DISS. ETH NO. 25370

Robust Methods for Accurate and Efficient 3D Modeling from Unstructured Imagery

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zürich
(Dr. sc. ETH Zürich)

presented by

Johannes Lutz Schönberger

MSc in Computer Science
University of North Carolina at Chapel Hill

born 17.02.1991
citizen of Germany

accepted on recommendation of

Prof. Dr. Marc Pollefeys
Prof. Dr. Jan-Michael Frahm
Prof. Dr. Richard Szeliski
Prof. Dr. Tomas Pajdla

2018

Abstract

Modeling the world in 3D from imagery has been a long-standing goal in photogrammetry and computer vision. The acquisition of 3D models is highly relevant for a variety of applications, including inspection and monitoring in quality control, heritage preservation, mapping and navigation, virtual tourism, mixed reality, autonomous robots, and many more. Over the last decades, many strides have been made towards the goal of building robust image-based 3D modeling systems. While these systems already achieve great results in many domains, factors such as robustness, accuracy, completeness, and efficiency remain the key challenges towards general-purpose image-based 3D reconstruction. This thesis proposes several methods that advance the state of the art in the different stages of an image-based 3D modeling pipeline. Thorough experimental evaluation on challenging real-world data and standardized benchmarks form the basis to derive and analyze the developed methods. The presented methods advance the state of the art in image-based 3D modeling in terms of robustness, accuracy, completeness, and efficiency of the employed algorithms and produced results. The integration of the different methods into an end-to-end image-based 3D modeling pipeline leads to one of the currently best-performing 3D modeling systems from unstructured imagery. Tailoring the developed pipeline and its products for different applications further underlines the robustness and generalization of the presented system.

Zusammenfassung

Die bildbasierte 3D Rekonstruktion ist seit jeher ein wichtiges Ziel im Bereich der Photogrammetrie und Computer Vision. Die Erfassung von 3D Modellen ist von großer Bedeutung in einer Vielzahl von Anwendungsbereichen, wie zum Beispiel bei der Inspektion und Überwachung in der Qualitätskontrolle, bei der Erhaltung von Kultur- und Naturdenkmälern, beim virtuellen Tourismus oder im Bereich von Mixed Reality, Computerspielen, Robotik, etc. Während der letzten Jahrzehnte wurden große Fortschritte gemacht mit dem Ziel ein robustes und vollautomatisches bildbasiertes 3D Rekonstruktionssystem zu entwickeln. Obwohl die entwickelten Methoden und Systeme bereits sehr gute Resultate in vielen Anwendungsbereichen erzielen, bleiben Faktoren wie Robustheit, Genauigkeit, Vollständigkeit und Effizienz die Schlüsselprobleme beim Erstellen eines universellen bildbasierten 3D Rekonstruktionssystem. Diese Dissertation präsentiert mehrere neue Methoden, die den Stand der Technik im Bereich der bildbasierten 3D Rekonstruktion weiter vorantreiben. Sorgfältige Experimente basierend auf anspruchsvollen und standardisierten Testdatensätzen stellen die Basis für die Entwicklung, Herleitung und Evaluierung der vorgestellten Methoden. Die präsentierten Methoden verbessern den Stand der Technik im Hinblick auf die Robustheit, Genauigkeit, Vollständigkeit und Effizienz der eingesetzten Algorithmen und Resultate. Die Integration der verschiedenen Methoden in ein vollständiges Programm resultiert in einem der derzeit besten 3D Rekonstruktionssystemen für unstrukturierte Bilddaten. Die Anpassung des entwickelten Systems und die Weiterverwendung der 3D Modelle in verschiedenen Szenarios und Applikationen ist ein weiterer Beweis für die Robustheit und Generalisierung der vorgestellten Methoden.

Acknowledgements

During my doctoral studies at ETH Zürich and UNC Chapel Hill, I had the pleasure to collaborate with and learn from many wonderful people. First, I would like to thank my supervisors Marc Pollefeys and Jan-Michael Frahm, who provided me with exceptional guidance and support throughout my doctoral studies. The opportunity and freedom to work on interesting and challenging research projects were extremely motivating. Besides my advisors, I would like to thank Rick Szeliski and Tomas Pajdla for their valuable feedback and comments as coexaminers of this thesis. Furthermore, I would like to especially thank my closest colleagues and friends True Price, Torsten Sattler, Filip Radenović, Jared Heinly, Ondrej Chum, Alex Berg, Enrique Dunn, Andreas Geiger, and Sudipta Sinha. Over the last couple of years, we had lots of fruitful discussions and collaborations, in which I learnt a lot and I very much enjoyed working with them. I would also like to express my gratitude to all the people that have worked together with me or supported me during my doctoral studies at ETH Zürich and UNC Chapel Hill, and of which many have become friends: Akash Bapat, Andrea Cohen, Daniel Thul, Danielle Luterbacher, Dinghuang Ji, Dominik Honegger, Enliang Zheng, Federico Camposeco, Friedrich Fraundorder, Hans Hardmeier, Hyojin Kim, Ian Cherabier, Jiri Matas, Ke Wang, Konrad Schindler, Marcel Geppert, Martin Oswald, Nektarios Lianos, Pablo Speciale, Silvano Galliani, Susanne Keller, Thomas Schöps, Thorsten Steenbock, Timo Hinzmann, Vagia Tsiminaki, Yagiz Aksoy, and Zhen Wei. Last but not least, I would like to express my sincere gratitude to my parents Helga and Kurt and my partner Pelin for their love and unconditional support throughout my life. Without all the amazing people in my life, this thesis would not have been possible. To them, I dedicate this thesis.

Johannes Schönberger
Zürich, August 2018

Contents

Part I

Introduction

1 Overview

1.1 Motivation

The problem of image-based 3D modeling is one of the important challenges in the fields of photogrammetry and computer vision. The goal of image-based 3D modeling is to derive useful geometric and semantic information from camera images. Modeling the real world in 3D is highly relevant and finds widespread use in many applications, such as inspection and monitoring in quality control, digital heritage preservation, digital mapping and navigation, virtual tourism, gaming, mixed reality, autonomous robots, and many more. The various applications have different aims and thus pose different requirements to a 3D modeling system. For example, inspection tasks usually require an accurate reconstruction of the geometry to enable precise measurements. On the contrary, it is of prime importance for visualization purposes to provide a visually pleasing and realistic experience for the user, independent of the accuracy of the underlying 3D model.

To accommodate the needs by different applications, a variety of technologies have been developed that enable 3D modeling, while each of them comes with its own benefits and drawbacks for certain applications. The approaches can generally be categorized into active (e.g., lidar, radar, etc.) and passive acquisition methods (i.e., cameras). As a passive acquisition method, cameras are especially power efficient and do not require direct physical contact with the real world. Furthermore, with the increasing availability of cameras as commodity sensors in consumer devices, the cost of camera hardware has fallen significantly over the last years. This development means that virtually everyone nowadays owns a digital camera and contributes to the ever-increasing wealth of visual data of the world. Personal storage devices and cloud storage services host a continuously updated digital image of the real world. Organizing and utilizing this extremely rich and diverse data source for 3D modeling means immense potential but also comes with significant challenges for image-based 3D reconstruction systems.

In principle, an image-based 3D reconstruction system aims to replicate the spatial understanding of the human visual system in order to extract 3D information from 2D images acquired by a camera. For a human, it is natural to build an accurate and complete 3D representation of the real world on the fly, but abstracting the underlying problem in a computer program is extremely hard. While many of the underlying problems in image-based 3D modeling are nowadays well understood after decades of research, there are still many problems for which we have not yet developed a deep understanding. The lack of a full understanding of the entire problem makes it especially challenging to translate the existing theory into robust and

efficient algorithms. To make image-based 3D modeling feasible in practice, the typical approach is to decompose the reconstruction process into several smaller, more tractable sub-problems. However, even with theoretically infinite computational time, the recovery of the true 3D model is often not possible, because arbitrarily many different 3D reconstructions may produce the same set of 2D images. For example, without prior information, we cannot recover the size of an object purely from images, because the images of an object are identical if the object is either tiny and close to the camera or tall and far away from the camera. Therefore, it is typically necessary to encode prior assumptions about the world into the 3D modeling process, e.g., that an object has a certain size or that most objects have a smooth surface. While tremendous progress has been made in understanding and modeling the problem over the last decades, we still have not accomplished to design a reliable and general-purpose reconstruction system.

Motivated by the above observations, this thesis focuses on the development of robust algorithms for the accurate and efficient 3D modeling of the real world from unstructured imagery. In the following sections in this chapter, we first give a brief historic overview of prior work in the field of image-based 3D modeling before introducing a high-level description of a typical image-based 3D reconstruction pipeline. Finally, we define the scope of the thesis and give an overview of its main contributions.

1.2 History

This brief historic overview is not comprehensive and is limited to some of the milestones of automatic image-based 3D modeling systems in the recent history. We refer the reader to the following chapters and related literature for a more specific and more comprehensive overview of prior work.

Over the last decades, image-based 3D modeling has seen tremendous evolution in the research community. The first works on image-based 3D modeling from calibrated cameras focused on the reconstruction from two views [133, 388], while later works noticed the benefits of leveraging the redundancy from more images [20, 27, 28, 76, 120, 296, 313, 329, 338]. Later, the early self-calibrating reconstruction systems [36, 78, 79, 86, 222, 249, 250, 300] served as the foundation for the first systems on unstructured image collections [275, 310] and urban scenes [251]. Inspired by these works, increasingly large-scale reconstruction systems have been developed for hundreds of thousands [3] and millions [89, 252, 288, 365] to recently a hundred million unstructured Internet images [129]. A number of other non-commercial (Bundler [310], VisualSfM [365], TheiaSfM [326], OpenMVG [224], MVE [91], etc.) and commercial software packages (Pix4D [245], PhotoScan [7], RealityCapture [317], etc.) were released based on the aforementioned innovations in the research community. While the existing systems have tremendously improved in terms of reconstruction quality and efficiency over the last decades, robustness, accuracy, completeness, and scalability still remain the key challenges in image-based 3D modeling that prevent

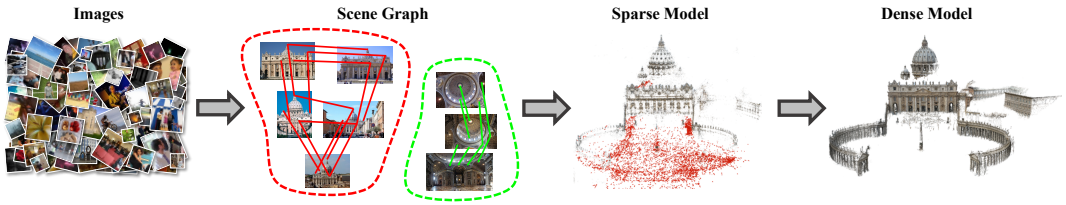

Figure 1.1: Overview of an image-based 3D modeling pipeline.

its use as a general-purpose method. This thesis builds upon the existing theory and presents several improvements that advance the state of the art towards the ultimate goal of a general-purpose reconstruction system.

1.3 System

Image-based 3D modeling aims to recover the 3D structure of a scene from its projection into a set of 2D images. In other words, image-based 3D modeling solves the inverse problem of 3D rendering in computer graphics. Recovering the 3D structure from a set of 2D images is an inherently hard and often ill-posed problem, since important information is lost in the process of capturing the properties of a 3D scene using 2D images. To make this problem feasible, the process of image-based 3D modeling is typically decomposed into a three-stage pipeline (see Figure 1.1). Starting from an unstructured collection of images, the first step searches for scene overlap and corresponding structure in the images. The resulting scene graph connects corresponding objects in multiple images and serves as the input for the subsequent 3D modeling stage. For reasons of computational efficiency, the modeling stage usually first recovers a coarse (or sparse) and then a dense 3D model of the scene. In the sparse modeling stage, the intrinsic (zoom level, distortion, etc.) and extrinsic (location and orientation) camera calibration of overlapping images is recovered alongside a (sparse) 3D point cloud of the scene. The dense modeling stage then uses the sparse model and the obtained camera calibrations to reconstruct a richer representation of the scene, e.g., in the form of a dense point cloud or a textured surface mesh. To understand this entire system in more detail, the next chapter first introduces the underlying concepts and principles in computer vision, (multi-view) geometry, and optimization. Based on these fundamental building blocks, we introduce the individual components of a traditional image-based 3D modeling pipeline in more detail. The following chapters then present the core contributions of this thesis that enable robust image-based 3D modeling from unstructured imagery in large-scale environments.

1.4 Scope

This thesis considers image-based 3D modeling of a rigid scene from multiple, unstructured images captured by a perspective camera. As opposed to image-based 3D modeling of dynamic scenes from a single or very few viewpoints, rigid scene reconstruction algorithms that leverage multiple viewpoints typically require fewer prior assumptions. This thesis therefore focuses on developing general algorithms using weak geometric priors for the accurate geometric modeling of the world from unstructured imagery in uncontrolled environments. The input to most of the developed methods in this thesis are unstructured in the sense that there is no prior knowledge about viewpoints, captured scenery, or camera hardware and calibration. One of the major challenges in this setting is the robust 3D modeling from such a diverse and complex data source. Crowd-sourced Internet images are primarily considered as one of the potential unstructured data sources in this thesis. The rapidly increasing amount of visual data on the Internet is a main motivation for the focus on developing efficient algorithms in order to obtain a complete and accurate 3D model of the entire world.

1.5 Contributions

The following list highlights the main contributions that are presented in this thesis.

- A thorough experimental evaluation [285] of learned and hand-crafted local feature descriptors is conducted to better understand their performance and impact across a wide range of scenarios in image-based 3D modeling.
- Three algorithms [283, 287, 288] are presented that drastically speedup and robustify the two-view reconstruction procedure in the correspondence search stage, which is an essential component and often the runtime bottleneck of an image-based 3D reconstruction system. Experiments on large-scale datasets demonstrate drastic speedups over the state of the art.
- Several algorithmic improvements to the incremental sparse reconstruction paradigm are presented [284]. These improvements advance the state of the art in terms of robustness, accuracy, completeness, and scalability towards the goal of building a general-purpose pipeline.
- A learning-based approach to improve dense stereo reconstruction is proposed. The presented algorithm fuses scanline optimization proposals in semi-global matching and replaces the brittle and heuristic scanline aggregation steps in the traditional semi-global matching algorithm. Evaluations on several benchmarks demonstrate state-of-the-art performance and robust generalization across different scenarios.
- A multi-view stereo algorithm [289] for the reconstruction of dense point clouds in highly unstructured environments is presented. The algorithm uses a joint

PatchMatch and variational optimization strategy and achieves state-of-the-art results on several recent 3D modeling benchmarks [167, 290].

- An end-to-end reconstruction system [288] from large-scale image collections is presented. The approach tightly integrates the image retrieval and 3D modeling components. The system takes a single query image of the scene as input and incrementally reconstructs a detailed and accurate 3D model of the entire scene based on a large, crowd-sourced database of images.

- An illumination robust 3D modeling system is presented [252] that produces complete and accurate dense 3D reconstructions in the presence of mixed day- and nighttime images. Our proposed system is able to reconstruct a complete color representation for the dense model surfaces by leveraging the corresponding appearance characteristics of the daytime and nighttime images.

- A robust visual localization approach is presented [286] that is based on joint semantic and geometric understanding of the 3D world. The proposed approach demonstrates reliable loop closure and localization in image-based 3D models even under extreme viewpoint and appearance changes. As such, this work is an important step towards robust, life-long localization in applications such as autonomous robots or mixed reality.

To facilitate reproducibility and future research, the implementation of most of the presented methods are released as open-source software[1] and bundled in the program *COLMAP*. The software includes a fully automatic image-based 3D modeling pipeline for sparse and dense scene reconstruction from unstructured images. The contributions and the content in this thesis are based on the material published in the following peer-reviewed papers:

[76] I. Cherabier*, J. L. Schönberger*, M. Oswald, M. Pollefeys, and A. Geiger. "Learning Priors for Semantic 3D Reconstruction". In: *European Conference on Computer Vision (ECCV). *Equal contribution.* 2018.

[89] A. Cohen*, J. L. Schönberger*, P. Speciale, T. Sattler, J. Frahm, and M. Pollefeys. "Indoor-Outdoor 3D Reconstruction Alignment". In: *European Conference on Computer Vision (ECCV). *Equal contribution.* 2016.

[154] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. "Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset)". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2015.

[159] T. Hinzmann, J. L. Schönberger, M. Pollefeys, and R. Siegwart. "Mapping on the Fly: Real-time 3D Dense Reconstruction, Digital Surface Map and Incremental Orthomosaic Generation for Unmanned Aerial Vehicles". In: *Field and Service Robotics - Results of the 11th International Conference.* 2015.

[1] Available for download at `https://github.com/colmap/colmap`.

[234] N. Lianos, J. L. Schönberger, M. Pollefeys, and T. Sattler. "VSO: Visual Semantic Odometry". In: *European Conference on Computer Vision (ECCV)*. 2018.

[297] T. Price, J. L. Schönberger, Z. Wei, M. Pollefeys, and J.-M. Frahm. "Augmenting Crowd-Sourced 3D Reconstructions using Semantic Detections". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[298] F. Radenović, J. L. Schönberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas. "From Dusk till Dawn: Modeling in the Dark". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[334] J. L. Schönberger, A. C. Berg, and J.-M. Frahm. "Efficient Two-View Geometry Classification". In: *German Conference on Pattern Recognition (GCPR)*. 2015.

[335] J. L. Schönberger, A. C. Berg, and J.-M. Frahm. "PAIGE: PAirwise Image Geometry Encoding for Improved Efficiency in Structure-from-Motion". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[336] J. L. Schönberger and J.-M. Frahm. "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[337] J. L. Schönberger, F. Fraundorfer, and J.-M. Frahm. "Structure-from-motion for MAV image sequence analysis with photogrammetric applications". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2014.

[338] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. "Comparative Evaluation of Hand-Crafted and Learned Local Features". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[339] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. "Semantic Visual Localization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[340] J. L. Schönberger*, T. Price*, T. Sattler, J.-M. Frahm, and M. Pollefeys. "A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval". In: *Asian Conference on Computer Vision (ACCV)*. **Equal contribution*. 2016.

[341] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm. "From Single Image Query to Detailed 3D Reconstruction". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[342] J. L. Schönberger, S. Sinha, and M. Pollefeys. "Learning to Fuse Proposals from Multiple Scanline Optimizations in Semi-Global Matching". In: *European Conference on Computer Vision (ECCV)*. 2018.

[343] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*. 2016.

[344] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

1.6 Outline

Chapter 2 introduces the underlying principles in computer vision, (multi-view) geometry, and optimization that are a pre-requisite to understand the remainder of the thesis. The rest of the thesis is structured roughly in the order in which the respective contributions appear in an image-based 3D modeling system (correspondence search, sparse reconstruction, dense reconstruction, and systems and applications). First, Chapter 3 presents a thorough comparative evaluation of learned and handcrafted local features in the context of image-based 3D modeling. Next, Chapter 4, Chapter 5, and Chapter 6 propose three algorithmic speedups for sparse two-view geometry estimation. Chapter 7 presents several algorithmic improvements to the sparse incremental reconstruction paradigm. Chapter 8 describes a learning-based approach to improve dense two-view stereo reconstruction based on the semi-global matching algorithm. Chapter 9 proposes a multi-view stereo approach for the reconstruction of dense 3D models from unstructured imagery. Chapter 10 introduces an end-to-end reconstruction system that takes a single query image and produces a detailed 3D reconstruction of the scene. Chapter 11 introduces an illumination robust 3D modeling system, which enables complete and accurate 3D modeling in the presence of mixed day- and nighttime images. Chapter 12 presents a system for the robust visual localization in 3D models under extreme viewpoint and illumination changes. Finally, Chapter 13 and Chapter 14 conclude the thesis and provide an outlook to open problems and future work.

2 Principles

This chapter introduces the underlying concepts and principles in computer vision, (multi-view) geometry, and optimization, which are necessary to understand the remainder of the thesis. In the first section of this chapter, we focus on the problem of how to abstract the content of an image in a computer and how this is relevant to the task of image-based 3D modeling. In the second section of this chapter, we define the geometry of the image formation process, while the third section introduces relevant optimization algorithms. The last section of this thesis then combines all the different principles and formalizes a full image-based 3D modeling system.

2.1 Image Description

Most problems in computer vision boil down to a correspondence problem between raw input images (e.g., in the form of a matrix of color values) and abstract concepts (e.g., high-level semantic categories or low-level numeric vectors). Associating raw images with abstract concepts enables to make predictions about single images (e.g., describe the captured scene content) which in turn allows for establishing correspondence between multiple images (e.g., images were taken at the same location). In image-based 3D modeling, we are interested in reasoning about the geometric relationship of multiple images in order to reconstruct the 3D structure of the captured scene. One of the core problems in image-based 3D modeling is thus the robust and efficient geometric association of multiple related images. This association process is typically solved in a two-stage approach: first, the contents of all the input images are described independently. Using these independent image descriptions, one then tries to find similar images that depict the same scene structure.

In the literature, there are different approaches to describe the content of an image. On the one hand, approaches can be categorized based on whether they describe the content globally (e.g., "this image shows a mountain") or locally (e.g., "this region/pixel in the image shows a tree"). On the other hand, approaches differ in the specificity of the description, i.e., whether the description is on the category-level (e.g., tree, mountain, etc.) or on the instance-level (e.g., General Sherman Tree, Mount Everest, etc.). In image-based 3D modeling, we make use of image description approaches from different ends of the spectrum. On a high level, local approaches enable to reconstruct the precise geometric relation between a pair of images on the instance-level, while global approaches are less specific and less accurate but enable the efficient association and categorization of large collections of images. In the following, we briefly review the basic principle of both local and

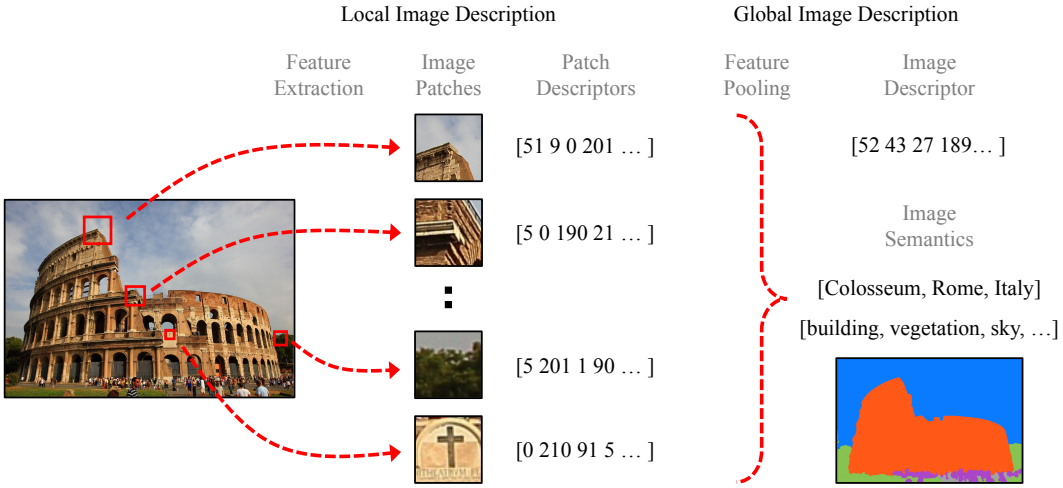

Figure 2.1: An overview of the concept of local and global image description.

global approaches, which are most relevant to the task of image-based 3D modeling. Figure 2.1 visualizes the concepts of the two image description concepts.

2.1.1 Local Image Description

Local image description decomposes an image into locally distinct features to describe its content. Local features can be, for example, points, edges, or blobs and are typically described using the image content in their immediate neighborhood. Ideally, these features should be distinctive and repeatably recognizable in different images of the same object, i.e., their description should be invariant under radiometric and geometric changes [211].

In image-based 3D modeling, sparsely detected point features are predominant because they allow for a robust and efficient parameterization of the geometric reconstruction problem. Point features are typically detected for distinctive locations in the image, e.g., at corners or crossings. To identify points of the same object in different images, one extracts and correlates information about their geometry (position, orientation, size, etc.) and appearance (e.g., a descriptor vector as a list of colors of a small patch around the point). Inherently, such a representation is not invariant under different illumination or viewing conditions. Therefore, the appearance description of a feature is typically normalized with respect to its geometry and a standard approach is to encode color gradient information instead of absolute color values. Usually, local image description approaches encode the content of an image using several hundreds or thousands of local feature points with their low-level geometry and appearance properties. By matching the local features of two different images depicting the same scene, one can establish pixel-level correspondence between two or more image regions. As explained later in this chapter, pixel-level correspondence information is a necessary input in order to determine the relative

geometric relation between images.

The Scale Invariant Feature Transform (SIFT) arguably has been the the most popular representative of hand-crafted local features for more than a decade. SIFT [200], its derivatives [47, 80, 345], and more recently automatically learned local features [24, 178, 301, 302] are the gold standard in terms of robustness [285]. Alternatively, binary features provide better efficiency at the cost of reduced robustness [127]. In Chapter 3, we provide an extensive comparative evaluation of hand-crafted and learned local features in the context of image-based 3D modeling.

2.1.2 Global Image Description

Global image description abstracts the contents of an image in a single representation, e.g., a high-level semantic category or a list of numerical values. Naturally, this representation is much more compact than local feature description and thus enables a more efficient categorization and association of many images. A popular approach to obtain a global description of an image is to summarize information from local features into a fixed-size histogram. This approach is inspired by the bag-of-words model in natural language processing and information retrieval [118]. Here, a local image feature is the equivalent of a word and the global image descriptor vector is simply a (sparse) histogram of the word occurrences over the visual vocabulary [231]. The visual vocabulary can be learned using hierarchical clustering from a large corpus of local image features. In order to find images with similar image content, one simply determines the overlap of their visual word occurrences, e.g., using term frequency-inverse document frequency (tf-idf) voting. This can be efficiently done for large image collections using an inverted index from word to image occurrences. An immediate advantage of this model is that local and global image description is tightly coupled, as the same local features can be used in both models. Recently, alternative approaches automatically learned from data have been developed for global image description [172, 253]. Rather than pooling information from distinct local features, end-to-end trainable convolutional neural networks directly learn to regress a fixed-size numerical vector in an embedding space or a semantic description of the image. Nearest neighbor search in the learned embedding space and semantic reasoning thereby allows for efficient association of images.

2.2 Multi-View Geometry

This section reviews some of the basic principles of multi-view geometry that are relevant to the task of image-based 3D modeling. Basic knowledge of projective geometry and its stratification into the affine, metric, and Euclidean subspaces is a pre-requisite to understand this section. We refer the interested reader to existing literature for an in-depth review about these topics [123]. We first derive the geometric properties of a single view and then generalize this understanding to two and more views.

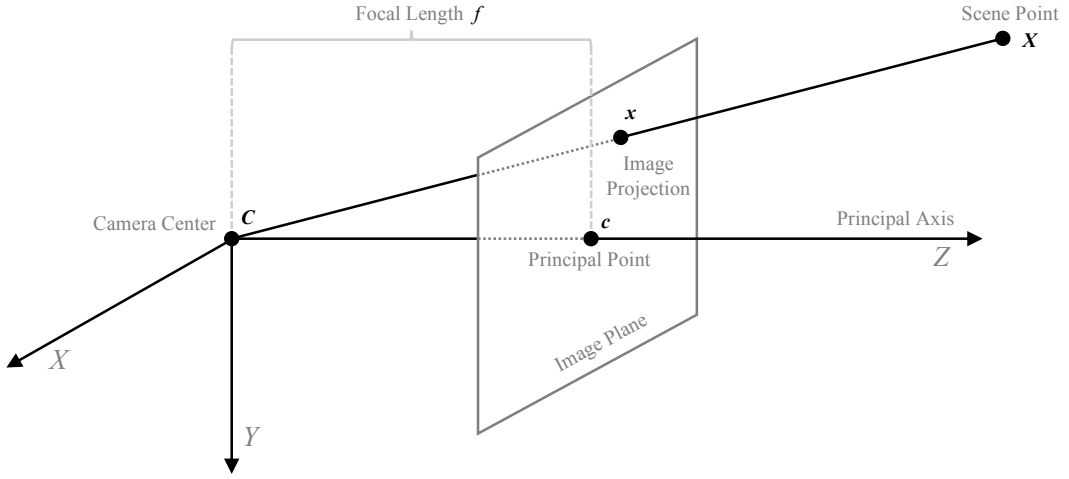

Figure 2.2: The geometry of a pinhole camera with the local camera coordinate system defined by the axes (X, Y, Z). The scene point $\boldsymbol{X}$ is projected onto the image plane at $\boldsymbol{x}$ through the pinhole camera projection center $\boldsymbol{C}$. The image plane is orthogonally offset to the plane (X, Y) by the distance of the focal length f.

2.2.1 Pinhole Camera

At its core, image-based 3D modeling aims to invert the process of capturing an image of a real world scene, i.e., we aim to recover the 3D structure of the real world from its projection into 2D images. Towards this goal, this section first introduces the geometry of the image capturing process, while the following sections then derive the theory necessary to invert this process. This thesis considers a perspective camera projection model, in which a 2D imaging plane captures the light rays emitted from the 3D scene points $\boldsymbol{X} \in \mathbb{R}^3$. When using an ideal pinhole camera, all the captured light rays pass through a single center of projection $\boldsymbol{C} \in \mathbb{R}^3$ (the aperture). Mathematically, this projection process can be formulated as

$$\boldsymbol{x} \simeq \lambda \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \boldsymbol{P}\boldsymbol{X} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \end{bmatrix} \boldsymbol{X} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}^T\boldsymbol{C} \end{bmatrix} \boldsymbol{X} \ , \tag{2.1}$$

where $\boldsymbol{P}$ is a 3×4 rank-3 matrix and defines the projection from the scene point $\hat{\boldsymbol{X}}$ in homogeneous coordinates $\boldsymbol{X} \in \mathbb{P}^3$ to an observation $\boldsymbol{x} \in \mathbb{P}^2$ in the projective imaging plane. The 3×3 rotation matrix $\boldsymbol{R} \in SO(3)$ and the translation vector $\boldsymbol{T} \in \mathbb{R}^3$ define the Euclidean transformation from world to camera coordinate system. These parameters are typically denoted as the extrinsic camera calibration, while the intrinsic camera calibration is encoded in the upper triangular matrix

$$\boldsymbol{K} = \begin{bmatrix} f & s & c_u \\ 0 & af & c_v \\ 0 & 0 & 1 \end{bmatrix} \ , \tag{2.2}$$

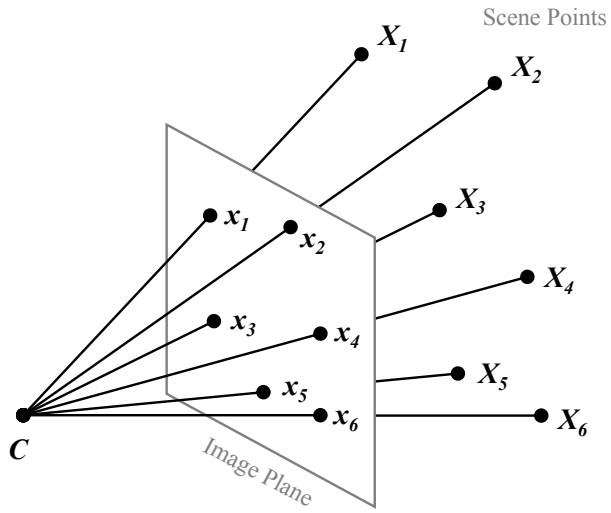

Figure 2.3: Pinhole camera calibration from 6 2D-3D correspondences.

where $\boldsymbol{c} = (c_u, c_v) \in \mathbb{R}^2$ defines the location of the principal point and f is the focal length in pixels with an anisotropy a and a shearing factor s due to non-rectangular pixels. The geometry of the pinhole camera projection process is visualized in Figure 2.2.

2.2.2 Camera Calibration

Camera calibration aims to recover the intrinsic and/or extrinsic parameters of a camera. Given 2D image observations $\boldsymbol{x}$ and their corresponding 3D scene points $\boldsymbol{X}$, the 12 parameters of the projection matrix $\boldsymbol{P}$ can be estimated using the linear relation

$$\boldsymbol{x} \simeq \boldsymbol{P}\boldsymbol{X} = \begin{bmatrix} \boldsymbol{P}_1^T \\ \boldsymbol{P}_2^T \\ \boldsymbol{P}_3^T \end{bmatrix} \boldsymbol{X} \tag{2.3}$$

from Equation 2.1. Reordering the equations with the direct linear transform allows to eliminate the unknown scaling factor λ and we obtain the homogenous system of equations

$$\begin{matrix} \boldsymbol{P}_3^T \boldsymbol{X} u = \boldsymbol{P}_1^T \boldsymbol{X} \\ \boldsymbol{P}_3^T \boldsymbol{X} v = \boldsymbol{P}_2^T \boldsymbol{X} \end{matrix} \quad \Rightarrow \quad \boldsymbol{0} = \boldsymbol{A} \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^T & 0 & -\boldsymbol{X}^T u \\ 0 & \boldsymbol{X}^T & -\boldsymbol{X}^T v \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{bmatrix} . \tag{2.4}$$

This linear system can be solved from at least 6 2D image to 3D point correspondences, because every equation contributes 2 constraints for a total of 12 unknowns (see Figure 2.3). The direct linear transform solution lies in the null space of $\boldsymbol{A}$ and thus corresponds to the eigenvector corresponding to the smallest singular value of the decomposition SVD($\boldsymbol{A}$). The projection matrix $\boldsymbol{P}$ can finally be decomposed

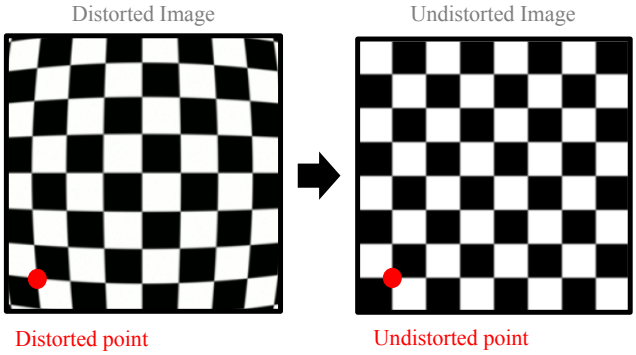

Figure 2.4: Visualization of lens distortion effects in the image.

into its intrinsic and extrinsic components using an RQ-decomposition followed by a normalization of the Q matrix [123], since the calibration matrix $\boldsymbol{K}$ is an upper triangular and $\boldsymbol{R}$ an orthonormal matrix.

Note that the projection model described previously assumes an ideal pinhole camera, which is in practice not sufficient to model the complex lens distortion effects present in a real-world camera. Typically, these distortions are parameterized using, for example, radial coefficients $k_1, k_2, ...$ of higher order polynomials, such as

$$\tilde{\boldsymbol{x}} = f_{\text{dist}}(\boldsymbol{x}) = \begin{bmatrix} u & v \end{bmatrix}^T (1 + k_1 r^2 + k_2 r^4 + ...) + ... \quad \text{with} \quad r^2 = u^2 + v^2 \ . \tag{2.5}$$

Here, the undistorted point $\boldsymbol{x}$ is the projection of a 3D point $\boldsymbol{X}$ using an ideal pinhole camera (see Equation 2.1), while the function f_{dist} applies a lens distortion correction (see Figure 2.4). In case the lens distortion of a camera is known a priori, the camera calibration described in Equation 2.3 and Equation 2.4 should be done with undistorted points $\boldsymbol{x} = f_{\text{dist}}^{-1}(\tilde{\boldsymbol{x}})$. Otherwise, the camera calibration routine must include an estimation of the lens distortion parameters, rendering the estimation process significantly more difficult. Calibrating cameras with complex distortion models is beyond the scope of this review section and we refer the interested reader to related photogrammetry and computer vision literature [123, 211].

For accurate and robust image-based 3D modeling, it is crucial to properly model the camera projection process. Choosing an appropriate camera model for the task at hand is important to obtain high quality reconstruction results. For example, oftentimes the intrinsic parameters of a complex distortion model can be determined a priori using a precise but time-intensive calibration routine. Whereas in other scenarios, it is necessary to calibrate each image independently during the reconstruction process, thereby prohibiting the use of complex distortion models due to insufficient observational evidence from a single image.

2.2.3 Triangulation

In image-based 3D modeling, we are ultimately interested in recovering the 3D structure $\boldsymbol{X}$ of the scene from 2D observations $\boldsymbol{x}$ in the image. Simply inverting

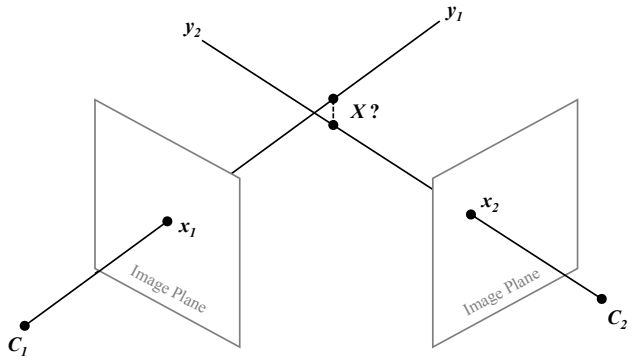

Figure 2.5: Two-view triangulation of the point $\boldsymbol{X}$ from two corresponding image observations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ by intersection of their viewing rays $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$.

Equation 2.1 is not feasible, since any two points along the viewing ray $\boldsymbol{y} = \boldsymbol{K}^{-1}\boldsymbol{x} \in \mathbb{P}^2$ are equal in projective space, i.e.

$$\boldsymbol{y}_1 = \lambda \boldsymbol{y}_2 \quad \text{with} \quad \lambda \neq 0 \ . \tag{2.6}$$

One of the key challenges in image-based 3D modeling is thus to recover the unknown scale factor λ, since depth information is lost in the 3D to 2D projection process. In other words, we must recover the distance λ from the camera location $\boldsymbol{C}$ to the observed scene point $\boldsymbol{X}$ along the viewing ray $\boldsymbol{y}$ of a pixel $\boldsymbol{x}$. Given the intrinsic and extrinsic calibration of a camera and a known scale factor λ, the 3D location $\bar{\boldsymbol{X}}$ of an observed image point $\boldsymbol{x}$ is computed as

$$\bar{\boldsymbol{X}} = \lambda \boldsymbol{R}^T \boldsymbol{K}^{-1} \boldsymbol{x} + \boldsymbol{C} \ . \tag{2.7}$$

Without prior knowledge about the depth λ, the point location can be determined by intersecting the corresponding viewing rays from multiple image observations in different cameras (see Figure 2.5). This intersection process is called triangulation and, similar to the camera calibration in Equation 2.4, it can be estimated by reordering Equation 2.1 using the direct linear transform as

$$\begin{matrix} \boldsymbol{P}_3^T \boldsymbol{X} u = \boldsymbol{P}_1^T \boldsymbol{X} \\ \boldsymbol{P}_3^T \boldsymbol{X} v = \boldsymbol{P}_2^T \boldsymbol{X} \end{matrix} \quad \Rightarrow \quad \boldsymbol{0} = \begin{bmatrix} \boldsymbol{P}_3^T u - \boldsymbol{P}_1^T \\ \boldsymbol{P}_3^T v - \boldsymbol{P}_2^T \end{bmatrix} \boldsymbol{X} \tag{2.8}$$

from a minimum of two observations in different images. This equation is over-determined even in the minimal two-view case with 4 linear equations and 3 unknown point coordinates. This additional degree of freedom stems from the fact that two viewing rays will likely not perfectly intersect in 3D space due to measurement noise in the image observation $\boldsymbol{x}$ or an inaccurate camera calibration $\boldsymbol{P}$. Also note that the system of equations becomes singular if camera centers are identical. Geometrically, all viewing rays are parallel in this case and any point along the coinciding viewing rays is a valid solution.

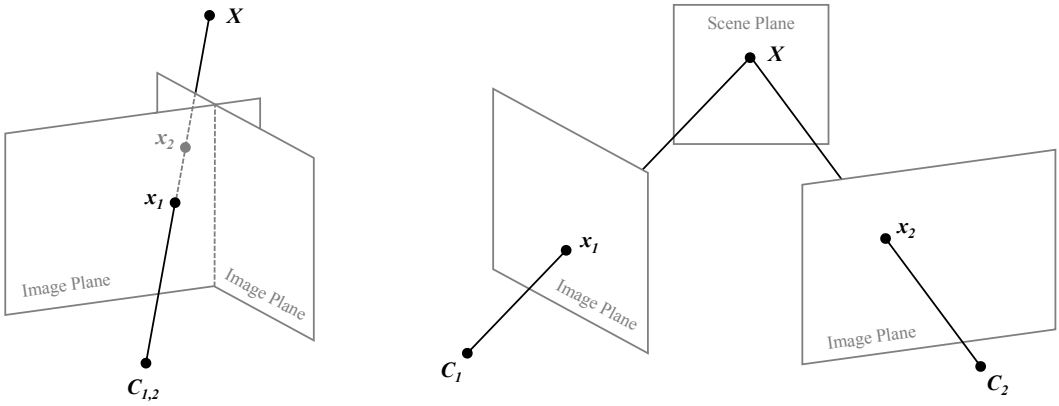

Figure 2.6: The homography maps points points from one plane to another plane and thereby describes the two-view geometry for a purely rotating camera (left) and a planar scene (right).

2.2.4 Two-View Geometry

The previous two sections described the process of recovering 3D scene structure from known camera calibrations and vice versa. In practical image-based 3D modeling, however, none of the two are known a priori and we must recover them simultaneously. This section derives the the theory of the homography and epipolar geometry, which can be used to describe the geometric relation between two views of a camera undergoing different types of motion without explicitly recovering the Euclidean structure of the scene. The homography describes the two-view geometry of a purely rotating camera or a camera with arbitrary motion capturing a planar scene. Epipolar geometry describes the two-view geometry of a non-stationary camera viewing an arbitrary scene with known and unknown intrinsics, respectively. As we will see later in this thesis, these two concepts are important components of any image-based 3D modeling system.

Homography

The 2D homography h is a projective transformation in $\mathbb{P}^2$ that preserves lines in projective space, i.e., any three points $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\boldsymbol{y}_3$ are collinear if and only if they are collinear in $h(\boldsymbol{y}_1)$, $h(\boldsymbol{y}_2)$, $h(\boldsymbol{y}_3)$. In other words, the 2D homography maps points $\boldsymbol{x}_1$ on one plane to points $\boldsymbol{x}_2$ on another plane

$$\boldsymbol{x}_2 \simeq \lambda_2 \begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix}^T = h(\boldsymbol{x}_1) = \boldsymbol{H}\boldsymbol{x}_1 = \begin{bmatrix} \boldsymbol{H}_1^T \\ \boldsymbol{H}_2^T \\ \boldsymbol{H}_3^T \end{bmatrix} \boldsymbol{x}_1 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \boldsymbol{x}_1 \ . \quad (2.9)$$

The homography matrix $\boldsymbol{H}$ has 8 degrees of freedom due to the projective ambiguity λ and can therefore by estimated from at least 4 image correspondences in two views

by solving the homogeneous system of equations

$$\begin{matrix} \boldsymbol{H}_3^T \boldsymbol{x}_1 u_2 = \boldsymbol{H}_1^T \boldsymbol{x}_1 \\ \boldsymbol{H}_3^T \boldsymbol{x}_1 v_2 = \boldsymbol{H}_2^T \boldsymbol{x}_1 \end{matrix} \quad \Rightarrow \quad \mathbf{0} = \begin{bmatrix} \boldsymbol{x}_1^T & 0 & -\boldsymbol{x}_1^T u_2 \\ 0 & \boldsymbol{x}_1^T & -\boldsymbol{x}_1^T v_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \end{bmatrix} . \tag{2.10}$$

By interpreting the homography as a mapping between two image planes, it can be used to describe the two-view geometry in two specific configurations. In the scenario of a purely rotating camera, the projection center and the viewing rays remain constant. If we define a local coordinate system in two images of such a camera, then we can find a homography that maps points from the first to the second image plane (see Figure 2.6). Note that the second image plane may coincide with an arbitrary plane of the scene and vice versa. As such, the homography can also describe the two-view geometry of an arbitrarily moving camera observing a planar scene: we can first construct a homography that maps from the first image to the scene plane and then concatenate a second homography that maps from the scene plane back to the second image (see Figure 2.6).

Without loss of generality, we set the extrinsic parameters of the camera of the first image as $\boldsymbol{R}_1 = \boldsymbol{I}$ and $\boldsymbol{T}_1 = \mathbf{0}$. Under these assumptions, the homography can be decomposed as

$$\boldsymbol{H} = \boldsymbol{K}_2 \left(\boldsymbol{R}_2 - \frac{\boldsymbol{T}_2 \boldsymbol{N}^T}{d} \right) \boldsymbol{K}_1^{-1} \tag{2.11}$$

in both discussed geometric configurations. Here, $\boldsymbol{N}$ is the unit normal vector of the scene plane and d the orthogonal distance of the scene plane to the projection center of the first camera. Due to the inherent scale ambiguity in Equation 2.9 and in the term $\frac{\boldsymbol{T}_2}{d}$ in Equation 2.11, the homography can generally not recover the scale of the scene and the camera motion. Note that this property is intrinsic to the image-based 3D reconstruction problem, in which one cannot recover the scale of the scene without prior knowledge.

The homography reduces to $\boldsymbol{H} = \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{K}_1^{-1}$, if the camera undergoes pure rotational motion or if the observed scene plane is infinitely far away. For calibrated cameras with known $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$, the homography simplifies to $\tilde{\boldsymbol{H}} = \boldsymbol{R}_2 - \frac{\boldsymbol{T}_2 \boldsymbol{N}^T}{d}$. An affinity is a special case of the homography modeling an orthographic camera when the focal lengths approach infinity and therefore $h_{31} = h_{32} = 0$ and $h_{33} = 1$. Further, if we constrain the upper left sub-block defined by $h_{11}, h_{12}, h_{21}, h_{22}$ to an orthogonal or orthonormal matrix, we respectively obtain a similarity or an isometry. This completes the hierarchy of projective transformations and enables to recover the relation between two views in specific geometric configurations without explicitly recovering the 3D Euclidean structure of the scene.

Epipolar Geometry

The previous section introduced the homography, which allows us to model the two-view geometry in case of a purely rotating camera or a planar scene. In this section,

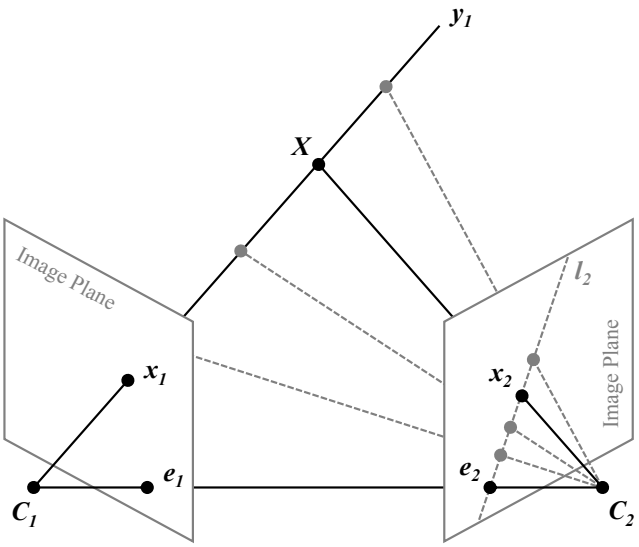

Figure 2.7: The fundamental matrix maps points from one image to lines in the other image and describes the two-view geometry of a general scene under non-stationary camera motion.

we define the concept of epipolar geometry, which allows us to model the two-view geometry of a general scene under non-stationary camera motion, where the centers of projection $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ of two cameras are different (see Figure 2.7).

Towards this goal, we first define the epipole, which is the projection of the projection center of the other camera in the current image, i.e.

$$\boldsymbol{e}_1 = \boldsymbol{P}_1 \boldsymbol{C}_2 \quad \text{and} \quad \boldsymbol{e}_2 = \boldsymbol{P}_2 \boldsymbol{C}_1 \ . \tag{2.12}$$

For any observation $\boldsymbol{x}_1$ of a 3D scene point $\boldsymbol{X}$ in the first image, we can construct the epipolar line through the epipole and the observation as $\boldsymbol{l}_1 = \boldsymbol{e}_1 \times \boldsymbol{x}_1$. Extending the epipolar line along the viewing ray to the scene gives the epipolar plane $\Pi = \boldsymbol{P}_1^+ \boldsymbol{l}_1$, where $\boldsymbol{P}_1^+$ is the pseudo-inverse of $\boldsymbol{P}_1$. Notice that the set of epipolar planes for all image observations defines a pencil of planes around the line segment between the two projection centers of the cameras. The intersection of the pencil of epipolar planes with the image planes defines two pencils of epipolar lines through the two epipoles. The intersection of the epipolar plane Π with the image plane of the second camera leads to the epipolar line in the second image $\boldsymbol{l}_2 = \boldsymbol{P}_2 \Pi$. Any 3D point $\boldsymbol{X}$ on the epipolar plane Π is observed in the two images along the corresponding epipolar lines $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$. The epipolar constraint formalizes this relation by enforcing that the observation $\boldsymbol{x}_2$ of $\boldsymbol{X}$ must lie on the epipolar line $\boldsymbol{l}_2$ in the second image, i.e.

$$0 = \boldsymbol{x}_2^T \boldsymbol{l}_2 = \boldsymbol{x}_2^T \boldsymbol{P}_2 \Pi = \boldsymbol{x}_2^T \boldsymbol{P}_2 \boldsymbol{P}_1^+ \boldsymbol{l}_1 = \boldsymbol{x}_2^T \boldsymbol{P}_2 \boldsymbol{P}_1^+ \left(\boldsymbol{e}_1 \times \boldsymbol{x}_1\right) = \boldsymbol{x}_2^T \boldsymbol{P}_2 \boldsymbol{P}_1^+ \left[\boldsymbol{e}_1\right]_\times \boldsymbol{x}_1 \ , \tag{2.13}$$

where $\left[\boldsymbol{e}_1\right]_\times$ is a skew-symmetric matrix representing the cross product.

This concludes the definition of epipolar geometry which can be summarized in the fundamental matrix

$$\boldsymbol{F} = \boldsymbol{P}_2 \boldsymbol{P}_1^{+} \left[\boldsymbol{e}_1\right]_{\mathrm{x}} = \boldsymbol{K}_2^{-T} \left[\boldsymbol{T}_{12}\right]_{\times} \boldsymbol{R}_{12} \boldsymbol{K}_1^{-1} = \begin{bmatrix} \boldsymbol{F}_1^T \\ \boldsymbol{F}_2^T \\ \boldsymbol{F}_3^T \end{bmatrix} \tag{2.14}$$

with the relative extrinsic calibration $\boldsymbol{R}_{12} = \boldsymbol{R}_2 \boldsymbol{R}_1^T$ and $\boldsymbol{T}_{12} = \boldsymbol{T}_2 - \boldsymbol{R}_{12} \boldsymbol{T}_1$ between the two cameras. The fundamental matrix maps points from one image to lines in the other image. The matrix has rank 2 with 7 degrees of freedom and can be estimated up to scale using the epipolar constraint from a minimal set of 7 point correspondences in the general case. While the minimal estimator requires to solve a cubic polynomial equation, the fundamental matrix can be estimated from 8 point correspondences by rearranging the homogeneous linear Equation 2.13 as

$$\boldsymbol{0} = \boldsymbol{x}_1 \otimes \boldsymbol{x}_2 = \begin{bmatrix} \boldsymbol{x}_1^T u_2 & \boldsymbol{x}_1^T v_2 & \boldsymbol{x}_1^T \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \end{bmatrix} \quad . \tag{2.15}$$

The solution to this equation can be determined using the singular value decomposition. This system of equations is over-determined in non-critical configurations (8 equations for 7 degrees of freedom) and, in the presence of noise, the rank 2 constraint is generally not satisfied. In this case, the obtained matrix maps points to epipolar lines that do not exactly intersect in a single epipole. The optimal solution to this problem is to find the closest rank 2 approximation of the obtained matrix. Furthermore, solving this system of equations in practice requires coordinate normalization due to limited numerical precision [119].

The essential matrix is a specialization of the fundamental matrix in the case of calibrated cameras with known intrinsics $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$. In the calibrated case, the matrix has only 5 degrees of freedom (3 for rotation and 2 for translation up to scale) and is defined as

$$\boldsymbol{F} = \boldsymbol{K}_2^{-T} \boldsymbol{E} \boldsymbol{K}_1^{-1} \quad \Leftrightarrow \quad \boldsymbol{E} = \boldsymbol{K}_2^T \boldsymbol{F} \boldsymbol{K}_1 = \left[\boldsymbol{T}_{12}\right]_{\times} \boldsymbol{R}_{12} \quad . \tag{2.16}$$

A minimal set of 5 point correspondences is necessary to estimate the essential matrix. Similar to the fundamental matrix, it can be estimated from a non-minimal set of 8 point correspondences, whereas the minimal 5-point algorithm leads to a degree 10 polynomial equation [230]. The essential matrix can be decomposed into 4 possible relative camera poses

$$\boldsymbol{R}_{12}(\pm\pi) = \boldsymbol{U} \boldsymbol{R}_z^T(\pm\tfrac{\pi}{2}) \boldsymbol{V}^T \tag{2.17}$$

$$\left[\boldsymbol{T}_{12}(\pm\pi)\right]_{\times} = \boldsymbol{U} \boldsymbol{R}_z(\pm\tfrac{\pi}{2}) \boldsymbol{\Sigma} \boldsymbol{V}^T \tag{2.18}$$

using the singular value decomposition $\mathrm{SVD}(\boldsymbol{E}) = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$ and $\boldsymbol{R}_z \in SO(3)$ as the rotation matrix around the z-axis

$$\boldsymbol{R}_z(\pm\tfrac{\pi}{2}) = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad . \tag{2.19}$$

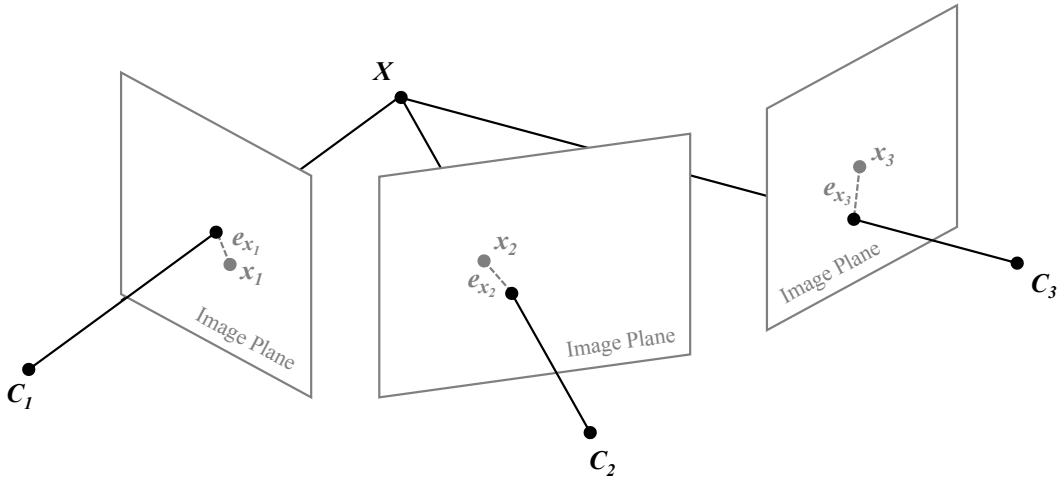

Figure 2.8: Bundle adjustment is the joint refinement of scene structure $\boldsymbol{X}$ and camera calibration $\boldsymbol{P}$ by minimizing the reprojection errors $\boldsymbol{e_x}$ of the image observations $\boldsymbol{x}$.

Only one solution of the 4 poses is geometrically plausible and leads to triangulated points in front of the cameras. Similar to the homography, the essential matrix can only be determined up to scale and thus its decomposition defines the translation between two cameras up to an arbitrary factor.

2.3 Non-Linear Estimation

The previous sections introduced linear estimation algorithms for geometric reconstruction problems from minimal sets of measurements. Linear algorithms are efficient to compute but generally not sufficient to obtain accurate and robust reconstruction results. The main reason being that, inherently, measurements are subject to observational noise, which causes uncertain estimation results. To mitigate the impact of measurement noise on the estimation results, it is crucial to leverage the redundancy of many measurements and to correctly model the noise statistics. In most settings, linear estimators are not optimal, because they do not model the noise statistics correctly. In the following, we first derive non-linear cost functions for the core geometric reconstruction problems that lead to optimal maximum likelihood estimators under a Gaussian noise assumption on the image observations. The second part of this section then describes optimization algorithms for the efficient and robust minimization the resulting non-linear cost functions.

2.3.1 Multi-View Geometry

Optimal estimators should correctly model the statistical properties of the underlying noise and should be able to leverage the redundancy from all available measure-

ments. Using more than the minimal set of measurements to increase redundancy is straightforward with all of the presented linear estimation algorithms for the geometric reconstruction problems. Additional measurements simply add more rows to the constraint matrices and, in the presence of measurement noise, the smallest singular value corresponding to the least-squares solution will be non-zero. Note, however, that all presented algorithms use an algebraic error, which lacks proper geometric or statistical justification.

Assuming zero-mean Gaussian noise on the image observations $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\boldsymbol{x}}^2)$, the maximum likelihood estimate of the structure $\boldsymbol{X}$ and the motion $\boldsymbol{P}$ is

$$\boldsymbol{X}^*, \boldsymbol{P}^* = \underset{\boldsymbol{X},\boldsymbol{P}}{\arg\min} \|\boldsymbol{x} - \frac{1}{\lambda}\boldsymbol{P}\boldsymbol{X}\| = \underset{\boldsymbol{X},\boldsymbol{P}}{\arg\min} \|\boldsymbol{e}_{\boldsymbol{x}}\| \ , \tag{2.20}$$

where $\frac{1}{\lambda}$ projects the homogeneous camera coordinate to the normalized image plane (see Equation 2.1). This non-linear least-squares optimization problem is called bundle adjustment [342] and minimizes the reprojection error between image measurements and the structure projected to the image (see Figure 2.8). The reprojection error $\boldsymbol{e}_{\boldsymbol{x}}$ has an intuitive geometric interpretation and its minimization leads to the maximum likelihood estimate under the Gaussian noise assumption.

The joint optimization of structure and motion requires a good initialization of $\boldsymbol{X}$ and $\boldsymbol{P}$ due to the highly non-linear nature of the bundle adjustment cost function. A typical approach for initialization is to first recover the motion using two-view geometry estimation and to then triangulate the structure from the computed motion before jointly refining both in a final bundle adjustment. In the following, we specify the optimal cost functions for the problems of camera calibration, triangulation, and two-view geometry estimation.

Camera Calibration and Triangulation

Solving Equation 2.4 and Equation 2.8 using the direct linear transform minimizes an algebraic error, which lacks proper geometric and statistical justification. Assuming zero-mean Gaussian noise $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\boldsymbol{x}}^2)$, the maximum likelihood estimate of the structure $\boldsymbol{X}$ given the motion $\boldsymbol{P}$ and vice versa is

$$\boldsymbol{P}^* = \underset{\boldsymbol{P}}{\arg\min} \|\boldsymbol{x} - \frac{1}{\lambda}\boldsymbol{P}\boldsymbol{X}\| \tag{2.21}$$

$$\boldsymbol{X}^* = \underset{\boldsymbol{X}}{\arg\min} \|\boldsymbol{x} - \frac{1}{\lambda}\boldsymbol{P}\boldsymbol{X}\| \ . \tag{2.22}$$

Compared to the full bundle adjustment problem, these geometric cost functions have fewer parameters but they still have many local minima and are thus difficult to optimize. For example, even for the minimal case of two views, the cost function of optimal triangulation may have as many as 3 local minima and maxima and thus the solution is expensive to compute in closed-form [122]. The standard approach is therefore to first determine an initial estimate of the structure and motion using

minimal, non-optimal estimators, such as the direct linear transform algorithm. These initial estimates are typically close to the global minimum and can then be efficiently refined by minimizing the optimal geometric cost functions above. A variety of different methods have been proposed for robust and efficient triangulation (e.g., [2, 8, 121, 122, 158, 189, 201, 235]) and, in Section 7.4, this thesis presents a novel sampling-based triangulation approach from an arbitrary number of views.

Two-View Geometry

Formulating the maximum likelihood estimators for two-view geometry problems is relatively more difficult than for camera calibration and triangulation: the homography or the fundamental and essential matrix transform geometric entities from one image to another, i.e., image observations appear on both sides of the equations. Assuming measurement errors on the image observations in both images, the geometric cost function [71] for the homography is defined as

$$\boldsymbol{H}^* = \arg\min_{\boldsymbol{H}} \|\boldsymbol{x}_2 - \frac{1}{\lambda_2}\boldsymbol{H}\boldsymbol{x}_1\| + \|\boldsymbol{x}_1 - \frac{1}{\lambda_1}\boldsymbol{H}^{-1}\boldsymbol{x}_2\| \ . \tag{2.23}$$

Note that assuming measurement noise in both images requires to compute the transfer error symmetrically in two directions. Under a Gaussian noise assumption, the maximum likelihood estimate of the homography can be estimated by minimizing the reprojection error in the total least-squares sense, which assumes errors on the image observations in both images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

$$\boldsymbol{H}^*, \hat{\boldsymbol{x}}_1^*, \hat{\boldsymbol{x}}_2^* = \arg\min_{\boldsymbol{H}, \hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2} \|\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1\| + \|\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2\| \quad \text{subject to} \quad \hat{\boldsymbol{x}}_2 = \boldsymbol{H}\hat{\boldsymbol{x}}_1 \ . \tag{2.24}$$

Recall that the fundamental and essential matrix map points in one image to epipolar lines in the other image. A geometrically more meaningful cost function than Equation 2.15 is the symmetric epipolar distance, which minimizes the orthogonal distance $d(\boldsymbol{x}, \boldsymbol{l})$ from point $\boldsymbol{x}$ to the epipolar line $\boldsymbol{l}$ as

$$\boldsymbol{F}^* = \arg\min_{\boldsymbol{F}} d\left(\boldsymbol{x}_2, \boldsymbol{F}\boldsymbol{x}_1\right)^2 + d\left(\boldsymbol{x}_1, \boldsymbol{F}^T\boldsymbol{x}_2\right)^2 \ . \tag{2.25}$$

In case all points are identically zero-mean Gaussian distributed, the maximum likelihood estimate for the fundamental matrix minimizes the reprojection error

$$\boldsymbol{F}^* = \arg\min_{\boldsymbol{F}} \|\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1\| + \|\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2\| \quad \text{subject to} \quad \hat{\boldsymbol{x}}_2^T\boldsymbol{F}\hat{\boldsymbol{x}}_1 = \boldsymbol{0} \ . \tag{2.26}$$

To obtain the optimal image correspondences $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$, the Hartley-Sturm triangulation algorithm can be used [122]. However, as discussed previously, optimal triangulation is expensive to compute and a more efficient cost function uses the Sampson error

$$\boldsymbol{F}^* = \arg\min_{\boldsymbol{F}} \frac{\boldsymbol{x}_2^T\boldsymbol{F}\boldsymbol{x}_1}{(\boldsymbol{F}\boldsymbol{x}_1)_1^2 + (\boldsymbol{F}\boldsymbol{x}_1)^2 + (\boldsymbol{F}^T\boldsymbol{x}_2)_1^2 + (\boldsymbol{F}^T\boldsymbol{x}_2)^2} \ , \tag{2.27}$$

which is a first-order approximation to the reprojection error.

2.3.2 Optimization Algorithms

In the previous sections, we defined several non-linear cost functions for different geometric reconstruction problems, but we did not specify how those cost functions can be minimized in order to find the desired estimates. Note that all presented cost functions have in common that they have a quadratic error function, as they are modeled in the least-squares sense. There are several approaches to minimize a non-linear cost function $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^n$ over some variables $\boldsymbol{\theta} \in \mathbb{R}^m$ in the least-squares sense, i.e.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{f}(\boldsymbol{\theta})\|^2 \ . \tag{2.28}$$

In general, the shape of such a cost function is non-linear and has many local minima. Since the global minimization of highly non-linear cost functions is intractable, the standard approach is to iteratively solve tractable approximations to the original cost function. For example, gradient descent methods start from an initial estimate $\boldsymbol{\theta}_0^*$ and iteratively minimize the cost by taking small steps down along the local linear approximation of the cost function, i.e., the gradient of the variables

$$g(\boldsymbol{\theta}) = \nabla \frac{1}{2}\|f(\boldsymbol{\theta})\|^2 = \boldsymbol{J}^T \boldsymbol{f}(\boldsymbol{\theta}) \ , \tag{2.29}$$

where $\boldsymbol{J}_{ij} = \delta_j \boldsymbol{f}_i(\boldsymbol{\theta})$ is the $n \times m$ Jacobian matrix summarizing the partial derivatives of $\boldsymbol{f}$ with respect to $\boldsymbol{\theta}$. Minimizing the first-order linearization

$$\boldsymbol{f}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \approx \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{J}(\boldsymbol{\theta})\Delta\boldsymbol{\theta} \tag{2.30}$$

of the cost function as

$$\Delta\boldsymbol{\theta}^* = \arg\min_{\Delta\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{J}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}\|^2 \tag{2.31}$$

leads to the iterative update rule $\boldsymbol{\theta}_{t+1}^* = \boldsymbol{\theta}_t^* + \Delta\boldsymbol{\theta}_t^*$ at step t starting from an initial estimate $\boldsymbol{\theta}_0^*$. Note that simple algorithms like naive gradient descent assume a constant step size for the update $\Delta\boldsymbol{\theta}$ and thus they often do not converge or are extremely slow. More advanced trust region or line search methods adaptively determine the step size along $\Delta\boldsymbol{\theta}$ and therefore have much better convergence properties.

For most geometric reconstruction problems and especially for bundle adjustment, the trust region method Levenberg-Marquardt [188, 204] is the most efficient algorithm. On a high level, Levenberg-Marquardt is a hybrid between gradient descent and Gauss-Newton. By expanding and rearranging Equation 2.31, we obtain the normal equations of the Gauss-Newton method

$$\boldsymbol{J}(\boldsymbol{\theta})^T \boldsymbol{J}(\boldsymbol{\theta})\Delta\boldsymbol{\theta} = -\boldsymbol{J}(\boldsymbol{\theta})^T \boldsymbol{f}(\boldsymbol{\theta}) \ . \tag{2.32}$$

The Levenberg-Marquardt algorithm is based on an augmented version of the normal equations

$$\left(\boldsymbol{J}(\boldsymbol{\theta})^T \boldsymbol{J}(\boldsymbol{\theta}) + \lambda \boldsymbol{D}(\boldsymbol{\theta})\right) \Delta\boldsymbol{\theta} = -\boldsymbol{J}(\boldsymbol{\theta})^T \boldsymbol{f}(\boldsymbol{\theta}) \ , \tag{2.33}$$

where $\lambda > 0$ is a damping parameter between the gradient descent and Gauss-Newton method. The $m \times m$ matrix $\boldsymbol{D}(\boldsymbol{\theta})$ should measure the quality of the update step $\Delta\boldsymbol{\theta}$, i.e., whenever the linear approximation is good, the algorithm should take large steps along the gradient. In order to avoid slow convergence, Marquardt [204] proposed to use the augmented Hessian as a measure of the cost curvature to scale the gradient as $\boldsymbol{D}(\boldsymbol{\theta}) = \operatorname{diag}\left(\boldsymbol{J}(\boldsymbol{\theta})^T \boldsymbol{J}(\boldsymbol{\theta})\right)$. Solving the linear equation $\boldsymbol{J}(\boldsymbol{\theta})\Delta\boldsymbol{\theta} = -\boldsymbol{f}(\boldsymbol{\theta})$ directly using, e.g., a Cholesky factorization, is typically not efficient for large problems with many parameters and cost terms. Exploiting the sparsity structure of $\boldsymbol{J}(\boldsymbol{\theta})^T \boldsymbol{J}(\boldsymbol{\theta})$ is key to solving the normal equations efficiently for large problems. For example, in bundle adjustment problems, only a subset of 3D points is visible in every image, i.e., each cost term only depends on a small subset of the unknown parameters, leading to a sparse Jacobian matrix. We refer the reader to the optimization literature [232] for further details on Levenberg-Marquardt and other state-of-the-art optimization algorithms.

2.3.3 Robust Estimation

The previous sections introduced maximum likelihood estimators under a Gaussian noise assumption on the image observations. Violating this assumption leads to biased or grossly wrong estimates. In practice, the input used to setup any of the presented estimation problems stems from imperfect, heuristic procedures. For example, as pre-processing to two-view geometry estimation, we must first establish correspondence between image observations in two overlapping images, e.g., using purely appearance-based local feature matching (see Section 2.1.1). If the overlapping structure has repetitive scene elements, it is likely that we sometimes establish incorrect correspondence. Notice that making such incorrect decisions is not modeled by Gaussian noise, because it typically leads to gross outliers with comparatively large error terms $\boldsymbol{f}(\boldsymbol{\theta})$. Such outlier measurements often have an especially large leverage on the estimation results, because of the squared error term in the cost function of Equation 2.31. A common approach to avoid biased estimates within traditional non-linear least-squares optimization is to down-weight the influence of outliers in the cost function by using robust kernels. Alternatively, the random sample consensus (RANSAC) algorithm determines the optimal model by iteratively maximizing the set of inlier measurements for a given measurement noise level.

Robust Kernels

To reduce the influence of individual measurements in Equation 2.28, we first extend the cost function as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_i \rho_i \left(\|\boldsymbol{f}_i(\boldsymbol{\theta})\|^2 \right) \quad , \tag{2.34}$$

by introducing $\rho_i : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ as a kernel function that can be used to reduce the influence of individual error terms. The idea is to choose a suitable kernel function ρ_i that transforms a set of non-Gaussian error terms into a Gaussian error distribution.

Close to the optimal result $\boldsymbol{\theta}^*$, the error term $\boldsymbol{f}(\boldsymbol{\theta})$ will be small for inliers (Gaussian measurements) and large for outliers (non-Gaussian measurements). A suitable kernel function to transform a non-Gaussian heavy-tailed distribution of errors into a Gaussian down-weights large outlier residuals and remains linear for inliers. In case of an identity kernel $\rho_i(x) = x$, the above equation becomes the original least-squares problem and assumes only Gaussian error terms. The following paragraphs introduce some of the most common kernel functions for robust estimation.

Iteratively reweighted least-squares is an algorithm that iteratively updates weights for the individual error terms from the current optimal estimates. The weight for an error terms in the current iteration t is computed as $\rho_i(x) = w_i x$ based on the estimated error in the previous iteration $w_i = \left|\boldsymbol{f}_i(\boldsymbol{\theta}^*_{t-1})\right|^{p-2}$ with $p = 1$ being equivalent to least absolute deviation. The Huber loss produces exactly the quadratic loss for measurements with errors below a certain residual threshold τ and is defined as

$$\rho_i(x) = \begin{cases} x & \text{if } x \leq \tau^2 \\ \tau\left(\sqrt{x} - \frac{\tau}{2}\right) & \text{if } x > \tau^2 \end{cases} . \tag{2.35}$$

The Huber loss is not smooth and has rather long tails, i.e., the modeled error distribution has rather short tails and thus the Huber loss is still susceptible to gross outliers. A smooth approximation to the Huber loss is the Pseudo-Huber loss, which is defined as $\rho_i(x) = \tau^2\left(\sqrt{1 + \frac{x}{\tau^2}} - 1\right)$. The Cauchy loss is a smooth and more robust alternative and it is computed as $\rho_i(x) = \log\left(1 + \frac{x}{\tau}\right)$.

In general, introducing robust kernels into least-squares enables the use of the same efficient optimizers as in any other non-linear least-squares problem. However, usually the convergence properties of such robustified problems are suboptimal: if the initial estimate $\boldsymbol{\theta}^*_0$ is far from the true solution, inliers may be incorrectly down-weighted and thus the iterative optimizer has little chance to converge to the correct solution. Lifting the kernel functions into a higher dimensional space [378] overcomes this issue. This approach has been shown to have better convergence properties in general and especially for bundle adjustment problems.

Random Sample Consensus

The random sample consensus (RANSAC) algorithm takes a fundamentally different approach to robust estimation. RANSAC aims to estimate a model under which a maximum number of inlier observations are explained by the underlying measurement noise assumption without outliers. More formally, RANSAC aims to maximize the following objective function

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{S} = \arg\max_{\boldsymbol{\theta}} \sum \rho(e_i^2) \quad \text{with} \quad \rho(e^2) = \begin{cases} 1 & \text{if } e^2 \leq \tau^2 \\ 0 & \text{if } e^2 > \tau^2 \end{cases} , \tag{2.36}$$

where e_i is the residual of a single measurement and $\mathcal{S}$ is a count of the number of inlier measurements, given an a priori inlier residual threshold τ. The residual threshold could, for example, be a reprojection error in pixels when estimating the

calibration of a camera. Under the Gaussian measurement noise assumption, a typical choice would be $\tau = 3\sigma$, i.e., only measurements which are likely to be explained by the underlying noise model are considered inliers.

RANSAC employs a hypothesize-and-verify strategy in order to maximize the objective function $\mathcal{S}$. The algorithm repeatedly hypothesizes a model from a minimal set of M random measurements and then verifies the model by counting the number of inliers $\mathcal{S}$ on all measurements. The aim is that at least one of the minimal sets of measurements, which are sampled in each iteration of the algorithm, contains only inliers and therefore produces a good estimate of the model. The model with the maximal inlier support $\mathcal{S}$ is returned as the final estimate.

Given a total of N measurements with I inliers, the probability of randomly sampling an outlier-free minimal set of measurements is

$$\frac{\binom{I}{M}}{\binom{N}{M}} \approx \epsilon^M \quad \text{with} \quad \epsilon = \frac{I}{N} \ . \tag{2.37}$$

The number of RANSAC iterations K required to hypothesize an outlier-free model is proportionally related to this equation as $K \sim \frac{1}{P}$. A typical stopping criterion for RANSAC iterates until

$$K \geq \frac{\log\left(1-\eta\right)}{\log\left(1-\epsilon^M\right)} \ , \tag{2.38}$$

resulting in a confidence η that at least one outlier-free combination of measurements has been hypothesized. Since the underlying distribution of measurements is typically not known a priori, the inlier ratio ϵ^M is set to a worst-case scenario initially and it is updated during the iteration as soon as a model with more inliers is verified. Especially for data with high outlier contamination, it is crucial to hypothesize the model from as few measurements as possible, because this increases the chance of sampling an outlier-free set of measurements, leading to a potentially good estimate of the model. Note that even an outlier-free set of measurements does not necessarily lead to a good estimate of the model due to measurement noise and the low redundancy in the sampled set of measurements or no redundancy if the sampled set is minimal. One solution to the problem is to explicitly model the uncertainty through error propagation [256]. Another solution uses local optimization [69], which accounts for the mentioned issue by refining the model on the initial set of inliers for the model estimated from the minimal set. The refined model is then verified again on all measurements and usually has bigger support as the increased redundancy from more measurements produces a more accurate estimate of the model. Several other variants of RANSAC exist [254] for better efficiency, more robustness, or fewer prior assumptions.

Minimal Solvers

As evident in Equation 2.36, RANSAC has an exponential runtime complexity in the number of model parameters and it is thus essential to hypothesize models from a minimal set of measurements. In the previous sections, we already introduced many linear estimation algorithms that facilitate the estimation of geometric reconstruction problems using few measurements. However, in many cases, the presented linear algorithms are not the minimal solutions, i.e., they are over-determined and do not lead to an efficient execution of RANSAC. For example, the fundamental matrix only has 7 and the essential matrix 5 degrees of freedom, whereas the described linear algorithm requires 8 points. Another example is camera calibration with known intrinsics, which only requires 3 2D-3D point correspondences, while the linear algorithm for the full uncalibrated case requires 6 2D-3D point correspondences. The minimal solutions to most geometric reconstruction problems are not linear anymore and require to solve complex polynomial equations, which is typically a computationally demanding process. Transforming the resulting polynomial equations into a Gröbner basis has been shown to be a particularly efficient way to design minimal solvers [173, 174]. Based on this theory, many different solvers have been proposed over the last years (e.g., [46, 50, 102, 175, 176, 179, 327]), that are specifically targeted at different types of reconstruction problems.

Model Selection

Many estimation problems require a model selection process, if the underlying functional model of a process is not known a priori. For example, together with the homography, the fundamental and essential matrix enable to describe the geometric configuration of any two views. The homography describes the geometric relation for pure rotational motion or a planar scene, while the essential and fundamental matrix describe the geometric relation for a non-stationary camera capturing a general scene. For an accurate and robust estimation of the two-view geometry, it is crucial to select the appropriate model for the given scenario. In practice, however, we have no prior knowledge of the captured scene structure or the specific camera motion for a given image pair. In fact, it is not even guaranteed that an arbitrary image pair has visual overlap at all. Consequently, we must perform model selection to solve the two-view reconstruction problem. A commonly used approach to perform model selection between $\boldsymbol{H}$, $\boldsymbol{E}$, and $\boldsymbol{F}$ is the Geometric Robust Information Criterion (GRIC) [341], which measures the goodness of fit for a maximum likelihood estimate as

$$\text{GRIC} = \sum_i \rho(e_i^2) + ND\ln(R) + M\ln(RN) \tag{2.39}$$

$$\text{with} \quad \rho(e^2) = \min\left(\frac{e^2}{\sigma^2}, 2\,(R-D)\right) \quad . \tag{2.40}$$

Here, e_i are the N measurement residuals, σ is the measurement standard deviation, M the number of model parameters, R the data dimensionality, and D the dimen-

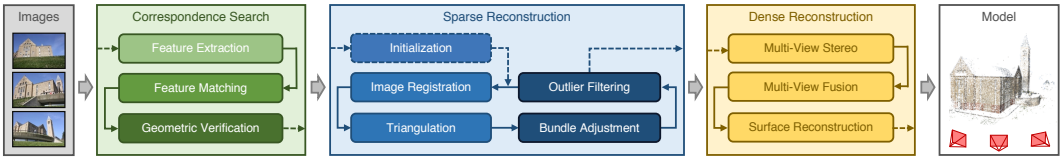

Figure 2.9: Traditional image-based 3D modeling pipeline using local features for correspondence search, an incremental algorithm for sparse modeling, and multi-view stereo for dense modeling.

sion of the structure. To decide on the correct model for a given image pair, one computes the GRIC measure for all estimates $\boldsymbol{H}$, $\boldsymbol{E}$, $\boldsymbol{F}$ and then simply selects the model with the best (lowest GRIC) score. If the goodness of fit is bad for all three models, the given image pair is not well explained by any of the discussed geometric transformations. This is typically the case for image pairs without visual overlap or, for example, in the case of dynamic scenes where observed 3D points change their position between the image capture. Note that the same concept applies for other model selection problems as well, e.g., when selecting different models for camera calibration.

2.4 Image-Based 3D Modeling Pipeline

The previous sections introduced some of the core concepts and algorithms for geometric reconstruction problems. In this section, we connect these different concepts and algorithms into a full pipeline for 3D reconstruction (see Figure 2.9) from an unstructured collection of images

$$\mathcal{I} = \{I_i \mid i = 1...N_I\} \ , \tag{2.41}$$

where each image I is captured by a perspective camera and is, for example, represented by a matrix of colors. Traditional image-based 3D modeling uses a sequential processing pipeline with an incremental sparse reconstruction component (see Figure 1.1). In addition to the incremental paradigm [3, 89, 310, 365], there are also hierarchical [108] and global approaches [75, 223, 328, 362]. These alternatives are typically less robust in unstructured scenarios and this thesis therefore focuses on the incremental approach. The first stage in an image-based 3D modeling system is correspondence search and commonly starts with feature extraction and matching followed by geometric verification. The resulting scene graph serves as the foundation for the reconstruction stage, which seeds the model with a specifically selected two-view reconstruction before incrementally registering new images, triangulating scene points, filtering outliers, and refining the reconstruction using bundle adjustment. The final dense reconstruction stage uses the estimated sparse scene model and the camera calibrations to recover a richer representation of the scene, e.g., in the form of a dense point cloud or textured surface mesh. The following sections elaborate on this process and define the notation used throughout the thesis.

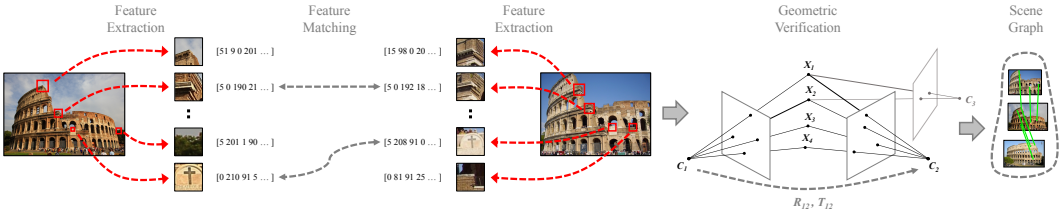

Figure 2.10: Correspondence search starts with feature extraction, which is followed by feature matching and geometric verification. Chaining the verified matches and two-view geometries produces the scene graph.

2.4.1 Correspondence Search

Starting from an unordered collection of images $\mathcal{I}$, the first stage in an image-based 3D reconstruction pipeline determines the two-view geometry for image pairs with scene overlap (see Figure 2.10). The output is a scene graph with images and scene points as nodes in the graph. Bidirectional edges between images specify scene overlap and are annotated with an appropriate model describing the two-view geometry. Undirected edges indicate visibility between images and scene points.

Feature Extraction

For each image I_i in the input collection $\mathcal{I}$, the pipeline first detects sets of local image features

$$\mathcal{F}_i = \{(\boldsymbol{g}_j, \boldsymbol{d}_j) \mid j = 1...N_{F_i}\} \tag{2.42}$$

represented by the local feature geometry $\boldsymbol{g}_j$ (e.g., location, orientation, scale) and an appearance descriptor $\boldsymbol{d}_j$ (e.g., a histogram of gradients). These features should be invariant under radiometric and geometric changes [211], such that, in the next stage, features of the same object can be uniquely recognized in other images under different appearance.

Feature Matching

The second stage discovers which pairs of the input images see the same part of the captured scene by leveraging the features $\mathcal{F}_i$ as an appearance description of the images. The naïve approach tests every possible combination of image pairs for scene overlap by exhaustively comparing all pairs of local features in each image pair. An image pair is considered to have scene overlap, if a sufficient number of local features are similar with respect to to their appearance description $\boldsymbol{d}_j$. This approach has computational complexity $O(N_I^2 N_{F_i}^2)$ and is prohibitively expensive for large image collections. A popular approach with an effective computational complexity of $O(N_I N_{F_i})$ uses hierarchical indexing of the local features in a vocabulary tree to obtain a global description of the image. The compact global image description is then used to efficiently find the visually most similar looking images for each image

in the collection in linear time with respect to the number of images. For each visually similar image pair, the hierarchical index can then be further used to find correspondence between local features in linear time with respect to to the number of features. Beyond this traditional approach, there exist a variety of other approaches that tackle the problem of scalable and efficient matching [3, 89, 126, 129, 198, 283, 365]. The output of this stage is a set of potentially overlapping image pairs

$$\mathcal{C} = \{(I_a, I_b, \mathcal{M}_{ab}) \mid I_a, I_b \in \mathcal{I},\ a < b\} \tag{2.43}$$

and their associated putative feature correspondences $\mathcal{M}_{ab} \subset \mathcal{F}_a \times \mathcal{F}_b$ between each potentially overlapping image pair.

Geometric Verification

The third stage geometrically verifies the potentially overlapping image pairs $\mathcal{C}$. Since feature matching is based solely on appearance, it is not guaranteed that corresponding features actually map to the same scene point due to wrong decisions in the feature matching process. Therefore, this stage verifies the putative feature matches by trying to robustly estimate a transformation that maps feature points between images based on their two-view geometry. Depending on the spatial configuration of an image pair, different mappings describe its geometric relation. A homography $\boldsymbol{H}$ describes the transformation of a purely rotating or a moving camera capturing a planar scene [123]. Epipolar geometry [123] describes the relation for a moving camera through the essential matrix $\boldsymbol{E}$ (calibrated) or the fundamental matrix $\boldsymbol{F}$ (uncalibrated), and can be extended to three views using the trifocal tensor [123]. If a valid transformation maps a sufficient number of features between the images, they are considered geometrically verified. Since the correspondences from feature matching are often highly outlier-contaminated, robust estimation techniques, such as RANSAC [85], are required. The output of this stage is a set of geometrically verified image pairs

$$\bar{\mathcal{C}} = \left\{ \left(I_a, I_b, \bar{\mathcal{M}}_{ab}, \boldsymbol{G}_{ab}\right) \mid (I_a, I_b, \mathcal{M}_{ab}) \in \mathcal{C},\ |\bar{\mathcal{M}}_{ab}| \geq N_{ab} \right\} \tag{2.44}$$

with their associated inlier correspondences $\bar{\mathcal{M}}_{ab}$ and their description of the geometric relation $\boldsymbol{G}_{ab} \in \{\boldsymbol{H}, \boldsymbol{F}, \boldsymbol{E}\}$. To decide on the appropriate geometric relation G_{ab}, decision criterions like GRIC [341] or methods like QDEGSAC [88] can be used. A minimum number of N_{ab} inlier correspondences are required in order to filter non-overlapping image pairs. Two-view scene points $\mathcal{X}_{ab}$ can be constructed by triangulation of the inlier correspondences $\bar{\mathcal{M}}_{ab}$.

The set of geometrically verified images $\bar{\mathcal{C}}$ can be transformed into the scene graph [129, 198, 282, 311], which serves as the input to the subsequent reconstruction stage. In the scene graph, overlapping images and scene points are nodes. Directed edges between images specify scene overlap and are annotated with an appropriate model describing the two-view geometry. Undirected edges indicate visibility between images and scene points. Note that scene points can be visible in more than two views by considering transitive correspondence information.

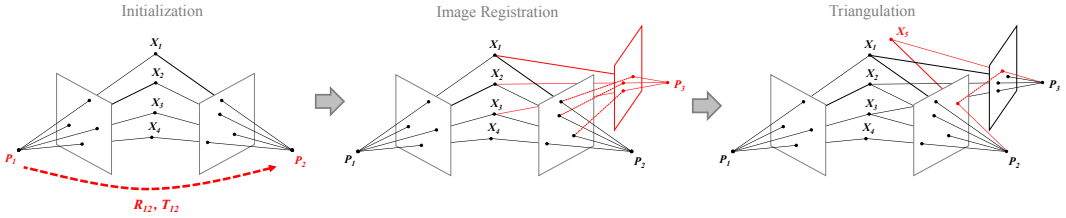

Figure 2.11: The initialization, image registration, and triangulation stages of an incremental sparse reconstruction pipeline.

2.4.2 Sparse Reconstruction

The input to the incremental reconstruction stage is the scene graph generated by the correspondence search stage. The outputs are the intrinsic and extrinsic calibration

$$\mathcal{P} = \{\boldsymbol{P}_c = \boldsymbol{K}_c \begin{bmatrix} \boldsymbol{R}_c & \boldsymbol{T}_c \end{bmatrix} \mid c = 1...N_P\} \tag{2.45}$$

for reconstructed images in a common reference frame and the triangulated scene structure as a set of points

$$\mathcal{X} = \{\boldsymbol{X}_k \in \mathbb{R}^3 \mid k = 1...N_X\} \ . \tag{2.46}$$

Note that, without prior information about the scene, the datum of the reference frame of the reconstruction has an arbitrary scale, orientation, and location. Intuitively, one might think that the absolute extrinsic camera calibrations could be determined by simply chaining the relative two-view transformations of overlapping images in the scene graph. The inherent problem with this approach is the scale ambiguity of each two-view reconstruction and the accumulation of errors due to imperfect two-view reconstructions. There are two fundamentally different approaches to solve this problem: incremental and global reconstruction methods. On the one hand, incremental algorithms initialize the reconstruction from a two-view reconstruction and then repeatedly register new images into the existing reconstruction followed by triangulation, filtering, and a refinement using bundle adjustment to reduce accumulated errors. On the other hand, global reconstruction methods formulate the recovery of the absolute camera motion as a joint optimization over all the relative two-view geometries in the scene graph. This optimization is decomposed into the independent recovery of global, absolute camera orientations before estimating the global, absolute camera locations. In a final stage, global methods refine the reconstruction using a full global bundle adjustment. Typically, global reconstruction methods are more efficient but, in general, not as robust as incremental methods. The main reason for the slower runtime of incremental methods, especially for large reconstructions, are the repeated bundle adjustments. However, the repeated bundle adjustments and continuous refinement of the scene during the incremental reconstruction typically lead to more robustness with respect to uncalibrated input images and outliers in the two-view reconstruction. To enable the

reconstruction of highly unstructured input images, this thesis considers the incremental reconstruction paradigm, which is described in further detail in the following sections (see also Figure 2.11).

Initialization

The incremental procedure initializes the reconstruction with a carefully selected image pair [29, 309] from the scene graph. Choosing a suitable two-view geometry from the scene graph is critical, since the reconstruction may never recover from a bad initialization. For example, choosing an image pair with pure rotational motion results in a degenerate triangulation due to parallel viewing rays. Moreover, the robustness, accuracy, and performance of the incremental reconstruction algorithm directly depends on the seeding location of the initialization process. Initializing from a densely connected location in the scene graph with many overlapping images typically results in a more robust and accurate reconstruction due to increased redundancy. In contrast, initializing from a more sparsely connected location results in lower runtimes, since the repeated bundle adjustments deal with overall sparser problems over the entire reconstruction process. The output of the initialization stage is a metric reconstruction of two images (i.e., $|\mathcal{P}| = 2$) and a set of two-view scene points triangulated from the verified feature correspondences.

Image Registration

Starting from a metric reconstruction, new images can be registered to the current scene by solving the camera calibration problem using feature correspondences to triangulated points in already registered images (2D-3D correspondences). The camera calibration problem involves estimating the absolute orientation $\boldsymbol{R}_c$ and location $\boldsymbol{T}_c$ and, for an uncalibrated camera, its intrinsic parameters $\boldsymbol{K}_c$ with an optional set of parameters accounting for lens distortion effects. In each iteration of the incremental reconstruction algorithm, the set of registered images $\mathcal{P}$ is thus extended by the pose $\boldsymbol{P}_c$ of a newly reconstructed image. Since the 2D-3D correspondences are often outlier-contaminated, the pose for calibrated cameras is usually estimated using RANSAC and a minimal pose solver, e.g. [102, 185]. For uncalibrated cameras, various minimal solvers, e.g. [46], or sampling-based approaches, e.g. [146], exist. Similar to the initialization strategy, the order in which images are reconstructed has essential impact on the robustness and efficiency of the algorithm. In Section 7.3, we present an efficient and robust method for selecting the next best view in incremental reconstruction, which enables for accurate pose estimation and reliable triangulation.

Triangulation

In order to register a new image to the reconstruction, it must observe existing scene points in the current reconstruction. In addition, it may also observe scene points

that are not yet reconstructed. If those points are seen by at least one other registered image, triangulation of these points can extend the scene coverage by extending the set of points $\mathcal{X}$. In turn, those points can be used to register new images. In an incremental reconstruction pipeline, image registration and triangulation rely on each other's outputs, since scene points are needed as input for camera calibration and vice versa. It is therefore crucial to perform the two steps in an alternating fashion.

Bundle Adjustment

Image registration and triangulation are separate procedures, even though their products are highly correlated – uncertainties in the camera pose propagate to triangulated points and vice versa, and additional triangulations may improve the initial camera pose through increased redundancy. Without further refinement, incremental reconstruction results usually drift quickly to a non-recoverable state. To mitigate accumulated errors, bundle adjustment is used as a joint non-linear refinement of camera parameters $\mathcal{P}$ and point parameters $\mathcal{X}$. Bundle adjustment is a computationally demanding step in the incremental reconstruction pipeline. The special structure of parameters in bundle adjustment problems motivates the Schur complement trick [43], in which one first solves the reduced camera system and then updates the points via back-substitution. This scheme is commonly more efficient, since the number of cameras is usually smaller than the number of points. In general, there are two choices for solving the system: exact and inexact step algorithms. Exact methods solve the bundle adjustment system by storing and factoring it as a dense or sparse matrix [59, 199] with a space complexity of $O(N_P^2)$ and a time complexity of $O(N_P^3)$. Inexact methods approximately solve the system, usually by using an iterative solver, e.g. preconditioned conjugate gradients, which has $O(N_P)$ time and space complexity [6, 366]. Direct algorithms are the method of choice for up to a few hundred cameras but they are too expensive in large-scale settings. While sparse direct methods reduce the complexity by a large factor for sparse problems, they are prohibitive for large unordered image collections due to typically much denser connectivity graphs [6, 311]. In this case, indirect algorithms are the method of choice.

Outlier Filtering

Despite careful filtering with robust estimation during two-view geometry reconstruction, image registration, and triangulation, outliers remain in the reconstruction even after bundle adjustment. For example, repetitive features along epipolar lines oftentimes cause mismatches and survive robust two-view geometry estimation as valid triangulations. Using the redundancy from more than two views allows to more strictly filter those mismatches. Frequent outlier filtering is crucial in that it avoids misregistrations of images and accumulated drift in the following iterations.

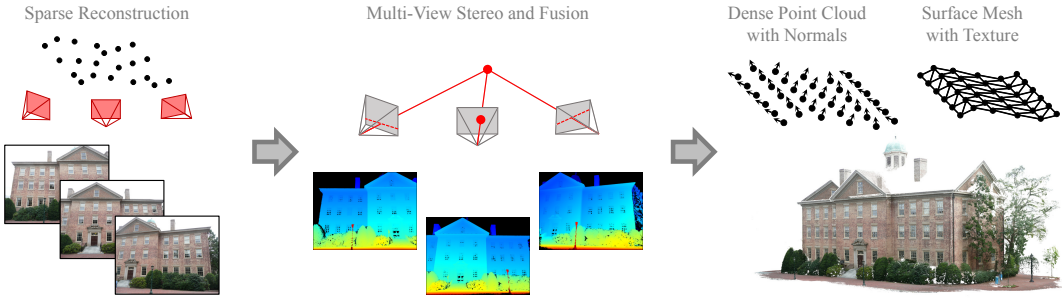

Figure 2.12: An overview of dense reconstruction using multi-view stereo for depth and normal map extraction, followed by a fusion and surface meshing step.

2.4.3 Dense Reconstruction

For efficiency reasons, the sparse reconstruction stage typically uses only relatively few distinct image features to recover the structure and motion. As a consequence, the reconstructed set of scene points is rather sparse and only represents a very rough approximation of the captured real world. The goal of the final dense reconstruction stage is to densify this sparse approximation to a richer representation, such as a dense point cloud or a textured surface mesh.

There are several different approaches to dense modeling, but the underlying principle of all such methods is to reconstruct a scene surface that best explains the reprojected appearance in the reconstructed images. One major difference between many methods lies in the parameterization of the problem. One type of approaches parameterizes the problem in image space and recovers per-pixel depth (and normal) estimates (e.g., [49, 97, 99, 110, 144, 289]). The other type of approaches optimizes for the surface directly in the scene space (e.g., [95, 116, 347, 377]). Inherently, the two parameterizations have several advantages for specific application scenarios. First, the number of parameters to estimate in image space are limited by the number of pixels in the image, while a parameterization of problem in scene space is not clearly confined. Therefore a parameterization in scene space is often not easily scalable for large scenes, as there is no prior knowledge about which parts of the scene are occupied or empty. In contrast, every pixel in an image observes an object and thus must naturally have a corresponding scene point and therefore also a valid depth (and normal) value. Nevertheless, parameterizing the problem directly in scene space has the potential benefit of avoiding redundant computation, when a surface is observed from many cameras. Additionally, it allows for advanced multi-view occlusion reasoning through raycasting and, in many instances, it is easier to formulate higher-level semantic priors directly in scene space.

This thesis focuses on methods for efficient and robust modeling from unstructured and large-scale image collections. The following sections thus focus on a dense modeling pipeline that is a hybrid of the two discussed parameterizations. The de-

scribed algorithm decomposes the problem into three main steps (see Figure 2.12): the recovery of dense depth maps for each image independently, the fusion of those depth maps into a globally consistent point cloud, and a final surface meshing and texturing step. Due to the large variety of approaches in dense modeling, the following description purposefully stays on a relatively high conceptual level and we refer the reader to existing literature [94] for a more comprehensive overview.

Multi-View Stereo

The goal of multi-view stereo is to compute a depth (and normal) estimate for every pixel of every camera calibrated during the sparse reconstruction stage. Given the already estimated camera calibration in the sparse reconstruction stage, the problem of recovering a depth value for each pixel for a pair of images simplifies significantly, since the full and precise epipolar geometry between the views is already known. Two-view stereo algorithms leverage the known epipolar geometry to compare every pixel in a reference image with all pixels along the corresponding epipolar line in the other image. Oftentimes, a small patch around each pixel is considered in order to infer more robust measures of appearance similarity. Theoretically, the most similar pixel along the epipolar line uniquely determines the location of the pixel in the scene through triangulation. In practice though, often many pixels along the epipolar line have similar appearance due to texture-less surfaces, repetitive structures, specularities, etc. Furthermore, some pixels might not be visible in the other image at all because of occlusion. Therefore, the naive approach of simply selecting the most similar pixels usually leads to noisy depth (and normal) maps with many outliers.

A common method to counteract this problem is to enforce surface smoothness priors. That means whenever one cannot find a sufficiently similar pixel in the other image or when there are multiple ambiguous pixels, we prefer a smooth surface over a noisy one. For example, a low-level geometric prior might simply encourage similar depths for neighboring pixels while a high-level semantic prior might encourage a flat vertical surface for walls. Such priors usually produce more accurate results, because they constrain the solution space by coupling the estimation of related pixels through incorporating prior scene knowledge. Another approach to tackling the problem of ambiguity is to accumulate evidence over more than two views. Multi-view stereo generalizes the concept of two-view stereo to multiple views. By constructing the multiple epipolar geometries to all other overlapping views, one can determine the corresponding epipolar lines to multiple views and aggregate a stronger and more robust measure of appearance similarity. Note that in the multi-view scenario, it is important to compute the appearance similarity in a robust manner in order to handle occlusion, because the scene is usually not completely visible in all images. Selecting suitable neighboring views for computing the appearance consistency is a crucial step in multi-view stereo. On the one hand, it is important to have sufficient scene overlap with little radiometric and geometric distortions. On the other hand, a sufficient baseline between the views is required for accurate triangulation. Usually,

view selection is done based on shared visibility information from the sparse scene reconstruction.

Per-pixel surface normal information can either be deduced from gradient information in the estimated depth map or by directly incorporating normal inference into the depth estimation procedure. A small patch around each pixel in the reference image corresponds to a small, oriented plane at a certain depth in the scene. Constructing the corresponding homography for a specific scene plane allows to warp the reference image patch into a (potentially non-rectangular) region in the neighboring views. The joint maximization of appearance similarity over the location along the epipolar lines and the scene plane orientation yields an estimate of depth and normal for each pixel. This parameterization also enables the use of more accurate second-order smoothness priors along oriented surfaces.

Multi-View Fusion

The second step in the dense modeling stage fuses the depth and normal estimates from image space into a dense point cloud in scene space. The fusion step is necessary in order to obtain a globally consistent and compact representation of the scene. While priors in multi-view stereo reduce the noise and the amount of outliers of the depth and normal maps significantly, the fusion step further increases the quality of the results by leveraging the redundancy of multiple views. Enforcing consistency among the depth and normal estimates in multiple views enables to filter most of the remaining outliers. Moreover, fusing consistent pixels from multiple views leads to a more compact and accurate representation of the scene. The output of the fusion step is a dense point cloud with surface appearance and normal information. This representation can be used for multiple applications, including visualization and rendering or image-based localization.

Surface Reconstruction

For many practical applications, a point cloud is an inadequate representation and instead a surface mesh model is required. Examples include collision detection in robotics or games, efficient and high-fidelity rendering, etc. For most scenes, a mesh parameterization leads to a much more compact representation due to the structure of the real world. While a point cloud requires to densely model regular surfaces with many point instances, a suitable surface mesh parameterization can potentially model an entire object using a single entity. A simple triangular mesh can typically very well approximate the geometry of a scene with locally planar surfaces. Creating surface texture maps from the images typically leads to superior rendering quality as compared to point cloud renderings.

Part II

Correspondence Search

3 Evaluation of Hand-Crafted and Learned Local Features

Matching local image features is a crucial step in many computer vision applications and is especially important for many stages of an image-based 3D modeling pipeline, e.g., in Structure-from-Motion and Multi-View Stereo [4, 129, 252, 284, 288, 289], image retrieval [243, 270, 337, 340], and image-based localization [266, 271, 386]. In many of these applications, the overall system performance strongly depends on the quality of the initial feature matching stage. Consequently, determining which local feature descriptors offer the most discriminative power and the best matching performance is of significant interest to a large part of the computer vision community.

For more than a decade, SIFT [200] has arguably been the most popular feature descriptor for such tasks. Recently, the ability of neural networks to learn feature representations from data that are superior to prior hand-crafted ones has led to significant progress in the field of computer vision, e.g., in object detection and recognition [106, 172, 259]. Consequently, neural networks have also been applied to the problem of descriptor learning [24, 115, 178, 301] in order to derive more discriminative representations for local features. The resulting methods demonstrate clear improvements over standard hand-crafted representations, such as SIFT [200], SURF [26], or DAISY [334]. However, there is usually no direct comparison with more advanced hand-crafted SIFT variants such as RootSIFT [15], RootSIFT-PCA [47], or DSP-SIFT [80]. Moreover, learned descriptors are typically evaluated on the patch classification benchmark from Brown et al. [44]. The task measures how well a descriptor can distinguish between related and unrelated patches based on their distance in descriptor space. Yet, a better performance on this benchmark does not necessarily imply a better matching quality, as shown by Balntas et al. [24]. For example, pruning steps such as Lowe's ratio test [200] or mutual nearest neighbor constraints might compensate for a higher false positive matching rate in terms of descriptor distance. Similarly, reaching a better average matching performance does not automatically imply a better performance in terms of subsequent processing steps. In the context of image-based 3D modeling, finding additional correspondences for image pairs where SIFT already provides enough matches does not necessarily result in more accurate or complete reconstructions. At the same time, descriptors with a better average matching performance might still not find enough correspondences to be able to handle hard image pairs where SIFT fails.

In this section, we present a thorough experimental evaluation of learned and advanced hand-crafted feature descriptors in order to better understand their per-

formance. In detail, this section makes the following contributions:

- We provide a more detailed study of the matching performance of the different descriptors using a wider range of evaluation criteria and scenes than previous evaluations such as [24].
- Besides analyzing the matching quality in isolation of further processing steps, we also investigate the impact of different descriptors on the challenging and more practical task of image-based reconstruction. For example, this allows us to better determine whether learned descriptors can help to register hard images, e.g., photos depicting the scene under strong viewpoint or illumination changes. In addition, we are interested to understand to what extent a better matching performance affects the outcome of further processing stages, e.g., the accuracy and completeness of the models produced by SFM and MVS.
- Our evaluation confirms that, as expected, learned descriptors often surpass SIFT on all evaluation metrics. However, we also observe that advanced versions of hand-crafted descriptors [47, 80] perform on par or better than the state-of-the-art learned feature descriptors, especially in the more complex SFM scenarios. As such, our section demonstrates that there is still significant room for improvement for learning more powerful feature descriptors.
- To facilitate further research in developing better descriptors, we make our benchmark publicly available.[1] This includes a large database corresponding local image patches.

3.1 Related Work

In the following, we provide a detailed overview of descriptor learning methods and a review of the hand-crafted descriptors used as baselines. In addition, we discuss the existing evaluation protocols and their limitations.

3.1.1 Descriptor Learning

Descriptor learning is usually formulated as a supervised learning problem. Given a set $\mathcal{P}$ of positive pairs and a set $\mathcal{N}$ of negative pairs, the objective is to learn a representation in which the descriptors belonging to the same physical object are close in descriptor space while unrelated descriptors are far apart. The approaches often differ in the exact definition of this property. For example, Simonyan et al. [302] use the *margin constraint*

$$d(\boldsymbol{p}_1, \boldsymbol{p}_2) + \tau < d(\boldsymbol{n}_1, \boldsymbol{n}_2) \;\; \forall (\boldsymbol{p}_1, \boldsymbol{p}_2) \in \mathcal{P} \;\; , (\boldsymbol{n}_1, \boldsymbol{n}_2) \in \mathcal{N} \;\; , \tag{3.1}$$

where $d(,)$ is a distance metric (usually L_2) and $\tau \in \mathbb{R}_{>0}$ is a margin. This approach can easily be extended to different types of positives and negatives, e.g., by using a

[1] `http://www.cvg.ethz.ch/research/local-feature-evaluation/`

larger margin τ_2 for random negative pairs and a smaller one $\tau_1 < \tau_2$ for negative pairs with a small initial distance [244]. Enforcing a small intra-class variance for descriptors belonging to the same physical point and a large inter-class variance for unrelated descriptors can also be expressed via a *hinge embedding* [221] or *contrastive* loss [113]

$$l(\boldsymbol{d}_1, \boldsymbol{d}_2) = \begin{cases} d(\boldsymbol{d}_1, \boldsymbol{d}_2) & \text{if } (\boldsymbol{d}_1, \boldsymbol{d}_2) \in \mathcal{P} \\ \max\left(0, \tau - d(\boldsymbol{d}_1, \boldsymbol{d}_2)\right) & \text{if } (\boldsymbol{d}_1, \boldsymbol{d}_2) \in \mathcal{N} \end{cases}, \tag{3.2}$$

which tries to enforce a minimum distance $\tau > 0$ between unrelated descriptors. As an alternative to working with pairs of descriptors, it is also possible to operate on triplets $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{n})$, with $(\boldsymbol{p}_1, \boldsymbol{p}_2) \in \mathcal{P}$ and $(\boldsymbol{p}_1, \boldsymbol{n}), (\boldsymbol{p}_2, \boldsymbol{n}) \in \mathcal{N}$. Potential cost functions are the *margin ranking loss* [355]

$$l(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{n}) = \max\left(0, \tau + d((\boldsymbol{p}_1, \boldsymbol{p}_2) - d(\boldsymbol{p}_1, \boldsymbol{n})\right) \tag{3.3}$$

and the *ratio loss* [140]

$$l(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{n}) = \left(\frac{e^{d_p}}{e^{d_p} + e^{d_n}}\right)^2 + \left(\frac{e^{d_n}}{e^{d_p} + e^{d_n}}\right)^2, \tag{3.4}$$

where $d_p = d(\boldsymbol{p}_1, \boldsymbol{p}_2)$ and $d_p = d(\boldsymbol{p}_1, \boldsymbol{n})$. The latter tries to enforce that the distance between related descriptors is significantly smaller than the distance to an unrelated descriptor, without explicitly specifying a margin.

The input to the descriptor learning algorithm varies between the different approaches. For example, methods based on metric learning [356] often use a fixed descriptor representation as input and learn a discriminative metric for comparing descriptors [44, 244, 302, 319]. In contrast, approaches that learn a new descriptor representation usually operate on raw image patches [24, 301, 302, 384].

One way to obtain the large amount of training data required for learning is to extract positive and negative pairs from 3D models [44, 178, 319]. As a result of the reconstruction, each 3D point is associated with at least two image descriptors and their corresponding local patches. Consequently, the measurements from a single point form positive pairs while measurements from different 3D points are used to define negative pairs. While SFM already uses a descriptor, e.g. SIFT, to compute the pairwise feature matches used for reconstruction, the resulting models can still be used to learn more discriminative descriptors: Due to the transitivity of matching, a 3D point might be associated with patches A, B, and C. Correspondences might initially be obtained between A and B and between B and C, but not between A and C, e.g., due to a large viewpoint or illumination change. Thus, the data is suitable to learn a better descriptor that is able to directly match between A and C. An alternative to using SFM or MVS models is to use image retrieval techniques [243] to obtain the positive and negative pairs [244, 302].

3.1.2 Learned Descriptors

Learning Patch and Descriptor Embeddings

Given an image patch, descriptor learning can be formulated as finding a discriminative embedding into a new space. For example, PCA-SIFT [161] uses principal component analysis (PCA) to embed a gradient image of a patch while Lepetit and Fua [184] embed patches using a random forest. Obviously, embeddings can also be applied to already existing descriptors, e.g., (Root)SIFT-PCA [47] employs PCA to project (Root)SIFT descriptors into a lower dimensional space. Philbin et al. [244] learn both linear and non-linear discriminative projections into lower dimensional spaces based on margin constraints. The non-linearity is implemented using a neural network with a single hidden layer. Simonyan et al. [302] model the problem of learning a discriminative projection into a low-dimensional space as a convex optimization problem. The resulting linear projections outperform the non-linear ones from Philbin et al. [244]. While the other methods learn embeddings into Euclidean spaces, Strecha et al. [319] propose a discriminative projection into a binary space where Hamming distances can be computed very efficiently. For our experimental evaluation, we use both RootSIFT-PCA and the projection learned by Simonyan et al. (in conjunction with the ConvOpt descriptor [302]). The former serves as a baseline method representing advanced hand-crafted descriptors.

Learning Pooling Regions

Hand-crafted and learned descriptors are constructed by applying a series of filter banks to an image patch, followed by pooling (e.g., into histogram bins in the case of SIFT) and normalization. Fixing the arrangement, e.g., on a polar grid, and the positions of the pooling regions, Brown et al. [44] learn descriptors by optimizing over the remaining pooling parameters such as the size of the regions. Following a similar approach, Trzcinski et al. [343] employ a boosting approach to learn binary descriptors from a set of weak learners that represent pooling strategies. Simonyan et al. [302] model the problem of learning the pooling regions as a convex energy minimization problem based on the margin constraint from Equation 3.1. The size of the resulting *Convex Optimization* (*ConvOpt*) descriptor is controlled by enforcing sparsity when selecting a subset of the pooling regions. More complex descriptors can be trained by combining the learning of pooling regions with learning (linear) discriminative projections [44, 302]. In this section, we use the *ConvOpt* descriptor, combined with a discriminative projection into a lower dimensional space [302], as a representative of approaches that learn pooling regions. It is selected since it outperforms both Brown et al. [44] and Trzcinski et al. [343]. As a baseline for hand-crafted descriptors, we employ *DSP-SIFT* [80], a variant of SIFT that pools gradients over multiple scales rather than only the scale at which the SIFT feature was detected.

Learning Filter Banks

While the approaches described in the previous paragraph [44, 302, 343] use a fixed set of filters and learn the pooling regions, approaches based on Convolutional Neural Networks (CNN) [24, 301] fix the pooling strategy and instead learn the filter banks. Simo-Serra et al. [301] use a siamese architecture [41] with a 3-layer CNN to minimize the contrastive loss from Equation 3.2. Simo-Serra et al. notice that most randomly sampled negative patch pairs are easy to separate. In order to train their *Deep Descriptor* (*DeepDesc*), they thus mine for hard positive and negative pairs that can be used during learning. While Simo-Serra et al. [301] use pairs of patches, Balntas et al. [24] use a triplet network [140] consisting of two convolutional followed by one fully connected layer. Their *TFeat* descriptor is trained using hard-negative mining and Balntas et al. propose versions based on the margin ranking loss from Equation 3.3 or the ratio loss from Equation 3.4. We use both *DeepDesc* and *TFeat* trained with the margin ranking loss for our evaluation.

Joint Descriptor and Metric Learning

The approaches described above learn functions that map local image patches to discriminative descriptors embedded in a Euclidean space. As such, they employ the L_2 distance to compare descriptors. An alternative strategy is to jointly learn a descriptor representation and a distance metric that can be used to compare them [115, 382, 384]. Such approaches are potentially more powerful as deep neural networks can be used to implement a non-linear distance metric. However, this strength is also a great draw-back as it requires a forward pass through the learned model for comparing each pair of descriptors. Not only is such a pass computationally more complex than computing a single L_2 distance, but the network also prevents the use of traditional spatial subdivision schemes for fast (approximate) nearest neighbor search, such as kd-trees or hierarchical k-means trees [226]. This limits the scalability of methods that jointly learn a descriptor representation and a metric for comparison. In this section, we evaluate datasets with millions of descriptors. Consequently, we focus on learned descriptors that can be efficiently compared via the L_2 distance.

Joint Detector and Descriptor Learning

The methods discussed above take an image patch as input and compute the corresponding feature descriptor as output. Hence, they are not tied to a single detector providing the patch but could easily be combined with any feature detector. However, jointly optimizing both the descriptor and detector should provide better results as the detector is trained to fire on regions that can be matched by the descriptor and vice versa. Recently, Yi et al. [178] proposed such an approach by combining the *DeepDesc* descriptor with a Difference-of-Gaussians (DoG)-like detector [200]. We include their LIFT feature in our evaluation.

3.1.3 Evaluation Protocols

Mikolajczyk et al. [216] evaluate affine region detectors by introducing standard metrics and small-scale datasets under various photometric and geometric image transformation. Later, Mikolajczyk and Schmid [215] extend this evaluation to several local descriptors. As a superset of this evaluation, Heinly et al. [127] evaluate binary descriptors and propose additional metrics and datasets.

Most learned descriptors are evaluated on the patch pair classification benchmark [44], which measures the ability of a descriptor to discriminate positive from negative patch pairs. The standard protocol of the benchmark is to generate the ROC curve by thesholding the distance values between pairs of patches. The final reported number is the false positive rate at 95% true positive rate (FPR95). However, as shown by Balntas et al. [24], a better FPR95 score does not automatically translate to better nearest neighbor matching because of usual filtering steps, such as Lowe's ratio test or the mutual nearest neighbor constraint. In practice, feature matching is typically followed by a geometric verification stage to prune outliers [4, 129, 243, 266, 270, 284, 288]. Due to the exponential complexity in the number of outliers [85], it is practically more important to have good precision for manageable runtimes of geometric verification. The authors of LIFT and TFeat make a first step to provide more insight into the practicality of the descriptors in a real-world application. Both evaluate their performance in terms of image-based reconstruction on the Strecha benchmark [319]. As we will show in this section, this dataset is rather easy and provides only little practical insight.

To facilitate comparability with the evaluations by Mikolajczyk and Heinly et al., we follow their benchmark protocol to evaluate the raw matching performance on a per image pair basis. As the core contribution of this section, we also study the impact of matching performance in the more practical setting of an image-based reconstruction pipeline [284, 289] using challenging small- and large-scale datasets. As part of the image-based reconstruction pipeline, SFM uses descriptor matching in the first stage to produce a graph of corresponding features in multiple views. Hence, all subsequent stages strongly rely on a good descriptor representation. Motivated by this, we derive evaluation metrics in all stages of the pipeline: feature matching, geometric verification, image retrieval, and sparse and dense modeling, in order to give new practical insights into the performance of the evaluated descriptors, as detailed in following section.

3.2 Evaluation

In the first part of this section, we detail and motivate the proposed evaluation protocol. The second part then presents and discusses the results of the evaluation.

3.2.1 Setup and Protocol

The following paragraphs describe the setup of our evaluation to ensure repeatability of the experiments. The entire protocol is provided to the public as an evaluation framework to foster future research in feature learning.

Evaluated Descriptors

We evaluate the performance of RootSIFT (short *SIFT*) [15] as a baseline descriptor, and RootSIFT-PCA (short *SIFT-PCA*) [47] and *DSP-SIFT* [80] as two representatives of advanced hand-crafted features. To evaluate the learned descriptors, we selected four state-of-the-art methods from the different groups of descriptor learning approaches: *ConvOpt* [302], *DeepDesc* [301], *TFeat* [24], and *LIFT* [178]. All features are evaluated using the same standardized test setup, as specified in the following.

Feature Detection

To ensure comparability between the evaluated descriptors, we use the standard SIFT keypoint detector for all descriptors but LIFT, which implements its own DoG-like detector. The SIFT detector uses DoG and we use 4 octaves starting with a two times up-sampled version of the original image, 3 scales per octave, a peak threshold of $\frac{0.02}{3}$, an edge threshold of 10, and a maximum of 2 detected orientations per keypoint location. These values have been optimized for the purpose of SFM and are, e.g., used as defaults in COLMAP [284, 289]. Following standard procedure by the original methods, we then extract 64×64 pixel patches as the input to each descriptor. Note that all descriptors have been learned based on DoG keypoints. We experimented with different detector settings for LIFT and found that the defaults by the authors performed best. On average, DoG detects 5,262 and LIFT 4,173 features for the images in the Oxford5k dataset [243].

Descriptor Matching

Throughout all experiments, the L_2 distance serves as an efficient distance metric to calculate the similarity between two descriptors. To compute the correspondences between pairs of images, we enforce mutual nearest neighbors, i.e., a corresponding descriptor in one image must be the nearest neighbor for the corresponding descriptor in the other image and vice versa. This has been shown to reduce the amount of false correspondences for ambiguous structures and significantly improved the results for all descriptors [127, 284]. In contrast to standard practice in SIFT matching, we do not enforce the ratio test by pruning descriptors whose top-ranked nearest neighbors are very similar. The reason being that the ratio test is highly dependent on the distribution of descriptor distances [200]. Preliminary experiments showed that the ratio test is not generally applicable to any of our evaluated descriptors but SIFT. For the smaller datasets with up to 2,000 images, we exhaustively compute

correspondences between all pairs of images. For the larger datasets, we use Bag-of-Words (BoW) to match each image only against a fixed number of top-ranked neighbor images. For the nearest neighbor search, we employ a state-of-the-art image retrieval system [287] using Hamming embedding [148] and visual burstiness weighting [150]. Following standard procedure, we ensure that the vocabulary is trained on a completely unrelated image collection. Correspondingly, we use a vocabulary of 262,144 words with a branching factor of 512 trained offline on Oxford5k [243] for all the experiments. To ensure a good quantization of the descriptor space and to evaluate the performance of each descriptor on the task of image retrieval, we train a custom vocabulary for each descriptor.

Geometric Verification

Descriptor matching as described in the previous paragraph is solely based on appearance information. For the purpose of SFM and to quantify the matching performance on a per image pair basis, we estimate the two-view geometry and determine the resulting inlier correspondences using the multi-model geometric verification approach described in [284]. Moreover, we are interested in quantifying the matching performance in the practical context of image-based reconstruction. Towards this goal, we use the successfully verified image pairs with a minimum of 15 inlier feature correspondences as the input to COLMAP [284, 289]. While both the sparse and dense reconstruction results provide insight into the practicality of the descriptors in a real-world application, SFM also implements a much stricter and more accurate geometric verification tool using multi-view information, as compared to the initial two-view verification. Hence, we also evaluate key metrics of the resulting sparse and dense reconstructions produced by SFM and MVS, as detailed in the following.

Matching Metrics

Equivalent to the binary descriptor evaluation by Heinly et al. [127], we first evaluate the raw matching performance on a per image pair basis using the standard metrics *Putative Match Ratio*, *Precision*, *Matching Score*, and *Recall*. First, the *Putative Match Ratio = #Putative Matches / #Features* quantifies the selectivity of the descriptor in terms of the fraction of the detected features initially identified as a match. Second, the *Precision = #Inlier Matches / #Putative Matches* defines the inlier ratio of the putative matches, as determined by geometric verification. The *Matching Score = #Inlier Matches / #Features* defines the number of initial features that will result in inlier matches. Last, the *Recall = #Inlier Matches / #True Matches* describes the number of identified ground-truth matches. We refer the reader to Heinly et al. [127] for more details and an in-depth motivation of these metrics.

Reconstruction Metrics

In addition to evaluating the raw matching performance on individual image pairs, we also evaluate the performance of the different descriptors in the practical and more challenging setting of image-based reconstruction. Typically, the image-based reconstruction pipeline first uses SFM to calibrate the cameras of the input images and to infer a sparse model of the scene. Then, the output of SFM serves as the input to MVS to obtain a dense representation of the scene, e.g., in the form of depth maps, a dense point cloud, or a meshed surface model. Generally, the ultimate goal of image-based reconstruction is to produce high-quality 3D models. The quality of SFM results strongly depends on accurate and complete two-view correspondences as input, and MVS relies on an accurate and complete SFM reconstruction [284]. Thus, SFM and MVS results are good indicators for the descriptor performance in the initial feature matching stage. Furthermore, by chaining two-view correspondences into a graph of feature tracks [284], SFM can exploit multi-view redundancy to more reliably verify the validity of correspondences. To evaluate the completeness and accuracy of the reconstruction results, we determine a number of key metrics: First, the number of *registered images* and *sparse points* quantify the completeness of the reconstruction. A larger number of registered images enables more complete MVS reconstruction and a larger number of 3D points with many image observations constitute a more complete and accurate scene representation. Second, we determine the number of *observations per image*, i.e., the number of verified image projections of sparse points, and the *track length*, i.e., the number of verified image observations per sparse point. These two metrics are crucial for an accurate calibration of the cameras and reliable triangulation, as they provide redundancy in the estimation. Third, bundle adjustment stands at the core of SFM as a joint non-linear refinement of the cameras and points. The overall *reprojection error* in bundle adjustment indicates the accuracy of the reconstruction and is mainly impacted by the accuracy and redundancy of the input data, which depend on the completeness of the graph of feature correspondences and the keypoint localization accuracy. For a subset of the datasets, ground-truth camera locations are available, and we evaluate the mean *metric pose accuracy* of the camera locations by aligning the reconstructed model to the ground-truth using robust 3D similarity transformation estimation. Last, the MVS problem boils down to dense correspondence estimation between multiple views. To produce accurate and complete results, MVS requires an accurate intrinsic and extrinsic camera calibration. Moreover, more registered images provide additional multi-view photo-consistency constraints and lead to more complete results. Hence, we determine the number of reconstructed *dense points* as a single measure of the overall completeness of the reconstruction and the accuracy of the SFM results. In addition, we have ground-truth depth maps for a subset of the datasets to also directly evaluate the metric accuracy and absolute completeness of the dense reconstruction results.

Datasets

We evaluate all descriptors on existing small- and large-scale benchmark datasets. For the two-view evaluation, we follow the evaluation protocol and the datasets provided by Heinly et al. [127]. The benchmark tests the descriptor performance with respect to different types and levels of photometric and geometric image transformations (image blur, exposure, white balance, JPEG compression, scale and/or rotation, planar and non-planar geometry, illumination, etc.). For the reconstruction evaluation, we employ various existing benchmark datasets. The well-known MVS benchmark by Strecha et al. [322] (*Fountain* and *Herzjesu*) consists of around 10 high-resolution images per dataset with highly accurate ground-truth camera locations and dense depth maps. To evaluate the completeness and accuracy of the depth maps, we follow the evaluation protocol by Hu and Mordohai [144]. Next, we evaluate the performance on the *South Building* dataset [116], which consists of 128 highly overlapping images with mostly repetitive scene structure captured by the same camera in a structured pattern around the building. Finally, Internet photo collections present the descriptors with more challenges due to the high variance in the input data. We test the descriptors on the large-scale Internet datasets by Wilson and Snavely [362]. Each dataset contains several thousand images of well-known landmarks across the world collected from Flickr. To simulate a harder matching and reconstruction scenario, each dataset is embedded into a distractor set of unrelated images. As such, the descriptors must generalize well to the heterogeneity of Internet data to robustly handle effects such as large illumination and viewpoint changes, repetitive structure, image compression and distortion artifacts, or unrelated distractor images. Finally, we evaluate the reconstruction performance on the large-scale *Cornell* dataset by Crandall et al. [75]. The dataset consists of 6,514 unstructured and uncalibrated images of the Cornell campus. The images were taken in a relatively sparse pattern during different seasons and times of the day and thus pose extreme challenges to the descriptors in terms of illumination and viewpoint changes. A subset of 348 images is equipped with ground-truth camera locations obtained through surveying methods that we use to evaluate the pose accuracy. We use the *Oxford5k* dataset [243] to train the visual vocabulary for image matching.

Implementation

To enable comparability in the timings, all experiments were conducted on the same machine with two 14-core Intel E5-2697 2.60GHz CPUs, 512GB of RAM, and 4 NVIDIA Titan X. We use the SIFT implementation by VLFeat [349] and, for all other descriptors, the open-source implementations and models provided by the authors. Traditionally, the descriptor learning models are trained on the multi-view correspondence dataset by Brown et al. [44]. We choose their best-performing pretrained models, if multiple are provided. The descriptor matching uses an efficient GPU implementation, and we use COLMAP [284, 289] for the SFM and MVS evaluation, while CMVS [95] is used to cluster the larger datasets into more manageable

	SIFT	SIFT-PCA	DSP-SIFT	ConvOpt	DeepDesc	TFeat	LIFT
Dimensionality	128	80	128	73	128	128	128
Size [bytes]	128	320	512	292	512	512	512
Platform	CPU	CPU	CPU	GPU	GPU	GPU	GPU
Extraction [s]	9.3	10.5	23.7	49.9	24.3	11.8	212.3
Matching [s]	0.14	0.11	0.14	0.10	0.14	0.14	0.14

Table 3.1: Key properties of the evaluated local feature descriptors. Average timings reported for the Oxford5k dataset. Extraction speed includes keypoint detection and are specified per image. Matching speed is specified per image pair.

image clusters for the dense reconstruction.

3.2.2 Results and Discussion

Computational Efficiency

Table 3.1 summarizes the key performance properties for each descriptor including timings, memory requirements, etc. on the Oxford5k dataset. The memory footprint and the descriptor dimensionality have important implications for the required storage capacity for large-scale datasets, since we evaluate datasets containing thousands of images with millions of descriptors. For example, the raw SIFT keypoints and descriptors for Cornell already comprise $\approx$ 11GB of data. Furthermore, the descriptor dimensionality impacts the speed of the descriptor matching, which in practice has squared complexity in terms of the number of features per image when using efficient exhaustive GPU matching. Due to its low dimensionality, ConvOpt provides $\approx$ 40% faster feature matching. Among the different descriptors, there is a large variance in extraction speed. In theory, when implemented efficiently, both SIFT-PCA and DSP-SIFT have only small overhead over standard SIFT. While ConvOpt is relatively slow to extract, it is significantly faster in the matching stage due to its low dimensionality. Conversely, TFeat is relatively fast to extract and slower in the matching stage, similar to the other descriptors with 128 dimensions. LIFT is the slowest method by a large margin. In general, the extraction of the hand-crafted descriptors is much faster as compared to the learned features despite running on the CPU. As such, the learned features are currently not a practical alternative for processing millions of images, such as in the streaming-based reconstruction pipeline by Heinly et al. [129] who report a throughput of 20 images per second on a single GPU.

Image Matching

Table 3.2 shows the results for the datasets and metrics of the descriptor evaluation benchmark by Heinly et al. [127]. The results give insight into which image transformations are particularly challenging for the descriptors. We observe that

	SIFT	SIFT-PCA	DSP-SIFT	ConvOpt	DeepDesc	TFeat	LIFT
			Putative Match Ratio in %				
Blur	3.7	**5.7**	**7.0**	5.2	4.6	4.2	**6.5**
JPEG	20.9	**29.3**	**34.0**	26.8	24.4	22.9	**27.5**
Exposure	33.0	**34.1**	**35.3**	32.8	10.4	31.2	**34.9**
Day-Night	5.5	**6.8**	**6.2**	**7.2**	3.6	5.5	5.4
Scale	12.1	**25.2**	**23.4**	**23.8**	23.0	21.5	19.6
Rotation	**12.8**	**17.6**	**17.3**	10.0	11.9	8.7	1.3
Scale-rotation	2.4	**6.0**	**5.8**	**4.7**	4.5	3.7	2.0
Planar	5.9	**10.0**	**10.1**	**9.4**	7.7	8.0	8.0
Non-planar	7.8	**8.8**	**8.7**	**8.4**	7.4	7.2	8.3
Internet	3.2	**4.6**	**4.4**	4.3	2.7	3.4	**4.8**
			Precision in %				
Blur	43.8	**46.5**	**48.4**	45.2	41.9	**46.3**	44.5
JPEG	**98.5**	**96.5**	**98.3**	94.1	91.6	95.8	95.9
Exposure	**99.3**	**98.0**	**98.6**	96.6	68.0	97.3	97.5
Day-Night	**93.8**	**80.4**	**77.8**	73.9	37.8	76.5	71.2
Scale	43.0	**95.5**	**95.5**	92.2	89.1	**94.3**	89.1
Rotation	**33.2**	**33.1**	**33.1**	32.2	32.3	32.3	7.9
Scale-rotation	32.8	**46.7**	**46.8**	42.3	39.1	**43.9**	18.7
Planar	33.9	**37.3**	**39.9**	**34.3**	32.5	33.6	33.2
Non-planar	**43.3**	**42.2**	**43.1**	38.4	34.5	39.3	40.4
Internet	**39.8**	**40.3**	**39.7**	35.6	27.2	36.6	37.1
			Matching Score in %				
Blur	3.7	**5.5**	**6.8**	4.9	4.1	4.0	**6.2**
JPEG	20.8	**28.8**	**33.7**	26.1	23.5	22.6	**27.1**
Exposure	32.8	**33.5**	**34.9**	31.8	9.1	30.5	**34.2**
Day-Night	5.3	**5.9**	**5.5**	**5.8**	1.8	4.7	4.3
Scale	11.7	**24.4**	**22.8**	**22.6**	21.3	20.7	18.2
Rotation	**12.8**	**17.5**	**17.2**	9.7	11.6	8.5	0.9
Scale-rotation	2.4	**5.8**	**5.6**	**4.3**	3.9	3.5	1.6
Planar	5.7	**9.6**	**9.9**	**8.7**	6.9	7.5	7.4
Non-planar	7.7	**8.4**	**8.4**	**7.8**	6.5	6.9	7.7
Internet	3.1	**4.1**	**4.0**	3.5	1.8	2.8	**4.1**
			Recall in %.				
Blur	17.0	**22.4**	**27.2**	**20.0**	16.9	17.0	17.9
JPEG	37.9	**51.6**	**62.8**	46.6	41.0	39.2	**51.5**
Exposure	**79.0**	**81.0**	**84.1**	76.5	18.2	73.1	64.0
Day-Night	25.6	**29.2**	**26.2**	**28.9**	8.4	22.9	19.3
Scale	22.4	**84.0**	73.9	**76.1**	71.9	68.9	**98.4**
Rotation	**20.8**	**28.5**	**28.1**	16.1	19.1	14.1	2.3
Scale-rotation	6.4	**16.4**	**15.2**	**12.0**	10.9	9.6	5.3
Planar	11.4	**18.0**	**18.6**	16.4	13.3	14.2	**17.9**

Table 3.2: Evaluation results for the descriptor benchmark by Heinly et al. [127]. **First**, **second**, **third** best results highlighted in bold.

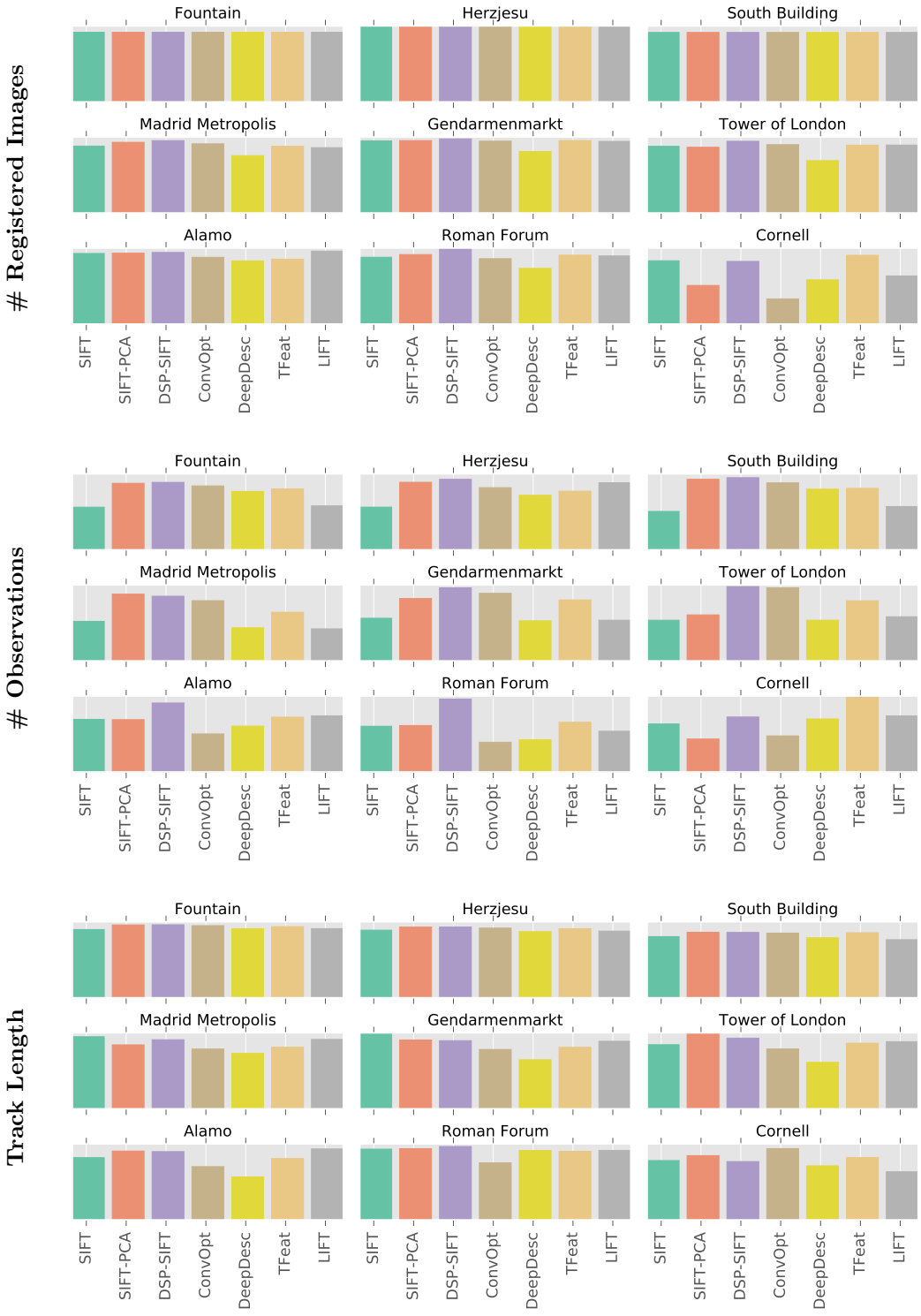

Figure 3.1: Qualitative visualization of sparse reconstruction statistics from Table 3.3. Visualization continued in Figure 3.2.

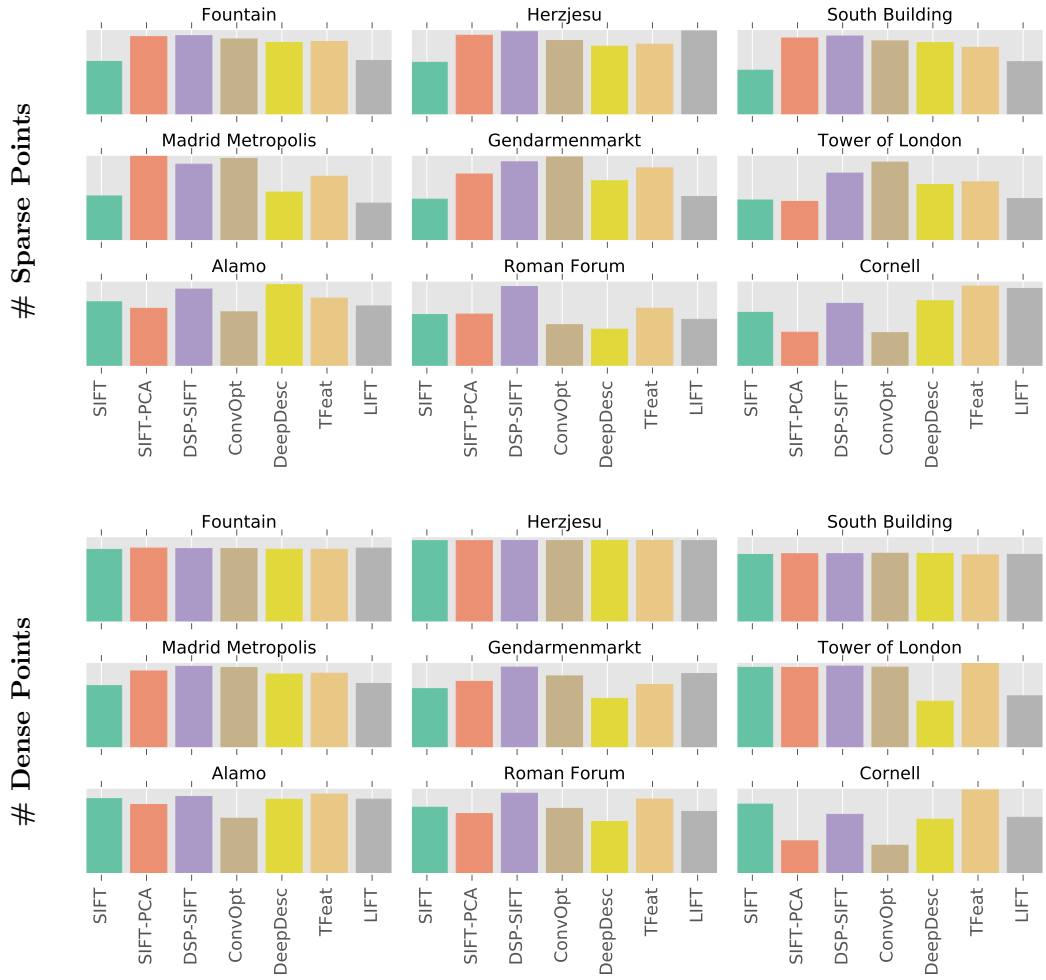

Figure 3.2: Qualitative visualization of dense reconstruction statistics from Table 3.3. Visualization continued in Figure 3.1.

all descriptors consistently perform worse across the different metrics in the case of image blur, day-night, and large viewpoint change. As expected, the learned descriptors typically outperform SIFT in terms of recall, while SIFT performs better in terms of precision. Surprisingly though, the advanced SIFT variants outperform the learned features for almost all metrics and matching scenarios. Notably, the performance of the learned descriptors often has a high variance across the different datasets, which indicates over-fitting for specific image transformations, e.g., due to a lack of training data depicting the entire appearance space of patches. Note that LIFT has problems with matching between rotated images, since it was trained on mostly upright Internet images. Among the learned descriptors, ConvOpt produces overall the best results and has the lowest variance across the different datasets. Table Table 3.3 presents the results for the image-based reconstruction benchmark and the *# Inlier Pairs* and *# Inlier Matches* metrics demonstrate a similar matching behavior in the large-scale setting. Next, we discuss, how the isolated matching performance impacts the image-based reconstruction results in practice.

Reconstruction

Table 3.3 lists the numerical values for the reconstruction evaluation, while Figure 3.1 and Figure 3.2 visualize the relative performance of the methods qualitatively. For the two smaller Strecha datasets (Fountain and Herzjesu), which were also evaluated by the authors of LIFT and TFeat, and the South Building dataset, the learned descriptors generally perform on par with or better than SIFT in terms of the number of sparse points, the number of image observations, and the mean track length. As a consequence of a better matching performance, the two advanced SIFT versions produce significantly better results than the other methods in these metrics. However, looking at the number of registered images, and the final dense modeling performance and accuracy metrics, all methods produce roughly the same reconstruction quality. We interpret these results as an indication that the Strecha and South Building datasets are rather easy benchmarks due to the structured camera setup with high overlap, same illumination conditions, etc. The higher variance in the results for the larger-scale Internet datasets confirms this interpretation. Here, Madrid Metropolis, Gendarmenmarkt, and Tower of London were matched exhaustively, whereas the images in Alamo, Roman Forum, and Cornell were only matched against the 100 nearest neighbors found using image retrieval. The matching and reconstruction results therefore also test the discriminative power of the descriptors in the context of BoW-based image retrieval. In the more challenging case of Internet photos, the matching performance directly impacts the ability to obtain complete and accurate models. Opposed to our observations in the raw matching evaluation, where SIFT produces inferior results as compared to the learned descriptors, in the reconstruction evaluation, SIFT performs typically on par with the learned descriptors. This implies that a better matching performance does not necessarily lead to better reconstruction results. DSP-SIFT performs best among all the methods, both in terms of sparse and dense reconstruction results. It consistently produces

		# Images	# Registered	# Sparse Points	# Observations	Track Length
Fountain	*SIFT*	11	11	10,004	44K	4.49
	SIFT-PCA		11	**14,608**	**70K**	**4.80**
	DSP-SIFT		11	**14,785**	**71K**	**4.80**
	ConvOpt		11	**14,179**	**67K**	**4.75**
	DeepDesc		11	13,519	61K	4.55
	TFeat		11	13,696	64K	4.68
	LIFT		11	10,172	46K	4.55
Herzjesu	*SIFT*	8	8	4,916	19K	4.00
	SIFT-PCA		8	**7,433**	**31K**	**4.19**
	DSP-SIFT		8	**7,760**	**32K**	**4.19**
	ConvOpt		8	6,939	28K	**4.13**
	DeepDesc		8	6,418	25K	3.92
	TFeat		8	6,606	27K	4.09
	LIFT		8	**7,834**	**30K**	3.95
South Building	*SIFT*	128	128	62,780	353K	5.64
	SIFT-PCA		128	**107,674**	**650K**	**6.04**
	DSP-SIFT		128	**110,394**	**664K**	**6.02**
	ConvOpt		128	**103,602**	**617K**	5.96
	DeepDesc		128	101,154	558K	5.53
	TFeat		128	94,589	566K	**5.99**
	LIFT		128	74,607	399K	5.35
Madrid Metropolis	*SIFT*	1,344	440	62,729	416K	**6.64**
	SIFT-PCA		**465**	**119,244**	**702K**	5.89
	DSP-SIFT		**476**	**107,028**	**681K**	**6.36**
	ConvOpt		**455**	**115,134**	**634K**	5.51
	DeepDesc		377	68,110	348K	5.11
	TFeat		439	90,274	512K	5.68
	LIFT		430	52,755	337K	**6.40**
Gendarmenmarkt	*SIFT*	1,463	950	169,900	1,010K	**5.95**
	SIFT-PCA		**953**	272,118	**1,477K**	**5.43**
	DSP-SIFT		**975**	**321,846**	**1,732K**	**5.38**
	ConvOpt		945	**341,591**	**1,601K**	4.69
	DeepDesc		809	244,925	949K	3.88
	TFeat		**953**	**297,266**	1,445K	4.86
	LIFT		942	180,746	964K	5.34
Tower of London	*SIFT*	1,576	702	142,746	963K	6.75
	SIFT-PCA		692	137,800	1,090K	**7.91**
	DSP-SIFT		**755**	**236,598**	**1,761K**	**7.44**
	ConvOpt		**719**	**274,987**	**1,732K**	6.30
	DeepDesc		551	196,990	964K	4.90
	TFeat		714	**206,142**	**1,424K**	6.91
	LIFT		**715**	147,851	1,045K	**7.07**
Alamo	*SIFT*	2,915	743	120,713	1,384K	11.47
	SIFT-PCA		**746**	108,553	1,377K	**12.69**
	DSP-SIFT		**754**	**144,341**	**1,815K**	**12.58**
	ConvOpt		703	102,044	1,001K	9.81
	DeepDesc		665	**152,537**	1,207K	7.92
	TFeat		683	**127,642**	**1,443K**	11.31
	LIFT		**768**	112,984	**1,477K**	**13.08**
Roman Forum	*SIFT*	2,364	1,407	242,192	1,805K	**7.45**
	SIFT-PCA		**1,463**	**244,556**	**1,834K**	**7.50**
	DSP-SIFT		**1,583**	**372,573**	**2,879K**	**7.73**
	ConvOpt		1,376	195,305	1,173K	6.01
	DeepDesc		1,173	174,532	1,275K	7.31
	TFeat		**1,450**	**271,902**	**1,963K**	7.22
	LIFT		1,434	220,026	1,608K	7.31
Cornell	*SIFT*	6,514	**4,999**	1,010,544	6,317K	6.25
	SIFT-PCA		3,049	640,553	4,335K	**6.77**
	DSP-SIFT		**4,946**	1,177,916	**7,233K**	6.14
	ConvOpt		1,986	632,613	4,747K	**7.50**
	DeepDesc		3,489	**1,225,780**	6,977K	5.69
	TFeat		**5,428**	**1,499,117**	**9,830K**	**6.56**
	LIFT		3,798	**1,455,732**	**7,377K**	5.07

Table 3.3: Results for our local feature benchmark. **First**, **second**, **third** best results highlighted in bold. Results continued in Table 3.4.

		Reproj. Error	# Inlier Pairs	# Inlier Matches	# Dense Points	Pose Error	Dense Error
Fountain	*SIFT*	**0.30px**	49	76K	2,970K	0.002m (0.002m)	0.77 (0.90)
	SIFT-PCA	0.39px	**55**	**124K**	**3,021K**	0.002m (0.002m)	0.77 (0.90)
	DSP-SIFT	0.41px	54	**129K**	2,999K	0.002m (0.002m)	0.77 (0.90)
	ConvOpt	0.37px	**55**	**114K**	**2,999K**	0.002m (0.002m)	0.77 (0.90)
	DeepDesc	**0.35px**	**55**	93K	2,972K	0.002m (0.002m)	0.77 (0.90)
	TFeat	**0.35px**	54	103K	2,969K	0.002m (0.002m)	0.77 (0.90)
	LIFT	0.59px	**55**	83K	**3,019K**	0.002m (0.002m)	0.77 (0.90)
Herzjesu	*SIFT*	**0.32px**	27	28K	2,373K	0.004m (0.004m)	0.57 (0.73)
	SIFT-PCA	0.42px	**28**	**47K**	2,372K	0.004m (0.004m)	0.57 (0.73)
	DSP-SIFT	0.45px	**28**	**50K**	**2,376K**	0.004m (0.004m)	0.57 (0.73)
	ConvOpt	0.40px	**28**	42K	2,375K	0.004m (0.004m)	0.57 (0.73)
	DeepDesc	**0.38px**	**28**	34K	**2,380K**	0.004m (0.004m)	0.57 (0.73)
	TFeat	**0.38px**	**28**	38K	**2,377K**	0.004m (0.004m)	0.57 (0.73)
	LIFT	0.63px	**28**	**46K**	2,375K	0.004m (0.004m)	0.57 (0.73)
South Building	*SIFT*	**0.42px**	1K	1,003K	1,972K	–	–
	SIFT-PCA	0.54px	**3K**	**2,019K**	1,993K	–	–
	DSP-SIFT	0.57px	3K	**2,079K**	**1,994K**	–	–
	ConvOpt	0.51px	**4K**	**1,856K**	**2,007K**	–	–
	DeepDesc	**0.48px**	**6K**	1,463K	**2,002K**	–	–
	TFeat	**0.49px**	3K	1,567K	1,960K	–	–
	LIFT	0.78px	3K	1,168K	1,975K	–	–
Madrid Metropolis	*SIFT*	**0.53px**	14K	1,740K	435K	–	–
	SIFT-PCA	0.57px	**27K**	**3,597K**	**537K**	–	–
	DSP-SIFT	0.64px	**21K**	**3,155K**	**570K**	–	–
	ConvOpt	0.57px	**29K**	**3,148K**	**561K**	–	–
	DeepDesc	**0.53px**	19K	1,570K	516K	–	–
	TFeat	**0.54px**	18K	2,135K	522K	–	–
	LIFT	0.76px	13K	1,498K	450K	–	–
Gendarmenmarkt	*SIFT*	**0.64px**	28K	3,292K	1,104K	–	–
	SIFT-PCA	0.69px	**43K**	**5,137K**	1,240K	–	–
	DSP-SIFT	0.74px	**56K**	**7,648K**	**1,505K**	–	–
	ConvOpt	0.70px	**56K**	**6,525K**	**1,342K**	–	–
	DeepDesc	**0.68px**	31K	2,849K	921K	–	–
	TFeat	**0.66px**	39K	4,685K	1,181K	–	–
	LIFT	0.83px	27K	2,495K	**1,386K**	–	–
Tower of London	*SIFT*	**0.53px**	18K	3,211K	1,126K	–	–
	SIFT-PCA	0.60px	12K	2,455K	1,124K	–	–
	DSP-SIFT	0.64px	**33K**	**8,056K**	**1,143K**	–	–
	ConvOpt	0.62px	**39K**	**7,542K**	**1,129K**	–	–
	DeepDesc	**0.55px**	25K	2,745K	653K	–	–
	TFeat	**0.57px**	**28K**	**5,333K**	**1,182K**	–	–
	LIFT	0.72px	23K	4,079K	729K	–	–
Alamo	*SIFT*	0.54px	**23K**	**7,671K**	**611K**	–	–
	SIFT-PCA	0.55px	12K	4,669K	564K	–	–
	DSP-SIFT	0.66px	**16K**	**10,115K**	**629K**	–	–
	ConvOpt	**0.48px**	3K	850K	452K	–	–
	DeepDesc	**0.48px**	16K	4,196K	607K	–	–
	TFeat	**0.52px**	16K	6,356K	**648K**	–	–
	LIFT	0.73px	**23K**	**9,117K**	607K	–	–
Roman Forum	*SIFT*	0.61px	**25K**	**6,063K**	**3,097K**	–	–
	SIFT-PCA	0.61px	16K	4,322K	2,799K	–	–
	DSP-SIFT	0.71px	**26K**	**9,685K**	**3,748K**	–	–
	ConvOpt	**0.55px**	11K	2,111K	3,043K	–	–
	DeepDesc	**0.60px**	9K	1,834K	2,434K	–	–
	TFeat	**0.61px**	**19K**	**5,584K**	**3,477K**	–	–
	LIFT	0.75px	17K	4,732K	2,898K	–	–
Cornell	*SIFT*	**0.53px**	71K	25,603K	**12,970K**	**1.537m (0.793m)**	–
	SIFT-PCA	**0.54px**	26K	13,793K	6,135K	11.498m (1.088m)	–
	DSP-SIFT	0.67px	73K	26,150K	**11,066K**	**2.943m (1.001m)**	–
	ConvOpt	0.57px	42K	18,615K	5,321K	5.824m (0.904m)	–
	DeepDesc	**0.55px**	**73K**	**28,845K**	10,159K	3.832m (0.695m)	–
	TFeat	0.59px	**89K**	**40,640K**	**15,605K**	**2.126m (0.593m)**	–
	LIFT	0.71px	**81K**	**39,812K**	10,512K	3.113m (0.712m)	–

Table 3.4: Results for our local feature reconstruction benchmark. Pose error as mean (median) over all images. Dense error for 2cm (10cm) threshold [144]. **First**, **second**, **third** best results highlighted in bold. Results continued in Table 3.3.

the most complete sparse reconstruction in terms of the number of registered images and reconstructed sparse points, while the dense models have the most points as a result of accurate camera registration. The mean reprojection error is similarly good for the descriptors that use the DoG keypoint detector, with a slightly larger error for DSP-SIFT, which is potentially caused by the descriptor pooling across multiple scales leading to more robustness with respect to inaccurate keypoint localization. Surprisingly, LIFT produces the largest reprojection error and relatively short tracks for all datasets, indicating inferior keypoint localization performance as compared to the hand-crafted DoG method. In addition, even though it was trained on the Roman Forum model, it does not perform better than DSP-SIFT or TFeat.

Figure 3.3 and Figure 3.4 show the sparse and dense reconstructions for the Gendarmenmarkt dataset. We observe that more challenging datasets result in significantly different reconstructions for the different local feature approaches. While the differences are not always prominent in the sparse reconstructions, the dense models often differ significantly. This evidences that dense reconstructions are a necessary indicator of the quality of a reconstruction. DSP-SIFT produces the most complete results. Notice the horizontal facades in the lower part of the screenshots, which are captured by comparatively few images. The reconstruction by TFeat contains some gross outlier structures due to incorrectly registered cameras. The results demonstrate that more challenging datasets are needed to meaningfully evaluate the performance of local features in practice.

3.3 Summary

This chapter presented a thorough experimental evaluation of learned and advanced hand-crafted local image feature descriptors to better understand their performance across a wide range of scenarios. The evaluation demonstrated that advanced hand-crafted local features still perform on par or better than recent learned local features in the context of image-based 3D reconstruction. The current generation of learned descriptors shows a high variance across different datasets and applications, which clearly evidences the necessity to evaluate a descriptor's discriminative power over a wide range of datasets and scenarios. To overcome the demonstrated limitations, we believe that the next generation of learned descriptors needs more training data and different training objectives. To facilitate future research, we made our full evaluation pipeline and a large training dataset of patches publicly available.

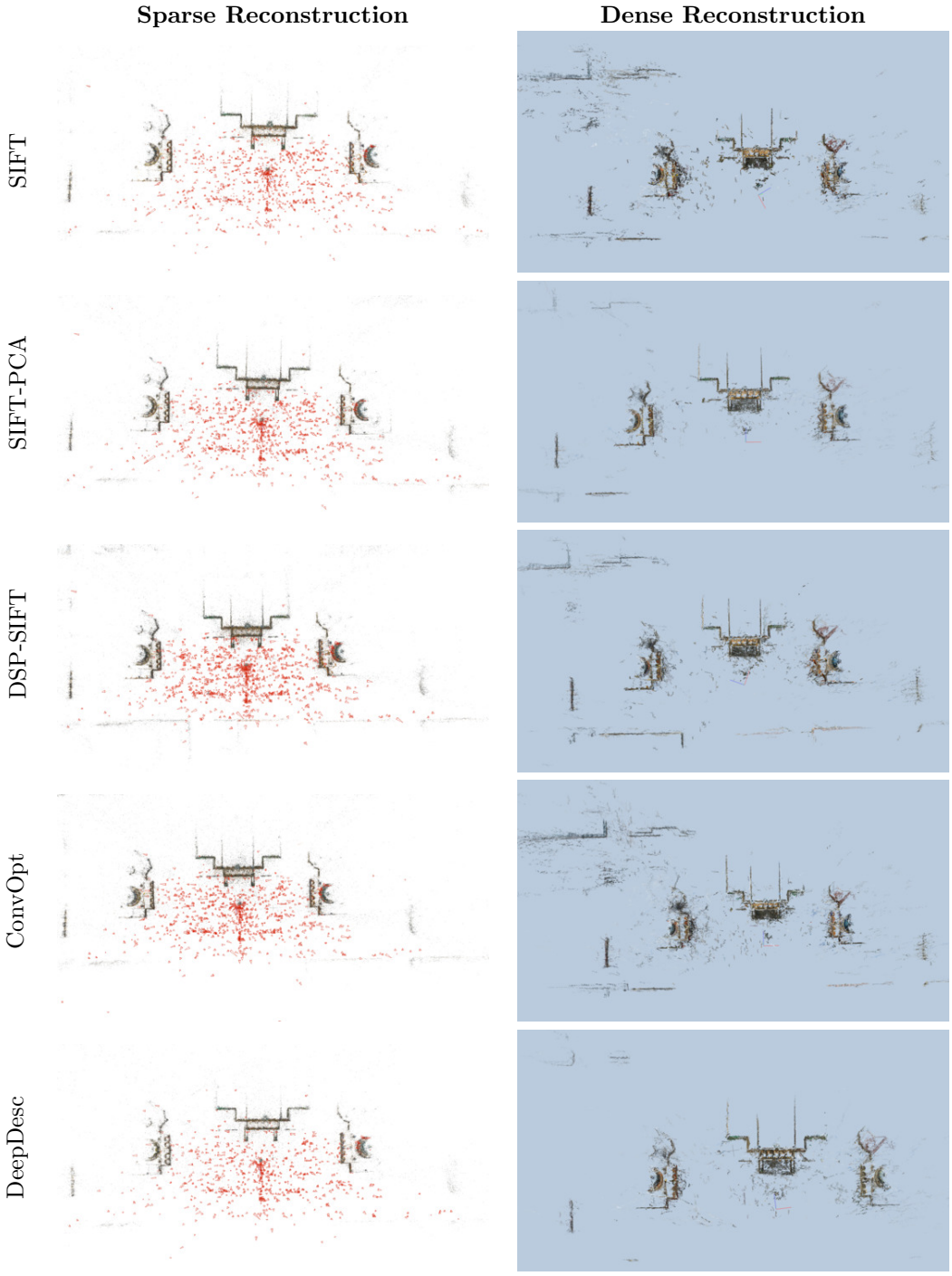

Figure 3.3: Sparse and dense reconstruction results of Gendarmenmarkt. SIFT with 950 registered images, SIFT-PCA with 953 registered images, DSP-SIFT with 975 registered images, ConvOpt with 945 registered images, and DeepDesc with 809 registered images. Results continued in Figure 3.4. Best viewed digitally for comparing details.

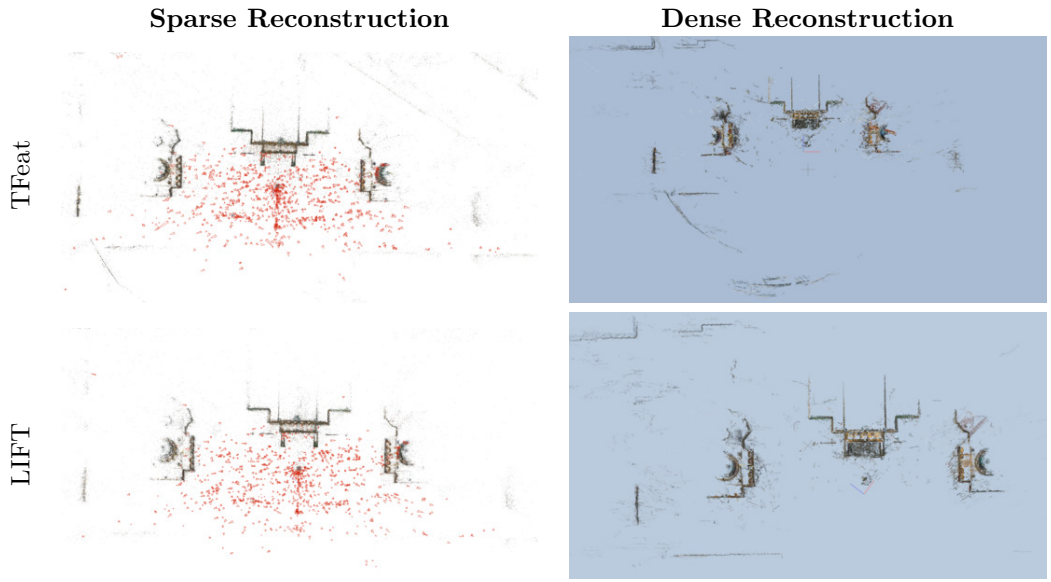

Figure 3.4: Sparse and dense reconstruction results of Gendarmenmarkt. TFeat with 953 registered images and LIFT with 942 registered images. Results continued in Figure 3.3. Best viewed digitally for comparing details.

4 A Vote-and-Verify Strategy for Fast Geometric Verification

The runtime of correspondence search in large-scale 3D modeling from unstructured image collections is usually dominated by feature matching and geometric verification due to the quadratic complexity in the number of images and features per image (see Section 2.4). A commonly used approach with better computational complexity in the feature matching component is to use image retrieval techniques to only match each image against its visually most similar looking images and to efficiently find matches between local features of similar looking images. While this drastically speeds up the feature matching process, it often results in lower quality feature matches and thus leads to a high runtime of geometric verification due to a higher outlier ratio (see Section 2.3.3). This section overcomes this limitation by presenting an algorithm for efficient and accurate geometric verification in image retrieval.

Image retrieval, i.e., finding relevant database images for a given query picture, is a fundamental problem in computer vision and has many applications beyond image-based 3D modeling [252, 284, 288], such as object retrieval [15, 307], location recognition [14, 270, 340], image-based localization [265, 274], automatic photo annotation [101], view clustering [357, 358], and loop-closure [181]. Although methods that use compact image representations [151, 152, 241] have gained popularity, especially in combination with deep learning [12, 111, 253], state-of-the-art systems [15, 64, 149, 150, 217, 335] still follow the *Bag-of-Words* (BOW) paradigm [307] proposed over a decade ago. The BOW model represents each image as a set of *visual words* obtained by quantizing the local feature space. Visually similar database images can then be found by searching for photos with similar visual words, usually implemented efficiently using inverted files [307]. For the sake of computational efficiency, BOW models generally only consider the presence and absence of visual words in an image and largely ignore their spatial configuration. Thus, a subsequent *spatial verification* phase [243] is typically used to filter retrieved photos whose visual words are not spatially consistent with the words in the query image. If not implemented efficiently, this spatial verification step quickly becomes the bottleneck of an image retrieval pipeline.

Spatial verification computes a geometric transformation, e.g., a similarity or affine transformation [243], from feature correspondences between the query and database images. The correspondences are obtained from common visual word assignments of the query and database features. This often leads to many wrong correspondences, especially in the presence of repetitive structures or when using small

vocabularies to reduce quantization artifacts. This makes traditional RANSAC-based approaches [85] that estimate transformations from multiple matches infeasible, as their runtime grows exponentially with the percentage of outliers. A key insight for fast spatial verification is to leverage the local feature geometry to hypothesize a geometric transformation from a single correspondence [200, 243]. This significantly reduces the number of hypotheses that need to be verified. Spatial verification can be accelerated by replacing the hypothesize-and-verify framework with a Hough voting scheme based on quantizing the space of transformations [19, 200]. These methods approximate the similarity between two images by the number of matches falling into the same bin in the voting space. Quantization artifacts are typically handled by using hierarchical voting schemes [19] or by allowing each match to vote for multiple bins [368].

Recent work demonstrates that a better verification accuracy can be obtained by explicitly incorporating verification into voting schemes, e.g., as a more detailed verification step [190] or when casting multiple votes [368]. However, in this section, we show that even such advanced voting schemes still achieve lower accuracy than classic hypothesize-and-verify approaches [243]. To close this gap, we propose a novel spatial verification approach that incorporates voting into a hypothesize-and-verify framework. In detail, we propose a hierarchical voting approach to efficiently identify promising transformation hypotheses that are subsequently verified and refined on all matches. Instead of finding the correct hypothesis through random sampling, we use a progressive sampling strategy on the most probable hypotheses. Furthermore, our approach offers multiple advantages over voting-based methods: First, rather than explicitly handling quantization artifacts in the voting space, our approach only requires that a reasonable estimate for the true transformation can be obtained from some matches falling into the same bin. Quantization artifacts are then automatically handled by the subsequent verification and refinement stages. As a result, our approach is rather insensitive to the quantization of the voting space and the visual vocabulary size. Second, in contrast to voting-based methods, which usually return only a similarity score, our approach explicitly returns a transformation and a set of inliers. Hence, it can be readily combined with query expansion (QE) schemes [15, 64, 66, 220] as well as further reasoning based on the detected inliers [146, 270]. Experimental evaluation on existing and new datasets show that our approach achieves accuracy equivalent to state-of-the-art hypothesize-and-verify methods while providing runtimes faster than state-of-the-art voting-based methods. The new query and distractor image datasets and the source code for our method are released to the public to facilitate further research.[1]

4.1 Related Work

In the following, we discuss prior work on spatial verification in the context of image retrieval. We classify these works based on whether they employ *weak geometric*

[1] `https://github.com/vote-and-verify`

models, use RANSAC's [85] *hypothesize-and-verify framework*, or follow a *Hough voting*-based approach. In this context, our method can be seen as a hybrid between the latter two types of approaches, as it replaces RANSAC's hypothesis generation stage with a voting scheme.

4.1.1 Weak Geometric Models

Instead of explicitly estimating a geometric transformation between two images, methods based on weak geometric models either use partial transformations [149, 190] or local consistency checks [267, 306, 369]. For a feature match between two images, both Sivic & Zisserman [306] and Sattler et al. [267] define a consistency score based on the number of matches shared in the spatial neighborhoods of the two features. A threshold on this score is then used to prune matches. The geometry of the local features, i.e., their position, orientation, and scale, can be used to hypothesize a similarity transformation from a single correspondence [200, 243]. Jegou et al. [149] focus on the change in scale and orientation predicted by a correspondence. They quantize the space of changes into a fixed set of bins and use Hough voting to determine a subset of matches that are all similar in terms of either scale or orientation change. Pairwise Geometric Matching (PGM) of Li et al. [190] uses a two-stage procedure to handle noise in the voting process caused by inaccurate feature frames. First, voting provides a rough estimate for the orientation and scale change between the images, as well as putative matches. PGM then checks whether the transformations obtained from correspondence pairs are consistent with the initial estimate. To improve the runtime, Li et al. perform a pruning step that enforces 1-to-1 correspondences. PGM's computational complexity is $\mathcal{O}(n + m^2)$, where n is the total number of matches and m is the number of matches falling into the best bin. $O(n)$ is required for voting for the best bin and $O(m^2)$ for pairwise verification. If all matches are correct, $m = n$ holds, and the complexity is $O(n^2)$. This worst case actually happens in practice, e.g., when the query is a crop of a database image. In comparison, while we also apply pruning, our vote-and-verify strategy has $\mathcal{O}(n)$ complexity and achieves both faster runtimes and better verification accuracy. A drawback to methods based on weak geometric models is they only determine a similarity score and do not identify individual feature correspondences. Thus, query expansion (QE) [64, 66], which transforms features from the database into the query image, cannot be directly applied.

4.1.2 Hypothesize-and-Verify Methods

Probably the most popular approach to compute a geometric transformation in the presence of outliers is RANSAC [85] or one of its many variants [63, 68, 180, 254, 267]. However, matching features through their visual word assignments usually generates many wrong correspondences. This makes it impractical to use any RANSAC variant that samples multiple matches in each iteration, as the runtime grows exponentially with the outlier ratio. Philbin et al. [243] therefore propose a

more efficient spatial verification approach, termed Fast Spatial Matching (FSM), exploiting the fact that a single correspondence already defines a transformation hypothesis. Consequently, they generate and evaluate all possible n transformations and apply local optimization [63, 180] whenever a new best model is found. FSM is the *de facto* standard for spatial verification and is used in most state-of-the-art retrieval pipelines [15, 64, 149, 150, 217, 335]. Its main drawback is the $\mathcal{O}(n^2)$ computational complexity due to evaluating all n hypotheses on all n correspondences. In practice, FSM can be accelerated by integrating it into a RANSAC framework and using early termination once the probability of finding a better model falls below a given threshold. Still, the worst-case complexity remains $\mathcal{O}(n^2)$. Our approach uses voting to efficiently identify promising transformation hypotheses in time $\mathcal{O}(n)$, and we show that it is sufficient to progressively sample a constant number of these hypotheses, resulting in an overall complexity of $\mathcal{O}(n)$. As such, our approach can be seen as borrowing the hypothesis prioritization of PROSAC [68] and using voting to achieve linear complexity. PROSAC orders matches based on matching quality and initially only generates transformation hypotheses from matches more likely to be correct. However, PROSAC is not directly applicable in our scenario, since matching via visual words does not provide an estimate of the matching quality. The output of FSM is a set of inliers and a geometric transformation, which can be used as input to QE. Our approach provides the same output and can thus directly be combined with existing QE methods.

4.1.3 Hough Voting-Based Approaches

Lowe [200] uses Hough voting to identify a subset of all putative matches, consisting of all matches whose corresponding similarity transformation belongs to the largest bin in the voting space. Next, they apply RANSAC on this consistent subset of matches to estimate an affine transformation. To reduce the complexity to $\mathcal{O}(n)$, Avrithis & Tolias [19] propose to restrict spatial verification to the voting stage. To mitigate quantization artifacts, their Hough Pyramid Matching (HPM) uses a hierarchical voting space, where every match votes for a single transformation on each level. For each match, they then compute a strength score based on the number of other correspondences falling in the the same bins and aggregate these strengths into an overall similarity score for the two images. Wu & Kashino propose to handle quantization artifacts by having each match vote for multiple bins [368]. For each feature match m, their Adaptive Dither Voting (ADV) scheme finds neighboring correspondences, similar to [267, 306]. If m is consistent with the transformation hypothesis of its neighbor, a vote is cast in the neighbor's bin. This additional verification step helps ADV to avoid casting unrelated votes, resulting in better accuracy compared to HPM. Since ADV finds a fixed number of nearest neighbors in the image space, its computational complexity is $\mathcal{O}(n \log n)$. Similar to methods based on weak geometric models, both HPM and ADV only provide an overall similarity score and thus cannot be directly combined with standard QE. In addition, our approach achieves superior verification accuracy at faster runtimes than ADV.

4.1.4 Verification during Retrieval

All previously discussed methods operate as a post-processing step after image retrieval. Ideally, spatial verification should be performed during the retrieval process to detect and reject incorrect votes. While there exist a variety of approaches that directly integrate geometric information [65, 156, 297, 336, 393], this usually comes at the price of additional memory requirements or longer retrieval times. Thus, most state-of-the-art approaches still apply spatial verification only as a post-processing step [15, 64, 217, 335].

4.2 Algorithm

One inherent problem of hypothesize-and-verify methods is that finding a good hypothesis through either random sampling or exhaustive search often takes quite some time. In this section, we propose to solve this problem by finding promising transformation hypotheses through voting. Starting with the most promising ones, we verify a fixed number of these hypotheses on all matches, i.e., perform inlier counting. As in FSM [243], local optimization [63, 180] is applied every time a new best transformation is found.

At first glance, the voting stage of our approach may seem identical to existing voting-based approaches [19, 190, 368], but there are two important differences between previous methods and our approach: i) Existing works [19, 190, 368] use voting with the aim of identifying *all* geometrically consistent matches. As such, handling quantization artifacts in the voting space is very important and the methods are rather sensitive to the number of bins in the voting space. In contrast, we only require to find transformations that need to be geometrically consistent with *some* of the matches. Matches missed due to quantization are then automatically detected during inlier counting and are subsequently used during local optimization to refine the model. ii) Instead of only using the number of matches associated with the same bin, we are also interested in the transformation induced by these matches. To this end, we find that simply taking the transformation defined by the center of a bin does not provide an accurate enough hypothesis, even for high levels of quantization. Consequently, we propose a refinement process to obtain more accurate transformation hypotheses. As a result, our approach is rather insensitive to the quantization level.

In the following, we first recapitulate the process of computing a similarity transformation from a single feature match (see Section 4.2.1). We next detail the voting procedure (see Section 4.2.2) and explain how to derive transformation hypotheses from the voting space (see Section 4.2.3). Section 4.2.4 then describes the verification and local optimization stage, while Section 4.2.5 analyses the computational complexity of our method. Finally, Algorithm 1 gives an overview of our proposed method.

Algorithm 1 Proposed Vote-and-Verify algorithm: VERIFY(VOTE($m_{i=1...n}$)).

procedure VOTE($m_{i=1...n}$)
  for $i = 1...n$ **do**
    for $l = 1...L$ **do**
      Vote for $\boldsymbol{C}(m_i, l)$ with $w(l)$ in Hough space
  Find bins $b_{t=1...T}$ with highest votes $w(b_t)$
  return Hypotheses $\boldsymbol{T}(b_{t=1...T})$ ordered in decreasing number of votes
procedure VERIFY($\boldsymbol{T}(b_{t=1...T})$)
  Set initial score $\hat{s} = 0$ and transformation $\hat{\boldsymbol{T}}$ as invalid
  for $t = 1...T$ **do**
    Verify $\boldsymbol{T}(b_t)$ and count the number of inliers s_t
    if $s_t > \hat{s}$ **then**
      Refine $\boldsymbol{T}(b_t)$ with local optimization and update s_t
      Update best score $\hat{s} = s_t$ and transformation $\hat{\boldsymbol{T}} = \boldsymbol{T}(b_t)$
      Update probability p of finding better $\hat{\boldsymbol{T}}$
      if $p < \hat{p}$ **then**
        break
  return Effective inlier count $\hat{s}_{\text{eff}}$ for $\hat{\boldsymbol{T}}$

4.2.1 From Local Features to Similarity Transformation

Consider a local image feature f, *e.g.*, a SIFT feature [200], defined by its local feature frame $f = (f_x, f_y, f_\sigma, f_\theta)$. Here, $(f_x, f_y)^T$ and f_σ are the location and scale of the detected feature in the image, while f_θ denotes the feature orientation. Following [19], each feature f is associated with a canonical coordinate frame in which the feature is located at the origin with unit scale and zero orientation. The transformation $\boldsymbol{M}_f(\boldsymbol{x})$ maps a location $\boldsymbol{x} = (x, y)^T$ in the image to a position in the canonical frame as

$$\boldsymbol{M}_f(\boldsymbol{x}) = \frac{1}{f_\sigma}\boldsymbol{R}(f_\theta)\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}f_x\\f_y\end{bmatrix}\right) = \frac{1}{f_\sigma}\begin{bmatrix}\cos f_\theta & \sin f_\theta\\ -\sin f_\theta & \cos f_\theta\end{bmatrix}\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}f_x\\f_y\end{bmatrix}\right) \quad , \tag{4.1}$$

where the rotation matrix $\boldsymbol{R}(f_\theta)$ performs a clockwise rotation by f_θ degrees. Consequently, a feature match $m = (f^{\mathcal{Q}}, f^{\mathcal{D}})$ between a query image $\mathcal{Q}$ and a database image $\mathcal{D}$ defines a similarity transformation between the two images:

$$\begin{aligned}
\boldsymbol{M}_{(f^{\mathcal{Q}}, f^{\mathcal{D}})}(\boldsymbol{x}) &= \boldsymbol{M}_{f^{\mathcal{Q}}}^{-1}\left(\boldsymbol{M}_{f^{\mathcal{D}}}(\boldsymbol{x})\right) && (4.2)\\
&= \frac{f_\sigma^{\mathcal{Q}}}{f_\sigma^{\mathcal{D}}}\boldsymbol{R}(f_\theta^{\mathcal{Q}})^T\boldsymbol{R}(f_\theta^{\mathcal{D}})\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}f_x^{\mathcal{D}}\\f_y^{\mathcal{D}}\end{bmatrix}\right) + \begin{bmatrix}f_x^{\mathcal{Q}}\\f_y^{\mathcal{Q}}\end{bmatrix} && (4.3)\\
&= \sigma\boldsymbol{R}(\theta)\boldsymbol{x} + \boldsymbol{t} \ . && (4.4)
\end{aligned}$$

In Equation 4.4, $\sigma = f_\sigma^{\mathcal{Q}}/f_\sigma^{\mathcal{D}}$, $\theta = f_\theta^{\mathcal{Q}} - f_\theta^{\mathcal{D}}$, and $\boldsymbol{t} = (t_x, t_y)^T$ define the relative scale, rotation angle, and translation of the similarity transformation. Thus, each

match $m = (f^{\mathcal{Q}}, f^{\mathcal{D}})$ can be associated with a 4-dimensional coordinate

$$\boldsymbol{C}(m) = [\sigma, \theta, t_x, t_y] \ . \tag{4.5}$$

4.2.2 From Similarity Transformation to Hough Voting

Each of the n feature matches between a query and database image defines a transformation hypothesis. We are interested in determining a fixed-sized set of transformations that is consistent with as many of these n hypotheses as possible. This can be done very efficiently using Hough voting, as similar transformations are likely to fall into the same voting bin. This allows us to obtain a set of promising transformation hypotheses from the best scoring bins. However, standard Hough voting suffers from quantization artifacts, caused by inaccuracies in the detected feature frames. This is especially problematic when only few matches are correct, in which case it becomes harder to distinguish between bins corresponding to the underlying geometric transformation and bins receiving votes from wrong matches. Thus, we use a hierarchical voting scheme similar to [19], which we describe in the following.

Following Avrithis & Tolias [19], we quantize each of the four similarity transformation parameters independently. The Hough voting space of transformations is then defined as the product space of the four individual quantizations. We discretize the space at different resolution levels $l = 0...L$ and use n_x, n_y, n_σ, and n_θ bins for translation, scale, and orientation at the finest resolution $l = 0$. Each successive resolution level divides the number of bins in half until only two bins are left for each dimension. Out of the four dimensions, only the rotation space is naturally bounded by $[0, 2\pi)$, while translation and scale in theory can take any value from $\mathbb{R}^2$ and $\mathbb{R}^+$, respectively. In practice, the space of possible scale changes σ is bounded, since feature detectors only consider a few octaves of scale space [19]. A feature match inducing a large scale change between two images can usually be safely discarded as an incorrect correspondence. Consequently, we only consider scale changes in the range $[1/\sigma_{\max}, \sigma_{\max}]$ [19]. In addition, we bound the translation parameters by

$$\max(|t_x|, |t_y|) \leq \max(W, H) \ , \tag{4.6}$$

where W and H are the width and height of the query image. Matches violating at least one constraint are ignored [19].

Given a feature match $m = (f^{\mathcal{Q}}, f^{\mathcal{D}})$ with transformation parameters σ, θ, and $\boldsymbol{t}$ as defined above, we obtain the corresponding Hough space coordinate as

$$\begin{aligned}
C_x(t_x, l) &= 2^{-l} \lfloor n_x \left(t_x + \max(W, H)\right) / \left(2 \max(W, H)\right) \rfloor \ , && (4.7)\\
C_y(t_y, l) &= 2^{-l} \lfloor n_y \left(t_y + \max(W, H)\right) / \left(2 \max(W, H)\right) \rfloor \ , && (4.8)\\
C_\sigma(\sigma, l) &= 2^{-l} \lfloor n_\sigma \left(\log_2(\sigma) + \log_2(\sigma_{\max})\right) / \left(2 \log_2(\sigma_{\max})\right) \rfloor \ , && (4.9)\\
C_\theta(\theta, l) &= 2^{-l} \lfloor n_\theta \left(\theta + \pi\right) / \left(2\pi\right) \rfloor \ . && (4.10)
\end{aligned}$$

For uniform sampling in the scale space, we linearize the scale change using the logarithmic function. The factor 2^{-l} normalizes the Hough coordinates to the respective

resolution level of the voting space. Here, each match $m = (f^{\mathcal{Q}}, f^{\mathcal{D}})$ determines a coordinate $\boldsymbol{C}(m, l)$ at level l in the voting space, with

$$\boldsymbol{C}(m, l) = [C_x(t_x, l), C_y(t_y, l), C_\sigma(\sigma, l), C_\theta(\theta, l)] \ . \tag{4.11}$$

The match m then contributes a level-dependent weight $w(l) = 2^{-l}$ to the score of its corresponding voting bin at level l. Next, we describe how to use these scores to detect the T most promising transformation hypotheses.

4.2.3 Hypothesis Generation

The goal of our voting scheme is to provide a set of transformation hypotheses for subsequent verification. As described in the previous section, the center of any bin in the hierarchical representation defines a transformation, and we could simply pick the transformations corresponding to the bins with the highest scores. For coarser levels in the hierarchy, however, it is unlikely that the center of the bin is close to the actual transformation between the images. As such, we only hypothesize transformations corresponding to bins at level $l = 0$. To mitigate quantization artifacts, we propagate the scores from coarser levels to the bins at level 0. Each bin b at level 0 uniquely defines a path through the hierarchy to the coarsest level L. The total score for this bin is computed by summing the scores of all bins along this path as $w(b) = \sum_{l=0\ldots L} w(b, l)$. Finally, we simply select the T bins that received the highest scores $w(b)$.

As we will show in Section 4.3, the naive approach of associating each bin at level 0 with the transformation defined by the bin's center coordinate does not perform well, even when using a reasonably deep hierarchy. In other words, the center coordinate of a bin can be rather far away from the true image transformation. In order to obtain a better estimate, we use the mean transformation of all matches falling into the bin instead. Let $\mathcal{M}(b)$ be the matches falling into bin b. Following Equation 4.5, the mean transformation is defined as

$$\boldsymbol{T}(b) = \frac{1}{|\mathcal{M}(b)|} \sum_{m \in \mathcal{M}(b)} \boldsymbol{C}(m) \tag{4.12}$$

and can be computed efficiently during voting without a significant memory overhead by maintaining a running average. Intuitively, one can think of this as local optimization (*cf.* [63]) on the level of hypothesis generation.

It is well-known that outliers, i.e., wrong matches falling into a bin, significantly impact the computation of the mean. As a more robust alternative, one could use the median transformation instead. However, the mean can be computed much more efficiently, and experiments in Section 4.3.3 show that its performance is very robust to the choice of the quantization resolution.

4.2.4 Accurate and Efficient Hypothesis Verification

The scores associated with each of the T similarity transformation hypotheses only provides an estimate on how well the transformation explains the matches. As a

next step, we thus perform detailed hypothesis verification using inlier counting. For this stage, we follow FSM [243] and consider a match an inlier to a transformation if its two-way reprojection error is below a threshold and if the scale change between two corresponding features induced by the transformation is consistent with the scales of the two features. The $t = 1...T$ transformation hypotheses are verified in decreasing order of their scores, and we apply local optimization [63, 180] every time a new best model is found. The latter step refines the transformation by drawing a constant number of non-minimal sets of inliers to obtain a least squares estimate for the transformation. If there are at least $s = 3$ inliers, we estimate an affine transformation in local optimization. Affine transformations have been shown to perform better than similarity transformations [243], as they can handle more general geometric configurations.

We progressively verify the hypotheses ordered in decreasing number of votes until the probability $p = (1-e)^t$ of finding a better model at the current inlier ratio e falls below a threshold. Our method is a variant of PROSAC [68] using Hough voting for ordering and 1-point sampling with a fixed-size hypothesis set.

One advantage over voting-based methods is that we explicitly determine the set of inliers. This allows us to use image similarity functions that are more discriminative than simply counting the number of inliers [270]. One such function is the *effective inlier count* [146], which has been shown to outperform raw inlier counting for image-based localization [265] and image retrieval [270] tasks. The effective inlier count is defined as

$$s_{\text{eff}} = \frac{|\cup_{i=1}^{\hat{n}} A_i|}{\sum_{i=1}^{\hat{n}} |A_i|} \hat{n} \ . \tag{4.13}$$

Here, $\hat{n}$ is the number of inlier matches, A_i is a region centered around the i-th inlier feature in the query image, $|\cup_{i=1}^{\hat{n}} A_i|$ is the area of the union of all regions, and $\sum_{i=1}^{\hat{n}} |A_i|$ is the maximum area that could be covered if none of the regions would overlap. In our experiments, we use square regions of size 24×24px.

4.2.5 Computational Complexity

For a fixed number of levels in the hierarchy, each match votes for a fixed number of bins, where each transformation bin can be computed in time $\mathcal{O}(1)$. Performing a single vote is also a constant time operation, because it only requires incrementing a counter and updating the running mean for the transformation. Consequently, voting can be done in time $\mathcal{O}(n)$ for n matches. Since T is a constant, finding and evaluating the T best hypotheses also requires time $\mathcal{O}(n)$. Each hypothesis is evaluated on all matches, which can again be done in time $\mathcal{O}(n)$. Local optimization is performed at most once for each hypothesis using a fixed-sized non-minimal set, i.e., each least squares transformation can be computed in constant time as well. Thus, the overall computational complexity is linear in the number n of matches.

4.3 Experimental Evaluation

In this section, we first introduce the datasets and describe our experimental setup. Next, we perform an ablation study and analyze the impact of the different parameters on the performance of our approach. Finally, we provide an extensive comparison with state-of-the-art spatial verification methods.

4.3.1 Query and Distractor Datasets

Following standard procedure [190, 368], we primarily evaluate on the Oxford5k [243] and Paris6k [242] datasets. These image sets, collected from Flickr, contain ∼5k and ∼6k images, respectively, each consisting of 11 distinct landmarks, with 5 query images per landmark. In addition, we created the new *World5k* dataset consisting of 5320 images from 61 landmarks around the world, where the images were obtained from the Yahoo 100M images dataset [332]. The landmark images were selected based on geo-tags and using overlap information from Heinly et al. [129] for ground-truth. Each landmark has between 30 and 100 database images and 1 to 3 associated query photos, resulting in a total of 163 query images. Different from Oxford5k and Paris6k, which represent an object retrieval task where query photos are obtained by cropping regions from database images, our query images are full-resolution and are not contained in the database.

To simulate larger datasets and thus create harder retrieval scenarios, it is common to combine the individual datasets with an additional "distractor" dataset of ∼100k unrelated images collected from Flickr [243]. It has been shown that adding this Flickr100k distractor set significantly impacts the retrieval performance. However, the Flickr100k (F100k) dataset mostly contains very unrelated photos, obtained by searching for generic terms, such as "graffiti", "uk", or "vacation". We thus collected four additional distractor sets from the Yahoo 100M images dataset. The first distractor set consists of 140k images taken between 2km and 50km from the center of the University of Oxford. The other three distractor sets consist of images taken from 30 cities around the UK (171k images), 30 cities throughout Europe (179k), and 30 cities across the US (233k). The collections were formed using the set of geo-tagged images within a 1km radius (2km for the US images) of the geographic center of the city. We expect to find more geometrically consistent matches for distractor images, e.g., due to buildings with similar architectural styles, on the new distractor datasets compared to the Flickr100k set. As such, we expect that our new distractor sets represent more challenging scenarios for spatial verification. In the following, we refer to the distractor sets as *Ox* (Oxford), *UK*, *EU* (Europe), and *US*.

4.3.2 Experimental Setup

Retrieval System

We employ a state-of-the-art retrieval system using Hamming embedding [149] and visual burstiness weighting [150]. We use vocabularies containing 20K, 200K, and

T		10						20						30					
$n_{x/y}$		16		32		64		16		32		64		16		32		64	
n_σ	n_θ	mAP	time	mAP	time	mAP	time	mAP	time	mAP	time	mAP	time	mAP	time	mAP	time	mAP	time
	8	**80.1**	0.7	79.7	0.7	79.8	0.8	79.9	0.8	79.9	0.9	79.9	0.9	80.0	0.9	79.9	0.9	80.0	1.0
8	16	79.9	0.7	79.7	0.8	79.8	0.8	80.0	0.8	79.8	0.9	79.9	0.9	80.0	0.8	79.9	0.9	80.0	1.0
	32	79.7	0.7	79.8	0.8	79.9	0.8	80.0	0.9	79.9	0.9	79.9	0.9	79.9	0.9	79.9	0.9	80.0	1.0
	8	79.9	0.7	79.7	0.7	79.8	0.8	80.0	0.8	79.9	0.9	79.9	0.9	**80.1**	0.8	80.0	0.9	**80.1**	0.9
16	16	79.9	0.8	79.8	0.7	79.8	0.8	80.0	0.9	79.9	0.9	79.9	0.9	80.0	1.0	80.0	0.9	80.0	0.9
	32	79.7	0.7	79.9	0.8	79.8	0.8	80.0	0.8	79.9	0.9	80.0	0.9	80.0	0.9	80.0	0.9	80.0	0.9
	8	79.9	0.7	80.0	0.8	79.8	0.8	80.0	0.8	80.0	0.9	79.9	0.9	80.0	0.9	80.0	0.9	**80.1**	0.9
32	16	79.7	1.0	79.9	0.7	79.7	0.8	80.0	0.9	80.0	0.9	79.9	0.9	80.0	0.8	80.0	0.9	**80.1**	0.9
	32	79.8	0.7	79.7	0.9	80.0	0.8	79.9	0.8	80.0	1.0	80.0	0.9	80.0	0.9	80.0	1.1	**80.1**	0.9

Table 4.1: The impact of the number of voting bins (n_σ, n_θ, n_x, n_y) and the number T of verified transformation hypotheses on the performance and efficiency of our method. Results, obtained on Oxford5k, with the highest mAP (80.1%) are highlighted in green.

1M words to quantize RootSIFT descriptors [15, 200] extracted from keypoints provided by an upright Hessian affine feature detector [239]. Following standard procedure [335], we ensure that the vocabulary used for each dataset has been trained on another image collection. Correspondingly, we use a vocabulary trained on Oxford5k for the experiments on Paris6k and our new dataset. A vocabulary trained on Paris6k is then used for all experiments performed on the Oxford5k dataset. After retrieval, the top-1000 ranked database images are considered for spatial verification and re-ranked based on the similarity scores computed during verification. We enforce 1-to-1 matches prior to verification [190], since initial experiments showed that this significantly improved verification efficiency and quality for all spatial verification methods.

Evaluation Protocol

We follow the standard evaluation procedure and assess the verification performance using mean average precision (mAP), which essentially averages the area under the precision-recall curves. For each verification approach, we report the total time in seconds required to verify the 1000 top-ranked retrievals for all query images. We ignore retrieval and setup time, e.g., the time required to enforce a 1-to-1 matching, since this is separate from verification. To facilitate comparability, we run single-threaded implementations on an Intel E5-2697 2.7GHz CPU with 256GB RAM.

4.3.3 Ablation Study

We evaluate the impact of the different parameters of our approach on its verification performance and efficiency. All experiments presented in this section have been performed on the Oxford5k dataset without any distractor images.

Impact of the Level of Quantization

As a first experiment, we evaluate the impact of the number of bins for rotation (n_θ), scale (n_σ) and translation (n_x, n_y) as well as the number T of transformation

	–	F100k	Ox	UK	EU	US	Ox+UK	Ox+EU	Ox+US	UK+EU	UK+US	EU+US	Ox+UK+EU	Ox+UK+US	Ox+EU+US	UK+EU+US	Ox+UK+EU+US	All
mAP [%]																		
Pure Retrieval	74.8	65.4	63.4	60.0	59.2	59.2	57.5	56.6	56.8	55.9	55.9	55.2	54.3	54.3	53.7	53.5	52.2	51.7
FSM Aff	**77.2**	71.1	68.8	66.4	67.0	67.6	64.8	65.3	65.7	64.5	64.7	65.3	63.3	63.4	64.0	63.7	62.6	62.2
+ Eff. Inl. Eval	**77.3**	**71.8**	**70.0**	**67.6**	**68.2**	**68.8**	**66.3**	**66.8**	**67.2**	**66.0**	**66.1**	**66.8**	**64.9**	**64.9**	**65.7**	**65.2**	**64.3**	**63.9**
+ Eff. Inl. Post	**77.2**	**71.5**	69.4	67.0	**67.6**	68.2	65.6	**66.1**	66.5	**65.3**	65.4	66.1	64.2	64.2	**65.0**	**64.5**	63.5	63.2
FSM-R Aff	**77.2**	71.1	68.8	66.4	67.0	67.6	64.8	65.3	65.7	64.5	64.7	65.3	63.3	63.4	64.0	63.7	62.6	62.2
+ Eff. Inl. Eval	**77.4**	**71.8**	**70.0**	**67.5**	**68.2**	**68.7**	**66.3**	**66.8**	**67.2**	**66.0**	**66.1**	**66.8**	**64.9**	**64.9**	**65.7**	**65.2**	**64.3**	**64.0**
+ Eff. Inl. Post	**77.2**	**71.6**	69.4	67.0	67.5	68.1	**65.7**	66.0	66.5	**65.3**	65.4	66.0	64.2	64.2	**65.0**	64.4	63.5	63.3
FSM Sim	76.6	70.6	68.3	65.9	66.5	66.9	64.2	64.6	65.1	64.0	64.2	64.8	62.7	62.8	63.4	63.2	62.1	61.8
+ Eff. Inl. Eval	76.9	71.2	**69.5**	66.9	**67.6**	68.2	65.6	**66.1**	**66.7**	**65.3**	**65.5**	**66.2**	**64.3**	**64.4**	**65.0**	**64.6**	**63.7**	**63.4**
+ Eff. Inl. Post	76.7	70.9	68.9	66.2	66.9	67.6	64.8	65.3	65.9	64.5	64.7	65.3	63.4	63.4	64.1	63.7	62.8	62.6
FSM-R Sim	76.6	70.5	68.2	65.8	66.4	66.9	64.1	64.6	65.1	63.9	64.1	64.7	62.7	62.8	63.4	63.2	62.1	61.8
+ Eff. Inl. Eval	76.9	71.2	**69.5**	66.9	67.5	68.2	65.6	**66.1**	66.6	**65.3**	**65.5**	66.1	64.2	**64.3**	**65.0**	**64.6**	**63.6**	63.3
+ Eff. Inl. Post	76.7	70.8	68.9	66.2	66.8	67.6	64.7	65.2	65.8	64.4	64.7	65.3	63.3	63.4	64.0	63.6	62.7	62.5
HPM	72.8	63.6	61.3	57.9	57.9	58.1	56.0	56.0	56.2	55.2	55.1	55.1	54.1	54.0	54.0	53.7	52.8	52.5
ADV	75.8	70.0	67.8	65.3	65.5	66.5	63.7	63.9	64.7	63.4	63.7	63.9	62.2	62.5	62.7	62.6	61.6	61.2
PGM	74.8	64.7	62.2	58.8	58.6	58.1	55.6	55.7	55.5	54.6	54.3	54.3	52.7	52.5	52.6	52.1	50.7	50.2
Ours	76.8	70.5	68.1	65.5	66.2	66.4	63.7	64.2	64.4	63.4	63.4	64.0	62.0	62.1	62.7	62.3	61.1	60.8
+ Eff. Inl. Post	**77.2**	**71.6**	**69.9**	**67.2**	**68.0**	**68.4**	**66.0**	**66.5**	**66.8**	**65.6**	**65.6**	**66.3**	**64.4**	**64.4**	**65.2**	**64.6**	**63.7**	63.3
Runtime [s]																		
FSM Aff	25.9	27.9	40.8	44.5	44.5	43.5	52.1	52.9	52.1	53.6	54.8	52.0	53.6	53.8	56.2	55.5	54.1	54.8
+ Eff. Inl. Eval	173.4	184.3	199.3	232.9	228.1	213.6	238.2	236.7	222.5	249.5	241.0	238.2	252.6	245.1	245.4	252.6	257.8	258.1
+ Eff. Inl. Post	23.3	27.5	41.2	45.4	45.2	43.3	52.0	52.9	52.3	54.6	53.5	53.8	54.6	54.8	56.5	54.5	55.3	56.0
FSM-R Aff	2.5	2.9	3.7	4.3	4.2	4.1	5.1	5.1	5.2	5.3	5.4	5.4	5.6	5.5	5.6	5.7	5.9	6.1
+ Eff. Inl. Eval	87.2	100.3	112.6	142.9	138.8	123.6	148.2	145.6	131.8	157.4	150.2	147.7	161.0	154.1	152.7	161.8	165.4	165.6
+ Eff. Inl. Post	2.6	3.0	3.9	4.6	4.4	4.3	5.5	5.6	5.4	5.7	5.6	5.6	6.0	6.0	6.0	6.1	6.3	6.6
FSM Sim	25.2	30.1	44.3	48.4	48.4	47.8	56.8	57.0	57.3	58.5	57.7	57.8	58.9	59.4	61.2	58.9	59.5	60.1
+ Eff. Inl. Eval	180.6	192.2	208.6	241.3	235.9	221.2	247.9	244.9	231.8	258.5	249.9	245.9	262.0	255.3	253.9	260.8	266.3	265.5
+ Eff. Inl. Post	25.3	30.2	44.6	48.8	48.5	48.2	57.3	58.0	57.8	59.6	58.8	58.8	59.6	60.9	61.3	61.1	60.7	61.6
FSM-R Sim	2.4	2.7	3.5	4.1	4.0	3.9	4.8	4.9	4.9	5.2	5.0	5.1	5.4	5.3	5.4	5.5	5.6	5.8
+ Eff. Inl. Eval	86.4	99.6	111.8	141.9	137.5	122.2	146.6	144.3	130.6	156.2	148.7	145.6	159.9	153.0	151.5	160.1	164.0	164.1
+ Eff. Inl. Post	2.5	2.8	3.7	4.4	4.2	4.1	5.2	5.2	5.3	5.5	5.3	5.5	5.8	5.7	5.8	5.9	6.3	6.3
HPM	**0.6**	**0.5**	**0.7**	**1.0**	**0.9**	**0.8**	**1.0**	**1.0**	**0.8**	**1.1**	**1.0**	**0.9**	**1.0**	**0.9**	**1.0**	**1.0**	**1.0**	**1.3**
ADV	**0.9**	1.1	**1.3**	**1.8**	**1.7**	**1.6**	2.0	2.0	1.8	2.3	2.0	2.1	2.4	2.3	2.3	2.4	2.5	2.5
PGM	14.8	11.4	11.6	12.4	12.3	12.5	12.1	12.1	11.9	11.8	12.0	11.8	12.0	12.1	12.3	12.0	11.8	11.8
Ours	**0.8**	**0.6**	**0.7**	**1.0**	**0.9**	**0.8**	**1.1**	**1.1**	**1.0**	**1.2**	**1.2**	**1.1**	**1.3**	**1.2**	**1.2**	**1.4**	**1.4**	**1.4**
+ Eff. Inl. Post	**0.8**	**0.7**	**0.8**	**1.1**	**1.0**	**0.9**	**1.2**	**1.2**	**1.1**	**1.4**	**1.3**	**1.2**	**1.4**	**1.3**	**1.4**	**1.5**	**1.5**	**1.6**

Table 4.2: Verification accuracy and efficiency measured on the Oxford5k dataset using a vocabulary of 1M visual words. The best, second-best, and third-best results are highlighted for each column.

	–	F100k	EU	US	EU+US	UK+EU+US	Ox+UK+EU+US	All
mAP [%]								
Pure Retrieval	70.2	60.1	55.5	54.0	50.7	49.0	48.1	47.8
FSM Aff	**72.2**	64.5	**60.1**	59.6	56.9	55.7	55.2	**54.9**
+ Eff. Inl. Eval	**72.4**	**64.9**	**60.6**	**60.1**	**57.5**	**56.3**	**55.7**	**55.4**
+ Eff. Inl. Post	72.1	**64.6**	60.0	59.6	56.9	55.6	55.0	**54.9**
FSM-R Aff	**72.2**	64.5	**60.1**	59.6	56.9	55.7	55.2	54.9
+ Eff. Inl. Eval	**72.3**	**64.7**	**60.4**	**59.9**	**57.3**	**56.1**	**55.5**	**55.2**
+ Eff. Inl. Post	72.0	64.5	59.8	59.4	56.7	55.4	54.9	54.7
FSM Sim	72.0	64.0	59.7	59.3	56.6	55.4	54.9	54.5
+ Eff. Inl. Eval	72.1	64.4	**60.1**	**59.8**	**57.0**	**55.8**	**55.3**	54.8
+ Eff. Inl. Post	71.9	64.0	59.5	59.2	56.5	55.2	54.7	54.3
FSM-R Sim	71.9	63.9	59.6	59.2	56.5	55.2	54.7	54.3
+ Eff. Inl. Eval	72.0	64.2	59.9	59.6	56.9	55.7	55.1	54.6
+ Eff. Inl. Post	71.8	63.8	59.3	59.0	56.2	55.0	54.5	54.0
HPM	70.0	60.5	54.8	54.4	50.9	49.5	48.8	48.5
ADV	69.4	61.9	58.5	58.3	55.7	54.3	53.8	53.4
PGM	69.9	59.7	54.7	53.1	49.8	48.0	47.3	46.9
Ours	71.5	63.4	58.6	58.2	55.5	54.4	53.8	53.6
+ Eff. Inl. Post	72.0	64.3	59.7	59.3	56.7	55.5	54.9	54.6
Runtime [s]								
FSM Aff	35.9	46.9	59.6	57.9	80.4	84.5	86.2	88.7
+ Eff. Inl. Eval	324.6	415.1	432.6	434.0	492.5	529.0	540.1	581.0
+ Eff. Inl. Post	35.9	47.2	60.2	58.8	81.7	84.2	88.1	89.9
FSM-R Aff	6.4	8.6	10.0	10.2	13.6	15.6	15.4	16.7
+ Eff. Inl. Eval	181.1	243.2	285.0	286.9	335.2	366.4	375.6	386.6
+ Eff. Inl. Post	6.6	8.9	10.4	10.6	14.2	16.1	16.4	17.6
FSM Sim	38.8	46.8	64.3	62.2	86.6	87.1	92.1	87.6
+ Eff. Inl. Eval	361.2	440.7	468.4	471.2	530.3	561.9	575.1	603.0
+ Eff. Inl. Post	39.2	48.4	64.6	63.1	88.3	88.9	91.0	88.5
FSM-R Sim	6.3	8.3	9.6	9.8	13.2	14.8	14.9	16.1
+ Eff. Inl. Eval	178.3	241.2	280.2	283.8	331.0	361.9	370.7	381.3
+ Eff. Inl. Post	6.5	8.6	10.1	10.3	13.9	15.6	15.7	17.0
HPM	**1.1**	**1.3**	**1.4**	**1.5**	**2.0**	**2.4**	**2.4**	**2.7**
ADV	2.0	2.7	3.2	3.1	4.3	4.9	5.2	5.3
PGM	15.5	15.1	15.1	14.9	15.0	15.7	15.5	15.3
Ours	**1.0**	**1.4**	**1.7**	**1.7**	**2.4**	**2.8**	**2.9**	**3.1**
+ Eff. Inl. Post	**1.2**	**1.7**	**2.1**	**2.1**	**2.7**	**3.2**	**3.1**	**3.5**

Table 4.3: Verification accuracy and efficiency measured on the Paris6k dataset using a vocabulary of 1M visual words. The best, second-best, and third-best results are highlighted for each column.

	–	EU	US	EU+US	UK+EU+US	Ox+UK+EU+US
mAP [%]						
Pure Retrieval	97.4	93.0	93.3	91.0	90.3	89.9
FSM Aff	**98.1**	**95.8**	**96.5**	**95.2**	**94.9**	**94.8**
+ Eff. Inl. Eval	**98.1**	**95.9**	**96.7**	**95.4**	**95.1**	**95.0**
+ Eff. Inl. Post	**98.0**	95.6	96.4	95.0	94.7	94.5
FSM-R Aff	**98.1**	**95.8**	**96.5**	**95.2**	**94.9**	**94.8**
+ Eff. Inl. Eval	**98.1**	**95.8**	**96.6**	**95.3**	**95.0**	**94.9**
+ Eff. Inl. Post	**97.9**	95.5	96.3	94.9	94.6	94.4
FSM Sim	**98.1**	**95.7**	**96.5**	95.1	94.8	94.6
+ Eff. Inl. Eval	**98.1**	**95.9**	**96.7**	**95.3**	**95.1**	**94.9**
+ Eff. Inl. Post	**97.9**	95.4	96.3	94.8	94.5	94.4
FSM-R Sim	**98.0**	95.6	96.4	95.0	94.7	94.6
+ Eff. Inl. Eval	**98.0**	**95.8**	**96.6**	**95.2**	**94.9**	**94.8**
+ Eff. Inl. Post	**97.9**	95.3	96.2	94.7	94.4	94.2
HPM	96.6	92.1	92.4	90.7	90.2	89.9
ADV	**97.9**	95.1	95.7	94.2	93.9	93.8
PGM	97.0	91.8	91.7	89.4	88.7	88.1
Ours	97.8	95.1	95.6	94.3	94.0	93.9
+ Eff. Inl. Post	97.8	95.3	96.0	94.8	94.5	94.3
Runtime [s]						
FSM Aff	197.3	292.2	282.7	374.3	382.1	383.1
+ Eff. Inl. Eval	2709.5	3543.6	3465.8	3879.3	4067.2	4185.8
+ Eff. Inl. Post	197.9	300.4	289.0	385.3	392.3	396.1
FSM-R Aff	38.3	61.4	61.3	85.6	91.5	93.6
+ Eff. Inl. Eval	1312.7	2079.1	2007.9	2356.2	2508.8	2599.1
+ Eff. Inl. Post	39.7	64.6	64.2	89.9	96.2	98.6
FSM Sim	216.9	322.8	309.9	418.8	423.0	420.4
+ Eff. Inl. Eval	2874.7	3702.7	3618.0	4032.9	4221.9	4332.6
+ Eff. Inl. Post	218.6	331.8	318.2	432.1	436.3	440.4
FSM-R Sim	37.9	59.4	59.3	82.8	88.3	90.7
+ Eff. Inl. Eval	1303.7	2055.8	1986.7	2325.2	2475.3	2563.4
+ Eff. Inl. Post	39.4	62.7	62.4	87.4	93.8	95.8
HPM	**6.4**	**9.7**	**9.8**	**12.0**	**12.7**	**13.5**
ADV	14.8	22.2	21.6	27.4	29.1	33.1
PGM	44.8	45.5	45.4	46.9	48.3	48.9
Ours	**6.3**	**11.0**	**10.6**	**13.2**	**13.9**	**15.0**
+ Eff. Inl. Post	**7.6**	**13.4**	**13.1**	**15.6**	**16.3**	**17.3**

Table 4.4: Verification accuracy and efficiency measured on our new dataset using a vocabulary of 1M visual words. The best, second-best, and third-best results are highlighted for each column.

hypotheses that are verified. Table 4.1 shows the results obtained for different parameter configurations. For this experiment, we used the mean transformation per bin to generate the hypotheses (see Section 4.2.3). As can be seen from the table, the verification performance of our approach is rather insensitive against the number of bins and verified transformations, although increasing T has a slightly positive impact on the measured mAP. Naturally, increasing the number of bins and T also increases the overall runtime, but the increase is rather small. We also experimented with fewer (2 and 4) and more (128) bin sizes but found that the former resulted in a significant drop in mAP while the latter did not noticeably improve mAP. For all following experiments, we use $T = 30$ transformation proposals, $n_x = 64$ and $n_y = 64$ translation bins, $n_\sigma = 32$ scale bins, and $n_\theta = 8$ rotation bins.

Impact of Refining the Transformation Hypotheses

In Section 4.2.3, we proposed to use the mean transformation for the bins in the voting space, arguing that the transformation defined by the center coordinate is not accurate enough. We measure an mAP of 76.2 when using the center coordinate and an mAP of 80.1 when using the mean transformation. At the same time, we do not observe an increase in runtime when computing the running mean. This clearly confirms our approach for refining the transformation hypotheses. In addition, we also experimented with using the median transformation per bin. As expected, the measured mAP increases to 80.3 since the median is less affected by outliers than the mean. However, this increase comes at significantly slower runtimes of 2.5 seconds, compared to 0.9 seconds when using the mean. This increase is caused by the fact that computing the median requires the individual transformations to be stored and then partially sorted.

4.3.4 Comparison

In the next experiments, we verify the claim that our approach achieves a verification accuracy similar to hypothesize-and-verify methods while obtaining faster runtimes than voting-based methods. Towards this goal, we compare our approach against state-of-the-art methods for spatial verification. Hypothesize-and-verify approaches are represented by different variants of FSM [243]: The original *FSM* method exhaustively evaluates each transformation hypothesis obtained from a single feature match. The 1-point-RANSAC version of FSM (*FSM-R*) randomly samples from the hypotheses and terminates spatial verification once the probability of finding a better hypothesis falls below a threshold of $\hat{p} = 0.99$. For both FSM and FSM-R, we use two variants that either estimate an affine (*Aff.*) or a similarity (*Sim.*) transformation from each correspondence. All variants use local optimization to estimate an affine transformation from the inlier matches, independent of the type of transformation estimated from the individual correspondences. Besides ranking transformation hypotheses based on their numbers of inliers, we also evaluate FSM and FSM-R in combination with the effective inlier count (*cf.* Equation 4.13). We

	–	F100k	Ox	UK	EU	US	Ox+UK	Ox+EU	Ox+US	UK+EU	UK+US	EU+US	Ox+UK+EU	Ox+UK+US	Ox+EU+US	UK+EU+US	Ox+UK+EU+US	All
mAP [%]																		
Pure Retrieval	76.2	66.4	64.1	60.3	60.3	59.6	57.6	57.4	57.2	56.3	55.9	55.8	54.4	54.3	54.1	53.7	52.3	51.7
FSM Aff	**79.9**	75.1	72.9	**70.3**	70.9	71.4	68.6	69.2	69.7	68.3	68.6	69.1	67.2	67.3	67.9	67.5	66.4	66.3
+ Eff. Inl. Eval	**79.8**	**75.9**	**73.7**	**71.0**	**71.8**	**72.5**	**69.7**	**70.3**	**71.0**	**69.6**	**69.9**	**70.5**	**68.6**	**68.8**	**69.5**	**69.1**	**68.1**	**67.9**
+ Eff. Inl. Post	79.6	75.2	73.3	70.1	**71.1**	71.7	68.8	69.6	70.2	**68.5**	68.8	**69.5**	67.5	67.7	68.4	67.8	66.8	66.6
FSM-R Aff	**79.9**	75.3	72.9	**70.3**	70.9	71.4	68.6	69.2	69.6	68.3	68.6	69.1	67.2	67.3	67.9	67.4	66.4	66.3
+ Eff. Inl. Eval	**79.8**	**75.9**	**73.7**	**71.0**	**71.8**	**72.5**	**69.7**	**70.3**	**70.9**	**69.6**	**69.9**	**70.5**	**68.6**	**68.8**	**69.4**	**69.0**	**68.1**	**67.7**
+ Eff. Inl. Post	79.6	75.2	73.1	70.0	71.0	71.7	68.6	69.5	70.0	68.3	68.7	69.4	67.3	67.5	68.3	67.6	66.6	66.3
FSM Sim	**79.8**	74.8	72.3	69.4	70.2	70.9	67.6	68.3	68.9	67.3	67.7	68.3	66.1	66.4	67.0	66.5	65.5	65.3
+ Eff. Inl. Eval	**79.9**	**75.4**	**73.5**	**70.3**	**71.2**	**72.1**	**69.0**	**69.8**	70.5	**68.6**	**69.0**	**69.7**	**67.7**	**68.0**	**68.7**	68.1	**67.2**	66.7
+ Eff. Inl. Post	79.1	74.4	72.5	69.1	70.0	71.0	67.7	68.5	69.2	67.2	67.8	68.5	66.2	66.6	67.4	66.6	65.7	65.3
FSM-R Sim	**79.8**	74.9	72.3	69.3	70.2	70.9	67.6	68.3	68.9	67.3	67.7	68.3	66.1	66.4	67.0	66.5	65.5	65.3
+ Eff. Inl. Eval	**79.8**	75.3	**73.4**	70.1	71.0	71.9	68.8	69.5	70.2	68.3	68.8	**69.5**	67.4	67.8	68.5	67.8	67.0	66.4
+ Eff. Inl. Post	79.0	74.3	72.4	69.0	69.9	70.8	67.6	68.3	69.0	67.1	67.6	68.3	66.1	66.5	67.2	66.5	65.5	65.2
HPM	73.4	65.4	63.3	59.7	59.6	60.0	58.1	58.1	58.2	57.1	57.1	57.4	56.0	56.0	56.3	55.8	54.9	54.4
ADV	78.5	73.7	71.9	68.2	68.8	70.0	66.8	67.4	68.3	66.1	66.5	67.0	65.1	65.4	65.9	65.2	64.3	64.0
PGM	76.0	64.6	62.5	58.7	59.1	57.9	55.5	55.3	54.9	54.6	53.9	53.8	52.6	52.3	52.0	51.4	50.2	49.4
Ours	**80.1**	74.5	71.9	68.7	69.5	69.7	67.0	67.6	67.8	66.6	66.7	67.2	65.3	65.4	65.9	65.4	64.4	64.1
+ Eff. Inl. Post	**79.8**	**75.7**	**73.5**	**70.5**	**71.1**	**72.3**	**69.2**	**69.7**	**70.6**	**68.6**	**69.2**	**69.7**	**67.6**	**68.1**	68.6	**67.9**	67.0	**66.8**
Runtime [s]																		
FSM Aff	31.1	43.4	44.1	46.3	45.9	49.7	68.0	68.1	70.1	69.9	70.5	68.7	73.5	71.2	69.9	69.8	73.3	73.9
+ Eff. Inl. Eval	309.6	356.4	381.0	456.5	446.8	408.6	476.8	467.1	438.0	500.7	488.6	475.3	523.4	495.0	484.7	512.1	522.7	506.2
+ Eff. Inl. Post	31.0	43.2	44.5	46.8	46.1	49.3	68.9	68.4	71.0	70.4	71.3	71.1	73.9	71.2	71.5	71.0	75.5	77.2
FSM-R Aff	4.3	5.7	6.6	8.4	8.0	7.5	10.4	9.9	9.7	11.2	11.1	10.6	12.1	11.2	10.7	11.3	12.0	12.0
+ Eff. Inl. Eval	193.1	237.1	258.2	333.9	324.8	282.5	347.9	339.2	303.9	370.1	357.5	344.1	387.0	363.2	354.4	381.0	390.0	378.7
+ Eff. Inl. Post	4.4	5.9	6.9	8.9	8.4	7.9	11.3	10.7	10.4	12.2	11.8	11.4	13.2	12.2	11.7	12.4	14.2	15.0
FSM Sim	33.0	45.0	47.5	48.8	47.8	51.0	70.9	70.9	73.5	72.2	73.3	72.1	75.9	73.4	74.2	73.4	76.3	76.1
+ Eff. Inl. Eval	317.3	367.3	389.6	476.3	454.6	416.4	485.7	476.2	446.9	508.1	497.5	482.6	531.2	502.1	491.8	520.8	530.7	516.1
+ Eff. Inl. Post	33.1	45.6	47.4	53.7	48.2	50.2	72.2	71.6	75.0	73.4	74.9	72.9	77.4	74.6	73.6	75.3	78.4	79.2
FSM-R Sim	4.2	5.5	6.3	8.1	7.6	7.2	10.1	9.5	9.5	10.8	10.6	10.1	11.6	10.8	10.2	10.9	11.6	11.5
+ Eff. Inl. Eval	192.5	236.6	257.0	332.4	323.2	281.2	346.0	336.4	301.7	368.7	356.2	341.3	385.9	361.4	352.9	378.1	386.7	376.2
+ Eff. Inl. Post	4.3	5.8	6.7	9.0	8.1	7.7	10.8	10.3	10.4	11.8	11.5	11.0	12.6	11.7	11.2	11.9	13.6	14.3
HPM	**1.1**	**1.4**	**1.7**	**1.9**	**1.8**	**1.6**	**2.2**	**2.2**	**1.9**	**2.3**	**2.2**	**2.2**	**2.3**	**2.2**	**2.1**	**2.4**	**2.5**	**2.6**
ADV	2.3	3.0	3.4	4.6	4.3	3.9	5.1	4.9	4.6	5.5	5.5	5.3	6.0	5.4	5.0	5.1	**5.4**	**5.3**
PGM	15.0	15.2	15.4	15.6	15.6	15.9	15.8	15.7	16.2	16.0	16.2	15.9	16.7	16.1	16.1	16.0	16.1	15.5
Ours	**0.9**	**1.6**	**1.7**	**2.2**	**2.0**	**1.8**	**2.4**	**2.3**	**2.2**	**2.5**	**2.4**	**2.4**	**2.8**	**2.7**	**2.9**	**2.9**	**3.1**	**2.8**
+ Eff. Inl. Post	**1.2**	**1.8**	**1.9**	**2.5**	**2.3**	**2.0**	**2.7**	**2.7**	**2.5**	**3.0**	**2.8**	**2.9**	**3.4**	**3.4**	**3.6**	**3.5**	5.6	5.9

Table 4.5: Verification accuracy and efficiency measured on the Oxford5k dataset using a vocabulary of 200K visual words. The best, second-best, and third-best results are highlighted for each column.

	–	F100k	EU	US	EU+US	UK+EU+US	Ox+UK+EU+US	All
mAP [%]								
Pure Retrieval	71.2	60.2	56.2	54.6	51.2	49.5	48.7	47.7
FSM Aff	74.2	**66.0**	62.1	62.0	**59.1**	**57.8**	57.1	**56.2**
+ Eff. Inl. Eval	**74.5**	**66.4**	**62.5**	**62.5**	**59.5**	**58.3**	**57.6**	**56.6**
+ Eff. Inl. Post	73.9	65.6	61.5	61.4	58.3	57.1	56.4	55.6
FSM-R Aff	74.2	**66.0**	62.1	**62.1**	**59.1**	**57.8**	**57.2**	**56.2**
+ Eff. Inl. Eval	**74.3**	**66.2**	**62.3**	**62.3**	**59.3**	**58.1**	**57.4**	**56.4**
+ Eff. Inl. Post	73.7	65.3	61.3	61.2	58.2	56.9	56.3	55.4
FSM Sim	74.1	65.7	61.8	61.8	58.8	57.5	56.9	55.9
+ Eff. Inl. Eval	**74.4**	**66.0**	**62.2**	**62.1**	**59.1**	**57.8**	**57.2**	**56.2**
+ Eff. Inl. Post	73.8	65.2	61.3	61.2	58.2	56.8	56.2	55.3
FSM-R Sim	73.9	65.4	61.5	61.5	58.6	57.3	56.7	55.7
+ Eff. Inl. Eval	74.1	65.6	61.8	61.7	58.8	57.5	56.9	55.9
+ Eff. Inl. Post	73.6	64.9	60.9	60.9	57.9	56.6	55.9	55.0
HPM	70.8	60.8	56.1	55.5	52.2	50.8	50.2	49.3
ADV	71.5	62.9	60.1	60.1	57.3	56.0	55.3	54.2
PGM	70.7	59.3	54.8	53.6	50.0	48.1	47.1	46.2
Ours	73.4	64.9	60.9	60.7	57.8	56.6	56.0	54.9
+ Eff. Inl. Post	73.9	65.6	61.9	61.8	58.9	57.5	57.0	55.9
Runtime [s]								
FSM Aff	55.9	64.7	82.2	87.2	125.1	131.6	137.6	118.6
+ Eff. Inl. Eval	545.1	668.6	782.8	800.9	901.4	964.0	987.3	982.0
+ Eff. Inl. Post	55.6	67.4	83.1	89.5	127.5	133.5	144.9	124.2
FSM-R Aff	10.8	17.5	19.5	21.0	26.7	30.5	31.6	33.8
+ Eff. Inl. Eval	314.0	463.0	529.1	541.2	614.8	671.1	688.1	723.2
+ Eff. Inl. Post	11.0	17.8	20.1	21.7	27.8	32.0	35.2	39.2
FSM Sim	61.7	68.5	91.4	97.2	137.4	142.8	151.6	120.9
+ Eff. Inl. Eval	570.0	674.0	816.6	834.8	925.8	1005.1	1013.7	983.4
+ Eff. Inl. Post	62.0	68.4	93.0	98.3	140.0	155.9	152.8	126.7
FSM-R Sim	10.3	16.6	18.7	20.1	25.4	29.4	30.3	32.3
+ Eff. Inl. Eval	310.2	459.5	523.6	536.2	608.7	663.2	679.3	715.2
+ Eff. Inl. Post	10.6	17.2	19.3	20.9	26.7	31.0	33.9	37.9
HPM	**2.2**	**2.3**	**2.8**	**2.8**	**3.8**	**4.2**	**4.4**	**4.5**
ADV	3.7	5.4	6.0	6.0	7.7	8.5	8.6	**9.1**
PGM	19.1	19.8	19.7	19.9	20.3	20.7	22.1	23.5
Ours	**2.5**	**2.8**	**3.3**	**3.3**	**4.4**	**4.6**	**4.9**	**5.4**
+ Eff. Inl. Post	**2.9**	**3.2**	**3.7**	**3.8**	**5.0**	**5.6**	**8.4**	10.0

Table 4.6: Verification accuracy and efficiency measured on the Paris6k dataset using a vocabulary of 200K visual words. The best, second-best, and third-best results are highlighted for each column.

	–	EU	US	EU+US	UK+EU+US	Ox+UK+EU+US
mAP [%]						
Pure Retrieval	97.1	92.3	92.5	90.1	89.4	88.9
Aff	**98.2**	**95.8**	**96.6**	**95.2**	**94.9**	**94.7**
+ Eff. Inl. Eval	**98.2**	**95.9**	**96.8**	**95.3**	**95.0**	**94.9**
+ Eff. Inl. Post	97.9	95.5	96.4	94.9	94.6	94.4
Aff RANSAC	**98.2**	**95.8**	**96.6**	**95.2**	**94.8**	**94.7**
+ Eff. Inl. Eval	**98.1**	**95.8**	**96.6**	**95.2**	**94.9**	**94.7**
+ Eff. Inl. Post	97.8	95.4	96.3	94.7	94.4	94.3
Sim	**98.1**	**95.8**	**96.6**	**95.1**	**94.8**	94.6
+ Eff. Inl. Eval	**98.1**	**95.9**	**96.7**	**95.3**	**95.0**	**94.8**
+ Eff. Inl. Post	97.8	95.4	96.2	94.8	94.5	94.3
Sim RANSAC	**98.0**	**95.7**	96.4	95.0	94.7	94.5
+ Eff. Inl. Eval	**98.0**	**95.8**	**96.6**	**95.2**	**94.9**	**94.7**
+ Eff. Inl. Post	97.6	95.2	96.0	94.6	94.3	94.1
HPM	96.0	91.6	92.0	90.3	89.9	89.6
ADV	97.5	94.2	94.9	93.2	92.9	92.7
PGM	96.4	90.8	90.4	88.0	87.3	86.7
Ours	97.8	95.2	95.8	94.4	94.1	93.9
+ Eff. Inl. Post	97.8	95.4	96.1	94.7	94.4	94.3
Runtime [s]						
Aff	255.6	341.4	323.1	437.6	450.1	450.7
+ Eff. Inl. Eval	4234.8	6009.0	5599.8	6372.5	6629.2	6744.8
+ Eff. Inl. Post	256.0	345.2	325.9	441.8	455.7	457.0
Aff RANSAC	55.7	99.5	92.1	123.6	132.2	136.5
+ Eff. Inl. Eval	2485.8	4103.6	3757.2	4361.7	4574.0	4671.2
+ Eff. Inl. Post	57.8	104.0	96.3	128.7	138.6	143.8
Sim	270.5	360.7	339.7	466.5	478.3	475.2
+ Eff. Inl. Eval	4378.2	6146.8	5732.3	6501.5	6758.4	6864.2
+ Eff. Inl. Post	273.5	366.1	344.3	471.9	485.8	483.3
Sim RANSAC	54.8	96.2	88.9	119.3	128.1	132.2
+ Eff. Inl. Eval	2479.8	4087.6	3741.5	4338.6	4544.2	4639.7
+ Eff. Inl. Post	57.2	101.3	93.5	124.9	134.9	139.5
HPM	**12.4**	**17.1**	**16.4**	**21.6**	**21.9**	**22.3**
ADV	24.8	35.7	32.4	39.9	42.1	46.4
PGM	54.5	57.5	54.9	57.9	58.5	59.9
Ours	**13.7**	**19.8**	**19.7**	**26.7**	**27.5**	**23.4**
+ Eff. Inl. Post	**16.5**	**24.3**	**23.8**	**29.7**	**30.8**	**28.0**

Table 4.7: Verification accuracy and efficiency measured on our new dataset using a vocabulary of 200K visual words. The best, second-best, and third-best results are highlighted for each column.

again evaluate two variants. The first variant uses the effective inlier count instead of the standard inlier count during verification (*Eff. Inl. Eval*). The second variant simply applies the effective inlier count as a post-processing step (*Eff. Inl. Post*) on the best transformation found by FSM and FSM-R. The effective inlier count of this hypothesis is then used for re-ranking after spatial verification.

We also compare our approach against the current state-of-the-art approaches for voting-based verification: HPM [19], ADV [368], and PGM [190]. Since the three methods do not return inlier matches, they cannot be combined with the effective inlier count. Notice that the results reported in [190, 368] are not directly comparable due to using different types of features and vocabularies of different sizes trained on different datasets. Thus, our results were obtained with our own implementations of HPM (without idf-weighting), ADV, and PGM.

Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6, Table 4.7, Table 4.8, Table 4.9, and Table 4.10 present the accuracy and runtimes on the Oxford5k, Paris5k, and new datasets for different vocabulary sizes, respectively. There are multiple interesting insights to be gained from our results: Both FSM and FSM-R outperform HPM, ADV, and PGM in terms of mAP. The result is especially pronounced on the Paris6k dataset (see Table 4.6). Using early stopping (FSM-R) rather than evaluating all possible transformations (FSM) significantly accelerates the verification without a significant impact on mAP. In fact, FSM-R is not more than a factor-of-4 slower than ADV, which is surprising given that one of the main arguments [19] for voting-based methods is that they are about an order of magnitude faster than FSM. Moreover, there is little difference between using an affine or similarity transformation for both FSM and FSM-R, likely due to the local optimization step. Compared to both ADV and PGM, our method achieves faster runtimes, which is most pronounced on our new dataset, where more features are found in each image. At the same time, our method also achieves a better accuracy. Especially on the Oxford5k and Paris6k datasets, our approach performs nearly as well as FSM and FSM-R, which are significantly slower than our method.

Influence of the Distractor Sets

Table 4.5, Table 4.6, and Table 4.7 report the impact of combining each dataset with various combinations of the distractor sets. While using the effective inlier count provides little benefits without distractors, we observe a noticeable gain when adding distractors and thus making the problem harder. Naturally, the best results are obtained by directly incorporating the count into the verification stage of FSM and FSM-R. However, this comes at significant runtime costs since it needs to be evaluated often. Yet, using the count as a post-processing step incurs only negligible cost. Combining the effective inlier count with our method further increases the accuracy of our method.

The distractor set showing the largest decrease in mAP in combination with the Oxford5k dataset was the image set from 30 cities around the UK and not *Ox*. This is somewhat counter-intuitive, since it might be expected that the distractor set

	–	F100k	Ox	UK	EU	US	Ox+UK	Ox+EU	Ox+US	UK+EU	UK+US	EU+US	Ox+UK+EU	Ox+UK+US	Ox+EU+US	UK+EU+US	Ox+UK+EU+US	All
mAP [%]																		
Pure Retrieval	72.5	62.7	61.0	55.1	54.3	55.9	53.0	52.6	53.6	50.9	51.0	50.9	49.7	49.8	49.7	48.7	47.8	47.4
FSM Aff	**82.0**	**78.1**	**75.8**	**71.6**	**72.5**	**75.0**	**70.0**	**70.8**	**72.5**	**69.0**	**70.0**	**70.6**	**67.7**	**68.7**	**68.9**	**67.8**	**66.1**	**65.5**
+ Eff. Inl. Eval	**81.4**	**77.8**	**75.5**	**71.3**	**71.9**	**74.6**	**69.7**	**70.4**	**72.1**	**68.7**	**69.7**	**70.2**	**67.5**	**68.4**	**68.6**	**67.5**	**65.9**	**65.5**
+ Eff. Inl. Post	79.5	75.7	73.5	69.3	70.1	72.3	67.8	68.7	70.3	66.8	67.7	68.4	65.8	66.6	66.9	65.7	64.2	63.7
FSM-R Aff	**81.8**	**77.9**	**75.5**	**71.3**	**72.0**	**74.6**	**69.6**	**70.4**	**72.1**	**68.6**	**69.7**	**70.1**	**67.3**	**68.3**	**68.4**	**67.4**	**65.7**	65.2
+ Eff. Inl. Eval	81.2	77.6	**75.1**	70.9	71.6	**74.2**	69.4	**70.2**	**71.7**	68.4	69.3	69.8	67.2	68.1	68.3	67.2	65.6	**65.3**
+ Eff. Inl. Post	79.2	75.5	73.1	68.9	69.6	71.9	67.4	68.4	69.8	66.5	67.4	68.0	65.5	66.3	66.5	65.4	63.9	63.5
FSM Sim	81.2	77.5	75.0	**71.0**	71.8	74.1	69.3	70.1	71.6	68.4	**69.4**	69.9	67.1	67.9	68.2	67.3	65.5	65.1
+ Eff. Inl. Eval	80.6	77.0	74.8	**71.0**	71.8	74.0	69.3	**70.2**	**71.7**	68.4	69.3	69.9	67.2	68.0	68.3	67.3	**65.7**	**65.5**
+ Eff. Inl. Post	78.2	74.8	72.6	68.6	69.3	71.7	67.2	67.9	69.7	66.2	67.0	67.6	65.1	65.9	66.3	65.2	63.6	63.3
FSM-R Sim	80.6	77.3	74.5	70.5	71.2	73.6	68.9	69.6	71.1	67.9	68.9	69.4	66.6	67.5	67.7	66.8	65.1	64.9
+ Eff. Inl. Eval	80.2	76.9	74.5	70.6	71.4	73.6	69.0	69.8	71.3	68.0	68.9	69.6	66.9	67.7	68.0	66.9	65.4	**65.4**
+ Eff. Inl. Post	77.5	74.6	71.7	67.7	68.4	70.8	66.3	67.0	68.8	65.3	66.2	66.8	64.3	65.1	65.5	64.4	62.9	63.1
HPM	70.3	64.3	62.8	58.7	58.0	60.4	57.6	57.0	58.8	56.0	56.8	56.5	55.2	55.9	55.6	54.9	54.1	53.7
ADV	73.5	66.6	64.7	59.7	59.5	63.1	58.6	58.4	61.0	57.1	58.4	58.2	56.4	57.5	57.3	56.5	55.7	55.5
PGM	72.1	60.7	58.8	51.8	51.8	52.0	49.5	49.9	50.0	47.6	47.2	47.9	46.6	46.4	46.7	45.8	44.7	44.3
Ours	80.6	76.1	73.1	69.2	69.9	71.7	67.4	68.0	69.3	66.6	67.4	67.9	65.2	65.9	66.2	65.3	63.8	63.5
+ Eff. Inl. Post	79.8	76.3	73.8	69.7	70.0	72.4	68.2	68.7	70.4	67.0	68.0	68.3	65.9	66.8	66.9	65.8	64.5	64.0
Runtime [s]																		
FSM Aff	94.2	118.8	126.2	157.9	162.4	144.2	182.7	180.8	168.5	192.1	182.6	185.0	191.6	182.1	180.6	190.9	191.9	189.9
+ Eff. Inl. Eval	1493.8	1751.3	1807.1	2149.0	2200.7	1930.2	2191.4	2163.7	2001.0	2280.4	2183.7	2212.9	2253.4	2164.3	2146.6	2249.9	2238.0	2218.4
+ Eff. Inl. Post	94.2	119.0	126.4	158.3	162.7	145.2	184.0	183.0	171.4	193.7	184.3	187.5	194.0	184.2	182.1	192.8	193.3	192.5
FSM-R Aff	53.1	71.1	75.6	99.8	101.7	84.5	104.9	103.2	91.8	111.1	105.0	107.7	110.9	103.5	102.0	109.5	109.6	108.9
+ Eff. Inl. Eval	1230.9	1430.5	1470.7	1740.0	1786.8	1573.7	1761.0	1739.8	1612.4	1825.4	1752.7	1783.9	1822.6	1734.7	1721.0	1795.9	1784.6	1765.6
+ Eff. Inl. Post	53.9	72.2	76.7	101.4	103.4	86.1	107.4	105.4	94.0	114.0	107.1	109.7	113.1	105.7	104.3	111.9	111.6	111.4
FSM Sim	96.0	119.9	127.2	158.3	163.8	148.0	188.8	187.9	176.1	198.6	188.0	193.4	196.5	187.3	184.8	195.5	195.3	192.9
+ Eff. Inl. Eval	1522.8	1777.6	1832.0	2180.5	2233.0	1955.3	2221.8	2196.2	2029.2	2313.4	2212.4	2245.8	2281.4	2192.7	2178.0	2279.2	2265.2	2244.9
+ Eff. Inl. Post	97.1	120.4	128.3	160.8	165.7	149.0	190.7	188.9	177.8	200.6	190.0	195.3	199.2	189.7	188.0	198.4	197.5	195.4
FSM-R Sim	51.8	68.3	72.6	95.7	98.1	81.3	101.0	99.6	88.5	107.4	100.8	103.8	106.2	99.4	97.8	105.4	105.3	104.5
+ Eff. Inl. Eval	1239.3	1431.7	1469.0	1734.5	1789.7	1573.7	1755.5	1739.3	1609.6	1823.4	1751.9	1786.4	1813.2	1728.0	1720.3	1795.4	1782.1	1762.2
+ Eff. Inl. Post	52.6	69.4	73.8	97.3	99.9	82.9	103.4	102.1	90.6	109.6	103.1	106.2	108.7	101.7	100.1	108.0	107.6	107.1
HPM	**7.7**	**9.5**	**9.5**	**11.9**	**11.5**	**10.2**	**14.6**	**14.3**	**12.7**	**15.2**	**14.6**	**14.3**	**15.0**	**14.2**	**14.6**	**14.7**	**14.8**	**14.8**
ADV	14.1	15.9	16.6	19.9	21.0	**18.5**	22.1	21.4	20.6	22.9	21.8	23.0	23.2	21.9	21.6	22.8	22.5	22.3
PGM	17.8	18.9	18.7	19.8	20.7	19.5	20.7	20.7	**20.4**	21.3	21.0	21.1	20.9	20.6	20.7	21.0	**21.2**	20.7
Ours	**7.9**	**9.6**	**9.9**	**12.2**	**12.0**	**10.5**	**13.6**	**13.4**	**12.7**	**14.3**	**13.9**	**13.5**	**15.2**	**14.4**	**14.2**	**15.1**	**14.8**	**14.6**
+ Eff. Inl. Post	**8.2**	**10.6**	**11.1**	**13.7**	**13.6**	**12.0**	**15.6**	**15.3**	**14.3**	**16.4**	**15.7**	**15.5**	**17.1**	**16.0**	**15.8**	**16.8**	**16.2**	**16.1**

Table 4.8: Verification accuracy and efficiency measured on the Oxford5k dataset using a vocabulary of 20K visual words. The best, second-best, and third-best results are highlighted for each column.

	–	F100k	EU	US	EU+US	UK+EU+US	Ox+UK+EU+US	All
mAP [%]								
Pure Retrieval	68.5	56.9	52.1	51.5	47.6	45.8	45.2	44.2
FSM Aff	**74.2**	**65.6**	**60.6**	**61.7**	**58.1**	**56.6**	**55.9**	**54.9**
+ Eff. Inl. Eval	**74.4**	**65.6**	**60.6**	**61.7**	**58.0**	**56.5**	**55.9**	**54.8**
+ Eff. Inl. Post	73.2	64.4	59.3	60.4	56.8	55.3	54.7	53.7
FSM-R Aff	**74.1**	**65.4**	**60.5**	**61.6**	**58.0**	**56.5**	**55.9**	**54.7**
+ Eff. Inl. Eval	**74.1**	**65.2**	**60.3**	**61.3**	**57.7**	**56.2**	**55.5**	54.4
+ Eff. Inl. Post	73.0	64.0	59.0	60.1	56.5	55.0	54.4	53.4
FSM Sim	73.9	**65.2**	60.2	**61.3**	57.6	**56.2**	**55.6**	54.6
+ Eff. Inl. Eval	73.8	65.1	60.0	61.2	57.5	56.0	55.4	54.4
+ Eff. Inl. Post	72.6	63.8	58.7	59.8	56.3	55.0	54.3	53.2
FSM-R Sim	73.5	64.9	59.8	60.9	57.3	55.9	55.3	54.4
+ Eff. Inl. Eval	73.4	64.8	59.7	60.8	57.1	55.7	55.0	54.2
+ Eff. Inl. Post	72.2	63.4	58.3	59.5	56.0	54.6	54.0	52.8
HPM	68.7	58.4	52.9	53.8	50.1	48.7	48.1	47.2
ADV	67.2	56.3	53.9	55.3	51.7	50.2	49.3	48.3
PGM	67.1	55.4	50.5	48.9	45.5	43.8	43.2	42.3
Ours	73.0	63.7	58.6	59.6	56.0	54.7	54.0	53.0
+ Eff. Inl. Post	73.1	63.8	58.8	60.1	56.4	55.0	54.3	53.3
Runtime [s]								
FSM Aff	145.0	223.6	234.9	224.1	282.3	309.7	341.2	349.4
+ Eff. Inl. Eval	1815.1	2396.9	2530.6	2500.3	2747.4	2776.0	2875.6	3002.9
+ Eff. Inl. Post	145.6	223.7	235.3	224.8	285.5	310.8	344.2	352.6
FSM-R Aff	78.3	128.6	135.0	126.7	151.2	165.8	182.0	185.7
+ Eff. Inl. Eval	1367.1	1697.9	1834.5	1746.6	1896.4	1932.2	1981.0	2021.5
+ Eff. Inl. Post	79.2	129.9	136.6	128.2	153.7	168.3	185.0	188.4
FSM Sim	143.5	218.8	230.6	218.6	284.0	306.2	334.1	343.2
+ Eff. Inl. Eval	1827.4	2422.3	2539.5	2528.2	2792.4	2789.7	2888.0	3031.2
+ Eff. Inl. Post	144.1	220.2	231.2	220.2	285.4	308.6	339.8	347.1
FSM-R Sim	75.9	123.6	130.0	121.2	147.4	160.3	173.6	177.6
+ Eff. Inl. Eval	1366.3	1693.0	1829.3	1737.2	1892.9	1920.8	1966.0	2005.7
+ Eff. Inl. Post	76.9	125.0	131.8	123.0	150.3	162.7	176.4	180.3
HPM	**8.6**	**12.5**	**15.0**	**13.4**	**17.4**	**17.7**	**18.5**	**18.8**
ADV	16.3	21.1	22.8	21.6	25.4	26.6	27.6	28.0
PGM	22.4	24.2	24.8	25.0	26.1	26.6	27.1	27.8
Ours	**9.3**	**13.1**	**15.7**	**13.6**	**17.3**	**18.4**	**19.5**	**19.8**
+ Eff. Inl. Post	**9.8**	**14.4**	**17.4**	**15.1**	**19.3**	**20.6**	**21.8**	**22.2**

Table 4.9: Verification accuracy and efficiency measured on the Paris6k dataset using a vocabulary of 20K visual words. The best, second-best, and third-best results are highlighted for each column.

	–	EU	US	EU+US	UK+EU+US	Ox+UK+EU+US
mAP [%]						
Pure Retrieval	94.8	87.8	88.4	85.4	84.5	84.1
FSM Aff	**97.8**	**94.1**	**95.2**	**93.1**	**92.4**	**92.2**
+ Eff. Inl. Eval	**97.6**	**94.0**	**95.2**	**93.1**	**92.4**	**92.1**
+ Eff. Inl. Post	97.1	93.4	94.6	92.5	91.8	91.5
FSM-R Aff	**97.7**	**94.0**	**95.2**	**93.1**	**92.4**	**92.1**
+ Eff. Inl. Eval	97.4	93.8	94.9	92.8	**92.2**	91.9
+ Eff. Inl. Post	96.9	93.1	94.4	92.2	91.6	91.3
FSM Sim	**97.7**	**94.0**	**95.1**	**93.0**	**92.4**	**92.1**
+ Eff. Inl. Eval	**97.6**	**93.9**	**95.1**	**92.9**	**92.3**	**92.0**
+ Eff. Inl. Post	97.0	93.3	94.5	92.4	91.8	91.5
FSM-R Sim	97.5	93.8	**95.0**	92.8	**92.2**	91.9
+ Eff. Inl. Eval	97.3	93.6	94.8	92.7	92.0	91.8
+ Eff. Inl. Post	96.8	93.1	94.3	92.2	91.5	91.3
HPM	92.9	87.4	88.2	86.0	85.3	85.0
ADV	90.8	77.9	79.2	76.1	74.9	74.3
PGM	93.8	86.3	86.1	83.3	82.5	82.0
Ours	96.4	92.3	93.3	91.3	90.7	90.4
+ Eff. Inl. Post	96.3	92.3	93.4	91.3	90.7	90.5
Runtime [s]						
FSM Aff	1056.3	1939.3	1787.9	2320.6	2520.9	2708.6
+ Eff. Inl. Eval	17921.0	25481.0	24010.0	26954.8	27758.7	28482.6
+ Eff. Inl. Post	1059.4	1947.3	1794.0	2334.1	2542.2	2742.2
FSM-R Aff	570.0	1146.6	1051.3	1306.0	1421.4	1490.0
+ Eff. Inl. Eval	13560.9	18038.3	17125.0	18657.7	19023.4	19352.0
+ Eff. Inl. Post	576.1	1156.2	1060.3	1323.2	1436.2	1522.0
FSM Sim	1040.9	1918.5	1766.6	2364.3	2527.8	2694.6
+ Eff. Inl. Eval	18097.0	25675.6	24188.3	27230.4	28026.4	28741.6
+ Eff. Inl. Post	1046.1	1928.3	1775.6	2385.7	2546.2	2723.1
FSM-R Sim	547.2	1100.0	1006.0	1279.3	1373.2	1435.0
+ Eff. Inl. Eval	13576.6	17941.0	17041.0	18595.3	18944.4	19263.3
+ Eff. Inl. Post	554.2	1110.1	1015.6	1296.7	1389.9	1464.1
HPM	**50.0**	**88.4**	**83.2**	**126.8**	**130.9**	**136.9**
ADV	101.5	150.4	139.3	172.3	175.8	183.9
PGM	90.3	118.0	112.8	**131.6**	**135.8**	**140.6**
Ours	**51.8**	**94.2**	**93.8**	**121.3**	**149.4**	**126.0**
+ Eff. Inl. Post	**57.2**	**106.4**	**107.7**	143.4	172.7	147.4

Table 4.10: Verification accuracy and efficiency measured on our new dataset using a vocabulary of 20K visual words. The best, second-best, and third-best results are highlighted for each column.

consisting of images taken close to Oxford would be likely to contain images similar to the query. However, since there are fewer cities in this area, it turns out that those (typically non-urban) images are more easily discarded during spatial verification, compared to image sets targeted toward city centers. At the same time, the UK distractor set proved harder than the Europe and US sets, despite its smaller size. The Paris6k dataset had a similar drop in accuracy using our targeted distractor sets, compared to the Flickr100k distractor set.

4.4 Summary

In this chapter, we presented a novel method for fast spatial verification in image retrieval. Our method is a hybrid of voting-based approaches for efficient hypotheses generation and inlier counting-based methods for accurate hypothesis verification. Comprehensive experiments on large-scale and unstructured image collections demonstrate high robustness to the choice of parameters. Our method achieves superior performance in balancing precision and efficiency versus the state of the art. The better performance in image retrieval directly translates to a better performance of the overall image-based 3D modeling pipeline. Furthermore, we studied the impact of distractor image distribution, which is especially relevant for 3D modeling from unstructured Internet imagery. To facilitate future research, we introduced a new unstructured query image set, which was released to the public alongside the source code of our method.

5 Pairwise Image Geometry Encoding

In the previous chapter, we proposed an approach that improves the efficiency of geometric verification in image retrieval. This and the next chapter focus on the goal of avoiding to spend time in the geometric verification stage altogether. The underlying idea is to efficiently decide on whether an image pair has scene overlap or not and then to only perform geometric verification for those image pairs that are likely to overlap. In contrast to purely appearance-based image retrieval techniques, the proposed approach in this and the next chapter integrates approximate geometric information into the decision process. By jointly considering appearance and geometry, we achieve superior accuracy in detecting scene overlap without sacrificing efficiency.

Over the last years, large-scale image-based 3D modeling has seen tremendous evolution in terms of efficiency in all stages (see Figure 1.1 and Figure 5.2) of processing [4, 75, 89, 129, 288, 311, 362, 365]. Generally, major computational effort in an image-based 3D modeling pipeline is spent on feature matching and geometric verification. In these two stages, it is essential to discover a sufficient number of image correspondences that link together all parts of the scene to obtain complete and large-scale reconstructions. In addition, robust and accurate alignment is aided by finding multiple redundant image-to-image connections across the entire scene. However, exhaustively searching for these overlapping pairs is infeasible for large-scale image collections due to quadratic computational complexity in the number of images and features. Moreover, as the number of registered images grows, the scalability of bundle adjustment algorithms becomes a significant performance bottleneck.

This chapter evaluates different techniques for reducing the cost of correspondence search, in particular feature matching and geometric verification. Usually, the major-

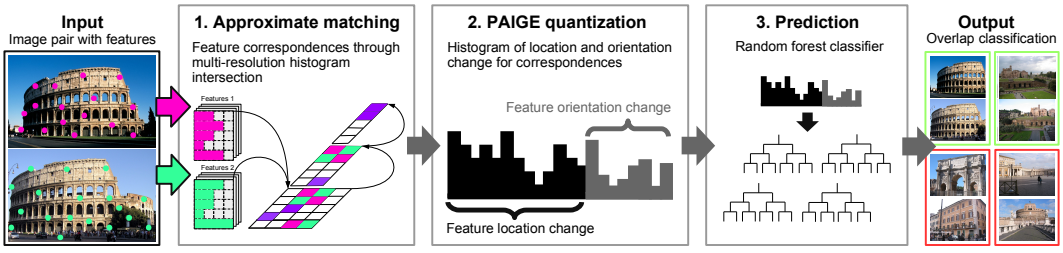

Figure 5.1: The proposed framework for extracting our proposed PAIGE feature, and its application for efficient scene overlap prediction in Structure-from-Motion.

ity of image pairs in unordered Internet photo-collections do not have scene overlap, so rejecting those pairs dominates execution time, even though such pairs are not useful for 3D reconstruction. Consequently, various approaches have been proposed to efficiently find overlapping pairs in noisy datasets and only forward those pairs to Stages 2 and 3. A downside of sending fewer image pairs to Stages 2 and 3 is that enough images with overlapping geometry must be processed to produce accurate camera alignment and complete reconstructions. Hence, it is essential to find the right trade-off between computational efficiency and sufficient image connectivity.

Despite the impressive progress in reducing the cost of the matching, relatively little attention has been paid in comparing the techniques. The goals of this chapter are therefore twofold: First, we present a comprehensive analysis and evaluation of various state-of-the-art matching techniques; second, we use the insights gained from this evaluation to propose the PAirwise Image Geometry Encoding (PAIGE) to build a scalable framework (see Figure 5.1) for the efficient recognition of the relative viewing geometry, all without explicit feature matching and without reconstructing the actual camera configuration using geometric verification. The proposed encoding is based on the geometric shape properties of local features, which are efficiently inferred from approximate feature correspondences. A subsequent classification strategy leverages the encoding to only perform matching and geometric verification for image pairs that are identified as overlapping. As demonstrated in comprehensive experiments, this novel approach leads to a further speedup of large-scale image-based 3D modeling as compared to the existing state of the art.

5.1 Related Work

Large-scale image-based 3D modeling systems have tremendously advanced in terms of increased robustness and reduced runtime. A variety of methods have been proposed to improve the efficiency in different stages of image-based 3D modeling pipelines (see Figure 5.2).

5.1.1 Feature Extraction

While SIFT [200] is a popular choice for robust feature detection and description, the slightly more efficient SURF features are a commonly used alternative [26]. In addition, a number of binary features have also been proposed [9, 186, 262]. These binary features lead to a significant speedup of the extraction and the subsequent matching stage as well as a reduced memory footprint.

5.1.2 Feature Matching

Various methods have been proposed to reduce the number of image pairs considered in the matching module. Frahm et al. [89] leverage iconic image selection through GIST clustering to find similar images. Agarwal et al. [4] employ image retrieval systems [231] to only match against similar images and then use approximate nearest

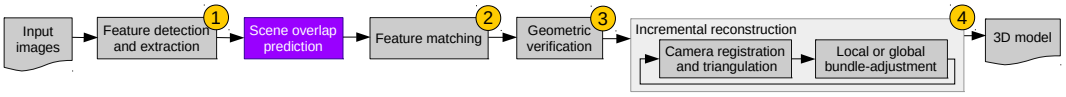

Figure 5.2: The proposed prediction framework (purple) integrated into a typical reconstruction pipeline.

neighbor feature matching. Furthermore, Chum et al. advance in the field of efficient image retrieval [64] and improve retrieval results with a randomized data mining method [67]. Another improvement to retrieval systems was developed by Chao et al. [52], who employ an online learning strategy to rerank retrieval results. Krapac et al. [171] and Jégou et al. [153] encode spatial information of features in bag-of-words models as used in retrieval systems. Orthogonally, Raguram et al. [258] use GPS tags to match images only to spatially nearby ones. Wu [365] follows a preemptive matching strategy by filtering image pairs that fail to match on a reduced feature set. Beyond that, Lou et al. [198] develop a scalable method to find connected components in large datasets. Most recently, Hartmann et al. [125] propose to predict the matchability of individual features to reduce the number of feature comparisons during feature matching; Havlena et al. [126] inspired by [243, 318] directly use the assignments of individual features to visual words in a vocabulary tree as verified correspondences as an input for the reconstruction stage, skipping the pairwise image matching stage altogether.

5.1.3 Geometric Verification

Apart from the advancements in fast essential matrix estimation [230], a number of efficient RANSAC [85] variants have been developed [68, 208, 243, 254]. Furthermore, other approaches propose to improve the efficiency of geometric verification through voting [19, 267, 306, 368] an online learning approach [257].

5.1.4 Sparse Reconstruction

Snavely et al. [311] compute skeletal subsets of images to reduce the runtime of incremental reconstruction, whereas Agarwal et al. [6] and Wu et al. [366] progress in the field of bundle adjustment by developing efficient and scalable algorithms for multi-core machines. Complementary to the efforts in incremental SFM, Gherardi et al. [108] propose a hierarchical SFM pipeline with balanced branching and merging. Sinha et al. [305] compute two-view reconstructions from vanishing points followed by efficient 3D model merging, while Crandall et al. [75] describe a replacement for traditional incremental reconstruction systems by finding a coarse initial solution for bundle adjustment using a discrete-continuous optimization approach based on GPS initializations. Recently, Wilson et al. [362] propose to estimate camera translations by solving simplified lower-dimensional problems with epipolar geometry averaging.

For the comparative evaluation of matching techniques in this work, we choose one popular representative of each family of approaches. The above described approaches

can be categorized into three different families of approaches. The first family, approximate matching techniques, describe images as a whole and avoid exhaustive pairwise image matching [4, 52, 64, 67, 89, 198]. The second family, exhaustive matching techniques, try to either preemptively filter image pairs [365] or reduce the cost of feature matching [125]. The third family consists of approaches that try to avoid pairwise matching and verification altogether [126]. We rely on publicly available implementations of the methods; implementations that have already been successfully applied in large-scale 3D reconstruction. In the following, we briefly describe the chosen approaches; an evaluation of their performance on several large-scale datasets (see Table 5.1) is given in Section 5.2.

5.1.5 Image Retrieval

Image retrieval has been extensively employed in large-scale image-based 3D modeling [4, 89, 198]. Hence, we use it as a representative of the first family. Image retrieval is often efficiently implemented using vocabulary trees [15, 64, 149, 150, 217, 231, 335], an instance of bag-of-words (BOW) models, which try to describe images as a whole. Features are hierarchically quantized and indexed in the vocabulary tree. Similarity from indexed images to a query image is measured using, e.g., tf-idf, co-occurrence, or burstiness scoring. In large-scale image-based 3D modeling systems, vocabulary trees are leveraged to match every image only against a number of most similar images (approximate nearest neighbors), effectively eliminating the quadratic computational cost in the number of images of exhaustive pairwise matching. The number of retrieved images is determined by retrieving a fixed number of images N_R and/or thresholding the similarity score. However, BOW similarities are noisy, due to faulty quantization and feature detection. As a consequence, it is difficult to find good similarity thresholds, which is why, in our analysis, we retrieve a fixed number of nearest neighbors per query image. We employ the implementation of Agarwal et al. [4], a tf-idf weighted vocabulary tree using min-distance metric and 1M visual words (branching factor 10, depth 5) trained from approximately 100M features (unrelated to evaluation datasets). We denote this method as *Retrieval* N_R.

5.1.6 Preemptive Matching

Preemptive matching is a representative of the exhaustive matching techniques, follows the idea that matching a small subset of the features is effective in determining whether an image pair has overlap. The method assumes that features detected at higher scales are more repeatable and stable across images; hence, if a small number N_P of L_P higher scale features match, the image pair is said to have overlap. Full putative feature matching is only performed for those pairs that pass this preemptive filtering stage. On the one hand, this strategy theoretically allows us to find all possible image pairs. On the other hand, it still has quadratic computational complexity in the number of features and images. However, the work for individual image pairs dramatically decreases (e.g. by a factor of 10,000 when $L_P = 100$), since

feature matching is itself quadratic in the number of features. We use the implementation of Wu [365] and denote it as *Preemptive* N_P, setting $L_P = 100$, as suggested by the author.

5.1.7 Vocabulary Matching

Vocabulary matching is a representative of the third family, skips the pairwise image and feature matching stages altogether by using the indexing of multiple features to the same visual word in a precomputed vocabulary tree as implicit matches. Feature matches between image pairs are then generated by the pairwise combination of all assigned features per visual word. A symmetric clustering matrix is used to find connected components in an image collection. To avoid ambiguous matches, only one visual word may appear in each image. For reasons of efficiency and to reflect the importance of a visual word with respect to frequency of its occurrence (similar to the motivation of tf-idf weighting), the method discards visual words that appear in too many images (the authors propose a threshold of 1%). Given a sufficiently large visual vocabulary, correspondences from assignments of features to visual words are stronger than from pairwise putative matching. Note that this method requires significantly more visual words than in standard vocabulary trees in order to achieve good performance. In addition, the proposed approach is infeasible for very large image collections with millions of images, since the clustering matrix cannot be stored in memory, as noted by the authors. We use the implementation and visual vocabulary provided by Havlena et al. [126], and denote the method as *VocMatch*.

5.2 Evaluation

In this section, we evaluate the previously described approaches on different large-scale datasets (see Table 5.1). We propose to formulate the problem of finding overlapping image pairs as a classification problem, where we try to learn a model that separates image pairs with scene overlap (positive) from image pairs without scene overlap (negative). The objective of an optimal method is to minimize the ratio of false over true positives (overhead), whereas the true positives should comprise all relevant image pairs of the dataset. Since in Internet photo collections typically only a small fraction of the images are relevant and therefore an even smaller fraction of image pairs actually match, the effective runtime of a method is determined by the overhead. Hence, the goal of an optimal matching strategy is to produce minimal overhead while finding all true positives. In the end, the effective utility of a matching method for image-based 3D modeling is related to the completeness and stability of the resulting reconstructions.

The evaluation datasets comprise five crowd-sourced image collections (London Eye, San Marco, Tate Modern, Time Square, and Trafalgar) [52], and a well-studied dataset of Rome [193]. These collections contain a diverse set of viewpoints, rather than a single dominant one. The first five datasets are contaminated with a large number of irrelevant images that do not match to the actual landmarks. Contrary,

	Images	Pairs	Verified pairs
London Eye	7,047	24,826,581	319,591 (1.29%)
San Marco	7,792	30,353,736	237,130 (0.78%)
Tate Modern	4,813	11,580,078	119,483 (1.03%)
Time Square	6,426	20,643,525	140,193 (0.68%)
Trafalgar	6,981	24,363,690	285,022 (1.17%)
Rome	16,179	130,871,931	– (–)

Table 5.1: Evaluation datasets.

the Rome dataset only consists of relevant images, which should register to at least one landmark. For all experiments, we use SIFT features (Hessian-Affine [239] for *VocMatch*, Difference of Gaussian for all other methods). We consider an image pair as geometrically verified (i.e. it has scene overlap) if the putative SIFT matches (max. distance ratio of 0.8 between top two matches, max. cosine distance of 0.7, and mutual best matching) have at least 20 inliers in essential matrix estimation with RANSAC (4px Sampson error threshold). As a baseline approach, we exhaustively compute the ground-truth image pairs, with N_G denoting the number of verified pairs. The performance of each method is quantified in several measures obtained from the confusion matrix (N_{TP}: true positives, N_{FP}: false positives). First, we measure how many of the ground-truth image pairs are found (N_{TP}/N_G). Second, we measure the overhead of finding these pairs (N_{FP}/N_G). Third, we measure the required time by isolating the runtime of the respective method including the subsequent feature matching and geometric verification stages. For the matching procedure, we use an optimized GPU implementation, and for geometric verification a multi-threaded RANSAC CPU implementation. To quantify the impact of the reduction of each method, we measure the completeness of 3D reconstruction in terms of the total number of registered cameras. All experiments were performed on the same machine with 2x12 physical cores, 256GB RAM, and a NVIDIA GeForce GTX TITAN Z graphics card. I/O overhead is excluded from the timing for all methods.

The results of the experiments are summarized in Table 5.2 and Figure 5.8. All methods significantly reduce the runtime and number of evaluated pairs compared to exhaustive matching. But they also produce a significant number of false negatives, i.e. they eliminate correct image pairs. Nevertheless, the reconstruction system is still able to produce quality reconstructions with the number of registered images being related to the number of verified image pairs. In this regard, we also observe that the number of registered images and, qualitatively, the stability of the reconstructed models saturates at some point. In other words, image-based 3D reconstruction does not substantially gain from finding all true positives. In the following, we briefly discuss the individual results of each approach.

5.2.1 Retrieval

As can be seen from Table 5.2, vocabulary trees work relatively well in terms of precision, when only retrieving a few nearest neighbors. However, when more images are retrieved, such methods tend to yield many false positives, resulting in a large computational overhead in matching. Otherwise, in case only a few images are retrieved, the overhead of indexing and querying images in the vocabulary tree becomes more relevant to the overall runtime. While theoretically possible [4], it is comparatively challenging to efficiently scale the indexing and querying of a vocabulary tree across distributed machines for large-scale datasets.

5.2.2 Preemptive

Due to quadratic feature matching cost, we must limit N_P to a low number for reasons of efficiency. Consequently, the threshold L_P must be chosen very low ($L_P = 4, N_P = 100$ as proposed by Wu [365]) to find relevant pairs. Thus, a small change in L_P has great effect on the performance of this filtering strategy – both in terms of efficiency and precision (compare *Preemptive 3* and *Preemptive 4*). Moreover, a small subset of the features may not adequately represent the entire image, resulting in a noisy classifier. Beyond that, image pairs with small overlap (e.g., due to large scale change or different viewpoint centers) will likely fail to pass the filtering, because of the the low number of features, which may be spatially distributed across the entire image. This method can be relatively easily scaled across multiple cores and distributed machines.

5.2.3 VocMatch

While this method drastically speeds up the computation of pairwise matching, it also makes some assumptions about the underlying structure of the image collection. Since the method discards highly frequent visual words, image collections of popular landmarks with many redundant viewpoints may produce less stable reconstructions due to the lack of long feature tracks. Moreover, the length of feature tracks depends on the relation of the number of features in the dataset and the codebook size of the vocabulary tree. We find that the track lengths of 3D points during reconstruction are significantly shorter for *VocMatch* than for the other methods, i.e. the estimated point locations are more uncertain. Setting the maximum frequency of visual words and using the right codebook size is difficult because there is usually no a priori knowledge about the distribution of images in crowd-sourced datasets. We can observe the impacts of the a priori assumptions by considering the high variance of the performance across the different datasets.

Summarizing the above evaluation, we conclude, that there is no need to find all true positive image pairs to produce good reconstructions in terms of stability and completeness. In fact, we only need a comparatively small fraction of the ground-truth image pairs. While the true positive rate accounts for some of the runtime, the methods' overall runtimes are mostly determined by the overhead, since RANSAC

is especially expensive for false positive pairs. Moreover, scalability becomes especially important for large-scale datasets. None of the existing approaches provides sufficient recall with low overhead and fast runtime to produce quick reconstructions for large datasets. Considering this analysis, we propose an efficient and scalable learning-based approach to preemptively predict the geometric relation between an image pair with low overhead (i.e. low false positive rate). To represent the image with a larger subset of features, we develop a hierarchical, approximate matching scheme that reduces the quadratic to amortized linear complexity. Based on the resulting implicit feature correspondences, we compute the PAirwise Image Geometry Encoding (PAIGE). A subsequent classification procedure leverages the encoding to predict whether an image pair has overlap or not. Finally, standard putative matching and geometric verification is only performed for image pairs that are predicted to overlap.

5.3 Pairwise Image Geometry

In this section, we develop PAIGE, a new approach for quickly predicting whether two images have scene overlap. Toward this goal, we begin by analyzing how pairwise image geometry relates to the pattern of correspondences between feature points in two images (see Section 5.3.1). To use this intuition in a fast approach, we first hash all the features in an image into a fixed-sized data structure (see Section 5.3.3), and then compute an approximate descriptor of the pattern between corresponding feature points (see Section 5.3.4). The PAIGE approach works by learning to predict scene overlap from this representation. The whole process, from hashing the descriptor of approximate correspondences to evaluating the classifier, is linear in the number of features per image (see Section 5.3.5), and is relatively light-weight in terms of actual computation. The hashing process preserves enough information about the geometry between pairs of images to allow PAIGE to produce accurate and fast predictions about whether two images should be sent onward to the computationally more expensive putative matching stage.

5.3.1 Feature Correspondence and Pairwise Geometry

We define pairwise image geometry as the relative motion between two images. The relative motion between an image pair can be determined up to unknown scale by estimating the essential matrix [123] for a freely moving and by a homography for a purely rotating camera. Hence, an image pair has scene overlap, if we can estimate its relative motion from corresponding feature points (see Section 2.2.4).

Traditional reconstruction systems require the extraction of sparse image features, preferably invariant under radiometric and geometric transformations. Current practice uses local features that estimate four properties [127]: location $\{\bar{x}, \bar{y}\}$, orientation o, scale s, and the descriptor $\boldsymbol{f}$.

The central observation underlying the PAIGE approach is, that when images of the same structure are taken from different viewpoints, corresponding features

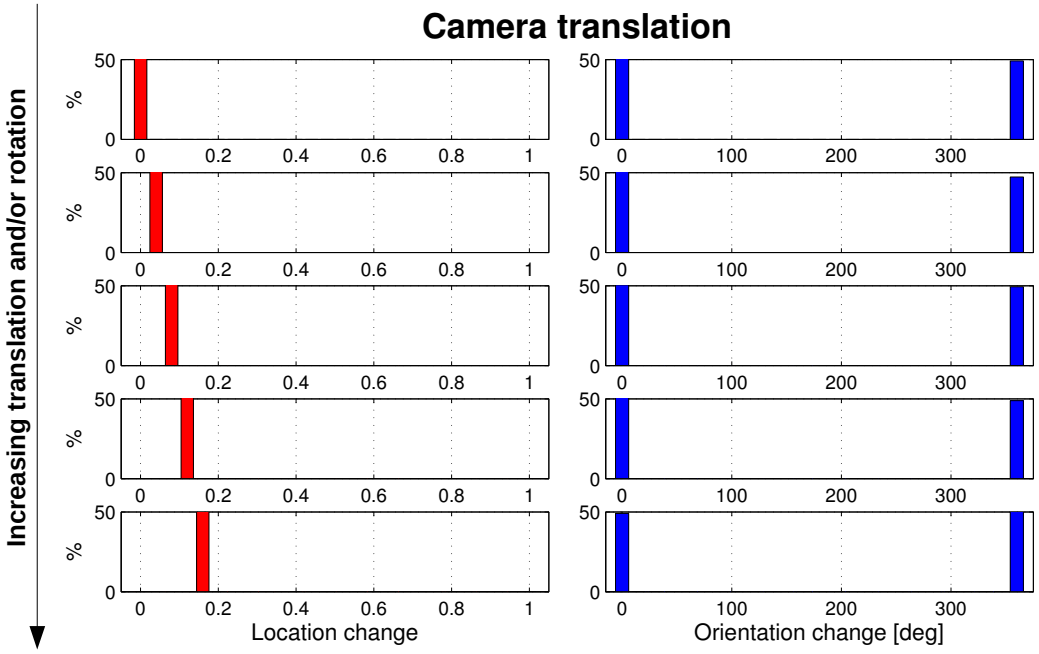

Figure 5.3: SIFT feature location and orientation change histograms for pure translational sidewards camera motion.

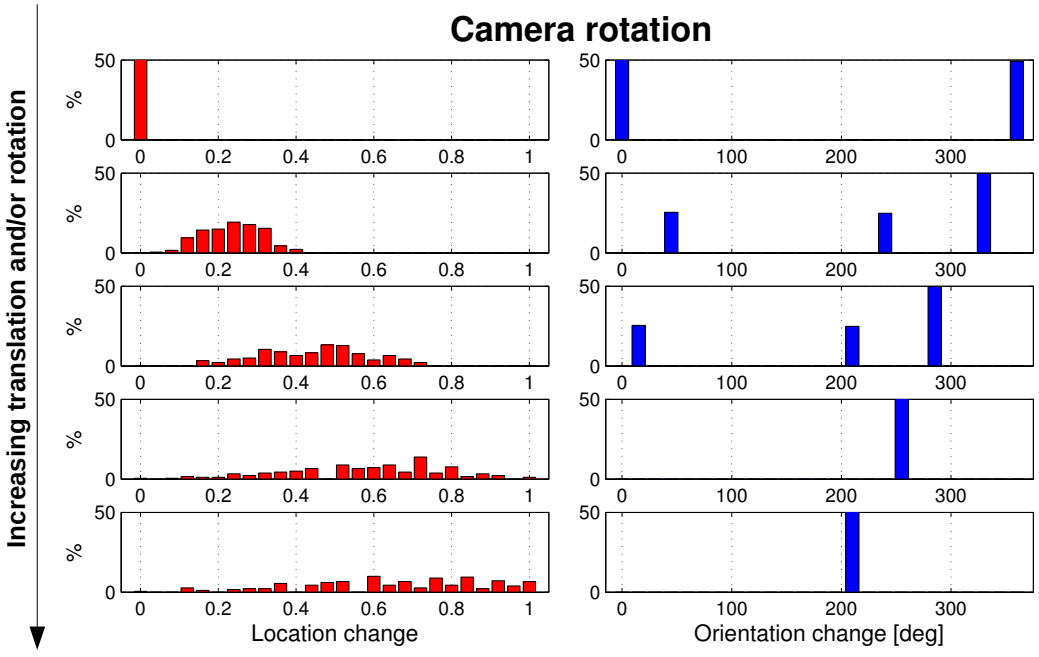

Figure 5.4: SIFT feature location and orientation change histograms for pure rotational camera motion in $[0°; 150°]$ around the viewing direction.

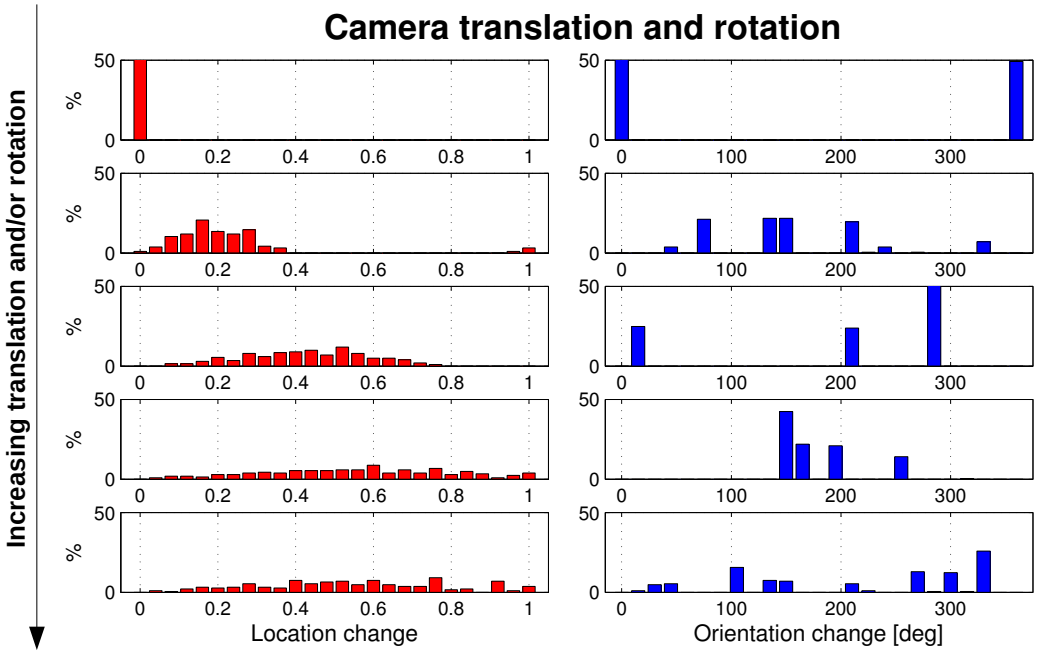

Figure 5.5: SIFT feature location and orientation change histograms for joint translational and rotational camera motion.

change in their geometric shape in recognizable patterns. Figure 5.3, Figure 5.4, and Figure 5.5 visualize patterns in the changes between features in a synthetic experiment, demonstrating the relation of pairwise geometry and the properties of corresponding features.

To produce Figure 5.3, Figure 5.4, and Figure 5.5, we find feature correspondences for a pair of rendered images of a reference pattern with 256 feature points, with the first camera held stationary and the viewpoint of the second camera increasingly transformed. For this image pair, we calculate the displacement for each feature using normalized image coordinates, such that $\{x, y\} \in [0, 1]$ (to handle zoom), and measure rotation of features in degrees. Next, histograms quantize the distribution of these two measures of feature transformation. We observe, that the location change histogram (see Figure 5.3) is sufficient to recognize purely translational camera motion. However, the location change alone does not distinguish between purely rotational motion (see Figure 5.4) and a combination of translational and rotational camera motion (see Figure 5.5). Considering histograms of both feature location and orientation change allows us to separate the two cases.

Based on this motivation, the following sub-sections describe how to efficiently find approximate feature correspondences and then to leverage estimates of the corresponding location and orientation changes to predict whether an image pair has scene overlap.

5.3.2 Approximate Feature Transformations

Computing the histograms shown in Figure 5.3 used knowledge of the exact feature correspondences between a pair of images. Our approach approximates this correspondence. Conceptually, we make two levels of relaxations to perform this approximation. First, instead of the exact correspondence, we can consider the motion (translation and rotation) between a feature in one image and any very similar features in the other image. As the next relaxation, we can consider individual dimensions of a feature descriptor. Each time two descriptors match in one dimension, we use their translation and rotation to increment the appropriate bins of the translation and rotation histograms. This later relaxation seems as if it would introduce a large number of spurious increments to the histograms, but because non-matching features rarely agree on many dimensions, unlike closely matching features, the noisy additions are spread out. In practice, this representation works well, as shown in the experiments (see Section 5.6). Furthermore, this approach allows computing the histogram of translations and rotations from approximate correspondences to be done in two stages: 1) Hashing all of the features in an image into one fixed-size data structure. 2) Using two of these data structures to compute an approximate histogram of translations and rotations between features in two images.

5.3.3 Hashing the Features from A Single Image

Consider a collection of images

$$\boldsymbol{X} = \{\boldsymbol{F}_1, \ldots, \boldsymbol{F}_m\} \ , \tag{5.1}$$

where each image is represented by its set of local features

$$\boldsymbol{F}_i = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{n_i}\} \tag{5.2}$$

of cardinality n_i of d-dimensional feature descriptors $\boldsymbol{f}_j \in \mathbb{R}^d$. For simplicity, we assume that all feature descriptors f_k in each $\boldsymbol{f}_j$ are non-negative, and that $\left\|\boldsymbol{f}_j\right\| = 1$. We quantize a given $\boldsymbol{F}$ into

$$\mathcal{F}(\boldsymbol{F}) = [\boldsymbol{H}_0(\boldsymbol{F}), \boldsymbol{H}_1(\boldsymbol{F}), ..., \boldsymbol{H}_{r-1}(\boldsymbol{F})] \tag{5.3}$$

as a concatenation of r differently weighted 2-dimensional multi-resolution histograms $\boldsymbol{H}_i(\boldsymbol{F}) \in \boldsymbol{M}_{d \times b_i}(\mathbb{R})$. Each $\boldsymbol{H}_i$ has 1-dimensional histograms with b_i bins for each of the d dimensions of the feature vectors $\boldsymbol{f}_j$, hence is $d \times b_i$-dimensional. The 1-dimensional histograms span the space $f_k \in [0, 1]$ using $b_i = 2^i$ bins of width $\Delta b = 2^{-i}$. To populate the histograms, each $\boldsymbol{f}_j \in \boldsymbol{F}_i$ contributes its assigned weight η (which varies depending on the task, see below) once to each of the d locations in each $\boldsymbol{H}_i$. Overall, the hashed descriptor, $\mathcal{F}(\boldsymbol{F}_i)$, for a set of features $\boldsymbol{F}_i$ from an image, has dimension $d \sum_{i=0}^{r-1} 2^i$ that does not depend on the number of feature descriptors, n_i, for the image.

This representation can be leveraged to establish approximate correspondences between two entities $\boldsymbol{F}_a$ and $\boldsymbol{F}_b$ by intersecting their respective $\mathcal{F}(\boldsymbol{F}_a)$ and $\mathcal{F}(\boldsymbol{F}_b)$.

The more similar two features $\boldsymbol{f}_a \in \boldsymbol{F}_a$ and $\boldsymbol{f}_b \in \boldsymbol{F}_b$ are, the more they will contribute to corresponding bins in $\mathcal{F}(\boldsymbol{F}_a)$ and $\mathcal{F}(\boldsymbol{F}_b)$.

This approach is potentially prone to over-estimating the number of correspondences, since it finds matches separately in all marginals of $\boldsymbol{f}$, which might result in duplicate and false matches. However, if a pair of feature vectors $\boldsymbol{f}$ are very close (e.g., for a true correspondence), they will agree in more dimensions than dissimilar features. As the dimension of the descriptors increases this effect becomes stronger. Section 5.3.4 explains, how we account for the differing similarity across levels by weighting the relevance of correspondences based on the resolution of the feature histogram. The approximate matching scheme can naturally deal with sets of unequal cardinalities, since a feature in the smaller set is implicitly matched to multiple features in the larger set.

The proposed scheme borrows ideas from the pyramid match approach [112], but differs in a fundamental way. The pyramid match approach treats the descriptor vector as a whole, and in practice it is therefore often implemented using a sparse histogram. In our approach, we hash each dimension of the descriptor separately, resulting in a more efficient, fixed-size histogram implementation. Moreover, the traditional pyramid match approach intersects the raw counts of overlapping features, while we use weighted histograms. In the next section, we will see how to use this weighted matching scheme to encode the geometric properties in the PAIGE descriptor.

5.3.4 Feature Quantization

The PAirwise Image Geometry Encoding (PAIGE) is defined as the function

$$\mathcal{P}\colon \boldsymbol{X} \times \boldsymbol{X} \rightarrow \mathbb{R}^{d_\mathcal{P}} \tag{5.4}$$

and quantifies the distribution of location and orientation changes between an image pair $(\boldsymbol{F}_a, \boldsymbol{F}_b)$ based on its feature correspondences $(\boldsymbol{f}_a \in \boldsymbol{F}_a, \boldsymbol{f}_b \in \boldsymbol{F}_b)$. The approximate matching scheme described in Section 5.3.3 is used to implicitly establish these correspondences. Therefore, we compute separate multi-level histograms $\{\mathcal{F}_x, \mathcal{F}_y, \mathcal{F}_o, \mathcal{F}_1\}$ in a computationally efficient manner for the respective cases $\eta \in \{x, y, o, 1\}$. In other words, we quantize the geometric information of a single image in separate histograms, and count the number of elements (the features) per bin with $\eta = 1$. The image locations are normalized using the dimensions of the image, such that

$$\Delta x \in [-1, 1] \tag{5.5}$$

$$\Delta y \in [-1, 1] \tag{5.6}$$

$$\Delta o \in [-2\pi, 2\pi] \quad . \tag{5.7}$$

In the next step, we average the location and orientation histograms to account for the fact that multiple features might populate the same bin

$$\overline{\mathcal{F}} = \left[\frac{\mathcal{F}_x}{\mathcal{F}_1}, \frac{\mathcal{F}_y}{\mathcal{F}_1}, \frac{\mathcal{F}_o}{\mathcal{F}_1}\right] \tag{5.8}$$

For all pairwise combinations of images $(\boldsymbol{F}_a, \boldsymbol{F}_b)$ in $\boldsymbol{X}$, we can thereby efficiently calculate the approximate change in location and orientation per marginal bin as

$$\Delta\overline{\mathcal{F}} = \overline{\mathcal{F}}_a - \overline{\mathcal{F}}_b \tag{5.9}$$

We describe the distribution of these location and orientation changes using the PAIGE feature, which is defined as a concatenation of uniformly spaced, weighted histograms

$$\boldsymbol{h}\left(\boldsymbol{F}_a, \boldsymbol{F}_b\right) = [\boldsymbol{h}_{\Delta x}, \boldsymbol{h}_{\Delta y}, \boldsymbol{h}_{\Delta o}] \tag{5.10}$$

populated from $\Delta\overline{\mathcal{F}}$. The dimensionality of PAIGE is

$$d_{\mathcal{P}} = d_{\Delta x} + d_{\Delta y} + d_{\Delta o} \tag{5.11}$$

since $\boldsymbol{h}_{\Delta x} \in \mathbb{R}^{d_{\Delta x}}$, $\boldsymbol{h}_{\Delta y} \in \mathbb{R}^{d_{\Delta y}}$, $\boldsymbol{h}_{\Delta o} \in \mathbb{R}^{d_{\Delta o}}$. Finally, the encoding is normalized to achieve invariance with respect to the number of feature correspondences

$$\mathcal{P}\left(\boldsymbol{F}_a, \boldsymbol{F}_b\right) = \frac{\boldsymbol{h}\left(\boldsymbol{F}_a, \boldsymbol{F}_b\right)}{\left\|\boldsymbol{h}\left(\boldsymbol{F}_a, \boldsymbol{F}_b\right)\right\|} \tag{5.12}$$

The weight ω a populated bin in $\mathcal{F}$ contributes to PAIGE depends on the similarity of the approximate correspondences. As shown in Section 5.3.3 the similarity of a match is dependent on the resolution and thus the level i of the histogram $\boldsymbol{H}_i$ in $\mathcal{F}$. Hence, we choose the weight as $\omega_i = 2^{i-r}$. PAIGE is naturally robust against mismatches, since it is dominated by fine-grained correspondences in the higher-resolution histogram levels. Additionally, it is able to capture the overall location and orientation changes through the correspondences in the coarser histogram levels. PAIGE is intentionally designed as a non-symmetric function, i.e. $\mathcal{P}\left(\boldsymbol{F}_a, \boldsymbol{F}_b\right) \neq \mathcal{P}\left(\boldsymbol{F}_b, \boldsymbol{F}_a\right)$, since this allows us to encode the direction of relative camera motion.

5.3.5 Computational Efficiency

The described matching approach enables us to find approximate feature correspondences without performing exhaustive pairwise feature matching, which is quadratic in the number of features $O(n^2)$. More precisely, the population of $\mathcal{F}$ is $O(drn)$, since the d marginals of $\boldsymbol{f}$ contribute to a maximum of r histograms. The normalization step and the PAIGE quantization are performed for each element in $\mathcal{F}$, and thus are $O(2^{r+1}d)$. Typically, it is $n \gg r$ and $n \gg d$. Hence, the amortized computational complexity of quantizing PAIGE is $O(n)$. Note that we hash every image in a collection in the fixed-sized data structure $\mathcal{F}$ independently, and then reuse it for the exhaustive pairwise computation of PAIGE to reduce the computational effort.

5.4 Classification

Based on the proposed PAIGE feature (see Equation 5.12), we next design a binary classifier to predict scene overlap. In doing so, we try to learn a model that separates

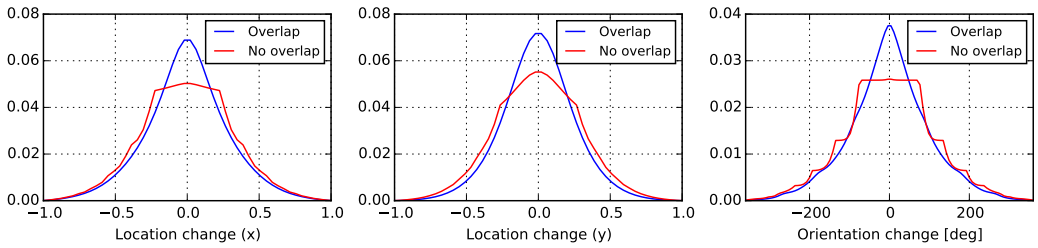

Figure 5.6: Average PAIGE feature for the London Eye dataset, separated into location and orientation parts.

image pairs with scene overlap (positive) from image pairs without scene overlap (negative). Choosing a suitable classifier depends on two main factors. First, the joint distribution of location and orientation change is expected to be complex over the complete space of possible pairwise image configurations. Hence, we need a classifier that is able to discriminate this complex parameter space. Second, the main motivation for the proposed method is a speed improvement over the traditional approach of exhaustive feature matching and geometric verification; therefore, the classifier should require minimal computational effort for maximal benefit. In our experiments, random forests [10, 40, 139] gave the best results in terms of accuracy and computational efficiency.

We use SIFT to extract invariant features at different scales. Note, any other invariant features could be employed alternatively. The 128-dimensional descriptors $\boldsymbol{f}$ are normalized and stored with 8-bit precision. We use $r = 9$ as the number of multi-resolution histograms; the number of bins of the finest-resolution histogram therefore equals the descriptor discretization. Empirically, the dimensionality of PAIGE is chosen as $d_{\Delta x} = d_{\Delta y} = 50$ and $d_{\Delta o} = 100$.

5.5 Training

Large-scale Internet photo-collections from several different landmarks across the world and a set of sequential image sequences acquired by mobile video cameras (to counter the orientation bias of crowd-sourced images) serve as the dataset for training the random forest classifier. Note, the training dataset is disjoint from the evaluation datasets. Ground-truth data is extracted by exhaustive pairwise image matching and subsequent geometric verification for approximately 30M image pairs. Then, PAIGE is extracted for all image pairs a and b in the forward $\mathcal{P}(\boldsymbol{F}_a, \boldsymbol{F}_b)$ and backward $\mathcal{P}(\boldsymbol{F}_b, \boldsymbol{F}_a)$ directions. Hence, for each image pair, we generate two training samples with the same label. Analogously, when we classify an image pair, we can extract the forward and backward PAIGE features with requiring only small additional computational overhead, since we need only invert the order of subtraction in Equation 5.9. We then classify both features and use the more confident prediction as the final classification result. Due to the fact that most image pairs in unordered

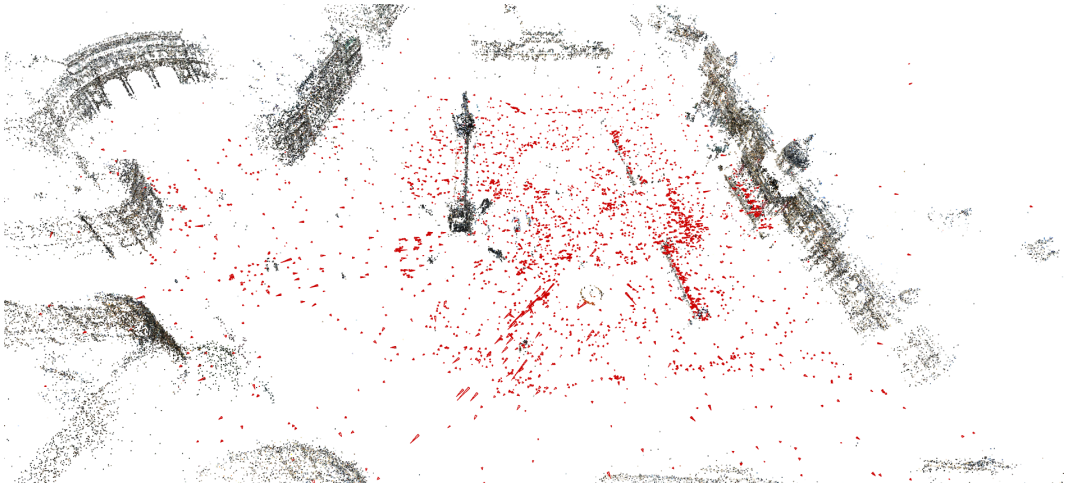

Figure 5.7: Reconstruction based on PAIGE for the Trafalgar dataset.

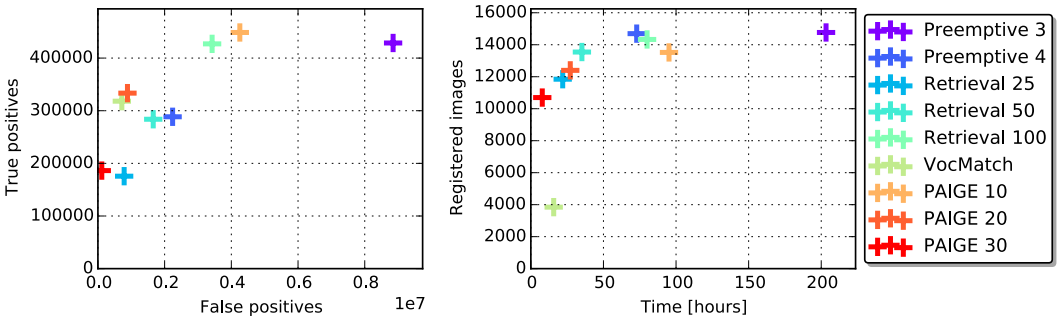

Figure 5.8: Visualization of the overall evaluation results in Table 5.2.

collections do not have scene overlap, we reduce (via random sub-sampling) the number of negative samples with the goal of training classifiers with differently tuned properties in terms of the expected overhead. We denote these versions as *PAIGE* $N_{\mathcal{P}}$, where $N_{\mathcal{P}}$ is the ratio of negative over true training samples. Using 3-fold cross-validation, we determined design choices for the classifier, including using a forest with 50 decision trees, entropy as the splitting criterion, and considering all features when searching for the best split at each node in a tree. A minimum number of three samples per leaf is enforced to avoid over-fitting.

5.6 Evaluation

We perform the same experiments for PAIGE as for the other methods. The PAIGE feature for overlapping vs. non-overlapping pairs is shown in Figure 5.6. The results in Table 5.2 show, that *PAIGE 30* has the lowest false positive rate of any method and hence the lowest overhead in reconstruction cost, while still reconstructing nearly

	Time	Prec.	Found	Overhead	Reg. images
Retrieval 25	**17h24m**	**0.22**	**0.13**	**3.59**	**11845**
London Eye	3h58m	0.29	0.14	2.40	2354
San Marco	3h41m	0.28	0.17	2.55	3392
Tate Modern	2h30m	0.18	0.12	4.63	1429
Time Square	3h16m	0.14	0.12	6.01	2014
Trafalgar	3h59m	0.17	0.11	4.85	2656
Rome	6h43m	–	–	–	15412
Retrieval 50	**28h39m**	**0.18**	**0.21**	**4.68**	**13544**
London Eye	5h25m	0.25	0.24	2.99	3280
San Marco	5h48m	0.23	0.28	3.41	3703
Tate Modern	3h28m	0.14	0.20	6.01	1481
Time Square	6h57m	0.11	0.18	8.34	2169
Trafalgar	7h0m	0.14	0.18	6.40	2911
Rome	9h4m	–	–	–	15366
Retrieval 100	**61h34m**	**0.14**	**0.32**	**6.39**	**14531**
London Eye	11h55m	0.20	0.37	3.95	3331
San Marco	17h7m	0.17	0.41	4.87	3929
Tate Modern	6h32m	0.11	0.31	8.01	1647
Time Square	16h6m	0.08	0.26	11.86	2463
Trafalgar	9h54m	0.10	0.26	8.75	3161
Rome	17h41m	–	–	–	15388
Preemptive 3	**164h41m**	**0.05**	**0.32**	**17.59**	**14767**
London Eye	38h6m	0.08	0.36	11.14	3418
San Marco	42h6m	0.06	0.34	16.24	3815
Tate Modern	14h33m	0.07	0.35	13.56	1825
Time Square	34h46m	0.03	0.24	34.53	2401
Trafalgar	35h10m	0.04	0.38	22.00	3308
Rome	223h12m	–	–	–	15401
Preemptive 4	**58h26m**	**0.13**	**0.21**	**6.48**	**14694**
London Eye	13h32m	0.21	0.25	3.83	3477
San Marco	13h58m	0.16	0.23	5.23	3744
Tate Modern	6h28m	0.18	0.23	4.61	1930
Time Square	10h35m	0.06	0.13	14.84	2351
Trafalgar	13h53m	0.09	0.25	9.57	3192
Rome	80h43m	–	–	–	15298
VocMatch	**11h15m**	**0.32**	**0.24**	**2.17**	**4247**
London Eye	2h15m	0.30	0.43	2.35	1353
San Marco	2h52m	0.29	0.37	2.47	1474
Tate Modern	1h21m	0.39	0.17	1.59	637
Time Square	2h46m	0.28	0.05	2.60	316
Trafalgar	2h1m	0.76	0.10	0.32	467
Rome	7h48m	–	–	–	12944
PAIGE 10	**84h31m**	**0.11**	**0.36**	**8.47**	**13520**
London Eye	18h0m	0.13	0.35	6.70	3167
San Marco	20h25m	0.10	0.36	9.25	3544
Tate Modern	9h54m	0.14	0.45	6.17	1490
Time Square	4h31m	0.23	0.28	3.38	2193
Trafalgar	31h42m	0.07	0.48	13.07	3126
Rome	83h55m	–	–	–	15298
PAIGE 20	**24h15m**	**0.29**	**0.27**	**2.44**	**12198**
London Eye	6h28m	0.30	0.28	2.33	2905
San Marco	4h22m	0.35	0.26	1.83	3145
Tate Modern	3h42m	0.30	0.35	2.37	1338
Time Square	1h8m	0.92	0.23	0.09	1999
Trafalgar	8h36m	0.18	0.32	4.41	2811
Rome	22h49m	–	–	–	14725
PAIGE 30	**7h18m**	**0.63**	**0.15**	**0.59**	**10697**
London Eye	1h35m	0.81	0.16	0.24	2508
San Marco	1h20m	0.84	0.15	0.19	2770
Tate Modern	0h55m	0.70	0.18	0.42	1204
Time Square	0h43m	0.99	0.14	0.01	1747
Trafalgar	2h44m	0.35	0.18	1.84	2468
Rome	5h14m	–	–	–	14566

Table 5.2: Precision ($N_{TP}/(N_{TP} + N_{FP})$, found pairs (N_{TP}/N_G), overhead (N_{FP}/N_G), and number of registered images for all evaluated matching approaches.

as much as any other technique. At the other end of the spectrum, *PAIGE 10* has the highest true positive rate and results in nearly the highest reconstruction completeness at modest computational cost. Interestingly, PAIGE outperforms the other methods on the Time Square dataset, for which we find that the predictions of positives and negatives are much more separated and confident than for the other datasets. The clear separation is caused by the many video screens (dynamic scenes) and the day/night images, resulting in clearly incorrect pairwise geometry and sets of SIFT features that clearly cannot be aligned. The resulting models of PAIGE are stable and cover the entire scenes (see Figure 5.7).

5.7 Summary

In this chapter, we conducted a comprehensive evaluation of state-of-the-art matching methods for correspondence search in image-based 3D modeling. Based on the insights of this evaluation, we proposed PAIGE, a novel learning-based approach to identify overlapping image pairs for improved efficiency in the matching stage of an image-based 3D modeling pipeline. We showed that approximate correspondence information contains enough information to reliably predict the pairwise image geometry, resulting in significant speedups compared to traditional, exact correspondence approaches. Moreover, we demonstrated that learning-based methods can effectively support 3D reconstruction for improved efficiency in the correspondence search stage. In the next chapter, we explore how to leverage the concept of PAIGE for other modules in image-based 3D modeling, e.g., to improve the robustness of incremental reconstruction by inferring geometric information between image pairs.

6 Efficient Two-View Geometry Classification

In the previous chapter, we introduced a learning-based approach for increasing the efficiency in the correspondence search stage. The underlying idea of the approach was to efficiently predict scene overlap from approximate correspondence information. In this chapter, we borrow the same concept to predict more informative measures of scene overlap. Multiple experiments explore the utility of the proposed approach for increasing the robustness and efficiency in other modules of the incremental sparse reconstruction stage of an image-based 3D modeling pipeline (see Section 2.4).

Incremental reconstruction systems (see Figure 6.2) typically start with feature detection and extraction, followed by feature matching and geometric verification of successfully matched pairs by the assessment of the relative viewing configuration. The major computational effort is spent on these two stages. Next, the incremental reconstruction seeds the model with a carefully selected initial two-view reconstruction and then the procedure incrementally registers new cameras from 2D-3D correspondences, triangulates new 3D features, and refines the reconstruction using a non-linear optimization, known as bundle adjustment. The input to the incremental reconstruction procedure is the scene graph of relative, pairwise epipolar transformations between overlapping images. Information about the relative geometric configuration, such as small or large baseline and forward or sidewards motion, is essential for the robustness of the pipeline, since the incremental reconstruction procedure is highly dependent on the order in which images are registered. A suitable initial image pair and similarly a suitable next-best-view during the incremental extension depends on the relative viewing geometry, i.e., uncertainty of 3D features and camera parameters. However, assessment of the relative viewing geometry for every overlapping image pair in a dataset is computationally expensive. This chapter presents a technique for efficiently recognizing image pairs that work well for incremental reconstruction, thereby significantly improving efficiency for geometric verification as well as improving reconstruction robustness.

The relative geometric configuration of overlapping image pairs serves as the input to the incremental reconstruction procedure. Geometric verification attempts to estimate the relative viewing geometry for pairs of overlapping images. Usually, the majority of image pairs in large-scale, unordered photo collections do not have scene overlap, and thus rejecting invalid pairs dominates execution time. Determining the relative viewing geometry for large image sets comes at significant computational expense, especially if the overlap between most images is sparse. However, it is a

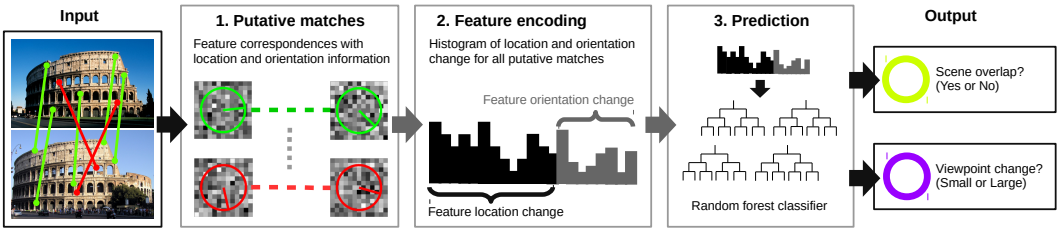

Figure 6.1: The proposed framework for extracting PAIGE, and its application for scene overlap and viewpoint change prediction.

necessary step, as unfavorable initializations or an unfortunate order in camera registrations, e.g., pairs resulting in high camera and/or point uncertainty, can lead to failures in registration and bundle adjustment due to weak geometry, local minima, degeneracies, etc.

The traditional procedure to assess the two-view geometry (see Section 2.2.4 and Section 2.4) in geometric verification comprises fundamental or essential matrix estimation [230] followed by triangulation of 3D points [123], multi-model estimation strategies like GRIC [341], or extended RANSAC procedures for model selection such as QDEGSAC [88]. The essential matrix reveals the entire two-view geometry of calibrated cameras up to unknown scale. Triangulation of 3D points, GRIC, or QDEGSAC then determine the properties of the relative viewing geometry, e.g., the amount and direction of viewpoint change. However, while efficient on a per pair basis, these methods are computationally expensive for a large number of image pairs.

In this chapter, we design an encoding of local image characteristics and build a framework (see Figure 6.1) for the efficient recognition of image pairs with scene overlap and the prediction of the stability of their two-view geometry, all without explicitly reconstructing the actual camera configuration using essential matrix estimation. The approach is based on the geometric shape properties of putative feature correspondences. In Section 6.5, we experimentally demonstrate the utility of the proposed framework for a variety of incremental reconstruction modules, e.g. reducing the set of image pairs for which to perform geometric verification and efficient search for stable initial image pairs in large datasets.

6.1 Related Work

Over the last years large-scale image-based 3D modeling systems have tremendously advanced in terms of increased robustness and reduced runtime. A variety of methods to reduce runtime in different stages of the reconstruction pipeline (see Figure 6.2) have been proposed. However, current state-of-the-art systems typically still spend major time in feature matching and geometric verification. To reduce the number of image pairs in the exhaustive matching module, Frahm et al. [89] leverage iconic image selection through clustering of similar images, Agarwal et al.

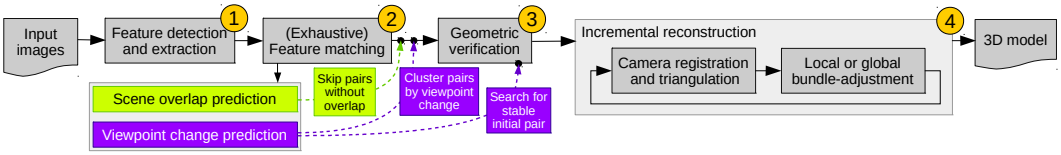

Figure 6.2: The stages of a typical incremental reconstruction pipeline, and applications of our proposed scene overlap and viewpoint change predictor in green and purple.

[3] employ image retrieval systems [15, 64, 149, 150, 217, 231, 335] to only match against similar images, Raguram et al. [258] use GPS tags to match images only to spatially nearby ones, and Wu [365] proposes a preemptive matching strategy. Recently, Hartmann et al. [125] proposed to predict the matchability of individual features to reduce the number of feature comparisons during exhaustive matching. Moreover, in the previous chapter, we proposed a learning-based approach to predict scene overlap based on approximate feature correspondences. However, these techniques still yield a significant amount of image pairs that have no scene overlap, and the set of images contains many redundant viewpoints. Despite the variety of approaches, they all rely on elaborate two-view reconstructions on their potentially reduced set of images in the geometric verification stage. Apart from algorithmic advancements on estimation techniques [230, 254], only Raguram et al. [257] tried to specifically improve runtime of geometric verification using an online learning strategy. However, their approach suffers from a significant loss of image registrations.

Complementary to these previous efforts, we propose a new method to further improve the efficiency in image-based 3D modeling by significantly reducing the runtime of the geometric verification module (Stage 3 in Figure 6.2). Our method can detect overlapping image pairs before geometric verification and for overlapping image pairs it can efficiently classify the geometric two-view configuration in terms of the amount of viewpoint change. We achieve this by extending the method of the previous chapter, which poses the problem of scene overlap detection as a classification task. Similar to this method, we exploit the observation that when images are taken from different viewpoints, corresponding features change in scale, location, and rotation in recognizable patterns. However, instead of approximate correspondences through histogram intersection, we leverage the more reliable feature correspondences from putative matching enabling a less noisy encoding and more accurate prediction. Even though our method builds on the idea of the previous chapter, both approaches can be used together as filters for feature matching and geometric verification in the same SFM pipeline.

6.2 Two-view Geometry

Traditional techniques to derive the two-view geometry (see Section 2.2.4) include feature matching (Stage 2 in Figure 6.2) followed by the robust estimation of epipolar

geometry (Stage 3 in Figure 6.2). The essential matrix reveals the relative viewing geometry [123], but its estimation is computationally expensive [230] due to outlier feature matches and the non-linearity of the estimation problem. RANSAC [85] or its more efficient variants [68, 70, 254, 255] are usually used for robust estimation. RANSAC can deal with large fractions of outliers, but has exponential computational complexity in the number of model parameters and the inlier ratio (see Equation 2.38). Hence, the runtime of two-view geometry estimation quickly rises for image pairs with few inliers from feature matching, which is commonly the case for unordered Internet photo collections [258] (see Section 6.4). Moreover, RANSAC becomes infinitely expensive for image pairs without overlap since those pairs have no inliers. Hence, traditionally a minimum inlier ratio is assumed to set an upper bound for the number of RANSAC iterations. Efficiently detecting image pairs that do not have scene overlap prior to traditional geometric verification can significantly reduce the runtime of the geometric verification stage.

The essential matrix reveals the relative transformation between two views up to an unknown scale. To derive more information about the relative viewing geometry, such as the amount of viewpoint change or the type of motion, further processing is necessary. Two-view scene reconstruction enables to determine the amount of viewpoint change through scene analysis, such as triangulation angle calculation. Alternatively, decision criterions like GRIC [341] or an extend RANSAC procedure like QDEGSAC [88] can be used to avoid degenerate viewing configurations. These methods are computationally expensive. In this chapter, we propose a more efficient method to classify the amount of viewpoint change without an explicit reconstruction of the scene.

6.3 Feature Representation

Our proposed feature representation builds upon the PAIGE feature by Schönberger et al. [283]. In this section, we describe our adaptions and extensions to their method for the efficient prediction of the two-view geometry.

PAIGE takes the extracted features from the feature extraction stage, performs approximate feature matching through histogram intersection, and predicts scene overlap for an image pair by exploiting statistics from corresponding feature properties. Only overlapping image pairs are then forwarded to the computationally expensive pairwise image matching module. Analogous to their approach, we exploit the fact that corresponding features change in scale, location x, and orientation o in recognizable patterns when images are taken at different viewpoints. However, our approach leverages the more precise feature correspondences produced by explicit feature matching, which enables us to produce a less noisy encoding for more accurate prediction.

For each putative feature correspondence of a matched image pair a and b, we determine the normalized image coordinates $\boldsymbol{x}_a, \boldsymbol{x}_b$, such that $x_i \in [0, 1]^2$. Normalization is necessary due to possibly different image resolutions of image a and b.

Next, we calculate the displacement for each correspondence as

$$\Delta x = \|\boldsymbol{x}_a - \boldsymbol{x}_b\| \tag{6.1}$$

We quantize the distribution of feature displacements in a $d_{\Delta x}$-dimensional histogram $\boldsymbol{h}_{\Delta x}$ with evenly spaced bins in the interval $[0, 1]$. Analogously, for each feature correspondence, we calculate the change in feature orientation

$$\Delta o = |o_a - o_b| \mod 2\pi \tag{6.2}$$

and we quantize the distribution of orientation changes in an $d_{\Delta o}$-dimensional histogram $\boldsymbol{h}_o$ with evenly spaced bins in the interval $[0, 2\pi]$. We normalize each of the histograms

$$\bar{\boldsymbol{h}}_x = \frac{\boldsymbol{h}_x}{\|\boldsymbol{h}_x\|} \quad \text{and} \quad \bar{\boldsymbol{h}}_o = \frac{\boldsymbol{h}_o}{\|\boldsymbol{h}_o\|} \tag{6.3}$$

for invariance with respect to the number of feature correspondences. Finally, we use the concatenation of the normalized histograms as our proposed encoding

$$\mathcal{P}(a, b) = \begin{bmatrix} \bar{\boldsymbol{h}}_x & \bar{\boldsymbol{h}}_o \end{bmatrix} \tag{6.4}$$

Similarly to PAIGE, we do not represent scale changes in the feature, as we found it does not improve the discriminative power of the feature. The next section describes a classification strategy leveraging this feature representation for scene overlap and triangulation angle prediction.

6.4 Classification

Based on the proposed encoding in Section 6.3, we now describe a classification strategy to answer the following two questions for any given image pair: *Is there scene overlap ($\mathcal{C}_\mathcal{A}$)?* and *Is there a stable two-view geometry ($\mathcal{C}_\mathcal{B}$)?*. We choose random forests [40] as a classification method as it gave best results in terms of accuracy and computational efficiency.

6.4.1 Training

For training, we use an existing 3D reconstruction of an unordered Internet image collection, which is computed using exhaustive feature matching and geometric verification. To collect training data, we calculate the mean triangulation angle $\bar{\alpha}_{ab}$ for each image pair $\{a, b\}$ with scene overlap and extract the proposed feature $\mathcal{P}(a, b)$.

Specifically, we use 3D reconstructions of 17 unordered Internet photo collections from different locations across the world (Rome, Notre Dame, Stonehenge, etc.) and a set of temporally sequential image sequences acquired by video cameras (to account for the orientation bias of crowd-sourced images) to serve as a training and test dataset (see Table 6.1). The dataset consists of 1,602,996 matched (≥ 30 putative feature correspondences) out of all 73,542,704 possible image pairs, of which

	Total pairs	Matched pairs	Verified pairs	e_{all}	e_{geo}	d_0	d_1	d_2
Train & Test	73,542,704	1,602,996	449,207	47%	70%	2,357,586,073 (100%)	295,230,950 (12.5%)	194,851,427 (8.3%)
Oxford	82,944	21,574	16,303	56%	70%	72,445,847 (100%)	14,557,490 (20.0%)	9,604,943 (13.2%)
Louvre	693,889	252,798	4,539	27%	65%	613,625,401 (100%)	72,898,480 (11.9%)	48,212,996 (7.9%)
Acropolis	8,767,521	439,609	16,492	29%	78%	1,139,606,104 (100%)	117,886,481 (10.3%)	77,105,077 (6.8%)

Table 6.1: Evaluation datasets with average inlier ratio for matched (e_{all}) and verified pairs (e_{geo}). Number of RANSAC iterations for geometric verification without classifiers $\mathcal{C}_\mathcal{A}$ and $\mathcal{C}_\mathcal{B}$ (d_0), after classifier $\mathcal{C}_\mathcal{A}$ (d_1), and after classifiers $\mathcal{C}_\mathcal{A}$ and $\mathcal{C}_\mathcal{B}$ (d_2). d_0, d_1, d_2 for *Train & Test* only averaged over held-out test set.

449,207 pairs have a geometrically verified overlap (≥ 15 inliers for essential matrix estimation). Table 6.1 lists the minimum number of RANSAC iterations (see Section 6.2) for essential matrix estimation of all matched image pairs with RANSAC confidence $\eta = 0.99$, $M = 5$ model parameters for essential matrix estimation, and a minimum inlier ratio $\epsilon_{min} = 0.28$. As a result of these parameters, RANSAC runs for a maximum of $d_{max} = 2674$ iterations for each image pair. The maximum number of iterations is reached for $< 5\%$ of the pairs, since $> 95\%$ of the pairs have an inlier ratio $> 28\%$. We employ SIFT features and use the ratio test for robust matching [200]. Note that SIFT could be replaced by any other feature that provides location and orientation properties. The quantization of the location and orientation histograms include all 110,587,256 putative feature matches for all image pairs, including 51,968,824 geometric inliers and 58,618,432 outliers, i.e. overall inlier ratio $e_{all} = 47\%$ and $e_{geo} = 70\%$ for geometrically verified pairs. We use a 172-dimensional feature vector $\mathcal{P}(a, b)$ with $d_{\Delta x} = 100$ and $d_{\Delta o} = 72$. Figure 6.3 visualizes the distribution of triangulation angles and Figure 6.4 the average feature vector $\mathcal{P}(a, b)$ over all image pairs. We find a significant amount of pairs with only a small viewpoint change, caused by popular viewpoints of famous landmarks and less stable feature matching for large viewpoint changes. As expected, the overall location and orientation change is higher for wide baselines than for small baselines, and the orientation change for images without overlap is significantly larger.

To answer the two binary classification problems $\mathcal{C}_\mathcal{A}$ and $\mathcal{C}_\mathcal{B}$, we divide the set of image pairs into three different categories: small and large mean triangulation angle (using an angle threshold), and no scene overlap (pairs with failed geometric verification). Next, the dataset is split in randomly permuted training (70%) and test samples (30%). Two random forests were trained on the training dataset, using 50 decision trees each, entropy as the splitting criterion, and considering $\sqrt{172} \approx 13$ features when looking for the best split at each node in the tree. A minimum number of 5 samples per leaf is enforced to avoid over-fitting. The parameters were determined with a 5-folded cross-validation on the training set. The trained random forests can efficiently decide on the two classification problems $\mathcal{C}_\mathcal{A}$ and $\mathcal{C}_\mathcal{B}$ and the

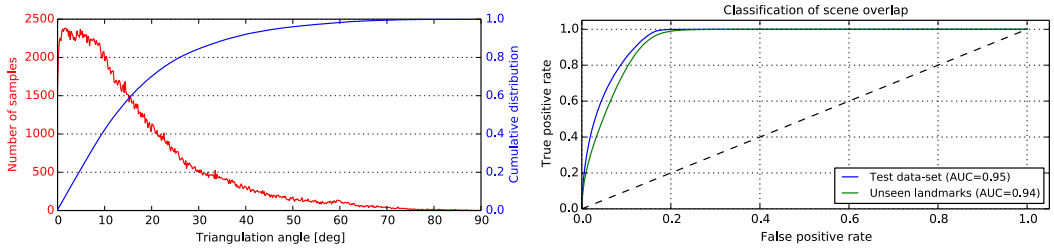

Figure 6.3: *Left:* Triangulation angle distribution for geometrically verified image pairs. *Right:* Performance evaluation for scene overlap classification $\mathcal{C}_{\mathcal{A}}$.

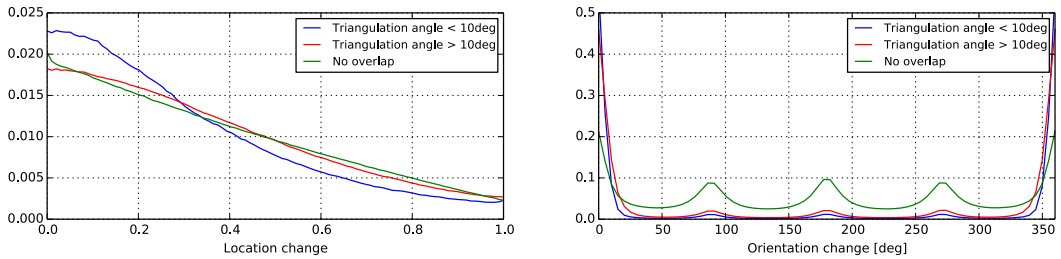

Figure 6.4: Location and orientation change distributions of PAIGE for the entire dataset.

integration of the proposed classifiers in a typical SFM pipeline is demonstrated in Section 6.5.

6.4.2 Performance Evaluation

On a conventional desktop computer the training time for both classifiers is approximately 5min, and the classification frequency averages at around 200K pairs per second including quantization and prediction, compared to around 20K pairs per second for the PAIGE approach [283]. We evaluate the classification performance on the held-out test set (30%), and three unordered Internet photo collections of completely unseen landmarks (see Table 6.1) at different geo-locations (Oxford, Louvre, Acropolis). Figure 6.5 demonstrates the performance for both classifiers $\mathcal{C}_{\mathcal{A}}$ and $\mathcal{C}_{\mathcal{B}}$. For both classifiers $\mathcal{C}_{\mathcal{A}}$ and $\mathcal{C}_{\mathcal{B}}$, we find minimal bias towards the trained landmarks. In another evaluation, the performance of classifier $\mathcal{C}_{\mathcal{B}}$ is evaluated on the unseen landmarks with respect to different triangulation angle thresholds by only considering overlapping image pairs using $\mathcal{C}_{\mathcal{A}}$. Figure 6.5 shows that our method generalizes well. Next, we demonstrate the applicability of the two classifiers $\mathcal{C}_{\mathcal{A}}$ and $\mathcal{C}_{\mathcal{B}}$ within the context of SFM.

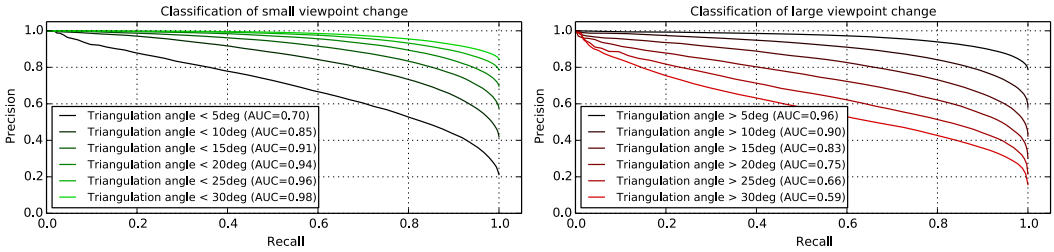

Figure 6.5: Performance evaluation for triangulation angle classification $\mathcal{C}_{\mathcal{B}}$ using different angle thresholds. Area under curve as AUC.

6.5 Efficient Structure-from-Motion

In the following, we show the integration of the proposed method into a typical SFM system (see Figure 6.2) with respect to the datasets in Table 6.1. We demonstrate that the classifiers significantly improve the computational performance by reducing the set of images being evaluated by the geometric verification module. Furthermore, we show the utility for the efficient search of stable initial image pairs in large datasets.

6.5.1 Scene Overlap Prediction

In Section 6.2, we have seen that the number of RANSAC iterations is exponentially dependent on the outlier ratio. Hence, we spend a majority of the runtime to evaluate pairs with no scene overlap. For these pairs RANSAC reaches the maximum number of iterations, leading to a significant computational burden. Our proposed method allows to filter these pairs prior to geometric verification, preventing the high computational effort for pairs that do not contribute to the final 3D model. Assuming we filtered all pairs with no scene overlap for the unseen landmarks (see Table 6.1) using a perfect classifier, and run RANSAC only for the remaining pairs, we can reduce the number of iterations by a factor of 35. For our classifier $\mathcal{C}_{\mathcal{A}}$, we enforce a precision of ≥ 0.99 for classifying pairs with no scene overlap using an appropriate prediction confidence, and thereby lower the recall to 81%. The geometric verification module thus only misses approximately 1.7% of actually overlapping pairs. Notice that the missing image pairs have negligible impact, considering that most images are still contributing to the final model through other transitive image pairs. Using these parameters, we achieve a 7.8x speedup of geometric verification for the training & test-set, and overall an 8.9x speedup for the unseen landmarks compared to the potential speedup of 35 using a perfect classifier. Since the computational effort for the classification is insignificant compared to geometric verification (3 to 4 orders of magnitudes faster), this speedup directly propagates to the overall geometric verification runtime. Note that the performance improves even more, if we verify very weak image pair connections, since we assume a minimum inlier ratio of 28% ($d_{max} = 2674$). The reported runtimes are a vast improvement over previous

efforts [257], which achieve a 70% speedup but lose 26% of image registrations, in contrast to our 9-fold speedup with 1.7% loss. Due to the less noisy encoding based on putative feature matches, our approach misses significantly fewer image pairs than the PAIGE [283] approach, which loses 38-90% of actually overlapping image pairs. Note that both approaches could be applied together in the same pipeline, since PAIGE operates as a filter to feature matching and this approach as a filter to geometric verification. On average, we find that exhaustive matching and geometric verification spends 52% in Stage 2 using a GPU SIFT implementation and 48% in Stage 3 using a multi-threaded CPU RANSAC implementation. Ideally, the PAIGE approach [283] can eliminate the runtime of Stage 2 for sparsely connected image collections. Our proposed approach in this chapter reduces the runtime of Stage 3 by a factor of 9. Combining the two approaches, we can effectively eliminate the original cost of Stages 2 and 3 compared to standard exhaustive matching.

6.5.2 Redundant Viewpoint Detection

In SFM systems, we achieve redundancy by tracking a 3D feature over multiple images. Corresponding features between two images cannot only be verified with direct pairwise geometric verification, but also by bridging the track using an intermediate image, that has the same point in common. Especially for small viewpoint changes, the continuation of tracks over multiple images is very likely. Beyond that, uncertainty and reliability of parameter estimates in bundle adjustment only improve up to a certain redundancy [21, 342], i.e. the resulting 3D models do not gain from high redundancy in the same way as we spend an unproportional amount of increased computational effort. For outlier-detection in SFM, it is typically critical to have at least 3-4 observations per 3D point. Leveraging these facts and classifier $\mathcal{C}_{\mathcal{B}}$, we can detect clusters of images with small viewpoint change. Next, we select one iconic image in the cluster with the most points in common, and finally only perform geometric verification from the iconic image to the rest of the images in the cluster rather than exhaustive verification between all pairs. Moreover, for very large clusters, we can limit the number of images for geometric verification, and simply register the remaining images with respect to the final model using 2D-3D pose estimation [85]. In both datasets, we see 40% of image pairs (282,387) with small viewpoint change ($\bar{\alpha} < 10^{\circ}$). To find clusters, we build an undirected graph of all pairs with small viewpoint change using images as nodes and small viewpoint change as edges. In this graph, we find 6,404 disjoint maximal cliques [42, 56, 339] in the training & test-set. These cliques are similar to the clusters described by Frahm et al. [89], but our clusters are based on viewpoint change rather than GIST similarity [234]. By only considering edges from the iconic to the remaining images in a clique, we reduce the pairwise geometric verifications from 97,564 to 20,426. In addition, we further decrease this number to 14,469 by only considering images up to a maximum cluster size of 10, i.e. we improve geometric verification runtime by 30% from 282,387 to 199,292 pairs. This technique is especially beneficial for very dense datasets, as often encountered in Internet photo collections.

6.5.3 Search for Optimal Initial Pairs

Searching for a good initial pair as a seed for incremental reconstruction is computationally expensive, since it involves essential matrix estimation followed by triangulation of feature correspondences, and the calculation of triangulation angles or uncertainty estimates. With state-of-the-art essential matrix solvers [230] and linear triangulation [121, ch. 12.2], around 10-50 two-view reconstructions can be computed per second [257] using the parameters as in Section 6.4. As opposed to the traditional approach, our classifier $\mathcal{C}_{\mathcal{B}}$ enables us to efficiently search for stable pairs through an entire dataset at significantly reduced computational cost. In the unseen landmarks of our evaluation dataset, we find 14,886 stable pairs (out of 17,330 true stable pairs) with $\bar{\alpha} > 20°$, where 83% of the reported pairs are actually stable. We use these pairs as initial seeds for the incremental reconstruction by ranking the reported stable pairs based on the number of putative feature matches to attain higher initial redundancy. On the one hand, our method leads to significantly faster search for initial pairs and, on the other hand, it allows us to search for optimal initial image pairs globally.

6.6 Summary

In this chapter, we extended the PAIGE approach to efficient two-view geometry classification. The presented extension of PAIGE further improves the computational efficiency and robustness of different modules in an image-based 3D modeling system. Experiments demonstrated a speedup for geometric verification by an order of magnitude over the traditional exhaustive approach, while only loosing less than 1.7% of the valid image pairs. Compared to PAIGE, this approach provides an order of magnitude faster prediction performance, while achieving significantly better prediction accuracy. PAIGE and our approach are complementary methods that can both be integrated into the same reconstruction pipeline to speedup feature matching and geometric verification. Furthermore, the framework significantly reduces runtime of incremental reconstruction for very dense photo collections and we demonstrated the utility for the efficient, global search of image pairs for the robust initialization of incremental reconstruction.

Part III

Sparse Reconstruction

7 Structure-from-Motion Revisited

The previous chapters presented and evaluated several approaches that drastically improve upon the correspondence search stage in terms of efficiency, completeness, and robustness. A better performance in correspondence search leads to a more complete and more accurate scene graph and thus positively impacts the performance of the subsequent sparse and dense reconstruction stages. Despite sophisticated algorithms and careful filtering in two-view geometry estimation, the scene graph is typically still rather noisy and contains many outliers. The goal of the incremental reconstruction procedure (see Section 2.4.2) is to robustly combine the individual two-view reconstructions into a globally consistent and accurate 3D model of the scene. In this chapter, we first provide an overview of the current limitations and challenges of incremental reconstruction algorithms. Next, we present several algorithmic improvements to the incremental reconstruction process that enable more robust 3D modeling from unstructured image collections.

7.1 Challenges

While the previous state-of-the-art incremental reconstruction algorithms could handle the diverse and complex distribution of images in large-scale Internet photo collections [3, 89, 129, 310], they frequently fail to produce fully satisfactory results in terms of completeness and robustness. Oftentimes, the systems fail to register a large fraction of images that empirically should be registrable [89, 129], or the systems produce broken models due to image misregistrations or accumulated drift. First, this may be caused by correspondence search producing an incomplete or incorrect scene graph, e.g., due to approximations in feature matching or ambiguous scene structure, and therefore providing neither the necessary connectivity for complete models nor the sufficient redundancy for reliable estimation. Second, this may be caused by the reconstruction stage failing to register images due to missing or inaccurate scene structure – image registration and triangulation have a symbiotic relationship in that images can only be registered to existing scene structure and scene structure can only be triangulated from registered images [396]. Maximizing the accuracy and completeness of both at each step during the incremental reconstruction is a key challenge in an incremental reconstruction system. In this chapter, we address these challenges and significantly improve the reconstruction results over the current state of the art in terms of completeness, robustness, and accuracy while maintaining the scalability and efficiency, as demonstrated in Section 7.7.

7.2 Scene Graph Augmentation

We propose a multi-model geometric verification strategy to efficiently augment the scene graph with the appropriate geometric two-view relation. First, we estimate a fundamental matrix, given no prior intrinsic calibration of the camera. If at least N_F inliers are found, we consider the image pair as geometrically verified. Next, we classify the transformation by determining the number of homography inliers N_H for the same image pair. To approximate model selection methods like GRIC, we assume a moving camera in a general scene if

$$\frac{N_H}{N_F} < \epsilon_{HF} \ . \tag{7.1}$$

For calibrated images, we also estimate an essential matrix and its number of inliers N_E. For uncalibrated images, we determine sample different essential matrices by normalizing the image coordinates using different intrinsic hypotheses. The hypothesis leading to the maximal number of inliers N_E is chosen as the final estimate of the essential matrix. We assume correct intrinsic camera calibration if

$$\frac{N_E}{N_F} > \epsilon_{EF} \ . \tag{7.2}$$

In case of correct calibration and

$$\frac{N_H}{N_F} < \epsilon_{HF} \ , \tag{7.3}$$

we decompose the essential matrix, triangulate points from inlier correspondences, and determine the median triangulation angle α_m. Using α_m, we distinguish between the case of pure rotation (panoramic) and planar scenes (planar). Furthermore, a frequent problem in Internet photos are watermarks, timestamps, and frames (WTFs) [129, 360] that incorrectly link images of different landmarks. We detect such image pairs by estimating a similarity transformation with N_S inliers at the image borders. Any image pair that satisfies

$$\frac{N_S}{N_F} > \epsilon_{SF} \quad \text{and} \quad \frac{N_S}{N_E} > \epsilon_{SE} \tag{7.4}$$

is considered a WTF and not inserted to the scene graph. For valid pairs, we label the scene graph with the model type (general, panoramic, planar) alongside the inliers of the model with maximum support N_H, N_E, or N_F. The model type is leveraged to initialize the reconstruction only from non-panoramic and preferably calibrated image pairs. An already augmented scene graph enables to efficiently find an optimal initialization for a robust reconstruction process. In addition, we do not triangulate from panoramic image pairs during incremental reconstruction in order to avoid degenerate points and thereby improve robustness of triangulation and subsequent image registrations.

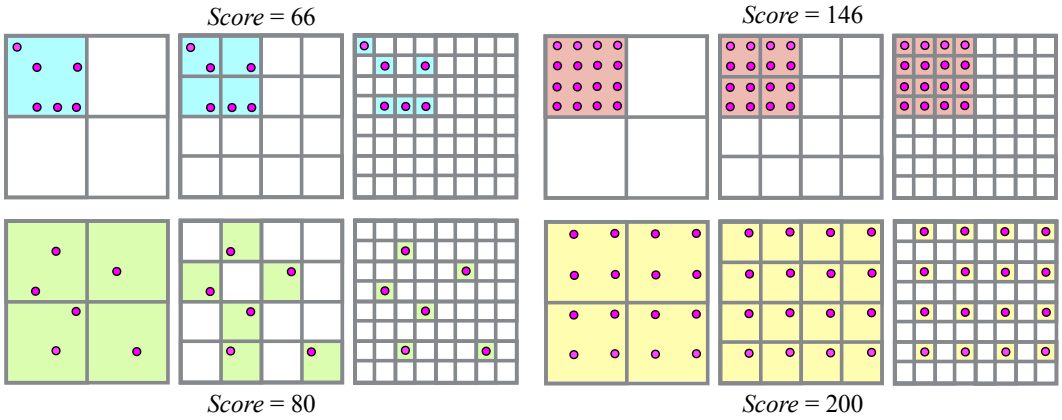

Figure 7.1: Scores for different number of points (left and right) with different distributions (top and bottom) in the image for $L = 3$.

7.3 Next Best View Selection

Next best view planning has been studied in the fields of computer vision, photogrammetry, and robotics [58]. For robust and accurate image-based 3D modeling, choosing the next best view aims to minimize the reconstruction error [83, 117]. Here, we propose an efficient next best view strategy following an uncertainty-driven approach that maximizes reconstruction robustness and accuracy.

Choosing the next best view in an incremental reconstruction pipeline is critical, as every decision impacts the remaining reconstruction. A single bad decision may lead to a cascade of camera mis-registrations and faulty triangulations. In addition, choosing the next best view greatly impacts both the quality of pose estimates and the completeness and accuracy of triangulation. An accurate pose estimate is essential for robust modeling, as point triangulations may fail if the pose is inaccurate. The decision of choosing the next best view is challenging, since for unstructured image collections there is usually no a priori information about scene coverage and camera parameters, and therefore the decision is based entirely on information derived from image appearance [83], two-view correspondences, and the incrementally reconstructed scene [117, 310].

A popular strategy is to choose the image that sees most triangulated points [309] with the aim of minimizing the uncertainty in camera resection. Haner et al. [117] propose an uncertainty-driven approach that minimizes the reconstruction error. Usually, the camera that sees the largest number of triangulated points is chosen, except when the configuration of observations is not well-conditioned. To this end, Lepetit et al. [185] experimentally show that the accuracy of the camera pose using PnP depends on the number of observations and their distribution in the image. For uncalibrated image collections, the standard camera calibration problem is extended to the estimation of intrinsic parameters in the case of missing or inaccurate prior calibration. A large number of 2D-3D correspondences provides this estimation with

redundancy [185], while a uniform distribution of points avoids bad configurations and enables the reliable estimation of intrinsics [211].

The candidates for the next best view are the not yet registered images that see a sufficient number of triangulated points. Keeping track of this statistic can be efficiently implemented using the scene graph. For unstructured image datasets, this graph can often be very dense, since many images may observe the same scene structure. Hence, there are many candidate views to choose from at each step in the reconstruction. Exhaustive covariance propagation as proposed by Haner et al. is not feasible, since the covariance would need to be computed and analyzed for each candidate at each step. Our proposed method approximates their uncertainty-driven approach using an efficient multi-resolution analysis.

Towards this goal, we must simultaneously keep track of the number of visible points and their distribution in each candidate image. More visible points and a more uniform distribution of these points should result in a higher ranking score $\mathcal{S}$ [146], such that images with a better-conditioned configuration of visible points are registered first. To achieve this goal, we discretize the image into a fixed-size grid with K_l bins in both dimensions. Each cell takes two different states: *empty* and *full*. Whenever a point within an *empty* cell becomes visible during the reconstruction, the cell's state changes to *full* and the score $\mathcal{S}_i$ of the image is increased by a weight w_l. With this scheme, we quantify the number of visible points. Since cells only contribute to the overall score once, we favor a more uniform distribution over the case when the points are clustered in one part of the image (i.e. only a few cells contain all visible points). However, if the number of visible points is $N_t \ll K_l^2$, this scheme may not capture the distribution of points well as every point is likely to fall into a separate cell. Consequently, we extend the previously described approach to a multi-resolution pyramid with $l = 1...L$ levels by partitioning the image using higher resolutions $K_l = 2^l$ at each successive level. The score is accumulated over all levels with a resolution-dependent weight $w_l = K_l^2$. This data structure and its score can be efficiently updated online. Figure 7.1 shows scores for different configurations, and Section 7.7 demonstrates improved reconstruction robustness and accuracy using this strategy.

7.4 Robust and Efficient Triangulation

A large number of methods for multi-view triangulation exist [2, 8, 121, 122, 158, 189, 201, 235]. These methods suffer from limited robustness or high computational cost in the setting of large-scale reconstruction from unstructured imagery. In thus section, we present a robust and efficient triangulation method, that overcomes the limitations of prior approaches.

Especially for sparsely matched image collections, exploiting transitive feature correspondences boosts triangulation completeness and accuracy, and hence improves subsequent image registrations. Efficient feature correspondence search techniques usually favor image pairs similar in appearance, and as a result two-view feature

correspondences often stem from image pairs with a small baseline. Leveraging transitivity can establish correspondences between images with larger baselines and thus enables more accurate triangulation. Hence, we form feature tracks by concatenating two-view feature correspondences.

A variety of approaches have been proposed for multi-view triangulation from noisy image observations [8, 122, 201]. While some of the proposed methods are robust to a certain degree of outlier contamination [2, 121, 158, 189, 235], to the best of our knowledge none of the approaches can handle the high outlier ratio often present in transitive feature tracks (see Figure 7.5). We refer to Kang et al. [158] for a detailed overview of existing multi-view triangulation methods. In this section, we propose an efficient, sampling-based triangulation method that can robustly estimate all points within an outlier-contaminated feature track.

Feature tracks often contain a large number of outliers due to erroneous two-view geometry verification of ambiguous matches along the epipolar line. A single mismatch merges the tracks of two or more independent points. For example, falsely merging a two-view feature track with another 10-view feature track results in an outlier ratio of $\frac{10}{2+10} \approx 83\%$. In addition, inaccurate camera poses may invalidate track elements due to high reprojection errors. Hence, for robust triangulation, it is necessary to find a consensus set of track elements before performing a refinement using multiple views. Moreover, to recover the potentially multiple points of a feature track from a faulty merge, a recursive triangulation scheme is necessary.

Traditional reconstruction pipelines, such as *Bundler*, sample all pairwise combinations of track elements, perform two-view triangulation, and then check if at least one solution has a sufficient triangulation angle. If a well-conditioned solution is found, multi-view triangulation is performed on the whole track, and it is accepted if all observations pass the cheirality constraint [124]. This method is not robust to outliers, as it is not possible to recover independent points merged into one track. Also, it has significant computational cost due to exhaustive pairwise triangulation. Our method overcomes both limitations.

To handle arbitrary levels of outlier contamination, we formulate the problem of multi-view triangulation using RANSAC. We consider the feature track

$$\mathcal{T} = \{T_n \mid n = 1...N_T\} \quad \text{with} \quad T_n = (\hat{\boldsymbol{x}}_n, \boldsymbol{P}_n\ ) \tag{7.5}$$

as a set of measurements with an a priori unknown ratio ϵ of inliers. A measurement T_n consists of the normalized image observation $\hat{\boldsymbol{x}}_n \in \mathbb{R}^2$ and the corresponding extrinsic camera calibration $\hat{\boldsymbol{P}}_n \in SE(3)$ defining the transformation from world to camera frame

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T\boldsymbol{T} \end{bmatrix} \tag{7.6}$$

with $\boldsymbol{R} \in \boldsymbol{SO}(3)$ and $\boldsymbol{T} \in \mathbb{R}^3$. Our objective is to maximize the support of measurements conforming with a well-conditioned two-view triangulation

$$\boldsymbol{X}_{ab} \simeq \tau\left(\bar{\boldsymbol{x}}_a, \bar{\boldsymbol{x}}_b, \boldsymbol{P}_a, \boldsymbol{P}_b\right) \text{ with } a \neq b, \tag{7.7}$$

where τ is any chosen triangulation method (in our case the direct linear transform method [123]) and $\boldsymbol{X}_{ab}$ is the triangulated two-view point. Note, that we do not triangulate from panoramic image pairs to avoid degenerate triangulation angles due to inaccurate pose estimates. A well-conditioned model satisfies two constraints. First, a sufficient triangulation angle α

$$\cos\alpha = \frac{\boldsymbol{T}_a - \boldsymbol{X}_{ab}}{\|\boldsymbol{T}_a - \boldsymbol{X}_{ab}\|} \cdot \frac{\boldsymbol{T}_b - \boldsymbol{X}_{ab}}{\|\boldsymbol{T}_b - \boldsymbol{X}_{ab}\|}. \tag{7.8}$$

Second, positive depths d_a and d_b with respect to the views $\boldsymbol{P}_a$ and $\boldsymbol{P}_b$ (also known as the cheirality constraint [124]), with the depth being defined as

$$d = \begin{bmatrix} p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_{ab} \\ 1 \end{bmatrix}, \tag{7.9}$$

where p_{mn} denotes the element in row m and column n of $\boldsymbol{P}$. A measurement T_n is considered to conform with the model if it has positive depth d_n and if its reprojection error

$$e_n = \left\| \bar{\boldsymbol{x}}_n - \begin{bmatrix} x'/z' \\ y'/z' \end{bmatrix} \right\| \text{ with } \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \boldsymbol{P}_n \begin{bmatrix} \boldsymbol{X}_{ab} \\ 1 \end{bmatrix} \tag{7.10}$$

is smaller than a certain threshold t. RANSAC maximizes $\mathcal{K}$ as an iterative approach and generally it uniformly samples the minimal set of size two at random. However, since it is likely to sample the same minimal set multiple times for short feature tracks, we define our random sampler to only generate unique two-view samples. To ensure with confidence η that at least one outlier-free minimal set has been sampled, RANSAC must run for at least K iterations. Since the a priori inlier ratio is unknown, we set it to a small initial value ϵ_0 and adapt K whenever we find a larger consensus set (adaptive stopping criterion). Because a feature track may contain multiple independent points, we run this procedure recursively by removing the consensus set from the remaining measurements. The recursion stops if the size of the latest consensus set is smaller than three. The evaluations in Section 7.7 demonstrate increased triangulation completeness at reduced computational cost for the proposed method.

7.5 Efficient and Robust Bundle Adjustment

To mitigate accumulated errors, it is necessary to frequently perform bundle adjustment after image registration and triangulation. Usually, there is no need to perform global bundle adjustment after each iteration, since the incremental reconstruction procedure only affects the model locally. Hence, we perform local bundle adjustment on the set of most-connected images and points after each image registration. Analogous to *VisualSFM*, we perform global bundle adjustment only after growing the model by a certain percentage, resulting in an amortized linear run-time of the incremental reconstruction algorithm.

7.5.1 Parameterization

To account for potential outliers, we employ the Cauchy function as a robust loss function ρ_j in local bundle adjustment. In contrast, global bundle adjustment does not use a robustifier in order to increase the convergence basin for drift correction. For problems up to a few hundred cameras, we use a sparse direct solver, and for larger problems, we rely on a solver based on preconditioned conjugate gradients. We use *Ceres Solver* [5] as an non-linear least-squares optimization framework and provide the option to share camera models of arbitrary complexity among any combination of images. For images without prior intrinsic calibration information, we rely on a simple camera model with one radial distortion parameter, as the estimation relies on pure self-calibration.

7.5.2 Filtering

After bundle adjustment, typically some observations do not conform with the reconstructed model. Accordingly, we filter image observations with large reprojection errors [310, 365]. Moreover, for each point, we check for well-conditioned geometry by enforcing a minimum triangulation angle over all pairs of viewing rays [310]. After global bundle adjustment, we also check for degenerate intrinsic camera parameters, e.g., caused by panorama images or artificially enhanced images. Typically, those cameras only have outlier observations or their intrinsic parameters converge to a bogus minimum. Hence, we do not constrain the focal length and distortion parameters to an *a priori* fixed range but let them freely optimize in bundle adjustment. Since principal point calibration is an ill-posed problem [1], we keep it fixed at the image center for uncalibrated cameras. Cameras with an abnormal field of view or large distortion coefficient magnitude are considered as corrupt and filtered after global bundle adjustment.

7.5.3 Re-Triangulation

Analogous to *VisualSFM*, we perform re-triangulation to account for drift effects prior to global bundle adjustment (pre bundle adjustment re-triangulation). Using re-triangulation, bundle adjustment often significantly improves camera and point parameters. As an improvement, we propose to extend the very effective re-triangulation before bundle adjustment with an additional re-triangulation after bundle adjustment. The purpose of this step is to improve the completeness of the reconstruction (see Section 2.4.2) by continuing the tracks of points that previously failed to triangulate, e.g., due to inaccurate poses etc. Instead of increasing the triangulation thresholds, we only continue tracks with observations whose errors are below the filtering thresholds. In addition, we attempt to merge tracks and thereby provide increased redundancy for the next bundle adjustment step.

7.5.4 Iterative Refinement

Traditional reconstruction pipelines, such as *Bundler* and *VisualSFM*, perform a single instance of bundle adjustment and filtering. Due to accumulated drift and the re-triangulation prior to bundle adjustment, usually a significant portion of the observations in bundle adjustment are outliers and subsequently filtered. Since bundle adjustment is severely affected by outliers, a second step of bundle adjustment can significantly improve upon the initial results. We therefore propose to perform bundle adjustment, re-triangulation, and filtering in an iterative optimization until the number of filtered observations and the number of re-triangulated points after bundle adjustment diminishes. In most cases, after the second refinement iteration, results improve dramatically and the optimization converges. Section 7.7 demonstrates that the proposed iterative refinement significantly boosts reconstruction completeness and accuracy.

7.6 Redundant View Mining

Bundle adjustment is a major performance bottleneck in the incremental reconstruction pipeline, since it is repeatedly executed on larger and larger problems during the process. Furthermore, especially for Internet photo collections, bundle adjustment spends significant time on optimizing many near-duplicate images. In this section, we propose a method that exploits the inherent characteristics of the incremental reconstruction algorithm and dense image collections for a more efficient parameterization of bundle adjustment by clustering redundant cameras into groups of high scene overlap.

Unstructured image datasets, such as Internet photo collections, usually have a highly non-uniform visibility pattern due to varying popularity of points of interest. Moreover, the resulting reconstructions are usually clustered into fragments of points that are co-visible in many images. A number of previous works exploit this fact in order to improve the efficiency of bundle adjustment, including Kushal et al. [177] who analyze the visibility pattern for efficient preconditioning of the reduced camera system. Ni et al. [229] partition the cameras and points into submaps, which are connected through separator variables, by posing the partitioning as a graph cut problem on the graph of connected camera and point parameters. Bundle adjustment then alternates between fixing the cameras and points and refining the separator variables, and *vice versa*. Another approach by Carlone et al. [54] collapses multiple points with a low-rank into a single factor imposing a high-rank constraint on the cameras, providing a computational advantage when cameras share many points.

Our proposed method is motivated by these previous works. Similar to Ni et al., we partition the problem into submaps whose internal parameters are factored out. We have three main contributions: First, an efficient camera grouping scheme leveraging the inherent properties of SFM and replacing the expensive graph-cut employed by Ni et al.. Second, instead of clustering many cameras into one submap,

we partition the scene into many small, highly overlapping camera groups. The cameras within each group are collapsed into a single camera, thereby reducing the cost of solving the reduced camera system. Third, as a consequence of the much smaller, highly overlapping camera groups, we eliminate the alternation scheme of Ni et al. by skipping the separator variable optimization.

In an incremental reconstruction algorithm images and points can be grouped into two sets based on whether their parameters were affected by the latest incremental model extension. For large reconstructions, most of the scene remains unaffected since the model usually extends locally. Global bundle adjustment naturally optimizes more for the newly extended parts while other parts only improve in case of drift [365]. Moreover, unstructured image collections often have an uneven camera distribution with many redundant viewpoints. Motivated by these observations, we partition the unaffected scene parts into groups

$$\mathcal{G} = \{G_r \mid r = 1...N_G\} \tag{7.11}$$

of highly overlapping images and parameterize each group G_r as a single, generalized camera with fixed relative extrinsic calibration. Images affected by the latest incremental model extension are grouped independently to allow for an optimal refinement of their parameters. Note that the independent parameterization results in the standard bundle adjustment formulation. For unaffected images during the latest incremental model extension, we create image groups of cardinality N_{G_r}. We consider an image as affected if it was added during the latest model extension or if more than a ratio ϵ_r of its observations have a reprojection error larger than r pixels (to refine re-triangulated cameras).

Images within a group should be as redundant as possible [229] for a maximum performance gain at minimal accuracy impact, and the number of co-visible points between images is a measure to describe their degree of mutual interaction [177]. For a scene with N_X points, each image can be described by a binary visibility vector $\boldsymbol{v}_i \in \{0,1\}^{N_X}$, where the n-th entry in $\boldsymbol{v}_i$ is 1 if point $\boldsymbol{X}_n$ is visible in image i and 0 otherwise. The degree of interaction between image a and b is calculated using bitwise operations on their vectors $\boldsymbol{v}_i$

$$V_{ab} = \frac{\|\boldsymbol{v}_a \wedge \boldsymbol{v}_b\|}{\|\boldsymbol{v}_a \vee \boldsymbol{v}_b\|} \ . \tag{7.12}$$

To build groups, we sort the images as

$$\bar{\mathcal{I}} = \{I_i \mid \|\boldsymbol{v}_i\| \geq \|\boldsymbol{v}_{i+1}\|\} \ . \tag{7.13}$$

We initialize a group of images G_r by removing the first image I_a from $\bar{\mathcal{I}}$ and finding the image I_b that maximizes V_{ab}. If an image I_b satisfies

$$V_{ab} > V \quad \text{and} \quad |G_r| < S \ , \tag{7.14}$$

it is removed from $\bar{\mathcal{I}}$ and added to group G_r. Otherwise, a new group of images is initialized. To reduce the time of finding I_b, we employ the heuristic of limiting the

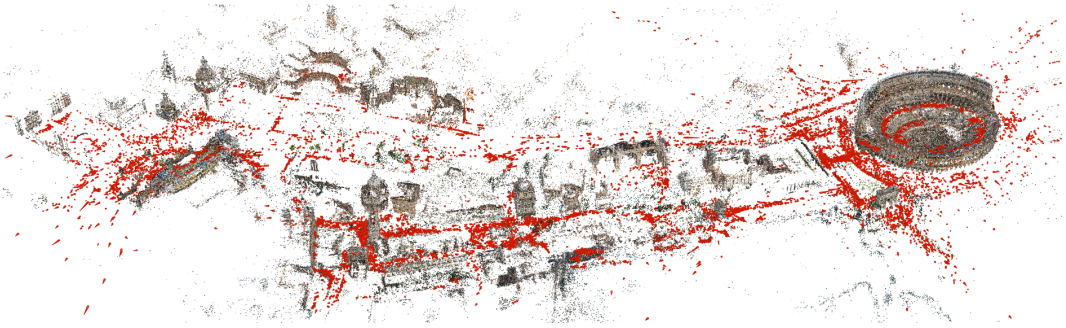

Figure 7.2: Sparse reconstruction result of the city of Rome produced by our proposed system with 21K registered images out of 75K unstructured Internet images.

search to the K_r spatial nearest neighbors with a common viewing direction in the range of $\pm\beta$ degrees, motivated by the fact that those images have a high likelihood of sharing many points.

Each group of images is then parameterized as a generalized camera with fixed relative extrinsic calibration. In bundle adjustment, we then only optimize over the common absolute extrinsic calibration $\boldsymbol{G}_r \in SE(3)$ of the generalized camera. The overall bundle adjustment optimization problem (see Equation 2.20) for grouped and ungrouped images is defined as

$$\boldsymbol{G}_r^*, \boldsymbol{X}^* = \underset{\boldsymbol{G}_r, \boldsymbol{X}}{\arg\min} \left\| \boldsymbol{x} - \boldsymbol{P}_c \boldsymbol{G}_r \boldsymbol{X} \right\|^2 \tag{7.15}$$

using the optimized parameters $\boldsymbol{G}_r$ and a fixed camera calibration $\boldsymbol{P}_c$. We initialize the group parameters as $\boldsymbol{G}_r = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$ and fix the relative extrinsics $\boldsymbol{P}_c$ to the current estimate of the absolute projection matrix of the image. The projection matrix of an image c in group r can be updated after bundle adjustment as the concatenation of the group and image pose

$$\boldsymbol{P}_{cr} = \boldsymbol{P}_c \boldsymbol{G}_r \ . \tag{7.16}$$

For an efficient concatenation of the rotational components of $\boldsymbol{G}_r$ and $\boldsymbol{P}_c$, we rely on quaternions. A larger group size leads to a greater performance benefit due to a smaller relative overhead of computing $\boldsymbol{P}_c \boldsymbol{G}_r$ over $\boldsymbol{P}_c$. Note that even for the case of a group size of two images, we observe a computational benefit in bundle adjustment. In addition, the performance benefit depends on the problem size, as a reduction in the number of cameras affects the cubic computational complexity of direct methods more than the linear complexity of indirect methods (see Section 2.4.2).

7.7 Experimental Evaluation

We run experiments on a large variety of datasets to evaluate both the proposed components and the overall system compared to state-of-the-art incremental (*Bundler*

	# Input Images	# Registered Images			
		Theia	*Bundler*	*VSFM*	*Ours*
Rome	74,394	–	13,455	14,797	20,918
Quad	6,514	–	5,028	5,624	5,860
Dubrovnik	6,044	–	–	–	5,913
Alamo	2,915	582	647	609	666
Ellis Island	2,587	231	286	297	315
Gendarmenmarkt	1,463	703	302	807	861
Madrid Metropolis	1,344	351	330	309	368
Montreal Notre Dame	2,298	464	501	491	506
NYC Library	2,550	339	400	411	453
Piazza del Popolo	2,251	335	376	403	437
Piccadilly	7,351	2,270	1,087	2,161	2,336
Roman Forum	2,364	1,074	885	1,320	1,409
Tower of London	1,576	468	569	547	578
Trafalgar	15,685	5,067	1,257	5,087	5,211
Union Square	5,961	720	649	658	763
Vienna Cathedral	6,288	858	853	890	933
Yorkminster	3,368	429	379	427	456

Table 7.1: Number of registered images for state-of-the-art sparse reconstruction pipelines on large-scale unstructured Internet photo collections [75, 193, 362].

	# **Points** (Avg. Track Length)			
	Theia	*Bundler*	*VSFM*	*Ours*
Rome	–	5.4M	12.9M	5.3M
Quad	–	10.5M	0.8M	1.2M
Dubrovnik	–	–	–	1.35M
Alamo	46K (6.0)	127K (4.5)	124K (8.9)	94K (11.6)
Ellis Island	29K (4.9)	39K (4.1)	61K (5.5)	64K (6.8)
Gendarmenmarkt	87K (3.8)	93K (3.7)	138K (4.9)	123K (6.1)
Madrid Metropolis	47K (5.0)	27K (3.2)	48K (5.2)	43K (6.6)
Montreal Notre Dame	154K (5.4)	135K (4.6)	110K (7.1)	98K (8.7)
NYC Library	66K (4.1)	71K (3.7)	95K (5.5)	77K (7.1)
Piazza del Popolo	36K (5.2)	34K (3.7)	50K (7.2)	47K (8.8)
Piccadilly	197K (4.9)	197K (3.9)	245K (6.9)	260K (7.9)
Roman Forum	261K (4.9)	281K (4.4)	278K (5.7)	222K (7.8)
Tower of London	140K (5.2)	151K (4.8)	143K (5.7)	109K (7.4)
Trafalgar	381K (4.8)	196K (3.7)	497K (8.7)	450K (10.1)
Union Square	35K (5.3)	48K (3.7)	43K (7.1)	53K (8.2)
Vienna Cathedral	259K (4.9)	276K (4.6)	231K (7.6)	190K (9.8)
Yorkminster	143K (4.5)	71K (3.9)	130K (5.2)	105K (6.8)

Table 7.2: Number of reconstructed points for state-of-the-art sparse reconstruction pipelines on large-scale unstructured Internet photo collections [75, 193, 362].

	Time [s]			
	Theia	*Bundler*	*VSFM*	*Ours*
Rome	–	295,200	6,012	10,912
Quad	–	223,200	2,124	3,791
Dubrovnik	–	–	–	3,821
Alamo	874	22,025	495	882
Ellis Island	94	12,798	240	332
Gendarmenmarkt	202	465,213	412	627
Madrid Metropolis	95	21,633	203	251
Montreal Notre Dame	207	112,171	418	723
NYC Library	194	36,462	327	420
Piazza del Popolo	89	33,805	275	380
Piccadilly	1,427	478,956	1,236	1,961
Roman Forum	1,302	587,451	748	1,041
Tower of London	201	184,905	497	678
Trafalgar	1,494	612,452	3,921	5,122
Union Square	131	56,317	556	693
Vienna Cathedral	764	567,213	899	1,244
Yorkminster	164	34,641	661	997

Table 7.3: Runtimes for state-of-the-art sparse reconstruction pipelines on large-scale unstructured Internet photo collections [75, 193, 362].

[310], *VisualSFM* [365]) and global SFM systems (*DISCO* [75], *Theia* [326][1]). The 17 datasets contain a total of 144,953 unordered Internet photos, distributed over a large area and with highly varying camera density. In addition, Quad [75] has ground-truth camera locations. Throughout all experiments, we use RootSIFT features and match each image against its 100 nearest neighbors using a vocabulary tree trained on unrelated datasets. To ensure comparability between the different methods, all evaluated pipelines use the same correspondence search and the timing results are produced on the same 48-core 2.7GHz machine with 256GB RAM.

7.7.1 Next Best View Selection

A synthetic experiment (see Figure 7.3) evaluates how well the score $\mathcal{S}$ reflects the number and distribution of points. We use $L = 6$ pyramid levels, and we generate Gaussian-distributed image points with spread σ and location μ. A larger σ and a more central μ corresponds to a more uniform distribution and correctly produces a higher score. Similarly, the score is dominated by the number of points when there are few and otherwise by their distribution in the image. Another experiment (see Figure 7.4) compares our method (*Pyramid*) to existing strategies in terms of the reconstruction error. The other methods are *Number* [310], which maximizes the number of triangulated points, and *Ratio* which maximizes the ratio of visible over

[1]Results for *Theia* kindly provided by the authors [326].

	Avg. Reproj. Error [px]			
	Theia	*Bundler*	*VSFM*	*Ours*
Rome	–	–	–	0.81
Quad	–	–	–	0.73
Dubrovnik	–	–	–	0.71
Alamo	1.47	2.29	0.70	0.68
Ellis Island	2.41	2.24	0.71	0.70
Gendarmenmarkt	2.19	1.59	0.71	0.68
Madrid Metropolis	1.48	1.62	0.59	0.60
Montreal Notre Dame	2.01	1.92	0.88	0.81
NYC Library	1.89	1.84	0.67	0.69
Piazza del Popolo	2.11	1.76	0.76	0.72
Piccadilly	2.33	1.79	0.79	0.75
Roman Forum	2.07	1.66	0.69	0.70
Tower of London	1.86	1.54	0.59	0.61
Trafalgar	2.09	2.07	0.79	0.74
Union Square	2.36	3.22	0.68	0.67
Vienna Cathedral	2.45	1.69	0.80	0.74
Yorkminster	2.38	2.61	0.72	0.70

Table 7.4: Average reprojection error results for state-of-the-art sparse reconstruction pipelines on large-scale unstructured Internet photo collections [75, 193, 362].

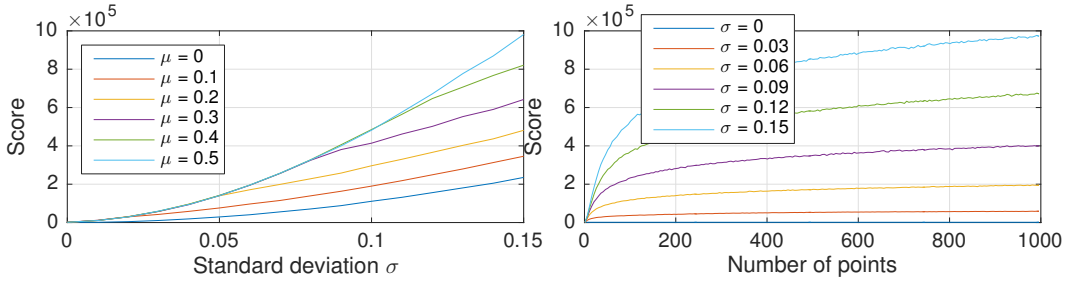

Figure 7.3: Next best view scores using our proposed hierarchical data structure for truncated Gaussian distributed points $\boldsymbol{x}_j \in [0,1] \times [0,1]$ with mean μ and standard deviation σ. Left: score $\mathcal{S}$ with respect to uniformity. Right: score $\mathcal{S}$ with respect to number of points for $\mu = 0.5$.

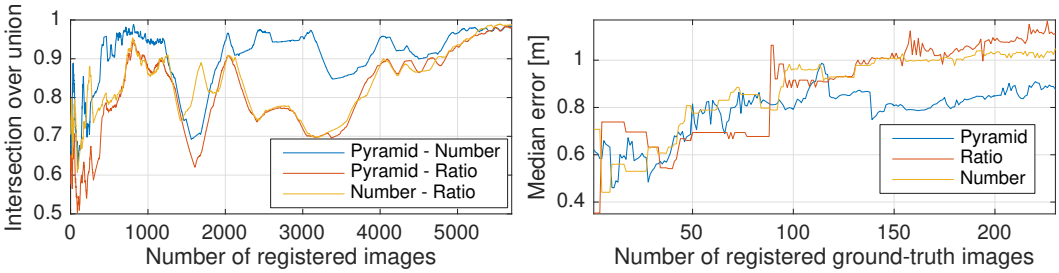

Figure 7.4: Next best view results for Quad dataset [75] during incremental Structure-from-Motion. Left: shared number of registered images. Right: reconstruction error in meters by aligning reconstruction to ground truth.

	#Points	#Elements	Avg. Track Length	#Samples
Bundler	713,824	5.58 M	7.824	136.34 M
Non-recursive				
Exhaustive	861,591	7.64 M	8.877	120.44 M
RANSAC, η_1	861,591	7.54 M	8.760	3.89 M
RANSAC, η_2	860,422	7.46 M	8.682	3.03 M
Recursive				
Exhaustive	894,294	8.05 M	9.003	145.22 M
RANSAC, η_1	902,844	8.03 M	8.888	12.69 M
RANSAC, η_2	906,501	7.98 M	8.795	7.82 M

Table 7.5: Robust triangulation results for Dubrovnik dataset [193] using our proposed sampling-based approach with the confidence parameters $\eta_1 = 0.99$ and $\eta_2 = 0.5$.

potentially visible points. After each image registration, we measure the number of registered images shared between the strategies (intersection over union) and the reconstruction error as the median distance to the ground-truth camera locations. While all strategies converge to the same set of registered images, our method produces the most accurate reconstruction by choosing a better registration order for the images.

7.7.2 Robust and Efficient Triangulation

The experiment on the Dubrovnik dataset in Figure 7.5 and Figure 7.5 evaluates our method on 2.9M feature tracks composed from 47M verified matches. We compare against *Bundler* and an exhaustive strategy that samples all pairwise combinations in a track. We set $\alpha = 2°$, $t = 8$px, and $\epsilon_0 = 0.03$. To avoid combinatorial explosion, we limit the exhaustive approach to 10K iterations, i.e. $\epsilon_{min} \approx 0.02$ with

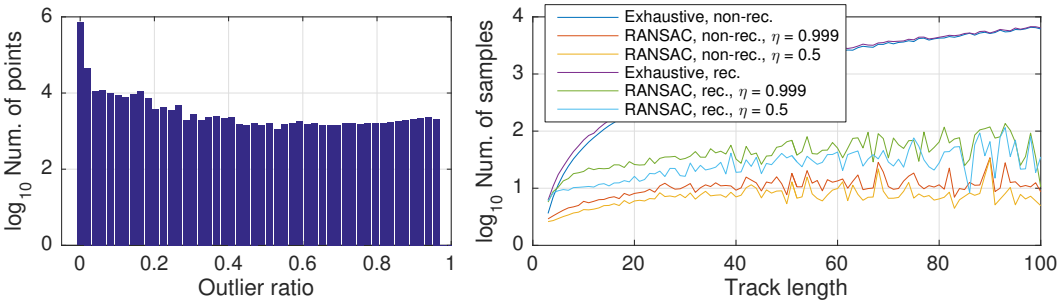

Figure 7.5: Triangulation statistics for Dubrovnik dataset [193]. Left: outlier ratio distribution of feature tracks. Right: average number of samples required to triangulate point with certain track length.

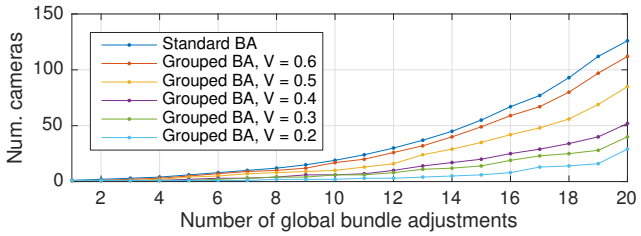

Figure 7.6: Number of cameras for Dubrovnik dataset [193] in standard and grouped bundle adjustment using the parameters $\epsilon_r = 0.02$, $S = 10$, and varying scene overlap V.

$\eta = 0.999$. The diverse inlier ratio distribution (as determined with the recursive, exhaustive approach) evidences the need for a robust triangulation method. Our proposed recursive approaches recover significantly longer tracks and overall more track elements than their non-recursive counterparts. Note that the higher number of points for the recursive RANSAC-based methods corresponds to the slightly reduced track lengths. The RANSAC-based approach yields just marginally inferior tracks but is much faster (10-40x). By varying η, it is easy to balance speed against completeness.

7.7.3 Redundant View Mining

We evaluate redundant view mining on an unordered collection of densely distributed images. Figure 7.6 shows the growth rate of the parameterized cameras in global bundle adjustment using a fixed number of bundle adjustment iterations. Depending on the enforced scene overlap V, we can reduce the time for solving the reduced camera system by a significant factor. The effective speedup of the total runtime is 5% ($V = 0.6$), 14% ($V = 0.3$) and 32% ($V = 0.1$), while the average reprojection error degrades from $0.26px$ (standard bundle adjustment) to $0.27px$, $0.28px$, and $0.29px$, respectively. The reconstruction quality is comparable for all choices of

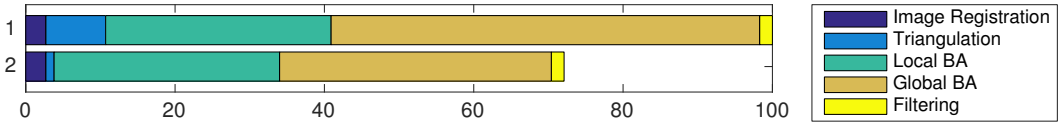

Figure 7.7: Average relative runtimes using standard global bundle adjustment and exhaustive, recursive triangulation (1), and our proposed grouped bundle adjustment and sampling-based recursive triangulation (2). Runtime for initialization and all next best view selection strategies is smaller than 0.1%.

$V > 0.3$ and increasingly degrades for a smaller V. Using $V = 0.4$, the runtime of the entire pipeline for Colosseum reduces by 36% yet results in an equivalent reconstruction.

7.7.4 System Performance

Table 7.1, Table 7.2, Table 7.3, and Table 7.4 demonstrate the performance of the overall system and thereby also evaluate the performance of the individual proposed components of the system. For each dataset, we report the largest reconstructed components, which are also visualized in Figure 7.2, Figure 7.8, and Figure 7.9, and Figure 7.10. *Theia* is the fastest method, while our method achieves slightly worse timings than *VisualSFM* and is more than 50 times faster than *Bundler*. Figure 7.7 shows the relative timings of the individual modules. For all datasets, we significantly outperform any other method in terms of completeness, especially for the larger models. Importantly, the increased track lengths result in higher redundancy in bundle adjustment. In addition, we achieve the best pose accuracy for the *Quad* dataset: *DISCO* 1.16m, *Bundler* 1.01m, *VisualSFM* 0.89m, and *Ours* 0.85m. Figure 7.11 shows an overlay of the camera locations in the Rome dataset with an aerial image. Note that the Rome reconstruction spans several hundreds of meters and our reconstruction suffers from almost no pose errors due to drift.

7.8 Summary

This chapter proposed several improvements over the traditional incremental sparse reconstruction paradigm that overcomes key challenges to make a further step towards a general-purpose image-based 3D modeling system. The proposed components of the algorithm improve the state of the art in terms of completeness, robustness, accuracy, and efficiency. Comprehensive experiments on challenging large-scale datasets demonstrated the performance of the individual components and the overall system.

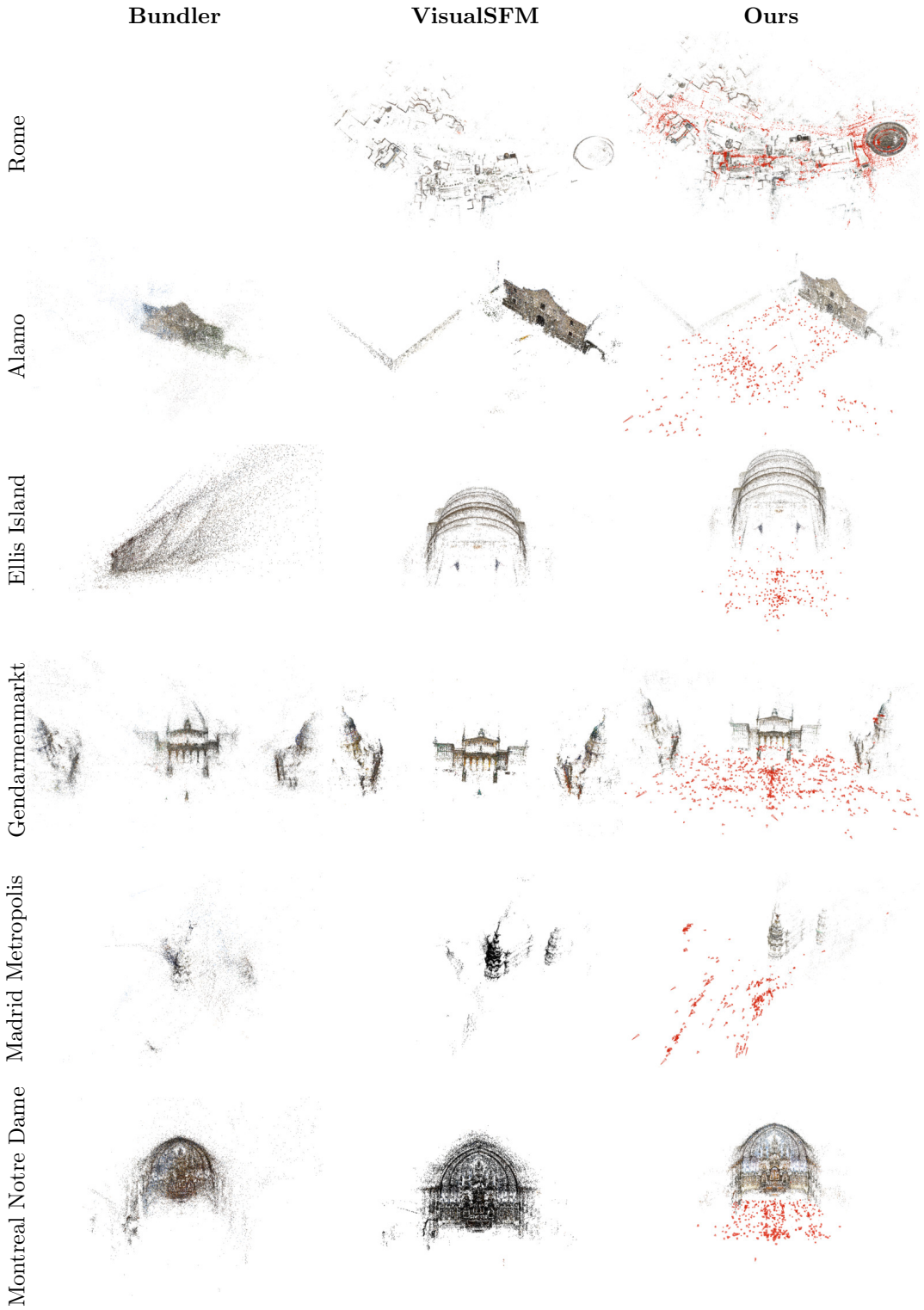

Figure 7.8: Visualization of sparse models produced from unstructed Internet images for the different incremental reconstruction pipelines *Bundler*, *VisualSFM*, and *Ours*. More results in Figure 7.9 and Figure 7.10.

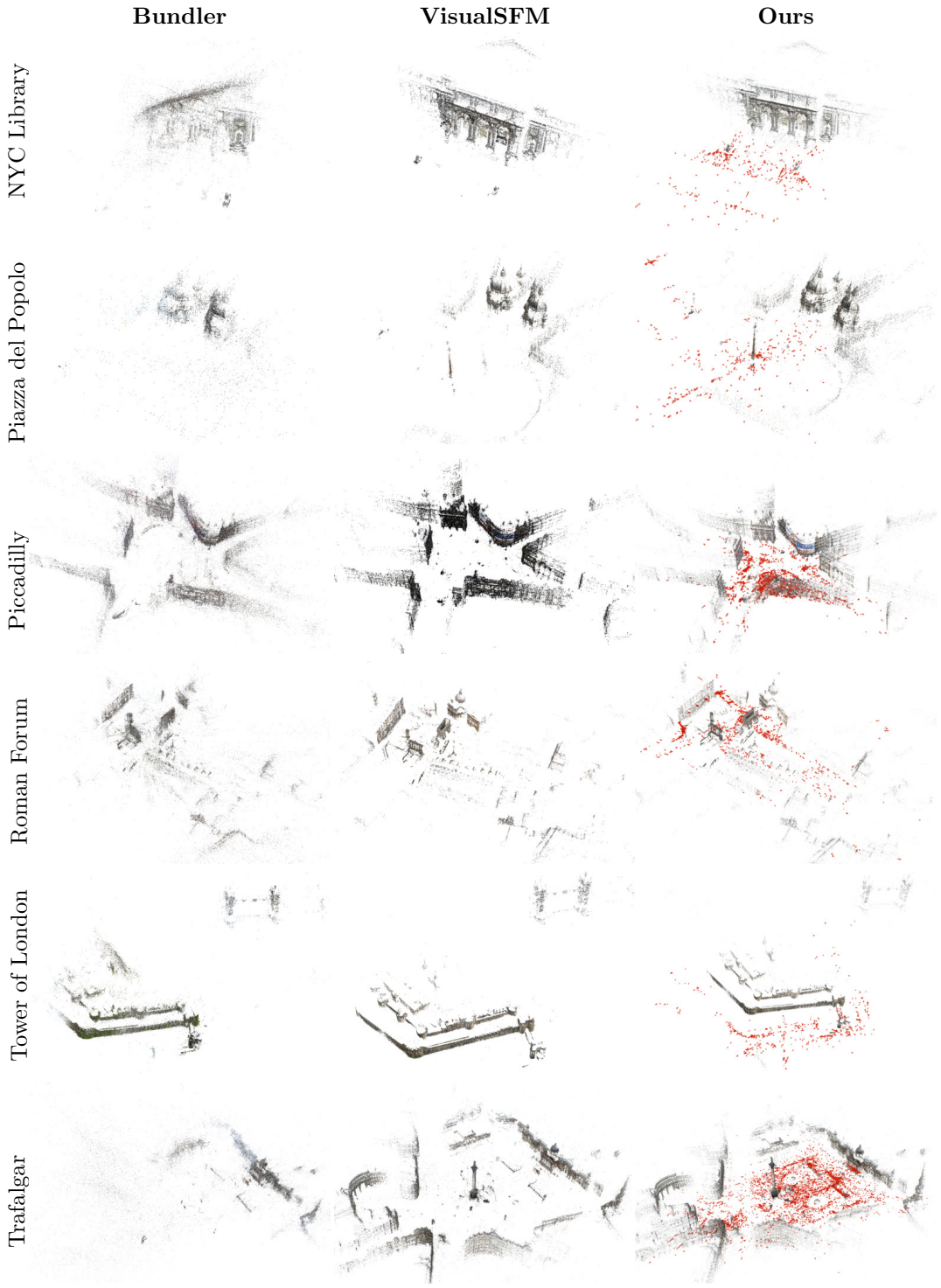

Figure 7.9: Visualization of sparse models produced from unstructed Internet images for the different incremental reconstruction pipelines *Bundler*, *VisualSFM*, and *Ours*. More results in Figure 7.8 and Figure 7.10.

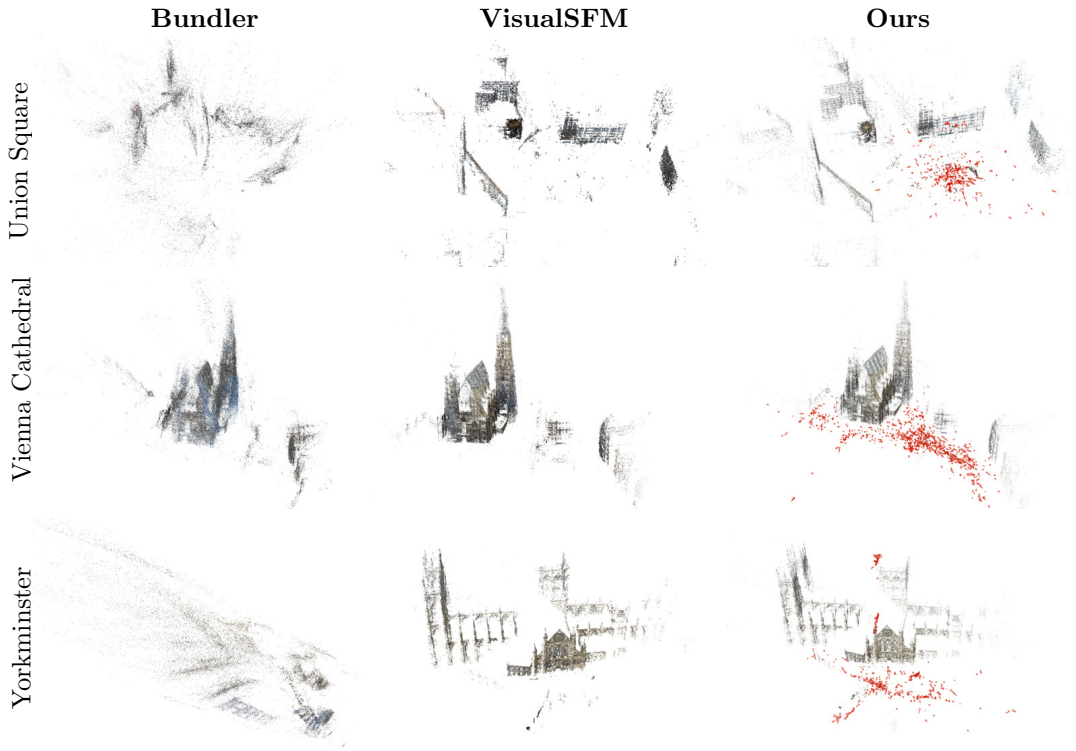

Figure 7.10: Visualization of sparse models produced from unstructed Internet images for the different incremental reconstruction pipelines *Bundler*, *VisualSFM*, and *Ours*. More results in Figure 7.8 and Figure 7.9.

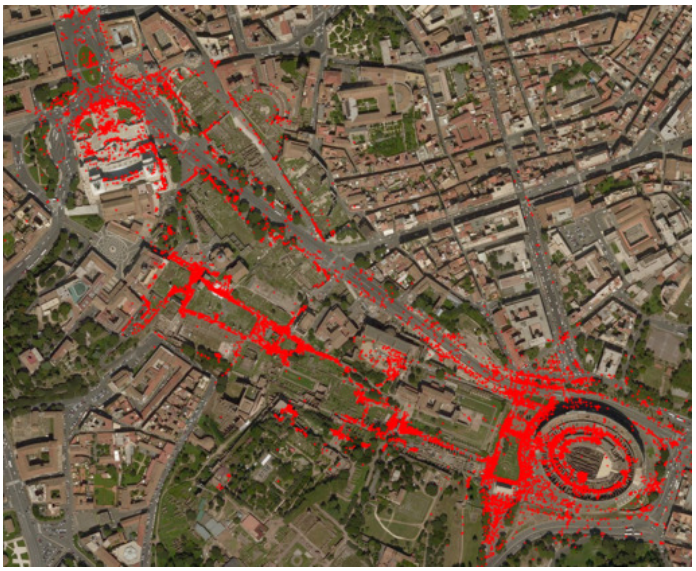

Figure 7.11: Camera locations of our sparse reconstruction of Rome in red accurately aligned with rectified aerial image from Apple Maps.

Part IV

Dense Reconstruction

8 Learning to Fuse Proposals from Multiple Scanline Optimizations in Semi-Global Matching

This chapter focuses on the robust and accurate dense stereo reconstruction using the Semi-Global Matching algorithm. Semi-Global Matching (SGM) is a popular stereo matching algorithm proposed by Hirschmüller [134] that has found widespread use in applications ranging from 3D mapping [135, 261], robot and drone navigation [141, 281], and assisted driving [90]. The technique is efficient and parallelizable and suitable for real-time stereo reconstruction on FPGAs and GPUs [25, 105, 141]. SGM incorporates regularization in the form of smoothness priors, similar to global stereo methods but at lower computational cost. The main idea in SGM is to approximate a 2D Markov random field (MRF) optimization problem with several independent 1D scanline optimization problems corresponding to multiple canonical scanline directions in the image (typically 4 or 8). These 1D problems are optimized exactly using dynamic programming (DP) by aggregating matching costs along the multi-directional 1D scanlines. The costs of the minimum cost paths for the various directions are then summed up to compute a final aggregated cost per pixel. Finally, a winner-take-all (WTA) strategy is used to select the disparity with the minimum aggregated cost at each pixel.

Summation of the aggregated costs from multiple directions and the final WTA strategy are both ad-hoc steps in SGM that lack proper theoretical justification. The summation was originally proposed to reduce 1D streaking artifacts [134] but is ineffective for weakly textured slanted surfaces and also generally inadequate when multiple scanline optimization solutions are inconsistent.

Our main motivation in this work is to devise a better strategy to fuse 1D scanline optimization costs from multiple directions. We argue that the scanline optimization solutions should be considered as independent disparity map proposals and the WTA step should be replaced by a more general fusion step. Figure 8.1 shows two of the eight scanline optimization solutions for the Adirondack pair from the Middlebury 2014 dataset [276]. While both solutions suffer from directional bias due to their respective propagation directions, each solution is accurate in certain image regions where the other one is inaccurate. For example, the horizontal pass produces accurate disparities near the left occlusion boundaries of the chair, whereas the diagonal pass performs better on the right occlusion edges. In those regions, the final SGM solution is slightly worse. The error plot in Figure 8.1 quantifies this observation for the entire image. Whereas SGM is more accurate than each scanline optimization

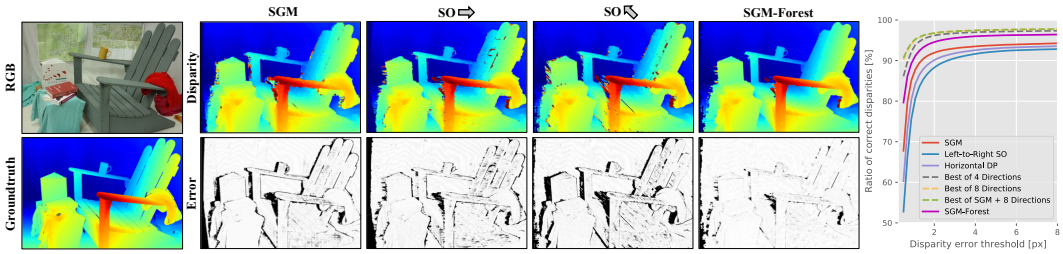

Figure 8.1: **Fusing Multiple Scanline Proposals.** Left: Visualization of disparity maps from SGM, two (out of 8) scanline optimizations (SO) and our proposed SGM-Forest method. While SGM is more accurate than each SO on the whole image, each SO solution is better in some specific areas. SGM-Forest identifies the best SO proposal at each pixel and produces the best overall result. Right: Error plots for SGM, SO and SGM-Forest solutions (solid line) and upper bounds for oracles making optimal selections (dotted line). In this example, SGM-Forest gets close to the upper bounds.

individually, the *joint accuracy* of all scanlines is much higher than SGM. Here, joint accuracy refers to a theoretical upper bound of the achievable accuracy of an oracle, which has access to ground truth and selects the best out of all the scanline solution proposals.

Based on this insight, we formulate the fusion step as the task of selecting the best amongst all the scanline optimization proposals at each pixel in the image. We propose to solve this task using supervised learning. Our method, named *SGM-Forest*, uses a per-pixel random forest classifier. As shown in Figure 8.1, it gets close to the theoretical upper bound and significantly outperforms SGM.

The per-pixel classifier in SGM-Forest is trained on a low-dimensional input feature that encodes a sparse set of aggregated cost samples. Specifically, these cost values are sampled from the cost volumes computed during the scanline optimization passes. The sampling locations correspond to the disparity candidates for all scanline directions at each pixel. In fact, the proposals need not be limited to the usual scanline directions. Including the SGM solution and two horizontal scanline optimization solutions from the right image as additional proposals improves accuracy further. We train and evaluate the forest using ground truth disparity maps provided by stereo benchmarks [276, 278, 290]. At test time, the random forest predicts the disparity proposal to be selected at each pixel. Inference is fast and parallelizable and thus has small overhead. The forest automatically outputs per-pixel posterior class probabilities from which suitable confidence maps are derived, for use in a final disparity refinement step.

Thus, the main contribution in this chapter is a new, efficient learning-based fusion method for SGM that directly predicts the best amongst all the 1D scanline optimization disparity proposals at each pixel based on a small set of scanline op-

timization costs. SGM-Forest uses this fusion method instead of SGM's sum-based aggregation and WTA steps and our results shows that it consistently outperforms SGM in many different settings. We evaluate SGM-Forest on three stereo benchmarks. Currently, it is ranked 1st on ETH3D [290] and is competitive on Middlebury 2014 [276] and KITTI 2015 [106]. We run extensive ablation studies and show that our method is extremely robust to dataset bias. It outperforms SGM even when the forests are trained on datasets from different domains.

8.1 Related Work

In this section, we review SGM and learning-based methods for stereo. We then compare and contrast our proposed SGM-Forest to closely related works.

SGM was built on top of earlier methods such as 1D scanline optimization [233, 278, 381] and dynamic programming stereo [212] with a new aggregation scheme to fix the lack of proper 2D regularization in those methods. However, a proper derivation of the aggregation step remained elusive until Drory et al. [81] showed its connection to non-loopy belief propagation on a special graph structure. Veksler [350] and Bleyer et al. [33] advanced dynamic programming stereo to tree structures connecting all pixels, but those methods have not been widely adopted. SGM has been extended to improve speed and accuracy [25, 84, 105, 131, 132, 136, 141, 279, 304] reduce memory usage [137, 141, 182], and used to compute optical flow [331, 371].

Scharstein and Pal [277] were one of the first to use learning in stereo. They trained a conditional random field (CRF) on Middlebury 2005–06 datasets to model the relationship between the CRF's penalty terms and local intensity gradients in the image. The KITTI and Middlebury 2014 [106, 276] benchmarks encouraged much work on learning. In particular, CNNs have been trained to compute robust matching costs [61, 202, 352]. Zbontar and Lecun were the first; they proposed MC-CNN [352] and reported higher accuracy when using MC-CNN in conjunction with SGM for regularization and additional post-processing steps. Newer methods combined MC-CNN with better optimization but as a result are much slower. The method of Taniai et al. [330] uses iterative graph cut optimization and MC-CNN-acrt [352] and is the current state of the art on Middlebury.

End-to-end training of CNNs is nowadays popular on KITTI [109, 163, 210, 237] but is almost never tested on Middlebury. In one rare case, moderate results were reported [168]. In contrast, our method generalizes across three benchmarks [106, 276, 290] on which it consistently outperforms baseline SGM. Furthermore, we train three separate models on Middlebury 2005–06, KITTI, and ETH3D. All three outperform SGM when tested on the Middlebury 2014 training set. SGM-Net [293] is a CNN-based method for improving SGM. SGM-Net performs more accurate scanline optimization by using a CNN to predict the parameters of the underlying scanline optimization objective. In contrast, we use regular scanline optimization but propose a learning-based fusion step using random forests.

Stereo matching has been solved by combining multiple disparity maps using MRF

fusion moves [35, 183, 330]. Fusion moves are quite general, but computationally expensive and need many iterations. This makes them slow. Alternatively, multiple disparity maps can also be fused using learning, based on random forests [315] and CNNs [246]. Other methods first predict confidence maps [143], often via learning [11, 114, 238, 247, 248, 314], and then use the predicted confidence values in a greedy fashion to combine multiple solutions. Drory et al. [81] proposed a different uncertainty measure for SGM but do not show how to use it. Unlike MRF fusion moves [183], our fusion method is not general. It combines a specific number and specific type of proposals but does so in a single efficient step.

Michael et al. [214] and Poggi and Mattoccia [247] (SGM-RF) proposed replacing SGM's sum-based aggregation with a weighted sum, setting smaller weights in areas with 1D streaking artifacts. The former work [214] proposes using global weights per scanline direction. SGM-RF [247] is more effective as it predicts per-pixel weights for each scanline direction using random forests based on disparity-based features. However, SGM-RF was not evaluated on the official test sets of the Middlebury 2014 and KITTI 2015 benchmarks. Mac Aodha et al. [11] also used random forests to fuse optical flow proposals using flow-based features.

Our SGM-Forest differs from these methods in several ways. First, it avoids predicting confidence separately for each proposal [11, 247] but instead directly predicts the best proposal at each pixel. The forest is invoked only once at each pixel and has information from all the scanline directions. This makes inference more effective. Furthermore, the features used by our forest are directly obtained by sampling the aggregated cost volumes of each scanline optimization problem at multiple selective disparities. This is much more effective than handcrafted disparity-based features [247, 315]. Finally, our confidence maps derived from posterior class probabilities are normalized and hence better for refining the disparities during post-processing. Häusler et al. [114] aim to detect unreliable disparities and suggest adding SGM's aggregated (summed) costs to their handcrafted disparity-based features. In contrast, we focus on fusing multiple proposals and propose to sample all the cost volumes for each independent scanline optimization at multiple disparities to better exploit contextual information.

8.2 Semi-Global Matching

We now review SGM as proposed by Hirschmüller [135] for approximate energy minimization of a 2D Markov Random Field (MRF)

$$E(D) = \sum_{\boldsymbol{p}} C_{\boldsymbol{p}}(d_{\boldsymbol{p}}) + \sum_{\boldsymbol{p},\boldsymbol{q}\in\mathcal{N}} V(d_{\boldsymbol{p}}, d_{\boldsymbol{q}}) \ , \tag{8.1}$$

where $C_{\boldsymbol{p}}(d)$ is a unary data term that encodes the penalty of assigning pixel $\boldsymbol{p} \in \mathbb{R}^2$ to disparity $d \in \mathcal{D} = \{d_{\min}, \ldots, d_{\max}\}$. The pairwise smoothness term $V(d, d')$ penalizes disparity differences between neighboring pixels $\boldsymbol{p}$ and $\boldsymbol{q}$. In SGM, the

term V is chosen to have the following specific form

$$V(d,d') = \begin{cases} 0 & \text{if } d = d' \\ P_1 & \text{if } |d-d'| = 1 \\ P_2 & \text{if } |d-d'| \geq 2 \ , \end{cases} \tag{8.2}$$

which favors first-order smoothness, i.e., has a preference for fronto-parallel surfaces. Minimizing the 2D MRF is NP-hard. Therefore, SGM instead solves multiple scanline optimization problems, each of which involves solving the 1D version of Equation 8.1 along 1D scanlines in 8 cardinal directions $\boldsymbol{r} = \{(0,1),(0,-1),(1,0),...\}$. For each direction $\boldsymbol{r}$, SGM computes an aggregated matching cost

$$L_{\boldsymbol{r}}(\boldsymbol{p},d) = C_{\boldsymbol{p}}(d) + \min_{d' \in \mathcal{D}}(L_{\boldsymbol{r}}(\boldsymbol{p}-\boldsymbol{r},d') + V(d,d')) \ . \tag{8.3}$$

The definition of $L_{\boldsymbol{r}}(\boldsymbol{p},d)$ is recursive and is typically started from a pixel on the image border. An aggregated cost volume $S(\boldsymbol{p},d)$ is finally computed by summing up the eight individual aggregated cost volumes

$$S(\boldsymbol{p},d) = \sum_{\boldsymbol{r}} L_{\boldsymbol{r}}(\boldsymbol{p},d) \ . \tag{8.4}$$

The final disparity map is obtained using a WTA strategy by selecting per-pixel minima in the aggregated cost volume

$$d_{\boldsymbol{p}} = \arg\min_{d} S(\boldsymbol{p},d) \ . \tag{8.5}$$

The steps in Equation 8.4 and Equation 8.5 are accurate when the costs from different scanline directions are mostly consistent with respect to each other. However, these steps are likely to fail as the scanlines become more inconsistent. To overcome this problem, we propose a novel fusion method to robustly compute the disparity $d_{\boldsymbol{p}}$ from the multiple scanline costs $L_{\boldsymbol{r}}(\boldsymbol{p},d)$.

8.3 Learning To Fuse Scanline Optimization Solutions

We start by analyzing some difficult examples for scanline optimization in order to motivate our fusion method and then describe the method in detail.

8.3.1 Scanline Optimization Analysis

Figure 8.2 shows four scanlines from the left ADIRONDACK image with the corresponding x–d slices of the the unary cost C and the four horizontal and vertical aggregated scanline costs $L_{\boldsymbol{r}}$ alongside their respective WTA solutions. Notice the patterns in the $L_{\boldsymbol{r}}$ cost slices for the different passes. When the smoothness prior is effective, the noisy unary costs get filtered, producing strong minima at the correct disparities. However, when the unary costs are weak and the prior is ineffective, multiple noisy minima are present or the minimum is at an incorrect location. We now investigate these problematic cases in further detail.

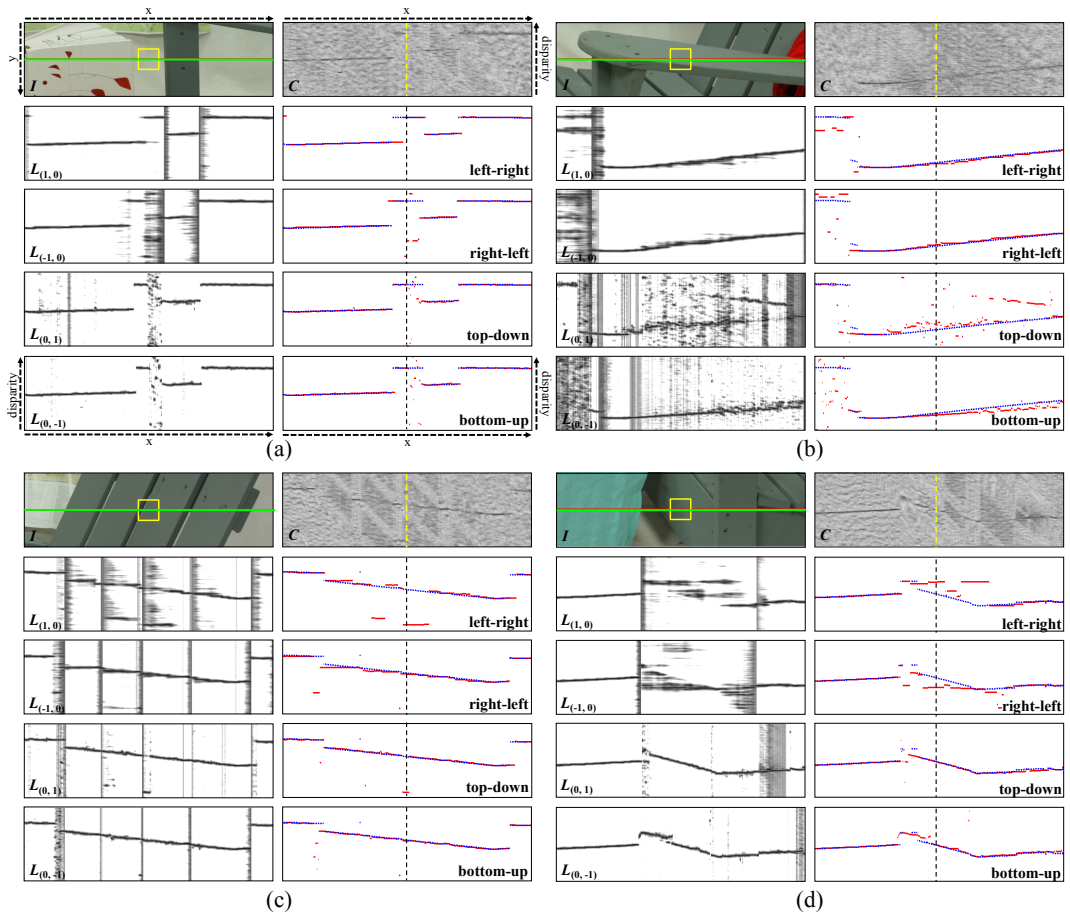

Figure 8.2: **1D Scanline Optimization Costs.** Each of the four subfigures shows the following – Top Left: Image and reference scanline section in green centered around yellow patch. Top Right: x–d slice of unary cost volume C along the reference scanline and ray of reference patch center in yellow. Bottom: Aggregated costs $L_{\boldsymbol{r}}$ for four scanline directions on the left and the corresponding disparities on the right. The WTA solution is shown in red whereas the ground truth disparity is in blue.

Weak Texture

Figure 8.2 (a)–(d) focus on weakly textured image patches. Whenever the unary cost is weak, the smoothness prior in the 1D optimization favors propagating several equally likely disparity estimates along the propagation direction. This effect is seen clearly on the vertical wooden plank in Figure 8.2(d) in the horizontal passes. Here, the left-right propagation continues the solution from the left occlusion boundary to the right, while the right-left solution continues from the corner of the chair to the left. In contrast, the two vertical passes are in agreement at the correct disparity as the surface *along that propagation direction* is indeed fronto-parallel.

Slanted Surface

Figure 8.2 (b), (c), (d) show examples of weakly textured slanted surfaces, where the 1D scanline solutions are typically biased and jump at random pixel locations, leading to inconsistent solutions in different scanlines. A prominent example is the arm rest in Figure 8.2 (b), where the left-right pass underestimates the disparity, whereas the right-left and bottom-up passes overestimate the disparity. In this case, there is no clear outlier in the solution but final cost summation leads to a biased estimate. Notice also the asymmetry in the two vertical passes where the bottom-up direction has a much more consistent solution while the top-down solution jumps at random locations. On weakly textured slanted surfaces, adjacent scanlines solutions are mostly inconsistent leading to noisy disparity maps and well-known streaking artifacts.

Occlusion

Figure 8.2 (a) is centered around a region which is occluded in the right image. In this case, the unary cost is invalid and the only pass producing a correct prediction is the left-to-right direction. Here, the occluded surface is fronto-parallel and the smoothness prior is likely to propagate the correct disparity to the occluded region. Typically, only a small subset of scanlines results are correct in occluded areas, whereas SGM's standard cost summation is not robust and therefore produces gross outliers (see Figure 8.1).

Repetitive Structure

The wooden planks on the chair's backrest in Figure 8.2 (c) are repetitive and produce multiple ambiguous local cost minima. In this example, the solutions of the left-right and top-down directions are incorrectly estimated, since the centered patch is almost identical to the symmetric patch on the right-most wooden plank. Notice also that the right-left and bottom-up directions are much less susceptible to this specific ambiguity problem.

These examples show that the joint distribution of aggregated costs over the disparity range at each pixel appears to provide strong clues about which scanline

proposal or which subset of proposals are likely to be correct. This insight forms the basis of our fusion model which is described next.

8.3.2 Definition of Fusion Model

The disparities of the different scanline solutions are often inconsistent, especially in areas of weak data cost. Yet, in almost all cases there is at least one scanline that is either correct or is very close to the correct solution. The main challenge for robust and accurate scanline fusion is to identify the scanlines which agree on the correct estimate. In our proposed approach, we cast the fusion of scanlines as a classification problem that chooses the optimal estimate from the given set of candidate scanlines. Typically, the pattern at which specific scanlines perform well is consistent and repeatable. We aim to encode these patterns into rules that can identify the correct solution from a given set of candidate solutions. However, manually hand-crafting these rules is unfeasible and error-prone, which is why we resort to automatically learning these rules from training data in a supervised fashion. To facilitate the learning of these rules, we provide the model with discriminative signals that allow for a robust and efficient disparity prediction. Our proposed model takes sparse samples from a set of proposal cost volumes $K_n(\boldsymbol{p}, d)$ (e.g., the optimized scanline costs $L_{\boldsymbol{r}}(\boldsymbol{p}, d)$) and concatenates them into a per-pixel feature vector $\boldsymbol{f}_{\boldsymbol{p}}$. This feature vector is then fed into a learned model that predicts a disparity estimate $\hat{d}_{\boldsymbol{p}}$ together with a posterior probability $\hat{\rho}_{\boldsymbol{p}}$, which we use as a confidence measure for further post-processing.

More specifically, our model is defined as

$$(\hat{d}_{\boldsymbol{p}}, \hat{\rho}_{\boldsymbol{p}}) = F(f_{\boldsymbol{p}}) \text{ with } d_{\boldsymbol{p}} \in \mathbb{R}_0^+,\, \rho_{\boldsymbol{p}} \in [0,1],\, \boldsymbol{f}_{\boldsymbol{p}} \in \mathbb{R}^{N+N^2} \;, \tag{8.6}$$

where N is the number of proposal costs $K_n(\boldsymbol{p}, d)$ under consideration. For all $n = 1...N$ proposals $K_n(\boldsymbol{p}, d)$, the feature $\boldsymbol{f}_{\boldsymbol{p}}$ stores the location of its per-pixel WTA solution $d^*_{\boldsymbol{p}}(n) = \arg\min_d K_n(\boldsymbol{p}, d)$ and the corresponding costs $K_m(\boldsymbol{p}, d^*_{\boldsymbol{p}}(n))$ in all proposals $m = 1...N$. Overall, the feature is composed of N WTA solutions and the N^2 sparsely sampled costs. For each disparity proposal $d^*_{\boldsymbol{p}}(n)$, we thereby encode its relative significance with respect to the other proposals in a compact representation. The intuition is that when multiple proposals agree, their minima $d^*_{\boldsymbol{p}}(n)$ are close and their respective costs $K_m(\boldsymbol{p}, d^*_{\boldsymbol{p}}(n))$ are low.

Note that the naïve approach of concatenating the per-pixel costs of all proposals into a feature vector is not feasible for two reasons. First, we want a light-weight feature representation and model with small runtime overhead with respect to regular SGM. However, the naïve approach would result in a very high-dimensional feature representation of size $N \cdot |\mathcal{D}|$ (e.g., $8 \cdot 256 = 2048$ for 256 disparity candidates and 8 scanlines), which would require a complex model and eliminate the computational efficiency of SGM. In contrast, our proposed feature vector is only $8 + 8^2 = 72$-dimensional in case of 8 scanline proposals. Second, we strive to learn a generalizable model, which uses a fixed-size feature representation during training and inference even though the disparity range $\mathcal{D}$ may vary between scenes. An

alternative to our proposed approach would be to resample the cost proposals to a small, constant size and concatenate them into a feature vector. The major drawback of the resampling based approach is that it causes a drop in accuracy. the loss of accuracy in disparity values due to the resampling. In summary, our proposed feature encodes discriminative signals for our classification task without sacrificing efficiency, compactness, or accuracy.

8.3.3 Random Forests for Disparity and Confidence Prediction

Given ground truth disparities, there are many ways to learn the model $F(\boldsymbol{f_p})$ using supervised learning. The first principal design decision is whether to pose the problem as a classification or regression task. Arguably, classification problems are often considered as easier to solve. As shown in Figure 8.1, at least one of the different scanline solutions is often accurate. We therefore chose to formulate a N-class classification task that predicts the best solution from the set of candidates $d^*_{\boldsymbol{p}}(n)$. This approach gave much better results than modeling the problem as a regression task. The second principal design decision is the specific type of classifier to use, e.g., k-NN, support vector machines, decision trees, neural nets, etc. In our experiments, random forests provided the best trade-off between accuracy and efficiency (see Section 8.4.2 and Table 8.1).

At test time, we first perform 1D scanline optimization to construct the proposal cost volumes $K_n(\boldsymbol{p}, d)$, from which we build the per-pixel feature vectors $\boldsymbol{f_p}$. In the second stage, we simply feed the feature vectors $\boldsymbol{f_p}$ of all pixels $\boldsymbol{p}$ through our model to obtain a posterior probability $\rho_{\boldsymbol{p}}(n)$ for each proposal n. We select the proposal with the maximum posterior probability $n^*_{\boldsymbol{p}} = \arg\max_n \rho_{\boldsymbol{p}}(n)$ as our initial disparity estimate $d^*_{\boldsymbol{p}}(n^*)$ for pixel $\boldsymbol{p}$. To further refine this initial estimate, we find the subset of disparity proposals close to the initial estimate and their corresponding posteriors:

$$\mathcal{D}^*_{\boldsymbol{p}} = \{(d^*_{\boldsymbol{p}}(k), \rho_{\boldsymbol{p}}(k)) \,|\, k = 1...N \wedge |d^*_{\boldsymbol{p}}(k) - d^*_{\boldsymbol{p}}(n^*)| < \epsilon_d\} \tag{8.7}$$

When multiple scanlines agree on a solution, the inlier set $\mathcal{D}^*_{\boldsymbol{p}}$ contains multiple elements, even for small disparity thresholds ϵ_d. The final per-pixel disparity estimate $\hat{d}_{\boldsymbol{p}}$ and confidence measure $\hat{\rho}_{\boldsymbol{p}}$ are computed as

$$\hat{d}_{\boldsymbol{p}} = \frac{\sum_k \rho_{\boldsymbol{p}}(k)\, d^*_{\boldsymbol{p}}(k)}{\sum_k \rho_{\boldsymbol{p}}(k)} \;\text{ and }\; \hat{\rho}_{\boldsymbol{p}} = \sum_k \rho_{\boldsymbol{p}}(k) \tag{8.8}$$

Note that the final disparity estimate has sub-pixel precision. Moreover, all steps are fully parallelizable on the pixel level and therefore suitable for real-time FPGA implementations (see Section 8.4.2 and Section 8.4.5). Next, we will describe our spatial edge-aware filtering scheme for disparity refinement.

8.3.4 Confidence-based Spatial Filtering

The random forest produces a per-pixel estimate for disparity and confidence. In a final filtering step, we now enhance the spatial smoothness of the disparity and

confidence maps. Towards this goal, we define the adaptive local neighborhood

$$\mathcal{N}_{\boldsymbol{p}} = \{\boldsymbol{q} \,|\, \|\boldsymbol{q} - \boldsymbol{p}\| < \epsilon_{\boldsymbol{p}} \wedge \hat{\rho}_{\boldsymbol{q}} > \epsilon_{\rho} \wedge |I(\boldsymbol{p}) - I(\boldsymbol{q})| < \epsilon_I\} \tag{8.9}$$

centered around each pixel $\boldsymbol{p}$, where $I(\boldsymbol{q})$ is the image intensity at pixel $\boldsymbol{q}$. The filtered disparity and confidence estimates are finally given as

$$\bar{d}_{\boldsymbol{p}} = \text{median}\, \hat{d}_{\boldsymbol{q}} \;\text{ and }\; \bar{\rho}_{\boldsymbol{p}} = \text{median}\, \hat{\rho}_{\boldsymbol{q}} \;\text{ with }\; \boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}} \tag{8.10}$$

The filter essentially computes a median on the selective set of neighborhood pixels $\mathcal{N}_{\boldsymbol{p}}$ which have high confidence and similar color as the center pixel $\boldsymbol{p}$. In the next section, we experimentally demonstrate the performance of our proposed approach with respect to baseline SGM and other state-of-the-art methods.

8.4 Experiments

We report a thorough evaluation of SGM-Forest on three stereo benchmarks – Middlebury 2014, KITTI 2015, and ETH3D 2017 [106, 276, 290]. Our evaluation protocol contrasts to most top-ranked stereo methods which often evaluate only on one benchmark [109, 163, 210, 237, 293, 330]. In all our experiments, SGM-Forest outperforms SGM by a significant margin and ranks competitively against the state-of-the-art learning-based and global stereo methods, which are computationally more expensive. It also robustly generalizes across different dataset domains.

8.4.1 Implementation Details

Scanline Optimization and SGM

To facilitate an unbiased comparison, we use the same SGM implementation throughout all experiments. We compare three different matching costs (NCC, MC-CNN-fast [352], MC-CNN-acrt [352]) as the unary term C, which is quantized to 8 bits for reduced memory usage using linear rescaling to the range $[0, 255]$. Image intensities are given in the range $[0, 255]$. For NCC, we use a patch size of 7×7. We follow standard procedure and improve the right image rectification using sparse feature matching before computing the matching cost. The smoothness term $V(d, d')$ uses the constant parameters $P_1 = 100$ and $P_2 = P_1(1 + \alpha e^{-|\Delta I|/\beta})$, where $\alpha = 8$, $\beta = 10$, and ΔI is the intensity difference between neighboring pixels. The choice of P_2 favors large disparity jumps at occlusion boundaries with large intensity changes.

SGM-Forest

In all our experiments, we train random forests with 128 trees, a maximum depth of 25, and the Gini impurity measure to decide on the optimal data split. We set $\epsilon_d = 2$, $\epsilon_{\rho} = 0.1$, $\epsilon_{\boldsymbol{p}} = 5$, and $\epsilon_I = 10$. These optimal parameters were decided using parameter grid search and 3-fold cross validation on the Middlebury 2014 training scenes. To showcase the generalization robustness of our approach, we

Method	Left View Scanlines	Right View Scanlines	Filtering	Training Dataset	bad 0.5px [%]	bad 1px [%]	bad 2px [%]	bad 4px [%]	Time [s]
non-occluded									
SGM	all			–	50.85	23.04	8.89	5.16	**3.0**
SGM – $\min_d L_{\boldsymbol{r}}(\boldsymbol{p}, d)$	all			–	52.18	25.45	11.81	7.79	3.1
SGM – $\min_d \text{median}_{\boldsymbol{r}} L_{\boldsymbol{r}}(\boldsymbol{p}, d)$	all			–	63.25	31.81	9.90	8.24	3.2
SGM-SVM	all			M	48.68	21.88	8.57	5.09	323.7
SGM-MLP	all			M	47.77	21.83	8.53	5.08	21.0
SGM-Forest	horiz+vert			M	47.36	21.30	8.49	4.93	5.7
	top-down			M	47.45	21.20	8.38	4.94	5.8
	bottom-up			M	47.65	21.54	8.54	4.98	5.8
	all			M	46.67	20.85	8.40	4.89	6.1
	all	•		M	46.49	20.81	8.23	4.72	6.3
	all	•	•	E	46.80	20.32	8.17	4.79	8.2
	all	•	•	K	46.48	20.45	8.09	4.81	8.2
	all	•	•	M	**46.08**	**19.99**	**7.78**	**4.41**	8.2
all									
SGM	all			–	65.58	36.08	20.66	16.24	**3.0**
SGM – $\min_d L_{\boldsymbol{r}}(\boldsymbol{p}, d)$	all			–	66.79	38.35	23.32	18.36	3.1
SGM – $\min_d \text{median}_{\boldsymbol{r}} L_{\boldsymbol{r}}(\boldsymbol{p}, d)$	all			–	67.53	39.75	23.34	18.12	3.2
SGM-SVM	all			M	60.89	32.59	20.31	16.16	323.7
SGM-MLP	all			M	60.49	32.61	20.25	16.14	21.0
SGM-Forest	horiz+vert			M	61.09	32.69	18.02	13.19	5.7
	top-down			M	61.31	32.85	18.31	13.37	5.8
	bottom-up			M	61.38	32.91	18.42	13.43	5.8
	all			M	60.28	32.15	17.90	13.14	6.1
	all	•		M	60.18	32.08	17.69	12.91	6.3
	all	•	•	E	59.89	30.69	16.78	11.67	8.2
	all	•	•	K	59.70	30.61	16.72	11.67	8.2
	all	•	•	M	**59.20**	**30.58**	**16.57**	**11.62**	8.2

Table 8.1: Validation performance on *non-occluded* (top) and *all* (bottom) pixels on the Middlebury 2014 training set (15 half resolution pairs). Rows 1–5 show results for SGM baselines. Rows 6–14 report ablation studies for SGM-Forest. Bottom three rows show results for the best SGM-Forest setting, trained on different datasets. Letters M, K, and E refer to Middlebury 2005–06, KITTI, and ETH3D, respectively. The matching cost is always MC-CNN-acrt. Runtimes exclude matching cost and timed on same CPU.

		all				non-occluded			
Dataset	**Method**	0.5px	1px	2px	4px	0.5px	1px	2px	4px
Adirondack	SGM	61.97	32.25	13.02	8.04	58.96	26.92	6.47	1.68
	SGM-Forest (raw)	44.39	20.41	9.48	5.43	40.75	15.65	4.92	1.50
	SGM-Forest (filt.)	41.59	17.78	7.09	3.31	38.23	13.77	3.98	1.06
ArtL	SGM	69.06	29.55	24.66	22.73	60.64	11.49	5.99	4.21
	SGM-Forest (raw)	60.83	29.57	20.62	16.99	52.57	15.43	5.97	3.30
	SGM-Forest (filt.)	60.27	29.39	20.61	15.88	52.97	16.73	7.33	3.55
Jadeplant	SGM	70.71	47.98	33.15	28.02	62.57	33.55	15.09	9.30
	SGM-Forest (raw)	57.24	39.74	31.68	26.57	45.53	23.84	14.62	9.23
	SGM-Forest (filt.)	56.53	38.72	30.39	24.74	45.80	23.94	14.60	8.81
Motorcycle	SGM	60.53	31.23	14.00	11.33	55.97	23.40	4.62	2.37
	SGM-Forest (raw)	44.17	18.21	10.04	7.84	39.23	11.45	3.66	2.22
	SGM-Forest (filt.)	42.91	17.24	8.84	6.39	38.61	11.58	3.74	2.18
MotorcycleE	SGM	61.71	32.16	14.46	11.46	57.35	24.55	5.24	2.49
	SGM-Forest (raw)	44.29	18.53	10.14	7.94	39.46	12.06	3.97	2.34
	SGM-Forest (filt.)	42.95	16.96	8.78	6.38	39.06	11.68	3.94	2.25
Piano	SGM	65.09	38.72	19.91	15.01	62.19	33.68	13.54	8.60
	SGM-Forest (raw)	48.49	26.54	17.31	12.23	44.59	21.42	12.58	8.35
	SGM-Forest (filt.)	47.89	26.40	16.57	11.27	43.94	21.29	11.92	7.64
PianoL	SGM	68.87	44.65	26.00	20.46	66.27	40.06	20.03	14.32
	SGM-Forest (raw)	57.86	35.96	24.04	18.37	54.76	31.51	19.43	14.34
	SGM-Forest (filt.)	57.32	35.04	22.59	16.94	54.15	30.49	17.99	13.10
Pipes	SGM	62.90	35.34	19.33	17.24	55.95	23.45	5.01	3.14
	SGM-Forest (raw)	43.13	22.79	16.22	14.04	33.22	10.26	4.08	2.70
	SGM-Forest (filt.)	41.89	21.60	15.48	13.41	32.26	9.62	4.04	2.71
Playroom	SGM	69.82	46.35	25.60	18.44	65.04	37.88	14.07	6.22
	SGM-Forest (raw)	59.01	37.03	23.58	16.85	52.65	27.57	12.92	6.48
	SGM-Forest (filt.)	59.00	36.89	23.04	16.00	52.73	27.58	12.51	5.88
Playtable	SGM	65.50	36.95	21.32	15.16	61.53	29.72	12.60	6.10
	SGM-Forest (raw)	51.42	26.71	16.49	10.59	47.61	21.13	10.74	5.14
	SGM-Forest (filt.)	51.69	26.37	15.20	9.02	48.12	21.43	10.36	4.78
PlaytableP	SGM	64.64	34.79	20.38	14.54	60.62	27.39	11.63	5.51
	SGM-Forest (raw)	49.97	24.67	15.42	9.91	46.35	19.26	9.92	4.73
	SGM-Forest (filt.)	50.41	24.50	14.46	8.59	47.00	19.61	9.70	4.41
Recycle	SGM	63.64	34.19	15.66	10.63	60.88	29.36	9.75	4.84
	SGM-Forest (raw)	51.81	24.32	12.41	7.83	48.79	19.94	8.42	4.83
	SGM-Forest (filt.)	50.07	22.46	11.03	6.40	47.05	18.37	7.88	4.54
Shelves	SGM	75.40	55.59	39.45	31.38	73.85	52.80	35.77	27.39
	SGM-Forest (raw)	67.95	49.58	36.46	27.76	66.27	47.16	33.92	25.25
	SGM-Forest (filt.)	66.95	48.12	34.44	24.80	65.24	45.73	32.20	23.04
Teddy	SGM	64.62	20.14	13.62	11.94	60.55	11.06	3.91	2.23
	SGM-Forest (raw)	52.76	20.92	12.29	8.52	47.59	12.64	4.12	1.90
	SGM-Forest (filt.)	51.07	20.42	12.45	8.62	45.83	12.15	4.32	2.23
Vintage	SGM	70.22	45.74	27.83	18.62	67.58	40.94	21.65	11.96
	SGM-Forest (raw)	61.13	42.89	30.69	23.23	57.81	38.25	25.57	18.41
	SGM-Forest (filt.)	60.99	42.42	29.96	22.13	57.66	37.76	24.80	17.24

Table 8.2: This table shows the per-dataset performance (percentage of bad pixels for a given error threshold) for the ablation study in Table 8.1. *SGM* here corresponds to row 1, *SGM-Forest (raw)* to row 10, and *SGM-Forest (filt.)* to row 13 in Table 8.1. Note that the numbers in Table 8.1 are averaged using the Middlebury benchmark weighting (half weight for scenes PianoL, Playtable, Playroom, Shelves, and Vintage).

		Middlebury 2014				KITTI 2015				ETH3D 2017			
Datacost	**Method**	0.5px	1px	2px	4px	0.5px	1px	2px	4px	0.5px	1px	2px	4px
						non-occluded							
NCC	SGM	54.15	28.59	15.23	10.14	59.70	32.28	13.09	6.17	30.94	14.78	8.62	5.67
	SGM-F.	**50.06**	**25.29**	**12.55**	**8.08**	**51.61**	**24.74**	**9.22**	**4.17**	**21.14**	**10.28**	**5.59**	**3.67**
MC-CNN-fast	SGM	51.22	23.49	10.58	6.85	57.53	29.82	11.28	4.80	24.70	8.56	4.14	2.57
	SGM-F.	**48.73**	**22.24**	**9.55**	**5.91**	**50.25**	**22.98**	**7.88**	**3.28**	**16.31**	**6.08**	**3.04**	**1.94**
MC-CNN-acrt	SGM	50.85	23.04	8.89	5.16	56.27	26.90	7.41	3.00	37.46	14.44	7.17	4.72
	SGM-F.	**46.08**	**19.99**	**7.78**	**4.41**	**46.16**	**18.82**	**5.76**	**2.56**	**26.26**	**11.05**	**6.56**	**4.71**
						all							
NCC	SGM	69.23	42.36	27.96	22.25	60.59	33.79	15.06	8.34	32.52	16.71	10.66	7.69
	SGM-F.	**64.00**	**37.22**	**22.85**	**17.09**	**52.39**	**25.80**	**10.11**	**4.69**	**22.48**	**11.26**	**6.36**	**4.35**
MC-CNN-fast	SGM	65.82	36.22	21.98	17.47	58.48	31.39	13.30	7.02	26.34	10.50	6.13	4.52
	SGM-F.	**62.04**	**32.96**	**18.22**	**13.16**	**51.03**	**24.05**	**8.73**	**3.78**	**17.62**	**7.17**	**3.66**	**2.51**
MC-CNN-acrt	SGM	65.58	36.08	20.66	16.24	57.24	28.55	9.54	5.26	39.03	16.34	9.14	6.67
	SGM-F.	**59.20**	**30.58**	**16.57**	**11.62**	**46.88**	**19.77**	**6.51**	**2.97**	**27.40**	**11.89**	**7.30**	**5.52**

Table 8.3: This table shows the validation performance using 3-fold cross-validation for different matching costs and datasets at different error thresholds. Our method (SGM-F.) outperforms baseline SGM in all settings.

train and evaluate our SGM-Forest on different dataset combinations. In all settings, the training and test scenes are non-overlapping and we provide a detailed list of training/test splits in the supplementary material. For learning our SGM-Forest model, we sample a maximum of 500K random pixels with ground-truth disparity uniformly in each training image.

8.4.2 Ablation Study

We now evaluate several aspects of our algorithm using an extensive ablation study summarized in Table 8.1, Table 8.3, and Table 8.2.

SGM Baseline

We compare our SGM baseline against two simple methods that robustify Equation 8.4 and Equation 8.5 (see Table 8.1): the first approach is SGM – $\min_d L_{\boldsymbol{r}}(\boldsymbol{p}, d)$ and selects the scanline solution with minimum cost as the disparity estimate, while the second approach SGM – $\min_d \text{median}_{\boldsymbol{r}}\, L_{\boldsymbol{r}}(\boldsymbol{p}, d)$ uses the robust median instead of summation for aggregating the costs from multiple scanlines. Both methods perform worse than baseline SGM, underlining the need for a more sophisticated fusion approach.

Input Proposals

The input to our algorithm is a set of proposal cost volumes $K_n(\boldsymbol{p}, d)$. As demonstrated in Figure 8.1, a single scanline performs worse than SGM while the best of

multiple scanlines is significantly better. In fact, our method is general and the input proposals to our system need not be limited to the canonical 1D scanline optimizations. We always consider the regular SGM cost volume $S(\boldsymbol{p}, d)$ as a proposal. Using only this proposal leads to a trivial 1-class classification problem and is equivalent to running baseline SGM (see Table 8.1). Adding the four horizontal and vertical scanlines from the left image as proposals improves the accuracy significantly, which is further boosted by adding the remaining 4 diagonal scanlines. Using only scanlines that propagate in the five top-down or five bottom-up directions degrades performance slightly but is still much better than regular SGM and enables real-time implementation of our algorithm on an FPGA [141]. We also experimented with running two horizontal scanline optimizations on the right image and warping the results to the left view to be used as two additional proposals. This is because the occluded pixels in the left image are invisible in the right image and the left occlusion edges are usually more accurately recovered in the right disparity map. These additional proposals provide a small but consistent improvement.

Classification Model

In Section 8.3.3, we argued that, for our task, random forests provide the best trade-off in terms of accuracy and efficiency. We experimented with many different classification models, including k-NN search, SVMs, (gradient boosted) decision trees, AdaBoost, neural nets, etc. In Table 8.1, we show results for two other well-performing models: SGM-SVM uses a linear SVM classifier and SGM-MLP is a multi-layer perceptron using 3 hidden layers with ReLU activation and twice the neurons after each layer followed by a final softmax layer for classification. SGM-MLP outperforms the SGM baseline but has slightly lower accuracy and efficiency on the CPU than SGM-Forest.

Filtering

The final step in our algorithm is the confidence-based spatial filtering of the disparity and confidence maps. While the biggest accuracy improvement stems from the initial fusion step (see Table 8.1), the final filtering further improves the results by eliminating spatially inconsistent outliers.

Efficiency

The reported runtimes in Table 8.1 show only a small computational overhead of SGM-Forest and our proposed filtering over baseline SGM, enabling a potential real-time implementation on the GPU or FPGA (see Section 8.4.5). Note that the runtimes exclude the matching cost computation, i.e., the overhead of SGM-Forest becomes negligible if, for example, MC-CNN-acrt is used.

Middlebury 2014 (MC-CNN-acrt)					
Method	**non-occl.**		**all**		**Time**
LocalExp	5.43%	#1	11.7%	#1	881s
3DMST	5.92%	#2	12.5%	#3	174s
MC-CNN+TDSR	6.35%	#2	12.1%	#3	657s
PMSC	6.71%	#4	13.6%	#4	599s
LW-CNN	7.04%	#5	17.8%	#15	314s
MeshStereoExt	7.08%	#6	15.7%	#9	161s
FEN-D2DRR	7.23%	#7	16.0%	#11	121s
APAP-Stereo	7.26%	#8	13.7%	#5	131s
SGM-Forest	**7.37%**	**#9**	**15.5%**	**#8**	**88s**
NTDE	7.44%	#10	15.3%	#7	152s

Middlebury 2014 (MC-CNN-fast)					
Method	**non-occl.**		**all**		**Time**
LocalExp	6.52 %	#1	12.1%	#1	846s
3DMST	7.08 %	#2	12.9%	#2	167s
APAP-Stereo	7.53%	#3	14.3%	#6	117s
FEN-D2DRR	7.89%	#4	14.1%	#4	73s
...					
MC-CNN-acrt	10.1%	#12	19.7%	#20	106s
...					
SGM-Forest	**11.1%**	**#19**	**17.8%**	**#14**	**9s**
...					
MC-CNN-fast	11.7%	#21	21.5%	#27	1s

Table 8.4: **Middlebury Benchmark.** Left: Official results for the top 10 performing methods using MC-CNN-acrt for our SGM-Forest. Our method achieves the best runtime among the top performing methods. Right: Inofficial results on the training scenes trained on Middlebury 2005–06 using MC-CNN-fast. SGM-Forest with MC-CNN-fast outperforms baseline SGM with MC-CNN-acrt but is an order of magnitude faster.

Generalization and Robustness

All results in Table 8.1 were obtained by training on Middlebury 2005–06 and evaluating on Middlebury 2014, which already demonstrates good generalization properties. Note that Middlebury 2014 images are much more challenging than those in Middlebury 2005–06. Moreover, we also evaluate cross-domain generalization by training on KITTI (outdoors) and ETH3D (outdoors and indoors) and evaluating on Middlebury 2014 (indoors). In both cases, our approach achieves almost the same performance as compared to training on Middlebury. Table 8.3 shows that SGM-Forest improves over baseline SGM in every single metric irrespective of matching cost and dataset. In contrast to most learning-based methods, we demonstrate that our learned fusion approach is general and extremely robust across different domains and settings: SGM-Forest performs well outdoors when trained on indoor scenes, handles different image resolutions, disparity ranges and diverse matching costs, and consistently outperforms baseline SGM by a large margin.

8.4.3 Benchmark Results

Unlike most existing methods, we evaluate SGM-Forest on three benchmarks and achieve competitive performance with respect to the state of the art. For all benchmark submissions, we use the best setting found in our ablation study, i.e., we include 8 (and 2) proposals from the left (and right) view and run disparity refinement.

Middlebury

Table 8.4 reports our results on Middlebury 2014. For the benchmark submission, we use MC-CNN-acrt matching costs and jointly train on Middlebury 2005–06 and

KITTI 2015		
Method	**Error**	**Time**
CNNF+SGM	3.60% (#9)	71.0s
SGM-Net	3.66% (#11)	67.0s
MC-CNN-acrt	3.89% (#12)	67.0s
SGM-Forest	**4.38% (#14)**	**6.0s**
MC-CNN-WS	4.97% (#18)	1.4s
SGM_ROB [135]	6.38% (#27)	0.1s
SGM+C+NL	6.84% (#31)	270.0s
SGM+LDOF	6.84% (#32)	86.0s
SGM+SF	6.84% (#33)	2700.0s
CSCT+SGM+MF	8.24% (#35)	6.4ms

ETH3D 2017			
Method	**non-occl.**	**all**	**Time**
SGM-Forest	**5.40%**	**4.96%**	**5.21s**
SGM_ROB [135]	10.08%	10.77%	0.15s
MeshStereo	11.94%	11.52%	159.24s
SPS-Stereo	15.83%	15.04%	1.59s
ELAS	17.99%	16.72%	0.13s

Table 8.5: **KITTI and ETH3D Benchmarks.** Left: KITTI results over all pixels for all ranked SGM variants. Our SGM-Forest uses MC-CNN-fast as matching cost and achieves high accuracy at comparatively low runtime. Right: ETH3D results over non-occluded and all pixels for all ranked methods. Our SGM-Forest uses MC-CNN-fast as matching cost and achieves the best accuracy at comparatively low runtime.

the training scenes of Middlebury 2014. Our method ranks competitively among the top ten methods in terms of accuracy but is significantly faster. In addition to our official submission, we also report unofficial results for MC-CNN-fast evaluated on the training scenes.[1] The models for this submission were trained only on the Middlebury 2005–06 scenes. Using MC-CNN-fast, SGM-Forest outperforms SGM by two percentage points on non-occluded pixels. Evaluated on all pixels, SGM-Forest with MC-CNN-fast outperforms baseline SGM with MC-CNN-acrt by two percentage points but SGM-Forest is an order of magnitude faster.

KITTI

Table 8.5 lists all SGM-based methods evaluated on KITTI. We use MC-CNN-fast for this submission and are ranked right behind the original MC-CNN-acrt method [352], CNNF+SGM [391], and SGM-Net [293]. However, our method is an order of magnitude faster even though our scanline optimization and the proposed additional steps are implemented on the CPU while MC-CNN-WS runs on the GPU. Note that CNNF+SGM and SGM-Net report results only on KITTI whereas our method generalizes across domains and datasets.

ETH3D

On this fairly new benchmark with diverse indoor and outdoor images, SGM-Forest is currently ranked 1st with competitive running times (see Table 8.5). Our submission uses MC-CNN-fast which was surprisingly more accurate than MC-CNN-acrt

[1] Only one submission per method is allowed on Middlebury 2014.

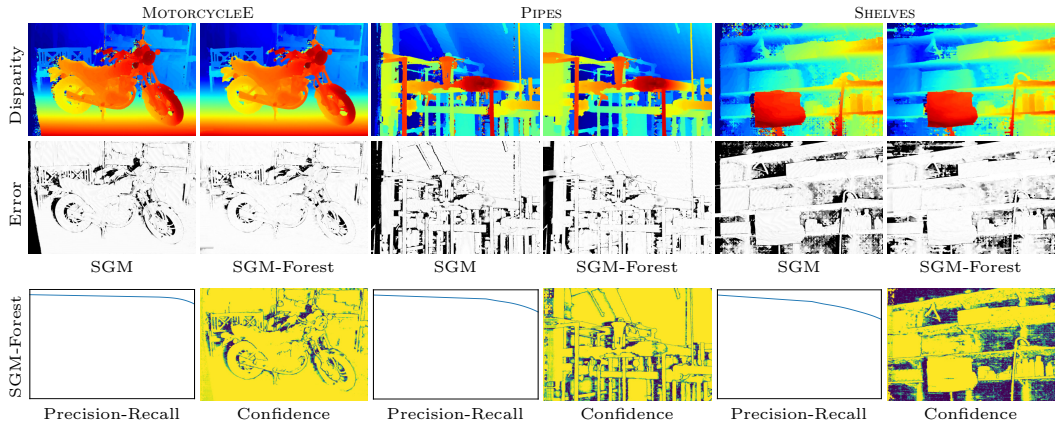

Figure 8.3: Qualitative Middlebury results for SGM and SGM-Forest. Absolute error maps are clipped to $[0\text{px}, 8\text{px}]$. Precision (Y) and Recall (X) in the interval $[0, 1]$. Confidence maps are log-scaled.

on ETH3D (also see Table 8.3). Here, our SGM-Forest submission has almost half the error as the original SGM method [135].

8.4.4 Qualitative Results

Figure 8.3 shows qualitative results for Middlebury. Compared to baseline SGM, our SGM-Forest produces less streaking artifacts and performs significantly better in occluded areas. High confidence regions in general correspond to low errors. This is further confirmed by the monotonically decreasing precision-recall curves, which were produced by thresholding on the predicted confidences. Figure 8.4 and Figure 8.5 show further qualitative results and comparisons between raw predictions and filtered results.

8.4.5 Limitations and Future Work

Our current SGM and random forest implementation is CPU-based and is not real-time capable since we buffer all scanline cost volumes before fusion. The learned forests in this chapter use 128 trees, so our method could be sped up easily by using fewer trees. In our experiments, even a single decision tree improved upon baseline SGM. An implementation of our method on the GPU would be straightforward, where SGM-MLP would probably outcompete SGM-Forest in efficiency at the cost of a small degradation in accuracy. Real-time implementation on embedded systems [141] requires a one-pass, buffer-less algorithm prohibiting the use of all 8 scanline directions. In Table 8.1, we demonstrated that our idea also works well for top-down/bottom-up directions only.

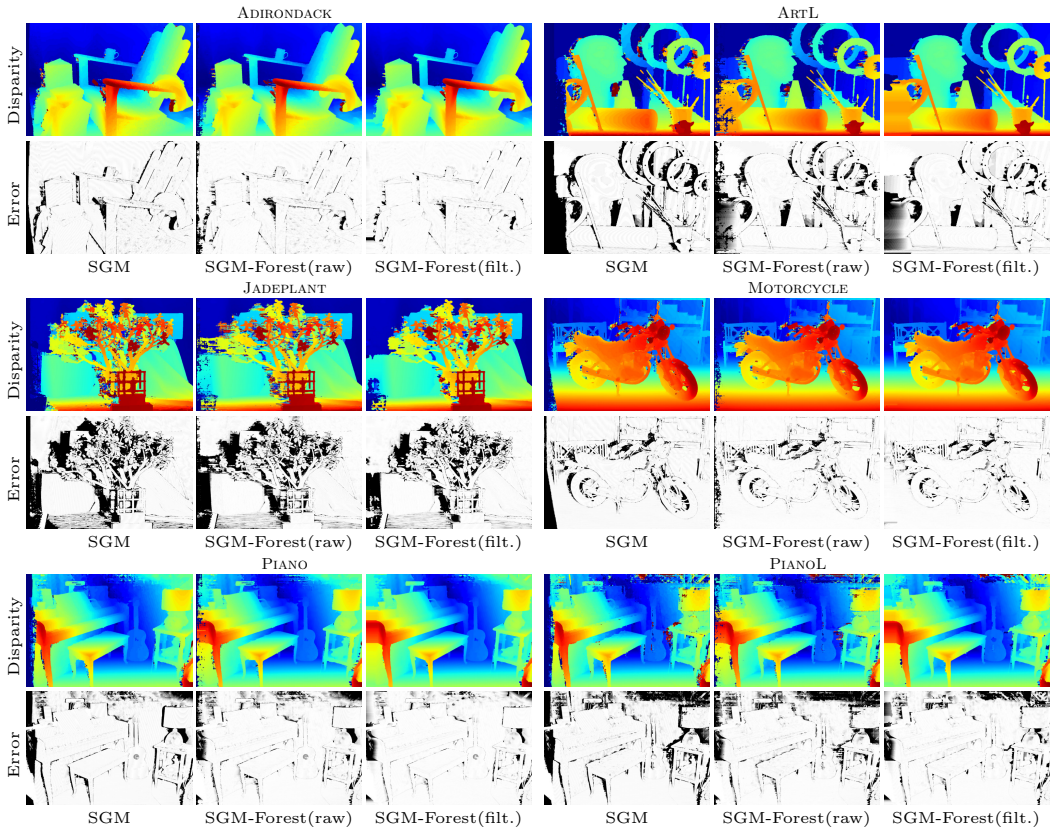

Figure 8.4: Qualitative Middlebury results for SGM and SGM-Forest. Absolute error maps clipped to $[0\text{px}, 8\text{px}]$. Refer to Figure 8.3 and Figure 8.5 for more results.

8.5 Summary

In this chapter, we proposed a learning-based approach to fuse scanline optimization proposals dense stereo reconstruction using SGM, replacing the brittle and heuristic scanline aggregation steps in standard SGM. Our method is efficient and accurate and ranks 1st on the ETH3D benchmark while being competitive on Middlebury and KITTI. We have demonstrated consistent improvements over SGM on three stereo benchmarks. The learning appears to be extremely robust and generalizes well across datasets. Our method can be readily integrated into existing SGM variants and allows for real-time implementation in practical, high-quality stereo systems.

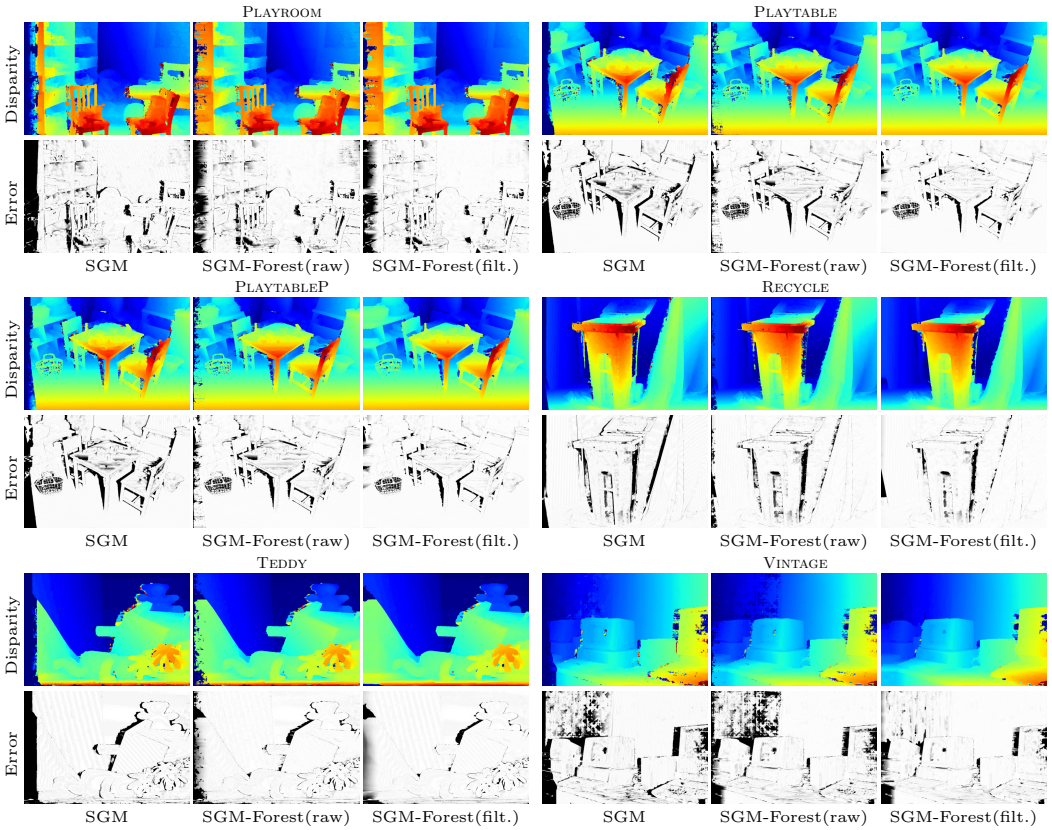

Figure 8.5: Qualitative Middlebury results for SGM and SGM-Forest. Absolute error maps clipped to $[0\text{px}, 8\text{px}]$. Refer to Figure 8.3 and Figure 8.4 for more results.

9 Pixelwise View Selection for Unstructured Multi-View Stereo

While the previous chapter focused on stereo reconstruction using a comparatively structured setup of rectified stereo images, this chapter proposes a method for the accurate and robust dense reconstruction of unstructured image collections. Large-scale 3D reconstruction from unstructured images and, in particular, Internet photos has seen a tremendous evolution in the sparse modeling [3, 89, 129, 275, 284, 288, 310, 396] and in the dense modeling stages [22, 93, 95, 97, 294, 295, 394]. Many applications benefit from a dense scene representation, e.g., classification [299], image-based rendering [57], localization [87], etc. Despite the widespread use of dense reconstruction, the efficient and robust estimation of accurate, complete, and aesthetically pleasing dense models in uncontrolled environments remains a challenging task. Dense pixelwise correspondence search is the core problem of stereo methods. Recovering correct correspondence is challenging even in controlled environments with known viewing geometry and illumination. In uncontrolled settings, e.g., where the input consists of crowd-sourced images, it is crucial to account for various factors, such as heterogeneous resolution and illumination, scene variability, unstructured viewing geometry, and mis-registered views.

Our proposed approach improves the state of the art in dense reconstruction for unstructured images. This work leverages the optimization framework by Zheng et al. [394] to propose the following core contributions:

- Pixelwise *normal estimation* embedded into an improved PatchMatch sampling scheme for improved accuracy, especially on slanted surfaces.
- Pixelwise view selection using triangulation angle, incident angle, and image resolution-based *geometric priors* for improved robustness.
- Integration of a "temporal" *view selection smoothness* term.

Figure 9.1: Dense point cloud reconstructions from unstructured Internet images for Louvre, France; Todai-ji, Japan; Paris Opera, France; and Astronomical Clock in Prague, Czech Republic.

- Adaptive window support through bilateral *photometric consistency* for improved occlusion boundary behavior.
- Introduction of a multi-view *geometric consistency* term for simultaneous depth and normal estimation and image-based fusion.
- Reliable depth and normal *filtering* and *fusion*. Outlier-free and accurate depth and normal estimates further allow for direct meshing of the resulting point cloud.

We achieve state-of-the-art results on benchmarks (Middlebury [292], Strecha [322]). To demonstrate the advantages of our method in a more challenging setting, we process SFM models of a world-scale Internet dataset [129]. The entire algorithm is released to the public as an open-source implementation.

9.1 Related Work

Stereo methods have advanced in terms of accuracy, completeness, scalability, and benchmarking – from the minimal stereo setup with two views [145, 157, 260, 278] to multi-view methods [49, 92, 93, 95, 96, 97, 147, 394]. Furthermore, the joint estimation of semantics [116], dynamic scene reconstruction [154, 205, 236, 252, 344], and benchmarking [278, 292, 294, 322]. Our method performs MVS with pixelwise view selection for depth/normal estimation and fusion. Here, we only review the most related approaches, within the large body of research in multi-view and two-view stereo.

MVS leverages multiple views to overcome the inherent occlusion problems of two-view approaches [323, 372, 397]. Accordingly, view selection plays a crucial role in the effectiveness of MVS. Kang et al. [159] heuristically select the best views with minimal cost (usually 50%) for computing the depth of each pixel. Strecha et al. [320, 321] probabilistically model scene visibility combined with a local depth smoothness assumption [321] in a Markov Random Field for pixelwise view selection. Different from our approach, their method is prohibitive in memory usage and does neither include normal estimation nor photometric and geometric priors for view selection. Gallup et al. [98] select different views and resolutions on a per-pixel basis to achieve a constant depth error. In contrast, our method simultaneously considers a variety of photometric and geometric priors improving upon the robustness and accuracy of the recently proposed depth estimation framework by Zheng et al. [394]. Their method is most closely related to our approach and is reviewed in more detail in Section 9.2.

MVS methods commonly use a fronto-parallel scene structure assumption. Gallup et al. [99] observed the distortion of the cost function caused by structure that deviates from this prior and combats it by using multiple sweeping directions deduced from the sparse reconstruction. Earlier approaches [31, 48, 376] similarly account for the surface normal in stereo matching. Recently, Bleyer et al. [34] use PatchMatch to estimate per-pixel normals to compensate for the distortion of the cost function.

In contrast to these approaches, we propose to estimate normals not in isolation but also considering the photometric and geometric constraints guiding the matchabilty of surface texture and its accuracy. By probabilistically modeling the contribution of individual viewing rays towards reliable surface recovery, we achieve significantly improved depth and normal estimates.

Depth map fusion integrates multiple depth maps into a unified and augmented scene representation while mitigating any inconsistencies among individual estimates. Jancoseck et al. [147] fuses multiple depth estimates into a surface and, by evaluating visibility in 3D space, they also attempt to reconstruct parts that are not directly supported by depth measurements. In contrast, our method aims at directly maximizing the estimated surface support in the depth maps and achieves higher completeness and accuracy (see Section 9.4). Gösele et al. [110] propose a method that explicitly targets at the reconstruction from crowd-sourced images. They first select camera clusters for each surface and adjust their resolution to the smallest common resolution. For depth estimation, they then use the four most suitable images for each pixel. As already noted in Zheng et al. [394], this early pre-selection of reduced camera clusters may lead to less complete results and is sensitive to noise. Our method avoids this restrictive selection scheme by allowing dataset-wide, pixelwise sampling for view selection. Zach et al. [377] proposed a variational depth map formulation that enabled parallelized computation on the GPU. However, their volumetric approach imposes substantial memory requirements and is prohibitive for the large-scale scenes targeted by our method. Beyond these methods, there are several large-scale dense reconstruction and fusion methods for crowd-sourced images, e.g., Furukawa et al. [93] and Gallup et al. [100, 395], who all perform heuristic pre-selection of views, which leads to reduced completeness and accuracy as compared to our method.

9.2 Pixelwise View Selection

This section reviews the framework by Zheng et al. [394] to introduce notation and context for our contributions. Since their method processes each row/column independently, we limit the description to a single image row with l as the column index. Their method estimates the depth θ_l for a pixel in the reference image X^{ref} from a set of unstructured source images

$$\boldsymbol{X}^{\text{src}} = \{X^m \mid m = 1 \dots M\} \quad . \tag{9.1}$$

The estimate θ_l maximizes the color similarity between a patch X_l^{ref} in the reference image and homography-warped patches X_l^m in non-occluded source images. The binary indicator variable $Z_l^m \in \{0, 1\}$ defines the set of non-occluded source images as

$$\bar{\boldsymbol{X}}_l^{\text{src}} = \{X^m \mid Z_l^m = 1\} \quad . \tag{9.2}$$

To sample $\bar{\boldsymbol{X}}_l^{\text{src}}$, they infer the probability that the reference patch X_l^{ref} at depth θ_l is visible at the source patch X_l^m using

$$P(X_l^m|Z_l^m,\theta_l) = \begin{cases} \frac{1}{NA}\exp\left(-\frac{(1-\rho_l^m(\theta_l))^2}{2\sigma_\rho^2}\right) & \text{if } Z_l^m = 1 \\ \frac{1}{N}\mathcal{U} & \text{if } Z_l^m = 0 \end{cases} \tag{9.3}$$

$$\text{with} \quad A = \int_{-1}^{1} exp\{-\frac{(1-\rho)^2}{2\sigma_\rho^2}\}d\rho \ , \tag{9.4}$$

where N is a constant canceling out in the inference. In the case of occlusion, the color distributions of the two patches are unrelated and follow the uniform distribution $\mathcal{U}$ in the range $[-1,1]$ with probability density 0.5. Otherwise, ρ_l^m describes the color similarity between the reference and source patch based on normalized cross-correlation (NCC) using fronto-parallel homography warping. The variable σ_ρ determines a soft threshold for ρ_l^m on the reference patch being visible in the source image. The state-transition matrix from the preceding pixel $l-1$ to the current pixel l is

$$P(Z_l^m|Z_{l-1}^m) = \begin{bmatrix} \gamma & 1-\gamma \\ 1-\gamma & \gamma \end{bmatrix} \tag{9.5}$$

and encourages spatially smooth occlusion indicators, where a larger γ enforces neighboring pixels to have more similar indicators. Given reference and source images $\boldsymbol{X} = \{X^{\text{ref}}, \boldsymbol{X}^{\text{src}}\}$, the inference problem then boils down to recover, for all L pixels in the reference image, the depths

$$\boldsymbol{\theta} = \{\theta_l \mid l = 1 \dots L\} \tag{9.6}$$

and the occlusion indicators

$$\boldsymbol{Z} = \{Z_l^m \mid l = 1 \dots L,\, m = 1 \dots M\} \tag{9.7}$$

from the posterior distribution $P(\boldsymbol{Z},\boldsymbol{\theta}|\boldsymbol{X})$ with a uniform prior $P(\boldsymbol{\theta})$. To solve the computationally infeasible Bayesian approach of first computing the joint probability

$$P(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{\theta}) = \prod_{l=1}^{L}\prod_{m=1}^{M}[P(Z_l^m|Z_{l-1}^m)P(X_l^m|Z_l^m,\theta_l)] \tag{9.8}$$

and then normalizing over $P(\boldsymbol{X})$, Zheng et al. use variational inference theory to develop a framework that is a variant of the generalized expectation-maximization (GEM) algorithm [228]. For the inference of $\boldsymbol{Z}$ in the hidden Markov-Chain, the forward-backward algorithm is used in the E step of GEM. PatchMatch-inspired [34] sampling serves as an efficient scheme for the inference of $\boldsymbol{\theta}$ in the M step of GEM. Their method iteratively solves for $\boldsymbol{Z}$ with fixed $\boldsymbol{\theta}$ and *vice versa* using interleaved row-/columnwise propagation. Full depth inference

$$\theta_l^{\text{opt}} = \arg\min_{\theta_l^*} \sum\nolimits_{m=1}^{M} P_l(m)(1-\rho_l^m(\theta_l^*)) \tag{9.9}$$

has high computational cost if M is large as PatchMatch requires the NCC to be computed many times. The probability

$$P_l(m) = \frac{q(Z_l^m = 1)}{\sum_{m=1}^{M} q(Z_l^m = 1)} \tag{9.10}$$

indicates the patch similarity between the source image m to the reference patch, while $q(\boldsymbol{Z})$ is an approximation of the real posterior $P(\boldsymbol{Z})$. Source images with small $P_l(m)$ are non-informative for the depth inference, hence Zheng et al. propose a Monte Carlo based approximation of θ_l^{opt} for view selection

$$\hat{\theta}_l^{\text{opt}} = \arg\min_{\theta_l^*} \frac{1}{|S|} \sum\nolimits_{m \in S} (1 - \rho_l^m(\theta_l^*)) \tag{9.11}$$

by sampling a subset of images $S \subset \{1 \dots M\}$ from the distribution $P_l(m)$ and hence only computing the NCC for the most similar source images.

9.3 Algorithm

In this section, we describe our novel algorithm that leverages the optimization framework reviewed in the previous section. We first present the individual terms of the proposed likelihood function, while Section 9.3.6 explains their integration into the overall optimization framework.

9.3.1 Normal Estimation

Zheng et al. [394] map between the reference and source images using fronto-parallel homographies leading to artifacts for oblique structures [99]. In contrast, we estimate per-pixel depth θ_l and normals $\boldsymbol{n}_l \in \mathbb{R}^3, \|\boldsymbol{n}_l\| = 1$. A patch at $\boldsymbol{x}_l \in \mathbb{P}^2$ in the reference image warps to a source patch at $\boldsymbol{x}_l^m \in \mathbb{P}^2$ as $\boldsymbol{x}_l^m = \boldsymbol{H}_l \boldsymbol{x}_l$ with the homography between the two views defined as

$$\boldsymbol{H}_l = \boldsymbol{K}^m (\boldsymbol{R}^m - \frac{\boldsymbol{T}^m \boldsymbol{n}_l^T}{d_l}) \boldsymbol{K}^{-1} \ . \tag{9.12}$$

Here, $\boldsymbol{R}^m \in SO(3)$ and $\boldsymbol{T}^m \in \mathbb{R}^3$ define the relative transformation from the reference to the source camera frame. $\boldsymbol{K}$ and $\boldsymbol{K}^m$ denote the calibration of the reference and source images, respectively, and $d_l = \boldsymbol{n}_l^T \boldsymbol{p}_l$ is the orthogonal distance from the reference image to the plane at the point $\boldsymbol{p}_l = \theta_l \boldsymbol{K}^{-1} \boldsymbol{x}_l$.

Given no knowledge of the scene, we assume a uniform prior $P(\boldsymbol{N})$ in the inference of the normals

$$\boldsymbol{N} = \{\boldsymbol{n}_l \mid l = 1 \dots L\} \ . \tag{9.13}$$

Estimating the normals $\boldsymbol{N}$ requires to change the terms $P(X_l^m | Z_l^m, \theta_l)$ and $P_l(m)$ from Equation 9.3 and Equation 9.11 to also depend on $\boldsymbol{N}$, as the color similarity

ρ_l^m is now based on slanted rather than fronto-parallel homographies. Consequently, the optimal depth and normal are chosen as

$$(\hat{\theta}_l^{\text{opt}}, \hat{\boldsymbol{n}}_l^{\text{opt}}) = \underset{\theta_l^*, \boldsymbol{n}_l^*}{\arg\min} \frac{1}{|S|} \sum\nolimits_{m \in S} (1 - \rho_l^m(\theta_l^*, \boldsymbol{n}_l^*)) \ . \tag{9.14}$$

To sample unbiased random normals in PatchMatch, we follow the approach by Galliani et al. [97]. With the additional two unknown normal parameters, the number of unknowns per pixel in the M step of GEM increases from one to three. While this in theory requires PatchMatch to generate many more samples, we propose an efficient propagation scheme that maintains the convergence rate of depth-only inference. Since depth θ_l and normal $\boldsymbol{n}_l$ define a local planar surface in 3D, we propagate the depth $\theta_{l-1}^{\text{prp}}$ of the intersection of the ray of the current pixel $\boldsymbol{x}_l$ with the local surface of the previous pixel $(\theta_{l-1}, \boldsymbol{n}_{l-1})$. This exploits first-order smoothness of the surface (cf. [130]) and thereby drastically speeds up the optimization since correct depths propagate more quickly along the surface. Moreover, different from the typical iterative refinement of normals using bisection as an intermediate step between full sweeps of propagations (cf. [34, 97]), we generate a small set of additional plane hypotheses at each propagation step. We observe that the current best depth and normal parameters can have the following states: neither of them, one of them, or both of them have the optimal solution or are close to it. By combining random and perturbed depths with current best normals and vice versa, we increase the chance of sampling the correct solution. More formally, at each step in PatchMatch, we choose the current best estimate for pixel l according to Equation 9.11 from the set of hypotheses

$$\{(\theta_l, \boldsymbol{n}_l), (\theta_{l-1}^{\text{prp}}, \boldsymbol{n}_{l-1}), (\theta_l^{\text{rnd}}, \boldsymbol{n}_l), (\theta_l, \boldsymbol{n}_l^{\text{rnd}}), (\theta_l^{\text{rnd}}, \boldsymbol{n}_l^{\text{rnd}}), (\theta_l^{\text{prt}}, \boldsymbol{n}_l), (\theta_l, \boldsymbol{n}_l^{\text{prt}})\}, \tag{9.15}$$

where θ_l^{rnd} and $\boldsymbol{n}_l^{\text{rnd}}$ denote randomly generated samples. To refine the current parameters when they are close to the optimal solution, we perturb the current estimate as $\theta_l^{\text{prt}} = (1 \pm \epsilon)\theta_l$ and $\boldsymbol{n}_l^{\text{prt}} = \boldsymbol{R}_\epsilon \boldsymbol{n}_l$. The variable ϵ describes a small depth random perturbation, and the rotation matrix $\boldsymbol{R}_\epsilon \in SO(3)$ perturbs the normal direction by a small angle subject to $\boldsymbol{p}_l^T \boldsymbol{n}_l^{\text{prt}} < 0$. Normal estimation improves both the reconstruction completeness and accuracy, while the new sampling scheme leads to both fast convergence and more accurate estimates (see Section 9.4).

9.3.2 Geometric Priors for View Selection

This section describes how to incorporate geometric priors in the pixelwise view selection for improved robustness in particular for unstructured imagery. On a high level, the proposed priors encourage the sampling of source images with sufficient baseline (*Triangulation Prior*), similar resolution (*Resolution Prior*), and non-oblique viewing direction (*Incident Prior*). In contrast to prior work (e.g. [93, 100, 110]), which decouples inference and per-image geometric priors by pre-selecting source images, we integrate geometric priors on a per-pixel basis into the inference. The

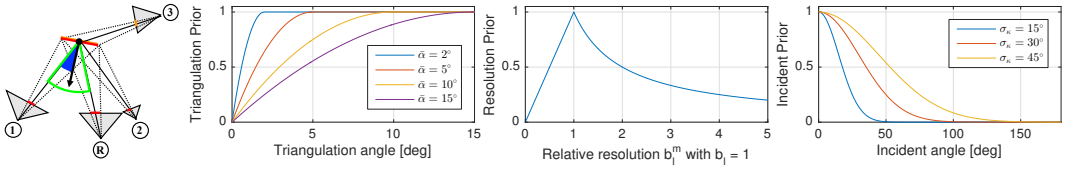

Figure 9.2: Left: Illustration of geometric priors for reference view (R) and three source views (1-3). View 1 has similar resolution (red), and good triangulation (green) and incident angle (blue), while view 2 is oblique and has lower resolution. View 3 cannot see the patch. Right: Geometric prior likelihood functions with different parameters.

motivation for per-pixel geometric priors is similar to inferring per-pixel occlusion indicators $\boldsymbol{Z}$. Since the pre-selection of source images is based on a sparse and therefore incomplete scene representation, the selected source views are often sub-optimal. Occlusion boundaries, triangulation angles, relative image resolution, and incident angle can vary significantly between a single pair of reference and source images (see Figure 9.2). Incorporating geometric priors in addition to the photometric occlusion indicators $\boldsymbol{Z}$ leads to a more comprehensive and robust pixelwise view selection. In the following, we detail the proposed priors and explain their integration into the optimization framework.

Triangulation Prior

Zheng et al. [394] sample source images purely based on color similarity. Consequently, the more similar the reference patch is to the source patch, the higher the selection probability in the view sampling. Naturally, image pairs with small viewpoint change, which coincides with small baseline, have high color similarity. However, image pairs with zero baseline do not carry information for depth inference, because reconstructed points can arbitrarily move along the viewing ray without changing the color similarity. Pure photometric view selection favors to sample these uninformative views. To eliminate this degenerate case, we calculate the triangulation angle

$$\alpha_l^m = \cos^{-1} \frac{(\boldsymbol{p}_l - \boldsymbol{c}^m)^T \boldsymbol{p}_l}{\|\boldsymbol{p}_l - \boldsymbol{c}^m\| \|\boldsymbol{p}_l\|} \quad \text{with} \quad \boldsymbol{c}^m = -(\boldsymbol{R}^m)^T \boldsymbol{T}^m \tag{9.16}$$

and $\alpha_l^m \in [0, \pi)$ between two intersecting viewing rays as a measure of the stability of the reconstructed point $\boldsymbol{p}_l$. Empirically, we choose the following likelihood function

$$P(\alpha_l^m) = 1 - \frac{(\min(\bar{\alpha}, \alpha_l^m) - \bar{\alpha})^2}{\bar{\alpha}^2} \tag{9.17}$$

to describe how informative a source image is for reconstructing the correct point. Intuitively, this function assigns low likelihood to source images for which the triangulation angle is below an a priori threshold $\bar{\alpha}$. Otherwise, no additional view selection preference is imposed (see Figure 9.2).

Resolution Prior

Unstructured datasets usually contain images captured by a multitude of camera types under diverse viewing geometry. As a consequence, images capture scene objects in a wide range of resolutions. To avoid under- and oversampling in computing ρ_l^m, the patches in the reference and source image should have similar size and shape [110]. Similar size is favorable as it avoids comparing images captured at vastly different resolutions, e.g., due to different zoom factors or distance to the object. Similar shape avoids significantly distorted source patches caused by different viewing directions. In the case of different shape, areas within the same source patch have different sampling rates. An approximate measure of the relative size and shape between the reference and source patch is

$$\beta_l^m = \frac{b_l}{b_l^m} \in \mathbb{R}^+ \ , \tag{9.18}$$

where b_l and b_l^m denote the areas covered by the corresponding patches. In our implementation, the reference patch is always square. If the size and shape of the patches is similar, β_l^m is close to the value 1. To quantify the similarity in resolution between two images, we propose the likelihood function

$$P(\beta_l^m) = \min(\beta_l^m, (\beta_l^m)^{-1}) \tag{9.19}$$

and integrate it into $P_l(m)$. Note that, at increased computational cost, undersampling could alternatively be handled by resampling of the source image patch.

Incident Prior

The inferred per-pixel normals provide geometric constraints on the solution space that we encode in the form of a prior. The estimated plane restricts the possible space of source camera locations and orientations. By construction, the camera location can only lie in the positive half-space defined by the plane $(\theta_l, \boldsymbol{n}_l^m)$, while the camera viewing direction must face towards the opposite normal direction. Otherwise, it is geometrically impossible for the camera to observe the surface. To satisfy this geometric visibility constraint, the incident angle of the source camera

$$\kappa_l^m = \cos^{-1} \frac{(\boldsymbol{p}_l - \boldsymbol{c}^m)^T \boldsymbol{n}_l^m}{\|\boldsymbol{p}_l - \boldsymbol{c}^m\| \|\boldsymbol{n}_l^m\|} \quad \text{with} \quad \kappa_l^m \in [0, \pi) \tag{9.20}$$

must be in the interval $0 \leq \kappa_l^m < \frac{\pi}{2}$. In our method, the likelihood function

$$P(\kappa_l^m) = \exp(-\frac{\kappa_l^{m\,2}}{2\sigma_\kappa^2}) \tag{9.21}$$

encodes the belief in whether this geometric constraint is satisfied. This associates some belief with a view even in the case where $\kappa_l^m \geq \frac{\pi}{2}$. The reason for this is, that in the initial inference stage, the variables θ_l and $\boldsymbol{n}_l^m$ are unknown and hence the geometric constraints are likely not yet correct.

Integration

Figure 9.2 visualizes the geometric priors, and Figure 9.4 shows examples of specific priors over all reference image pixels. We integrate the priors into the inference as additional terms in the Monte-Carlo view sampling distribution

$$P_l(m) = \frac{q(Z_l^m = 1)q(\alpha_l^m)q(\beta_l^m)q(\kappa_l^m)}{\sum_{m=1}^{M} q(Z_l^m = 1)q(\alpha_l^m)q(\beta_l^m)q(\kappa_l^m)}, \tag{9.22}$$

where $q(\alpha_l^m), q(\beta_l^m), q(\kappa_l^m)$ are approximations during the variational inference, in the sense that they minimize the KL-divergence to the real posterior [32]. The distributions need no normalization in the inference because we solely use them as modulators for the sampling distribution $P_l(m)$. This formulation assumes statistical independence of the individual priors as a simplifying approximation, which makes the optimization feasible using relatively simple models for well-understood geometric relations. Intuitively, non-occluded images with sufficient baseline, similar resolution, and non-oblique viewing direction are favored in the view selection. Section 9.4 evaluates the priors in detail and shows how they improve the reconstruction robustness especially for unstructured datasets.

9.3.3 View Selection Smoothness

The graphical model associated with the likelihood function in Equation 9.8 uses state-transition probabilities to model spatial view selection smoothness for neighboring pixels in the propagation direction. Due to the interleaved inference using alternating propagation directions, Z_l^m suffers from oscillation, leading to striping effects as shown in Figure 9.5. To reduce the oscillation effect of $Z_{l,t}^m$ in iteration t, we insert an additional "temporal" smoothness factor into the graphical model. In this new model, the state of $Z_{l,t}^m$ depends not only on the state of its neighboring pixel $l-1$ but also on its own state in the previous iteration $t-1$. The temporal state-transition is defined as

$$P(Z_{l,t}^m | Z_{l,t-1}^m) = \begin{bmatrix} \lambda_t & 1-\lambda_t \\ 1-\lambda_t & \lambda_t \end{bmatrix} , \tag{9.23}$$

where a larger λ_t enforces greater temporal smoothness during the optimization. In fact, as the optimization progresses from $t = 1 \dots T$, the value of the estimated $Z_{l,t-1}^m$ should stabilize around the optimal solution. Therefore, we adaptively increase the state-transition probability as $\lambda_t = \frac{t}{2T} + 0.5$, i.e., the inferred $Z_{l,t}^m$ in iterations $t = 1$ and $t = T-1$ have maximal and minimal influence on the final value $Z_{l,T}^m$, respectively. The two state-transitions are jointly modeled as

$$P(Z_{l,t}^m | Z_{l-1,t}^m, Z_{l,t-1}^m) = P(Z_{l,t}^m | Z_{l-1,t}^m) P(Z_{l,t}^m | Z_{l,t-1}^m). \tag{9.24}$$

Figure 9.5 shows the evolution of $Z_{l,t}^m$ during the optimization and demonstrates the reduced oscillation, which effectively also leads to less noisy view sampling.

9.3.4 Photometric Consistency

Zheng et al. [394] employ NCC to compute the color similarity ρ_l^m. NCC is statistically optimal for Gaussian noise but is especially vulnerable to producing blurred depth discontinuities [138]. Inspired by prior works [34, 373], we diminish these artifacts by using a bilaterally weighted adaption of NCC. We compute ρ_l^m between a reference patch $\boldsymbol{w}_l$ at $\boldsymbol{x}_l$ with a corresponding source patch $\boldsymbol{w}_l^m$ at $\boldsymbol{x}_l^m$ as

$$\rho_l^m = \frac{\mathrm{cov}_w(\boldsymbol{w}_l, \boldsymbol{w}_l^m)}{\sqrt{\mathrm{cov}_w(\boldsymbol{w}_l, \boldsymbol{w}_l)\ \mathrm{cov}_w(\boldsymbol{w}_l^m, \boldsymbol{w}_l^m)}} \tag{9.25}$$

with the weighted covariance

$$\mathrm{cov}_w(\boldsymbol{x}, \boldsymbol{y}) = E_w(\boldsymbol{x} - E_w(\boldsymbol{x}))\ E_w(\boldsymbol{y} - E_w(\boldsymbol{y})) \tag{9.26}$$

and the weighted average

$$E_w(\boldsymbol{x}) = \frac{\sum_i w_i x_i}{\sum_i w_i}\ . \tag{9.27}$$

The per-pixel weight

$$w_i = \exp(-\frac{\Delta g_i^2}{2\sigma_g^2} - \frac{\Delta x_i^2}{2\sigma_x^2}) \tag{9.28}$$

indicates the likelihood that a pixel i in the local patch belongs to the same plane as its center pixel at l. It is a function of the grayscale color distance $\Delta g_i = |g_i - g_l|$ and the spatial distance $\Delta x_i = \|\boldsymbol{x}_i - \boldsymbol{x}_l\|$, whose importance is relatively scaled by the Gaussian dispersion σ_g and σ_x. By integrating the bilaterally weighted NCC into the term $P(X_l^m | Z_l^m, \theta_l, \boldsymbol{n}_l)$, our method achieves more accurate results at occlusion boundaries, as shown in Section 9.4.

9.3.5 Geometric Consistency

MVS typically suffers from gross outliers due to noise, ambiguities, occlusions, etc. In these cases, the photometric consistency for different hypotheses is ambiguous as large depth variations induce only small cost changes. Spatial smoothness constraints can often reduce but not fully eliminate the resulting artifacts. A popular approach to filter these outliers is to enforce multi-view depth coherence through left-right consistency checks as a post-processing step [34, 97].

In contrast to most approaches, we integrate multi-view geometric consistency constraints into the inference to increase both the completeness and the accuracy. Similar to Zhang et al. [392], we infer the best depth and normal based on both photometric and geometric consistency in multiple views. Since photometric ambiguities are usually unique to individual views (except textureless surfaces), exploiting the information from multiple views can often help to pinpoint the right solution. We compute the geometric consistency between two views as the forward-backward reprojection error

$$\psi_l^m = \|\boldsymbol{x}_l - \boldsymbol{H}_l^m \boldsymbol{H}_l \boldsymbol{x}_l\|\ , \tag{9.29}$$

where $\boldsymbol{H}_l^m$ denotes the projective backward transformation from the source to the reference image. It is composed from the source image estimates $(\theta_l^m, \boldsymbol{n}_l^m)$ interpolated at the forward projection $\boldsymbol{x}_l^m = \boldsymbol{H}_l \boldsymbol{x}_l$. Intuitively, the estimated depths and normals are consistent if the reprojection error ψ_l^m is small. Due to computational constraints, we cannot consider the occlusion indicators in the source image for the backward projection. Hence, to handle occlusion in the source image, we employ a robustified geometric cost in

$$\xi_l^m = 1 - \rho_l^m + \eta \min\left(\psi_l^m, \psi_{\max}\right) \tag{9.30}$$

using $\eta = 0.5$ as a constant regularizer and $\psi_{\max} = 3\text{px}$ as the maximum forward-backward reprojection error. Then, the optimal depth and normal is chosen as

$$(\hat{\theta}_l^{\text{opt}}, \hat{\boldsymbol{n}}_l^{\text{opt}}) = \underset{\theta_l^*, \boldsymbol{n}_l^*}{\arg\min} \frac{1}{|S|} \sum\nolimits_{m \in S} \xi_l^m(\theta_l^*, \boldsymbol{n}_l^*). \tag{9.31}$$

The geometric consistency term is modeled as $P(\theta_l, \boldsymbol{n}_l | \theta_l^m, \boldsymbol{n}_l^m)$ in the likelihood function, and Section 9.3.6 shows how to integrate its inference into the overall optimization framework. Experiments in Section 9.4 demonstrate how this formulation improves both the accuracy and the completeness of the results.

Algorithm 2 Overview of the proposed depth and normal estimation algorithm.

Function: *Coordinate Descent*	**Function**: *Sweep*
Input: Images $\boldsymbol{X}$	**Input**: Images $X^{\text{ref}}, \boldsymbol{X}^{\text{src}}$
Output: Depths $\boldsymbol{\theta}$, Normals $\boldsymbol{N}$	**Output**: Depths $\boldsymbol{\theta}^{\text{ref}}$, Normals $\boldsymbol{N}^{\text{ref}}$

For $m = 1$ to M	**For** $r = 1$ to 4 — Eq.
Set $X^{\text{ref}} = X^m$	Rotate reference image by 90°
Set $X^{\text{src}} = \boldsymbol{X} \setminus \{X^m\}$	**For** $l = L$ to 1
For $i = 1$ to I_1	**For** $m = 1$ to M
Sweep with Equation 9.14	Compute backward message $\overleftarrow{m}_l^m$ — 9.35
For $i = 1$ to I_2	**For** $l = 1$ to L
For $m = 1$ to M	**For** $m = 1$ to M
Set $X^{\text{ref}} = X^m$	Compute forward message $\overrightarrow{m}_l^m$ — 9.34
Set $X^{\text{src}} = \boldsymbol{X} \setminus \{X^m\}$	Compute $q(Z_l^m)$ — 9.33
Sweep with Equation 9.31	PatchMatch propagation/sampling — 9.22
	Estimate $(\theta_l^*, \boldsymbol{n}_l^*)$ over $q_l(\theta_l, \boldsymbol{n}_l)$ — 9.15/9.32
	For $m = 1$ to M
	Recompute forward message $\overrightarrow{m}_l^m$ — 9.34

9.3.6 Integration

This section contextualizes the individual terms of the proposed algorithm by explaining their integration into the overall optimization framework [394]. The joint likelihood function $P(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{N})$ of our proposed algorithm is defined as

$$\prod_{l=1}^{L} \prod_{m=1}^{M} [P(Z_{l,t}^m | Z_{l-1,t}^m, Z_{l,t-1}^m) P(X_l^m | Z_l^m, \theta_l, \boldsymbol{n}_l) P(\theta_l, \boldsymbol{n}_l | \theta_l^m, \boldsymbol{n}_l^m)]$$

over the input images $\boldsymbol{X}$, the occlusion indicators $\boldsymbol{Z}$, the depths $\boldsymbol{\theta}$, the normals $\boldsymbol{N}$, and is composed of several individual terms. First, the spatial and temporal smoothness term $P(Z_{l,t}^m|Z_{l-1,t}^m, Z_{l,t-1}^m)$ (see Section 9.3.3) enforces spatially smooth occlusion maps with reduced temporal oscillation during the optimization. Second, the photometric consistency term $P(X_l^m|Z_l^m, \theta_l, \boldsymbol{n}_l)$ uses bilateral NCC (see Section 9.3.4) and a slanted plane-induced homography (see Section 9.3.1) to compute the color similarity ρ_l^m between the reference and source images. Third, the geometric consistency term $P(\theta_l, \boldsymbol{n}_l|\theta_l^m, \boldsymbol{n}_l^m)$ to enforce multi-view consistent depth and normal estimates. The photometric and geometric consistency terms are computed using Monte-Carlo view sampling from the distribution $P_l(m)$ in Equation 9.22. The distribution encourages the sampling of non-occluded source images with informative and non-degenerate viewing geometry (see Section 9.3.2).

Analog to Zheng et al. [394], we factorize the real posterior $P(\boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{N}|\boldsymbol{X})$ in its approximation $q(\boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{N}) = q(\boldsymbol{Z})q(\boldsymbol{\theta}, \boldsymbol{N})$ [32]. Furthermore, for tractability, we constrain $q(\boldsymbol{\theta}, \boldsymbol{N})$ to the family of Kronecker delta functions

$$q(\theta_l, \boldsymbol{n}_l) = \delta(\theta_l{=}\theta_l^*, \boldsymbol{n}_l{=}\boldsymbol{n}_l^*) \ . \tag{9.32}$$

Variational inference then aims to infer the optimal member of the family of approximate posteriors to find the optimal $\boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{N}$. The validity of using GEM for this type of problem has already been shown in [228, 394]. To infer $q(Z_{l,t}^m)$ in iteration t of the E step of GEM, we employ the forward-backward algorithm as

$$q(Z_{l,t}^m) = \frac{1}{A}\overrightarrow{m}(Z_{l,t}^m)\overleftarrow{m}(Z_{l,t}^m) \tag{9.33}$$

with $\overrightarrow{m}(Z_{l,t}^m)$ and $\overleftarrow{m}(Z_{l,t}^m)$ being the recursive forward and backward messages

$$\overrightarrow{m}(Z_l^m) = P(X_l^m|Z_l^m, \theta_l, \boldsymbol{n}_l) \sum\nolimits_{Z_{l-1}^m} \overrightarrow{m}(Z_{l-1}^m) P(Z_{l,t}^m|Z_{l-1,t}^m, Z_{l,t-1}^m) \tag{9.34}$$

$$\overleftarrow{m}(Z_l^m) = \sum\nolimits_{Z_{l+1}^m} \overleftarrow{m}(Z_{l+1}^m) P(X_{l+1}^m|Z_{l+1}^m, \theta_{l+1}, \boldsymbol{n}_{l+1}) P(Z_{l,t}^m|Z_{l+1,t}^m, Z_{l,t-1}^m) \tag{9.35}$$

using an uninformative prior $\overrightarrow{m}(Z_0^m) = \overrightarrow{m}(Z_{L+1}^m) = 0.5$. The variable $q(Z_{l,t}^m)$ together with $q(\alpha_l^m), q(\beta_l^m), q(\kappa_l^m)$ determine the view sampling distribution $P_l(m)$ used in the M step of GEM as defined in Equation 9.22. The M step uses PatchMatch propagation and sampling (see Section 9.3.1) for choosing the optimal depth and normal parameters over $q(\theta_l, \boldsymbol{n}_l)$. Since geometrically consistent depth and normal inference is not feasible for all images simultaneously due to memory constraints, we decompose the inference in two stages. In the first stage, we estimate initial depths and normals for each image in the input set $\boldsymbol{X}$ according to Equation 9.14. In the second stage, we use coordinate descent optimization to infer geometrically consistent depths and normals according to Equation 9.31 by keeping all images but the current reference image as constant. We interleave the E and M step in both stages using row- and column-wise propagation. Four propagations in all directions denote a sweep. In the second stage, a single sweep defines a coordinate descent

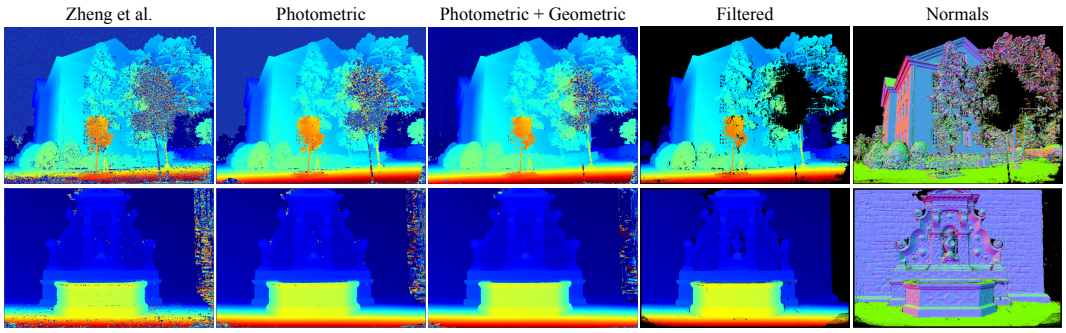

Figure 9.3: Reconstruction results for the *South Building* [62, 116] and *Fountain* [322] datasets. From left to right: Depth map by Zheng et al. [394], then ours only with the photometric term, with the photometric and geometric terms, and the final filtered depth and normal maps.

step, i.e., we alternate between different reference images after propagating through the four directions. Typically, the first stage converges after $I_1 = 3$ sweeps, while the second stage requires another $I_2 = 2$ sweeps through the entire image collection to reach a stable state. 2 provides an overview of the steps of our algorithm.

9.3.7 Filtering and Fusion

After describing the depth and normal inference, this section proposes a robust method to filter any remaining outliers, e.g., in textureless sky regions. In addition to the benefits described previously, the photometric and geometric consistency terms provide us with measures to robustly detect outliers at negligible computational cost. An inlier observation should be both photometrically and geometrically stable with support from multiple views. The sets

$$\mathcal{S}_l^{\text{pho}} = \{\boldsymbol{x}_l^m \mid q(Z_l^m) > \bar{q}_Z\} \tag{9.36}$$

$$\mathcal{S}_l^{\text{geo}} = \{\boldsymbol{x}_l^m \mid q(\alpha_l^m) \geq \bar{q}_\alpha, q(\beta_l^m) \geq \bar{q}_\beta, q(\kappa_l^m) > \bar{q}_\kappa, \psi_l^m < \psi_{\max}\} \tag{9.37}$$

determine the photometric and geometric support of a reference image pixel $\boldsymbol{x}_l$. To satisfy both constraints, we define the effective support of an observation as

$$\mathcal{S}_l = \{\boldsymbol{x}_l^m \mid \boldsymbol{x}_l^m \in \mathcal{S}_l^{\text{pho}}, \boldsymbol{x}_l^m \in \mathcal{S}_l^{\text{geo}}\} \tag{9.38}$$

and filter any observations with $|\mathcal{S}_l| < s$. In all our experiments, we set $s = 3$, $\bar{q}_Z = 0.5$, $\bar{q}_\alpha = 1$, $\bar{q}_\beta = 0.5$, and $\bar{q}_\kappa = P(\kappa = 90°)$. Figure 9.3 and Figure 9.8 show examples of filtered depth and normal maps.

The collection of support sets $\mathcal{S}$ over the observations in all input images defines a directed graph of consistent pixels. In this graph, pixels with sufficient support are nodes, and directed edges point from a reference to a source image pixel. Nodes are associated with depth and normal estimates and, together with the intrinsic and

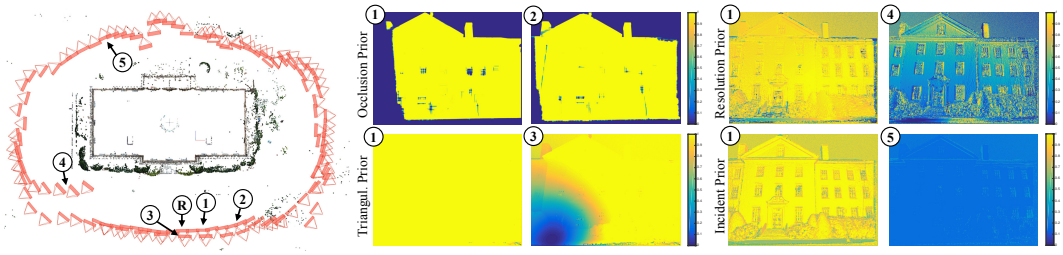

Figure 9.4: Photometric and geometric priors for the *South Building* dataset [116] between reference image (R) and each two selected source images (1-5).

extrinsic calibration, edges define a projective transformation from the reference to the source pixel. Our fusion finds clusters of consistent pixels in this graph by initializing a new cluster using the node with maximum support $|\mathcal{S}|$ and recursively collecting connected nodes that satisfy three constraints. Towards this goal, we project the first node into 3D to obtain the location $\boldsymbol{p}_0$ and normal $\boldsymbol{n}_0$. For the first constraint, the projected depth $\tilde{\theta}_0$ of the first node into the image of any other node in the cluster must be consistent (cf. Merrel et al. [213]) with the estimated depth θ_i of the other node such that

$$\frac{|\tilde{\theta}_0 - \theta_i|}{\tilde{\theta}_0} < \epsilon_\theta \quad . \tag{9.39}$$

Second, the normals of the two must be consistent such that $1 - \boldsymbol{n}_0^T \boldsymbol{n}_i < \epsilon_n$. Third, the reprojection error ψ_i of $\boldsymbol{p}_0$ with respect to the other node must be smaller than $\bar{\psi}$. Note that the graph can have loops, and therefore we only collect nodes once. In addition, multiple pixels in the same image can belong to the same cluster and, by choosing $\bar{\psi}$, we can control the resolution of the fused point cloud. When there is no remaining node that satisfies the three constraints, we fuse the cluster's elements, if it has at least three elements. The fused point has median location $\hat{\boldsymbol{p}}_j$ and mean normal $\boldsymbol{n}_j$ over all cluster elements. The median location is used to avoid artifacts when averaging over multiple neighboring pixels at large depth discontinuities. Finally, we remove the fused nodes from the graph and initialize a new cluster with maximum support $|\mathcal{S}|$ until the graph is empty. The resulting point cloud can then be colored (e.g. using the approach by Wächter et al. [353]) for visualization purposes and, since the points already have normals, we can directly apply meshing algorithms (e.g. using the Poisson reconstruction algorithm [160]) as an optional step.

9.4 Experiments

This section first demonstrates the benefits of the proposed contributions in isolation. Following that, we compare to other methods and show state-of-the-art results

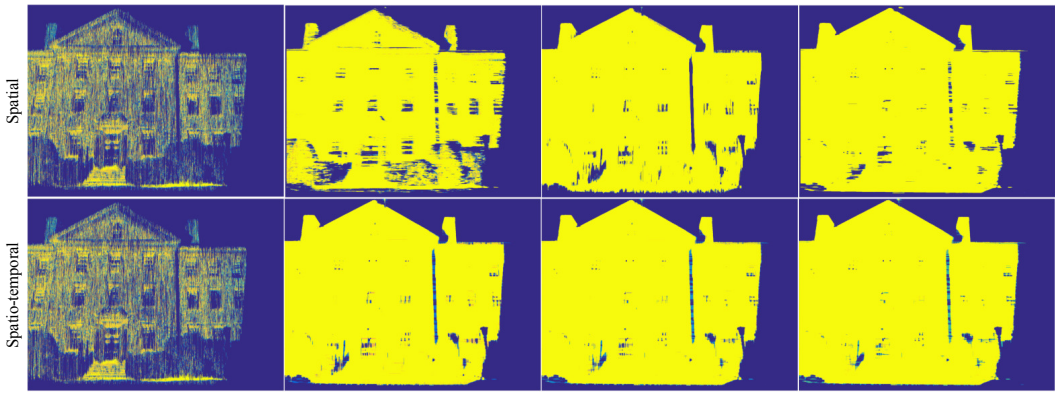

Figure 9.5: Comparison of spatial smoothness term [394] with our proposed spatial and temporal smoothness term for the occlusion variables $\boldsymbol{Z}$. Algorithm starts from the left with the first sweep and is followed by consecutive sweeps to the right.

on both low- and high-resolution benchmark datasets. Finally, we evaluate the performance of our algorithm in the challenging setting of large-scale Internet photo collections. The algorithm lends itself for massive parallelization on the row- and column-wise propagation and the view level. In all our experiments, we use a CUDA implementation of our algorithm on a NVIDIA Titan X GPU. We set $\gamma = 0.999$, leading to an average of one occlusion indicator state change per 1000 pixels. Empirically, we choose $\sigma_\rho = 0.6$, $\bar{\alpha} = 1°$, and $\sigma_k = 45°$.

9.4.1 Components

This section shows the benefits of the individual components in isolation based on the *South Building* dataset [116], which consists of 128 unstructured images with a resolution of 7MP. We obtain sparse reconstructions using SFM [129]. For each reference view, we use all 127 images as source views with an average runtime of 50s per sweep.

Normal Estimation

Figure 9.3 shows depth maps using fronto-parallel homographies (1st column) and with normal estimation (2nd to 5th columns), which leads to increased completeness and accuracy for depth inference of oblique scene elements, such as the ground. In addition, our method estimates more accurate normals than standard PatchMatch (see Figure 9.6). Due to the proposed PatchMatch sampling scheme, our algorithm requires the same number sweeps to converge and only approximately 25% more runtime due to more hypotheses as compared to Zheng et al. [394], who only estimate per-pixel depths.

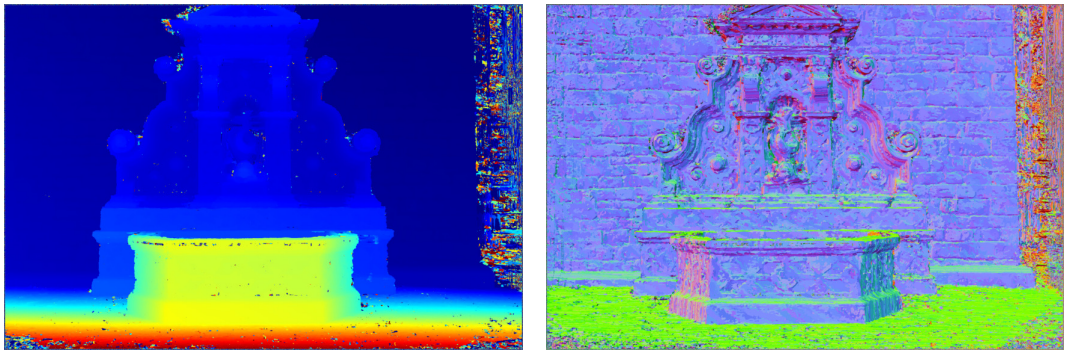

Figure 9.6: Reconstructed depth and normal maps for the *Fountain* dataset [322] using standard PatchMatch propagation [34]. See Figure 9.3 to compare against the results of our proposed approach.

Geometric Priors

Figure 9.4 demonstrates the benefit of each geometric prior. We show the likelihood functions for the reference view against one representative source image. For all priors, we observe varying likelihood within the same source image, underlining the benefit of pixel-wise view selection. The priors correctly downweigh the influence of source images with small triangulation angle, low resolution, or occluded views.

Selection Smoothness

Figure 9.5 shows that our temporal smoothness term effectively mitigates the oscillation of the pure spatial smoothness term. While the occlusion variables in the formulation by Zheng et al. [394] oscillate depending on the propagation direction, in our method they quickly converge in a stable state leading to more stable view sampling.

Geometric Consistency

Figure 9.3 demonstrates improved completeness when incorporating the geometric consistency term, and it also allows to reliably detect outliers for practically outlier-free filtered results. To measure the quantitative impact of our contributions, we obtain benchmark results by omitting a single component or combinations of components from the formulation (see Table 9.1). We observe that each component is important to achieve the overall accuracy and completeness of our method.

9.4.2 Benchmarks

Middlebury

This benchmark [278] consists of the *Dino* and *Temple* models captured at 640x480 under varying settings (*Full*, *Ring*, *Sparse*). For each reference image, we use all

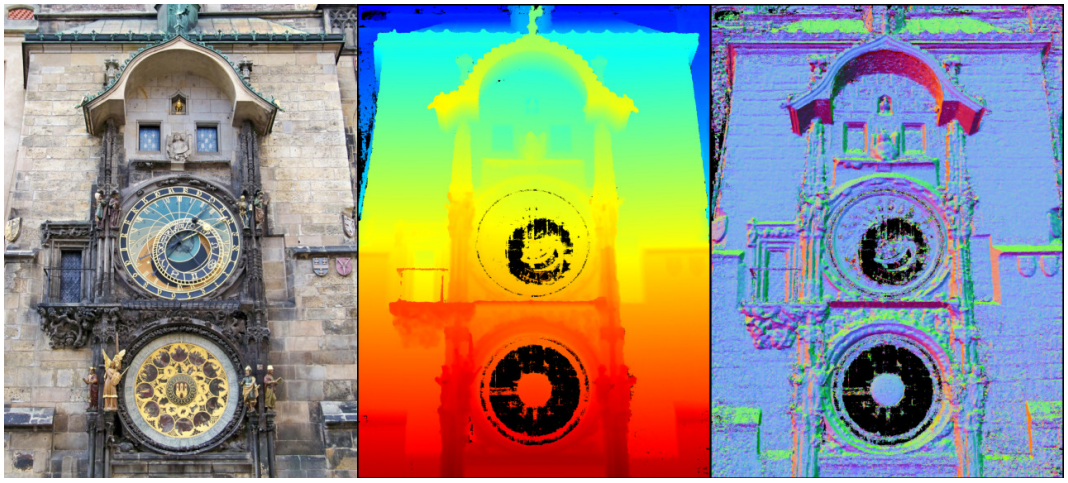

Figure 9.7: Reference image with filtered depth and normal maps for a crowd-sourced Internet image of the Astronomical Clock in Prague, Czech Republic.

views as source images at a runtime of approximately 40*s* per view for the *Full* models with around 300 images. We achieve excellent accuracy and completeness on both models[1]. Specifically, using the standard settings, we rank 1st for *Dino Full* (tied) and *Dino Sparse*, while achieving competitive scores for the *Temple* (4th for *Full*, 8th for *Ring*). Note that our method performs best for higher resolutions, as normal estimation needs large patch sizes. Also, we use basic Poisson meshing [160], underlining the highly accurate and outlier-free depth/normal estimates produced by our method.

Strecha

This benchmark [322] consists of high-resolution images with ground-truth, and we follow the evaluation protocol of Hu and Mordohai [144]. Figure 9.3 shows outputs for the *Fountain* dataset and, Table 9.1 lists the results quantifying both the accuracy and completeness. To maintain comparability against Zheng et al. [394], we evaluate our raw depth maps against the ground-truth. We produce significantly more accurate and complete results than Zheng et al., and we outperform the other methods in 3 of 4 categories, even though the results of [144, 147, 346] are evaluated based on the projection of a 3D surface obtained through depth map fusion.

ETH3D

Table 9.2 shows results for the ETH3D multi-view stereo benchmark, which covers a variety of indoor and outdoor scenes. The images were taken using a high-resolution

[1]Full results online at `http://vision.middlebury.edu/mview/eval/`.

	Fountain		Herzjesu	
	2cm	10cm	2cm	10cm
Zheng et al. [394]	0.769	0.929	0.650	0.844
Hu et al. [144]	0.754	0.930	0.649	0.848
Furukawa et al. [95]	0.731	0.838	0.646	0.836
Zaharescu et al. [383]	0.712	0.832	0.220	0.501
Tylecek et al. [346]	0.732	0.822	0.658	0.852
Jancosek et al. [147]	0.824	0.973	**0.739**	0.923
Galliani et al. [97]	0.693	0.838	0.283	0.455
Ours $\setminus N$	0.799	0.937	0.673	0.901
Ours $\setminus P$	0.824	0.972	0.686	0.928
Ours $\setminus S$	0.825	0.973	0.688	0.927
Ours $\setminus B$	0.826	0.973	0.690	0.929
Ours $\setminus PSB$	0.817	0.965	0.688	0.921
Ours $\setminus G$	0.804	0.949	0.679	0.907
Ours	**0.827**	**0.975**	0.691	**0.931**

Table 9.1: Strecha benchmark [322] with reported values from [144]. Ratio of pixels with error less than 2cm and 10cm. Ablation study for ours without normals (\N), without geometric priors (\P), without temporal smoothness (\S), without geometric consistency (\G), without bilateral NCC (\B), and with all components.

	all	high-res multi-view	indoor	outdoor	botanical garden	boulders	bridge	door	exhibition hall	lecture room	living room	lounge	observatory	old computer	statue	terrace 2
LTVRE	69.57	76.25	74.54	81.41	88.60	64.38	79.24	89.12	70.76	69.79	87.86	49.09	93.20	56.21	80.16	86.65
AMHMVS	67.68	75.89	73.93	81.77	88.71	66.61	88.62	89.01	69.81	66.78	86.70	41.95	91.51	55.64	78.14	87.17
Ours	66.92	73.01	70.41	80.81	87.13	65.63	88.30	84.19	62.96	63.80	87.69	38.04	92.56	46.66	74.91	84.24
OpenMVS	64.09	70.56	68.19	77.65	80.26	59.41	85.01	82.96	58.76	65.43	82.33	37.63	91.23	45.34	76.03	82.32
CMPMVS	51.72	70.19	68.16	76.28	73.15	57.55	79.47	83.92	48.02	56.48	84.81	43.13	86.42	62.67	81.81	84.89
Gipuma	–	45.18	41.86	55.16	77.03	51.52	32.50	80.52	32.46	27.12	35.02	9.52	75.50	22.31	60.21	38.45
PMVS	37.38	44.16	40.28	55.82	40.82	42.54	65.12	63.99	34.43	24.48	57.12	3.92	76.17	17.10	55.52	48.75
MVE	26.22	30.37	25.89	43.81	44.55	35.46	28.87	39.07	18.20	18.58	40.18	2.68	47.37	17.07	23.78	48.60
FPMVS	–	–	–	59.81	65.08	51.03	–	69.58	41.46	40.67	72.32	16.46	70.00	24.24	65.42	58.40

Table 9.2: F_1-score results for the ETH3D benchmark [290] on the test datasets using the default error threshold of 2cm.

DSLR camera in an unstructured setup. Ground truth reconstructions have been obtained using a high-precision laser scanner. The laser scans are accurately aligned to the images using a robust photometric optimization algorithm. For each DSLR image, the benchmark provides an accurate intrinsic and extrinsic calibration as an input to each method. The performance of each method is measured in terms of completeness and accuracy with respect to the ground truth laser scan point cloud. As a single performance metric, the F_1-score is used to combine the completeness and accuracy scores. Our method achieves top performance and is ranked third in the benchmark. Note that the top performing method explicitly regularizes the geometry in 3D and could be combined with our method easily, which does not perform any explicit regularization.

Tanks and Temples

Table 9.3 lists the results for the Tanks and Temples benchmark [167]. This benchmark measures the performance of an end-to-end image-based 3D modeling system, including the sparse and dense reconstruction stages. The authors of this benchmark acquired several highly accurate and complete laser scans of indoor and outdoor scenes, which are used as ground truth for the evaluation. The performance is measured on the quality of the final dense reconstruction output using the F_1-score, which combines precision and recall in a single metric. Our sparse and dense reconstruction pipeline achieves the best overall results on both the intermediate and advanced datasets among all of the evaluated approaches.

9.4.3 Internet Photos

We densely reconstruct models of 100M Internet photos released by Heinly et al. [129, 284] using a single machine with 4 NVIDIA Titan X. We process the 41K images at a rate of 70s per view using 2 threads per GPU and finish after 4.2 days in addition to the 6 days needed for sparse modeling using SFM. Whenever we reach

		Bundler + PMVS	*Ours*	*MVE*	*MVE + SMVS*	*OpenMVG + MVE*	*OpenMVG + OpenMVS*	*OpenMVG-G + OpenMVS*	*OpenMVG + PMVS*	*OpenMVG + SMVS*	*Pix4D*	*Theia-G + OpenMVS*	*Theia-I + OpenMVS*	*VisualSfM + CMPMVS*	*VisualSfM + OpenMVS*	*VisualSfM + PMVS*
Intermediate	Family	16.91	50.41	48.59	30.42	49.91	58.86	56.50	41.03	31.93	64.45	47.95	48.11	35.41	49.10	38.02
	Francis	4.34	22.25	23.84	16.64	28.19	32.59	29.63	17.70	19.92	31.91	19.52	19.38	14.11	21.38	12.93
	Horse	3.82	25.63	12.70	10.44	20.75	26.25	21.69	12.83	15.02	26.43	19.56	20.66	14.71	18.59	11.30
	Lighthouse	22.49	56.43	5.07	39.16	43.35	43.12	6.55	36.68	39.38	54.41	28.90	30.02	37.75	25.24	41.75
	M60	23.80	44.83	39.62	34.35	44.51	44.73	39.54	35.93	36.51	50.58	16.25	30.37	12.02	27.02	35.47
	Panther	21.54	46.97	38.16	37.90	44.76	46.85	28.48	33.20	41.61	35.37	21.54	30.79	24.29	24.64	34.19
	Playground	0.53	48.53	5.81	2.40	36.58	45.97	0.00	31.78	35.89	47.78	23.45	23.65	27.26	16.59	35.47
	Train	9.42	42.04	29.19	21.44	35.95	35.27	0.53	28.10	25.12	34.96	10.24	20.46	13.62	13.07	13.26
	Mean	12.86	42.14	25.37	24.09	38.00	41.70	22.86	29.66	30.67	**43.24**	23.43	27.93	22.40	24.45	27.80
	Rank	18.25	**4.50**	12.25	14.50	7.00	4.62	11.88	12.88	11.38	4.52	14.88	13.00	15.12	14.00	13.62
Advanced	Auditorium	0.00	16.02	4.11	0.97	14.70	9.79	1.89	4.54	6.96	10.83	5.74	6.23	4.70	7.94	4.68
	Ballroom	4.05	25.23	12.63	6.76	26.36	22.49	9.16	12.09	11.58	18.53	13.63	13.73	8.07	15.21	10.84
	Courtroom	10.30	34.70	27.93	16.97	32.48	26.54	24.61	21.00	19.82	33.21	16.08	18.43	13.17	21.21	16.36
	Museum	11.15	41.51	34.67	19.72	37.57	36.89	26.18	29.17	21.89	47.37	15.51	18.55	8.66	19.78	20.00
	Palace	2.71	18.05	13.58	7.74	3.65	14.64	4.02	6.76	8.90	14.47	6.43	10.61	3.89	9.10	7.32
	Temple	5.45	27.94	16.79	7.98	22.84	20.76	14.14	12.72	12.27	26.01	11.77	11.58	6.95	2.99	2.12
	Mean	5.61	**27.24**	18.29	10.02	22.93	21.85	13.33	14.38	13.57	25.07	11.53	13.19	7.57	12.70	10.22
	Rank	15.50	**2.33**	7.33	12.50	5.33	4.67	10.33	9.50	9.00	3.33	11.17	9.33	13.67	8.83	12.00

Table 9.3: F_1-score results for the Tanks and Temples benchmark [167] on the intermediate and advanced datasets.

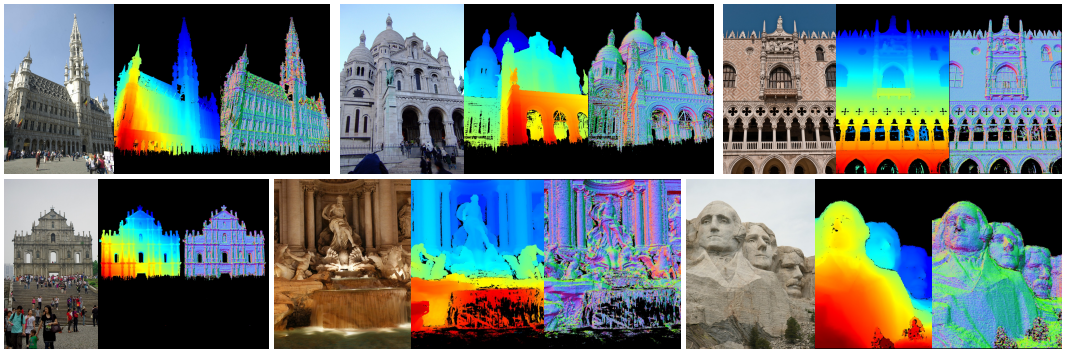

Figure 9.8: Reference image with filtered depth and normal maps reconstructed from unstructured Internet images.

Figure 9.9: Examples of dense point clouds for reconstructed Internet photo datasets produced by the pipeline by Heinly et al. [5]. From top left to bottom right: Milan Cathedral, Italy; Piazza dei Miracoli, Italy; Reichstag, Germany; Temple in Kyoto, Japan; St. Vitus Cathedral, Czech Republic; Piazza San Marco, Italy; St. Paul's Cathedral, England; Peter's Dome, Vatican; Pantheon, Italy; Sagrada Familia (front), Spain; Sagrada Familia (back), Spain; British Museum, England; Florence Cathedral, Italy; Sistine Chapel, Vatican; Piazza della Signora, Italy; Piazza Public, Italy.

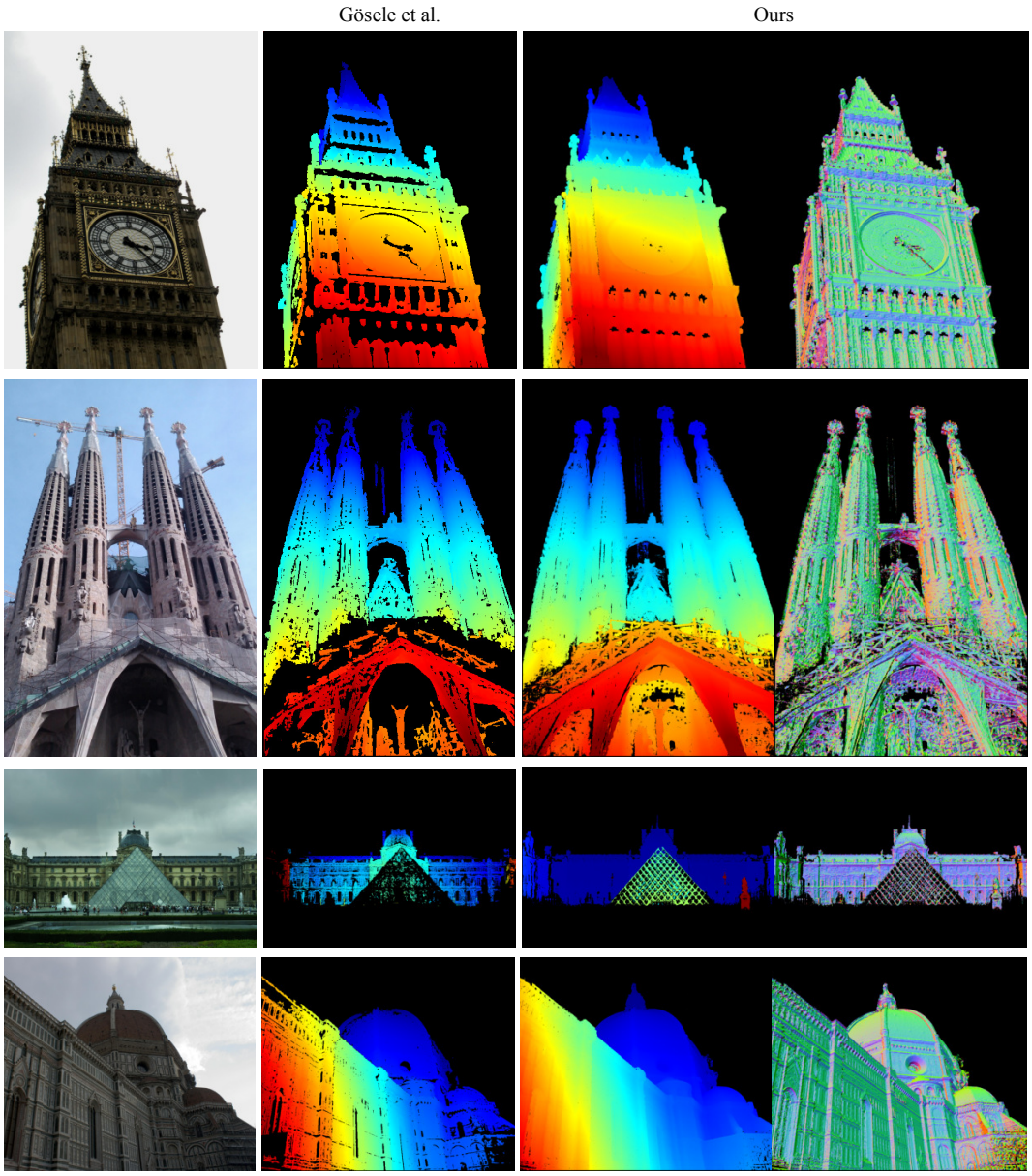

Figure 9.10: Comparison of our results against Gösele et al. [43] on Internet photo collections [5]. From left to right: reference image, depth map by Gösele et al., our depth and normal maps. From top to bottom: Big Ben, England; Sagrada Familia, Spain; Louvre, France; Florence Cathedral, Italy.

the GPU memory limits, we select the most connected source images ranked by the number of shared sparse points. Usually, this limit is reached for $\approx$ 200 images, while image sizes vary from 0.01MP to 9MP. The fusion and filtering steps consume negligible runtime. Figure 9.1 and Figure 9.9 show fused point clouds, Figure 9.8 and Figure 9.7 show depth/normal maps, while Figure 9.10 provides more results and comparisons against [93, 95, 110].

9.5 Summary

This chapter proposed a novel algorithm for robust and efficient dense reconstruction from unstructured image collections. Our proposed method estimates accurate depth and normal information using photometric and geometric information for pixelwise view selection and for image-based fusion and filtering. Extensive experiments on existing benchmarks and challenging Internet datasets demonstrated state-of-the-art results in comparison to other state-of-the-art dense reconstruction approaches.

Part V

Systems and Applications

10 From Single Image Query To Detailed 3D Model

As demonstrated in the previous chapters, the computer vision community has made great progress in the areas of image retrieval and image-based 3D modeling. Current image search engines operate on web-scale image collections and are able to localize specific objects and landmarks, and aid user-friendly content browsing. In the field of reconstructing scenes from images and videos, arguably the biggest steps have been made in 3D modeling from unstructured Internet photo collections. Typically, both components have been employed as mostly independent entities. In this work, we propose an end-to-end image-based 3D reconstruction system that tightly integrates the two components in order to produce a highly accurate and detailed 3D model of a scene. Our proposed system is a natural step forward in that it addresses the problem of obtaining a detailed 3D model of an object depicted in a single, user-provided photograph.

Image-based 3D modeling systems have been extended from modeling scenes from a few thousand images [308, 310] to modeling city-scale photo collections of millions of images [89, 129]. Early photo collection reconstruction systems leverage exhaustive matching of image pairs to determine possible overlapping image pairs. This is generally quadratic in the number of images and features. Hence, this approach does not scale and is not applicable to datasets containing thousands or even millions of images, which are commonly available. However, exhaustive matching guarantees the discovery of all possible camera overlaps. To achieve scalability, the current state-of-the-art large-scale reconstruction systems abandon the exhaustive pairwise overlap determination. Instead, modern systems leverage image retrieval algorithms [66, 67, 231], or image-clustering techniques to identify overlapping images during reconstruction, as demonstrated by the systems of Agarwal et al. [4] and Frahm et al. [89]. While the introduction of image retrieval was essential to boosting the scalability of reconstruction methods on large datasets, it also severely impacted the ability to reconstruct fine details of the scene. This problem stems from the fact that the image pairs showing the details are often absent from the retrieval results. This is unsatisfactory, as for applications such as photo field of view extension using unordered photo collections, recently proposed by Zhang et al. [390], it is desirable to have the details present in the reconstruction.

The lack of detail is a result of the employed retrieval approaches [66, 67, 231], which are tuned to obtain images similar in scale and appearance. In this chapter, we introduce a tightly-coupled retrieval and image-based 3D reconstruction system for large-scale reconstruction from unordered photo collections of several million

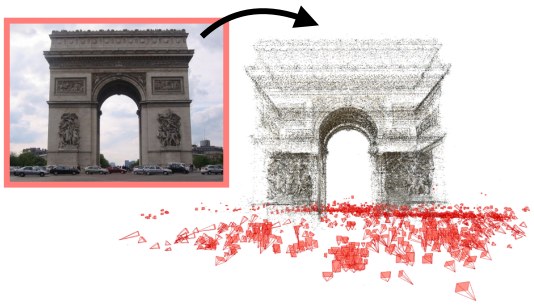

Figure 10.1: Arc de Triomphe, Paris. 3D reconstruction from a single input image (red inset) using 2,640 views around the landmark from a 7.4M image database. Only imagery is used (no GPS or text). The scene contains 395,431 points and the surface resolution reaches the order of 1mm in the most photographed areas.

Figure 10.2: Reconstruction of the Astronomical Clock in Prague, Czech Republic. Left: 3D model obtained from our retrieval and reconstruction system. Images illustrating the range of registered views from overview images to images of a specific architectural detail are shown alongside the model. Right: visualization of the surface resolution from high resolution in red (approximately 1mm surface resolution as obtained from a known object size in 3D) to low resolution in blue.

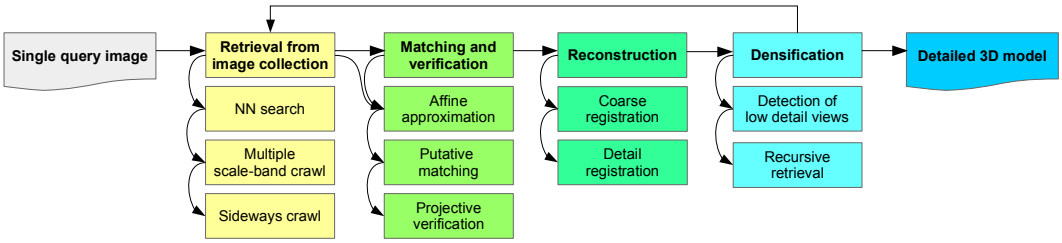

Figure 10.3: The proposed image-based 3D modeling pipeline that tightly couples image retrieval and Structure-from-Motion.

images, which not only recovers the coarse geometry of the scene but specifically focuses on modeling scene details. Our approach achieves this by combining SFM with retrieval across differently scaled scene images.

In order to achieve these detailed reconstructions, our system has to overcome the following challenges:

- Achieve a more balanced retrieval of overview and detailed images to provide the images needed for fine-detail reconstruction.
- Overcome the registration uncertainties that result from the large resolution differences between overview images and detailed images.

We resolve these challenges by proposing a tightly-coupled SFM and retrieval system. Establishing an interactive link between the reconstruction system and the retrieval system enables us to control the retrieval characteristics based on the current state-of-the-art 3D reconstruction. This allows us to specifically retrieve images that are required to overcome the challenges of SFM. Our resulting reconstructions from unordered Internet photo collections show high geometric detail while at the same time conveying the structure of the entire scene. An example reconstruction is shown in Figure 10.2.

10.1 Related Work

Our system simultaneously leverages retrieval and SFM algorithms to achieve the goal of detailed scene reconstruction. In this section, we discuss the relevant state-of-the-art in both areas, before introducing our method in more detail in the following sections.

Exploring a large unordered image collection by user defined image query – the problem of large-scale image retrieval – made significant progress during the last decade. Most of the approaches pose the problem as a nearest neighbor search in a descriptor space, such as bag-of-words [148, 231, 243, 307], VLAD [13, 151], Fischer vectors [240], or exhaustive matching [283, 365]. Recently, Mikulik et al. [219] pointed out that the nearest neighbor image search is not optimal for the user, who is typically looking for new image information rather than for near-duplicate images.

Novel formulations and efficient methods for extreme change of scale were proposed in [219] and for detailed image mining in [218]. We extend these ideas to identify initial sets of images suitable for 3D reconstruction and then leverage the obtained reconstructions to suggest further image retrieval goals. Instead of targeting the extreme scale changes that are attractive for a human user, the whole spectrum of scale transitions is sampled, which is more suitable for 3D reconstruction. Further, we propose an efficient retrieval method for content-based crawling around a landmark, mining for views connecting multiple sides of the landmark.

Scene reconstruction from Internet photo collections has been introduced in the seminal work of Snavely et al. [308, 310]. This was the first approach to show that SFM for such diverse and unordered collections of thousands of images is possible. The major limitation of this reconstruction system was its limited scalability due to exhaustive image pair overlap evaluation.

To overcome this lack of scalability, Li et al. [191] introduced an appearance-based clustering for grouping the images. This allowed modeling from tens of thousands of images on a single PC. Agarwal et al. [4] introduced a cloud computing algorithm to perform modeling from 150,000 images on 62 computers in less than 24 hours. The approach leveraged a vocabulary tree based search with query extension [66] to determine overlapping images, followed by approximate nearest neighbor feature matching. While providing scalability, such an approach severely impairs the retrieval of detailed images for registration and reconstruction. Lou et al. [198] proposed a modified vocabulary tree based retrieval enforcing diversity in the retrieval results. The proposed reweighting enhances scene coverage for the reconstruction with SFM, but it does not solve the problem of not retrieving detailed images.

Frahm et al. [4, 89] extended the approach of Li et al. [191] to scale to the reconstruction from millions of images. However, this approach also suffers from the use of recognition methods – gist-feature [234] based appearance grouping – that fail to obtain detailed images and thus severely limits the ability to produce fine-detail reconstructions.

Crandall et al. [75] proposed a global method that performs SFM based on a MRF optimization. In order to properly initialize this hybrid optimization, their approach requires approximate geo-location priors for the images. While this approach can retrieve geo-located detail images, a large fraction of Internet photo collection photos is not geo-located. Hence, this approach would be very restrictive in our scenario, and it would only register a fraction of the images compared to our approach.

10.2 System Overview

Our proposed pipeline (see Figure 10.3) handles image collections of the size of millions of unordered images. A single, user-provided query image serves as the input to our pipeline. In the first stage, the query image is used as the initial seed for image retrieval (see Section 10.3). The retrieval stage first finds nearest neighbor images, followed by a multiple scale-band crawl to obtain additional views

Figure 10.4: Terracotta Army, China. Samples of different scale-bands of the initial query image: context of the query image (zoom out – top left), two examples of mid-level detail (zoom in), and three detailed images for each of the mid-level band (rightmost). Two examples of the left and right side of the query are shown in the bottom left.

at different zoom levels (see Section 10.3.1). Furthermore, we expand the query by retrieving images to the left and right of the query image in order to obtain additional context around the query image (see Section 10.3.2). As a preparation for the subsequent reconstruction stage, we propose an efficient matching method (see Section 10.4), leveraging the by-products of the retrieval stage to intelligently avoid image pairs that do not overlap. The reconstruction stage employs several methods to overcome the challenges of detailed image registration (see Section 10.5). Finally, we perform additional densification (see Section 10.6) by identifying the low-detail parts of the model and recursively retrieving additional images for another round of reconstruction. In comprehensive experiments (see Section 10.8) on a dataset of millions of images, we demonstrate that the method produces large-scale, high-quality models that also capture the fine details of the scene.

10.3 Image Retrieval

The objective of retrieval for 3D reconstruction is to provide a matching graph with a variety of viewpoints (for the stability of the reconstruction), sequences of images providing a smooth transition between extreme viewpoints or scale changes (to be able to connect different parts and to help disambiguate duplicated structures), and mining for images of further structures in space and scale (to extend the reconstruction and improve the level of detail).

The retrieval engine builds upon bag-of-words representation with fast spatial verification [243]. Hessian affine features [216] are detected (1900 features per image on average) and described by the rotation-variant [239] SIFT descriptor [200]. The descriptors are vector-quantized into 16 million visual words using k-means with approximate nearest neighbor search [226]. Fast (several hundred image pairs per

second) spatial verification [243] then estimates an approximate affine transformation between query and result images. To enforce transformation consistency (scale change, translation), the scale and position information for each feature is included in the inverted file [148, 318].

In the proposed method, the initial matching graph is obtained from the image collection using the query image as an entry point. Depending on the intended result (either detailed reconstruction of the scene visible in the query image, or detailed reconstruction of the whole neighborhood of the query image) different mining techniques are used to generate the initial matching graph. Once a (partial) reconstruction is available, the same techniques are applied to incrementally extend the reconstruction; see Section 10.6.

10.3.1 Multiple Scale-Bands

To retrieve relevant images of various levels of detail (and/or different amounts of context), we build on the approach of hierarchical query expansion [218]. Unlike in [218], we are not interested in the extreme scale changes, but rather in an image sequence capturing a smooth transition in scale to support stable SFM estimation.

The hierarchical query expansion proceeds as follows. An initial query encouraging change of scale is issued with the query image. To reflect the scale change in image ranking, we use document at a time (DAAT) scoring [318] exploiting geometry stored in an inverted file. The results of this initial query are clustered in scale-space. Each spatial image cluster is then used to issue a new expanded query [66], which retrieves further details at the given location. Figure 10.4 shows four scale bands – context (zoom out), original scale, two examples of mid-level detail (zoom in), and three detailed images for each of the mid-level bands.

10.3.2 Sideways Crawl

Retrieving images of multiple scale-bands, starting from a single query image, yields the whole spectrum of image scales, but typically only from a single viewing direction. However, many interesting scene parts are often located around the corners of or next to the observed landmark. In this case, the reconstruction significantly benefits from additional sideways crawling around the landmark, in order to obtain more complete and stable models. The crawling is performed to the left or to the right with respect to the original query image. We propose a novel, efficient retrieval method for content-based crawling around an initial point of view. As a result, we successfully mine images connecting multiple sides of the landmark (see Figure 10.5), or images with a broader view of the whole area of interest in the case of indoor scenes (see Figure 10.4).

The sideways crawl retrieval consists of two stages. The first stage allows us to specifically crawl for images in different directions (left and right). In the second stage, the initial set of retrievals is extended by additional images from the desired direction.

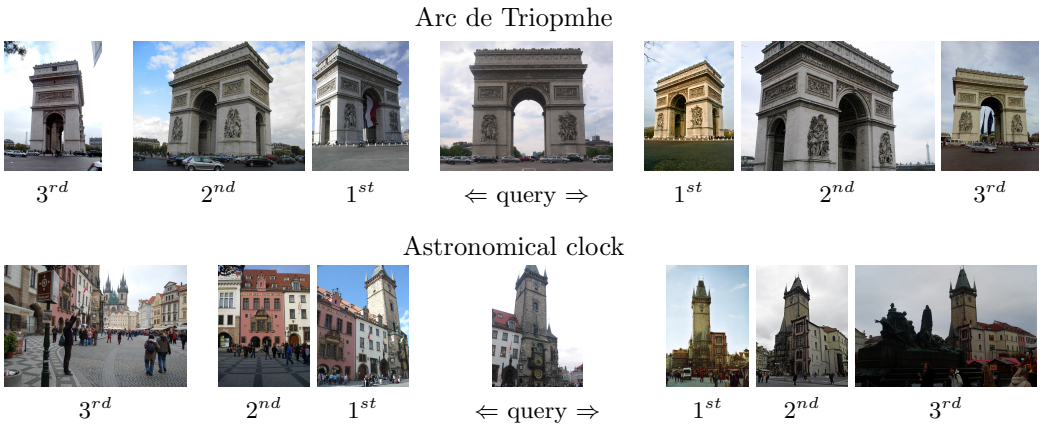

Figure 10.5: Sideways crawl of Arc de Triomphe and Astronomical Clock. Sample results of three recursive rightward and leftward queries extending the view of the original query image.

The first stage leverages the estimated geometric transformation (an affine transformation in our case) between the query image and the results. When taking a step to the right, for instance, features on the right-hand side of the query image should match a sufficient number of features on the left-hand side of the result image. This can be achieved by geometric re-ranking of the shortlisted results or more efficiently using the DAAT approach [318]. Additional geometric analysis, such as estimating the position of the horizontal vanishing point through homography fitting, can be performed at additional computational effort.

To retrieve a larger set of relevant images that contain novel image information, additional queries are executed using the top ranked images from the first stage. In these expanding queries, only features from areas not visible in the original query image are considered. Finally, retrieved images are merged and then re-ranked based on the amount of viewpoint change.

To obtain reconstruction of a landmark from all sides, the sideways crawl is repeated, as illustrated in Figure 10.5. In order to also reconstruct details on all sides of the landmark, each sideways step is followed by multiple scale-band mining. For further examples on the sideways crawl, see an indoor view of the Terracotta Army landmark in China (see Figure 10.4). Having introduced the image-retrieval system, we next detail our SFM system that exploits the unique characteristics of the proposed retrieval system.

10.4 Matching and Geometric Verification

The state-of-the-art SFM systems [4, 89] have achieved impressive results on city-scale reconstructions from unordered photo collections. One frequent reason for the lack of detail reconstruction is caused by the nature of conventional image re-

trieval. That is, when starting from an overview image of the scene, the images of the scene details are not retrieved as nearest neighbors, because of low overlap or a large number of similar views outranking the detail images [218]. The reason for not registering retrieved detail images in SFM is, that SFM will often require transition images that establish connectivity between the detail view with high surface resolution and the overview image with comparably low surface resolution in the area of the detail view. In this chapter, we propose a novel combination of a detail oriented retrieval and SFM system to address the challenge of obtaining 3D models from unordered photo collections that provide complete scene coverage and high geometric resolution for the details in the scene.

Our proposed image retrieval method has a major advantage over traditional vocabulary-based and clustering-based approaches. Since the method performs spatial verification through an affine transformation estimation during retrieval, it obtains a quantitative measure of the scene overlap early on. The number of inliers, treated as a similarity score, is a by-product of robust estimation with RANSAC [85].

Next, our geometric verification of image pairs, i.e. the test for a pairwise epipolar geometry, in SFM operates in projective space, estimating an essential or fundamental matrix for moving cameras and a homography for purely rotating cameras. Because the space of affine transformations is a subspace of the space of projective transformations, we can use the existence of an affine transformation between a set of correspondences as a proxy to assume the existence of a projective transformation for a larger set of correspondences between the same image pair. While, in theory, the projective transformations may not exercise all degrees of freedom, the chance of encountering these configurations are extremely low in practice. In fact, for more than 99.9% of image pairs in the experimental datasets (see Section 10.8) we can estimate a valid epipolar geometry or homography when there exists an affine transformation. Hence, if we enforce the existence of an affine transformation, geometric verification in SFM only has to process valid image pairs for reconstruction. This avoids the significant overhead caused by non-overlapping pairs. This makes RANSAC for geometric verification significantly faster, since RANSAC has exponential computational complexity in the number of model parameters and the outlier ratio of the measurements.

Leveraging the early similarity metric, provided by the retrieval system, significantly improves the performance and reliability of our SFM estimation. The reason being that our proposed pipeline can rank the retrieved images by the number of affine transformation inliers and the subsequent geometric verification only spends time on actually overlapping image pairs. This enables us to only match against a limited number of images instead of matching against much larger sets of nearest neighbors as in the case of traditional vocabulary tree based approaches. In all experiments, we attempt to verify a query image to a maximum number of 200 retrieved images, and we empirically found that nearest neighbor images with at least 8 affine transformation inliers to the query image have a very high likelihood of successful registration. Next, we describe the enhancements to our SFM algorithm to reliably obtain reconstructions of the geometric scene details.

10.5 Reconstruction of Details

Detailed scene reconstruction depends on accurate and reliable camera registration, which is especially challenging for the highest-resolution images in a photo collection. There are three major reasons for this: the dependence of incremental SFM on the order of camera registrations, the reduced redundancy of measurements during registration of the detailed views, and the often challenging geometric configurations for these views. In the following, we examine these challenges and describe our solutions to them.

Generally, the quality of SFM results are dependent on three main factors: First, to attain reliable and precise estimates, no parameter should rely on just a minimal or small set of measurements to enable the compensation of measurement noise (detailed views usually have significantly reduced correspondences to other images). Second, reliability provides us with the ability to detect outliers and determines the degree to which undetected outliers affect our estimates (outlier detection for detailed views is challenging due to their reduced redundancy). Third, the uncertainty of measurements propagates to the uncertainty of the estimated parameters, and the viewing geometry impacts the stability of estimation (very oblique or distant views generally have significantly higher measurement and thus registration uncertainties).

First, incremental SFM is heavily dependent on the order of camera registrations, due to the non-linear nature of bundle-adjustment. This effect becomes especially important for detailed scene reconstruction, since the different levels of detail are usually only sparsely connected or connected with forward motion views. Forward motion is a particularly challenging situation for SFM, caused by unstable viewing geometry. In our experience, seeding the reconstruction with one of the detailed views and then incrementally growing the model to include less detailed views fails in most of the cases or results in inferior-quality models. We therefore seed the reconstruction with an image that sees a maximal fraction of the scene, effectively ruling out the detailed and extremely zoomed-out views. From there, we gradually extend the model, avoiding abrupt scale and viewpoint changes by ranking cameras for registration based on the amount of currently visible scene structure.

Second, we discuss the effect of reduced redundancy that is often encountered for the images observing the details of the scene. Since images of detailed structure only see a small fraction of the scene, there are generally much fewer images that observe the same features. In this case, conventional nearest neighbor matching produces fewer and shorter tracks, as it fails to match a significant portion of image pairs from the already small set of images that see the same structure. Hence, bundle-adjustment must deal with a significantly reduced redundancy of the measurements, resulting in less accurate and less reliable camera resectioning and structure estimation. Redundancy, and thus the reliability in bundle-adjustment (or, more generally, in maximum likelihood estimation), is determined as the difference of the number of independent observations minus the effective degrees of freedom [342]. Hence, we are provided with two opportunities for improving redundancy: increasing the number of observations and reducing the degrees of freedom. Inherently, the em-

ployed retrieval system mitigates the effect of reduced redundancy, in that it reveals significantly more overlapping image pairs. In this manner, we are able to build significantly more and significantly longer tracks than before, which leads to significantly increased redundancy of the measurements compared with standard retrieval systems, such as the vocabulary tree based approaches. Additionally, we restrict ourselves to a relatively simple camera model with a total of 8 degrees of freedom (3 orientation, 3 translation, 1 focal length, and 1 first-order polynomial radial distortion parameter with fixed principal point at the image center).

Third, SFM methods face distinct challenges for difficult geometric configurations of the scene and/or the cameras. Many images of the geometric details of the scene are taken at high zoom levels, i.e. with large focal length. In this case, the viewing rays are close to parallel, resulting in high registration uncertainty along the viewing direction. Given that a relatively large displacement along the viewing direction causes only a small change in reprojection error. Hence, we need to have a good initial estimate of the focal length before starting the non-linear refinement in order to achieve convergence to the correct solution. For this purpose, we can use focal length information extracted from EXIF data, if available. However, crowd-sourced photos often lack this information due to modifications, such as resizing, cropping, etc. For large zoom factors, it is not enough to simply assume a default focal length [146, 272], inferred from the image dimensions, and use it as an initial estimate for a non-linear refinement. Rather, it is necessary to exhaustively sample an *a priori* specified space of focal lengths during 2D-3D pose estimation. If EXIF information is missing, we uniformly sample 50 focal lengths for opening angles between $[5°, 130°]$ using P3P RANSAC [269] and use the solution with the highest number of inliers, followed by a non-linear refinement of the pose.

However, even after these modifications, the camera registration occasionally still fails due to low redundancy, bogus EXIF information, or unfortunate configurations. We detect these cases in order to avoid a cascade of mis-registrations due to faulty triangulations from an initially bad camera. These cases can be detected in different stages of the SFM pipeline. First, we detect a small number of inliers during RANSAC pose estimation. Second, a non-linear refinement of an initial registration is performed; faulty registrations typically result in high cost in the non-linear refinement of the pose. Hence, we reject camera registrations that display any of the above properties. Third, bundle-adjustment usually converges to a local minimum for faulty registrations. As a result, it tries to minimize the cost through the use of extreme camera parameters. Whenever we refine the structure and motion in bundle-adjustment, we filter images that have abnormal camera parameters (opening angle outside $[5°, 130°]$, absolute value of radial distortion parameter greater than 1).

Figure 10.6: Single query images used for landmark reconstruction. From left to right: Astronomical Clock, Bridge of Sighs, Terracotta Army, Arc de Triomphe, Notre Dame, Sagrada Familia.

Figure 10.7: Dense reconstruction of Arc de Triomphe details.

10.6 Densification

Our proposed combined SFM and retrieval system achieves significantly more detailed reconstructions than the state-of-the-art reconstruction systems. However, some parts of the scene naturally have low detail. This occurs when the initial set of images, which is obtained by image retrieval without considering the full 3D scene information, does not provide a sufficient number of detailed images, or even none at all, in certain areas of the structure.

However, we wish to produce complete models with high detail across all parts of the structure. To overcome this limitation, we introduce an incremental strategy to extend the initial 3D model by explicitly mining for high detail in low-resolution parts of the originally reconstructed 3D model. To avoid redundant detail mining everywhere, we propose to perform detail mining on demand after the initial 3D reconstruction. After the initial reconstruction we are able to determine the density of the obtained sparse model and identify the low resolution parts of the 3D model. Then, we attempt to densify the reconstruction only for those parts.

To find images that cover the low-resolution parts, we first determine the highest model-resolution of every 3D point by calculating the spatial extent of every image observation in the world coordinate frame, i.e. back-projecting the image pixel to the 3D point into the world coordinate frame. As multiple images from different distances and at different zoom levels potentially see the same structure, the 3D point is assigned the maximum resolution of all of its observations. Given the resolution of the entire structure and the camera poses, we can then identify images that cover the low-resolution parts of the scene. Each image typically only sees a fraction of the entire scene. The median of the observed 3D point resolutions in an image provides us with a meaningful measure of the overall contribution of an image to the surface resolution it contributes, independent of distance to the individual structure or the zoom level. In a final step, we sort all images by their median resolutions and iteratively query for more detail for the top images until no further details are found or a sufficient resolution is achieved. Finally, we connect the new retrievals to the existing reconstruction by only matching the new images using the strategy described in Section 10.4.

10.7 Duplicate Scene Structure

Duplicate, symmetric, or repetitive scene structure is a common pattern in urban environments, posing challenges for incremental SFM due to a potential cascade of camera mis-registrations and faulty triangulations [128, 361, 379, 380]. These camera mis-registrations are caused by symmetric scene structure that is erroneously retrieved and registered by RANSAC based alignment [128].

The existing solutions for the correction of the problem caused by symmetric scene structure is formulated as post-processing of registered camera triplets [379] or post-processing of the entire model [128, 361]. The major drawback of all of

these approaches is, however, that mis-aligned cameras and faulty 3D points could potentially cause unstable models. Moreover, incremental SFM might prematurely stop the extension of the model when there are too many conflicting observations in some parts of the scene. Ideally, such mis-registrations are avoided during the incremental extension of the model.

We found, that, if a more gradual set of transition images is provided (such as in a video), the potential for the confusion caused by symmetric structures is significantly reduced. Given the ability of our retrieval system to crawl for images in different directions, we are able to provide a more gradual sequence of images to the SFM system. Hence, in practice we observed a significantly increased robustness to symmetric structures. For example, Arc de Triomphe, Notre Dame, and St. Vitus consistently produced symmetry issues with traditional retrieval approaches, while our system does not suffer from these effects.

10.8 Experimental Results

In our experiments, we use a generic database of over 7.4 million images downloaded from Flickr through keywords of famous landmarks, cities, countries, and architectural sites. We use single images as seeds for the retrieval (see Table 10.1 and Figure 10.6) and the subsequent reconstruction.

For the retrieval, the maximum number of verifications per query is set to 5000, the average timings for different types of queries are summarized in Table 10.2. Combining the different query types, we can retrieve a set of images for a given query image in the order of minutes. The retrieved collection is a concise set of images (less than 0.1% of the entire database) with a relatively small number of irrelevant images, evidenced by the ratio of registered over retrieved images. Please note, that the registered images are all part of the same connected component as the query image. The efficient matching (see Section 10.4) allows us to build the individual models in a matter of a few hours, since we only need to perform matching on the retrieved image pairs which are a fraction of the possible image pairs.

The individual components of the retrieval system have different effects on the obtained results. The sideways crawl helps to increase the extent of the 3D model and to disambiguate the repeated / symmetric structures. When the reconstruction is executed without sideways crawl (using a state-of-the-art pipeline), Arc de Triomphe, St. Vitus, and Notre Dame have symmetry issues and are not reconstructed as complete – for frontal images the top 100 images are still frontal. Zoom-out provides more context and we have observed that it also helps to disambiguate symmetric structures. For instance, if executed without zoom-out, the sides of Notre Dame's left tower are cross-matched. Zoom-in does not increase the correctness of matching, but it significantly increases the level of detail.

To quantify the amount of detail reconstruction, we determine the spatial resolution of every 3D point as described in Section 10.6. The surface resolution is mapped to jet color map for visualization, with red referring to the highest and blue to the

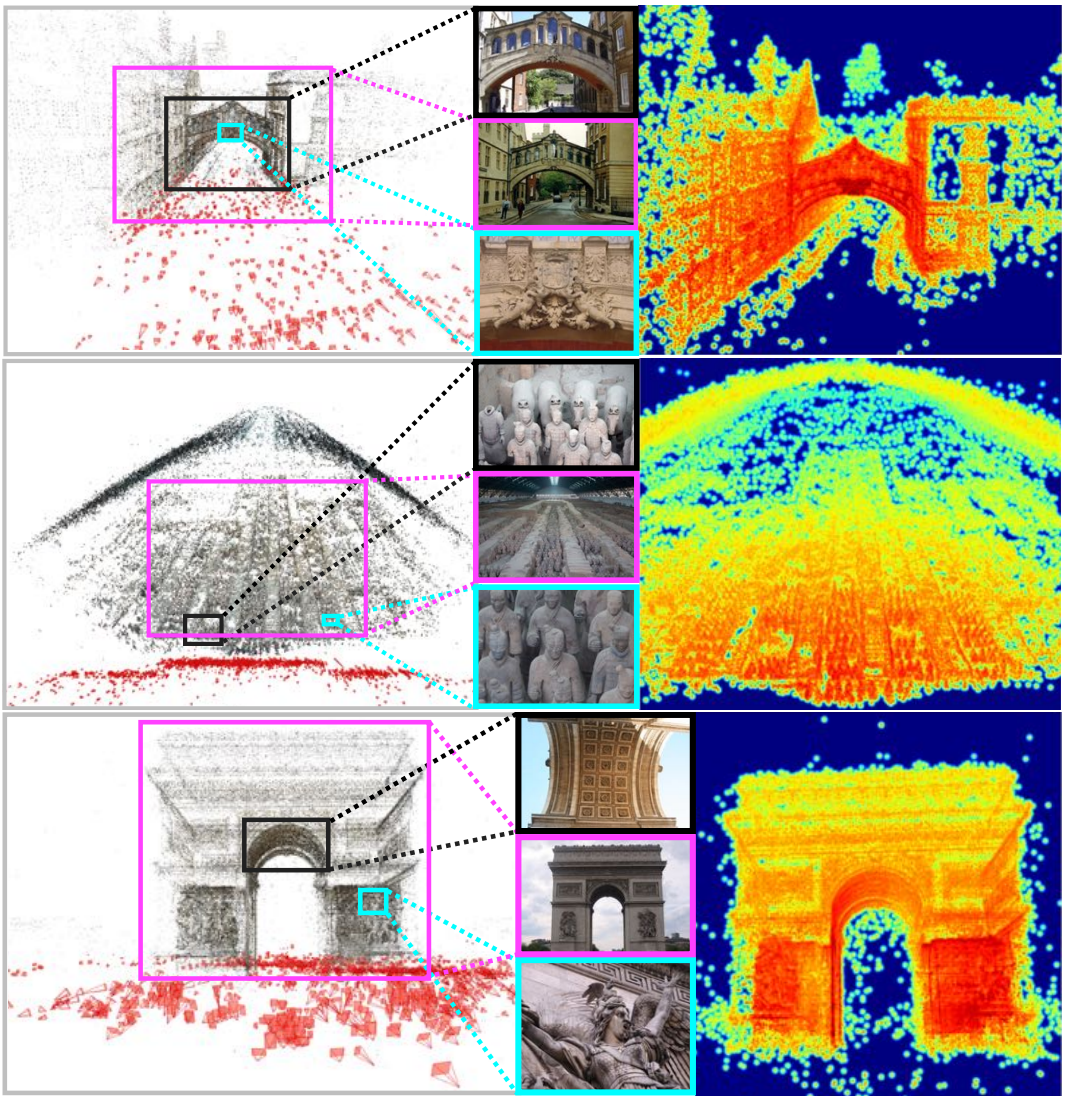

Figure 10.8: Reconstructions from top to bottom: Bridge of Sighs, UK; Terracotta army, China; Arc de Triomphe, France. Left: 3D model obtained from our retrieval and reconstruction system. Middle: registered images illustrating the range of views from overview images to images of a specific architectural detail. Right: visualization of the surface resolution from high resolution in red (approximately 1mm surface resolution) to low resolution in blue. More results in Figure 10.9.

	Retrieved	Registered	Pairs	Points
Astronomical Clock	10,443	8,163	572,412	830,238
Bridge of Sighs	2,077	1,018	70,473	117,182
Terracotta Army	2,781	2,099	113,747	167,715
Arc de Triomphe	3,744	2,640	179,346	395,431
Notre Dame	4,978	2,081	164,871	304,339
Sagrada Familia	11,783	7,129	617,362	364,510

Table 10.1: Details of reconstructed models with the number of retrieved and registered images, the number of verified image pairs, and the number of reconstructed 3D points.

Query type	Time
NN query, no QE	1 sec
NN query, with QE	5 sec
Multi scale-band crawl	2.8 min
Sideways crawl	5.6 sec

Table 10.2: Average query duration for the retrieval engine.

lowest resolution. Figure 10.2, Figure 10.8, and Figure 10.9 show the resolutions for a variety of scenes. Moreover, Figure 10.7 is an example of dense reconstruction using multi-view-stereo.

Another experiment shows that the choice of the query image is not critical. Seeding the Arc de Triomphe scene with two different images from opposing sides of the building results in models with 2640 and 2721 images (intersection over union 92%), which are visually near-identical.

To compare our system against full pairwise reconstruction, we injected the Dubrovnik6k dataset [193] with 6,036 images into the 7.4M image database. Starting from a single query image, our pipeline reconstructs 87% (4430 images) in the first 3 retrieval-SFM iterations, with respect to to full pairwise reconstruction on the isolated dataset (5102 registered images). Both approaches result in similar visual quality, but faster runtime for our pipeline, even though it operates on 7.4M images (compared to 6K for the pairwise approach).

10.9 Summary

In this chapter, we proposed a novel tightly coupled image retrieval and reconstruction system in order to produce detailed 3D models from unstructured image collections. Our method is able to seed the reconstruction from just a single image. The tight integration of reconstruction and retrieval enables us to retrieve

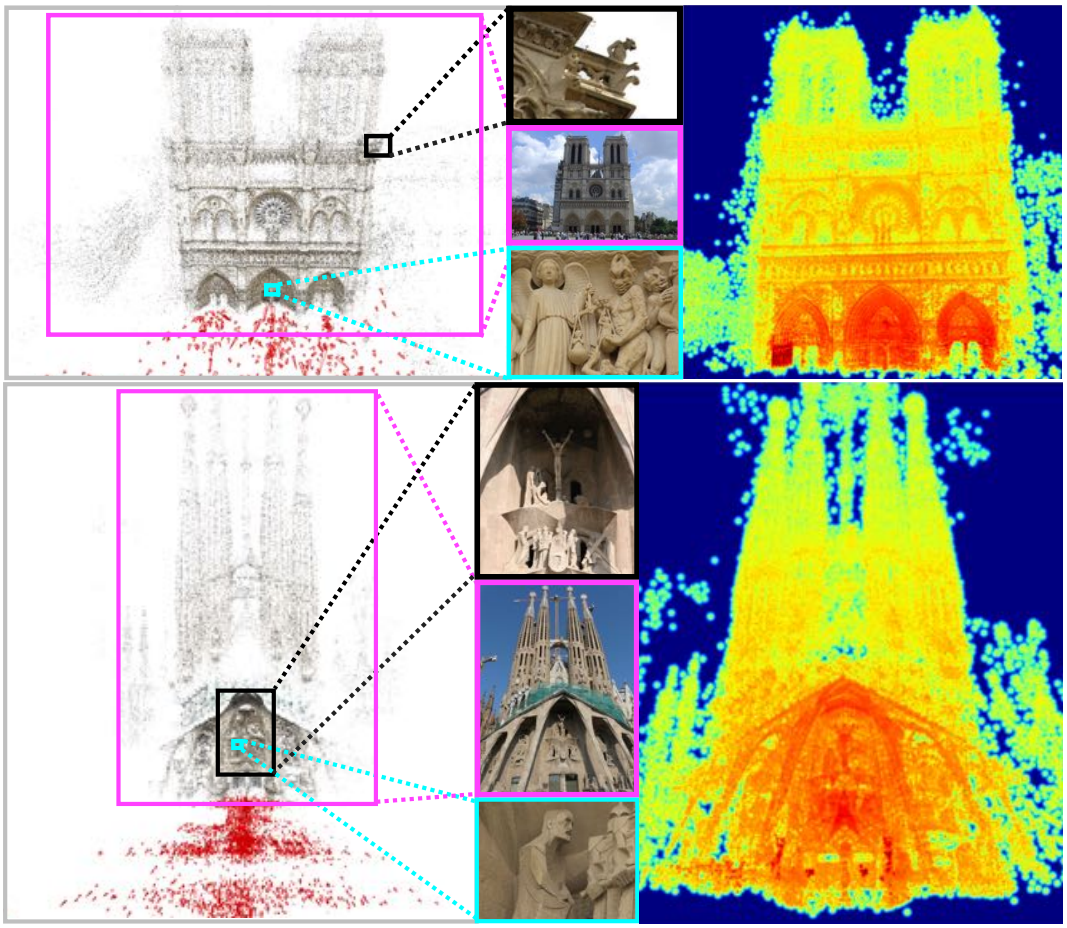

Figure 10.9: Reconstructions from top to bottom: Notre Dame, France; Sagrada Familia, Spain. Left: 3D model obtained from our retrieval and reconstruction system. Middle: registered images illustrating the range of views from overview images to images of a specific architectural detail. Right: visualization of the surface resolution from high resolution in red (approximately 1mm surface resolution) to low resolution in blue. More results in Figure 10.9.

image data suitable for reconstruction that the current state-of-the-art systems in 3D reconstruction from unstructured photo collections do not recover. Experiments demonstrated our method on a large variety of scenes from a collection of 7.4 million unstructured images downloaded from the Internet.

11 Illumination Robust 3D Modeling

Image retrieval and 3D reconstruction have made big strides in the past decade. In the previous chapter, we presented a system that combined image retrieval and Structure-from-Motion to achieve accurate and detailed 3D modeling from millions of unstructured images. Combining them cannot only tackle scale but also allows to reconstruct spatially complete models with high levels of detail [288]. A key observation is that an increasing number of images in the collections ease the registration of images taken under very different illumination conditions into a single 3D model. A feat that is not achieved by direct matching techniques, but rather by discovering sequences of matching images with a gradual change of the illumination, see Figure 11.2 for such a transition sequence.

A sparse 3D reconstruction of feature points, obtained from a mixed set of day and night images, is reliable and naturally occurs in large-scale photo collections. This is due to the presence of "transition" images and due to the fact, that some of the detected features after photometric normalization provide sufficiently stable matches across illumination transitions. Examples of such feature points, the corresponding image patches, and their normalized descriptor patches are shown in Figure 11.4.

However, while beneficial for SFM, the registration of mixed illumination images creates challenges for dense 3D reconstruction, which delivers poor results or even

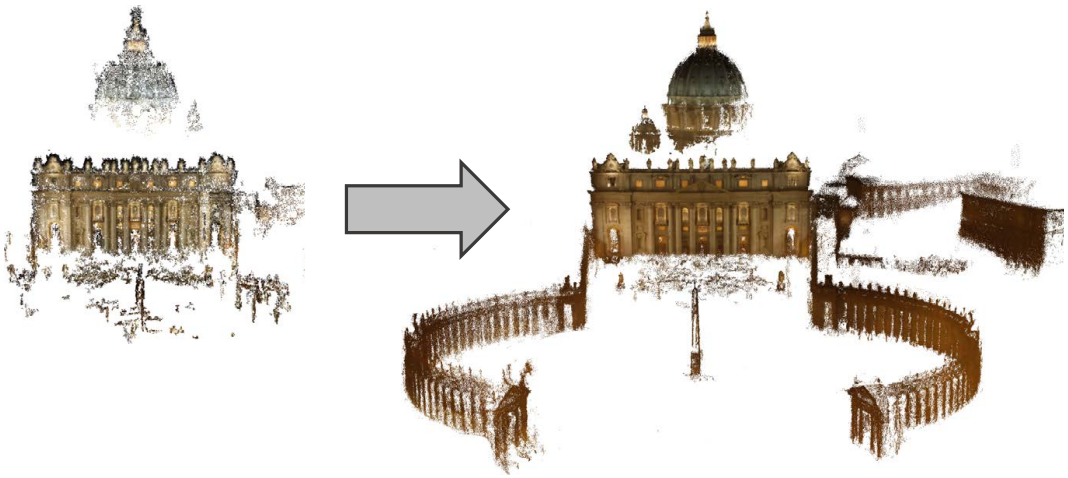

Figure 11.1: Night model of St. Peter's Cathedral in Rome reconstructed by our method. Left: Model obtained from night images only. Right: Fused, recolored model from day and night images.

Figure 11.2: Tyn Church, Prague. Registration of day and night images into the same model through smoothly varying illumination in intermediate images during dusk and dawn.

fails in the presence of day and night images [205, 206]. In particular, mixed illuminations cause erroneous dense correspondences due to accidental photo-consistency in multi-view stereo that distort the texture composition of the models.

As a *first* contribution of this chapter, we propose a method for automatically separating day and night images based on the sparse scene graph produced by SFM and a learned day/night color model. The separated sets of day and night images then allow to compute reliable dense reconstructions for each of the two modalities separately.

While two separate models on first sight may be seen as a drawback, we demonstrate that they often contain regions in which only one of the models provides reliable surface reconstruction. As expected, we observe that usually daytime images are significantly more frequent and, due to better illumination conditions, lead to overall superior models over nighttime models. Interestingly, we observed several situations where night images provide better reconstruction than their daytime counterparts: (i) when lights at night illuminate or texture areas that are shadowed or ambiguous during the day, and (ii) when areas with repeated and confusing textures are not lit during the night, allowing unambiguous dense matching in those areas. Our *second* contribution is to fuse the initially separated dense models into a superior model combining the strengths of both modalities.

Finally, as a *third* contribution, we introduce a method of color transfer to consistently re-color the composite 3D areas for each illumination condition, even for areas that were not reconstructed under the illumination, i.e., we will compute a nighttime color even for geometry that is only reconstructed in the day model.

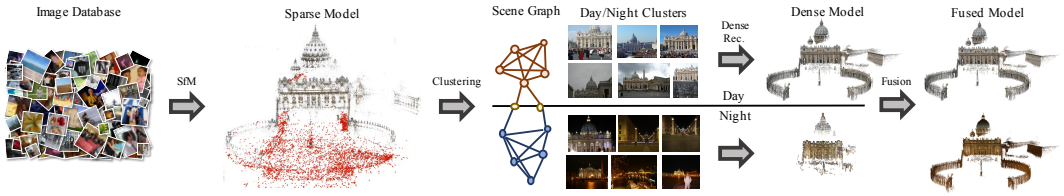

Figure 11.3: The proposed day/night modeling pipeline starting with sparse modeling to day-night clustering and the final dense modeling.

In summary, our contributions achieve a more complete and accurate dense 3D reconstruction for mixed day- and nighttime images that are typically present in Internet photo collections. Previously, the joint modeling of day and nighttime images caused disturbing artifacts or even lead to reconstruction failures. Additionally, we are able to reconstruct a complete color representation for the dense model surfaces leveraging the corresponding appearance characteristics of the daytime and nighttime images.

11.1 Related Work

The seminal paper of Snavely et al. [308, 310] first proposed reconstruction from unordered Internet photo collections. To determine overlapping views, Snavely et al. performed exhaustive pairwise geometric verification. While this ensures the highest possible discovery rate, it impairs the scalability of their system due to the quadratic complexity growth in the number of images. During the following years, several methods for tackling scalability of unordered photo collection reconstruction were proposed: appearance-based clustering methods for grouping the images [89, 191], vocabulary tree based approaches [4, 198], and most recently streaming based methods leveraging augmented appearance indexing [129]. Although the systems successfully scaled the reconstruction to tens of millions of images, they lost the ability to reconstruct details of the scene in the process. In the previous chapter, we proposed a method to overcome this limitation of not being able to reconstruct details. Their method leverages a tightly-coupled SFM and image retrieval system [220] to overcome the loss of fine details in the models while keeping the scalability of the state-of-the-art reconstruction systems. Our reconstruction system is inspired by this method. Snavely et al. [308, 310] empirically observed the difficulty in registering night images due to their noisiness and darkness. In our system, we overcome this limitation by registering night images mainly through transition images under intermediate illumination conditions during dusk and dawn (see Figure 11.2). Snavely's system [309] provided an option to manually select day or night images to explore similar viewpoints and illuminations. In contrast, our system automatically classifies and clusters day and night images. In addition, we use the clustering to improve reconstruction results.

Schindler and Dellaert [280] proposed a method for analyzing the point in time

at which a photo was taken. In contrast to our approach, their method was relying on observable changes of the scene geometry, e.g., construction or demolition of buildings, which typically happens over longer periods of time. Our method focuses on modeling the illumination changes over the course of a day. Recently, Matzen et al. [209] proposed an approach to model and extract temporal scene appearance changes in 3D reconstructions. They perform temporal segmentation of the 3D model to obtain objects whose appearance changed over time. The recovered object appearance changes (wall art, signs, billboards, storefronts, etc.) relate to scene texture changes but not to illumination changes due to their search of change over longer periods of time. In contrast, our algorithm aims at determining periodic short term (over the course of a day) temporal scene appearance and illumination changes. Hence, our proposed approach deals with much smaller appearance differences in segmented parts of the reconstruction. These changes are caused by different illuminations during daytime and nighttime and are not correlated with scene texture changes.

Martin-Brualla et al. [206] proposed to compute time-lapse mosaics from unordered Internet photo collections of landmarks. They observed the difficulties posed by the presence of night and day images in the same reconstruction. Specifically, they noted that mixing day and night images within the same model introduces "unrealistic twilight effects". In this chapter, we propose an approach that overcomes these failure cases and obtains a correct representation of the 3D model for both modes of illumination.

Ji et al. [155] proposed a system to automatically create illumination mosaics for a given outdoor scene from Internet photos. Their work strives to depict temporal variability of the observed scene by presenting a 2D image of the scene with varying illumination along the rows of the image. They perform a search for a chain of images that exercise smooth illumination variation and that are all related through a homography mapping. In contrast, our method considers all available images and not only the images related through homographies. Instead of illumination modeling in 2D, our approach achieves illumination separation and modeling in 3D for the entire scene. Moreover, the ordering of Ji et al. [155] heavily relies on the color of the sky shown in the images. Whereas our system can perform day-night separation even with no sky present in any of the images.

Veride et al. [351] learned a feature detector which is stable under significant illumination changes, facilitating the matching between day- and nighttime images. They observed that standard feature detectors exhibit significant temporal sensitivity, i.e., reduced repeatability under different illumination conditions. We exploit this temporal sensitivity "flaw" of the standard detectors to efficiently split a given 3D model into groups of cameras and points that have the highest illumination change across groups, i.e., a group for the day and another for the night.

11.2 Overview

Before delving into the details of our method for day and night model reconstruction, we provide an overview as illustrated in Figure 11.3. It starts with a database of unordered images. During the initial phase of the reconstruction, we build a sparse 3D model using SFM (see Section 11.3). In support of sparse modeling, we index all images in the database using a min-Hash and find reconstruction seeds by leveraging geometrically verified hash-collisions. Next, our SFM algorithm uses these seeds to build sparse 3D models for the scenes contained in the photo collection. Specifically, it uses a feedback loop to gradually extend the reconstruction by dedicated queries against the database. The resulting sparse model contains day and night images registered into the same model and represented as one scene graph.

In the next step, a dense scene model is obtained. Given the previously observed difficulties and artifacts caused by mixed day and night images, we deviate from the standard approach of directly proceeding to dense reconstruction. We first split the scene graph into day and night clusters to separate the images of the different illumination conditions (see Section 11.4). This in essence separates the scene graph into two scene graphs – one for daytime images and one for nighttime images. Then, we perform separate dense geometry estimation for the images in each of the scene graphs yielding two separate dense 3D models (see Section 11.5). Subsequently, the two dense models are aligned into one common model representing the overall dense scene geometry. As part of computing the dense scene geometry, we obtain the color information of the point cloud under the two illumination conditions, i.e., a daytime color and a nighttime color for each point. Given that not all parts of the common model are necessarily visible both at daytime and at nighttime, we then determine the missing color information through cross-illumination transfer. Specifically, we use one illumination condition to find similar patches with a corresponding color in the other illumination. The color information of the patches under one illumination is then used to compose the missing color information for the point under the other illumination.

11.3 Reconstruction

In this section, we detail our approach for efficiently reconstructing all 3D models contained in a given image database. We use the same generic database as used in the previous chapter with over 7.4 million images downloaded from Flickr through keywords of famous landmarks, cities, countries, and architectural sites. The approach starts with an initial clustering procedure to find putative spatially related images. These spatially related images are subsequently used to seed an iterative reconstruction process that repeatedly extends the 3D model through a tight integration of the image retrieval and SFM module similar to the approach presented in the previous chapter. In contrast to the previous system, our approach exhaustively builds models for the entire image database. Due to the massive number of images

in the database, exhaustive reconstruction imposes several challenges in terms of efficiency, which we address through an initial clustering procedure and a parallelized implementation.

11.3.1 Clustering

To seed our iterative reconstruction process efficiently, we find independent sets of spatially overlapping images using the clustering approach by Chum et al. [67]. This approach first indexes all database images in a min-Hash table and then uses spatially verified hash collisions as cluster seeds. Next, an incremental query expansion [66, 243] with spatial verification extends the initial clusters with additional images of the same landmark. The nearest-neighbor images in this query expansion step then define the graph of overlapping images, the so-called scene graph. Given that query expansion is a depth first search strategy, the resulting scene graph is only sparsely connected. However, in order to achieve a successful reconstruction, SFM requires a denser scene graph than provided by the clustering method. Therefore, we first densify the scene graph as described in the following section before using it in SFM. Compared to the approach in [288], which takes a single query image as input for the reconstruction, this clustering step reduces the number of query images dramatically. Rather than seeding the reconstruction with 7.4M query images, the clustering procedure identifies 19,546 individual landmarks used to initialize the subsequent reconstruction procedure and thereby reduces the number of seeds by 3 orders of magnitude.

11.3.2 Densification

Next, we densify the initially sparse scene graph for improved reconstruction robustness and completeness. In the spirit of the previous chapter, we leverage the spatially verified image pairs and their visual word matches along with an affine model to serve as hypotheses for subsequent exhaustive feature matching and epipolar verification. From this exhaustive verification, we not only obtain a higher number of feature correspondences but we also determine additional image pairs to densify the scene graph. More importantly, beyond the benefit of additional image pairs, the significantly increased number of feature correspondences is essential for establishing feature tracks from day to night images through dusk and dawn. Only through these transitive connections, we are able to reliably register day and night images into a single 3D model.

11.3.3 Structure-from-Motion

The densified scene graph is the input to the subsequent incremental SFM algorithm, which treats each edge in the graph as a putative image pair for reconstruction and attempts to reconstruct every connected component within a cluster. Connected components with less than 20 registered images are discarded for the purposes of

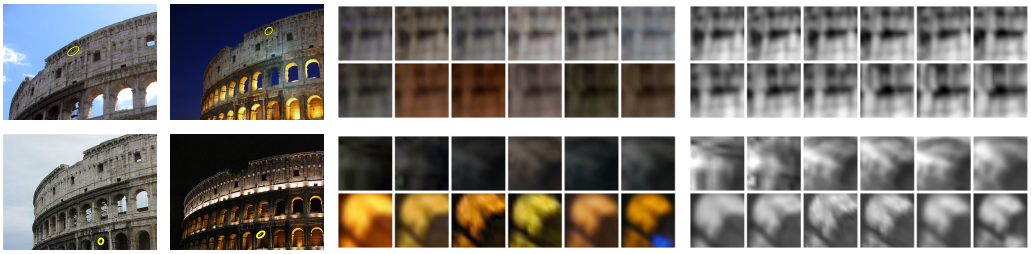

Figure 11.4: Colosseum, Rome. Two feature tracks containing both day and night images/features. Each row depicts two images labeled as day and night, respectively, followed by a subset of feature patches depicted in two rows, one for day and one for night features, respectively. Intensity normalized patches, grayscale versions used for SIFT description, are shown to the right of the respective color patches. Notice the variation in lighting conditions for day and night, expressed as a significant color difference of patches. Best viewed in color.

day/night modeling as they typically lack a sufficient number of transition images during dusk and dawn.

11.3.4 Extension

To boost registration completeness, a final extension step issues further queries for all registered images in each reconstructed connected component. If new images are found and spatially verified, we again perform scene graph densification and use SFM to register the new views into the previously reconstructed models. While significantly increasing the size of the reconstructed models, the extension process also improves the performance of the day/night modeling step. Typically, the initial set of images obtained in clustering often only contains images from one modality, i.e., either day or night, even though our large-scale image database contains images of both modalities for almost all landmarks. The iterative extension overcomes this problem by incrementally growing the model from day to night or vice versa through transition images during dusk and dawn (see Figure 11.2 for an example).

11.4 Day and Night Clustering

After the exhaustive 3D reconstruction stage of all landmarks in the database, we proceed with clustering the images inside each of the 3D models into two groups: day- and nighttime. For crowd-sourced data, the clustering cannot simply rely on embedded EXIF time stamps. In our experiments, the majority of images either have no time stamp information at all or the information is clearly corrupt. We speculate that most images are taken on vacation and people do not adjust the time zone in their cameras. For most landmarks with many registered images, day- and

nighttime images are registered into the same model as a result of the extension step (seeSection 11.3). It is well known that standard feature (keypoint) detectors [351] suffer under illumination sensitivity, i.e., the reliability of keypoint detectors degrades significantly when the images originate from outdoor scenes during different times of the day or generally different illumination conditions. In this case, the detectors commonly produce keypoints at different locations for day and night lighting conditions [351]. This even holds true when the images are taken from the same viewpoint. Our key insight is to exploit this behavior in order to split the images inside a SFM model into two groups. Our clustering is based on the number of commonly observed 3D points for each pair of images with similar viewpoints. This enables us to identify day and night images registered within a model. For efficient grouping, we leverage a bipartite visibility graph [193], as explained in the following sections.

11.4.1 Min-cut on Bipartite Visibility Graph

A 3D model produced by SFM can be interpreted as a bipartite visibility graph $\boldsymbol{\mathcal{G}} = (\mathcal{I} \cup \mathcal{P}, \mathcal{E})$ [193], where the images $i \in \mathcal{I}$ and the points $p \in \mathcal{P}$ are the vertices of the graph. The edges of the graph are then defined by the visibility relations between cameras and points, i.e., if a point p is visible in an image i, then there exists an edge $(i, p) \in \mathcal{E}$. We define the set of points observed by an image i as:

$$\mathcal{P}(i) = \{p \in \mathcal{P} \mid (i, p) \in \mathcal{E}\}. \tag{11.1}$$

Our day/night clustering separates the vertices of the graph (the cameras and points) into two groups: one corresponding to day cameras and points and the second for the night cameras and points. More formally, we define two label vectors representing the group assignment. Vector $\boldsymbol{\alpha_i}$ for the images and vector $\boldsymbol{\alpha_p}$ for the points:

$$\begin{aligned} \boldsymbol{\alpha_i} &= \{\alpha_i \in \{0, 1\} \mid i \in \mathcal{I}\}, \\ \boldsymbol{\alpha_p} &= \{\alpha_p \in \{0, 1\} \mid p \in \mathcal{P}\}, \end{aligned} \tag{11.2}$$

where label variables α_i and α_p correspond to image i and point p, and label $\alpha_i, \alpha_p = 0$ denotes day and label $\alpha_i, \alpha_p = 1$ night. We formulate the problem of separating day from night images as an energy optimization. We propose the following energy function $\mathbf{E}$ over the graph $\boldsymbol{\mathcal{G}}$ that measures the quality of the labeling $\boldsymbol{\alpha_i}$, $\boldsymbol{\alpha_p}$:

$$\mathbf{E}(\boldsymbol{\alpha_i}, \boldsymbol{\alpha_p}, \boldsymbol{\mathcal{G}}) = \sum_{i \in \mathcal{I}} U_i(\alpha_i) + \sum_{(i,p) \in \mathcal{E}} P_{i,p}(\alpha_i, \alpha_p). \tag{11.3}$$

The term $P_{i,p}(\alpha_i, \alpha_p)$ describes the pairwise potentials associated with the edges enforcing a smooth labeling of the cameras and points with respect to their mutually observed scene information. A standard Potts model is used for the pairwise potentials, that is $P_{i,p}(\alpha_i, \alpha_p) = 0$ for $\alpha_i = \alpha_p$ and $P_{i,p}(\alpha_i, \alpha_p) = 1$ otherwise. The 3D

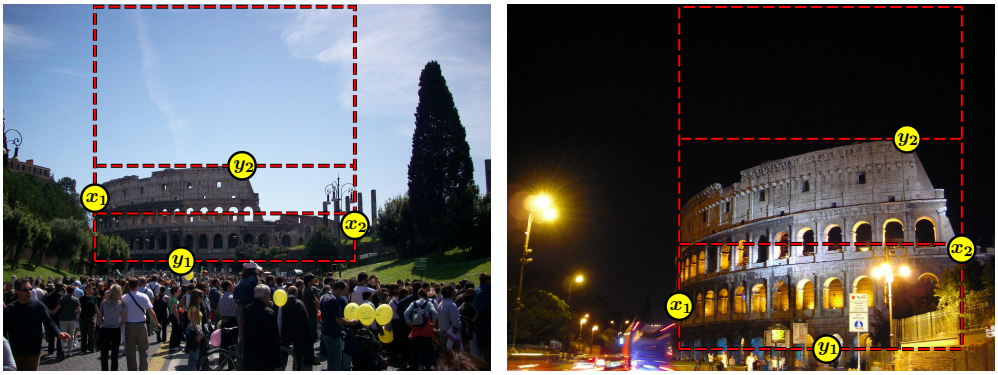

Figure 11.5: Colosseum, Rome. Examples of image color histogram description area. Coordinates of features reconstructed as 3D points define the bounding boxes used to compute three histograms. Using the 3D model information, we successfully segment out confusing background and are able to focus the description on the three important parts: sky, upper and lower part of the reconstructed landmark.

points incur no unary cost for being assigned either label. The unary cost $U_i(\alpha_i)$ for images is based on the day/night illumination model discussed below. The clustering of all images and points in a model is achieved by minimizing the objective

$$\boldsymbol{\alpha_i}, \boldsymbol{\alpha_p} = \arg\min_{\boldsymbol{\alpha_i}, \boldsymbol{\alpha_p}} \mathbf{E}(\boldsymbol{\alpha_i}, \boldsymbol{\alpha_p}, \boldsymbol{\mathcal{G}}) \tag{11.4}$$

using the min-cut/max-flow algorithm of Boykov et al. [37]. Figure 11.4 shows examples of 3D point tracks that contain both day and night labels.

11.4.2 Day and Night Illumination Model

We use a day/night illumination model to estimate the likelihood of an image being taken during day or night respectively. As a feature for the prediction, a spatial color histogram in the opponent color space [107]

$$\begin{aligned} I &= (R+G+B)/3, \\ O_1 &= (R+G-2B)/4+0.5, \\ O_2 &= (R-2G+B)/4+0.5, \end{aligned} \tag{11.5}$$

is used. To reduce the influence of occlusions and background clutter, a three-band spatial histogram is computed over a region of the image directly related to the reconstructed object, as depicted in Figure 11.5. The bottom two stripes of the histogram equally split the bounding box of feature points that have been reconstructed as 3D points in the model. The top band covers the sky area above the landmark, up to the top edge of the image.

The color is uniformly quantized and each spatial band of the histogram is separately normalized by the number of pixels per region. The final illumination descriptor is obtained by concatenating the color histograms for the three spatial bands. In our experiments, we use $n = 4$ bins per color channel resulting in an image descriptor of dimensionality $D = 3n^3 = 192$.

To classify the illumination descriptors into daytime and nighttime, a linear SVM [32] is trained on ground-truth labeled images of our largest model (Colosseum, Rome). The same trained SVM is used to compute the unary terms for each image i in all reconstructed models:

$$U_i(\alpha_i) = \begin{cases} 0 & \text{if } \alpha_i = \text{SVMp}(i), \\ c{\cdot}\text{SVMs}(i){\cdot}\,|\mathcal{P}(i)| & \text{otherwise,} \end{cases} \tag{11.6}$$

where SVMp(i) and SVMs(i) denote the SVM's label prediction and the absolute value of the prediction score of image i, respectively. The confidence constant c of the trained SVM has higher confidence for higher values $c > 0$ and in our experiments we set $c = 1$. The cardinality of the set of observed points $\mathcal{P}(i)$ is equal to the number of edges that connect image i to 3D points in the visibility graph.

The label of image i is decided based on the labels of its observed points (pairwise term) and by the confidence of the linear SVM prediction (unary term). In order for this process to be fair for all images, we multiply the SVM score by the number of observed points $|\mathcal{P}(i)|$ for the final unary term. This number defines the percentage of observed points that should have different labels to change the SVM prediction for the image.

11.5 Day and Night Modeling

After obtaining the image clustering, we first aim to reconstruct the separate models and then combine them into a joint model to produce consistent geometry and texture within each modality, as detailed in this section. Typically, there is an uneven distribution of day and night images, causing one of the modalities to have lower scene coverage. In addition, the different illumination conditions during day and night allow for reconstruction of details that are clearly visible during the day but not at night and vice versa. For example, many landmarks are lit during the night and a reconstruction of fine details is oftentimes possible for night images while during the day those structures are hidden in shadows. Hence, in the second step, we fuse the geometry of the two models in order to obtain better completeness in terms of scene coverage and reconstruction of fine details. To obtain consistent color for the fused model, we re-color the structure of the respective other modality through repainting of visible structure and inpainting of structures not covered by images. The following sections describe our proposed approach in detail.

11.5.1 Dense Reconstruction

For dense reconstruction, we first separate the sparse model into its day and night modalities based on the labels $\boldsymbol{\alpha_i}$ and $\boldsymbol{\alpha_p}$. For most models, there are enough images during day and night to allow for dense reconstruction in both modalities. We split the graph $\boldsymbol{\mathcal{G}}$ into two disjoint sub-graphs: $\boldsymbol{\mathcal{G}_d}$ for the day modality, and $\boldsymbol{\mathcal{G}_n}$ for the night modality. We separate the tracks of points that are visible in both day and night images. The two graphs serve as the input to the dense reconstruction system[1] by Furukawa and Ponce [93, 95]. Separate reconstruction of day and night images removes many of the disturbing artifacts present when using all images in a model (see Figure 11.6). To mitigate reconstruction artifacts caused by sky regions, we create segmentation masks using an improved version of the approach proposed by Ji et al. [155]. In distinction to their approach, we leverage the sparse point cloud as an additional clue for deciding whether parts of the image belong to the sky or not. The outputs of this step are separate models for day and night. In the next section, we describe an approach that fuses the two models and leverages the benefits of the respective other modality for increased model completeness and detail reconstruction.

11.5.2 Fusion

Typically, the scene coverage of day and night models are very different due to a multitude of reasons. First, parts of the scene may not be covered by any images in one of the modalities, e.g., caused by occlusion or lack of images. In addition, we found that for some scenes, parts of the reconstruction are not covered by images at all during the night due to restricted access in those areas, e.g., the inside of the Colosseum. A second reason for different scene coverage is the different illumination conditions causing dynamic range issues for the cameras that often prevent reliable reconstruction of scene parts, even though they are theoretically visible. Especially for night images, parts are often under-illuminated or lack any illumination at all. There is a similar issue for day images as well, e.g., shadows caused by intense sunlight often prevent reconstruction of structure. One such case is depicted inFigure 11.6. Using the default parameters, the dense reconstruction method by Furukawa and Ponce [95] is very conservative in terms of creating dense points, i.e., 3D structure only appears in high confidence areas. Therefore, geometric fusion of the two models enables the use of structure that is more accurately reconstructed from day or night images. As a first step, we perform alignment of the two models into the same reference frame using the correspondences from points that appear both in night and day images. However, such fused models contain both day and night points and thus suffer from inconsistent coloring. In the following section, we describe a joint repainting and inpainting procedure to color the fused day points in the night model and vice versa (see Figure 11.7). For simplicity, we explain the

[1] Note that at the time, we published the work in this chapter, the dense reconstruction system presented in Chapter 9 was not yet published.

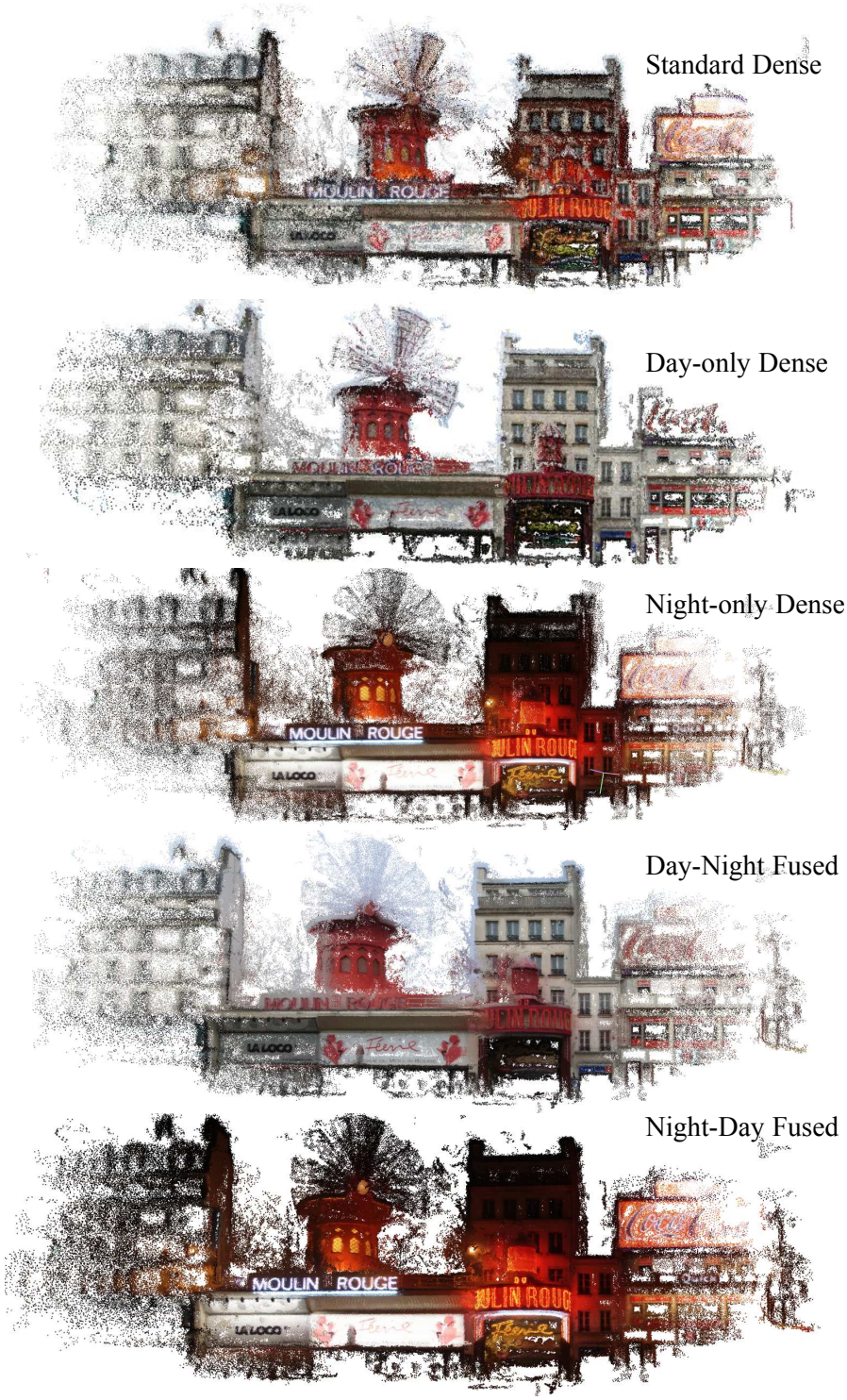

Figure 11.6: Moulin Rouge, Paris. Standard dense modeling using day and night images creates disturbing artifacts, while a separate modeling for day and night images produces consistent geometry and coloring. Fusion and recoloring improves completeness, appearance, and accuracy.

procedure for the case of coloring the fused point cloud using the night images, but the approach is analogous in the opposite direction.

Repainting

As explained in the previous section, many dense points are reconstructed in day but not in night models, even though they are covered by night images. We project these points into all night images and determine their color as the median of all projections. For occlusion handling, we enforce depth consistency with the sparse point cloud. The depth of the dense points must be within the 10th and 90th percentile of the depth range of the observed sparse points of an image. While this cannot account for fine-grained occlusions, in our experiments, the extracted colors are not affected by occluded observations due to the robust averaging of colors.

Inpainting

For those points that are not visible in any night image, we propose a novel inpainting method. The method learns the appearance mapping between known corresponding day and night patches to predict the color of unseen points. To establish dense correspondence between day and night patches, we first project all points into day and night images. Any point that projects both into day and night images defines a correspondence that we use to infer the appearance of a day point during the night. Each of the correspondences usually projects into multiple day and night images. An average color histogram is extracted from a 5×5 patch around the projected image location, for each correspondence between day and night images. While we tried to incorporate shape information as descriptors, we found color histograms to be sufficiently distinctive features and best performing for the task of inpainting. Using these histograms as input, we train a nearest-neighbor regressor to map from day patches to night patches. To inpaint the color of points that only project to day images, we extract the average day color histogram for that point and use our trained regressor to predict its most likely appearance during the night. This inpainting method enables us to obtain a model during the night that is as complete as during the day. In all our experiments, we use $N = 20$ nearest neighbors for the regression and $D = 96$ dimensional histograms for the appearance descriptor.

Blending

Even though we are using a robust average in the repainting step, low-coverage points sometimes suffer from abrupt changes in appearance in 3D space whenever the field of view of one image ends. To counteract this artifact, we propose to blend these points by predicting their appearance using the same mapping as in the inpainting step. We improve the color of any point with a track length $t < t_{min}$. The originally repainted color is then blended with the inpainted color based on the

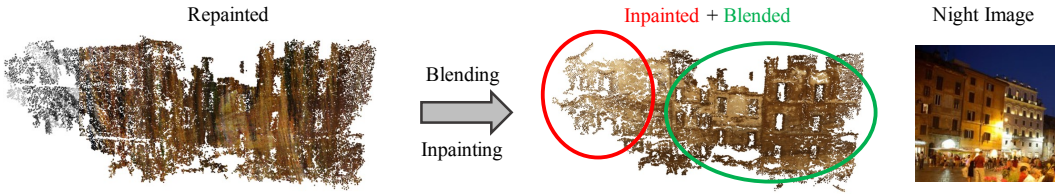

Figure 11.7: Pantheon, Rome. Example of repainting, inpainting, and blending for building facade that is not present in the original night reconstruction.

track length of the point. The blended color of a point is calculated as

$$\mathbf{c}_{bl} = \frac{t_{min} - t}{t_{min}} \cdot \mathbf{c}_{inp} + \frac{t}{t_{min}} \cdot \mathbf{c}_{rep}, \tag{11.7}$$

where $\mathbf{c}_{inp}$ and $\mathbf{c}_{rep}$ denote the inpainted and repainted colors, respectively. In all experiments we set $t_{min} = 10$.

11.6 Results

After describing our novel approach for day/night modeling, we now evaluate our method on the entire 7.4M image database and present results for a variety of scenes. Our experiments demonstrate that the proposed algorithm robustly generalizes to different illumination conditions.

11.6.1 Reconstruction

The iterative reconstruction process for the database of 7.4 million images converges in 3 iterations for all clusters in the database and takes around one week on a single desktop machine. We produce day and nighttime models for any reconstructed cluster that has a sufficient number of registered images, i.e., at least 30 day and 30 night images. We find 1,474 such models out of the initial set of 19,546 clusters used to seed the reconstruction pipeline. These models have 239,717 unique, registered images contained in 845 disjoint landmarks. The average ratio of day to nighttime images in the reconstructions is 9:1.

11.6.2 Clustering

To evaluate our clustering approach, we hand-labeled 13,931 images of 6 different landmarks present in the dataset using the two classes of labels “day” and “night” (see Table 11.1). For the sake of comparison, we also introduce a baseline method for image clustering into day- and nighttime images using k-means clustering with two clusters on the HSV color histograms of the images. Our clustering approach achieves almost perfect classification for the day and night images. Even in the challenging case with only few night images. We outperform k-means on all landmarks

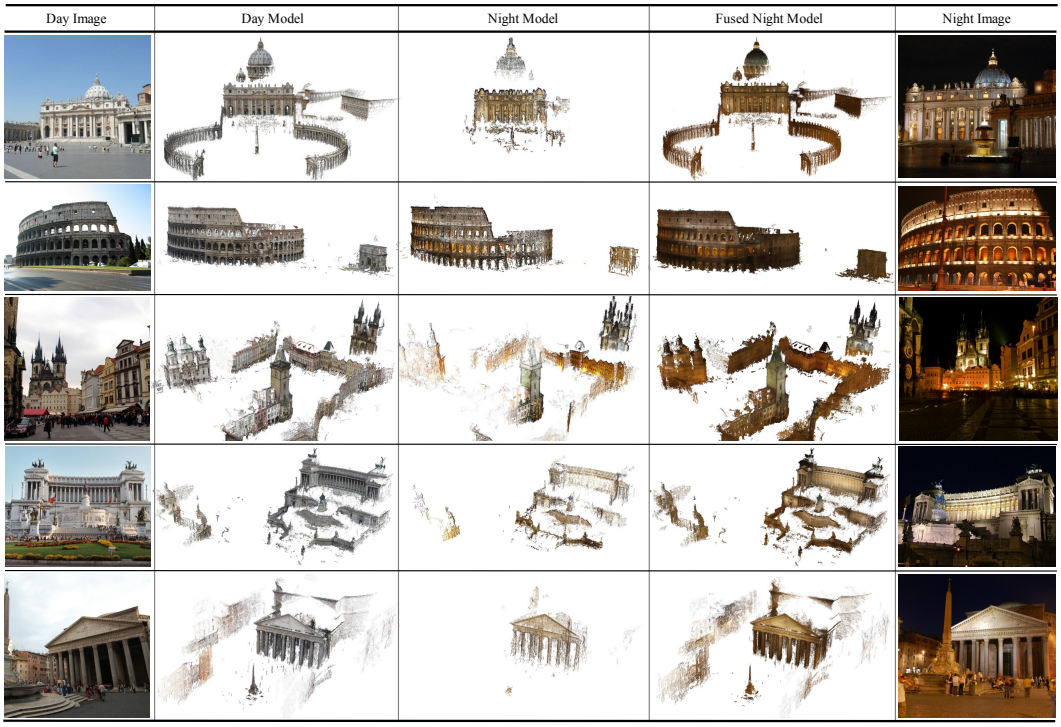

Figure 11.8: Example of reconstructions produced by our method for St. Peter's Basilica in Vatican, Colosseum in Rome, Astronomical Clock in Prague, Altare della Patria in Rome, and Pantheon in Rome.

and, most importantly, we can classify night images very accurately, which is crucial for avoiding artifacts in day/night modeling. This is even more notable considering that night images are significantly outnumbered in most of the models.

11.6.3 Geometric Fusion

Figure 11.8 impressively demonstrates the improved completeness and accuracy of night models by the geometric fusion. In addition, Figure 11.6 also depicts an example of the opposite direction, where the structure of day model is improved through the night model. We encourage the readers to view the supplementary material for additional impressions and videos.

11.6.4 Color Fusion

Figure 11.7 demonstrates the proposed repainting, inpainting, and blending method applied to a building facade in a low-coverage part of the Pantheon reconstruction. The structure is not reconstructed in the original night model (see Figure 11.8). Hence, the entire structure consists of repainted points from the day reconstruction. In addition, our method effectively inpaints structure that is not visible in any night

			Ours		Baseline	
Landmark	**# Day**	**# Night**	**TP**	**FP**	**TP**	**FP**
Spanish Steps	1030	92	98.91	3.26	93.48	14.13
Moulin Rouge	880	754	87.00	0.93	85.81	1.33
Castel St'Angelo	1400	129	99.22	6.20	93.02	6.98
Astronomical Clock	2243	1375	97.89	5.60	80.15	2.98
Altare d. Patria	1993	357	97.76	2.52	92.72	4.20
St. Peter's Basilica	1980	495	98.99	2.22	87.47	6.46

Table 11.1: Quantitative evaluation of clustering accuracy for night images. Ground-truth labels obtained through manual labeling. Clustering accuracy specified as true positives (TP) and false positives (FP).

images and removes artifacts through blending.

11.7 Summary

In this chapter, we introduced a novel algorithm that handles and benefits from the variety of scene illuminations naturally present in large-scale Internet photo collections. This is in stark contrast to previous methods that treated multiple illuminations as a nuisance or failure condition. We exploit the additional information to obtain a more complete and accurate 3D model and to create multi-illumination appearance information for the 3D model. The proposed method demonstrates that we can leverage the additional information provided by the different illuminations to boost modeling quality for both geometry and appearance.

12 Robust Semantic Visual Localization

One of the core components and applications of any image-based 3D modeling pipeline [284, 288] is the localization of an image within an existing scene. Robust image-based localization under a wide range of viewing conditions is fundamental in enabling accurate and complete 3D modeling. Furthermore, image-based 3D models are the main input to a visual localization system, which is highly relevant for a wide range of applications, including autonomous robots [291] or mixed reality [166, 203], where visual localization is used for loop closure detection [77, 82, 389] and re-localization [193, 266] in SLAM [187, 194, 227].

Traditional image-based localization approaches based on low-level local image features (e.g., SIFT, SURF, etc.) provide limited robustness against strong viewpoint and illumination changes, as already demonstrated in Chapter 3. This limitation is one of the primary reasons for incomplete or degenerate 3D reconstructions. In the following, we propose an approach that enables robust visual localization in challenging scenarios by a joint geometric and semantic understanding of the scene.

In general, there are three types of approaches to the localization problem: *Structure-based* methods represent the scene by a 3D model and estimate the pose of a query image by directly matching 2D features to 3D points [51, 192, 266, 325, 386] or by matching 3D features to 3D points, if depth information is available [164, 389]. *Image-based* methods model the scene as a database of images [60, 77, 273, 324, 340]. They use image retrieval techniques to identify the database images most relevant to the query, which are then used to estimate the pose from 2D-3D matches. *Learning-based* methods represent the scene by a learned model, which either predicts matches for pose estimation [38, 39, 298, 348] or directly regresses the pose [162, 354]. This work follows the structure-based approach and represents the database scene by a semantic 3D map. Given a query image together with its semantic segmentation and depth map, we construct a 3D semantic query map from which we extract local descriptors. Using 3D-3D matches between query and database descriptors, we align the maps to obtain the query pose estimate.

All of the approaches, including ours, explicitly or implicitly measure the (visual or structural) similarity between a query image and the database scene representation. Thus, they assume that the query and database images depict the scene under sufficiently similar conditions in viewpoint, illumination, and scene geometry. As shown in Figure 12.1, these assumptions are easily violated in practice. Different illumination within a single day causes strong variation in appearance while seasonal changes significantly affect the scene geometry. Similarly, strong viewpoint

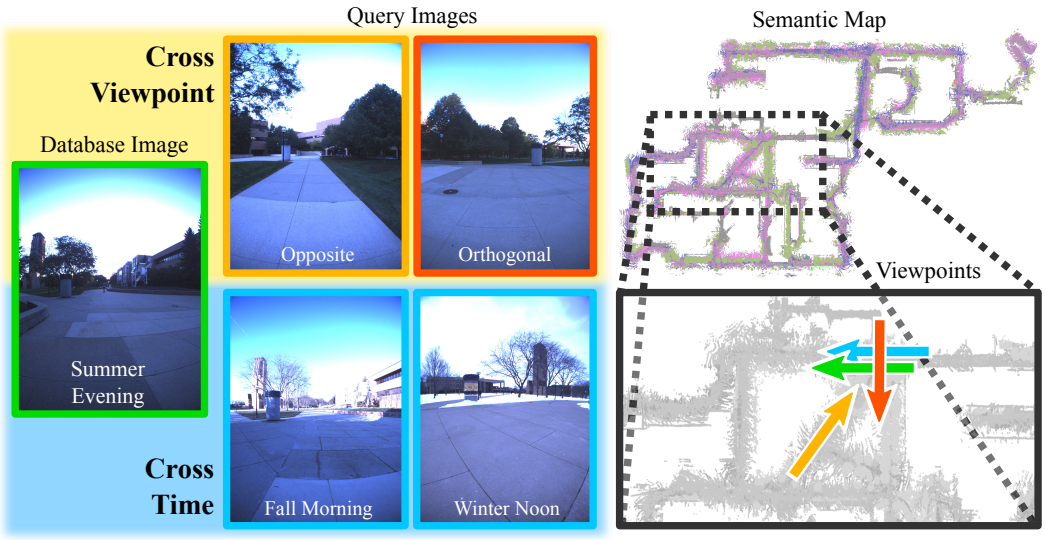

Figure 12.1: We propose a semantic localization technique which is able to match features over extreme appearance changes across viewpoints and time. In this example, the database contains only images captured in summer and from one particular viewpoint, yet our method correctly localizes images with strong viewpoint, illumination, and seasonal changes.

changes lead to severe perspective distortion and often result in little structural overlap between the query and the database. Yet, robustness against such changes is important, e.g., for AR devices or robots to re-localize robustly in a changing environment.

The main challenge in this setting is successful data association between the query and the database. Existing image- and structure-based methods use local features designed to be discriminative, e.g., [164, 301, 302, 389], such that descriptors of the same physical point are close in descriptor space while unrelated points are far apart. However, strong changes in viewing conditions, e.g., in appearance or geometry, demand for an invariant embedding which contradicts the discriminative learning objective of these approaches. In theory, such invariance could be implemented by learning a more complex descriptor comparison function [115, 384]. Yet, in practice, such methods do not scale well, as they require an expensive pairwise comparison of descriptors.

To overcome this limitation, we present a novel approach to descriptor learning that is based on a generative rather than a discriminative model. The core idea is to learn an embedding in Euclidean space that retains all information required to recover the scene appearance under different viewing conditions. Our embedding encodes high-level 3D geometric and semantic information and thus allows us to handle strong viewpoint changes as well as moderate changes in scene geometry, e.g., due to seasonal changes. More specifically, we propose to learn a generative

descriptor model based on the auxiliary task of 3D semantic scene completion. Given a partially observed scene, the goal of this auxiliary task is to predict the complete scene. A key insight of our work is that semantics provide strong cues for the scene completion task, resulting in drastically improved descriptors. We show that our descriptors can be learned in a self-supervised manner without explicit human labeling. The learned descriptors generalize to new datasets and different sensor types without re-training.

In summary, this work makes the following contributions:

- We propose a novel approach to visual localization based on 3D geometric and semantic information.
- We formulate a novel method to the descriptor learning problem based on a generative model for 3D semantic scene completion. The latent space of our variational encoder-decoder model serves as our descriptor and captures high-level geometric and semantic information.
- We demonstrate the effectiveness of our approach on two challenging problems: Accurate camera pose estimation under *strong viewpoint changes* and *illumination/seasonal changes*. Even without semantics, our approach outperforms state-of-the-art baselines by a significant margin, demonstrating the power of generative descriptor learning in localization. Incorporating semantic information leads to further improvements. To the best of our knowledge, ours is the first approach which reliably estimates accurate camera poses under such challenging conditions. In addition, our method generalizes to new datasets with different types of sensors without re-training.

12.1 Related Work

12.1.1 Traditional Approaches

Most existing large-scale localization methods use local features such as SIFT [200] to establish 2D-3D matches between features in a query image and points in a SFM model [192, 193, 266, 325, 386], These correspondences are then used to estimate the camera pose. Descriptor matching is typically accelerated using prioritization [193, 266] or efficient matching schemes [195, 203]. Co-visibility information [192, 266], an intermediate image retrieval step [146, 273], and geometric outlier filtering [51, 325, 386] aid in handling ambiguous features arising at large scale. If available, depth information can be used to remove perspective distortion effects before descriptor extraction [367, 387] or to directly extract descriptors in 3D [164, 263, 389]. However, even depth-based approaches fail in the presence of strong viewpoint or appearance changes due to a lack of visual or structural overlap. In contrast, we make our approach more robust to such drastic changes by learning a novel 3D descriptor specifically for these conditions. Recent learning-based methods for visual localization either learn to associate each pixel to a 3D point [38, 39, 298, 348] or

learn to directly regress the camera pose from an image [162, 354]. The principal drawback of both approaches is that they need to be retrained for each dataset. In contrast, our learned semantic descriptors generalize across datasets.

12.1.2 Semantic Localization

A popular strategy for semantic localization is to focus on features found on informative structures [170, 225] and to re-weight or discard ambiguous features [169]. Similarly, individual features [170] or Bag-of-Words representations [16, 303] can be enhanced by combining local features with semantics as a post-processing step. In contrast, our approach learns to combine semantics and geometry into a single and more powerful descriptor.

An alternative strategy to semantic localization is to use high-level features such as lane markings [291], object detections [17, 18, 264, 333], discriminative buildings structures [363], or the camera trajectory of a car [45]. These approaches need object databases or maps containing the same types of objects, which either requires careful manual annotation [291] or pre-scanning of objects [264]. In addition, the feature extraction and matching process is often a complex and hand-crafted solution tailored to specific objects [17, 18]. In contrast, our model learns a general semantic scene representation in a self-supervised fashion from data, eliminating the need for hand-crafted solutions or manual labeling.

12.1.3 Descriptor Learning

The traditional approach to descriptor learning in the general setting is to learn a discriminative embedding in Euclidean space from corresponding 2D patch samples [44, 115, 178, 301, 302]. The embedding function should produce similar descriptors for patches depicting the same physical structure and dissimilar descriptors for unrelated patches. The same approach also applies to 3D voxel volumes [389] and point clouds [164]. Typically, these descriptors are learned for local patches or local volumes in order to handle (partial) occlusions. In contrast, we are interested in learning descriptors with a larger spatial context in order to obtain a more powerful, high-level understanding of the scene. Consequently, we learn 3D descriptors for relatively large 3D semantic voxel volumes. The main challenge in our setting is that descriptors in the query and database map only have partial structural overlap due to their large spatial context and due to strong occlusions when matching under extreme viewpoint changes. We thus exploit the auxiliary task of semantic completion [312] to learn an embedding that is invariant to occlusions. In contrast to learning complex matching functions [115, 384] that are expensive to compute, our descriptor is embedded in Euclidean space and can be matched efficiently at large scale.

One application of our approach is localization under illumination and seasonal changes. There exists work on training local [197] or image-level [12, 60, 103, 104, 340] descriptors that are robust under such changes. Due to the challenge of ob-

taining accurately posed images under different conditions [268], these approaches are trained on data with little viewpoint changes, e.g., from webcams [60]. Thus, such approaches are not very robust under viewpoint variations. In contrast, we explicitly train on data with strong viewpoint changes and demonstrate that our model also generalizes to illumination and seasonal changes.

12.1.4 Semantic Model Alignment

The key idea of our method is to use the geometry and semantics to establish correspondences for pose estimation. Thus our approach is also related to methods aligning 3D models through semantic features [72, 73, 333, 375]. Cohen et al. [72, 73] use semantic features to stitch visually disconnected SFM models. Toft et al. [333] use a similar idea for camera pose refinement in localization. Yu et al. [375] use 3D object detections as features in a semantic ICP approach. These approaches use hand-selected semantic features, which are often ambiguous, e.g., there might be multiple cars in the scene. Hence, the association problem is solved either via brute-force search [73] or by assuming an initial alignment [72, 333, 375]. In contrast, our approach learns descriptors with a more general semantic scene understanding that can be matched efficiently at large scale.

12.1.5 Aerial-Ground Localization

A related problem to ours is tackled by work on matching ground-level imagery against overhead maps to obtain coarse location estimates under orthogonal viewpoint changes [55, 196, 359, 364]. However, these methods are specific to this problem and cannot be used for accurate ground-level to ground-level localization, which is the focus of this chapter.

12.2 Semantic Visual Localization

In this section, we describe our proposed method for semantic visual localization. The input to our system is a set of color images with associated depth maps $\mathcal{I} = \{I_i\}$ and, for database images, their respective camera poses $\mathcal{P} = \{P_i\}$ with $P_i \in SE(3)$. Given the subset of database images $\mathcal{I}_D$ and their camera poses $\mathcal{P}_D$, we create a global 3D semantic map M_D in a pre-processing step. For a query $\mathcal{I}_Q$ of one or multiple images, we compute a local 3D semantic map M_Q and establish 3D-3D matches between M_Q and M_D to determine the unknown query poses $\mathcal{P}_Q$. This localization procedure should be robust to extreme viewpoint and illumination changes between the database and the query images. While lower-level radiometric and geometric information typically varies significantly under different viewpoints and illumination, semantic information is comparatively invariant to these types of transformations through higher-level scene abstraction. This is the main motivation for our proposed semantic visual localization method which comprises the following three steps: In

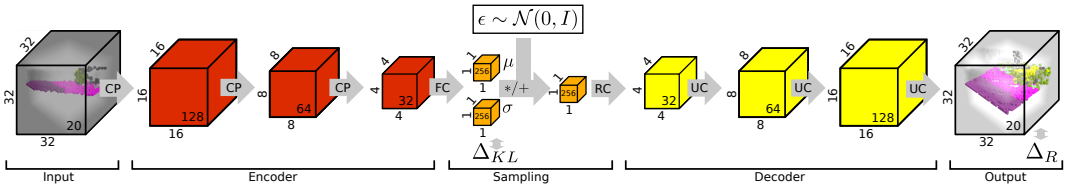

Figure 12.2: **Variational Encoder-Decoder Architecture.** Legend: CP = Convolution + Pooling, FC = Fully Connected, RC = Reshape + Convolution, UC = Upsampling + Convolution, Δ_{KL} = KL Divergence with respect to $\mathcal{N}(0, I)$, Δ_R = Reconstruction Loss. The numbers at the bottom right of each block denote the number of feature channels. The network takes incomplete semantic observations as input (left) and predicts completed semantic subvolumes (right). The latent code μ forms our descriptor.

an offline step, we learn robust local descriptors by exploiting semantic scene completion as an auxiliary task. During online operation, we use these local descriptors to establish 3D-3D matches between the query and database map. The matches are then used to estimate an alignment between the two maps, which defines a pose estimate for the query. In the following, we first describe how we construct the 3D semantic maps from which we learn and extract our proposed descriptors. We then explain the proposed descriptor matching and pose estimation stages.

12.2.1 Semantic Segmentation and Fusion

We first compute dense pixelwise semantic segmentations $\mathcal{S} = \{S_i\}$ for all input images, where each pixel of S_i is assigned a semantic class label $l \in \{1, \dots, L\}$. Next, we fuse the images into semantic 3D voxel maps M_D and M_Q for the database and query images [62, 116, 142]. Each voxel in the semantic 3D maps takes one of $L + 2$ labels, i.e., a voxel is either occupied with one of the L semantic classes or it is labeled as free space L_F or unobserved space L_U. The task of localization is to find the transformation $P \in SE(3)$ that best aligns a query to the database map.

Given a robust semantic classifier, e.g., trained specifically for different seasons, the semantic maps are inherently invariant to large illumination changes and geometric variations up to the voxel resolution. Note that using semantics, it is easy to determine reliable classes and, e.g., to ignore dynamic objects such as cars. While semantics abstract high-level scene information, large spatial context is needed for an unambiguous, instance-level understanding of the scene. However, a larger spatial context inherently leads to missing observations due to occlusions. For example, in the case of different viewpoints, the volumetric overlap between the query and database maps may be very small. In the extreme case of opposite viewing directions, there might be no structural overlap between the two maps. Hence, one main challenge for our pipeline is to robustly find matches between the query and database maps in the absence of common observations. In the following, we describe how to

learn an encoding of the database and query maps M_D and M_Q that is invariant to such missing observations through a semantic understanding of the scene.

12.2.2 Generative Descriptor Learning

The underlying goal of our localization method is to estimate the transformation P from 3D-3D matches between M_D and M_Q. Since the query and database maps typically differ in size and coverage, we find correspondences between subvolumes $v_D \in M_D$ and $v_Q \in M_Q$ of size V^3. To establish these correspondences, we learn a function that recognizes similar subvolumes. This function should be invariant to missing observations due to the relatively large size of the subvolumes, different viewpoints and moderate geometric deformations between the query and database map, dynamic objects in the scene, sensor noise, etc. In particular, the function should identify the same object even when seen from different viewpoints and under different illumination. We will show that semantic scene understanding is key to learning such an invariant function.

The two traditional approaches to solving this problem are to either learn a matching function $f(v_D, v_Q)$ [115, 384] or an embedding $f(v)$ [164, 389]. The latter approach aims to find an encoding function that maps the same subvolumes to similar points in (Euclidean) space. While a learned matching function in theory has more discriminative power, it also imposes high computational cost as it requires exhaustive pairwise comparisons, which is intractable at large scale. Thus, we learn an embedding that is evaluated only once per subvolume rather than per pair of subvolumes.

More concretely, we learn an encoding function $f(v) \in \mathbb{R}^N$ that maps a subvolume to a lower-dimensional descriptor which jointly encodes the scene semantics and geometry. To recognize the same object from different or even opposing viewpoints, this encoding must contain enough information to hallucinate the unobserved parts of the subvolume. Towards learning such a robust encoding, we define the auxiliary task of semantic scene completion. This auxiliary task is described by the function $h(v)$ that hallucinates the geometry and the semantics of the unobserved parts of its input. We use a 3D variational encoder-decoder $h(v) = g(f(v))$, where f is a neural network which encodes the incomplete subvolume and g is a neural network which hallucinates the complete subvolume. To learn the distribution of the space of subvolumes and to ensure that the same physical subvolumes map to nearby encodings in Euclidean space, we enforce a Gaussian prior on $(\mu, \sigma) = f(v)$ using variational sampling. Our formulation is similar to the original variational autoencoder [165] with the difference that we encode an incomplete sample and decode the complete sample.

For learning h, we generate training data using volumetric fusion. We first fuse all training images $\mathcal{I}_T$ into a volumetric representation. This yields a nearly *complete* representation M_T of the scene. In addition, we create *incomplete* volumetric representations M_{T_i} for each image $I_{T_i} \in \mathcal{I}_T$ individually. During stochastic gradient descent, we randomly sample incomplete subvolumes $\bar{v}$ in $\{M_{T_i}\}$ and find its cor-

responding complete subvolume $\hat{v}$ in M_T. The task of h is to denoise the observed parts and to hallucinate the unobserved parts of the incomplete subvolume. The learning objective is the semantic reconstruction loss $\Delta_R = \mathrm{E}(h(\bar{v}), \hat{v})$ using the categorical cross entropy measure $E(\cdot, \cdot)$. Together with the Gaussian prior, the overall training objective is defined as $\Delta = \Delta_R + \Delta_{KL}$ where Δ_{KL} measures the Kullback-Leibler divergence between the latent code f and $\mathcal{N}(0, I)$. The architecture of h is illustrated in Figure 12.2 and examples are shown in Figure 12.3.

We learn the model for a fixed voxel size using the same orientation for the incomplete and complete subvolumes $\bar{v}$ and $\hat{v}$. For additional data augmentation and robustness to noise, we jointly rotate the subvolumes using random orientations and perturb the occupancy of the incomplete subvolumes using dropout [316]. Note that no human labeling is required since we employ pre-trained semantic classifiers for this task. As described next, our semantic localization pipeline only uses the encoder part f of the full model h.

12.2.3 Bag of Semantic Words

The previous section described how to learn a discriminative function f that maps semantic subvolumes v to low-dimensional latent codes μ. We use this function to create a *Bag of Semantic Words* that encodes the semantic 3D scene layout of a map. The bag of semantic words $\mathcal{F}(M) = \{f(v_j)\}$ with $j = 1 \dots |M|$ is defined as the set of descriptors $f(v_j)$ computed for all occupied subvolumes v_j in M. We consider a subvolume to be occupied if at least one voxel within the subvolume (not necessarily the center voxel) is occupied. Note that given an *incomplete* map, the bag of semantic words is a description of its *complete* semantic scene layout. Our localization pipeline uses this representation to robustly match the query map to the database map, as detailed in the following.

12.2.4 Semantic Vocabulary for Indexing and Search

We establish correspondence between subvolumes in the query and database using nearest neighbor search in the descriptor space $f(v) \in \mathbb{R}^N$ using the Euclidean metric. For efficient semantic word matching, we build a semantic vocabulary [231] in an offline procedure using the bag of semantic words of the training dataset. We quantize the space of descriptors $\mathcal{F}(M_T)$ using hierarchical k-means and a N_B-dimensional Hamming embedding [148]. We index all semantic words of the database map $\mathcal{F}(M_D)$ into the resulting vocabulary tree, which serves as an efficient data structure for matching. To find matches between a given query image and the database, we find the top $K = 5$ nearest database words $\mathcal{D}$ for each query word $f(v_j) \in \mathcal{F}(M_Q)$ by traversing the vocabulary tree and finding nearest neighbors in Hamming space. Since our descriptors are *rotation variant*, as they are trained on aligned subvolumes (see Section 12.2.2), and, generally, we have no *a priori* knowledge about the orientation of the query, we perform the same query for a fixed set of orientation hypotheses $\theta \in SO(3)$ while the database remains fixed. The set of

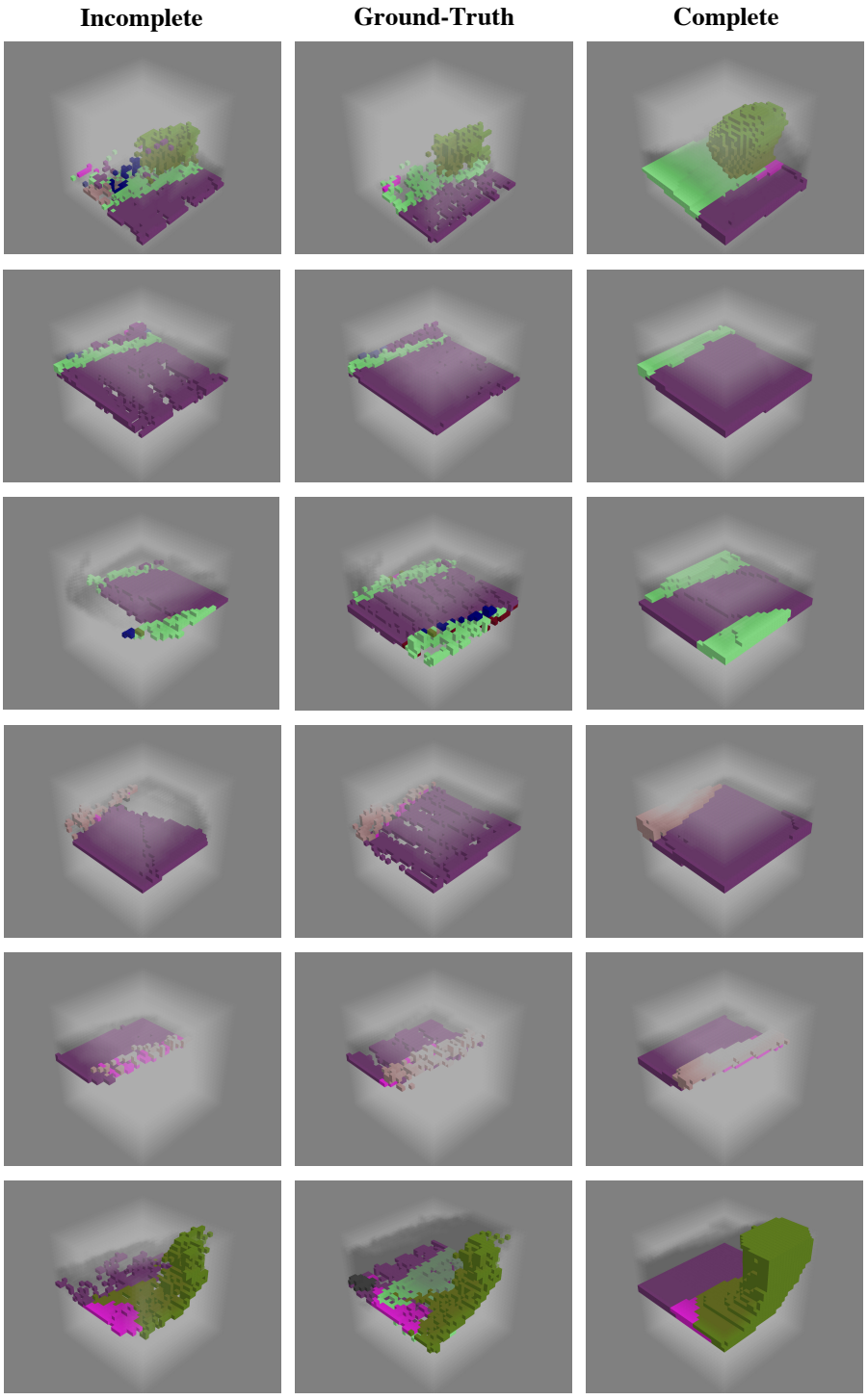

Figure 12.3: Example input $\bar{v}$ and output $h(\bar{v})$ from the KITTI dataset for our semantic completion auxiliary task. The incomplete input is completed using our encoder-decoder network h, while the multi-view fusion $\hat{v}$ is the ground-truth.

putative matches $\mathcal{D}(\theta)$ for the different orientations provide evidence for the location of the query. The next section details how to accurately localize the query based on this evidence using a joint semantic map alignment and verification.

12.2.5 Semantic Alignment and Verification

Given the putative matches $\mathcal{D}(\theta)$ from Section 12.2.4, we seek to find the transformation $P \in SE(3)$ that best aligns the query to the database map. Specifically, a good alignment is established if both the geometry (i.e., occupancy) as well as the semantics agree. Due to the rotation variance of our descriptors, a single 3D-3D match between the query and the database defines a transformation hypothesis P, which is composed of the rotation defined by θ and the translation $t \in \mathbb{R}^3$ defined by the spatial offset of the corresponding subvolumes. We exhaustively enumerate all transformation hypotheses defined by the matches. To verify a single transformation hypothesis, we then align the query to the database map using P and count the number of correctly aligned voxels of the query map. A correctly aligned voxel matches both in terms of geometry and semantics, i.e., an occupied voxel in the aligned query map must also be occupied in the spatially closest voxel in the database map. In addition, the spatial distance of their voxel centers must be smaller than κ and their semantic class labels must match exactly. We ignore unobserved voxels in both the query and the database map. To further refine the alignment, we use the iterative closest point algorithm [30], where closest points are defined as the set of correctly aligned voxels in the previous iteration. Finally, we rank the transformation hypotheses by the ratio τ of correctly aligned over occupied voxels in the query map. The top-ranked hypotheses define the query pose estimates as the output of our system.

12.3 Experiments

In this section, we compare our method to the state-of-the-art techniques on several large-scale localization benchmark datasets. The following sections explain the setup and results of our experiments in detail.

12.3.1 Datasets

KITTI

We evaluate the localization performance in the setting of extreme viewpoint changes on the KITTI odometry dataset [106] comprising 11 sequences with ground-truth poses. 6 of these sequences contain loops with extreme viewpoint changes. First, we evaluate on the traditional loop closure scenario including all images independent of viewpoint change. For this, we construct the database map from all images, while the query map is constructed from a single image. In addition, we also perform experiments when the database only contains images from significantly different

viewpoints (90° or 180°) and under different appearance (see Figure 12.4). In this case, all images with similar viewpoint are excluded from the database.

NCLT

In addition, we use the NCLT dataset [53] to evaluate the localization performance under extreme appearance changes caused by short-term illumination changes over several hours and long-term seasonal changes over several months. The dataset was acquired biweekly during 1.5 hour sessions over the course of 15 months and we selected 4 sequences that span the different modalities of daytime (morning, midday, afternoon, evening), weather (sunny, partially cloudy, cloudy), (no) foliage, and (no) snow.

12.3.2 Setup

For all evaluations, we compute gravity-aligned semantic maps by using either the integrated inertial sensors (NCLT) or through vanishing point detection in the images (KITTI). This reduces the space of orientation hypotheses θ to the rotation around the gravity axis. We use the raw output of an off-the-shelf semantic classifier [374] trained on the Cityscapes [74] dataset. This classifier segments the scene into $L = 19$ semantic classes and we only consider the maximum activation per pixel and discard any pixels with *sky* labels. We adapted the volumetric fusion approach by Hornung et al. [142] using (multi-view) stereo depth maps [135] (KITTI) and sparse lidar measurements (NCLT) for efficient large-scale semantic fusion. We extract subvolumes v of size 32^3 at a fixed voxel resolution of 30cm, resulting in a $10m^3$ spatial context. At this resolution, the bag of semantic words for a single image query map contains several thousand descriptors for a fixed set of 18 uniformly spaced orientation hypotheses θ. A single forward pass of one volume takes around 1ms on a NVIDIA Titan X GPU while the geometric verification has negligible performance impact in comparison, leading to an average throughput of around one query image per second, which is on par with the fastest baselines we compare to. For pose estimation, we empirically choose a maximum error of $\kappa = 3$ meters and a minimum overlap ratio of $\tau = 0.3$.

12.3.3 Training

Throughout all experiments, we employ the same semantic descriptor and vocabulary trained offline on two sequences of around 10k and 20k frames [370] with accurate ground-truth poses acquired over several kilometers. The dataset contains a large number of loops with extreme viewpoint changes and is independent of the evaluation datasets. Note that this dataset is much more similar to KITTI as compared to NCLT, since it is an autonomous driving dataset with a stereo camera pair that we use for volumetric fusion. Nevertheless, we show that our descriptor generalizes well to NCLT. We train our encoder-decoder architecture on 16M random

subvolumes using SGD and choose a latent code size of $N = 256$ as a trade-off between speed and accuracy. In our experiments, $N < 64$ led to significantly reduced reconstruction and localization performance, while $N > 512$ did not significantly improve the results. Our semantic vocabulary is represented by 2^{16} semantic words embedded in a $N_B = 64$ dimensional Hamming space, using a hierarchical branching factor of 256.

12.3.4 Baselines

In the following, we briefly present the chosen state-of-the-art baselines for our evaluation. We evaluate the localization performance on the ratio of correctly localized query images within a given error threshold. In addition, we show the rank-recall curves for an error threshold of 1m and 5m. For additional implementation details of the baselines, we refer the reader to the supplementary material.

SIFT and DSP-SIFT

We employ a state-of-the-art visual localization pipeline [270, 287] using SIFT [200]. This pipeline is based on a visual vocabulary tree embedded in a Hamming space [148] with visual burstiness weighting [150] and uses 2D-2D matching on the top-ranked retrievals to obtain 2D-3D correspondences for absolute pose estimation. Instead of SIFT, we also evaluate using DSP-SIFT [80], which has been shown to perform significantly better [23, 285].

MSER and VIP

Furthermore, we replaced the standard SIFT keypoint detector with MSER [207] features, which are designed for wide baseline matching. Then, we extracted DSP-SIFT features and kept the rest of the SIFT pipeline as described above. In addition, we experimented with Viewpoint Invariant Patches [367], where we rectified the ground plane to cancel the effect of perspective distortion before extracting SIFT features. However, we found that we could not match features for 90° and 180° viewpoint changes on the KITTI dataset. Our main insight from this experiment was that the geometric and radiometric distortions are too severe for low-level appearance matching.

DenseVLAD

To study the impact of image retrieval on the localization performance, we replaced the vocabulary tree based ranking with DenseVLAD [340] as a global image descriptor and otherwise use the SIFT pipeline as described. DenseVLAD produces state-of-the-art retrieval results under large viewpoint [273] and illumination changes [340].

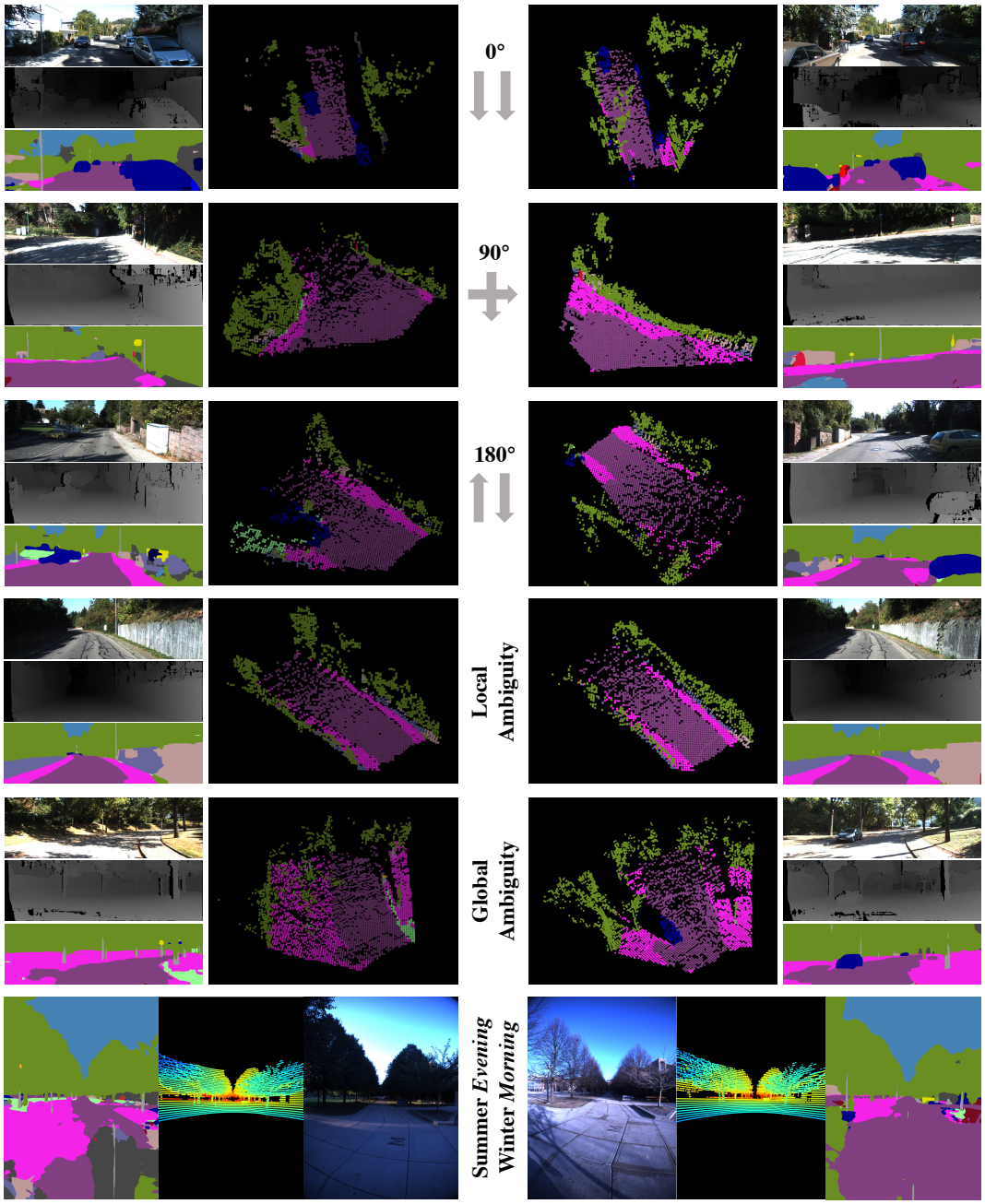

Figure 12.4: Example scenes in the KITTI and NCLT datasets for the different loop closure scenarios, including two failure cases caused by local and global ambiguities.

FPFH, CGF, and 3DMatch

FPFH [263], CGF [164], and 3DMatch [389] are state-of-the-art hand-crafted and learned geometric shape descriptors used for point cloud matching. We use them as a replacement for our descriptor while keeping the rest of the pipeline as proposed. Equivalent to our setup, we densely extract these shape descriptors using the same spatial descriptor radius and train a custom shape vocabulary on our training dataset. In addition, we fine-tune 3DMatch on our data using corresponding incomplete and complete volumes, resulting in a small performance boost.

PoseNet and DSAC

We train separate PoseNet [162] models for each sequence as a state-of-the-art representative of end-to-end pose regression methods. We also experimented with DSAC [39] but failed to obtain meaningful results even for the smallest KITTI sequence.

Ours

We compare the above state of the art against our semantic localization pipeline, denoted as *Ours (semantic)*. To quantify the performance impact of semantic information on the localization task, we additionally train a version of our descriptor and vocabulary using uniform semantics with only a single label. As a result, the fused 3D maps contain only occupied, free, and unobserved space labels. Opposed to the FPFH, CGF, and 3DMatch, this can be seen as an occlusion-aware geometric shape descriptor, denoted as *Ours (geometric)*. In addition, we show results for fusing multiple frames into the query map, denoted as *Ours (acc.)*. Opposed to fusing a single frame, we fuse five consecutive frames into a query map and therefore demonstrate the benefit of accumulating more evidence over time to obtain less noisy and more complete query maps.

12.3.5 Implementation Details

Ours

We use a batch size of 32 to train our encoder-decoder network for 2,000 epochs using ADADELTA [385] as an adaptive learning rate method for stochastic gradient descent. We set the initial learning rate to $\eta = 1$ without decay and set the hyperparameters to $\rho = 0.95$ and $\epsilon = 10^{-8}$. All convolutional layers use a filter size of $3 \times 3 \times 3$ using zero-padding and $ReLU$ activation followed by a $2 \times 2 \times 2$ max-pooling layer. The fully-connected layers are followed by a *tanh* activation function. Upsampling is implemented by repeating the data in the spatial domain by a factor of $2 \times 2 \times 2$. The final convolutional layer of the decoder is followed by a softmax activation. There is a total of around one million learned parameters in our network. For data augmentation, we apply a dropout of 10% on the voxels of the incomplete volume. The reconstruction loss $\Delta_R \in \mathbb{R}$ is emphasized by a factor

of 10 relative to the Gaussian prior loss $\Delta_{KL} \in \mathbb{R}$. In addition, for faster convergence during training, the reconstruction loss on occupied voxels is emphasized by a factor of 10. In the experiments, we use 18 uniformly spaced orientation hypotheses $\theta = \{0°, 20°, \ldots, 340°\}$ around the gravity axis.

SIFT

For SIFT feature detection, we use 4 octaves starting with a two times up-sampled version of the original image, 3 scales per octave, a peak threshold of $\frac{0.02}{3}$, an edge threshold of 10, and, due to the gravity-aligned input, an upright orientation assumption. The visual vocabulary is represented by 2^{16} visual words embedded in a $N_B = 64$ dimensional Hamming space and using a hierarchical branching factor of 256. Using these settings, we obtain several thousand descriptors per image. Localization is performed using a traditional image retrieval setup with two-view geometric verification on the top-ranked retrievals and 2D-3D camera pose estimation inside RANSAC followed by a non-linear refinement. A camera pose is considered as verified, if it has at least 15 2D-3D inlier correspondences. The 3D map is obtained through fusion of all database depth maps. Similar setups achieve state-of-the-art results [265, 270].

DSP-SIFT

We use the same feature detector as for standard SIFT and a total of 10 pooling scales uniformly spaced between $\frac{1}{6}$ and 3. We train a new visual vocabulary for nearest neighbor search. Otherwise, we use the same setup as for standard SIFT.

MSER

Using the same setup as for DSP-SIFT, we replaced the SIFT keypoint detector with MSER using a step size between 2 intensity threshold levels, region sizes between 30 and 14000 pixels varying by a maximum of 25%. We extract DSP-SIFT descriptors as described previously for the detected regions and train a new visual vocabulary for nearest neighbor search.

VIP

For our VIP experiments, we manually selected road regions in multiple images from different viewpoints, fitted a plane through the 3D road points, normalized image to a fronto-parallel viewpoint, densely extracted SIFT descriptors, and matched them exhaustively between the images. For all but very similar viewpoints, we failed to establish correct correspondences between the images. Our main insight from this experiment was that the geometric and radiometric distortions are too severe for low-level appearance matching. We therefore excluded VIP from the further evaluation.

DenseVLAD

Using the same setup as for DSP-SIFT, we extract a 4096 dimensional global image descriptor using DenseVLAD, which replaces the visual vocabulary based image retrieval pipeline. To find nearest neighbor images for a given query images, we exhaustively compare the global image descriptors from the query to the database and then perform two-view geometric verification on the top-ranked retrievals, equivalent to the DSP-SIFT experiment.

FPFH

Using the same keypoint locations and geometric verification approach as for our method, we extract 33 dimensional FPFH descriptors and train a new vocabulary for nearest neighbor search.

CGF

Equivalent to FPFH, we replaced our learned descriptors with 32 dimensional CGF descriptors and train a new vocabulary for nearest neighbor search. We consistently oriented the point cloud normals between the query and database maps towards the cameras.

3DMatch

For this experiment, we tried both the pre-trained 3DMatch models and also fine-tuned the descriptor using corresponding complete and incomplete subvolumes that we also used to train our descriptor. The fine-tuned model performs slightly better and we use it for our experiments. Equivalent to FPFH, we then replaced our learned descriptors with 512 dimensional 3DMatch descriptors and train a new vocabulary for nearest neighbor search.

PoseNet

For each database, we train a separate PoseNet model from scratch until convergence, which required more than 2 days of training for the largest models. We then regress the pose for each query image, which serves as the single, top-ranked pose hypothesis for the evaluation.

DSAC

For this experiment, we trained DSAC from scratch using the suggested initialization protocol, which took around 2 days for the smallest KITTI odometry sequence 04. However, we could not produce meaningful pose estimates and our main insight from investigating the issue was that the current DSAC approach has problems with repetitive structures and larger scale outdoor scenes. We therefore excluded DSAC from the further evaluation.

12.3.6 Results

Scene Completion

Figure 12.3 shows results of the semantic completion task. We attain an average semantic reconstruction accuracy of 87% on the test data. In most cases, the completed volume is a spatially smoothed approximation of the ground-truth. We conclude that the network learns powerful semantic representations of the world and meaningfully hallucinates the missing parts. Using this model for all further evaluations, we next demonstrate the performance of our approach on the task of localization.

0° Localization Scenario

First, we evaluated our method on the traditional loop closure scenario when images from similar viewpoints as the query image are indexed in the database. Figure 12.5 shows the results for KITTI in the 0° column and on the main diagonal for NCLT. Our method achieves state-of-the-art errors for the top-ranked localization proposals and clearly outperforms the other geometric shape descriptors, whereas our semantic and geometric descriptors perform roughly on par. While (DSP-)SIFT and DenseVLAD achieve lower performance for the top-ranked proposals, they clearly outperform all other methods when retrieving many images. This is not surprising as this task is rather easy. Both the query and the database images depict the scene from similar viewpoints and are taken close to each other in time. Thus, low-level image statistics encoded by, e.g., SIFT or DenseVLAD are very discriminative.

90° Localization Scenario

To evaluate the methods in a more challenging setting, we consider 90° trajectory intersections, which is a common scenario in most robotic applications, e.g., when a car passes a street crossing twice but in orthogonal directions. For this experiment, we excluded all images from the database that are not within a viewpoint change of $90° \pm 20°$ with respect to the query images. The query consists of a short sequence of 20 images and the results are averaged over all 57 intersection cases in the dataset. Figure 12.5 shows that this is indeed a very difficult scenario.

As expected, SIFT is not able to localize any query images due to the extreme viewpoint change, while DSP-SIFT and DenseVLAD perform slightly better. In contrast, FPFH and our method are able to localize a significant number of queries due to their invariance to appearance changes. Surprisingly, CGF and 3DMatch are not as robust as FPFH. Moreover, due to the different viewpoints, the geometry between the query and database map is not the same. Attributed to this fact, our approach achieves much better performance than existing geometric descriptors, because ours is specifically learned for invariance against missing observations. Moreover, our semantic descriptor outperforms our geometric descriptor by a large margin, highlighting the importance of semantic information for recognizing scenes

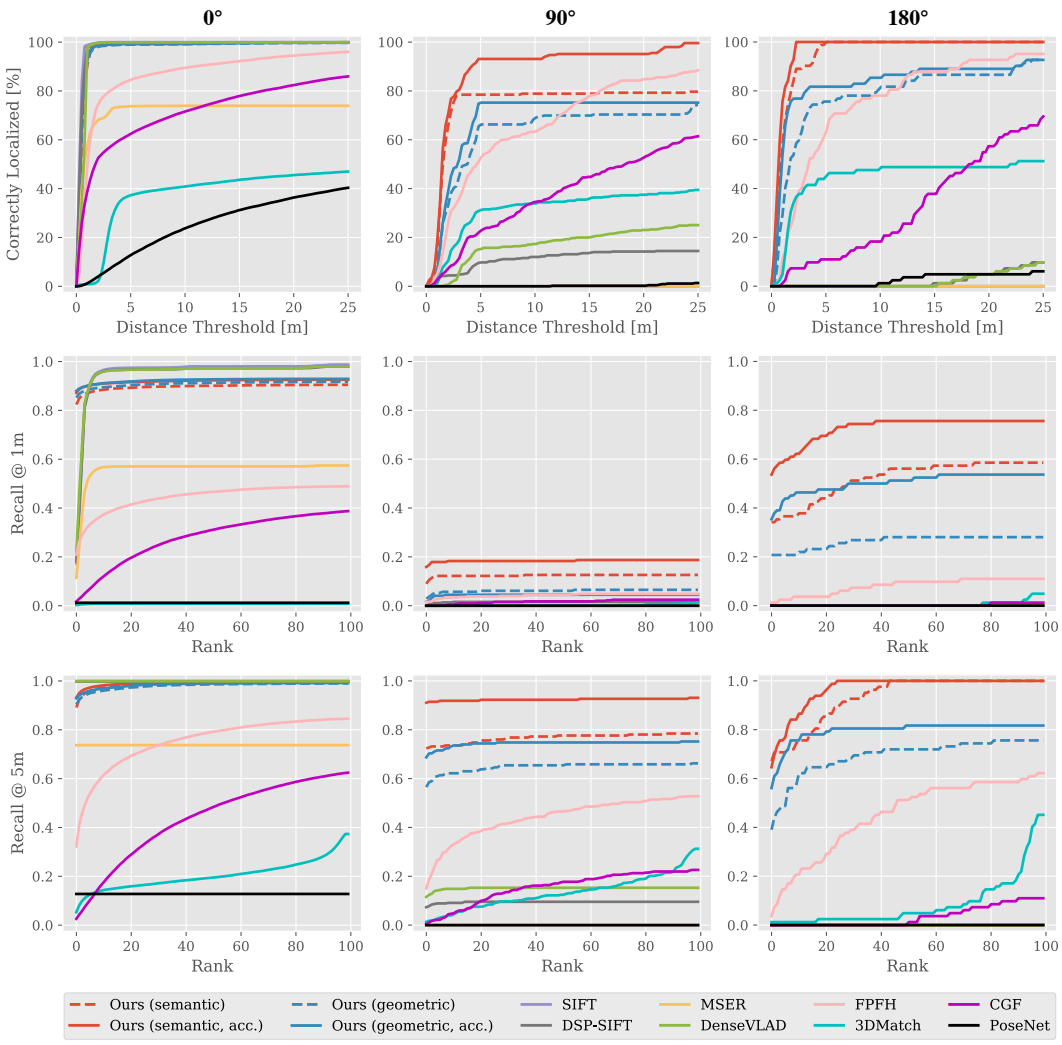

Figure 12.5: Localization results for the cross-viewpoint (0°, 90°, 180°) scenarios in the KITTI odometry dataset.

from different viewpoints. We obtain another significant boost in performance by accumulating evidence over a short window of five frames.

180° Localization Scenario

Equivalent to the previous experiment, we also detected 180° trajectory intersections, which, for example, occur when a car passes the same street but in opposite directions. Again, we excluded all images from the database that are not within a viewpoint change of $180° \pm 20°$ with respect to the query images. The results in Figure 12.5 are averaged over all such cases. In this scenario, there is almost no visual overlap between the query and the database apart from, e.g., the street or walls. Consequently, SIFT is not able to localize any of the query images, while the geometric approaches succeed in this task. Notably, this task seems to be easier than the 90° scenario. We attribute this to the fact that in KITTI the 180° intersections typically have larger structural overlap between the query and database as compared to the 90° intersections.

Cross-Time Localization

The experiments on NCLT in Figure 12.6 show that all but our method fail to robustly localize under extreme appearance changes caused by different geometry (foliage, snow) and illumination (time of day, weather). We observe that, in the cases where (DSP-)SIFT and DenseVLAD succeed, they mostly use stable features on buildings rather than vegetation or ground. In contrast, FPFH and 3DMatch are more robust to illumination changes. However, our method consistently outperforms all other methods across all scenarios. Note that our descriptor was trained on a KITTI-like dataset using stereo for map fusion instead of lidar used in NCLT. Furthermore, the same semantic classifier is employed for NCLT and KITTI. This demonstrates that our approach is robust to different types of input data and can be deployed in a wide range of settings without re-training. Looking at Figure 12.3, it is not surprising that our approach generalizes to this new task as the network mainly focuses on the overall scene geometry and semantics.

Failure Cases

While our method robustly localizes queries in all scenarios, we observed a few common failure cases. The two most systematic errors are joint semantic and geometric ambiguities at a local or global scale (see Figure 12.4), also commonly occurring in traditional approaches [192, 270]. Local ambiguities are caused by repetitive structures, causing wrong localization in the order of tens of meters. Global ambiguities are more rare and typically result in a localization error of hundreds of meters. Accumulating evidence over multiple frames significantly reduces their impact. Furthermore, our classifier is trained on Cityscapes which only depicts daytime images during spring/summer/fall. Training classifiers tailored to different modalities should further improve the performance of our approach.

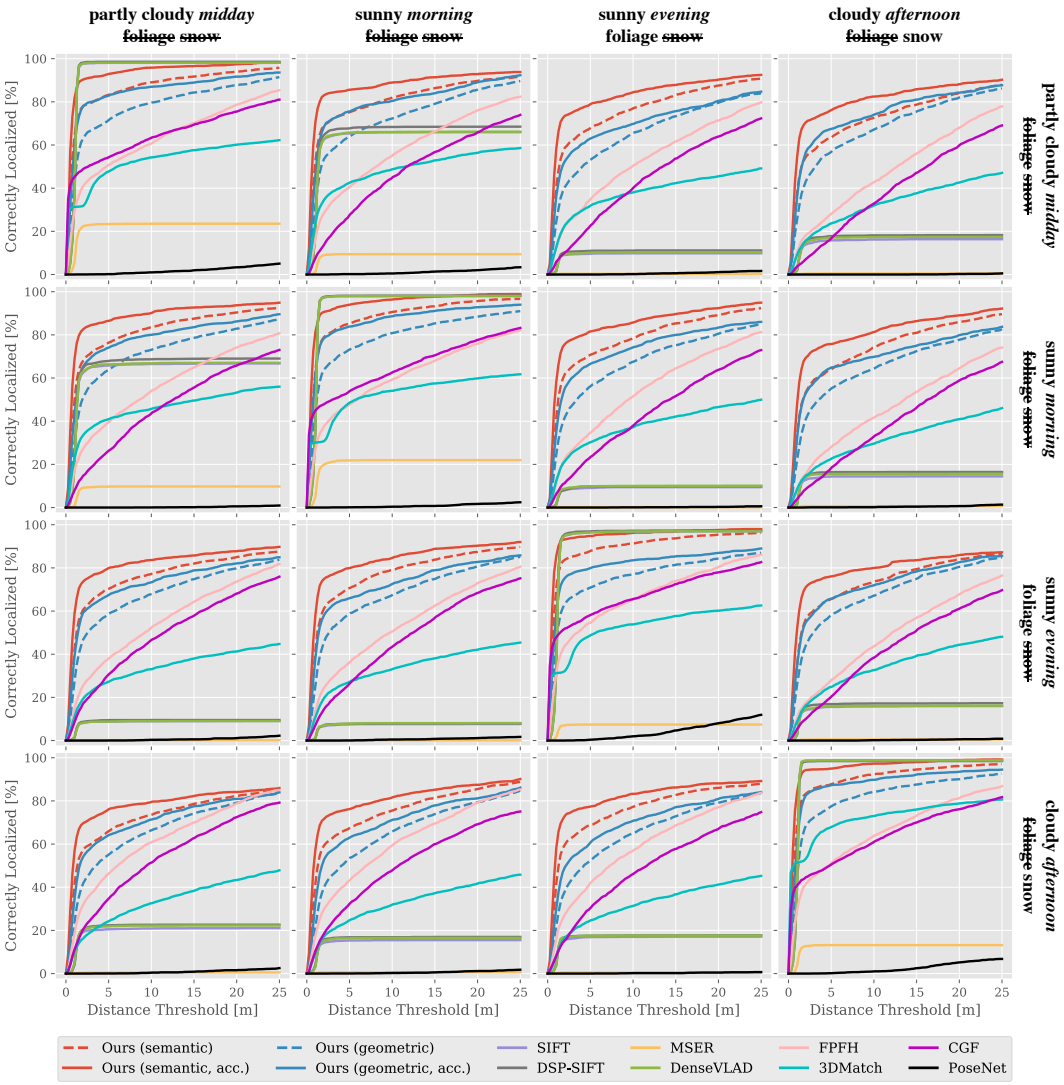

Figure 12.6: Localization results for the cross-time localization scenarios in the NCLT dataset. Striked-through text denotes the absence of a property.

12.4 Summary

In this chapter, we proposed a novel method for image-based localization using a joint semantic and geometric understanding of the 3D world. At its core lies a novel approach to learning robust 3D semantic descriptors. Being the first to demonstrate reliable loop closure and localization even under extreme viewpoint and appearance changes, we believe that our method is an important step towards robust, life-long localization in applications such as autonomous robots or mixed reality. While this chapter focused on static scenes, an avenue for future research is the exploration of robustness against strong geometric changes caused by scene dynamics.

Part VI

Conclusion

13 Summary

The work in this thesis considers the entire pipeline of an image-based 3D modeling system. The presented contributions build upon the fundamental building blocks of image-based 3D modeling that have been developed over the last decades. The aim of this thesis was to push the state of the art in terms of algorithmic efficiency, generalization robustness across different scenarios, and accuracy of the resulting 3D models. Towards this goal, we presented algorithmic improvements and new theoretical insights for the different stages in an image-based 3D modeling pipeline. Many of the proposed algorithmic contributions follow the common idea of tightly integrating classic, hand-crafted and automatically learned algorithms, derived by a deep understanding of the underlying problems. Throughout the thesis, thorough experimental evaluation lies at the core of all presented contributions. Experimental evaluation was instrumental in both generating novel insights as well as validating the performance of the proposed algorithms.

In Part I of the thesis, we introduced the basic principles in computer vision, (multi-view) geometry, and optimization that were a pre-requisite to understand the thesis. In Part II, we conducted a thorough experimental evaluation of learned and hand-crafted local feature descriptors to better understand their performance and impact across a wide range of scenarios in image-based 3D modeling. Next, we presented three different algorithms that drastically improve both the efficiency and robustness of two-view reconstruction from unstructured image collections. Part III introduced several algorithmic improvements to the incremental sparse reconstruction paradigm and demonstrated improvements over the state of the art in terms of robustness, accuracy, completeness, and scalability. In Part IV of the thesis, we first focused on robust and efficient dense stereo estimation using the semi-global matching algorithm. In the second chapter of Part IV, we presented an efficient and robust multi-view stereo algorithm for high-quality dense reconstruction from unstructured imagery. The last part of the thesis presented an end-to-end 3D reconstruction system, which tightly integrates an image retrieval and 3D modeling system. The system takes a single query image of the scene as input and incrementally reconstructs a detailed and accurate 3D model of the entire scene. Furthermore, we extended the system to enable illumination robust 3D modeling in the presence of mixed day- and nighttime images. Finally, we proposed a robust visual localization approach, which is based on joint semantic and geometric understanding of the 3D world. Experiments on several datasets demonstrated reliable loop closure and localization even under extreme viewpoint and appearance changes. As such, this chapter presented an important step towards robust, life-long localization in applications such as autonomous robots or mixed reality.

As part of the thesis, the developed methods and algorithms have been combined into an end-to-end image-based 3D modeling system for sparse and dense reconstruction from unstructured imagery. The individual components alone and the overall system pushes the state of the art in terms of robustness, efficiency, and accuracy, as demonstrated in numerous evaluations in this thesis and other recent 3D reconstruction benchmarks [167, 290]. The system is released as open-source software[1] in order to facilitate reproducibility and future research. Over the last years, the software has found widespread use among hobbyists as well as in industry and academia. It has been used for educational purposes in two tutorials at the International Conference on 3D Vision 2016 in Stanford and at the Conference on Computer Vision and Pattern Recognition 2017 in Hawaii. During the year of 2017 and the first half of 2018, the software has been downloaded more than 100,000 times[2].

[1] `https://github.com/colmap/colmap`
[2] Download statistics from `https://github.com/colmap/colmap` and `https://demuc.de`

14 Future Work

In this thesis, we pushed the state of the art in image-based 3D modeling. However, many open problems and challenges remain towards the ambitious goal of a truly general-purpose image-based 3D modeling system, that can produce highly accurate and photo-realistic reconstructions from any sensor and any scene. In this chapter, we discuss some of the fundamental problems that still exist today and we provide an outlook to potential future approaches that might overcome these problems.

Efficiency and Scalability. While the work in this thesis considered scalable algorithms for millions of images, the wealth of visual data around the world exceeds orders of magnitude more images. Despite the various speedups presented in this thesis and other recent works, even more efficient and scalable algorithms are necessary in order to leverage the entire wealth of visual data. This is especially relevant if we want to maintain an updated history of the world, which requires to constantly keep up with the ever-increasing amount of images.

Time and Dynamics. The scope of this thesis was restricted to the reconstruction of static scenes. However, the real world is dynamic and temporal changes are an essential part of it. Reconstructing the dynamics of the real-world is crucial in, for example, enabling temporal scene reasoning or in providing realistic virtual experiences. Early works in dynamic scene reconstruction already demonstrated its immense potential, though more work is needed to make these algorithms efficient and robust in the setting of unstructured images at large scale. Since dynamic reconstruction is often a highly under-constrained problem, it is necessary to enforce strong priors and the existing works typically used low-level geometric priors. In the future, it will be necessary to incorporate higher-level semantic priors to push the state of the art.

Semantics and Reasoning. The shown reconstructions in this thesis consist of purely low-level geometric and radiometric information. While this is sufficient for many tasks, a semantic 3D reconstruction of the scene can enable more sophisticated applications. For example, knowing the semantics of an object allows for realistic scene animation or the semantic scene completion of unobserved parts. Furthermore, many of the the current failure cases in image-based 3D modeling, such as textureless, reflective, or repetitive objects, require semantic reasoning to be reliably solved. Most of the current semantic reconstruction approaches work in controlled and small-scale environments. In the future, it will be necessary to make these methods more robust and efficient in order to apply them to unstructured and large-scale datasets.

Hand-Crafting and Learning. Traditionally, all the components of an image-based 3D modeling system have been hand-crafted through a detailed understanding of geometry, optimization, and the physics of the image formation process. Over the last years, more and more of these hand-crafted steps have been replaced by automatically learned algorithms, facilitated by the tremendous evolution in machine learning and in particular deep learning. In this thesis, we presented and evaluated several learned algorithms as a replacement of traditional hand-crafted approaches in order to improve the efficiency, robustness, and accuracy of the image-based 3D modeling pipeline. One fundamental challenge in developing learned algorithms is to ensure the generalization capability across different scenarios, which is especially important in the setting of unstructured imagery. In the future, it will be interesting to see which other parts of the pipeline can be replaced by learned components and whether it is, in principle, possible to build a fully learned reconstruction pipeline. In this regard, it also remains unclear whether a fully learned system will be superior, since decades of research provided us with a deep understanding of the underlying physical problem. Furthermore, the entire pipeline of image-based 3D modeling is a comparatively large and complex program. It is an open question how much capacity or memory is necessary to effectively model such a program automatically through learning. In the far future, we envision a system where automatically learned algorithms are tightly integrated with existing knowledge about geometry and optimization, with the potential of facilitating end-to-end learning of a joint geometric and semantic understanding of the world.

Acronyms

BOW	Bag of Words
CNN	Convolutional Neural Network
DLT	Direct Linear Transform
DOG	Difference of Gaussians
EM	Expectation Maximization
GEM	Generalized Expectation Maximization
GRIC	Geometric Robust Information Criterion
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimation
MVS	Multi-View Stereo
PCA	Principal Component Analysis
RANSAC	Random Sample Consensus
SFM	Structure-from-Motion
SGM	Semi-Global Matching
SIFT	Scale-Invariant Feature Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machine
VLAD	Vector of Locally Aggregated Descriptors

Glossary

$\mathbb{R}^n$	The set of n-dimensional real numbers
$\mathbb{Z}^n$	The set of n-dimensional complex numbers
$\mathbb{P}^n$	The n-dimensional projective space
a	A real or complex scalar
$\boldsymbol{a}$	A real or complex column vector of dimension $m \times 1$
$\boldsymbol{A}$	A real or complex matrix of dimension $m \times n$
$\boldsymbol{A}^T$	The transpose of a matrix
$\boldsymbol{A}^{-1}$	The inverse of a square matrix
$\boldsymbol{A}^+$	The pseudo-inverse of a matrix
$\lvert\boldsymbol{a}\rvert$	L_1-norm of a vector $\boldsymbol{a}$
$\lVert\boldsymbol{a}\rVert$	L_2-norm of a vector $\boldsymbol{a}$
$[\boldsymbol{a}]_\times$	Skew-symmetric cross-product matrix
$SE(3)$	The group of 3D Euclidean transformations
$SO(3)$	The group of 3D similarity transformations
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2

Bibliography

[1] L. Agapito, E. Hayman, and I. Reid. "Self-Calibration of a Rotating Camera with Varying Intrinsic Parameters". In: *British Machine Vision Conference (BMVC)*. 1998.

[2] S. Agarwal, N. Snavely, and S. Seitz. "Fast algorithms for L_∞ problems in multiview geometry". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[3] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. "Building rome in a day". In: *International Conference on Computer Vision (ICCV)*. 2009.

[4] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. "Building rome in a day". In: *Communications of the ACM (CACM)* (2011).

[5] S. Agarwal, K. Mierle, and Others. *Ceres Solver*. `http://ceres-solver.org`.

[6] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski. "Bundle adjustment in the large". In: *European Conference on Computer Vision (ECCV)*. 2010.

[7] Agisoft. *PhotoScan*. `http://www.agisoft.com/`.

[8] C. Aholt, S. Agarwal, and R. Thomas. "A QCQP Approach to Triangulation". In: *European Conference on Computer Vision (ECCV)*. 2012.

[9] A. Alahi, R. Ortiz, and P. Vandergheynst. "FREAK: Fast Retina Keypoint". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[10] Y. Amit and D. G. Y. "Shape quantization and recognition with randomized trees". In: *Neural Computation* (1997).

[11] O. M. Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. "Learning a confidence measure for optical flow". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2013).

[12] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. "NetVLAD: CNN architecture for weakly supervised place recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[13] R. Arandjelović and A. Zisserman. "All about VLAD". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[14] R. Arandjelović and A. Zisserman. "DisLocation: Scalable descriptor distinctiveness for location recognition ". In: *Asian Conference on Computer Vision (ACCV)*. 2014.

[15] R. Arandjelović and A. Zisserman. "Three things everyone should know to improve object retrieval". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[16] R. Arandjelović and A. Zisserman. "Visual Vocabulary with a Semantic Twist". In: *Asian Conference on Computer Vision (ACCV)*. 2014.

[17] S. Ardeshir, A. R. Zamir, A. Torroella, and M. Shah. "GIS-Assisted Object Detection and Geospatial Localization". In: *European Conference on Computer Vision (ECCV)*. 2014.

[18] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas. "Localization from semantic observations via the matrix permanent". In: *International Journal of Robotics Research (IJRR)* (2016).

[19] Y. Avrithis and G. Tolias. "Hough Pyramid Matching: Speeded-up geometry re-ranking for large scale image retrieval". In: *International Journal of Computer Vision (IJCV)* (2014).

[20] A. Azarbayejani, B. Horowitz, and A. Pentland. "Recursive estimation of structure and motion using relative orientation constraints". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 1993.

[21] W. Baarda. *Statistical concepts in geodesy*. 1967.

[22] C. Bailer, M. Finckh, and H. P. Lensch. "Scale robust multi view stereo". In: *European Conference on Computer Vision (ECCV)*. 2012.

[23] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[24] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. "Learning local feature descriptors with triplets and shallow convolutional neural networks". In: *British Machine Vision Conference (BMVC)*. 2016.

[25] C. Banz, H. Blume, and P. Pirsch. "Real-time semi-global matching disparity estimation on the GPU". In: *Mobile Vision Workshop, ICCV*. 2011.

[26] H. Bay, T. Tuytelaars, and L. V. Gool. "SURF: Speeded Up Robust Features". In: *European Conference on Computer Vision (ECCV)*. 2006.

[27] P. Beardsley, P. Torr, and A. Zisserman. "3D model acquisition from extended image sequences". In: *European Conference on Computer Vision (ECCV)*. 1996.

[28] P. A. Beardsley, A. Zisserman, and D. W. Murray. "Sequential updating of projective and affine structure from motion". In: *International Journal of Computer Vision (IJCV)* (1997).

[29] C. Beder and R. Steffen. "Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence". In: *Pattern Recognition* (2006).

[30] P. J. Besl and H. D. McKay. "A method for registration of 3D shapes". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (1992).

[31] S. Birchfield and C. Tomasi. "Multiway cut for stereo and motion with slanted surfaces". In: *International Conference on Computer Vision (ICCV)*. 1999.

[32] C. M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

[33] M. Bleyer and M. Gelautz. "Simple but Effective Tree Structures for Dynamic Programming-Based Stereo Matching". In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. 2008.

[34] M. Bleyer, C. Rhemann, and C. Rother. "PatchMatch Stereo-Stereo Matching with Slanted Support Windows". In: *British Machine Vision Conference (BMVC)*. 2011.

[35] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. "Object stereo—joint stereo matching and object segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[36] S. Bougnoux. "From projective to euclidean space under any practical situation, a criticism of self-calibration". In: *International Conference on Computer Vision (ICCV)*. 1998.

[37] Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2004).

[38] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. "Learning 6D Object Pose Estimation Using 3D Object Coordinates". In: *European Conference on Computer Vision (ECCV)*. 2014.

[39] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. "DSAC-Differentiable RANSAC for Camera Localization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[40] L. Breiman. "Random Forests". In: *Machine Learning* (2001).

[41] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. "Signature Verification using a Siamese Time Delay Neural Network". In: *Conference on Neural Information Processing Systems (NIPS)*. 1994.

[42] C. Bron and J. Kerbosch. "Algorithm 457: Finding All Cliques of an Undirected Graph". In: *Communications of the ACM (CACM)* (1973).

[43] D. C. Brown. *A solution to the general problem of multiple station analytical stereo triangulation*. 1958.

[44] M. Brown, G. Hua, and S. Winder. "Discriminative Learning of Local Image Descriptors". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2011).

[45] M. A. Brubaker, A. Geiger, and R. Urtasun. "Map-Based Probabilistic Visual Self-Localization". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2016).

[46] M. Bujnak, Z. Kukelova, and T. Pajdla. "A general solution to the P4P problem for camera with unknown focal length". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[47] A. Bursuc, G. Tolias, and H. Jégou. "Kernel local descriptors with implicit rotation matching". In: *International Conference on Multimedia Retrieval (ICMR)*. 2015.

[48] P. Burt, L. Wixson, and G. Salgian. "Electronically directed focal stereo". In: *International Conference on Computer Vision (ICCV)*. 1995.

[49] N. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. "Using multiple hypotheses to improve depth-maps for multi-view stereo". In: *European Conference on Computer Vision (ECCV)*. 2008.

[50] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler. "Hybrid Camera Pose Estimation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[51] F. Camposeco, T. Sattler, A. Cohen, A. Geiger, and M. Pollefeys. "Toroidal Constraints for Two Point Localization Under High Outlier Ratios". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[52] S. Cao and N. Snavely. "Learning to Match Images in Large-Scale Collections". In: *European Conference on Computer Vision (ECCV)*. 2012.

[53] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. "University of Michigan North Campus long-term vision and lidar dataset". In: *International Journal of Robotics Research (IJRR)* (2015).

[54] L. Carlone, P. F. Alcantarilla, H.-P. Chiu, Z. Kira, and F. Dellaert. "Mining Structure Fragments for Smart Bundle Adjustment". In: *British Machine Vision Conference (BMVC)*. 2014.

[55] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese. "Semantic cross-view matching". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[56] F. Cazals and C. Karande. "A note on the problem of reporting maximal cliques". In: *Theoretical Computer Science* (2008).

[57] S. E. Chen and L. Williams. "View interpolation for image synthesis". In: *Conference on Computer Graphics and Interactive Techniques*. 1993.

[58] S. Chen, Y. F. Li, J. Zhang, and W. Wang. *Active Sensor Planning for Multiview Vision Tasks*. 2008.

[59] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate". In: *Transactions on Mathematical Software (TOMS)* (2008).

[60] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford. "Deep Learning Features at Scale for Visual Place Recognition". In: *International Conference on Robotics and Automation (ICRA)*. 2017.

[61] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. "A deep visual correspondence embedding model for stereo matching costs". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[62] I. Cherabier*, J. L. Schönberger*, M. Oswald, M. Pollefeys, and A. Geiger. "Learning Priors for Semantic 3D Reconstruction". In: *European Conference on Computer Vision (ECCV). *Equal contribution*. 2018.

[63] O. Chum, J. Matas, and J. Kittler. "Locally Optimized RANSAC". In: *German Conference on Pattern Recognition (GCPR)*. 2003.

[64] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. "Total Recall II: Query Expansion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[65] O. Chum, M. Perdoch, and J. Matas. "Geometric min-Hashing: Finding a (Thick) Needle in a Haystack". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[66] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval". In: *International Conference on Computer Vision (ICCV)*. 2007.

[67] O. Chum and J. Matas. "Large-Scale Discovery of Spatially Related Images". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2010).

[68] O. Chum and J. Matas. "Matching with PROSAC-progressive sample consensus". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[69] O. Chum, J. Matas, and J. Kittler. "Locally optimized RANSAC". In: *Joint Pattern Recognition Symposium*. 2003.

[70] O. Chum, J. Matas, and S. Obdrzalek. "Enhancing RANSAC by generalized model optimization". In: *Asian Conference on Computer Vision (ACCV)*. 2004.

[71] O. Chum, T. Pajdla, and P. Sturm. "The geometric error for homographies". In: *Computer Vision and Image Understanding (CVIU)* (2005).

[72] A. Cohen, T. Sattler, and M. Pollefeys. "Merging the Unmatchable: Stitching Visually Disconnected SfM Models". In: *International Conference on Computer Vision (ICCV)*. 2015.

[73] A. Cohen*, J. L. Schönberger*, P. Speciale, T. Sattler, J. Frahm, and M. Pollefeys. "Indoor-Outdoor 3D Reconstruction Alignment". In: *European Conference on Computer Vision (ECCV). *Equal contribution*. 2016.

[74] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[75] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. "Discrete-Continuous Optimization for Large-Scale Structure from Motion". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[76] N. Cui, J. Weng, and P. Cohen. "Extended structure and motion analysis from monocular image sequences". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 1990.

[77] M. Cummins and P. Newman. "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance". In: *International Journal of Robotics Research (IJRR)* (2008).

[78] P. E. Debevec, C. J. Taylor, and J. Malik. "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach". In: 1996.

[79] F. Dellaert, S. Seitz, C. E. Thorpe, and S. Thrun. "Structure from motion without correspondence". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2000.

[80] J. Dong and S. Soatto. "Domain-size pooling in local descriptors: DSP-SIFT". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[81] A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht. "Semi-Global Matching: A Principled Derivation in Terms of Message Passing". In: *German Conference on Pattern Recognition (GCPR)*. 2014.

[82] R. Dubé, D. Dugas, E. Stumm, J. I. Nieto, R. Siegwart, and C. Cadena. "SegMatch: Segment based loop-closure for 3D point clouds". In: *International Conference on Robotics and Automation (ICRA)*. 2017.

[83] E. Dunn and J.-M. Frahm. "Next best view planning for active model improvement". In: *British Machine Vision Conference (BMVC)*. 2009.

[84] G. Facciolo, C. D. Franchis, and E. Meinhardt. "MGM: A Significantly More Global Matching for Stereovision". In: *British Machine Vision Conference (BMVC)*. 2015.

[85] M. A. Fischler and R. C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM (CACM)* (1981).

[86] A. Fitzgibbon and A. Zisserman. "Automatic camera recovery for closed or open image sequences". In: *European Conference on Computer Vision (ECCV)*. 1998.

[87] C. Forster, M. Pizzoli, and D. Scaramuzza. "Air-ground localization and map augmentation using monocular dense reconstruction". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2013.

[88] J.-M. Frahm and M. Pollefeys. "RANSAC for quasi-degenerate data (QDEGSAC)". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[89] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. "Building Rome on a cloudless day". In: *European Conference on Computer Vision (ECCV)*. 2010.

[90] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. G. Herrtwich. "Making Bertha See". In: *International Conference on Computer Vision (ICCV) Workshops*. 2013.

[91] S. Fuhrmann, F. Langguth, and M. Goesele. "MVE-A Multi-View Reconstruction Environment". In:

[92] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. "Reconstructing building interiors from images". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[93] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. "Towards internet-scale multi-view stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[94] Y. Furukawa, C. Hernández, et al. "Multi-view stereo: A tutorial". In: *Foundations and Trends in Computer Graphics and Vision* (2015).

[95] Y. Furukawa and J. Ponce. "Accurate, dense, and robust multiview stereopsis". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2010).

[96] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. "Manhattan-world stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[97] S. Galliani, K. Lasinger, and K. Schindler. "Massively Parallel Multiview Stereopsis by Surface Normal Diffusion". In: *International Conference on Computer Vision (ICCV)*. 2015.

[98] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. "Variable baseline/resolution stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[99] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. "Real-time plane-sweeping stereo with multiple sweeping directions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[100] D. Gallup, M. Pollefeys, and J.-M. Frahm. "3d reconstruction using an n-layer heightmap". In: *Joint Pattern Recognition Symposium*. 2010.

[101] S. Gammeter, T. Quack, and L. Van Gool. "I Know What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps". In: *International Conference on Computer Vision (ICCV)*. 2009.

[102] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. "Complete solution classification for the perspective-three-point problem". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2003).

[103] S. Garg, N. Suenderhauf, and M. Milford. "Don't Look Back: Robustifying Place Categorization for Viewpoint-and Condition-Invariant Place Recognition". In: *International Conference on Robotics and Automation (ICRA)*. 2018.

[104] S. Garg, N. Suenderhauf, and M. Milford. "LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics". In: *Robotics: Science and Systems (RSS)*. 2018.

[105] S. Gehrig, F. Eberli, and T. Meyer. "A real-time low-power stereo vision engine using semi-global matching". In: *Computer Vision Systems*. 2009.

[106] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[107] J.-M. Geusebroek, R. V. den Boomgaard, A. W. Smeulders, and H. Geerts. "Color invariance". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2001).

[108] R. Gherardi, M. Farenzena, and A. Fusiello. "Improving the efficiency of hierarchical structure-and-motion". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[109] S. Gidaris and N. Komodakis. "Detect, replace, refine: Deep structured prediction for pixel wise labeling". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[110] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. "Multi-view stereo for community photo collections". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[111] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. "Deep Image Retrieval: Learning global representations for image search". In: *arXiv* (2016).

[112] K. Grauman and T. Darrell. "The pyramid match kernel: discriminative classification with sets of image features". In: *International Conference on Computer Vision (ICCV)*. 2005.

[113] R. Hadsell, S. Chopra, and Y. LeCun. "Dimensionality reduction by learning an invariant mapping". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[114] R. Haeusler, R. Nair, and D. Kondermann. "Ensemble learning for confidence measures in stereo vision". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[115] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. "MatchNet: Unifying feature and metric learning for patch-based matching". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[116] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. "Joint 3D scene reconstruction and class segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[117] S. Haner and A. Heyden. "Covariance Propagation and Next Best View Planning for 3D Reconstruction". In: *European Conference on Computer Vision (ECCV)*. 2012.

[118] Z. S. Harris. "Distributional structure". In: *Word* (1954).

[119] R. Hartley. "In defense of the eight-point algorithm". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (1997).

[120] R. Hartley. "Lines and points in three views - an integrated approach". In: *ARPA Image Understanding Workshop*. 1994.

[121] R. Hartley and F. Schaffalitzky. "L_∞ minimization in geometric reconstruction problems". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004.

[122] R. Hartley and P. Sturm. "Triangulation". In: *Computer Vision and Image Understanding (CVIU)* (1997).

[123] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. 2003.

[124] R. I. Hartley. "Cheirality Invariants". In: *DARPA Image Understanding Workshop*. 1993.

[125] W. Hartmann, M. Havlena, and K. Schindler. "Predicting matchability". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[126] M. Havlena and K. Schindler. "VocMatch: Efficient Multiview Correspondence for Structure from Motion". In: *European Conference on Computer Vision (ECCV)*. 2014.

[127] J. Heinly, E. Dunn, and J.-M. Frahm. "Comparative evaluation of binary features". In: *European Conference on Computer Vision (ECCV)*.

[128] J. Heinly, E. Dunn, and J.-M. Frahm. "Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction". In: *European Conference on Computer Vision (ECCV)*. 2014.

[129] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. "Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset)". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[130] P. Heise, B. Jensen, S. Klose, and A. Knoll. "Variational PatchMatch Multi-View Reconstruction and Refinement". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[131] S. Hermann and R. Klette. "Iterative semi-global matching for robust driver assistance systems". In: *Asian Conference on Computer Vision (ACCV)*. 2012.

[132] S. Hermann, R. Klette, and E. Destefanis. "Inclusion of a second-order prior into semi-global matching". In: *Pacific-Rim Symposium on Image and Video Technology*. 2009.

[133] H. C. L. Higgins. "A Computer Algorithm for Reconstructing a Scene from Two Projections". In: *Nature* (1981).

[134] H. Hirschmüller. "Accurate and efficient stereo processing by semi-global matching and mutual information". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[135] H. Hirschmüller. "Stereo Processing by Semi-Global Matching and Mutual Information". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2008).

[136] H. Hirschmuller. "Stereo vision in structured environments by consistent semi-global matching". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[137] H. Hirschmüller, M. Buder, and I. Ernst. "Memory Efficient Semi-Global Matching". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2012.

[138] H. Hirschmüller and D. Scharstein. "Evaluation of stereo matching costs on images with radiometric differences". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2009).

[139] T. K. Ho. "The Random Subspace Method for Constructing Decision Forests". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (1998).

[140] E. Hoffer and N. Ailon. "Deep metric learning using triplet network". In: *International Workshop on Similarity-Based Pattern Recognition*. 2015.

[141] D. Honegger, H. Oleynikova, and M. Pollefeys. "Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2014.

[142] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* (2013).

[143] X. Hu and P. Mordohai. "A quantitative evaluation of confidence measures for stereo vision". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2012).

[144] X. Hu and P. Mordohai. "Least Commitment, Viewpoint-Based, Multi-view Stereo". In: *International Conference on 3D Vision (3DV)*. 2012.

[145] S. S. Intille and A. F. Bobick. *Disparity-space images and large occlusion stereo*. 1994.

[146] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. "From structure-from-motion point clouds to fast location recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[147] M. Jancosek and T. Pajdla. "Multi-view reconstruction preserving weakly-supported surfaces". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[148] H. Jégou, M. Douze, and C. Schmid. "Hamming embedding and weak geometric consistency for large scale image search". In: *European Conference on Computer Vision (ECCV)*. 2008.

[149] H. Jégou, M. Douze, and C. Schmid. "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search". In: *European Conference on Computer Vision (ECCV)*. 2008.

[150] H. Jégou, M. Douze, and C. Schmid. "On the burstiness of visual elements". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[151] H. Jégou, M. Douze, C. Schmid, and P. Pérez. "Aggregating local descriptors into a compact image representation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[152] H. Jégou and A. Zisserman. "Triangulation embedding and democratic aggregation for image search". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[153] H. Jégou, M. Douze, and C. Schmid. "Improving Bag-of-Features for Large Scale Image Search". In: *International Journal of Computer Vision (IJCV)* (2010).

[154] D. Ji, E. Dunn, and J.-M. Frahm. "3D reconstruction of dynamic textures in crowd sourced data". In: *European Conference on Computer Vision (ECCV)*. 2014.

[155] D. Ji, E. Dunn, and J.-M. Frahm. "Synthesizing Illumination Mosaics from Internet Photo-Collections". In: *International Conference on Computer Vision (ICCV)*. 2015.

[156] E. D. Johns and G.-Z. Yang. "Pairwise Probabilistic Voting: Fast Place Recognition without RANSAC". In: *European Conference on Computer Vision (ECCV)*. 2014.

[157] T. Kanade and M. Okutomi. "A stereo matching algorithm with an adaptive window: Theory and experiment". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (1994).

[158] L. Kang, L. Wu, and Y.-H. Yang. "Robust multi-view L_2 triangulation via optimal inlier selection and 3D structure refinement". In: *Pattern Recognition* (2014).

[159] S. B. Kang, R. Szeliski, and J. Chai. "Handling occlusions in dense multi-view stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2001.

[160] M. Kazhdan and H. Hoppe. "Screened poisson surface reconstruction". In: *Transactions on Graphics (TOG)* (2013).

[161] Y. Ke and R. Sukthankar. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004.

[162] A. Kendall, M. Grimes, and R. Cipolla. "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization". In: *International Conference on Computer Vision (ICCV)*. 2015.

[163] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. "End-to-end learning of geometry and context for deep stereo regression". In: *International Conference on Computer Vision (ICCV)*. 2017.

[164] M. Khoury, Q.-Y. Zhou, and V. Koltun. "Learning Compact Geometric Features". In: *International Conference on Computer Vision (ICCV)*. 2017.

[165] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *arXiv* (2013).

[166] G. Klein and D. Murray. "Parallel Tracking and Mapping on a Camera Phone". In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. 2009.

[167] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. "Tanks and temples: Benchmarking large-scale scene reconstruction". In: *Transactions on Graphics (TOG)* (2017).

[168] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock. "End-to-end training of hybrid CNN-CRF models for stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[169] J. Knopp, J. Sivic, and T. Pajdla. "Avoiding confusing features in place recognition". In: *European Conference on Computer Vision (ECCV)*. 2010.

[170] N. Kobyshev, H. Riemenschneider, and L. V. Gool. "Matching Features Correctly through Semantic Understanding". In: *International Conference on 3D Vision (3DV)*. 2014.

[171] J. Krapac, J. Verbeek, and F. Jurie. "Modeling spatial layout with fisher vectors for image categorization". In: *International Conference on Computer Vision (ICCV)*. 2011.

[172] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012.

[173] Z. Kukelova. "Algebraic methods in computer vision". PhD thesis. 2013.

[174] Z. Kukelova, M. Bujnak, and T. Pajdla. "Automatic generator of minimal problem solvers". In: *European Conference on Computer Vision (ECCV)*. 2008.

[175] Z. Kukelova, M. Bujnak, and T. Pajdla. "Real-time solution to the absolute pose problem with unknown radial distortion and focal length". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[176] Z. Kukelova, J. Heller, M. Bujnak, and T. Pajdla. "Radial distortion homography". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[177] A. Kushal and S. Agarwal. "Visibility based preconditioning for bundle adjustment". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[178] M. Kwang, E. Trulls, V. Lepetit, and P. Fua. "LIFT: Learned Invariant Feature Transform". In: *European Conference on Computer Vision (ECCV)*. 2016.

[179] V. Larsson, M. Oskarsson, K. Åström, A. Wallis, Z. Kukelova, and T. Pajdla. "Beyond Gröbner Bases: Basis Selection for Minimal Solvers". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[180] K. Lebeda, J. Matas, and O. Chum. "Fixing the Locally Optimized RANSAC". In: *British Machine Vision Conference (BMVC)*. 2012.

[181] G. H. Lee, F. Fraundorfer, and M. Pollefeys. "Structureless Pose-Graph Loop-Closure with a Multi-Camera System on a Self-Driving Car". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2013.

[182] Y. Lee, M.-G. Park, Y. Hwang, Y. Shin, and C.-M. Kyung. "Memory-Efficient Parametric Semiglobal Matching". In: *Signal Processing Letters* (2018).

[183] V. Lempitsky, C. Rother, S. Roth, and A. Blake. "Fusion moves for markov random field optimization". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2010).

[184] V. Lepetit and P. Fua. "Keypoint recognition using randomized trees". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2006).

[185] V. Lepetit, F. Moreno-Noguer, and P. Fua. "EPnP: An accurate O(n) solution to the PnP problem". In: *International Journal of Computer Vision (IJCV)* (2009).

[186] S. Leutenegger, M. Chli, and R. Y. Siegwart. "BRISK: Binary Robust invariant scalable keypoints". In: *International Conference on Computer Vision (ICCV)*. 2011.

[187] S. Leutenegger, P. T. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. "Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization". In: *Robotics: Science and Systems (RSS)*. 2013.

[188] K. Levenberg. "A method for the solution of certain non-linear problems in least squares". In: *Quarterly of Applied Mathematics* (1944).

[189] H. Li. "A practical algorithm for L_∞ triangulation with outliers". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[190] X. Li, M. Larson, and A. Hanjalic. "Pairwise Geometric Matching for Large-Scale Object Retrieval". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[191] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. "Modeling and recognition of landmark image collections using iconic scene graphs". In: *European Conference on Computer Vision (ECCV)*. 2008.

[192] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. "Worldwide Pose Estimation using 3D Point Clouds". In: *European Conference on Computer Vision (ECCV)*. 2012.

[193] Y. Li, N. Snavely, and D. P. Huttenlocher. "Location recognition using prioritized feature matching". In: *European Conference on Computer Vision (ECCV)*. 2010.

[194] N. Lianos, J. L. Schönberger, M. Pollefeys, and T. Sattler. "VSO: Visual Semantic Odometry". In: *European Conference on Computer Vision (ECCV)*. 2018.

[195] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. "Real-Time Image-Based 6-DOF Localization in Large-Scale Environments". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[196] T.-Y. Lin, S. Belongie, and J. Hays. "Cross-view image geolocalization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[197] C. Linegar, W. Churchill, and P. Newman. "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera". In: *International Conference on Robotics and Automation (ICRA)*. 2016.

[198] Y. Lou, N. Snavely, and J. Gehrke. "MatchMiner: Efficient Spanning Structure Mining in Large Image Collections". In: *European Conference on Computer Vision (ECCV)*. 2012.

[199] M. I. Lourakis and A. A. Argyros. "SBA: A software package for generic sparse bundle adjustment". In: *Transactions on Mathematical Software (TOMS)*. 2009.

[200] D. Lowe. "Distinctive image features from scale-invariant keypoints". In: (2004).

[201] F. Lu and R. Hartley. "A fast optimal algorithm for L_2 triangulation". In: *Asian Conference on Computer Vision (ACCV)*. 2007.

[202] W. Luo, A. G. Schwing, and R. Urtasun. "Efficient deep learning for stereo matching". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[203] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. "Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization". In: *Robotics: Science and Systems (RSS)*. 2015.

[204] D. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the society for Industrial and Applied Mathematics* (1963).

[205] R. Martin-Brualla, D. Gallup, and S. M. Seitz. "3D time-lapse reconstruction from internet photos". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[206] R. Martin-Brualla, D. Gallup, and S. M. Seitz. "Time-lapse mining from internet photos". In: 2015.

[207] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions". In: *BMCV* (2004).

[208] J. Matas and O. Chum. "Randomized RANSAC with sequential probability ratio test". In: *International Conference on Computer Vision (ICCV)*. 2005.

[209] K. Matzen and N. Snavely. "Scene chronology". In: *European Conference on Computer Vision (ECCV)*. 2014.

[210] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[211] C. McGlone, E. Mikhail, and J. Bethel. *Manual of photogrammetry*. 1980.

[212] G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. V. Gool. "A hierarchical symmetric stereo algorithm using dynamic programming". In: *International Journal of Computer Vision (IJCV)* (2002).

[213] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys. "Real-time visibility-based fusion of depth maps". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[214] M. Michael, J. Salmen, J. Stallkamp, and M. Schlipsing. "Real-time stereo vision: Optimizing semi-global matching". In: *Intelligent Vehicles Symposium (IV)*. 2013.

[215] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2005).

[216] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. "A comparison of affine region detectors". In: *International Journal of Computer Vision (IJCV)* (2005).

[217] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. "Learning Vocabularies over a Fine Quantization". In: *International Journal of Computer Vision (IJCV)* (2013).

[218] A. Mikulík, F. Radenović, O. Chum, and J. Matas. "Efficient Image Detail Mining". In: *Asian Conference on Computer Vision (ACCV)*. 2014.

[219] A. Mikulík, O. Chum, and J. Matas. "Image Retrieval for Online Browsing in Large Image Collections". In: *International Conference on Similarity Search and Applications (SISAP)*. 2013.

[220] A. Mikulík, F. Radenović, O. Chum, and J. Matas. "Efficient image detail mining". In: *Asian Conference on Computer Vision (ACCV)*. 2014.

[221] H. Mobahi, R. Collobert, and J. Weston. "Deep learning from temporal coherence in video". In: *ICML* (2009).

[222] R. Mohr, L. Quan, and F. Veillon. "Relative 3D reconstruction using multiple uncalibrated images". In: *International Journal of Robotics Research (IJRR)* (1995).

[223] P. Moulon, P. Monasse, and R. Marlet. "Global fusion of relative motions for robust, accurate and scalable structure from motion". In: *International Conference on Computer Vision (ICCV)*. 2013.

[224] P. Moulon, P. Monasse, R. Marlet, and Others. *OpenMVG – An Open Multiple View Geometry library*. `https://github.com/openMVG/openMVG`.

[225] A. Mousavian, J. Košecká, and J.-M. Lien. "Semantically Guided Location Recognition for Outdoors Scenes". In: *International Conference on Robotics and Automation (ICRA)*. 2015.

[226] M. Muja and D. G. Lowe. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration". In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. 2009.

[227] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *Transactions on Robotics (TRO)* (2015).

[228] R. M. Neal and G. E. Hinton. "A view of the EM algorithm that justifies incremental, sparse, and other variants". In: *Learning in Graphical Models*. 1998.

[229] K. Ni, D. Steedly, and F. Dellaert. "Out-of-core bundle adjustment for large-scale 3D reconstruction". In: *International Conference on Computer Vision (ICCV)*. 2007.

[230] D. Nister. "An efficient solution to the five-point relative pose problem". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2004).

[231] D. Nister and H. Stewénius. "Scalable Recognition with a Vocabulary Tree". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[232] J. Nocedal and S. Wright. *Sequential quadratic programming*. 2006.

[233] Y. Ohta and T. Kanade. "Stereo by intra-and inter-scanline search using dynamic programming". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (1985).

[234] A. Oliva and A. Torralba. "Modeling the shape of the scene: A holistic representation of the spatial envelope". In: *International Journal of Computer Vision (IJCV)* (2001).

[235] C. Olsson, A. Eriksson, and R. Hartley. "Outlier removal using duality". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[236] M. Oswald and D. Cremers. "A convex relaxation approach to space time multi-view 3d reconstruction". In: *International Conference on Computer Vision (ICCV) Workshops*. 2013.

[237] J. Pang, W. Sun, J. Ren, C. Yang, and Q. Yan. "Cascade residual learning: A two-stage convolutional neural network for stereo matching". In: *International Conference on Computer Vision (ICCV) Workshops*. 2017.

[238] M.-G. Park and K.-J. Yoon. "Leveraging stereo matching with learning-based confidence measures". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[239] M. Perdoch, O. Chum, and J. Matas. "Efficient representation of local geometry for large scale object retrieval". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[240] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. "Large-Scale Image Retrieval with Compressed Fisher Vectors". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[241] F. Perronnin and C. Dance. "Fisher kernels on visual vocabularies for image categorization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[242] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. "Lost in quantization: Improving particular object retrieval in large scale image databases". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[243] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. "Object retrieval with large vocabularies and fast spatial matching". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[244] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. "Descriptor Learning for Efficient Retrieval". In: *European Conference on Computer Vision (ECCV)*. 2010.

[245] Pix4D. *Pix4Dmapper*. `https://pix4d.com/`.

[246] M. Poggi and S. Mattoccia. "Deep stereo fusion: combining multiple disparity hypotheses with deep-learning". In: *International Conference on 3D Vision (3DV)*. 2016.

[247] M. Poggi and S. Mattoccia. "Learning a general-purpose confidence measure based on o (1) features and a smarter aggregation strategy for semi global matching". In: *International Conference on 3D Vision (3DV)*. 2016.

[248] M. Poggi and S. Mattoccia. "Learning to predict stereo reliability enforcing local consistency of confidence maps". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[249] M. Pollefeys. "Self-calibration and metric 3D reconstruction from uncalibrated image sequences". PhD thesis. 1999.

[250] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. "Visual modeling with a hand-held camera". In: *International Journal of Computer Vision (IJCV)* (2004).

[251] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. "Detailed real-time urban 3D reconstruction from video". In: *International Journal of Computer Vision (IJCV)* (2008).

[252] F. Radenović, J. L. Schönberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas. "From Dusk till Dawn: Modeling in the Dark". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[253] F. Radenović, G. Tolias, and O. Chum. "CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples". In: *European Conference on Computer Vision (ECCV)*. 2016.

[254] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm. "USAC: A universal framework for random sample consensus". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2013).

[255] R. Raguram, J.-M. Frahm, and M. Pollefeys. "ARRSAC: Adaptive Real-Time Random Sample Consensus". In: *European Conference on Computer Vision (ECCV)*. 2008.

[256] R. Raguram, J.-M. Frahm, and M. Pollefeys. "Exploiting uncertainty in random sample consensus". In: *International Conference on Computer Vision (ICCV)*. 2009.

[257] R. Raguram, J. Tighe, and J.-M. Frahm. "Improved Geometric Verification for Large Scale Landmark Image Collections". In: *British Machine Vision Conference (BMVC)*. 2012.

[258] R. Raguram, C. Wu, J.-M. Frahm, and S. Lazebnik. "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs". In: *International Journal of Computer Vision (IJCV)* (2011).

[259] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2014.

[260] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. "Fast cost-volume filtering for visual correspondence and beyond". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[261] M. Rothermel, K. Wenzel, D. Fritsch, and N. Haala. "SURE: Photogrammetric surface reconstruction from imagery". In: *Proceedings LC3D Workshop, Berlin*. 2012.

[262] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *International Conference on Computer Vision (ICCV)*. 2011.

[263] R. B. Rusu, N. Blodow, and M. Beetz. "Fast Point Feature Histograms (FPFH) for 3D registration". In: *International Conference on Robotics and Automation (ICRA)*. 2009.

[264] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[265] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. "Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition". In: *International Conference on Computer Vision (ICCV)*. 2015.

[266] T. Sattler, B. Leibe, and L. Kobbelt. "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2017).

[267] T. Sattler, B. Leibe, and L. Kobbelt. "SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter". In: *International Conference on Computer Vision (ICCV)*. 2009.

[268] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[269] T. Sattler, C. Sweeney, and M. Pollefeys. "On Sampling Focal Length Values to Solve the Absolute Pose Problem". In: *European Conference on Computer Vision (ECCV)*. 2014.

[270] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. "Large-Scale Location Recognition And The Geometric Burstiness Problem". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[271] T. Sattler, B. Leibe, and L. Kobbelt. "Fast image-based localization using direct 2d-to-3d matching". In: *International Conference on Computer Vision (ICCV)*. 2011.

[272] T. Sattler, C. Sweeney, and M. Pollefeys. "On Sampling Focal Length Values to Solve the Absolute Pose Problem". In: *European Conference on Computer Vision (ECCV)*. 2014.

[273] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. "Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?" In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[274] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. "Image Retrieval for Image-Based Localization Revisited". In: *British Machine Vision Conference (BMVC)*. 2012.

[275] F. Schaffalitzky and A. Zisserman. "Multi-view matching for unordered image sets, or How do I organize my holiday snaps?" In: *European Conference on Computer Vision (ECCV)*. 2002.

[276] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling. "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth". In: *German Conference on Pattern Recognition (GCPR)*.

[277] D. Scharstein and C. Pal. "Learning conditional random fields for stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[278] D. Scharstein and R. Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *International Journal of Computer Vision (IJCV)*. 2002.

[279] D. Scharstein, T. Taniai, and S. Sinha. "Semi-Global Stereo Matching with Surface Orientation Priors". In: *International Conference on 3D Vision (3DV)*. 2017.

[280] G. Schindler and F. Dellaert. "Probabilistic temporal inference on reconstructed 3D scenes". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[281] K. Schmid, T. Tomic, F. Ruess, H. Hirschmüller, and M. Suppa. "Stereo vision based indoor/outdoor navigation for flying robots". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2013.

[282] J. L. Schönberger, A. C. Berg, and J.-M. Frahm. "Efficient Two-View Geometry Classification". In: *German Conference on Pattern Recognition (GCPR)*. 2015.

[283] J. L. Schönberger, A. C. Berg, and J.-M. Frahm. "PAIGE: PAirwise Image Geometry Encoding for Improved Efficiency in Structure-from-Motion". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[284] J. L. Schönberger and J.-M. Frahm. "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[285] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. "Comparative Evaluation of Hand-Crafted and Learned Local Features". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[286] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. "Semantic Visual Localization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[287] J. L. Schönberger*, T. Price*, T. Sattler, J.-M. Frahm, and M. Pollefeys. "A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval". In: *Asian Conference on Computer Vision (ACCV). *Equal contribution*. 2016.

[288] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm. "From Single Image Query to Detailed 3D Reconstruction". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[289] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*. 2016.

[290] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[291] M. Schreiber, C. Knöppel, and U. Franke. "LaneLoc: Lane marking based localization using highly accurate maps". In: *Intelligent Vehicles Symposium (IV)*. 2013.

[292] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. "A comparison and evaluation of multi-view stereo reconstruction algorithms". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[293] A. Seki and M. Pollefeys. "SGM-Nets: Semi-Global Matching With Neural Networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[294] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M. Seitz. "The visual turing test for scene reconstruction". In: *International Conference on 3D Vision (3DV)*. 2013.

[295] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. "Occluding contours for multi-view stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[296] A Shashua. "On the trilinear tensor of three perspective views and its underlying geometry". In: *International Conference on Computer Vision (ICCV)*. 1995.

[297] X. Shen, Z. Lin, J. Brandt, and Y. Wu. "Spatially-constrained similarity measure for large-scale object retrieval". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2014).

[298] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[299] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. "Real-time human pose recognition in parts from single depth images". In: *Communications of the ACM (CACM)*. 2013.

[300] H.-Y. Shum, M. Han, and R. Szeliski. "Interactive construction of 3D models from panoramic mosaics". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 1998.

[301] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. "Discriminative learning of deep convolutional feature point descriptors". In: *International Conference on Computer Vision (ICCV)*. 2015.

[302] K. Simonyan, A. Vedaldi, and A. Zisserman. "Learning local feature descriptors using convex optimisation". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2014).

[303] G. Singh and J. Košecká. "Semantically Guided Geo-location and Modeling in Urban Environments". In: *Large-Scale Visual Geo-Localization*. 2016.

[304] S. N. Sinha, D. Scharstein, and R. Szeliski. "Efficient high-resolution stereo matching using local plane sweeps". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[305] S. N. Sinha, D. Steedly, and R. Szeliski. "A Multi-stage Linear Approach to Structure from Motion". In: *Trends and Topics in Computer Vision*. Ed. by K. N. Kutulakos.

[306] J. Sivic and A. Zisserman. "Efficient Visual Search Cast as Text Retrieval". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2009).

[307] J. Sivic and A. Zisserman. "Video Google: A Text Retrieval Approach to Object Matching in Video". In: *International Conference on Computer Vision (ICCV)*. 2003.

[308] N. Snavely, S. M. Seitz, and R. Szeliski. "Modeling the World from Internet Photo Collections". In: *International Journal of Computer Vision (IJCV)* (2007).

[309] N. Snavely. "Scene reconstruction and visualization from internet photo collections". PhD thesis. 2008.

[310] N. Snavely, S. Seitz, and R. Szeliski. "Photo tourism: exploring photo collections in 3D". In: *Transactions on Graphics (TOG)* (2006).

[311] N. Snavely, S. Seitz, and R. Szeliski. "Skeletal graphs for efficient structure from motion". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[312] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. "Semantic scene completion from a single depth image". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[313] M. E. Spetsakis and J. Y. Aloimonos. "Structure from motion using line correspondences". In: *International Journal of Computer Vision (IJCV)* (1990).

[314] A. Spyropoulos, N. Komodakis, and P. Mordohai. "Learning to detect ground control points for improving the accuracy of stereo matching". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[315] A. Spyropoulos and P. Mordohai. "Ensemble classifier for combining stereo matching algorithms". In: *International Conference on 3D Vision (3DV)*. 2015.

[316] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research (JMLR)* (2014).

[317] C. R. s.r.o. *RealityCapture*. `https://www.capturingreality.com/`.

[318] H. Stewénius, S. H. Gunderson, and J. Pilet. "Size Matters: Exhaustive Geometric Verification for Image Retrieval". In: *European Conference on Computer Vision (ECCV)*. 2012.

[319] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. "LDAHash: Improved Matching with Smaller Descriptors". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2012).

[320] C. Strecha, R. Fransens, and L. V. Gool. "Combined depth and outlier estimation in multi-view stereo". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[321] C. Strecha, R. Fransens, and L. V. Gool. "Wide-baseline stereo from multiple views: a probabilistic account". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004.

[322] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. "On benchmarking camera calibration and multi-view stereo for high resolution imagery". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[323] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. "Symmetric stereo matching for occlusion handling". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[324] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free". In: *Robotics: Science and Systems (RSS)*. 2015.

[325] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. "City-Scale Localization for Cameras with Known Vertical Direction". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2017).

[326] C. Sweeney. *Theia Multiview Geometry Library: Tutorial & Reference*. `http://theia-SFM.org`.

[327] C. Sweeney, V. Fragoso, T. Höllerer, and M. Turk. "gdls: A scalable solution to the generalized pose and scale problem". In: *European Conference on Computer Vision (ECCV)*. 2014.

[328] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys. "Optimizing the Viewing Graph for Structure-from-Motion". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[329] R. Szeliski and S. B. Kang. "Recovering 3D shape and motion from image streams using nonlinear least squares". In: *Journal of Visual Communication and Image Representation* (1994).

[330] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura. "Continuous 3D Label Stereo Matching using Local Expansion Moves". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2017).

[331] T. Taniai, S. N. Sinha, and Y. Sato. "Fast multi-frame stereo scene flow with motion segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[332] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. "YFCC100M: The New Data in Multimedia Research". In: *Communications of the ACM (CACM)* (2016).

[333] C. Toft, C. Olsson, and F. Kahl. "Long-Term 3D Localization and Pose from Semantic Labellings". In: *International Conference on Computer Vision (ICCV)*. 2017.

[334] E. Tola, V. Lepetit, and P. Fua. "Daisy: An efficient dense descriptor applied to wide-baseline stereo". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2010).

[335] G. Tolias, Y. Avrithis, and H. Jégou. "To aggregate or not to aggregate: Selective match kernels for image search". In: *International Conference on Computer Vision (ICCV)*. 2013.

[336] G. Tolias, Y. Kalantidis, Y. Avrithis, and S. Kollias. "Towards large-scale geometry indexing by feature selection". In: *Computer Vision and Image Understanding (CVIU)* (2014).

[337] G. Tolias, Y. Avrithis, and H. Jégou. "Image search with selective match kernels: aggregation across single and multiple images". In: *International Journal of Computer Vision (IJCV)* (2016).

[338] C. Tomasi and T. Kanade. "Shape and motion from image streams under orthography: a factorization method". In: *International Journal of Computer Vision (IJCV)* (1992).

[339] E. Tomita, A. Tanaka, and H. Takahashi. "The worst-case time complexity for generating all maximal cliques and computational experiments". In: *Theoretical Computer Science* (2006).

[340] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. "24/7 place recognition by view synthesis". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[341] P. H. Torr. "An assessment of information criteria for motion model selection". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 1997.

[342] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. Fitzgibbon. "Bundle adjustment – a modern synthesis". In: *International Workshop on Vision Algorithms*. 2000.

[343] T. Trzcinski, M. Christoudias, and V. Lepetit. "Learning Image Descriptors with Boosting". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2015).

[344] T. Tung, S. Nobuhara, and T. Matsuyama. "Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo". In: *International Conference on Computer Vision (ICCV)*. 2009.

[345] T. Tuytelaars and K. Mikolajczyk. "Local invariant feature detectors: a survey". In: *Foundations and Trends in Computer Graphics and Vision* (2008).

[346] R. Tylecek and R. Sara. "Refinement of Surface Mesh for Accurate Multi-View Reconstruction". In: *International Journal of Virtual Reality* (2010).

[347] A. O. Ulusoy, A. Geiger, and M. J. Black. "Towards probabilistic volumetric reconstruction using ray potentials". In: *International Conference on 3D Vision (3DV)*. 2015.

[348] J. P. C. Valentin, M. Nießner, J. Shotton, A. W. Fitzgibbon, S. Izadi, and P. H. S. Torr. "Exploiting uncertainty in regression forests for accurate camera relocalization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[349] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithm*. 2008.

[350] O. Veksler. "Stereo correspondence by dynamic programming on a tree". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[351] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. "TILDE: A Temporally Invariant Learned Detector". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[352] J. Žbontar and Y. LeCun. "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches". In: *Journal of Machine Learning Research (JMLR)* (2016).

[353] M. Waechter, N. Moehrle, and M. Goesele. "Let there be color! Large-scale texturing of 3D reconstructions". In: *European Conference on Computer Vision (ECCV)*. 2014.

[354] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. "Image-based localization using LSTMs for structured feature correlation". In: *International Conference on Computer Vision (ICCV)*. 2017.

[355] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. "Learning Fine-Grained Image Similarity with Deep Ranking". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[356] K. Q. Weinberger and L. K. Saul. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *Journal of Machine Learning Research (JMLR)* (2009).

[357] T. Weyand and B. Leibe. "Discovering Details and Scene Structure with Hierarchical Iconoid Shift". In: *International Conference on Computer Vision (ICCV)*. 2013.

[358] T. Weyand and B. Leibe. "Discovering Favorite Views of Popular Places with Iconoid Shift". In: *International Conference on Computer Vision (ICCV)*. 2011.

[359] T. Weyand, I. Kostrikov, and J. Philbin. "Planet-photo geolocation with convolutional neural networks". In: *European Conference on Computer Vision (ECCV)*. 2016.

[360] T. Weyand, C.-Y. Tsai, and B. Leibe. "Fixing WTFs: Detecting Image Matches Caused by Watermarks, Timestamps, and Frames in Internet Photos". In: *Winter Conf. on Applications of Computer Vision (WACV)*. 2015.

[361] K. Wilson and N. Snavely. "Network Principles for SFM: Disambiguating Repeated Structures with Local Context". In: *International Conference on Computer Vision (ICCV)*. 2013.

[362] K. Wilson and N. Snavely. "Robust Global Translations with 1DSFM". In: *European Conference on Computer Vision (ECCV)*. 2014.

[363] M. Wolff, R. T. Collins, and Y. Liu. "Regularity-Driven Facade Matching Between Aerial and Street Views". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[364] S. Workman, R. Souvenir, and N. Jacobs. "Wide-area image geolocalization with aerial reference imagery". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[365] C. Wu. "Towards linear-time incremental structure from motion". In: *International Conference on 3D Vision (3DV)*. 2013.

[366] C. Wu, S. Agarwal, B. Curless, and S. Seitz. "Multicore bundle adjustment". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[367] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. "3D Model Matching with Viewpoint-Invariant Patches (VIP)". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[368] X. Wu and K. Kashino. "Adaptive Dither Voting for Robust Spatial Verification". In: *International Conference on Computer Vision (ICCV)*. 2015.

[369] X. Wu and K. Kashino. "Robust Spatial Matching as Ensemble of Weak Geometric Relations". In: *British Machine Vision Conference (BMVC)*. 2015.

[370] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger. "Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[371] K. Yamaguchi, D. McAllester, and R. Urtasun. "Efficient joint segmentation, occlusion labeling, stereo and flow estimation". In: *European Conference on Computer Vision (ECCV)*. 2014.

[372] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2009).

[373] K.-J. Yoon and I.-S. Kweon. "Locally adaptive support-weight approach for visual correspondence search". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[374] F. Yu and V. Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *International Conference on Learning Representations (ICLR)*. 2016.

[375] F. Yu, J. Xiao, and T. A. Funkhouser. "Semantic alignment of LiDAR data at city scale". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[376] X. Zabulis and K. Daniilidis. "Multi-camera reconstruction based on surface normal estimation and best viewpoint selection". In: *International Conference on 3D Vision (3DV)*. 2004.

[377] C. Zach. "Fast and high quality fusion of depth maps". In: *International Conference on 3D Vision (3DV)*. 2008.

[378] C. Zach. "Robust bundle adjustment revisited". In: *European Conference on Computer Vision (ECCV)*. 2014.

[379] C. Zach, A. Irschara, and H. Bischof. "What can missing correspondences tell us about 3D structure and motion?" In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2008).

[380] C. Zach, M. Klopschitz, and M. Pollefeys. "Disambiguating visual relations using loop constraints". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[381] C. Zach, M. Sormann, and K. Karner. "Scanline Optimization for Stereo on Graphics Hardware". In: *International Conference on 3D Vision (3DV)*. 2006.

[382] S. Zagoruyko and N. Komodakis. "Learning to compare image patches via convolutional neural networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[383] A. Zaharescu, E. Boyer, and R. Horaud. "Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2011).

[384] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese. "Generic 3D Representation via Pose Estimation and Matching". In: *European Conference on Computer Vision (ECCV)*. 2016.

[385] M. D. Zeiler. "ADADELTA: an adaptive learning rate method". In: *arXiv* (2012).

[386] B. Zeisl, T. Sattler, and M. Pollefeys. "Camera Pose Voting for Large-Scale Image-Based Localization". In: *International Conference on Computer Vision (ICCV)*. 2015.

[387] B. Zeisl, K. Köser, and M. Pollefeys. "Automatic Registration of RGB-D Scans via Salient Directions". In: *International Conference on Computer Vision (ICCV)*. 2013.

[388] C. Zeller and O. Faugeras. "Camera self-calibration from video sequences: the Kruppa equations revisited". PhD thesis. 1996.

[389] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. "3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[390] C. Zhang, J. Gao, O. Wang, P. Georgel, R. Yang, J. Davis, J.-M. Frahm, and M. Pollefeys. "Personal photograph enhancement using Internet photo collections". In: *Transactions on Visualization and Computer Graphics (TVCG)* (2014).

[391] F. Zhang and B. W. Wah. "Fundamental Principles on Learning New Features for Effective Dense Matching". In: *Transactions on Image Processing* (2018).

[392] G. Zhang, J. Jia, T.-T. Wong, and H. Bao. "Recovering consistent video depth maps via bundle optimization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[393] Y. Zhang, Z. Jia, and T. Chen. "Image retrieval with geometry-preserving visual phrases". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[394] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm. "PatchMatch Based Joint View Selection and Depthmap Estimation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[395] E. Zheng, E. Dunn, R. Raguram, and J.-M. Frahm. "Efficient and Scalable Depthmap Fusion". In: *British Machine Vision Conference (BMVC)*. 2012.

[396] E. Zheng and C. Wu. "Structure from Motion Using Structure-less Resection". In: *International Conference on Computer Vision (ICCV)*. 2015.

[397] C. L. Zitnick and T. Kanade. "A cooperative algorithm for stereo matching and occlusion detection". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2000).