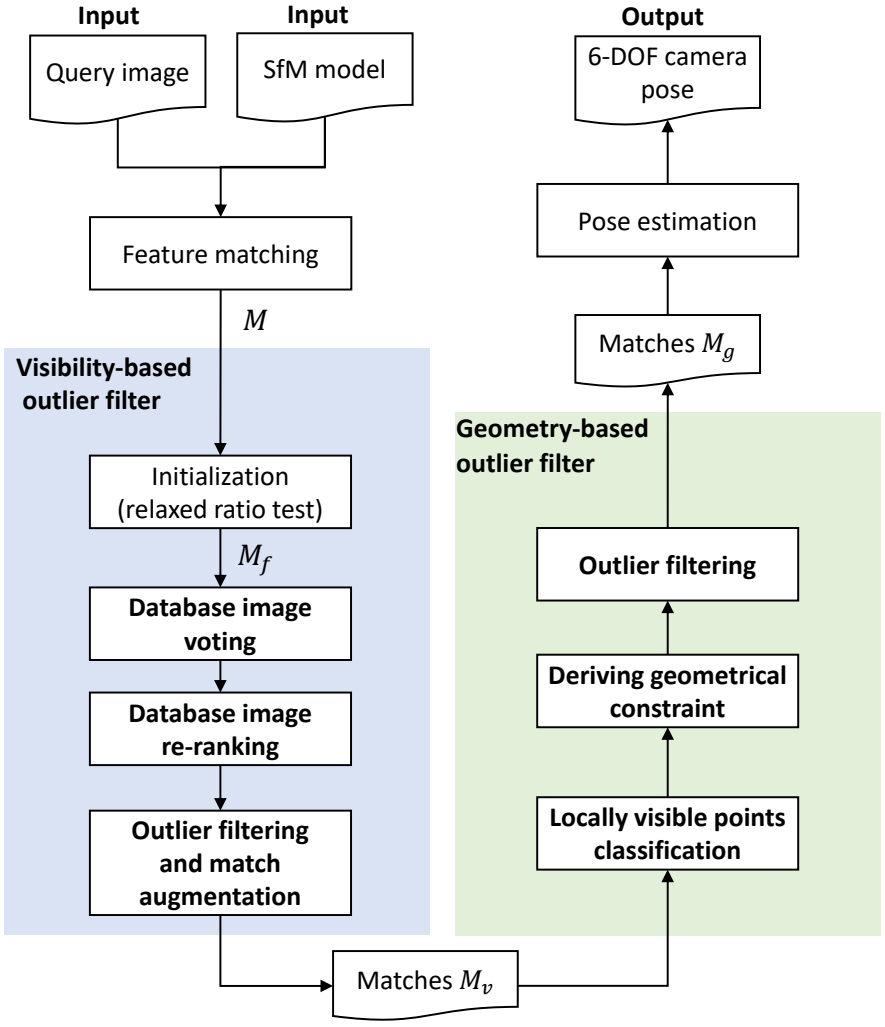


FIGURE 4.1: The localization pipeline with the proposed two-stage outlier filtering framework (in bold font).

subsequent geometry-based outlier filter is applied as the second stage. *Locally visible points* are classified and a novel geometrical constraint is derived based on *locally visible points*. The outliers can be further removed by integrating the derived geometrical constraint into a RANSAC-based pose estimation method. The final 6-DOF camera pose can be computed using the matches $\mathcal{M}_g$ generated from the geometry-based outlier filter.

4.2 Visibility-based Outlier Filter

The pipeline of the proposed visibility-based outlier filter is illustrated in Fig. 4.2. In the following, we will describe the proposed visibility-based outlier filter in detail.

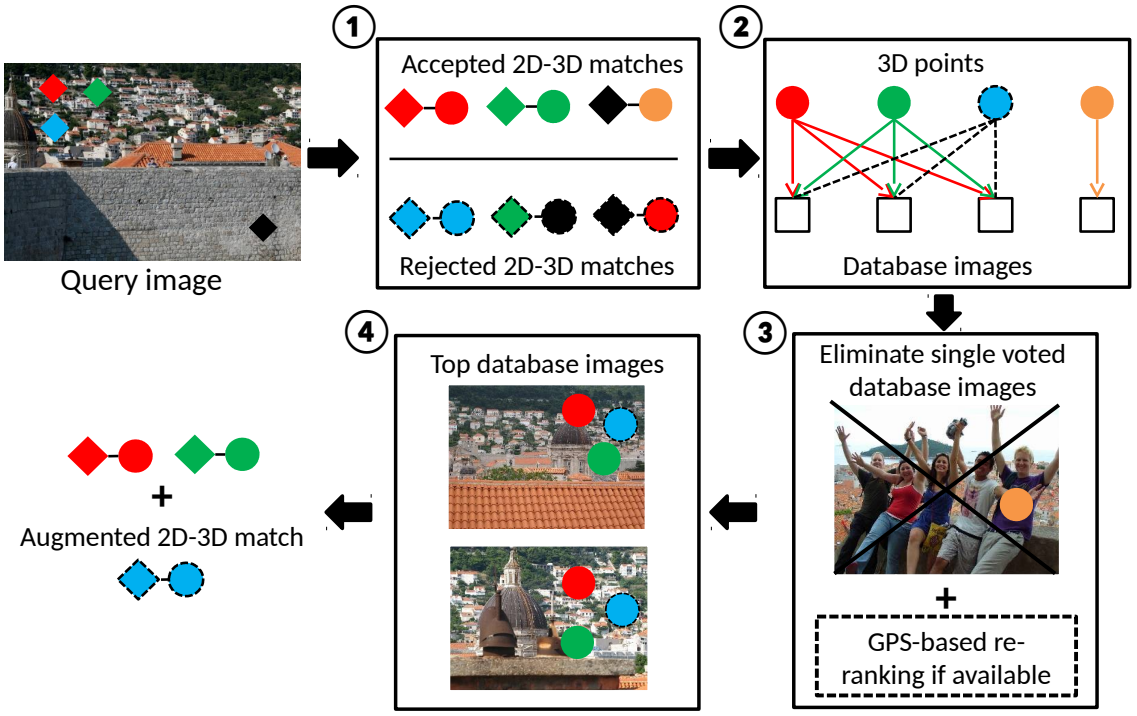


FIGURE 4.2: The pipeline of the proposed visibility-based outlier filter. 1: initialization with a relaxed ratio test (Section 4.2.1). 2: database image voting with the bipartite visibility graph (Section 4.2.2). 3: re-ranking by eliminating single voted database images (Section 4.2.3). In addition, the ranking can be optionally refined if GPS data is available. 4: outlier filtering and match augmentation (Section 4.2.4).

4.2.1 Initialization

In an SfM point cloud, each 3D point is associated with a set of 2D feature descriptors such as SIFT feature descriptors [Low04]. 2D-3D matches can be established by searching nearest neighbors in an SfM point cloud for each query feature descriptor. Let $\mathcal{M}$ be a set of initial 2D-3D matches established between a query image and an SfM point cloud. The visibility-based outlier filter starts with rejecting matches based on feature appearance. The SIFT ratio test is a widely utilized method to reject unreliable matches [Low04, SLK12, ZSP15, SEKO17, CSC+17]: let p^{1st} and p^{2nd} be the first and second nearest neighbors in an SfM point cloud for a query feature q. A match is

considered to be reliable if it satisfies the ratio test: $\|q - p^{1st}\|_2 / \left\|q - p^{2nd}\right\|_2 < \tau$. The threshold τ is usually set as 0.8 when matching between two images. However, due to the high density of feature space in a city-scale SfM point cloud, a correct match will often fail the SIFT ratio test and be rejected. In order to preserve more correct matches while rejecting wrong matches as much as possible, a relaxed SIFT ratio test should be applied. The relaxation can be done by either increasing the threshold of the SIFT ratio test or using an adaptive threshold [ZSP15]. Let $\mathcal{M}_f$ be the matches that are accepted by the relaxed SIFT ratio test. In practice, $\mathcal{M}_f$ contains much fewer ambiguous matches than the original matches $\mathcal{M}$. We therefore use $\mathcal{M}_f$ instead of $\mathcal{M}$ for the following database image voting procedure.

4.2.2 Database Image Voting

After obtaining the matches $\mathcal{M}_f$ with a relaxed SIFT ratio test, we aim to remove outliers by utilizing the visibility intrinsics of an SfM point cloud. In an SfM point cloud, the relationship between 3D points and database images can be modelled as a bipartite visibility graph $\mathcal{G} = (\mathcal{P}, \mathcal{D}, \mathcal{E})$. Each node $p \in \mathcal{P}$ represents a 3D point in the SfM point cloud, and each node $d \in \mathcal{D}$ represents a database image which is used to reconstruct the SfM point cloud. An edge $(p, d) \in \mathcal{E}$ exists if the 3D point p is visible in the database image d.

Leveraging the bipartite graph $\mathcal{G}$, each 2D-3D match $(q, p) \in \mathcal{M}_f$ can cast a vote to the database images that observe p. Thus, the votes for a database image d can be computed as follows:

$$\mathcal{V}(d) = \{(q, p) \mid (p, d) \in \mathcal{E}, (q, p) \in \mathcal{M}_f\}. \tag{4.1}$$

Ideally, a correct 2D-3D match (q, p) means that the query feature q should depict the same location as the 3D point p. Due to the continuity of geometry space, correct matches should be frequently co-visible. The co-visibility of correct matches makes the corresponding database images receiving high votes. Meanwhile, the weak co-visibility among wrong matches makes them randomly casted to irrelevant database images. In a city-scale dataset which contains a large number of database images, the votes that each database image can receive with wrong matches should be much smaller than the votes from correct matches. However, a database image that observes more 3D points is inherently more likely to receive votes from wrong matches. In order to avoid bias

towards database images with more visible 3D points, the original vote $|\mathcal{V}(d)|$ should be weighted by the number of 3D points that are seen by the database image d. Let $\mathcal{F}(d) = \{p \mid (p,d) \in \mathcal{E}\}$ be the 3D points observed by the database image d. The weighted votes $\mathcal{W}(d)$ of the database image d can be calculated as follows:

$$\mathcal{W}(d) = \frac{|\mathcal{V}(d)|}{|\mathcal{F}(d)|}. \tag{4.2}$$

In real-world scenes, there are various kinds of repetitive patterns, *e.g.* doors or windows, in a local region. It is possible that a query feature may establish multiple locally ambiguous 2D-3D matches in repetitive patterns. The locally ambiguous matches will falsely increase the weighted votes of the corresponding database images especially when the votes contain few correct matches. Unfortunately, the relaxed SIFT ratio test used in the initialization cannot entirely remove such locally ambiguous matches. In order to reduce the influence of the locally ambiguous matches in the database image voting procedure, we use an approach similar to Sattler *et al.* [SHR+15] to enforce that a query feature casts one unique vote to the same database image. Considering a query feature q that establishes local ambiguous matches to the database image d as $\{(q,p) \mid (p,d) \in \mathcal{E}\}$, we randomly choose one match from the locally ambiguous matches for casting vote to make sure that $\forall (q',p') \in \mathcal{V}(d) \setminus (q,p) : q \neq q'$.

4.2.3 Database Image Re-ranking

A database image with more weighted votes indicates that the corresponding matches are more likely to be correct. Thus the outlier filtering problem can be formulated as solving an image retrieval problem. Given a query image, the database images are ranked according to the corresponding weighted votes. Among the top rank database images, special attention should be paid on those, which receive only one single vote from the established matches. In a city-scale dataset, it is common that some database images can only see a small number of 3D points due to low image resolution or viewpoint uniqueness. For such database images, a single vote could produce a large weighted vote value and a top rank. To recap, our core idea is based on the fact that correct matches are frequently co-visible and thereby vote to the same database image. Since the single vote does not exhibit any co-visibility feature, we first eliminate all database images with $|\mathcal{V}(d)| \leq 1$ from the database image list. The top K database images $\mathcal{D}^K$ are selected for outlier filtering. In addition, for the datasets with additional

prior information such as GPS data, we can use the available data to further refine the ranking of database images. The Euclidean distance between the query image and each database image can be estimated using the associated GPS tags. We only select the top K database images whose Euclidean distances to the query image are below a threshold. In this work, we set this threshold as 300 meters as suggested by Zeisl *et al.* [ZSP15]. To avoid misunderstanding, all distances mentioned below are Euclidean distances.

4.2.4 Outlier Filter and Match Augmentation

In previous approaches [LSHF12, ZSP15, CSC+17], an inappropriate assumption is that the matches rejected by the relaxed SIFT ratio test are all wrong matches. Here, we point out that even though wrong matches take up the majority of the rejected matches by the relaxed SIFT ratio test, a portion of correct matches are mistakenly rejected. It is meaningful and beneficial to recover correct matches back to further improve the quality of the matches. After retrieving the top rank database images $\mathcal{D}^K$, a match in $\mathcal{M}_f$, which casts a vote to one of the database image in $\mathcal{D}^K$, can be safely selected into $\mathcal{M}_v$ as follows:

$$\mathcal{M}_v = \left\{(q,p) \mid (q,p) \in \mathcal{M}_f, (p,d) \in \mathcal{E} \wedge d \in \mathcal{D}^K\right\}. \tag{4.3}$$

Moreover, for a match in $\mathcal{M} \setminus \mathcal{M}_f$ which also casts a vote to one of the database image in $\mathcal{D}^K$, it can be recovered as long as the associated query feature has not been found in $\mathcal{M}_v$ yet. Therefore, for each match in $(q', p') \in \mathcal{M} \setminus \mathcal{M}_f$, we iteratively select it into $\mathcal{M}_v$ if $\forall (q,p) \in \mathcal{M}_v : q \neq q'$. Note that the recovered matches from $\mathcal{M} \setminus \mathcal{M}_f$ are not involved in the previous database image voting procedure.

4.3 Geometry-based outlier filter

Having obtained the matches $\mathcal{M}_v$ using the visibility-based outlier filter in the first stage, we now propose to further filter wrong matches using geometrical considerations. Our key observation is that visual occlusion is a common phenomenon in a city-scale SfM point cloud. Therefore, there are a large number of *locally visible points*, which

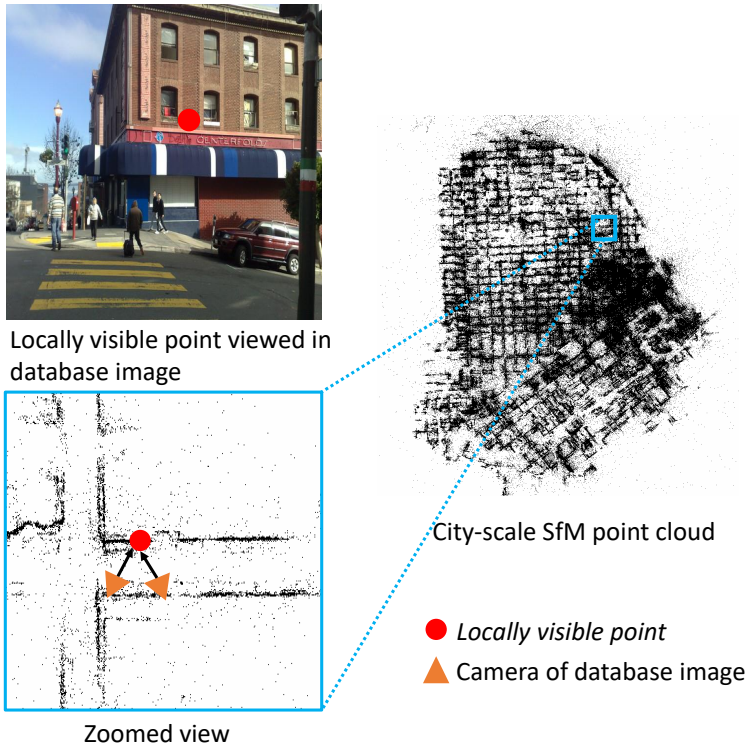


FIGURE 4.3: An illustration of a *locally visible point* in the *San Francisco* dataset [LSHF12]. A *locally visible point* (red) is observed by nearby cameras (orange) of the database images.

are only observed by database images whose camera positions lie nearby. Fig. 4.3 illustrates a typical example of a *locally visible point*. The restriction of cameras observing *locally visible points* enables us to derive a novel geometrical constraint that is simply based on the camera position. Different from traditional re-projection error measurement, the proposed geometrical constraint can serve as a more robust inlier evaluation measurement in RANSAC-based pose estimation, especially under large outlier ratio scenario. In this section, we will describe the proposed geometrical constraint and its application in detail.

4.3.1 A Data-driven Geometrical Constraint

In order to efficiently classify the *locally visible points*, we leverage the bipartite visibility graph $\mathcal{G}$. Let $\mathcal{I}(p)$ be the set of database images which observe the 3D point p as follows:

$$\mathcal{I}(p) = \{d \mid (p, d) \in \mathcal{E}\}. \tag{4.4}$$

Suppose $\mathcal{I}(p)$ is of size n, a 3D point p can be regarded as a *locally visible point* if the distance between p and the camera position of each database image in $\mathcal{I}(p)$ is below a defined distance threshold T_{local} as follows:

$$\forall c_i : \|c_i - p\|_2 \leq T_{local}, \tag{4.5}$$

where c_i represents the camera position of the i^{th} database image in $\mathcal{I}(p)$.

For each *locally visible point* in an SfM point cloud, we derive a geometrical constraint to restrict the position of a hypothetical camera, which can observe the *locally visible point*. We define a sphere of radius r around the *locally visible point* to represent the region that a hypothetical camera may appear. The radius should be smaller than the distance defined in Eq. 4.5 to ensure the locality. In addition, an adaptive radius can be defined based on the average camera-to-point distance from a *locally visible point* p to the camera position of each database image in $\mathcal{I}(p)$. The average camera-to-point distance is calculated using the equation:

$$\text{dist}(p) = \frac{\sum_{i=1}^{n}(\|c_i - p\|_2)}{n} \tag{4.6}$$

where c_i represents the camera position of the i^{th} database image in $\mathcal{I}(p)$. For cases when the average camera-to-point distance is much smaller than the local distance threshold T_{local} in Eq. 4.5, we define an adaptive radius as $r = \alpha\text{dist}(p)$. Therefore, the radius of the sphere is $r = \min(\alpha\text{dist}(p), T_{local})$. In this work, we empirically set $\alpha = 4$.

In addition, we apply the angle constraint [JDS08, ZSP15, SHR+15] based on the view direction since the SIFT feature descriptor is variant to a significant viewpoint change. For each database image that can observe the *locally visible point* p, we compute the viewing direction as a normalized vector pointing from the camera position c_i to p. For a camera that observes p, the angle between the current viewing direction and the viewing direction from one of the database images should be smaller than an angle threshold λ. Therefore the final derived geometrical constraint $Constraint(c_h, p)$ can be defined as follows:

$$Constraint(c_h, p) = \begin{cases} \|c_h - p\|_2 < \min(\alpha\text{dist}(p), T_{local}) \\ \exists c_i : \measuredangle(\overrightarrow{c_h p}, \overrightarrow{c_i p}) < \lambda, \end{cases} \tag{4.7}$$

where c_i represents the camera position of the i^{th} database image in $\mathcal{I}(p)$. The derived geometrical constraint is illustrated in Fig. 4.4.

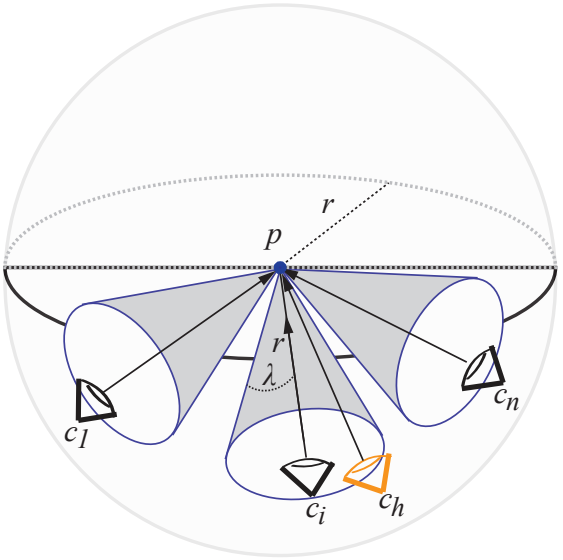


FIGURE 4.4: The derived geometrical constraint for a *locally visible point* p. For each camera position of the database image which observes p, we define a cone with height r and angle λ. A hypothetical camera c_h which observes p should lie inside at least one of the defined cones.

4.3.2 The Outlier Filter

In order to apply the derived geometrical constraint to filter outliers, a hypothetical camera position needs to be established. Assuming that the camera's internal calibration matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ of a query image is known in advance, we utilize a P3P pose solver [KSS11] to establish a hypothetical camera pose $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$, where $\mathbf{R}$ represents the rotation matrix and $\mathbf{t}$ is the translation vector. The hypothetical camera position can be computed as $\mathbf{c} = -\mathbf{R}\mathbf{t}$. Given the matches $\mathcal{M}_v$ generated after the visibility-based outlier filter, our goal is to find the camera position that is most likely to observe the 3D points associated with correct matches in $\mathcal{M}_v$. To this end, we adopt a standard RANSAC scheme [Fis81] to verify multiple camera position hypotheses. In each RANSAC iteration, the matches corresponding to *locally visible points* are regarded as inliers if they satisfy the geometrical constraint in Eq. 4.7. The matches corresponding to *non-locally visible points* are evaluated using the traditional re-projection error measurement as follows:

$$\|q - \mathbf{P}p\|_2 \leq \gamma. \tag{4.8}$$

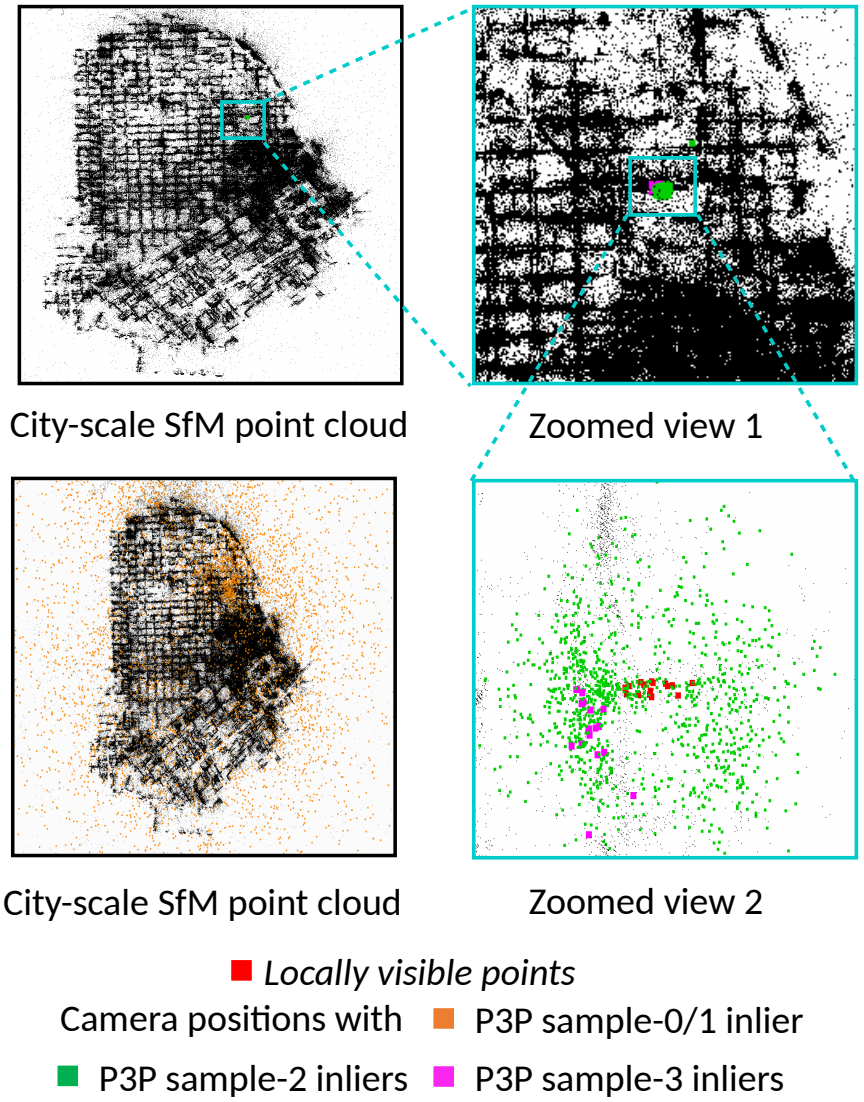


FIGURE 4.5: The distribution of camera positions in the geometry-based outlier filter for a query image that depicts a local scene. The camera positions with P3P samples (0 or 1 inlier) distributed throughout the whole SfM point cloud. It looks like many of these are clearly wrong, *e.g.* in the ocean. The distribution shows that a P3P sample with 2 inliers, which is much easier to be obtained than a P3P sample with 3 inliers under large outlier ratio scenarios, can provide us an approximate camera position to apply the proposed geometrical constraint. The data was generated by randomly sampling 10^5 trials using the image in Fig. 4.3.

The match corresponding to a *non-locally visible point* can be regarded as an inlier if the re-projection error is below the pixel threshold γ. Using the above hybrid inlier evaluation scheme, the camera model $\mathbf{P}^*$ with the largest number of inliers is returned. The inliers of $\mathbf{P}^*$ therefore are selected as $\mathcal{M}_g$ for the final pose estimation. The geometry-based outlier filter is summarized in Algorithm. 3.

By incorporating the derived geometrical constraint for *locally visible points*, our

Algorithm 3: The Geometry-based Outlier Filter

Require: $\mathcal{M}_v$, matches selected by the visibility-based outlier filter; $\mathcal{M}_{\mathcal{L}} \subseteq \mathcal{M}_v$, matches corresponding to the set of *locally visible points* $\mathcal{L}$.

Require: The camera internal matrix $\mathbf{K}$; re-projection error threshold γ; maximum RANSAC iterations $Iter$.

$Inlier_{max} \leftarrow 0$
for $j = 0; j < Iter$ **do**
  Randomly sample three matches from $\mathcal{M}_v$
  Compute the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ using P3P solver
  Obtain the projection matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ and the camera center $\mathbf{c} = -\mathbf{R}\mathbf{t}$
  Inliers $I_1 = Re\text{-}projection(\mathcal{M}_v \setminus \mathcal{M}_{\mathcal{L}}, \mathbf{P}, \gamma)$
  Inliers $I_2 = Constraint(\mathbf{c}, p), p \in \mathcal{L}$
  if $|I| \geq Inlier_{max}$ **then**
    $\mathbf{P}^* \leftarrow \mathbf{P}$, $Inlier_{max} \leftarrow |I|$
  end if
  $j \leftarrow j + 1$
end for
return The inliers of $\mathbf{P}^*$ as $\mathcal{M}_g$

geometry-based outlier filter is efficient in handling matches with large outlier ratio for two reasons. Firstly, traditional inlier evaluation method based on re-projection error requires an accurate 6-DOF camera pose. While in the proposed geometry-based outlier filter, the inlier evaluation for matches corresponding to *locally visible points* is relaxed since it only requires an approximate 3-DOF camera position. Secondly, for a query image that depicts a local scene, a P3P sample with three inliers is able to produce a theoretically correct camera position which lies nearby *locally visible points* corresponding to inliers. Inspired from Camposeco *et al.* [CSC+17], we observe that a P3P sample with only two inliers is able to produce an approximate camera position which lies nearby the theoretically correct camera positions. Fig. 4.5 shows an example of the camera position distribution with a query image that depicts a local scene. This relaxation on the number of inliers in a P3P sample significantly increases the probability of finding an approximate camera position to apply the derived geometrical constraint.

Suppose the inlier ratio of established 2D-3D matches is ε, the probability of obtaining a P3P sample with two inliers (P3P-2i) can be computed as ε^2. The traditional re-projection error measurement requires an accurate 6-DOF camera projection matrix, which is computed by a P3P sample with three inliers (P3P-3i). The probability of obtaining a P3P-3i sample can be computed as ε^3. Considering a large outlier ratio case, *e.g.* $\varepsilon < 0.1$, the probability of obtaining a P3P-2i sample is much larger than

obtaining a P3P-3i sample by a factor of $1/\varepsilon$. Therefore, the proposed geometrical constraint for locally visible points is more robust under large outlier ratio scenario comparing with traditional re-projection error measurement.

4.4 Experiments

We evaluate the proposed two-stage outlier filtering framework on two popular real-world datasets: the *San Francisco* dataset [CBK+11, LSHF12] and the *Dubrovnik* dataset [LSH10]. Table 4.1 summarizes the statistics of the datasets used in our experiments. The *San Francisco* dataset consists of 1.06 million street-view database images for image retrieval tasks. For 3D structure-based localization, we use the publicly available SF-0 SfM point cloud [LSHF12], which is built from 610k database images in the *San Francisco* dataset. The query images have a different spatial distribution compared to the database images, making feature matching in the *San Francisco* dataset difficult. In addition, each database image is associated with a precise GPS coordinate. The query images also are associated with GPS coordinates, in which some are not very precise. As far as we know, the *San Francisco* dataset is the most challenging dataset for 3D structure-based localization so far. Therefore, we mainly focus on evaluating our approach on this dataset. The *Dubrovnik* dataset has been widely studied by [LSH10, SLK11, SLK12, LSHF12] and almost all query images can be localized. Similar to recent works [ZSP15, CSC+17], we mainly focus on evaluating the pose accuracy on the *Dubrovnik* dataset. For a comprehensive comparison, we include the state-of-the-art approaches from three categories as follows:

- 3D structure-based approaches: Active search [SLK12], Co-occurrence [LSHF12], KVD [SEKO17], CPV [ZSP15], Hyperpoints [SHR+15] and Toroidal [CSC+17].
- Hybrid localization approaches which combine 2D image-based and 3D structure-based approaches: DenseVLAD + SfM [STS+17].
- Learning-based localization approach: PoseNet with novel geometrical loss functions (GLF), abbreviated as PoseNet (GLF) [KC+17].

TABLE 4.1: The statistics of the datasets used in our experiments.

Dataset	Database images	3D points	Query images
San Francisco (SF-0)	610k	30.34M	803
Dubrovnik	6k	1.89M	800

4.4.1 Evaluation on San Francisco Dataset

4.4.1.1 Implementation Details

In the feature matching step, we use the FLANN library [ML14] for approximate nearest neighbor searching between a query image and the SF-0 SfM point cloud. For fair comparison, we follow the 1-to-N matching strategy used in existing works [ZSP15, SEKO17]. For each query feature, at most 3 matches will be established. In the initialization, a match is verified with a variable search threshold, which is defined as 0.7 times the squared distance to the nearest neighbor in the query image itself. In the proposed visibility-based outlier filter, we empirically select the top 200 database images and perform the match augmentation scheme with the selected database images. In the proposed geometry-based outlier filter, we empirically set the distance threshold $T_{local} = 50$ meters and the angle threshold $\lambda = 60°$. Note that we used the same parameter setting in both the *San Francisco* dataset and the *Dubrovnik* dataset. We run a maximum of 1000 RANSAC iterations in the geometry-based outlier filter.

4.4.1.2 Evaluation Criteria

In the *San Francisco* dataset, all database images and query images are annotated with the ground truth building IDs. A query image is considered to be successfully localized if the final inliers are registered to the ground truth building IDs. We use the improved version of ground truth annotations reported by Arandjelović *et al.* [AZ14]. Note that there are 66 query images whose ground building IDs are missing in the SF-0 point cloud. We use the same evaluation criteria as customary used in previous work [CBK+11, LSHF12, ZSP15, SHR+15] that the performance is evaluated as the recall rate under 95% precision.

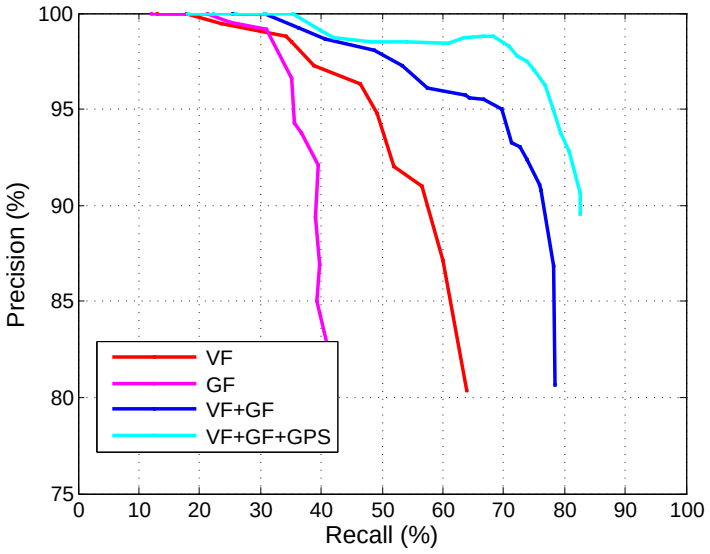


FIGURE 4.6: The experimental results of our method on the *San Francisco* dataset.

TABLE 4.2: The average match statistics of successfully localized query images in different stages of VF+GF in the *San Francisco* dataset.

Matches	$\mathcal{M}_f$	$\mathcal{M}_v$	$\mathcal{M}_g$	Final Inliers
Stage	Input	+VF	+VF+GF	+VF+GF+P3P
#Matches	4528	287	92	40
#Matches_correctIDs	76	102	84	38
%Matches_correctIDs	1.8%	30.4%	87.8%	90.2%

4.4.1.3 Overall Evaluation

Our method includes two major modules: the proposed visibility-based filter, abbreviated as VF and the proposed geometry-based filter, abbreviated as GF. In order to separately evaluate the impact of each module, we conduct several experiments on the *San Francisco* dataset with the following settings:

- VF: only use the visibility-based outlier filter.
- GF: only use the geometry-based outlier filter.
- VF+GF: use the visibility-based outlier filter and the subsequent geometry-based outlier filter.
- VF+GF+GPS: use the visibility-based filter and the subsequent geometry-based filter. Incorporate the GPS data in the visibility-based filter as described in Section 4.2.3.

TABLE 4.3: The comparison of our method with the state-of-the-art works on the *San Francisco* dataset. All the listed recall rates are measured at a 95% precision rate. The Vertical and Height assumptions mean that the camera's vertical direction with respect to the underlying SfM point cloud and the camera's approximate height are known in advance.

Method	Geometrical Assumptions	Recall Rate [%]	
		w/o GPS	w/ GPS
KVD [SEKO17]	Vertical+Height	68.0	-
CPV+P3P [ZSP15]	Vertical+Height	67.5	74.2
CPV [ZSP15]	Vertical+Height	68.7	73.7
Co-occurrence [LSHF12]	-	54.2	-
Hyperpoints [SHR+15]	-	61.9	-
Our method	-	**69.6**	**78.1**

After obtaining the set of matches using the above experimental settings, we use P3P-RANSAC [KSS11] to compute the final 6-DOF camera pose. Fig. 4.6 reports the experimental results using the above settings. For each setting, multiple recall@precision results are generated by varying the inlier threshold to determine whether a query image is successfully localized. We notice that GF achieves the worst performance among all settings. The reason is that the original matches are very noisy, i.e. below 1% inlier ratio. RANSAC used in GF requires too many iterations to find a reliable solution with such extremely noisy matches. The significant gain of VF+GF over VF indicates that the matches generated from VF may still contain a large number of outliers, which GF can remove efficiently. With VF+GF, we achieve a 69.6% recall at 95% precision. By incorporating the provided GPS data, the relevance between then selected top rank database images and the query image has been significantly improved. VF+GF+GPS can provide us a 78.1% recall at 95% precision.

In Table 4.2, we report the average match statistics of successfully localized query images using our full prior-free pipeline VF+GF. Since it is difficult to determine the number of inliers in the original matches with extremely large outlier ratios, we use the number of matches which are registered to the correct building IDs as an approximate upper bound of inliers. The ratio of the matches with correct building IDs among the whole matches can be used to evaluate the quality of the matches. The matches $\mathcal{M}_f$ after the initialization step have a very large outlier ratio, which make the pose estimation difficult. After applying VF, the quality of matches is significantly improved from a 1.8% ratio to 30.4% ratio. With the matching augmentation procedure in VF, the number of matches with correct building IDs increases from 76 to 102. However, for some query images the matches still contain a large number of wrong matches, which

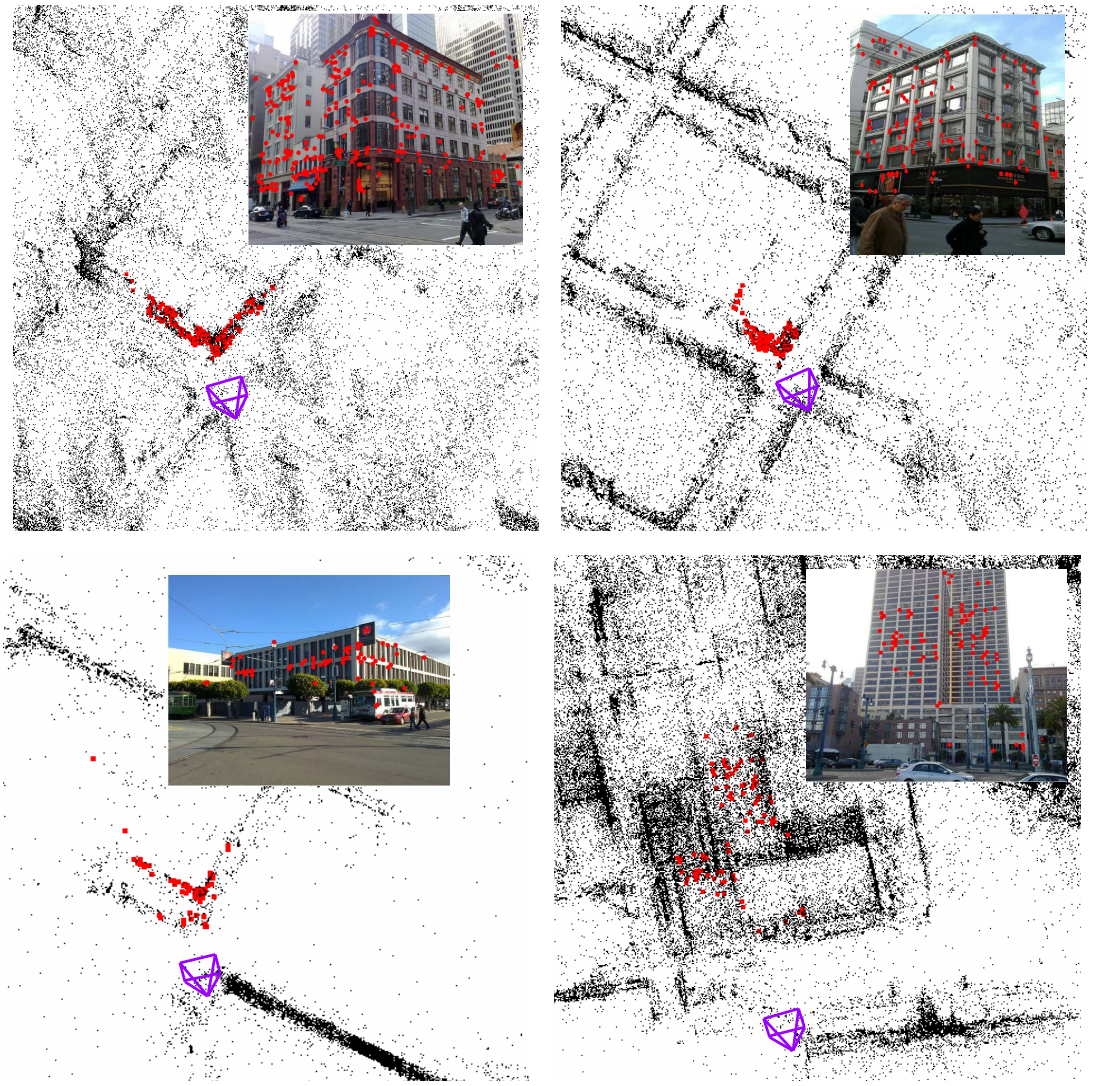


FIGURE 4.7: The exemplary query images and the corresponding estimated 6-DOF camera poses in the SF-0 SfM point cloud for the *San Francisco* dataset.

make VF obtain a lower recall rate compared with VF+GF. Due to the relaxation in both hypothesis and verification phase of RANSAC, GF is able to efficiently handle the matches with large outlier ratio, which are generated by VF. After VF+GF, the matches with correct building IDs are well preserved and the ratio significantly increases from 30.4% to 87.8%.

4.4.1.4 Comparison with state-of-the-art

Table 4.3 reports the comparison between our method and the state-of-the-art approaches. The performance is evaluated by the recall at 95% precision, which was

also used by related works [CBK+11, LSHF12, ZSP15, SHR+15]. Our method outperforms state-of-the-art 3D Structure-based methods in scenarios without and with GPS. Without additional assumptions about the camera's vertical direction and approximate height relative to the SfM point cloud, our method (GF+VF) achieves a 69.6% recall at 95% precision. By incorporating the GPS data, the recall at 95% precision increases to 78.1%. The localization performance achieved by our method (VF+GF) proves that the visibility intrinsics and geometry intrinsics in a city-scale SfM point cloud are not mutually exclusive and can be combined to remove outliers. Comparing with the 2D image-based approaches [AZ14, TSPO13, SHSP16], our method is able to provide a 6-DOF camera pose for a query image, as illustrated by Fig. 4.7. The Burstness [SHSP16] approach, which is 2D image-based, achieves a 72.4% recall at 95% precision. Note that in the Burstness approach [SHSP16], they leverage the original 1.06 million database images while the SF-0 SfM point cloud used in our method only contains the information of 610k database images. In addition, the GPS data of database images are leveraged for clustering locations.

4.4.1.5 Ablation Study of VF

To evaluate the impact of each individual component of the visibility-based outlier filter (VF), we conduct an ablation study on the *San Francisco* dataset with different VF schemes. Fig. 4.8 presents the experimental results of the re-ranking scheme in Section 4.2.3 and the match augmentation scheme in Section 4.2.4. We can notice that the re-ranking scheme improves the performance significantly. This improvement indicates that the top rank database images after re-ranking are more relevant to the query image, and are more likely to contain correct matches. The improvement of the recall rate proves that the match augmentation method is able to recover correct matches back that were previously removed. However, there is a drop of precision rate in the high precision regime ($> 95\%$). We found that the majority of the additional falsely localized query images caused by the match augmentation scheme are due to missing building ID annotations as shown in Fig. 4.9. The falsely localized query images with missing building IDs are registered to other locations with nearly identical appearances. In such cases, the 2D-3D matches recovered by the match augmentation step still have high reliability to ensure the consistency between the 2D query features and the features associated with the matched 3D points.

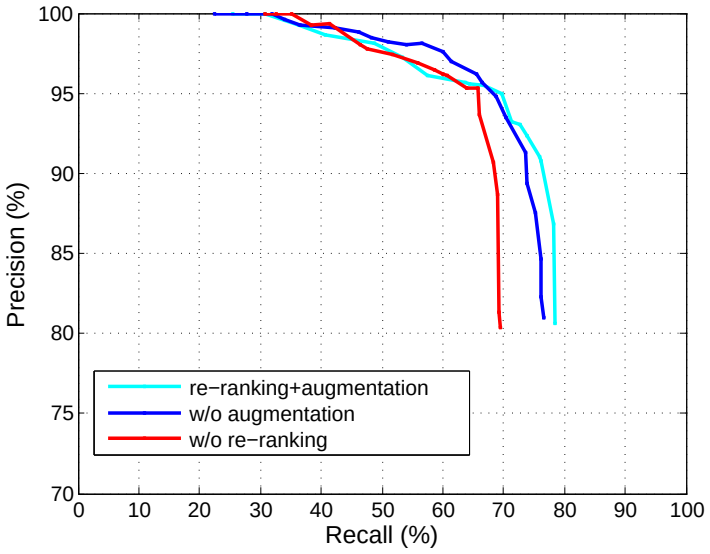
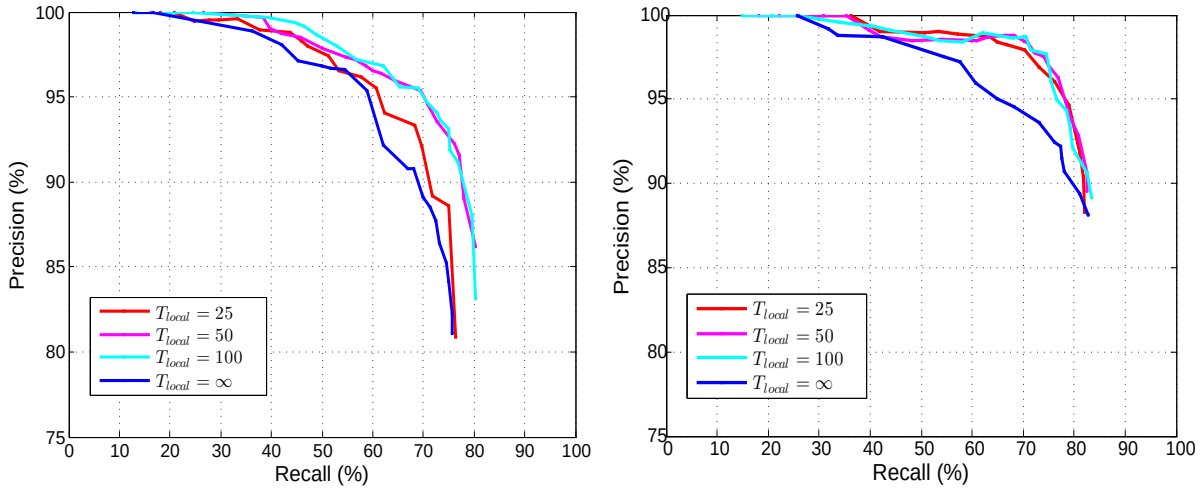


FIGURE 4.8: The performance comparison to evaluate the re-ranking scheme in Section 4.2.3 and the matching augmentation scheme in Section 4.2.4.

FIGURE 4.9: First row: the query images whose ground truth building ID annotations exist in the SF-0 SfM point cloud, the bounding box shows the corresponding pier marks of different building IDs. Second row: the falsely localized query images caused by the matching augmentation scheme. Their ground truth building ID annotations are missing in the SF-0 SfM point cloud. They are falsely registered to the same building IDs as images in the first row due to nearly identical appearance.

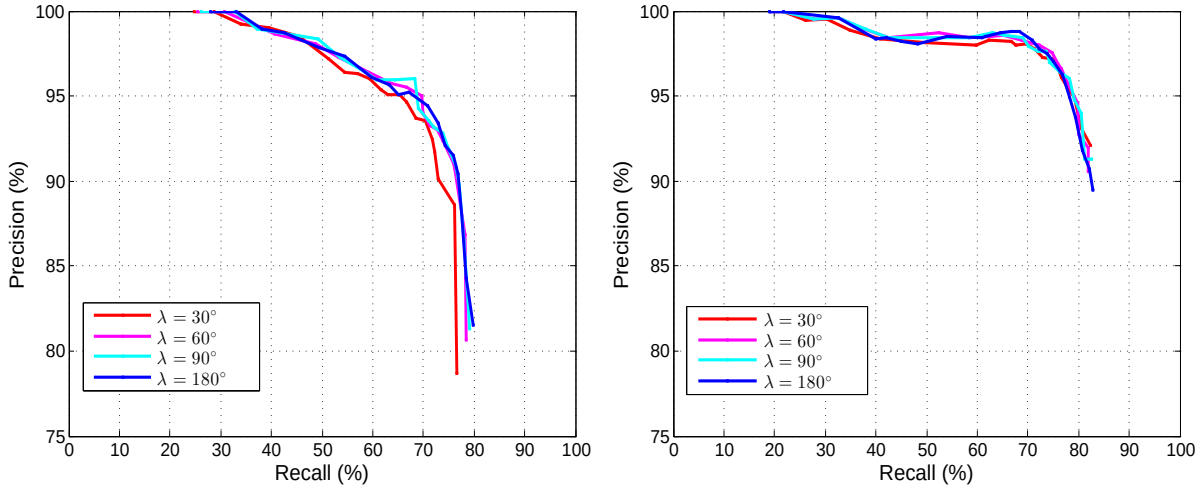
4.4.1.6 Ablation Study of GF

To evaluate the impact of each individual component of the geometry-based outlier filter (GF), we conduct an ablation study on the *San Francisco* dataset by varying the distance threshold T_{local} of the derived geometrical constraint in Eq. 4.7. Fig. 4.10 shows the experimental results using different T_{local} settings. All points are classified as *locally visible points* with $T_{local} = \infty$. We can notice that this setting significantly



(A) T_{local}, without GPS (B) T_{local}, with GPS

FIGURE 4.10: The ablation study of the proposed geometry-based outlier filter (GF) on the *San Francisco* dataset using different distance thresholds T_{local} in meter in both with GPS and without GPS scenarios.



(A) λ, without GPS (B) λ, with GPS

FIGURE 4.11: The ablation study of the proposed geometry-based outlier filter (GF) on the *San Francisco* dataset using different angle thresholds λ in both with GPS and without GPS scenarios.

decreases the localization performance. The main reason is that for *non-locally visible points* which can be seen by distant cameras, applying the derived geometrical constraint will result in that many wrong matches can easily satisfy the hybrid inlier evaluation measurement. The resultant matches with $T_{local} = \infty$ usually have a large outlier ratio, which make P3P-RANSAC difficult to obtain a reliable solution.

Looking at Fig. 4.10, it is necessary to define an appropriate distance threshold

TABLE 4.4: The statistics of *locally visible points* with different distance thresholds T_{local} in the *San Francisco* and *Dubrovnik* dataset.

The San Francisco Dataset				
T_{local}	25	50	100	∞
#Locally visible points	16.68M	24.60M	28.21M	30.34M
%Locally visible points	55%	81%	93%	100%
The Dubrovnik Dataset				
T_{local}	25	50	100	∞
#Locally visible points	0.27M	0.87M	1.24M	1.89M
%Locally visible points	14%	46%	66%	100%

T_{local} to ensure that the derived geometrical constraint is accurate for evaluating inliers with respect to *locally visible points*. To achieve this goal, we evaluate three distance thresholds. The statistics of *locally visible points* in the *San Francisco* dataset is shown in Table 4.4. We can notice that by setting $T_{local} = 50m$, 81% of 3D points are classified as *locally visible points*, which is compliant with the characteristics of the *San Francisco* dataset since most of the database images depict street-view scenes. As can be seen in Fig. 4.10A, by setting $T_{local} = 50m$ or $T_{local} = 100m$, our method achieves a significantly gain in both recall and precision comparing with $T_{local} = \infty$. This proves the benefit of the hybrid inlier evaluation measurement in GF. By setting $T_{local} = 25m$, the localization performance is worse than $T_{local} = 50m$ or $T_{local} = 100m$. The reason is that under such setting, several points that should be *locally visible points* are classified as *non-locally visible points* instead, thereby need the classic re-projection error measurement. Comparing with the inlier evaluation measurement using the derived geometrical constraint, the efficiency of classic re-projection error measurement relies more heavily on the quality of matches.

We also evaluate different T_{local} settings when incorporating the GPS data in VF as shown in Fig. 4.10B. The matches generated by VF usually have a larger inlier ratio than without GPS scenario. Therefore, the difference of performance among $T_{local} = 25, 50, 100$ is smaller than the cases without GPS. Applying the derived geometrical constraint to all points with $T_{local} = \infty$ still achieves the worst localization performance. We also evaluate the impact of the angle constraint in GF by varying the angle threshold λ as shown in Fig. 4.11A and Fig. 4.11B. There is a noticeable performance drop when $\lambda = 30°$, which indicates that this threshold is too strict and may reject correct matches. From the ablation study, we can notice that the derived geometrical constraint based on the distances between the camera positions and the *locally visible points* plays a major role in the geometry-based outlier filter.

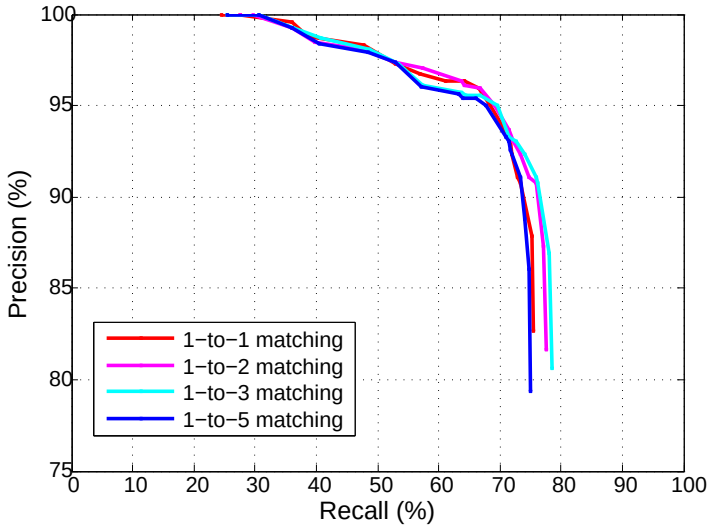


FIGURE 4.12: The localization performances using different 1-to-N matching schemes.

4.4.1.7 Scalability and Efficiency

To evaluate the scalability of our method, we conduct a experiment by varying the number of nearest neighbors in the 1-to-N matching scheme as shown in Fig. 4.12. As can be seen, finding only one nearest neighbor per query feature achieves the lowest recall due to insufficient correct matches. Among all the cases, $N = 2$ or $N = 3$ achieve the best performance, since these seem to provide a good balance between preserving correct matches and rejecting wrong matches. In general, our method shows its effectiveness in dealing with the very large outlier ratio scenario with multiple 1-to-N matching schemes. With 1-to-3 matching scheme, the computational time for the two-stage outlier filter (VF+GF) is close to 0.1 second.

4.4.2 Evaluation on Dubrovnik Dataset

In order to fairly compare with existing works, we adopt two feature matching schemes on the *Dubrovnik* dataset as follows:

- Scheme 1: the 1-to-3 matching scheme which is used in CPV [ZSP15]. Each query feature can find at most three nearest neighbors in the SfM point cloud. An adaptive distance threshold which is defined by 0.7 times the squared distance to the nearest neighbor in the underlying query image is set to reject ambiguous matches.

TABLE 4.5: The comparison of registered query images between our method and the state-of-the-art works on the *Dubrovnik* dataset.

Method	Query Image Statistics		
	#Images	$e < 18.3$	$e > 400$
P3P-RANSAC	628	596	11
Active search [SLK12]	796	704	9
KVD [SEKO17]	798	771	3
CPV [ZSP15]	798	725	2
CPV+P3P [ZSP15]	796	744	7
CPV+P3P+BA [ZSP15]	794	749	13
Toroidal [CSC+17]	800	739	8
DenseVLAD + SfM [STS+17]	-	-	-
PoseNet (GLF) [KC+17]	-	-	-
Our method (Scheme 1)	794	745	4
Our method (Scheme 2)	797	749	3

TABLE 4.6: The localization accuracy, robustness and efficiency comparison between our method and the state-of-the-art works on the *Dubrovnik* dataset. **V+H** means that the corresponding method relies on the prior information about camera's vertical direction and approximate height.

Method	Localization Error e [m]			Assumption	Time[s]
	1^{st} Quarter	Median	3^{rd} Quarter		
P3P-RANSAC	1.30	5.46	8.28	-	11.8
Active search [SLK12]	0.4	1.40	5.30	-	0.25
KVD [SEKO17]	-	0.56	-	**V+H**	5.06
CPV [ZSP15]	0.75	1.69	4.82	**V+H**	3.78
CPV+P3P [ZSP15]	0.19	0.56	2.09	**V+H**	-
CPV+P3P+BA [ZSP15]	0.18	0.47	1.73	**V+H**	-
Toroidal [CSC+17]	0.22	1.07	2.99	-	9.7
DenseVLAD + SfM [STS+17]	0.30	1.00	5.10	-	∼200
PoseNet (GLF) [KC+17]	-	7.9	-	-	0.005
Our method (Scheme 1)	0.29	0.69	2.15	-	2.6
Our method (Scheme 2)	0.28	0.70	2.10	-	1.4

- Scheme 2: each query feature can only find at most one nearest neighbor in the SfM point cloud, a squared distance ratio is set as 0.9 in the SIFT ratio test to reject ambiguous matches. This matching is the same with KVD [SEKO17] and Torodial [CSC+17].

We use the same evaluation criteria as [SLK12, ZSP15, SEKO17, CSC+17] to evaluate the localization result: a query image is successfully localized if the best camera pose returned by RANSAC has more than 11 inliers. The re-projection error threshold

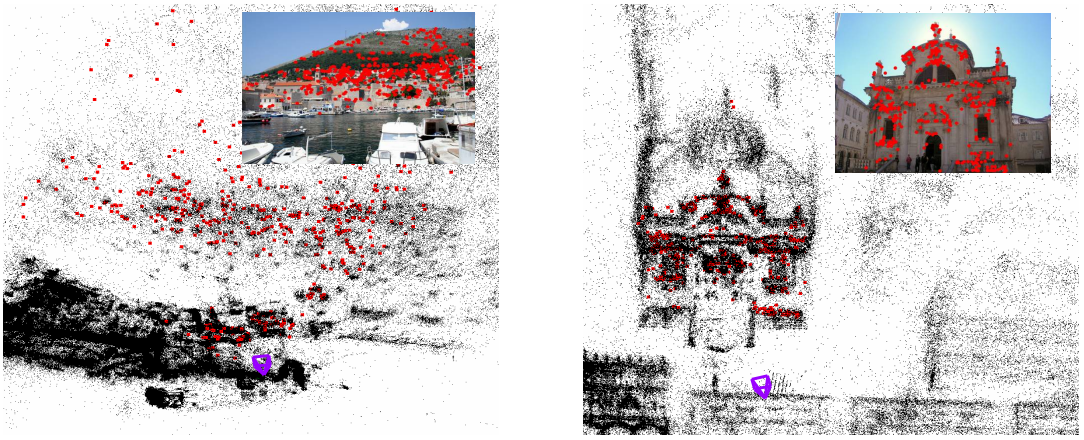


FIGURE 4.13: The exemplary query images with corresponding estimated 6-DOF camera poses and localization errors of the *Dubrovnik* dataset.

is set as 6 pixels. The pose accuracy can be measured with the ground truth 6-DOF camera poses provided by Li *et al.* [LSH10]. In the *Dubrovnik* dataset, we select the top 20 database images in the first stage to apply the visibility-based outlier filter.

Table 4.5 shows the localization performance of our method and other related works on the *Dubrovnik* dataset. Under Scheme 2, we achieve a slightly better performance considering the number of successfully localized images comparing with Scheme 1. This indicates that the matches established with Scheme 2 contain sufficient correct matches in the *Dubrovnik* dataset for an accurate pose estimation. As shown in Table 4.6, comparing with other methods that do not need any additional geometrical priors [SLK12, CSC+17, STS+17, KC+17], we achieve the state-of-the-art performance on the median and 3^{rd} quarter pose accuracy, and a comparable performance on the 1^{st} quarter pose accuracy comparing with the Toroidal approach [CSC+17]. Comparing with

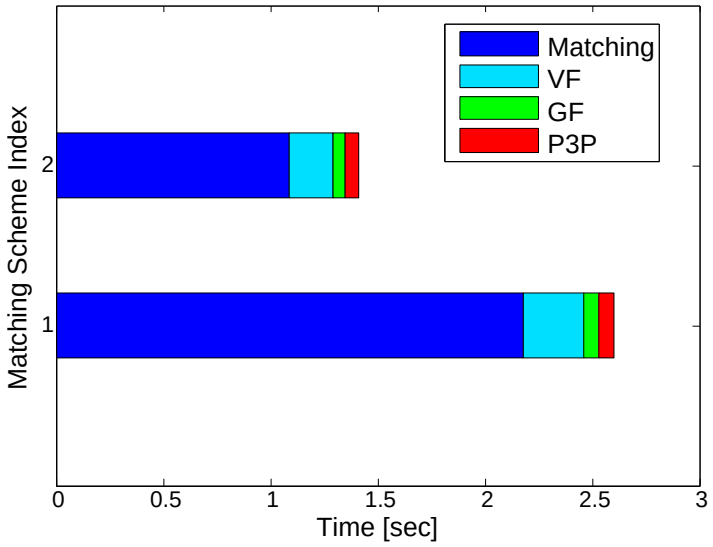


FIGURE 4.14: The computational time of our method with Scheme 1 and Scheme 2 on the *Dubronik* dataset (also reported in Table 4.6).

the methods [ZSP15, SEKO17] that rely on the assumption of the camera's vertical direction and approximate height, we are able to achieve competitive results. Fig. 4.13 shows the exemplary estimated 6-DOF camera poses in the *Dubrovnik* dataset using our method. In addition, our method has the third lowest computational time among all existing methods. The Active search [SLK12] method is efficient by establishing at most 100 2D-3D matches for each query image, which in the meantime reduces the pose accuracy. Fig. 4.14 gives the details of our method's computational time. As can be seen, the feature matching step occupies the majority of the computational time. The proposed visiblity-based outlier filter (VF) and the geometry-based outlier filter (GF) can be efficiently executed in less than half a second.

4.5 Summary

In this chapter, we study the match disambiguation problem to improve city-scale urban image-based localization. To this end, we propose a two-stage outlier filtering framework with a visibility-based outlier filter and a subsequent geometry-based outlier filter. We propose a novel re-ranking method to improve the relevancy of top ranked database images, which can serve as a proxy to filter wrong matches. We also present a match augmentation scheme to recover correct matches lost due to SIFT ratio test. For urban-like SfM point clouds, we propose a novel geometrical constraint to enhance the geometry-based outlier filter, especially in large outlier ratio scenarios. Comparing

with recent advanced outlier filtering approaches, our method does not rely on any geometrical prior from other additional sensors. We experimentally evaluate the proposed framework on two urban SfM datasets including the challenging *San Francisco* dataset, and the results show that the proposed framework is a prior-free, efficient and effective method for urban image-based localization.

Chapter 5

Accurate Image-based Localization under Binary Feature Representation

With the previous two chapters, we have shown how to separately handle the memory consumption (especially in extremely limited resources) and match filtering (especially in urban environments) problems for large-scale image-based localization. In this chapter, we make a step further by introducing a general framework [CLCZ19][1] that simultaneously addresses the memory consumption, match disambiguation and localization accuracy problems. In order to make the image-based localization system memory-efficient, our framework leverages a binary feature representation via Hamming Embedding [JDS08] on a visual vocabulary. To solve the challenging match disambiguation problem under binary feature representation, we use a cascade combining three types of match filters to disambiguate matches in a coarse-to-fine fashion. Meanwhile, the match disambiguation task is separated into two parallel tasks before deriving an auxiliary camera pose for final filtering. One task focuses on preserving potentially correct matches, while another focuses on obtaining high quality matches to facilitate subsequent match filtering. In addition, our framework improves the localization accuracy by a quality-aware spatial reconfiguration method for 2D-3D matches and a principal focal length selection method for RANSAC. Table 5.1 illustrates the

[1] **Wentao Cheng**, Weisi Lin, Kan Chen, Xinfeng Zhang, Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization. International Conference on Computer Vision, 2019.

Method	Feature Type	Match Disambiguation			Prior-free	SR
		Feature	Visibility	Geometry		
AS [SLK12]	SIFT	Strict	Yes	No	Yes	No
WPE [LSHF12]	SIFT	Relaxed	Yes	No	Yes	No
CSL [SEKO17]	SIFT	Relaxed	No	Yes	No*	No
CPV [ZSP15]	SIFT	Relaxed	No	Yes	No*	No
HFV [SHR+15]	SIFT	Strict	Yes	No	Yes	In RPE
EGM [LLD17]	SIFT+Binary	Relaxed	Yes	No	Yes	No
TC [CSC+17]	SIFT	Relaxed	No	Yes	Yes	No
SMC [TSH+18]	SIFT	Relaxed	No	Yes	No*	No
Ours	Binary	Relaxed	Yes	Yes	Yes	Before RPE

TABLE 5.1: Comparison of our proposed image-based localization framework to other image-based localization methods. No* means that the vertical direction of camera is known in advance, and RPE represents RANSAC-based Pose Estimation. SR represents spatial reconfiguration.

key conceptual differences between our image-based localization framework and state-of-the-art methods. We test our method on several widely used real-world datasets, and the experimental results show that we achieve the best localization accuracy with significantly lower memory requirements. Fig. 5.1 shows the complete pipeline of the proposed image-based localization framework. From Section 5.1 to Section 5.3, we describe three types of match disambiguation used in our framework in detail. In Section 5.4, we give a comprehensive evaluation and comparison considering effectiveness, compactness and accuracy.

5.1 Feature-wise Match Disambiguation

We start from feature-wise match disambiguation. Concretely, a feature-wise score is computed for each match. By setting a relaxed criteria, we reject obviously wrong matches to retain a feature-wise match pool that preserves potentially correct matches. Also, we use a strict criteria to obtain a set of feature-wisely confident matches to facilitate the subsequent visibility-wise match disambiguation.

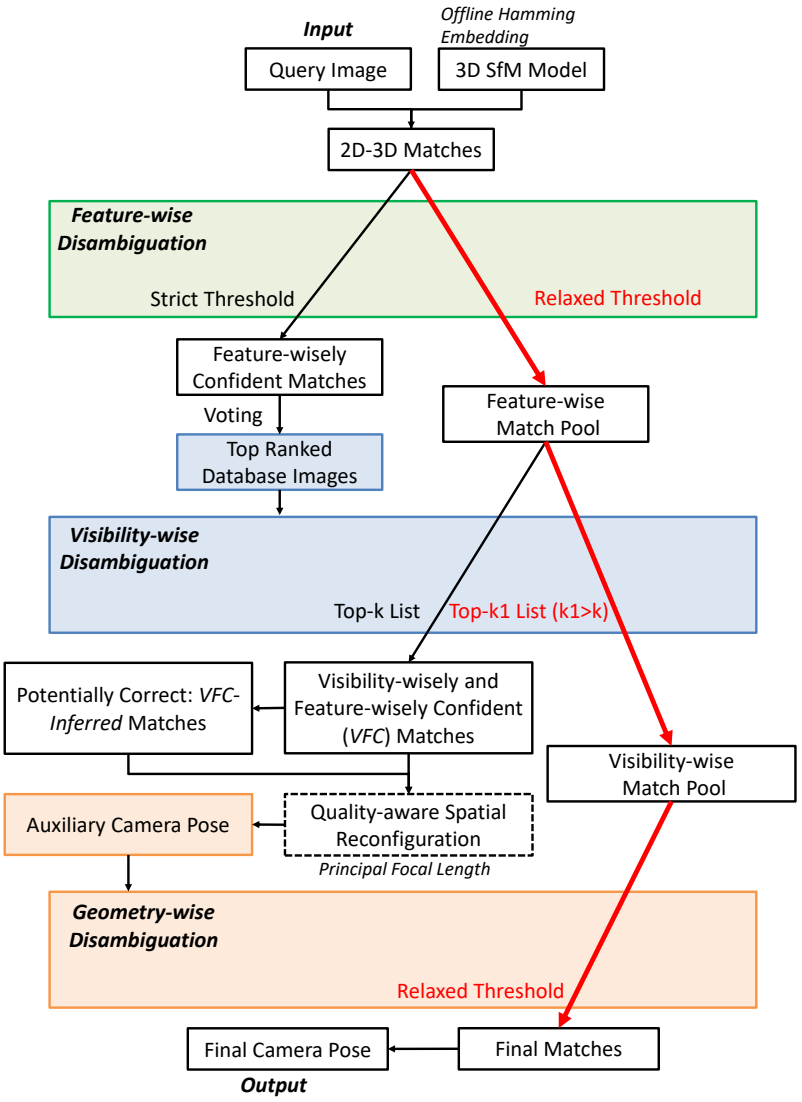


FIGURE 5.1: Overview of the pipeline of our proposed image-based localization framework.

5.1.1 Data Pre-processing

Let $\mathcal{P}$ be the 3D points in an SfM model. Each 3D point is associated with a set of SIFT descriptors. We first train a general visual vocabulary through k-means clustering algorithm. In the offline stage, the descriptors of a 3D point are assigned to their closest visual words through nearest neighbor search. For efficiency, we follow [SLK12] by representing the SIFT descriptors of a 3D point as integer mean descriptor per visual word. Subsequently, each integer mean descriptor is converted into a compact binary signature containing B bits using Hamming embedding [JDS08]. Given a query image, a set of SIFT descriptors are extracted, denoted as $\mathcal{Q}$. For each descriptor

$q \in \mathcal{Q}$, we first assign it to its closest visual word. With Hamming embedding, we also obtain the binary signature for descriptor q, denoted as s_q. For each 3D point $p \in \mathcal{P}$, if one of its associated integer mean descriptors is quantized into the same visual word with query descriptor q, a 2D-3D match can be established as $m = \{q \leftrightarrow p\}$. The Hamming distance of m can be measured as $h(s_q, s_p)$.

5.1.2 Bilateral Hamming Ratio Test

To evaluate the distinctiveness of the resultant 2D-3D matches, previous works mainly focus on the SfM model side by using a fixed Hamming distance threshold [SWLK12], a Gaussian weighting function [JPD+12], or density estimation [AZ14]. Few attentions have been paid on disambiguating matches on the query image side, where the corresponding feature space is easier to distinguish correct matches due to its sparsity. Inspired from Lowe's ratio test [Low04] for SIFT feature descriptor, we propose a bilateral Hamming ratio test that operates on both the query image and SfM model.

In order to prevent correct matches from being rejected in this step, we apply a coarse disambiguation scheme by using a large Hamming distance threshold τ. Therefore, for a match $m = \{q \leftrightarrow p\}$, the set of 3D points that can form a match with query descriptor q can be defined as $\mathcal{P}(q) = \{p \in \mathcal{P} | h(s_q, s_p) \leq \tau\}$. Similarly, the set of query descriptors that can form a match with 3D point p can be represented as $\mathcal{Q}(p) = \{q \in \mathcal{Q} | h(s_q, s_p) \leq \tau\}$. Our core idea is that a match should be distinctive if its corresponding Hamming distance is significantly lower than the average Hamming distance in $\mathcal{P}(q)$ and $\mathcal{Q}(p)$. To evaluate a match within the feature space of a query image, we apply an image side Hamming ratio test as follows:

$$t(m) = \frac{\sum_{j \in \mathcal{Q}(p)} h(s_j, s_p)}{h(s_q, s_p) |\mathcal{Q}(p)|^2}, \tag{5.1}$$

where one $|\mathcal{Q}(p)|$ in $|\mathcal{Q}(p)|^2$ is used to compute the average Hamming distance, and another is to penalize a match whose corresponding 3D point establish too many matches with different query descriptors. We notice that it is safe to reject a match when it is ambiguous in the feature space of query image. Therefore, we define a threshold φ, and reject matches if their corresponding scores are smaller than φ.

Similarly, to evaluate the distinctiveness of a match within the feature space of an

SfM model, we apply the model side Hamming ratio test as follows:

$$t'(m) = \frac{\sum_{j \in \mathcal{P}(q)} h(s_q, s_j)}{h(s_q, s_p)\,|\mathcal{P}(q)|}. \tag{5.2}$$

Since the term $|\mathcal{P}(q)|$ may vary dramatically with using different size of visual vocabularies, here we don't use it to penalize a match whose corresponding query descriptor can establish multiple matches with different 3D points. In addition, an SfM model usually contains orders of magnitude more descriptors than an image. This makes the model side Hamming ratio test prone to reject correct matches by directly setting a hard threshold. Therefore, we only apply $t'(m)$ as a soft scoring function to evaluate a match. The final bilateral Hamming ratio test can be defined as follows:

$$T(m) = \begin{cases} t'(m), & t(m) \geq \varphi \\ 0, & \text{otherwise.} \end{cases} \tag{5.3}$$

5.1.3 Aggregating Gaussian Weighting Function

In order to strengthen the feature distinctiveness, we propose an adapted version of Gaussian weighting function [JDS09] as follows:

$$w(h) = \begin{cases} (\frac{\sigma}{h})^2 e^{-(\frac{h}{\sigma})^2}, & 0.5\sigma < h \leq \tau \\ 4e^{-0.25}, & 0 < h \leq 0.5\sigma \\ 0, & \text{otherwise,} \end{cases} \tag{5.4}$$

where h is the Hamming distance of a match, and σ is usually set to one quarter of the binary feature dimension [AZ14]. By aggregating the Gaussian weighting function, the score for a match m therefore can be computed as follows:

$$E(m) = T(m)w(h(m)). \tag{5.5}$$

Overall, we can retain a feature-wise match pool $\mathcal{M} = \{m|E(m) > 0\}$, which focuses on preserving correct matches. We also obtain a set of Feature-wisely Confident (FC) matches $\mathcal{M}_{FC} = \{m|E(m) > \alpha\}, \alpha > 0$.

5.2 Visibility-wise Match Disambiguation

Given the feature-wise match pool $\mathcal{M}$ and Feature-wisely Confident (FC) matches $\mathcal{M}_{FC}$, here we describe how to leverage the visibility information in an SfM model to further disambiguate matches. In particular, we aim to achieve two purposes at this stage: 1) to reject wrong matches in $\mathcal{M}$ to retain a visibility-wise match pool that well preserves correct matches, 2) to select a set of high quality matches that are substantial to derive an auxiliary camera pose for later geometry-wise disambiguation.

The visibility information encoded in an SfM model can be represented as a bipartite visibility graph $\mathcal{G} = \{\mathcal{P}, \mathcal{D}, \mathcal{E}\}$. Each node $p \in \mathcal{P}$ represents a 3D point, and each node $d \in \mathcal{D}$ represents a database image. An edge $(p, d) \in \mathcal{E}$ exists if point p is observed in database image d. Intuitively, correct matches usually cluster in the database images that are relevant, a.k.a., visual overlap, to a given query image. Thus, the problem of disambiguating matches can be transferred as a problem of finding relevant database images.

5.2.1 Voting with *FC* Matches

Using the visibility graph $\mathcal{G}$, a 2D-3D match $m = \{q \leftrightarrow p\}$ can cast a vote to each database image that observes point p. In order to prevent ambiguous matches from interfering the voting procedure, we only use FC matches to vote database images. Inspired from [SHR+15], we also enforce a locally unique voting scheme. Let $\mathcal{M}_{FC}^{d} = \{m = \{q \leftrightarrow p\} \,|m \in \mathcal{M}_{FC}, (p, d) \in \mathcal{E}\}$ be the FC matches that vote for database image d. We enforce that a match for database image d can be added to $\mathcal{M}_{FC}^{d}$ only if its corresponding query descriptor has not appeared in $\mathcal{M}_{FC}^{d}$ before. In addition, we only consider database images that receive at least three votes to ensure high relevancy to the query image. After accumulating the match scores for a database image, we add a term frequency weight in order to penalize database images that observe a large number of 3D points. Let $\mathcal{P}^{d} = \{p|(p, d) \in \mathcal{E}\}$ be the set of 3D points that are observed by the database image d, the voting score can be defined as follows:

$$\mathcal{S}(d) = \frac{\sum_{m \in \mathcal{M}_{FC}^{d}} E(m)}{\sqrt{|\mathcal{P}^{d}|}}. \tag{5.6}$$

A larger voting score inherently indicates that the corresponding database image is more relevant to a given query image, hence more likely to find correct matches. We

first retrieve top-k ranked database images $d(k)$ with the largest voting scores. For a match $m \in \mathcal{M}$, we select it into the set $\mathcal{M}^{d(k)}$ if its corresponding 3D point is observed in at least one of the images in $d(k)$. Note that only visibility information is considered and we preserve both *FC* and *non-FC* matches in $\mathcal{M}^{d(k)}$. Similarly, we apply a relaxed criteria by using a larger k_1 to select another set of matches $\mathcal{M}^{d(k_1)}$, which may contain more correct matches but also are more noisy than $\mathcal{M}^{d(k)}$. $\mathcal{M}^{d(k_1)}$ will serve a visibility-wise match pool and later be disambiguated in Section 5.3.

5.2.2 Two-step Match Selection

Naturally, we can define the matches in $\mathcal{M}^{d(k)}$ as *Visibility-wisely Confident (VC)* matches. Due to the existence of feature-wisely ambiguous matches, *VC* matches may contain a large portion of outliers, making them difficult to compute an auxiliary camera pose. We propose a two-step match selection method to disambiguate *VC* matches. In the first step, we select the *FC* from *VC* matches as *Visibility-wisely and Feature-wisely Confident (VFC)* matches that can be defined as follows:

$$\mathcal{M}_{VFC}^{d(k)} = \left\{ m | m \in \mathcal{M}^{d(k)} \wedge E(m) \geq \alpha \right\}. \tag{5.7}$$

The *VFC* matches exhibit high confidence to be correct since they not only are observed in top ranked database images, but also are highly distinctive in feature space. The major difficulty is how to distinguish correct matches from the rest *Visibility-wisely but Not Feature-wisely Confident (VNFC)* matches that can be defined as follows:

$$\mathcal{M}_{VNFC}^{d(k)} = \mathcal{M}^{d}(k) \setminus \mathcal{M}_{VFC}^{d(k)}. \tag{5.8}$$

During the image voting procedure, we leverage the point-image relationship of the bipartite visibility graph $\mathcal{G}$. Here we will use the point-point relationship in $\mathcal{G}$ to help us disambiguate the *VNFC* matches. Intuitively, if a 3D point of one *VNFC* match exhibits a strong co-visibility relationship with 3D points of *VFC* matches in top ranked database images, it should be regarded as a potentially correct match. To this end, we engage the second step match selection to infer potentially correct matches from *VNFC* matches. For each database image $d \in d(k)$, we first count the number of *VFC* matches and *VNFC* matches, which we call ω_{VFC}^{d} and ω_{VNFC}^{d} respectively. If *VFC* matches occupy a larger portion compared with *VNFC* matches

Algorithm 4: Visibility-wise Match Disambiguation

Require: Matches $\mathcal{M}$ with feature-wise match scores $E(m)$, match score threshold α
Require: $\mathcal{M}_{VFC}^{d(k)} \leftarrow \emptyset$, $\mathcal{M}_{VFC\text{-}I}^{d(k)} \leftarrow \emptyset$,$\mathcal{M}^{d(k_1)} \leftarrow \emptyset$
/* explore **point-image** visibility */
Apply image voting with *FC* matches using Eq. 5.6
Retrieve top-k and top-k_1 ranked database images $d(k)$ and $d(k_1)$
Select all matches in $d(k_1)$ as $\mathcal{M}^{d(k_1)}$ for visibility-wise match pool
Select *VFC* matches $\mathcal{M}_{VFC}^{d(k)}$ with $d(k)$
/* explore **point-point** visibility */
for All $d \in d(k)$ **do**
    Compute the number of *VFC* matches ω_{VFC}^{d}
    Compute the number *VNFC* matches ω_{VNFC}^{d}
    for All $m \in \mathcal{M}_{VNFC}^{d}$ **do**
        update the match score $E(m)$ using Eq. 5.9
    end for
end for
for All $m \in \mathcal{M}_{VNFC}^{d(k)}$ **do**
    if $E(m) \geq \alpha$ **then**
        $\mathcal{M}_{VFC\text{-}I}^{d(k)} \leftarrow \mathcal{M}_{VFC\text{-}I}^{d(k)} \cup \{m\}$
    end if
end for
return $\mathcal{M}_{VFC}^{d(k)} \cup \mathcal{M}_{VFC\text{-}I}^{d(k)}$ and $\mathcal{M}^{d(k_1)}$

in one database image, each *VNFC* match should receive stronger promotion from *VFC* matches respectively. Therefore, for an *VNFC* match visible in database image d, we increase its match score as follows:

$$E(m) = E(m) + \frac{\alpha}{2} \ln(1 + \frac{\omega_{VFC}^{d}}{\omega_{VNFC}^{d}}). \tag{5.9}$$

After updating the scores for *VNFC* matches in all database images in $d(k)$, the larger the updated score, the more likely that corresponding *VNFC* match is correct. Using the previous match score threshold α, we can select a set of potentially correct matches from *VNFC* matches. Since these potentially correct matches are mainly inferred by exploring the visibility information with *VFC* matches, we call them *VFC-I* matches and they can be defined as follows:

$$\mathcal{M}_{VFC-I}^{d(k)} = \left\{ m | m \in \mathcal{M}_{VNFC}^{d(k)} \wedge E(m) \geq \alpha \right\}. \tag{5.10}$$

Therefore, the matches that we select from $\mathcal{M}^{d(k)}$ are the union of *VFC* and *VFC-I* matches. Algorithm 4 illustrates the process of visibility-wise match disambiguation.

5.3 Geometry-wise Match Disambiguation

In this section, we describe how to use the obtained *VFC* and *VFC-I* matches to compute an auxiliary camera pose, which facilitates geometry-wise match disambiguation in the visibility-wise match pool $\mathcal{M}^{d(k_1)}$.

5.3.1 Quality-aware Spatial Reconfiguration

A common way to estimate a camera pose is to use pose solvers inside RANSAC loops. The quality of input 2D-3D matches to pose solvers, i.e. the inlier ratio, is an essential factor for a robust and efficient camera pose estimation. It is also important to ensure that the input matches have a uniform spatial distribution. This becomes essentially important when the majority of input matches cluster in a highly textured region as shown in Fig. 5.2. Correct matches, rare but critical, in poorly textured regions are unlikely to be sampled in RANSAC hypothesis stage. This will significantly reduce the localization accuracy due to the difficulty of obtaining a non-degenerate pose hypothesis.

Our goal is to obtain a set of matches that simultaneously has a large inlier ratio and a uniform spatial distribution by selecting from *VFC* and *VFC-I* matches. To this end, we first divide the query image into 4×4 equally-sized bins, denoted as $\mathcal{B}$. The *VFC* and *VFC-I* matches are then quantized into $\mathcal{B}$ according to the image coordinates of their associated 2D query descriptors. To make the spatial distribution of selected matches more uniform, we apply a spatial reconfiguration method to penalize dense bins that have a large number of quantized matches and emphasize sparse bins that have few quantized matches. Let N_b be the number of matches that are quantized into bin $b \in \mathcal{B}$. Let R_b be the proportion of matches that can be selected from bin b, the spatial reconfiguration can be realized by computing R_b as follows:

$$R_b = \frac{\sqrt{N_b}}{\sum_{i \in \mathcal{B}} \sqrt{N_i}}. \tag{5.11}$$

To achieve an efficient camera pose estimation, we limit that overall at most N matches can be selected. Accordingly, for each bin b, the match selection quota is $R_b N$.

We now start to select the *VFC* and *VFC-I* matches. Starting with the first ranked database image, we first select the corresponding *VFC* matches according to each bin's

selection quota. After that, if there exist bins that still do not reach the selection quotas, we then select the *VFC-I* matches from these bins. Note that the *VFC-I* matches exhibit inferior quality than the *VFC* matches because of their confidence in only visibility. To ensure high quality of selected matches, the *VFC* matches should be dominant. Suppose the number of selected *VFC* matches is N_{VFC}, we restrict that at most $\beta N_{VFC}, \beta < 1$ *VFC-I* matches can be selected. We denote the selected matches after quality-aware spatial reconfiguration as $\mathcal{M}_s$. The quality-aware spatial reconfiguration is illustrated in Algorithm 5.

5.3.2 Auxiliary Camera Pose with Principal Focal Length

We then use the selected matches after quality-aware spatial reconfiguration to compute an auxiliary camera pose. Assuming a general scenario when the focal length of a given query image is unknown, we can adopt a 4-point pose solver (P4P) [BKP08] to estimate the extrinsic calibration and the focal length. In RANSAC-based camera pose estimation, the estimated camera pose usually is the pose hypothesis that is supported by the largest number of inliers. However, we noticed that this strategy becomes unreliable when few correct matches exist. In such case, a co-planar degenerated sample may result in that the estimated camera will lie far away from the scene with an unrealistic focal length. To tackle this unreliability problem, we propose a statistical verification scheme to find a reliable camera pose. Let ε be the largest number of inliers of a pose hypothesis after running a certain number of RANSAC+P4P loops. We store the top-10 pose hypotheses, whose corresponding inliers are more than 0.7ε. For a successful localization, we notice that most of the top hypotheses have numerically close focal length values. These focal length values, instead of the one with largest number of inliers, provide us a more stable and reliable camera pose estimation. Therefore, we propose to select the pose hypothesis whose focal length is the median value among the top pose hypotheses. We define the selected pose hypothesis $\mathcal{A}$ as an auxiliary camera pose, and its corresponding focal length as principal focal length f.

Disambiguation using Auxiliary Camera Pose. The computed auxiliary camera pose exhibits sufficient accuracy. Using it to recover potentially correct matches back can further improve the localization accuracy. We apply the auxiliary camera pose on the visibility-wise match pool $\mathcal{M}^{d(k_1)}$ to realize the geometry-wise disambiguation.

Algorithm 5: Quality-aware Spatial Reconfiguration

Require: top-k ranked database images $d(k)$, $\mathcal{M}_{VFC}^{d(k)}$, $\mathcal{M}_{VFC-I}^{d(k)}$
Require: 4×4 bins $\mathcal{B}$, match set threshold N, allocation ratio β, $\mathcal{M}_s \leftarrow \emptyset$, number of selected VFC matches $N_{VFC} \leftarrow 0$, number of selected $VFC-I$ matches $N_{VFC-I} \leftarrow 0$

/*__Bin-based spatial reconfiguration__*/
for all $b \in \mathcal{B}$ **do**
  Compute the proportion R_b using Eq. 5.11
  Occupied number $u_b \leftarrow 0$
end for
/*__Set all matches unchosen__*/
for all $m \in \mathcal{M}_{VFC}^{d(k)} \cup \mathcal{M}_{VFC-I}^{d(k)}$ **do**
  Chosen flag $c_m \leftarrow 0$
end for
/*__Pick VFC matches first__*/
for all $d \in d(k)$ **do**
  for all $m \in \mathcal{M}_{VFC}^{d}$ **do**
    if $c_m = 0$ **then**
      Obtain the bin index b for m
      /*__Do not exceed the bin quota__*/
      if $u_b < R_b N$ **then**
        $\mathcal{M}_s \leftarrow \mathcal{M}_s \cup \{m\}, c_m \leftarrow 1,$
        $N_{VFC} \leftarrow N_{VFC} + 1, u_b \leftarrow u_b + 1$
      end if
    end if
  end for
end for
/*__Pick VFC-I matches__*/
for all $d \in d(k)$ **do**
  for all $m \in \mathcal{M}_{VFC-I}^{d}$ **do**
    /*__Ensure a good match quality__*/
    if $c_m = 0$ and $N_{VFC}/N_{VFC-I} < \beta$ **then**
      Obtain the bin index b for m
      /*__Do not exceed the bin quota__*/
      if $u_b < R_b N$ **then**
        $\mathcal{M}_s \leftarrow \mathcal{M}_s \cup \{m\}, c_m \leftarrow 1,$
        $N_{VFC-I} \leftarrow N_{VFC-I} + 1, u_b \leftarrow u_b + 1$
      end if
    end if
  end for
end for
return $\mathcal{M}_s$

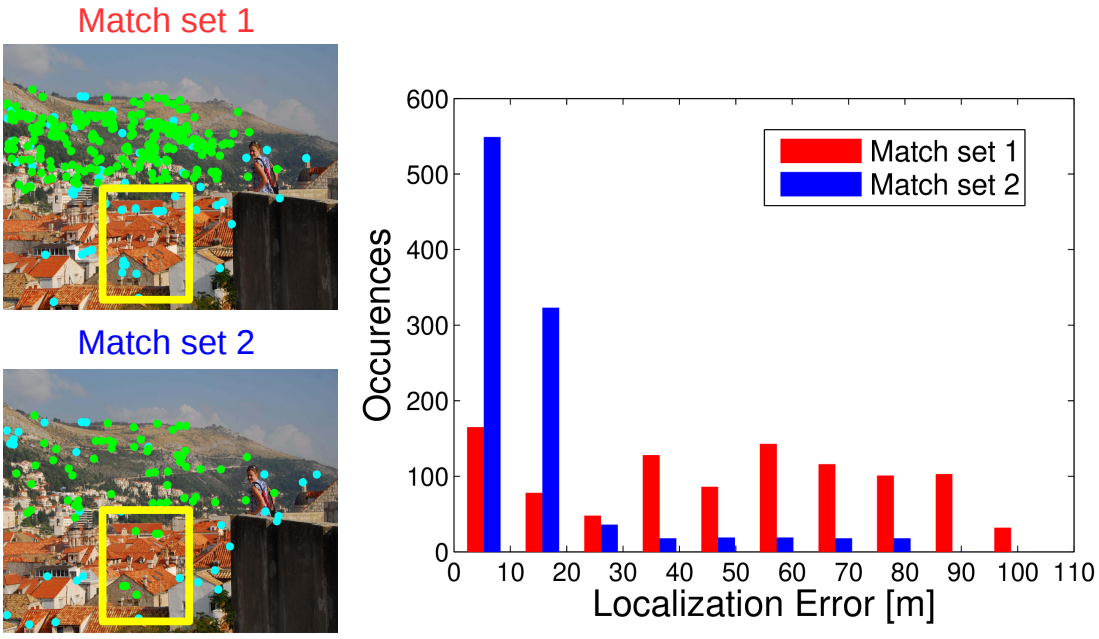


FIGURE 5.2: The influence of a uniform spatial distribution for matches. **Left top:** Original match set with 242 inliers shown in green and 64 outliers shown in cyan (inlier ratio is 0.79), matches are clustered in mountain area; **Left bottom:** a selection from original match set by applying spatial reconfiguration, this selection has 63 inliers and 31 outliers (inlier ratio is 0.67), matches are more uniformly distributed over the image; **Right:** Localization error statistics with these two match sets by running 1000 camera pose estimation trials. **Yellow box:** the inside correct but sparse matches are emphasized in match set 2.

We define a relaxed re-projection error threshold θ in case rejecting potentially correct matches. For a match $\{q \leftrightarrow p\} \in \mathcal{M}^{d(k_1)}$, let X_q and Y_p denote the coordinates of q in 2D space and p in 3D space respectively. As such, a match can be selected as a potentially correct match if the re-projection error is below the defined threshold as follows:

$$\|\mathcal{A}Y_p - X_q\|_2 \leqslant \theta. \tag{5.12}$$

Final Camera Pose Estimation. The matches selected by the auxiliary camera pose exhibit both high quality and high quantity. In addition, we have also obtained a reliable focal length value f. Based on these, we can directly apply a 3-point pose solver (P3P) [KSS11], which is much more efficient than 4-point pose solvers, to compute the final camera pose. The process of match disambiguation using auxiliary camera pose is illustrated in Algorithm 6.

Algorithm 6: Disambiguation using Auxiliary Camera Pose

Require: Re-projection error threshold θ, auxiliary camera pose $\mathcal{A}$, $\mathcal{I} \leftarrow \emptyset$
Require: $\mathcal{M}_s$: selected matches after quality-aware spatial reconfiguration
Require: Visibility-wise match pool $\mathcal{M}^{d(k_1)}$
1: Run RANSAC + P4P solver with $\mathcal{M}_s$
2: Obtain the largest inlier number ε_{max}
3: Store the top-10 pose hypotheses with $\varepsilon \geq 0.7\varepsilon_{max}$
4: Select the pose hypothesis $\mathcal{P}$ with median value focal length f
5: **for** all $m \in \mathcal{M}^{d(k_1)}$ **do**
6: **if** $\|\mathcal{A}Y_p - X_q\|_2 \leqslant \theta$ **then**
7: $\mathcal{I} \leftarrow \mathcal{I} \cup \{m\}$
8: **end if**
9: **end for**
10: Run final camera pose estimation using $\mathcal{I}$
11: **return** the pose hypothesis with largest number of inliers

TABLE 5.2: Summarization of the datasets used in the experiments.

Dataset	Image Capture	# 3D Points (# Sub-models)	# Images		6-DOF Query Poses
			Database	Query	
Dubrovnik	Free Viewpoint	1.89M (1)	6,044	800	SfM
Rome	Free Viewpoint	4.07M (69)	15,179	1,000	-
RobotCar Seasons	Trajectory	6.77M (49)	20,862	11,934	LIDAR Registered
Aachen Day-Night	Free Viewpoint	1.65M (1)	4,328	922	SfM

5.4 Experiments

5.4.1 Datasets and Evaluation Metrics

We evaluate our proposed image-based localization framework on four real-world datasets as summarized in Table 5.2. The Dubrovnik [LSH10] and Rome [LSH10] datasets were reconstructed from Internet photos via SfM techniques. The Dubrovnik dataset depicts a historical city with one single SfM model, while the Rome dataset depicts several landmarks with 69 unconnected SfM sub-models. For the Dubrovnik and Rome datasets, we adopt the same evaluation metric used in related works [LSH10, SLK11, SLK12, ZSP15, SEKO17, CSC+17, LLD17]. A query image is considered as successfully registered or localized if the best camera pose after RANSAC has at least 12 inliers. For the Dubrovnik dataset, the ground truth camera poses of query images were obtained from an SfM model reconstructed from all database and query images. The localization accuracy on the Dubrovnik dataset can be measured as the distance between estimated camera center position and the ground truth camera center position

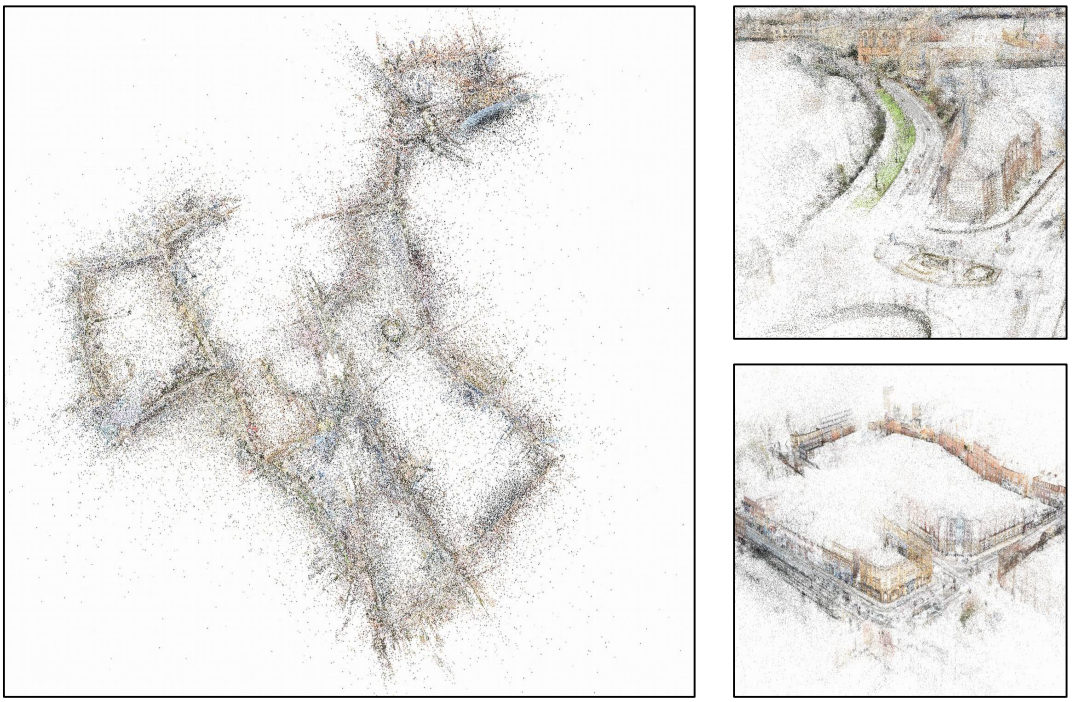


FIGURE 5.3: Left: the visualization of the RobotCar Seasons dataset [SMT+18]. Right: two zoomed-in visualizations.

of query image.

The RobotCar Seasons [SMT+18] dataset was reconstructed from images that were captured with cameras mounted on an autonomous vehicle. The RobotCar Seasons dataset covers a wide range of condition changes, *e.g.* weather, seasons, day-night, which make image-based localization on this dataset challenging. The ground truth camera poses of query images were obtained by aligning all 49 SfM sub-models to LIDAR point clouds. The visualization of this dataset is shown in Fig. 5.3. The Aachen Day-Night [SMT+18] dataset is an extension from the original Aachen dataset [SWLK12] by including carefully checked SfM ground truth poses for query images. To obtain the ground truth poses for night-time query images, they used hand-labeled 2D-3D matches by manually selecting a day-time query image from a similar viewpoint for each night-time query image. Fig. 5.4 shows several exemplary images in the RobotCar Seasons and Aachen Day-Night datasets. We follow the evaluation metric in [SMT+18] and report the percentage of query images localized within Um and V° from ground truth camera poses. To evaluate under different levels of localization accuracy, we use the three accuracy intervals defined in [SMT+18] as follows: High-precision $(0.25m, 2^\circ)$, Medium-precision $(0.5m, 5^\circ)$ and Coarse-precision $(5m, 10^\circ)$.

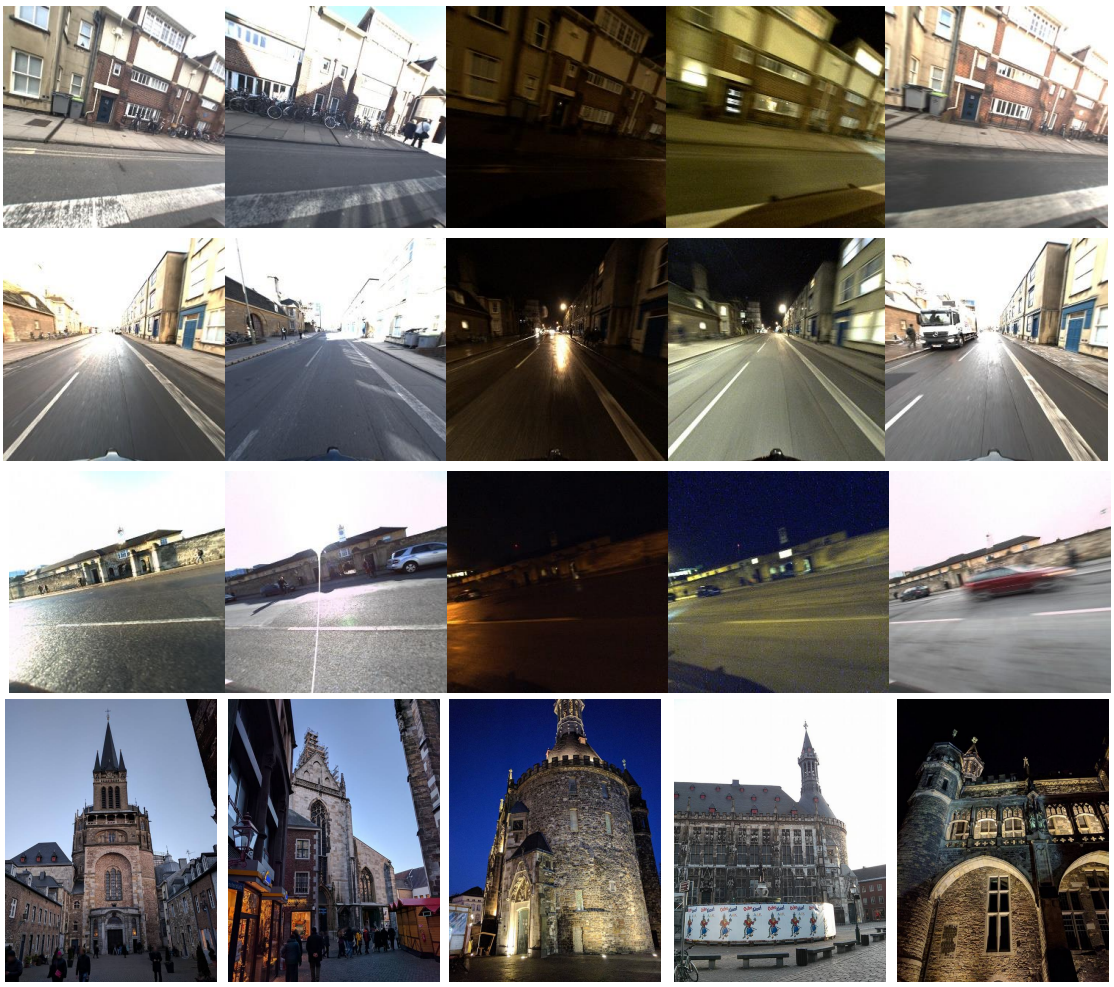


FIGURE 5.4: Exemplary images in the RobotCar Seasons (top three rows) and Aachen Day-Night (last row) datasets [SMT+18].

5.4.2 Implementation Details

For the Dubrovnik and Rome datasets, we use the same vocabulary containing 10k visual words trained by [SLK12]. For the RobotCar Seasons and Aachen Day-Night datasets, we follow [SMT+18] by training a vocabulary containing 10k visual words on all upright RootSIFT descriptors found in 1000 randomly selected database images in the reference SfM model. Our proposed method involves several parameters. For all datasets used in the experiments, we set $B = 64$, $\sigma = 16$, $\varphi = 0.3$ and $\alpha = 0.8$ for the feature-wise disambiguation step. In the visibility-wise disambiguation step, we set $k = 20$. In the geometry-wise disambiguation step, we set $N = 100$, $\beta = 0.33$, $\theta = 10$ pixels. The Hamming distance threshold τ in Eq. 5.4 is set to 19 for the Dubrovnik, Rome and RobotCar Seasons datasets. Since the query images in the Aachen Day-Night dataset are significantly more textured, we found that setting $\tau = 16$ is sufficient

TABLE 5.3: The comparison between our method and state-of-the-art methods on the Dubrovnik dataset

Method	Error Quartiles [m]			Localized images
	25%	50%	75%	
EGM [LLD17]	0.24	0.70	2.67	794
TC [CSC+17]	**0.22**	1.07	2.99	**800**
AS [SLK12]	0.40	1.40	5.30	796
Our method	**0.22**	**0.64**	**2.16**	794

to establish enough correct 2D-3D matches. In addition, we set $k_1 = 50$ for the Aachen Day-Night dataset and $k_1 = 100$ for other datasets. For computing the auxiliary camera pose and the final camera pose, we run both 1000 RANSAC iterations. For a fair comparison on the Dubrovnik and Rome datasets, we use a threshold of 4 pixels for final pose estimation. For a fair comparison on the RobotCar Seasons and Aachen Day-Night datasets, we use a 3-point pose solver to compute the auxiliary camera pose and a threshold of 4 pixels for final pose estimation. All experiments were conducted with a single CPU thread on a PC with an Intel i7-6800K CPU with 3.40 GHz and 32 GB RAM.

5.4.3 Comparison with State-of-the-art

We compare our method with state-of-the-art image-based localization methods such as EGM [LLD17], AS [SLK12] and TC [CSC+17]. Note that our comparison excludes CSL [ZSP15], CPV [ZSP15] and SMC [TSH+18], since their experimental results were obtained with synthetic data (they extracted the gravity direction from ground truth camera poses). On the RobotCar Seasons and Aachen Day-Night datasets, we also compare with two more image retrieval-based methods, namely DenseVLAD [AZ13] and NetVLAD [AGT+16]. They represent each image with global descriptors and estimate the camera pose by retrieving relevant database images.

Table 5.3 shows the comparison on the Dubrovnik dataset. As can be seen, our method outperforms state-of-the-art methods in localization accuracy. The 50% and 75% quartile errors of our method are significantly lower than other methods. In the meantime, we maintain a very competitive effectiveness, i.e. the number of successfully localized query images. Our method can successfully localize 794 out of 800 query images on the Dubrovnik dataset. On the Rome dataset, our method can successfully localize 991 out of 1000 query images.

Table 5.4 shows the percentage of query images localized within three pose accuracy

TABLE 5.4: The percentage of query images localized within three pose accuracy intervals of our proposed method compared with state-of-the-art localization methods on the RobotCar Seasons dataset.

	All Day	All Night
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10
AS [SLK12]	35.6 / 67.9 / 90.4	0.9 / 2.1 / 4.3
DenseVLAD [AZ13]	7.7 / 31.3 / 91.2	1.0 / 4.5 / **22.7**
NetVLAD [AGT+16]	6.4 / 26.3 / 91.0	0.4 / 2.3 / 16.0
Our method	**48.0** / **78.0** / **94.2**	**3.4** / **9.5** / 17.0

intervals of our proposed method compared with state-of-the-art localization methods on the RobotCar Seasons dataset. Our method significantly outperforms other methods in both High-precision $(0.25m, 2°)$ and Medium-precision $(0.5m, 5°)$ regimes. In High-precision regime, our method can localize 12.4% (= 48.0% - 35.6%) more query images comparing with AS. Interestingly, our method also outperforms retrieval-based methods DenseVLAD and NetVLAD in Coarse-precision regime $(5m, 10°)$ in day condition. This indicates that the top ranked database images voted by our method exhibit strong relevancy to provided query images. Yet, in night condition, when local feature matching easily fails, DenseVLAD and NetVLAD can localize more query images in Coarse-precision regime since they encode the global information for query images. Table 5.5 reports the comparison on the RobotCar Seasons dataset in different detailed conditions. Our method consistently achieves the best accuracy in High- and Medium-precision regimes with large margins. It is worthy to mention that in the Sun condition many query images are over exposed, making few feature descriptors found. In such condition, our method significantly outperforms AS in Coarse-precision regime. This is an interesting result since both our method and AS rely on local feature descriptors but not global image description. As shown in Table 5.6, our method outperform other methods in most scenarios on the Aachen Day-Night dataset.

We also investigate the memory consumption required in our method and other methods. In the following, let N_p be the number of 3D points in an SfM point cloud, N_d the number of feature descriptors (which is also the number of observing database images for all 3D points), N_m the number of integer mean descriptors based on the quantization results on a compact visual vocabulary (N_m is smaller than N_d), N_v the number of visual words contained in a visual vocabulary. TC needs to store all SIFT descriptors associated with 3D points. Overall, it requires the following memory consumption:

$$12 \cdot N_p + (128 + 4) \cdot N_d. \tag{5.13}$$

TABLE 5.5: The percentage of query images localized within three pose accuracy intervals of our proposed method compared with state-of-the-art localization methods on the RobotCar Seasons dataset (in different detailed conditions).

	Overcast Winter (DAY)	Sun (DAY)	Rain (DAY)
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
AS [SLK12]	33.1 / 71.5 / **93.8**	25.0 / 46.5 / 69.1	51.3 / 79.8 / 96.9
DenseVLAD [AZ13]	4.1 / 26.7 / 93.3	5.7 / 16.3 / 80.2	10.2 / 40.6 / 96.9
NetVLAD [AGT+16]	2.8 / 25.9 / 92.6	5.7 / 16.5 / **86.7**	9.0 / 35.9 / 96.0
Our method	**43.1** / **78.2** / 93.6	**36.1** / **62.8** / **86.7**	**59.6** / **83.1** / **97.9**
	Snow (DAY)	Dawn (DAY)	Dusk (DAY)
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
AS [SLK12]	36.4 / 72.2 / 93.7	36.2 / 68.9 / 89.4	44.7 / 74.6 / **95.9**
DenseVLAD [AZ13]	8.6 / 30.1 / 90.2	8.7 / 36.9 / 92.5	10.2 / 38.8 / 94.2
NetVLAD [AGT+16]	7.0 / 25.2 / 91.8	6.2 / 22.8 / 82.6	7.4 / 29.7 / 92.9
Our method	**54.2** / **84.9** / **95.3**	**51.1** / **78.1** / **92.8**	**56.3** / **83.2** / **95.9**
	Overcast Summer (DAY)	Night (NIGHT)	Night-rain (NIGHT)
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
AS [SLK12]	24.8 / 63.9 / 95.5	0.5 / 1.1 / 3.4	1.4 / 3.0 / 5.2
DenseVLAD [AZ13]	6.0 / 29.8 / 92.0	0.9 / 3.4 / **19.9**	1.1 / 5.5 / **25.5**
NetVLAD [AGT+16]	6.5 / 29.6 / 95.2	0.2 / 1.8 / 15.5	0.5 / 2.7 / 16.4
Our method	**36.5** / **76.5** / **97.8**	**2.3** / **6.6** / 15.3	**4.5** / **12.3** / 18.6

TABLE 5.6: The percentage of query images localized within three pose accuracy intervals of our proposed method compared with state-of-the-art localization methods on the Aachen Day-Night dataset.

	Day	Night
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10
AS [SLK12]	57.3 / 83.7 / **96.6**	19.4 / 30.6 / 43.9
DenseVLAD [AZ13]	0.0 / 0.1 / 22.8	0.0 / 2.0 / 14.3
NetVLAD [AGT+16]	0.0 / 0.2 / 18.9	0.0 / 2.0 / 12.2
Our method	**76.8** / **88.1** / 95.4	**25.5** / **35.7** / **52.0**

12 bytes are required to store the position of each 3D point. 128 bytes are required to store each SIFT feature descriptor. 4 bytes are required to store the corresponding database image index for each feature descriptor. For each 3D point, AS needs to store an integer mean (128-bytes) of SIFT descriptors per visual word for a 3D point. Overall, it requires the following memory consumption:

$$12 \cdot N_p + 4 \cdot N_d + (128 + 8) \cdot N_m + 128 \cdot N_v, \tag{5.14}$$

TABLE 5.7: The memory consumption (in GB) comparison between our method and other state-of-the-art methods.

Method	Memory Consumption			
	Dubrovnik	Rome	RobotCar Seasons	Aachen Day-Night
EGM [LSH10]	0.78	1.66	-	-
TC [CSC+17]	1.20	-	-	-
AS [SLK12]	0.75	1.60	2.72	0.76
Our method	**0.14**	**0.30**	**0.52**	**0.14**

where $8 \cdot N_m$ bytes are used to store the indexes of corresponding 3D points and visual words for integer mean feature descriptors, and $128 \cdot N_v$ bytes are used to store the compact visual vocabulary. Since EGM needs to store both 8-bytes binary signatures and integer mean feature descriptors, it requires the following memory consumption:

$$12 \cdot N_p + 4 \cdot N_d + (128 + 8 + 8) \cdot N_m + 128 \cdot N_v. \tag{5.15}$$

To store the information of feature descriptors, our method only needs to store a 8-bytes binary signature per visual word for each 3D point. Therefore, the overall memory consumption required by our method is as following:

$$12 \cdot N_p + 4 \cdot N_d + (8 + 8) \cdot N_m + 128 \cdot N_v. \tag{5.16}$$

Table 5.7 shows the comparison of memory consumption. Comparing to other methods, our method requires significantly lower memory consumption. For example, our method achieves superior performance but requires only $\sim 19\%$ of memory consumption of AS for the RobotCar Seasons dataset. For all datasets used in the experiments, the average computational time to localize a query image is less than 1 second, which validates the high efficiency of our method.

5.4.4 Ablation Study

We conduct an ablation study on the Dubrovnik dataset to evaluate the impact of key components in our method. We first implement a baseline that can disambiguate matches established from binary signatures. In the baseline implementation, a match is evaluated by Eq. 5.4. Then, we select all matches from top-20 ranked database images for computing the auxiliary camera pose, and we select all matches from top-100 ranked database images to obtain the visibility-wise match pool. We keep other components

TABLE 5.8: The comparison between our method and a baseline implementation on the Dubrovnik dataset.

Setting	Error quartiles [m]			Localized images
	25%	50%	75%	
Baseline	0.25	0.69	2.19	778
Our method	**0.22**	**0.64**	**2.16**	**794**

TABLE 5.9: The ablation study of quality-aware spatial reconfiguration (QSR) and principal focal length (PFL).

Setting	Error quartiles [m]			Localized images
	25%	50%	75%	
w/o QSR	0.26	0.74	2.53	793
w/o PFL	0.31	0.80	2.70	**794**
w/ QSR and PFL	**0.22**	**0.64**	**2.16**	**794**

unchanged, and the difference between our method and the baseline implementation is the bilateral Hamming ratio test in Section 5.1 and the two-step match selection in Section 5.2. We test with multiple Hamming distance thresholds in Eq. 5.4, and the baseline implementation achieves the best performance when setting the threshold to 11. Table 5.8 presents the comparison. Our method can successfully localize 16 more query images than the baseline implementation. This clearly demonstrates that the bilateral Hamming ratio test and the two-step match selection method are beneficial to better disambiguate matches.

We also conduct an experiment on the Dubrovnik dataset to investigate the impact of the quality-aware spatial reconfiguration (QSR) method and the principal focal length estimation (PFL) in Section 5.3. We first disable QSR and select the same number of *VFC* and *VFC-I* matches as when QSR enabled. Note that the matches in QSR disabled are selected with the largest match scores. As shown in Table 5.9, QSR significantly improves the localization accuracy. This indicates that obtaining a set of uniformly distributed matches before RANSAC-based pose estimation is essential for accurate image-based localization. To examine the benefit of PFL, we conduct an experiment with traditional RANSAC scheme when computing the auxiliary camera pose, i.e. the best camera pose is the one with largest number of inliers. As shown in Table 5.9, PFL also significantly improves the localization accuracy. This indicates that the auxiliary camera pose selected with PFL is more robust to apply geometry-wise match disambiguation.

5.5 Summary

In this chapter, we adopt the Hamming Embedding method to convert high-dimensional feature descriptors to binary ones. Though this can largely reduce the memory consumption of SfM models, the match disambiguation problem becomes more challenging. To break this dilemma, we propose a cascaded parallel filtering method to well disambiguate 2D-3D matched under binary representation. We first introduce a bilateral Hamming ratio test to better evaluate the feature-wise distinctiveness of matches. We also improve the visibility-wise match filter by exploring point-point relationship via a two-step match selection process. Through a quality-aware spatial reconfiguration algorithm, our method can easily generate a set of matches with both high quality and rational spatial distribution. Last, we propose to select the camera pose hypothesis using principal focal length value to improve the localization accuracy. The experiments on several benchmark datasets show that our method achieves a very competitive localization performance in a memory-efficient manner comparing with state-of-the-art. Because of high accuracy and compactness, the method proposed in this chapter unleashes the possibility for image-based localization applications to be used in the field of autonomous driving.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Image-based localization (*i.e. estimate the 6-DOF camera pose for an image*) using an SfM point cloud is a fundamental problem in many 3D vision tasks. As elaborate SfM algorithms have made large-scale reconstruction efficiently implemented in machines, there is a need for an image-based localization method to handle the challenges brought by the resultant large-scale SfM point clouds. In particular, we investigate three major challenges: memory consumption, match disambiguation and accurate camera pose estimation.

In Chapter 3, we have presented a novel framework for the SfM point cloud simplification problem. Based on evaluating the potential of a point cloud for establishing sufficient feature 2D-3D correspondences, we derive a fast prediction algorithm to predict the parameter K given an expected localization performance ratio. The derived number of 2D-3D correspondences that a query image has from a random view can serve a quality measure for an SfM point cloud, since it reflects the information sufficiency for image-based localization. By analyzing the process of the K-Cover algorithm, we claim that a single 3D point's visibility probability is an important factor to preserve the information of original 3D point clouds. An adaptive exponential weight scheme is proposed for the greedy heuristic selection process, which introduces a negligible computational overhead.

The experimental study on three benchmarks has confirmed the robustness of our prediction algorithm with respect to datasets with different attributes. The proposed

parameter prediction method has been shown to achieve less than 5% error in most cases when the provided performance ratio is above 70%. To the best of our knowledge, our method is the first work that provides parameter prediction for SfM point cloud simplification methods, which significantly improves their practical usability. In addition, the proposed adaptive exponential weighted K-Cover algorithm, in a data-driven manner, has been shown to outperform existing simplification methods in terms of localization performance.

In Chapter 4, we have considered the challenging match disambiguation problem in large-scale urban image-based localization scenarios. We have proposed a two-stage outlier filtering method that consists of an improved visibility-based outlier filter and a subsequent novel geometry-based outlier filter. In the first stage, we have demonstrated that through database image re-ranking and match augmentation, the performance of the visibility-based outlier filter can be significantly boosted. In the second stage, we have derived a novel data-driven geometrical constraint that is useful in generating a set of fine-grained matches. The geometrical constraint is motivated from a common phenomenon in urban environments: there exist many *locally visible points* that can only be observed by nearby cameras. Based on this constraint, a hybrid inlier evaluation measurement has been introduced in the geometry-based outlier filter for RANSAC-based camera pose estimation.

Comparing with previous outlier filtering approaches, our method is prior-free without requiring additional sensors to acquire the camera vertical direction. In addition, our method has been shown to effectively and efficiently handle 2D-3D matches of large outlier ratios, which frequently appear in urban scenes due to massive repetitive patterns. Consequently, our method has been shown to outperform state-of-the-art on the challenging *San Francisco* dataset in terms of urban image-based localization.

Finally, in Chapter 5, we have accounted for the image-based localization problem under binary feature representation. To handle the severe ambiguity of resultant matches, we have proposed a complete pipeline that sequentially purifies matches using feature, visibility and geometry intrinsics. Our novel bilateral hamming ratio test has been shown to better capture the feature-wise distinctiveness of matches established under binary feature representation. The proposed two-step match selection based on the visibility intrinsic has been verified to be suitable for preserving more correct matches. To improve the localization accuracy, we have introduced a quality-aware spatial reconfiguration method, which maintains a good balance between the spatial distribution and quality of matches.

Comprehensive experiments on four popular real-world datasets have demonstrated the benefit of our proposed image-based localization pipeline. Our method has been shown to achieve very competitive localization effectiveness and accuracy in a memory-efficient manner. The promising results on the *RobotCar Seasons* and *Aachen Day-Night* datasets have implied that our method can well disambiguate 2D-3D matches even in the challenging cross-season and day-night cases. The ablation study has verified the usefulness of each key element in our method.

6.2 Future Work

In this section, we propose three further directions for future work.

Hybrid Scene Compression: The availability of both SfM point cloud simplification and binary feature representation based methods leads to an interesting problem: can we adequately fuse these two kinds of methods to further reduce the memory consumption of large-scale SfM point clouds? During the SfM point cloud simplification process, a recent hybrid scene compression method [CCPS18] preserves the feature descriptors for the most informative 3D points. Meanwhile, less informative 3D points are grouped and represented by a common feature descriptor, thereby saving the memory consumption for storing more 3D points. Inspired from this method, future work could keep using high-dimensional feature representation for the informative 3D points so that matches will be reliably established against these points. For 3D points that are less informative, the binary feature representation could be adopted. As such, using the same amount of memory resources, more information could be preserved comparing with using only high-dimensional feature representation.

Long-Term Image-based Localization: In this thesis, we rely on traditional hand-crafted feature descriptors (*e.g.* SIFT) for establishing tentative 2D-3D matches. Even though proven to be robust under moderate geometrical changes, traditional hand-crafted feature descriptors still are sensitive to strong geometrical and photometric changes, *e.g.* illumination, to some extent. This leads to a challenge for long-term image based localization since photometric changes frequently happen due to day-night and season change. In future work, further improvements could be achieved by using more powerful feature detection and description methods [YTLF16, DMR18] based on neural networks. Moreover, it is promising to leverage feature descriptors [LCN16],

which are more suitable for long-term image-based localization due to training under various illumination changes. Except local feature descriptors, several global image representations [AGT+16], which are trained to be more robust under photometric changes, could also be adopted for future work.

Efficient Multi-sensor Fusion: As other types of sensors, *e.g.* Inertial measurement unit (IMU) except cameras become more available, future work should consider efficiently integrate priors from other sensors for image-based localization. Existing methods [ZSP15, SEKO17, TSH+18] relying on the IMU data bring in a heavy computation overhead due to exhaustively building and evaluating all possible hypothesis. In this regard, we believe that improving the efficiency of multi-sensor fusion, which could be achieved by prioritizing highly confident hypotheses using the data from other types of sensors, is a meaningful topic for image-based localization.

List of Publications

- **Wentao Cheng**, Weisi Lin, Kan Chen, Xinfeng Zhang, Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization. International Conference on Computer Vision (ICCV), 2019.
- **Wentao Cheng**, Kan Chen, Weisi Lin, Michael Goesele, Xinfeng Zhang, Yabin Zhang, A Two-stage Outlier Filtering Framework for City-Scale Localization using 3D SfM Point Clouds. IEEE Transaction on Image Processing, 28(10): 4857-4869, 2019.
- Yabin Zhang, Weisi Lin, Qiaohong Li, **Wentao Cheng**, Xinfeng Zhang, Multiple-Level Feature-Based Measure for Retargeted Image Quality. IEEE Transaction on Image Processing 27(1): 451-463, 2018.
- **Wentao Cheng**, Weisi Lin, Xinfeng Zhang, Michael Goesele, Ming-Ting Sun, A Data-Driven Point Cloud Simplification Framework for City-Scale Image-Based Localization. IEEE Transaction on Image Processing, 26(1): 262-275, 2016.
- **Wentao Cheng**, Weisi Lin, Ming-Ting Sun, 3D Point Cloud Simplification for Image-based Localization. IEEE International Conference on Multimedia & Expo Workshops (ICMEW) 2015: 1-6.

References

[AG80] Aloisio Araujo and Evarist Giné. *The central limit theorem for real and Banach valued random variables*, volume 431. Wiley New York, 1980. 34

[AGT+16] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 94, 95, 96, 103

[AKTS10] Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias, and Evaggelos Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 153–162. ACM, 2010. 19

[AMN+94] Sunil Arya, David M Mount, Nathan Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 573–582, 1994. 15

[AMN+98] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998. 36

[AZ13] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1578–1585. IEEE, 2013. 94, 95, 96

[AZ14] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, pages 188–204. Springer, 2014. 66, 70, 82, 83

[BKN+17] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017. 23

[BKP08] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 17, 88

[BKP11] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, pages 11–24, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 17

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 14

[CBK+11] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvä, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 737–744. IEEE, 2011. 19, 20, 65, 66, 70

[CCL+19] W. Cheng, K. Chen, W. Lin, M. Goesele, X. Zhang, and Y. Zhang. A two-stage outlier filtering framework for city-scale localization using 3d sfm point clouds. *IEEE Transactions on Image Processing*, pages 1–1, 2019. 7, 10, 53

[CCPS18] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. *arXiv preprint arXiv:1807.07512*, 2018. 102

[CLCZ19] Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded parallel filtering for memory-efficient image-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1032–1041, 2019. 9, 10, 79

[CLO+12] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012. 14

[CLS15] Wentao Cheng, Weisi Lin, and Ming-Ting Sun. 3d point cloud simplification for image-based localization. In *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015. 9, 38

[CLZ+16] Wentao Cheng, Weisi Lin, Xinfeng Zhang, Michael Goesele, and Ming-Ting Sun. A data-driven point cloud simplification framework for city-scale image-based localization. *IEEE Transactions on Image Processing*, 26(1):262–275, 2016. 7, 9, 26

[CM02] Ondrej Chum and Jırı Matas. Randomized ransac with td, d test. In *Proc. British Machine Vision Conference*, volume 2, pages 448–457, 2002. 18

[CM05] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005. 18

[CM08] Ondřej Chum and Jiří Matas. Optimal randomized ransac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008. 18

[CN12] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In *European conference on computer vision*, pages 130–143. Springer, 2012. 21, 33

[CS14] Song Cao and Noah Snavely. Minimal scene descriptions from structure from motion models. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 461–468. IEEE, 2014. 21, 24, 39, 42, 46, 47

[CSC+17] Federico Camposeco, Torsten Sattler, Andrea Cohen, Andreas Geiger, and Marc Pollefeys. Toroidal constraints for two-point localization under high outlier ratios. In *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, pages 4545–4553, 2017. 8, 21, 22, 30, 56, 59, 64, 65, 75, 76, 80, 91, 94, 97

[CT15] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 864–872, 2015. 13

[DBI18] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. *Computer Vision and Pattern Recognition (CVPR). IEEE*, 1, 2018. 23

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 102

[DRP+19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019. 23

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005. 13

[FBF76] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic time. *ACM Trans. Math. Software*, 3(SLAC-PUB-1549-REV. 2):209–226, 1976. 15

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. 3, 28

[FFW16] Youji Feng, Lixin Fan, and Yihong Wu. Fast localization in large-scale environments using supervised indexing of binary features. *IEEE Transactions on Image Processing*, 25(1):343–358, 2016. 22, 25

[Fis81] Martin A Fischler. Random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2, 15, 17, 62

[GHTC03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 17

[GLGP13] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2916–2929, 2013. 22

[HE08] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 19

[HHS13] Michal Havlena, Wilfried Hartmann, and Konrad Schindler. Optimal reduction of large image databases for location recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 676–683, 2013. 20

[HLON94] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International journal of computer vision*, 13(3):331–356, 1994. 17

[Hon11] Yili Hong. On computing the distribution function for the sum of independent and non-identical random indicators. *Dep. Statit., Virginia Tech, Blacksburg, VA, USA, Tech. Rep. 11_2*, 2011. 34

[HS+88] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 14

[HTDL13] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013. 13

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 11, 15, 16

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 13, 24

[JB09] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2419–2426, June 2009. 17

[JCT13] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 481–488, 2013. 13

[JDS08] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer, 2008. 22, 25, 61, 79, 81

[JDS09] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1169–1176. IEEE, 2009. 9, 20, 83

[JPD+12] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2012. 82

[KC+17] Alex Kendall, Roberto Cipolla, et al. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, volume 3, page 8, 2017. 23, 24, 65, 75, 76

[KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 23

[KS+04] Yan Ke, Rahul Sukthankar, et al. Pca-sift: A more distinctive representation for local image descriptors. *CVPR (2)*, 4:506–513, 2004. 14

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 13

[KSP10] Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In *European Conference on Computer Vision*, pages 748–761. Springer, 2010. 19

[KSS11] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011. 17, 62, 68, 90

[LCN16] Chris Linegar, Winston Churchill, and Paul Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 787–794. IEEE, 2016. 102

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 IEEE international conference on computer vision (ICCV)*, pages 2548–2555. Ieee, 2011. 22

[LLD17] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2391–2400. IEEE, 2017. 22, 30, 80, 91, 94

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3, 4, 14, 15, 56, 82

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 13

[LSH10] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *Computer Vision–ECCV*

2010, pages 791–804. Springer, 2010. xii, 2, 3, 20, 21, 24, 27, 30, 33, 40, 42, 65, 76, 91, 97

[LSHF12] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *Computer Vision–ECCV 2012*, pages 15–29. Springer, 2012. xiii, 8, 16, 21, 30, 59, 60, 65, 66, 68, 70, 80

[ML09] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009. 36

[ML14] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2227–2240, 2014. 15, 66

[MP07] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 13

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005. 14

[NS06] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006. 15, 19

[PCI+07] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 19

[Pol99] Marc Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, PhD thesis, ESAT-PSI, KU Leuven, 1999. 13

[PWN+13] Hyun Soo Park, Yu Wang, Eriko Nurvitadhi, James C Hoe, Yaser Sheikh, and Mei Chen. 3d point cloud reduction using mixed-integer quadratic programming. In *Computer Vision and Pattern Recognition Workshops*

(CVPRW), 2013 IEEE Conference on, pages 229–236. IEEE, 2013. 21, 24

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006. 14

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011. 14

[SBS07] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007. 19

[Sco12] John Scott. *Social network analysis*. Sage, 2012. 32

[SEKO17] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1455–1461, 2017. 8, 22, 25, 30, 56, 65, 66, 68, 75, 77, 80, 91, 103

[SF16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1, 13, 19

[SHR+15] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2102–2110, 2015. 8, 21, 58, 61, 65, 66, 68, 70, 80, 84

[SHSP16] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1582–1590, 2016. 20, 70

[SJS+18] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice

networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 23

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 19

[SLK11] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 667–674. IEEE, 2011. 13, 21, 30, 42, 65, 91

[SLK12] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. pages 752–765. Springer, 2012. 21, 30, 33, 46, 50, 56, 65, 75, 76, 77, 80, 81, 91, 93, 94, 95, 96, 97

[SMT+18] Torsten Sattler, Will Maddern, Akihiko Torii, Josef Sivic, Tomas Pajdla, Marc Pollefeys, and Masatoshi Okutomi. Benchmarking 6dof urban visual localization in changing conditions. 2, 2018. xiv, 92, 93

[SPGS18] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018. 23

[SSS06] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 1, 13, 19, 27

[ST93] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Cornell University, 1993. 14

[STS+17] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *CVPR 2017-IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 65, 75, 76

[SWLK12] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 6, page 7, 2012. 40, 82, 92

[SZ03] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003. 19

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 13, 24

[TLF10] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010. 14

[TLTD+18] Ngoc-Trung Tran, Dang-Khoa Le Tan, Anh-Dzung Doan, Thanh-Toan Do, Tuan-Anh Bui, Mengxuan Tan, and Ngai-Man Cheung. On-device scalable image-based localization via prioritized cascade search and fast one-many ransac. *IEEE Transactions on Image Processing*, 2018. 25

[TLTD+19] Ngoc-Trung Tran, Dang-Khoa Le Tan, Anh-Dzung Doan, Thanh-Toan Do, Tuan-Anh Bui, Mengxuan Tan, and Ngai-Man Cheung. On-device scalable image-based localization via prioritized cascade search and fast one-many ransac. *IEEE Transactions on Image Processing*, 28(4):1675–1690, 2019. 22

[TM+08] Tinne Tuytelaars, Krystian Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280, 2008. 14

[TSH+18] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 383–399, 2018. 80, 94, 103

[TSPO13] Akihiko Torii, Josef Sivic, Toma Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 883–890. IEEE, 2013. 70

[WHLT+17] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Int. Conf. Comput. Vis.(ICCV)*, pages 627–637, 2017. 23, 24

[WKP16] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016. 19

[WS14] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014. 13

[Wu13] Changchang Wu. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 127–134. IEEE, 2013. 13, 19

[YFST18] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2018. 23

[YTFO+18] Kwang Moo Yi, Eduard Trulls Fortuny, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, number CONF, 2018. 23

[YTLF16] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016. 102

[ZK06] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *3DPVT*, volume 6, pages 33–40. Citeseer, 2006. 19

[ZS10] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, pages 255–268. Springer, 2010. 20

[ZSP15] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2704–2712, 2015. 8, 21, 22, 25, 54, 56, 57, 59, 61, 65, 66, 68, 70, 74, 75, 77, 80, 91, 94, 103