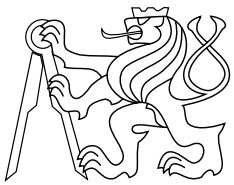
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

BACHELOR THESIS

3D Reconstruction of Indoor Scenes

Filip Šrajer

f.srajer@gmail.com

May 22, 2014

Available at
http://cmp.felk.cvut.cz/~srajefil/theses/bsc-filip-srajer.pdf

Thesis Advisor: Ing. Tomáš Pajdla, PhD.

This work has been supported by FP7-SPACE-2012-312377
PRoViDE and TA02011275 - ATOM grants.

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Filip Šrajer

Study programme: Open Informatics

Specialisation: Computer and Information Science

Title of Bachelor Project: 3D Reconstruction of Indoor Scenes

Guidelines:

1. Review the state of the art in 3D reconstruction in general and of indoor environments in particular. Focus on improving image matching and 3D reconstruction of indoor environments [3, 4].
2. Experiment with the standard approach to 3D reconstruction [1, 2] on indoor data and describe its limitations.
3. Suggest an improvement of [1, 2] for indoor environments and extend [1, 2].
4. Demonstrate that the new method improves 3D reconstruction on indoor environment compared to [1, 2].

Bibliography/Sources:

[1] N. Snavely, S. Seitz, and R. Szeliski: Phototourism: Exploring Photocollections in 3D. In SIGGRAPH, 2006.
[2] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski: Building Rome in a Day. In Proc. ICCV, 2009.
[3] A.G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun: Efficient Structured Prediction for 3D Indoor Scene Understanding. In Proc. CVPR, 2012.
[4] D. C. Lee, M. Hebert, and T. Kanade: Geometric Reasoning for Single Image Structure Recovery. In Proc. CVPR, 2009.

Bachelor Project Supervisor: Ing. Tomáš Pajdla, Ph.D.

Valid until: the end of the winter semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, November 21, 2013

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Filip Š r a j e r

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Rekonstrukce vnitřních prostor

Pokyny pro vypracování:

1. Prozkoumejte současné nejvyspělejší metody v rekonstrukci obecně a zaměřte se na rekonstrukci vnitřních prostor. Soustřeďte se na vylepšování korespondence v obrazech a rekonstrukci vnitřních prostor [3, 4].
2. Experimentujte se standardním přístupem k 3D rekonstrukci [1, 2] na datech z vnitřních prostor a popište jeho omezení.
3. Navrhněte vylepšení a rozšíření [1, 2] pro vnitřní prostory.
4. Demonstrujte, že nová metoda vylepšuje 3D rekonstrukci vnitřních prostor v porovnání s [1,2].

Seznam odborné literatury:

[1] N. Snavely, S. Seitz, and R. Szeliski: Phototourism: Exploring Photocollections in 3D. In SIGGRAPH, 2006.
[2] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski: Building Rome in a Day. In Proc. ICCV, 2009.
[3] A.G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun: Efficient Structured Prediction for 3D Indoor Scene Understanding. In Proc. CVPR, 2012.
[4] D. C. Lee, M. Hebert, and T. Kanade: Geometric Reasoning for Single Image Structure Recovery. In Proc. CVPR, 2009.

Vedoucí bakalářské práce: Ing. Tomáš Pajdla, Ph.D.

Platnost zadání: do konce zimního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 11. 2013

Acknowledgements

I would like to express my thanks to my advisor Tomáš Pajdla for introducing me to computer vision and for valuable guidance and comments which enabled me to finish this thesis. I would also like to thank Čeněk Albl for advice on 3D reconstruction, Alexander G. Schwing for help with estimation of layouts of indoor scenes and Prof. Marc Pollefeys for ideas and comments on improving 3D reconstruction of indoor scenes and evaluation of the proposed approach. Last but not least, I am grateful to my family for all their support.

Author's declaration

I declare that I have developed the presented work independently and that I have listed all information sources used in accordance with the Methodical Guidelines on Maintaining Ethical Principles During the Preparation of Higher Education Theses.

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ....

Podpis autora práce

Abstract

Recent work in Structure from Motion (SfM) has successfully built 3D models from unordered collections of images. Inspired by their success, we choose one of these as the baseline [55] for implementing our own 3D reconstruction pipeline. We introduce three improvements dealing with detection of a sufficient number of features for high resolution images, speeding up a standard RANSAC for epipolar geometry estimation and focal length estimation for sets of image files with insufficient information about cameras. We show that our pipeline performs just as well as, and in some cases better than, the baseline. To improve reconstruction of indoor scenes, we observe that traditionally employed features do not work well for significant appearance changes of local patches which is typical for large camera transformations. We propose to firstly understand the indoor scene as a whole and then exploit this knowledge to improve image matching. Inspired by recent success of monocular indoor scene reconstruction, we estimate a box-like scene model for every input image and rectify individual faces which are then utilized for matching. We show that using these additional matches brings a dramatic improvement in reconstructing challenging indoor scenes from images.

Abstrakt

Nedávné práce zabývající se rekonstrukcí poloh kamer a bodů úspěšně vytvořily 3D modely z neuspořádaných kolekcí obrázků. Tento úspěch nás inspiruje a volíme jednu z těchto prací jako vzor [55] pro naší implementaci provádějící 3D rekonstrukci. Poskytujeme tři vylepšení, která se zabývají detekcí dostačujícího počtu klíčových bodů v obrázcích vysokého rozlišení, zrychlením RANSAC algoritmu používaného pro výpočet epipolární geometrie a odhadnutím ohniskových vzdáleností pro kolekce obrázků s nedostatkem informací o kamerách. Demostrujeme, že podáváme stejné a někdy lepší výsledky než vzor. Pro vylepšení rekonstrukce vnitřních prostor pozorujeme, že tradičně používané reprezentace klíčových bodů nefungují dobře pro výrazné změny vzhledu lokálních ploch, což je typické pro velké změny pozice kamery. Navrhujeme nejprve globálně porozumět vnitřnímu prostoru jako celku a využít toho k vylepšení hledání korespondencí. Inspirováni nedávným úspěchem monokulární rekonstrukce vnitřních prostor, odhadneme model prostoru jako kvádr pro každý obrázek a rektifikujeme jednotlivé stěny, které použijeme pro hledání korespondencí. Demonstrujeme, že tyto dodatečné korespondence přináší významné vylepšení rekonstrukce vnitřních prostor.

Contents

1 Introduction

1.1 Motivation

One of the important research topics in computer vision is 3D reconstruction from images. It is employed in projects such as Google Earth [22], Photo Tours in Google Maps [23, 33], Nokia Maps 3D [11] and Microsoft Photosynth [40].

Having nothing more than a set of images, it is possible to detect features in every image and match them to get some sense of how the images relate to each other. Utilizing these matches, an iterative Structure from Motion (SfM) [24] can be applied returning a reconstructed sparse 3D point model of the scene as well as extrinsics and intrinsics of all the recovered cameras [24]. See Fig. 1.1 for a visualization of an output of a 3D reconstruction pipeline and its possible further use.

Reconstructing an indoor scene is particularly difficult due to its textureless surfaces such as uniformly-painted walls which can result in detecting an insufficient number of features. We focus on this issue. We also address the wide baseline stereo problem, *i.e.* matching images of the same scene taken from very different angles. Our approach to these challenges is encouraged by the quality of the results obtained from monocular scene understanding. Our aim is to use global scene structure to improve feature matching, which in turn improves the entire reconstruction.

1.2 Problem statement

Since Bundler [55] by Snavely *et al.* is a frequently utilized tool in the field of 3D reconstruction, we chose it to be our reference reconstruction pipeline and firstly implement our pipeline similarly. We notice a problem which occurs for high resolution images. In more detail, it is often possible to detect so many features in high resolution images that the detection itself and subsequent matching becomes too time consuming. We argue that it is not necessary to detect as many features as possible and propose an approach to scale the feature detector. Next, we realize that Bundler employs a standard RANSAC [19] for epipolar geometry estimation which can be further improved. That is why we replace the standard RANSAC by our variant of Progressive Sample Consensus (PROSAC) [13] which achieves computational savings and thus speeds up the process of epipolar geometry estimation. We also note that Bundler relies on the ability to estimate focal lengths of at least one matched image pair from EXIF tags of a set of image files. The estimation of focal length usually fails when Bundler does not have a record of a sensor size of a camera in its internal database or when EXIF tags of an image file do not contain any information whatsoever. We address this problem by using image resolution for estimating the focal lengths.

In addition, we focus on indoor scene reconstruction. More specifically, we aim at improving image matching for indoor scenes. In general, all approaches represent image keypoints by some kinds of features and use a distance metric to find visually similar local patches. As a similarity measure, the standard Euclidean distance is usually applied. Conversely, the exploited image features vary from variants of scale-invariant feature transform (SIFT) [39] and histograms of oriented gradients (HOG) [18] to GIST

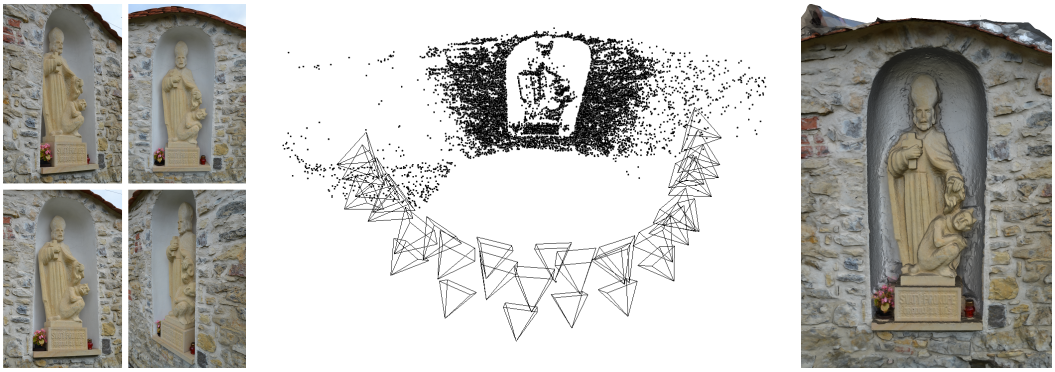


Figure 1.1 Sample images of a set of 31 images, a sparse 3D point model and cameras (visualized as pyramids) recovered by our 3D reconstruction pipeline using only the images and a textured 3D model which can be obtained using the result of our pipeline.

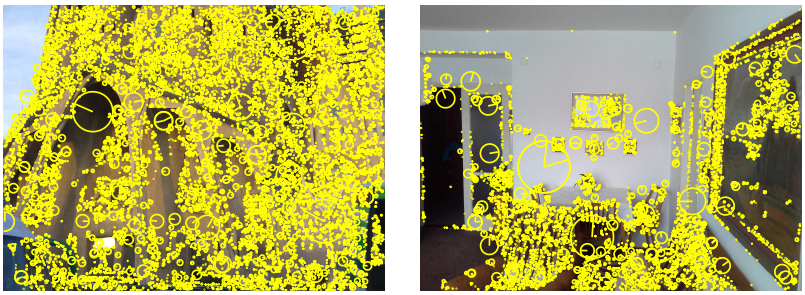


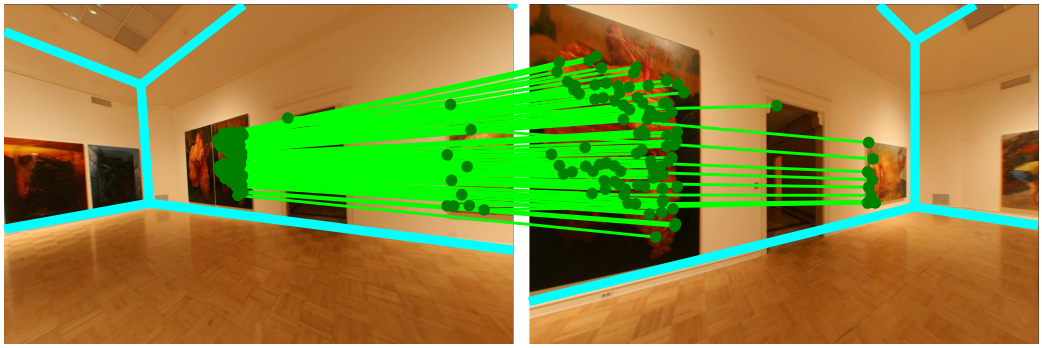
Figure 1.2 The image on the left shows detected keypoints on an outdoor scene (a part of a church) and the image on the right visualizes found keypoints on an indoor scene (a living room). The standard SIFT detector has found 12 136 on the outdoor scene and 4 999 on the indoor scene. The resolution of the image on the left is 1.77 Mpix. The resolution of the image on the right is 1.92 Mpix.

descriptors [46]. A noteworthy common property of these features is good representation of transformations such as scaling, rotation or illumination.

Hence, if we assume a small baseline, feature representations taking advantage of pixel intensities work very well in practice. For other situations, however, matching is often surprisingly difficult; especially if the observed scene is similar only on a higher level and features consist of pixels too different to be matched by standard matching approaches. Consider the translation and rotation of cameras between images in Fig. 1.3. Also, 3D reconstruction naturally has difficulties if too few features can be detected. The lack of features is often a problem for indoor scenarios; consider Fig. 1.2, for example. That is another reason why we want to improve feature detection and matching.

Many image matching approaches do not distinguish between indoor and outdoor scenarios. We argue that every application requires a custom approach and we subsequently suggest a solution specifically tailored for the indoor setting. Restricting ourselves to indoor environments enables us to employ the *Manhattan world assumption*, *i.e.*, we model a scene to be aligned according to three dominant and orthogonal directions defined by vanishing points. It has been shown [25, 64] that estimating four parameters is sufficient to roughly reconstruct the indoor room layout by means of a 3D box.

(a)



(b)

Figure 1.3 Image pairs that could not be matched by the standard approach were successfully matched using our indoor improvement. The figure visualizes matches verified by epipolar geometry and detected scene layouts. The images in (b) are from the data set of Furukawa [21].

Our solution utilizes this box-like indoor scene representation estimated from a single image. We employ rectification instead of sacrificing discriminative properties by only increasing the space of local transformations. In more detail, we rectify all faces of the box which enables us to find feature matches that undergo a large transformation between two images as illustrated in Fig. 1.3.

1.3 Thesis structure

In the following sections, we firstly discuss related 3D reconstruction state-of-the-art techniques in general and after that we focus on understanding indoor scene state-of-the-art methods. Secondly, we describe the reference 3D reconstruction pipeline, then build upon this explanation and discuss differences and enhancements made in our 3D reconstruction pipeline after which we present our improvement for indoor scene matching. Thirdly, we evaluate our pipeline and its extensions and we also evaluate the indoor improvement by comparing a standard run of our pipeline and a run enhanced with the indoor improvement. Last, we conclude by reviewing contributions of the thesis.

2 State of the art

We firstly review the state of the art in 3D reconstruction in general and secondly the work done in indoor 3D reconstruction.

2.1 3D reconstruction

3D scene modeling from images is an important problem of computer vision. Large progress has been made recently in understanding 3D scene modelling related key problems of geometry [24], optimization [60] and algebra [44].

3D reconstruction techniques differ by what assumptions are made about the input image set. Some approaches [3, 16, 41, 31, 57] assume that the image set is ordered as the image order gives a clue which image pairs have overlapping field of view and are therefore suitable for further processing. However, focusing on ordered sets of images only would be restrictive for us. We therefore take no such assumption which consequently allows us to process unordered as well as ordered sets of images. Unordered sets include images obtained from different sources or at different times. It could even be images returned by an image search on photo sharing sites like Flickr [67]. A typical image search could be: "Rome, Colosseum", "New York, Statue of Liberty", "Prague, St Wenceslas statue", *etc*.

First of all, we deem important to implement our own 3D reconstruction pipeline to be able to understand it properly and modify it prior to improving for a special setting of indoor environments. For that purpose, we are inspired by state of the art techniques in 3D reconstruction from unordered image sets [65, 10, 17, 2, 37, 20, 50, 62], especially by well known 3D modelling system Bundler [55]. Similarly to the mentioned approaches, we take an unordered image set as input. Since SIFT features [39] seem to work quite well for our purposes according to [55, 65, 10, 17, 2, 37, 20], we begin by detecting them in all images individually. Next, we match features of all possible pairs of images. Specially, we notice that approximate nearest neighbor package (ANN) by Arya *et al*. [5] is employed in [55, 2] and other approximate nearest neighbor implementations in [65, 10]. We leverage previously successfully deployed ANN [5] and also fast library for approximate nearest neighbor (FLANN) included in OpenCV library [9]. We follow with verification of matches by a distance ratio test [39] similarly to works [55, 2, 65, 20]. After that, matches are further verified by estimating epipolar geometry of every image pair as in [55, 2, 10, 37, 20, 50, 62]. Finally, focal lengths of cameras need to be estimated for initializing a Structure from Motion (SfM). We estimate the focal lengths from EXIF tags of the images as do [55, 2, 17, 37, 20]. When no EXIF information exists we use image resolution for the estimation as opposed to [55] which sets focal lengths of an initial camera pair to a predefined constant value. The last step, an incremental Structure from Motion (SfM), of the pipeline is implemented similarly to Bundler [55] with a difference being that we not only employ the sparse bundle adjustment library of Lourakis and Argyros [38] but also Center for Machine Perception (CMP) version of bundle adjustment [4] which is based on CERES solver [1].

Recent work has focused on improving 3D reconstruction of scenes containing repetitive structures or symmetries which is often a phenomenon for man-made objects and

buildings. Typical examples of repetitive structures range from windows on buildings (imagine a modern skyscraper) through wallpapers with repetitive patterns to scenes containing multiple identical objects (imagine a dining room with identical chairs). A problem of detecting repetitive patterns on non-planar surfaces is addressed by [29]. This approach utilizes multiple images and a set of 3D points reconstructed from them by Structure from Motion in order to rectify geometric deformations. The approaches [68, 30] exploit the idea of *missing correspondences*. Exploiting triplets of images, two images can be used to predict features in the third image. Absence of the predicted features enables additional inference about the triplet. An expectation maximization based algorithm is used in [49] to estimate camera poses and identify falsely-matched image pairs. The algorithm exploits geometric reasoning as well as image-based cues. In contrast, some researchers consider repetitive structures to be an important distinguishing feature instead of a difficulty. In [59], repetitive structures are detected in an image and subsequently used to re-weight the bag-of-visual-word model which standardly assumes independence of features. In [51], highly repetitive nature of urban environments is exploited. In more detail, multiple repetitive 2D patterns are detected and then matched to a database of textured facades in order to geo-tag a photograph. In [15], symmetries are detected using geometric and appearance cues and then symmetry constraints are imposed on the Structure from Motion to improve 3D reconstruction. Considering a dense 3D reconstruction from one image, it was shown in [32] that it is possible to detect a symmetry plane. They used the plane to create a virtual camera on the other side of the plane if the original camera was not situated directly on the plane. Then, they utilized both of the cameras for dense 3D reconstruction. Similarly, [66] improves a single-view dense reconstruction given an image and its detected repetitive structures. In order to improve the reconstruction, a repetition constraint is introduced to penalize the inconsistency between repetition intervals.

2.2 Indoor reconstruction

It is often very advantageous to assume something about a task instead of designing a general approach. This is true especially for object detection for which all the approaches are almost always based on SIFT [39], GIST [46] and HOG [18] features. Image matching which is a component of many applications is no exception to this principle.

Importantly, we note that small differences in viewpoint often result in arbitrary pixel dissimilarities. But a problem of designing a metric ignoring small details while distinguishing the important patterns due to which two patches appear similar is still challenging. We propose to imitate human behaviour by firstly understanding a new scene on a global level and subsequently identify small details. This approach contrasts commonly utilized techniques that directly focus on small details from the very beginning.

An important element for finding correspondences is the feature space and the similarity metric of features. Aforementioned representations like SIFT, GIST and HOG as well as various other wavelet and gradient decompositions and combinations such as spatial pyramids [34] are commonly employed. To address the mentioned locality restriction, matching techniques were dominantly improved in two directions. In the first one, the space of considered transformations is increased which influences computational efficiency and discriminative abilities. The researchers going in the second direction modify the similarity metric [42, 7, 6, 12, 8]. Other approaches formalize

image matching from a data perspective to learn a better visual similarity. Tieu and Viola [58] use boosting to learn image specific features and Hoiem *et al.* [28] employ a Bayesian framework to find close matches. Contrasting the mentioned work, which is based on multiple training examples, Shrivastava *et al.* [54] showed how to achieve cross-domain matching using structured support vector machines learned from a single example. Impressive results were demonstrated. Nevertheless all these approaches can still address only minor viewpoint changes.

To deal with a larger number of local transformations we follow the physiologic intuition by first investigating an observed scene from a more global perspective. We specifically design a solution for image matching of indoor scenes by leveraging the *Manhattan world* assumption, the restriction that scenes are aligned to three dominant and orthogonal vanishing points. This assumption was already utilized in indoor 3D reconstruction by Furukawa *et al.* [21] for a stereo algorithm which was employed in an automatic system capable of reconstructing and visualizing a house interior. Furthermore, Manhattan world assumption has enabled researchers to design methods that retrieve a 3D layout that fits the observed room layout even when given only a single image.

The Manhattan world assumption was also taken by methods for monocular scene layout estimation [25, 64, 35, 47, 52, 53]. As a consequence, [25, 64] introduced a simple parameterization of the 3D layout based on four variables. By exploiting the decomposition of the additive energy functions with an integral geometry technique [52], globally optimal inference of frequently utilized cost functions was shown to be possible [53]. Given high quality image cues known as geometric context [27] and orientation maps [36] accuracies exceeding 85% are achieved [53] on standard data sets [25, 26].

In contrast to recent work in image matching, which has been extended to better represent local transformations, we introduce the first work for indoor Structure from Motion to use global scene interpretation for rectification of local patches. Note that rectification based on global image properties has been done for outdoor facades in the context of image-based localization [14].

3 The proposed approach

3.1 Reference Bundler implementation

We have chosen Bundler v0.4 [55] to be our baseline of 3D reconstruction pipeline. With only image files observing a scene as input, this system outputs parameters of recovered cameras, including intrinsics (focal length and radial distortion coefficients) and extrinsics (camera rotation and translation). In addition, reconstructed 3D points (color and 3D position) are returned. This is achieved by computer vision techniques. Firstly, the features are detected in every image separately. Secondly, the features are matched between all pairs of images. Thirdly, the feature matches are verified. Last, an iterative SfM procedure is applied.

3.1.1 Feature detection

In this first step of the pipeline, some kinds of keypoints need to be detected in every image and after that a descriptor computed for every keypoint in every image.

SIFT keypoint detector [39] is utilized due to its invariance to image transformations. Besides the keypoint locations themselves, SIFT provides a local descriptor for each keypoint. This descriptor is a 128 dimensional vector. Bundler makes use of David Lowe's SIFT implementation for this process.

3.1.2 Feature matching

Next, having a set of SIFT descriptors for each image, the descriptors are matched for every image pair. An image pair is an unordered set of two images. For instance, image pair (img_1, img_2) is the same as (img_2, img_1).

Matching is done using approximate nearest neighbor (ANN) kd-tree package by Arya *et al.* [5]. To match the features of images img_1 and img_2, a kd-tree from feature descriptors of img_2 is created and subsequently a nearest neighbor is found for each feature descriptor of img_1 using the kd-tree. For efficiency, a priority search algorithm of ANN is employed, limiting each query to visit maximum of 200 bins in the kd-tree.

Furthermore, to eliminate false matches, a ratio test, described by Lowe [39], is exploited rather than thresholding the distance to the nearest neighbor. For a feature descriptor in img_1, the nearest neighbor and the second nearest neighbor is found in the set of descriptors of img_2 with distances d_1 and d_2 respectively. A match is then accepted only if $\frac{d_1}{d_2} \leq 0.6$. Additionally, if more than one feature descriptor in img_1 matches the same feature descriptor in img_2, all of these matches are removed as some of them must be spurious. We will refer to the resulting matches as *tentative*.

3.1.3 Verification of matches

Furthermore, all tentatively matched image pairs must be verified. The verification consists of two phases.

In the first, a fundamental matrix is estimated for each matched image pair using an eight-point algorithm [24] inside a RANSAC [19]. The RANSAC investigates a total

number of 2048 fundamental matrices. A tentative match is defined as an outlier if its residual for a generated fundamental matrix fails to lie within a predefined threshold. Additionally, the RANSAC is preliminarily terminated if a fundamental matrix supported by at least 95% of all the tentative matches is found. If the best epipolar geometry for a given image pair is supported by at least 16 inliers then it is accepted and outliers are disregarded. In the case when less than 16 inliers are found, the hypothesis is rejected as unreliable and no matches are taken into account. Additionally, the fundamental matrix is refined by running the Levenberg-Marquardt algorithm for non-linear least squares minimization (Nocedal and Wright [45]) of errors of all the inliers to the best fundamental matrix found by the RANSAC. We will refer to the resulting matches as *verified*.

The second phase utilizes the geometrically verified matches mentioned above. All the matches across all the images are organized into *tracks*. A track is a set of matching features across multiple images. Note that even a single match between two images is a track. Only consistent tracks are kept. A consistent track is a track that contains a maximum of one feature in one image.

3.1.4 Focal length estimation

Every image is considered to be taken by a different camera. That is why focal lengths are estimated for every camera separately. The focal length in pixels is computed using information from EXIF tags of an image file.

Firstly, information from the EXIF tags are extracted using Matthias Wandel's jhead program [63]. Out of all the extracted data, only camera make, camera model, focal length in millimeters, image resolution and CCD width are of interest.

Next, the width of the image sensor of the camera in millimeters is looked up in an internal database using the camera make and model information. In cases where the database does not contain an entry for the desired camera, the CCD width from the EXIF tags is used.

The focal length in pixels is then computed as

$$\text{focal length in pixels} = \max(\text{width}, \text{height}) \frac{\text{focal length in mm}}{\text{image sensor width in mm}} \tag{3.1}$$

where (width x height) is the image size. These estimates are convenient for the SfM initialization but sometimes they might be inaccurate. Unfortunately, EXIF tags do not always contain all the necessary information, so in the event of missing crucial information in EXIF tags, no estimate is provided and the camera is treated differently in the SfM.

3.1.5 Structure from Motion

Finally, a set of parameters of cameras and 3D locations of tracks can be recovered. For the camera parameters and the tracks, the reprojection error, *i.e.* the sum of distances between the projections of each track and its corresponding image features, is minimized. The minimization problem can be formulated as a non-linear least squares problem (see the following paragraph) and solved using bundle adjustment [60]. Algorithms for finding a solution to this non-linear problem, such as Nocedal and Wright [45] guarantee finding a local minimum. Unfortunately, large-scale SfM problems tend to get stuck in bad local minima. That is the reason why it is crucial to provide a good

initial estimate of the parameters. An approach of estimating initial camera pair parameters and then iteratively adding other cameras is chosen instead of estimating the parameters of all cameras and tracks at once.

Optimization We will now formulate the aforementioned minimization problem as a non-linear least squares problem as in [55, 56].

A perspective camera can be parameterized by an eleven-parameter projection matrix. It is possible to reduce the number of parameters by making the common additional assumptions that the pixels are square and that the center of projection is coincident with the image center. The parameters are: the 3D orientation (three), the camera center c (three), the focal length f (one). In addition, the system solves for two radial distortion parameters κ_1 and κ_2, so the total number of parameters per camera is nine.

An incremental rotation ω is used to parameterize the 3D rotation, where

$$\mathbf{R}(\theta, \hat{n}) = \mathbf{I} + \sin\theta \big[\hat{n}\big]_\times + (1 - \cos\theta)\big[\hat{n}\big]_\times^2, \quad \omega = \theta\hat{n} \tag{3.2}$$

is the incremental rotation matrix applied to an initial rotation and

$$\big[\hat{n}\big]_\times = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix} \tag{3.3}$$

The nine parameters are grouped into a vector $\mathbf{\Theta} = \begin{bmatrix} \omega & c & f & \kappa_1 & \kappa_2 \end{bmatrix}$. Each point is parameterized by a 3D position p.

To formulate the optimization problem, consider a set of n cameras, parameterized by $\mathbf{\Theta}_i$, a set of m 3D points (tracks) parameterized by p_j and a set of 2D projections (feature locations) q_{ij}, where q_{ij} is the observed projection of the j-th 3D point by the i-th camera.

Let $\mathbf{P}(\mathbf{\Theta}, p)$ be the equation mapping a 3D point p to its 2D projection in a camera with parameters $\mathbf{\Theta}$. $\mathbf{P}$ transforms p to homogeneous image coordinates and performs the perspective division that applies the radial distortion:

$$\begin{aligned} p'(\mathbf{\Theta}, p) &= \mathbf{KR}(p - c) \\ p_0(\mathbf{\Theta}, p) &= \begin{bmatrix} -p'_x/p'_z & -p'_y/p'_z \end{bmatrix}^T \\ \mathbf{P}(\mathbf{\Theta}, p) &= g_{rd}(p_0) \end{aligned} \tag{3.4}$$

where $\mathbf{K} = diag(f, f, 1)$ and g_{rd} is the distortion equation that maps a projected 2D point $q = (q_x, q_y)$ to a distorted point as follows

$$\begin{aligned} \rho^2 &= (\frac{q_x}{f})^2 + (\frac{q_y}{f})^2 \\ \alpha &= 1 + \kappa_1\rho^2 + \kappa_2\rho^4 \\ g_{rd}(q) &= \alpha q \end{aligned} \tag{3.5}$$

It is sought to minimize the sum of the reprojection errors:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\|q_{ij} - \mathbf{P}(\Theta_i, p_j)\| \tag{3.6}$$

w_{ij} is used as an indicator variable where $w_{ij} = 1$ if camera i observes 3D point j and $w_{ij} = 0$ otherwise.

Initialization To initialize the algorithm, an initial camera pair must be selected and subsequently the parameters of both cameras estimated.

In order to robustly estimate the initial two-frame reconstruction, the initial pair should have a lot of matches and a large baseline. To ensure that the pair has a large baseline, it is confirmed that the verified matches (see Sec. 3.1.3) of this pair cannot be well modeled by a homography. For the estimation of a homography, a RANSAC [19] is utilized. The RANSAC investigates a total of 256 homographies. A match is defined as an outlier if the distance between a feature from one image and a feature from the second image transformed by the generated homography exceeds a predefined pixel threshold. If less than 10 inliers support the best homography, an empty inlier set is returned.

The initial image pair is chosen as the one with the most matches satisfying the conditions of having at least 32 matches and the percentage of inliers to the best homography lower than 50%. In cases where no such pair exists, the initial pair is chosen as the one with the lowest percentage of inliers to the best homography and with at least 80 matches. Additionally to both of the rules, both images in the pair must have a focal length estimate computed from their EXIF tags (see Sec. 3.1.4). If there are no two matched images with a known focal length estimate, the initial image pair is selected according to the aforementioned rules but without the restriction on the focal length estimate. If even then no suitable image pair could be chosen, then considering a lexicographical ordering of image filenames, the first and the second image is selected.

Next, if the cameras have a known focal length, initial camera parameters of the initial pair are estimated using Nistér's five point algorithm [44]. With the camera parameters estimated, the 3D positions of tracks that correspond to the matched features between the pair are computed by triangulating the projections (features) in the images. When the focal lengths of the initial camera pair are unknown, then both camera centers are set to the origin and their rotation matrices are set to the identity matrices. Subsequently, the 3D positions of tracks corresponding to the matches between the pair are computed as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_{\text{proj}}/\text{default focal length} \\ y_{\text{proj}}/\text{default focal length} \\ 1 \end{bmatrix} \text{initial depth} \tag{3.7}$$

where x, y and z are 3D coordinates of a track, x_{proj} and y_{proj} are 2D coordinates of a feature corresponding to the track, the default focal length is set to 532 and the initial depth is set to 3. This basically means that the points are back projected to a constant depth.

Last, a two-frame bundle adjustment is applied.

Iterative procedure With the algorithm fully initialized, the procedure described in this paragraph is iterated until all the images are added to the reconstruction or until no more images can be added.

To start with, a camera observing the most tracks which were already reconstructed is found. In cases where this camera observes less than 16 tracks, the iterative procedure is stopped since only weakly connected cameras are left (this corresponds to the stopping criterion mentioned above). If the procedure was not stopped then let us call the number of tracks that are observed by the found camera $n_{\max}$. Then also all other cameras that observe at least 75% of $n_{\max}$ are found; all the found cameras are added to the optimization. To initialize parameters of the newly added cameras, for each of the cameras, a projection matrix is estimated using the direct linear transform (DLT) tech-

nique [24] in a RANSAC [19]. A total of 4096 projection matrices are investigated. An inlier is defined as having an error less than 4 pixels and a *weak inlier* less than 64 pixels; the error is the distance between a projection and a 3D point projected by the generated matrix. The best projection matrix is further refined using the Levenberg-Marquardt algorithm for non-linear least squares minimization of errors of all the inliers to the best matrix found by the RANSAC. If the resulting projection matrix is supported by at least 6 inliers then it is accepted as sufficiently reliable. When no projection matrix was found, then the camera is removed from the optimization. Next, camera rotation matrix, translation vector and calibration matrix $\mathbf{K}$ are computed from the projection matrix using QR decomposition. The acquired $\mathbf{K}$ can be used as an estimate of a focal length

$$f_1 = \frac{1}{2}(\mathbf{K}_{11} + \mathbf{K}_{22}) \tag{3.8}$$

If the focal length was estimated from EXIF tags, let us refer to it as f_2. For further use, f_2 is preferred on condition that

$$0.7f_1 < f_2 < 1.4f_1 \tag{3.9}$$

Otherwise f_1 is used, including cases in which f_2 could not be estimated. Finally, radial distortion parameters are initialized to 0. With a set of initial parameters including the rotation matrix, the translation vector, the focal length and the radial distortion parameters, a bundle adjustment step is applied, allowing only the new camera and the points it observes, which are weak inliers defined earlier in this paragraph, to change. The rest of the reconstructed model is fixed. Some of the input points (weak inliers) might get rejected in the process. If less than 8 points remain after the bundle adjustment step or the estimated focal length is smaller than $0.1width$, where $width$ is the image width, the new camera is removed from the optimization. An inlier in the bundle adjustment step is defined as a point with a reprojection error lower than a threshold which is set as

$$\min\left(\max\left(2.4\ d_{95}, 8\right), 16\right) \tag{3.10}$$

where d_{95} is a 95th percentile of reprojection errors of all points which were inliers in the previous round of the bundle adjustment (all weak inliers in the first round). That way it is ensured for errors less than 8 to be always inliers and the ones with error above 16 to be always outliers, with the exact threshold lying in between.

Furthermore, tracks observed by the cameras newly added to the optimization are also added to the optimization if they meet the following conditions. A track has to be observed by at least two cameras that are already in the optimization (including the new ones). Also, triangulating a track must give a well-conditioned estimate of its location. The conditioning is given by considering all pairs of rays that could be used for triangulation and finding a pair of rays with the maximum angle of separation. If the maximum angle is larger than 2 degrees the track is well-conditioned and is triangulated using least squares. If the reprojection error is smaller than a 16 pixel threshold, the triangulated track is accepted and added to the optimization.

Finally, a global bundle adjustment is employed to refine the whole model. The minimization is done using the sparse bundle adjustment library of Lourakis and Argyros [38]. After every run of this optimization, outlier tracks are detected and removed, and that is repeated beginning with the optimization until no outlier tracks are detected. An outlier track is a track associated with at least one feature for which the reprojection error exceeds a threshold, which is defined as

$$\min\left(\max\left(2.4\ d_{80}^{img_i}, 8\right), 16\right) \tag{3.11}$$

Figure 3.1 A visualization of standard SIFT keypoints on the left and upright on the right. The yellow circles represent individual keypoints. The scale of a keypoint is illustrated by the radius of its respective circle and an orientation by a line going from the center of the circle to a point on it. This example shows only keypoints of a scale higher than 10 to enable a nice visualization.

where $d_{80}^{img_i}$ is the 80th percentile of the reprojection errors for image img_i.

Additionally, all tracks which are in the optimization are pruned after the global bundle adjustment step. Similar to adding a new track to the optimization, the maximal angle of separation between all pairs of rays associated with the track is found. If this angle is smaller than 1 degree, the track is removed from the optimization. Note that it is possible for the track to be added to the optimization again if a new camera observing this track is added to the optimization.

3.2 Reconstruction pipeline implementation

This section presents our implementation of a 3D reconstruction pipeline. Since we use Bundler [55], described in detail in Sec. 3.1, as a reference reconstruction pipeline, we will now point out differences and improvements of our implementation.

3.2.1 Feature detection

We have seen fit to also use SIFT keypoint detector and descriptor [39]. We provide the user of our pipeline with two choices concerning implementation of the SIFT; both of them being publicly available open source projects. Firstly, one can choose the OpenCV library [9]. Specifically, we employ the currently newest version 2.4.8. Secondly, one can decide for the VLFeat library [61]. We again exploit the currently newest version, which is 0.9.18. In our experiments, we use the VLFeat library because its interface enables us to easily implement extensions presented below.

Additionally to the VLFeat library option, we provide a sub option. The user can choose to detect more discriminative upright SIFT keypoints, which are defined as standard SIFT keypoints with fixed orientation. Note that it makes them susceptible to rotations. For a visual comparison of the standard and the upright SIFT keypoints, see Fig. 3.1.

SIFT octaves We will now explain the basic idea of a few technical details of SIFT [39] so that even a reader unfamiliar with it would understand our next step. To detect keypoints, SIFT exploits so called scale spaces. To understand scale spaces, it is useful to consider a person looking at an object, *e.g.*, a tree, at different distances. From afar

they would see only the shape of the tree. As they come closer, they would be able to recognize more details such as different leaves, and subsequently even smaller details of the individual leaves. In SIFT, these viewpoints from different distances are simulated by resizing the original image. Keypoints are then detected in the different sizes of the image. The processing of one size of the image is referred to as an *octave*. Typically, the first octave is created by doubling the original image in size. The second one is then formed from the original image and all the following octaves are based on an image preceding them resized to half of its size. That means that the third octave is created from the original image resized to half size, the fourth octave from the original image scaled down to a quarter of its size and so on.

Scaling of the SIFT detector It is not always desirable to detect as many keypoints as possible since it significantly decreases computational efficiency. Therefore, a problem of detecting too many keypoints arises for high resolution images. Usually, one would need to resize all images to some reasonable size prior to inputting them into the pipeline. That is true even for Bundler [55]. Another way to approach this problem would be detecting all keypoints and then keeping only some of them. Note that some scoring would be necessary for choosing which to keep and which to discard.

Nonetheless, we argue that it is sufficient to exploit the SIFT octaves mentioned above. We notice that choosing a different first octave is equivalent to resizing the image a priori. Also, we reason that there is some function taking resolution of an image and returning first octave at which the SIFT keypoint detection should start. Next, consider an ordered sequence of octaves

$$o_{-1}, o_0, o_1, \ldots, o_k \tag{3.12}$$

and corresponding resolutions in megapixels of images they are formed from

$$r_{-1}, r_0, r_1, \ldots, r_k = 2r_0, r_0, \frac{1}{2}r_0, \frac{1}{4}r_0, \ldots, \frac{1}{2^k}r_0 \tag{3.13}$$

where r_0 is the resolution of the original image. We are looking for a function $f(r) = i$, where r is a resolution of an image in megapixels and i is an index of the starting octave. For this purpose, we collected a set of images of various sizes and manually determined which octaves are ideal to start at, where ideal stands for the detection of as many keypoints as possible but staying under 20000 keypoints. We plotted the measured data as points. Next, we chose to approximate them by a function, which fits the points well

$$f'(r) = a_0 + a_1\sqrt{r} + a_2 \log r, \quad a_0, a_1, a_2 \in \mathbb{R} \tag{3.14}$$

The parameters (a_0, a_1, a_2) were then estimated from the measured data by least squares and the result is

$$f'(r) = -1.5673 + 0.5497\sqrt{r} + 0.0319 \log r \tag{3.15}$$

To get the final function which returns an index of the ideal octave we compose auxiliary functions

$$f(r) = g(h(f'(r))) \tag{3.16}$$

where function h rounds a number to the nearest whole number and function g only keeps a number in interval -1 and k, *i.e.*

$$g(x) = \max(-1, \min(k, x)) \tag{3.17}$$

By providing the scaling of the SIFT detector, we remove the overhead connected to a resizing of images and in addition the user of our pipeline does not need to manually choose an ideal resolution of their images. We utilize the scaling approach for VLFeat [61] option since it offers easy interface for choosing the first octave. Nevertheless, we still enable the user to set the first octave manually.

3.2.2 Feature matching

Similar to the feature detection, we provide two possible choices of libraries. The two choices use different algorithms for the computation. First, we make use of the approximate nearest neighbor package (ANN) by Arya *et al.* [5] the same way as described in Sec. 3.1.2. Next, we utilize OpenCV [9] for two different algorithms. The first being a naive brute force matcher, which for each descriptor in the first set finds the closest descriptor in the second set by computing distances to all descriptors in the second set. The second, OpenCV matcher, is based on a fast library for finding approximate nearest neighbor (FLANN) [43]. FLANN is a library containing a collection of algorithms for fast nearest neighbor search in large data sets and for high dimensional features. It is described in detail by M. Muja *et al.* [43]. From all the possibilities, we choose to employ an algorithm utilizing randomized kd-trees. More concretely, four randomized kd-trees are constructed and searched in parallel. We employ FLANN in our experiments, since in our tests it has proved to be faster than both brute force and ANN and it tends to be as effective in matching as ANN. The brute force matcher is disregarded mainly because it is ineffective for large data sets. We always apply a ratio test after matching as detailed in Sec. 3.1.2.

3.2.3 Verification of matches

The verification of tentative matches is done exactly as detailed in Sec. 3.1.3, with one exception. When estimating epipolar geometry, we exploit the Progressive Sample Consensus (PROSAC) [13] framework, presented later in this section, instead of the standard RANSAC [19]. In addition, we use a stricter threshold of 5 pixels for determining inliers. To complement this, we supply the user of our pipeline with an option to use OpenCV implementation of estimating epipolar geometry. Nonetheless, this option does not take advantage of our PROSAC implementation.

PROSAC We are inspired by the Progressive Sample Consensus (PROSAC) detailed by Raguram *et al.* [48] and Chum and Matas [13]. Our interest lies in speeding up the the standard RANSAC procedure which was proved to be achieved by [48, 13]. PROSAC utilizes ordering of matches by a quality measure. For example, a distance between two matched features can be used as a quality score of a match. PROSAC then firstly tests the most promising hypotheses which often allows it to terminate earlier than the standard RANSAC would.

For illustration, see the pseudocode of our PROSAC implementation in Alg. 1. As mentioned above, our PROSAC implementation exploits match distances. Therefore, we sort matches in an ascending order by the distances prior to the main loop of PROSAC. That enables using the best matched features first which in turn allows us to generate the most promising hypotheses first.

In the following, we define the sampling strategy of our PROSAC implementation the same way as in [13]. The set of N matches is denoted by $\mathcal{U}_N$. The matches in $\mathcal{U}_N$

Algorithm 1 Progressive Sample Consensus (PROSAC), see Sec. 3.2.3

Input: A set of matches $\mathcal{U}_N$ sorted according to the distance measure

```
r := 0
n := m
i_max := 0
while r < K do
    1. Choice of a semi-random sample M_r of size m
    if r = T'_n  &&  n < N then
        n := n + 1,  see Eq. (3.23)
    if r < T'_n then
        Choose m − 1 matches from U_{n−1} at random and u_n into the sample
    else
        Choose m matches from U_N at random into the sample
    2. Hypothesis generation
    Compute a hypothesis from the sample M_r
    3. Hypothesis evaluation and possible termination
    Compute a number of supporting inliers i to the hypothesis
    if i > i_max then
        i_max := i
        Remember the hypothesis as the best one so far
        if r ≥ k  &&  the hypothesis is non-degenerate then
            if i > c  ||  Eq. (3.25) then
                Terminate
        if i > 0.95N then
            Terminate
    r := r + 1
end while
```

are sorted in ascending order with respect to the match distance function d

$$\mathbf{u}_i, \mathbf{u}_j \in \mathcal{U}_N : i < j \Rightarrow d(\mathbf{u}_i) \leq d(\mathbf{u}_j) \tag{3.18}$$

The set of n matches with the lowest distance is denoted $\mathcal{U}_n$. Next, imagine the standard RANSAC drawing T_N samples of size m out of N matches. Let $\{\mathcal{M}_i\}_{i=1}^{T_N}$ denote the sequence of samples $\mathcal{M}_i \subset \mathcal{U}_N$ that are uniformly drawn by RANSAC and let $\{\mathcal{M}_{(i)}\}_{i=1}^{T_N}$ be the same sequence sorted in ascending order according to the distance score

$$i < j \Rightarrow q(\mathcal{M}_{(i)}) \leq q(\mathcal{M}_{(j)}), \quad q(\mathcal{M}) = \max_{\mathbf{u}_i \in \mathcal{M}} d(\mathbf{u}_i) \tag{3.19}$$

If the samples are taken in order $\mathcal{M}_{(i)}$, the samples that are more likely to be uncontaminated are drawn earlier. Progressively, samples containing matches with bigger distances are drawn. After T_N samples, exactly all RANSAC samples $\{\mathcal{M}_i\}_{i=1}^{T_N}$ were drawn. Next, let T_n be an average number of samples from $\{\mathcal{M}_i\}_{i=1}^{T_N}$ that contain matches $\mathcal{U}_m$ only

$$T_n = T_N \frac{\binom{n}{m}}{\binom{N}{M}} = T_N \prod_{i=0}^{m-1} \frac{n-i}{N-i}, \text{then} \tag{3.20}$$

$$\frac{T_{n+1}}{T_n} = \frac{T_N}{T_N} \prod_{i=0}^{m-1} \frac{n+1-i}{N-i} \prod_{i=0}^{m-1} \frac{N-i}{n-i} = \frac{n+1}{n+1-m} \tag{3.21}$$

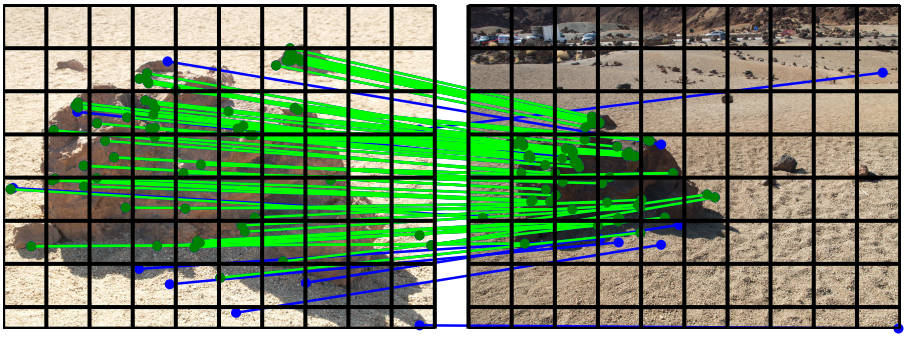


Figure 3.2 A visualization of the PROSAC non-degeneration check. The lines represent matches and the dots positions of matched keypoints. The color green stands for inliers and blue for outliers to the current hypothesis. This particular example shows a non-degenerate hypothesis.

Finally, the recurrent relation for T_{n+1} is

$$T_{n+1} = \frac{n+1}{n+1-m} T_n \tag{3.22}$$

There are T_n samples containing only matches from $\mathcal{U}_n$ and T_{n+1} samples containing matches from $\mathcal{U}_{n+1}$. Since $\mathcal{U}_{n+1} = \mathcal{U}_n \cup \{\mathbf{u}_{n+1}\}$, there are $T_{n+1} - T_n$ samples that contain a match $\mathbf{u}_{n+1}$ and $m-1$ matches drawn from $\mathcal{U}_n$. Therefore, a procedure that for $n = m \dots N$ draws $T_{n+1} - T_n$ samples consisting of a match $\mathbf{u}_{n+1}$ and $m-1$ matches drawn from $\mathcal{U}_n$ at random efficiently generates samples $\mathcal{M}_{(i)}$. As the values of T_n are not integer in general, we define $T'_m = 1$ and

$$T'_{n+1} = T'_n + \lceil T_{n+1} - T_n \rceil \tag{3.23}$$

To define our PROSAC procedure fully, we need to introduce a stopping criterion, since without it the procedure would not gain any computational savings in comparison with standard RANSAC. Let K be the maximum and k the minimum of tested hypotheses. We propose to terminate the algorithm when k hypotheses has already been tested, the best hypothesis found so far is non-degenerate and it is supported by at least c inliers. We set K to the number of rounds that the standard RANSAC would run, which in our case (see Sec. 3.1.3) is 2048. Next, we want to test at least k hypotheses where

$$k = \min\left(K, \max\left(100, \lfloor 0.1K \rfloor\right)\right) \tag{3.24}$$

We propose that 50 inliers are enough to support a good epipolar geometry, assuming that it is non-degenerate and therefore we set the threshold c on inliers to $\min(50, 0.95N)$, where N is the number of all matches.

In order to define a non-degenerate hypothesis, consider Fig. 3.2. Having two matched images, we propose to partition both of them into cells. In more detail, we make the cells have a square shape with one row having rectangular cells depending on the aspect ratio of an image. The length of one side of a square is given by the image resolution. We set a longer side of the image to have 10 rows of squares which defines the whole grid. Next, having the partitioning of the images, we calculate the area that is covered by the cells which contain at least one inlier for both images. In cases where for at least one of the images the area is at least one eighth of the whole image, we say that the

hypothesis is non-degenerate. This comes from an assumption that a photographer is considered to have captured a desired object in at least one eighth of a photograph.

In addition, we propose that if a generated hypothesis is supported by a number of inliers close under the threshold c and a PROSAC is still to run for many rounds, it can be terminated assuming that the minimum of k hypotheses was already tested and the currently best hypothesis is non-degenerate. In more detail, we threshold a ratio

$$\frac{1 - \frac{|I|}{c}}{K - r} < t \tag{3.25}$$

where r is a current round, I is a set of inliers and t is the threshold. We have empirically found that setting the value of t to 0.00015 works well for estimating epipolar geometry. For an even bigger speedup, we introduce a secondary stopping criterion. If an hypothesis supported by at least 95% of all matches is found, the procedure is terminated immediately since the remaining 5% is not worth the additional computation.

3.2.4 Focal length estimation

We estimate focal length the same way as explained in Sec. 3.1.4.

3.2.5 Structure from Motion

For details on this phase see Sec. 3.1.5.

The first difference in our implementation is the strategy used for choosing an initial pair. We use a similar approach as in Sec. 3.1.5, which means finding a camera pair with the most matches and simultaneously having at least 32 matches and the percentage of inliers to the best homograhy lower than 50%. If there is no such pair, the initial pair is chosen as the one minimizing the inlier percentage to the best homography and having at least 80 matches. We firstly apply this set of rules to image pairs where both of the images have focal lengths estimated from their EXIF tags. If this fails, we look for the initial pair in a set of image pairs where at least one of the two images has an estimated focal length. Only when both of these searches are not successful do we attempt to find the initial pair in a set of all matched image pairs. In cases where even the last search fails, we pick the first and the second image when considering ordering of image files lexicographically.

As opposed to Bundler (see Sec. 3.1.5), in the need of estimating camera parameters for an initial camera pair without estimated focal length from their EXIF tags we approach the problem differently. To begin with, we estimate the focal lengths as

$$\text{focal length in pixels} = 0.82 \text{ image width} \tag{3.26}$$

Next, instead of putting both of the cameras into origin and setting their rotation matrices to the identity matrices, we utilize the aforementioned estimate of the focal lengths and run the five point algorithm [44] as if the focal lengths were estimated correctly from EXIF tags. This proves to be a remarkable improvement for reconstructing a set of cameras for which the focal lengths could not be estimated from EXIF tags (see the experiments in Sec. 4.1). Nonetheless, since Eq. (3.26) does not provide an accurate estimate, we place no constraint on the focal to the optimization.

Finally, we do not want to restrict ourselves to one bundle adjustment package. Therefore, we incorporate into the pipeline the sparse bundle adjustment library of Lourakis and Argyros [38] as well as a CMP version of bundle adjustment [4] which is based on Ceres Solver [1]. We compare both of the packages in the experiments (see Sec. 4.1 and Sec. 4.2.1).

3.3 Indoor reconstruction

See Fig. 3.3 for the visualization of steps taken by the proposed approach. We are given a set of images (step 1). First of all, we estimate scene model for every image individually. That consists of detecting tree mutually orthogonal vanishing points using the algorithm from Hedau *et al*. [25]. Subsequently, detecting orientation maps [36] and geometric context [27] image cues which are visualized in step 2. Next, minimizing an energy function, which is based on the image cues, to estimate the room layout. For minimization, we employ a variant of a branch-and-bound algorithm [53] which is briefly described in Sec. 3.3.1. The resulting scene layout is visualized in step 3 of Fig. 3.3. Given the scene model, we rectify walls, floor and ceiling as detailed in Sec. 3.3.2 (step 4). and use them for detecting and matching SIFT features (step 5).

Next, as described in Sec. 3.3.3 we extract and subsequently match standard SIFT features [39] from the rectified floor and ceiling as well as upright SIFT from the rectified walls. We transform the obtained matches back to the original image and combine them with the result from standard feature matching (see step 5 of Fig. 3.3). Afterwards, we verify the matches and run the SfM procedure.

3.3.1 Scene estimation

We will now give a brief description of the scene layout estimation. Note that our exposition follows the approach described in [53] and we refer the interested reader to this work for additional details.

For the scene estimation task, let the room layout be referred to via variable y. Finding the best scene interpretation is commonly defined as the general energy minimization problem

$$y^* = \arg\min_{y \in \mathcal{Y}} E(y). \tag{3.27}$$

where y^* denotes the best interpretation. In order to specialize the general energy minimization, we first discuss the parameterization of an indoor scene. Hence we describe the space of all possible layouts $y \in \mathcal{Y}$. Secondly, we detail the energy function E. Finally, we discuss the optimization algorithm.

Scene parameterization Following the standard monocular scene understanding literature [25, 64, 35, 47, 52, 53], we use the Manhattan world assumption, *i.e*., we assume that the observed scene is described by three mutually orthogonal plane directions (vanishing points). Therefore, having one image as input we detect the three vanishing points (vp) using the algorithm of Hedau *et al*. [25]. Next, we parameterize the indoor scene as a 3D box using the three vanishing points. Note that at most three walls as well as floor and ceiling can be visible in an image. Hence, similar to [25, 64], we parameterize the 3D box by four parameters y_i, $i \in \{1, \ldots, 4\}$ each corresponding to an angle describing a ray r_i, $i \in \{1, \ldots, 4\}$ as visualized in Fig. 3.4. We point out that the rays r_1, r_2 are limited to lying either above or below the horizon which is the line that connects vp_0 and vp_2. Also, similar constraint is valid for the rays r_3 and r_4 in order to only parameterize valid layouts. For efficient computation, the possible angles $y_i \in \mathcal{Y}_i = \{1, \ldots, |\mathcal{Y}_i|\}$ are discretized such that the space of all valid layouts $\mathcal{Y} = \prod_{i=1}^{4} \mathcal{Y}_i$ is a product space describing a countable amount of possibilities. To ensure a sufficiently dense discretization, the number of discrete states $|\mathcal{Y}_i|$ is made dependent on the location of the vanishing points while making sure that the area within the image domain covered by successive rays is smaller than 3000 pixel.

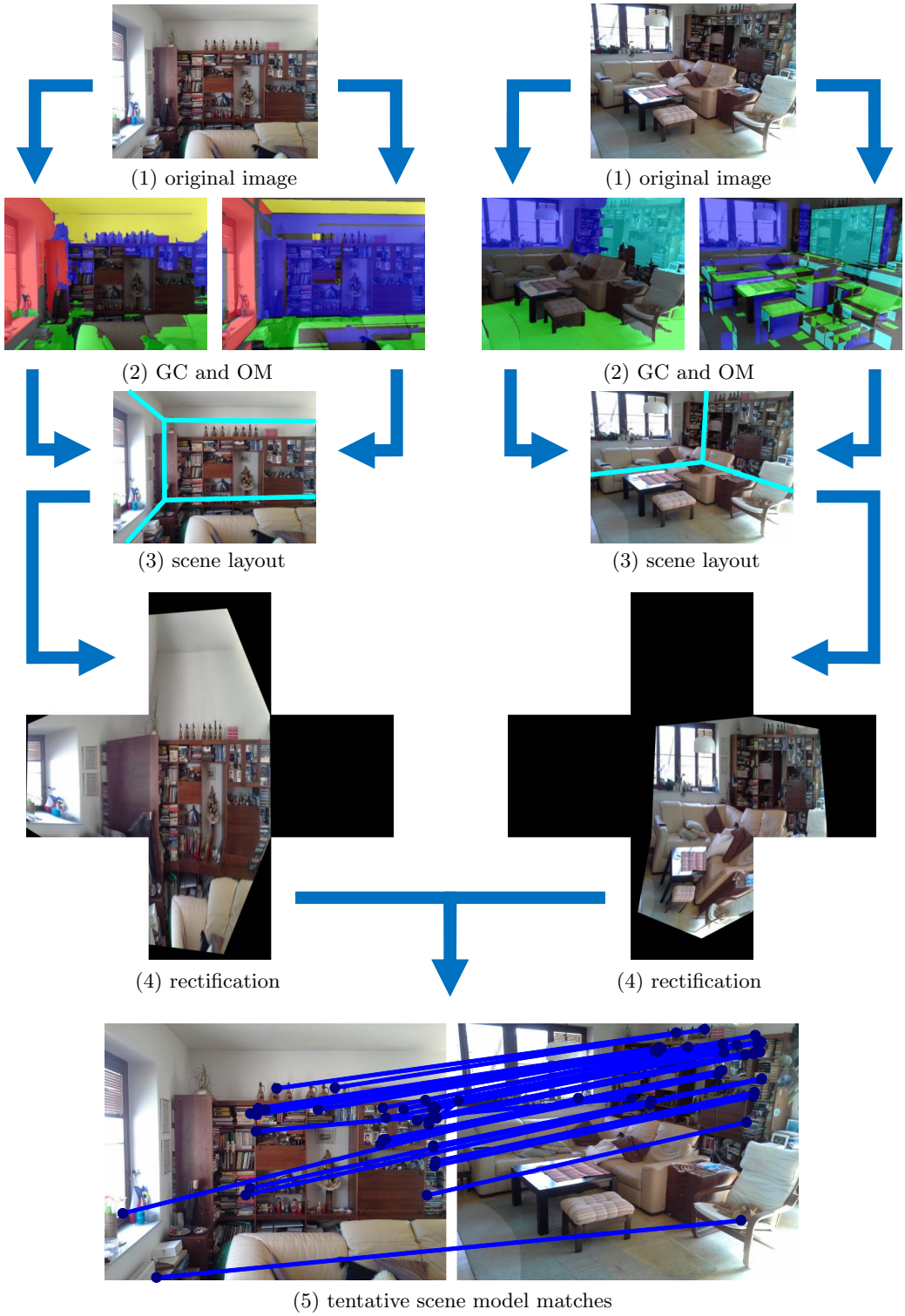


Figure 3.3 The proposed procedure. Given input images (1) we extract orientation maps (OM) and geometric context (GC) (2) which enable an optimization to find a scene layout (3). The scene estimate enables rectification of the detected faces (floor, ceiling and walls) (4) used for keypoint detection and matching (5). This particular example could not be matched by standard approach whereas we obtained 33 tentative matches.

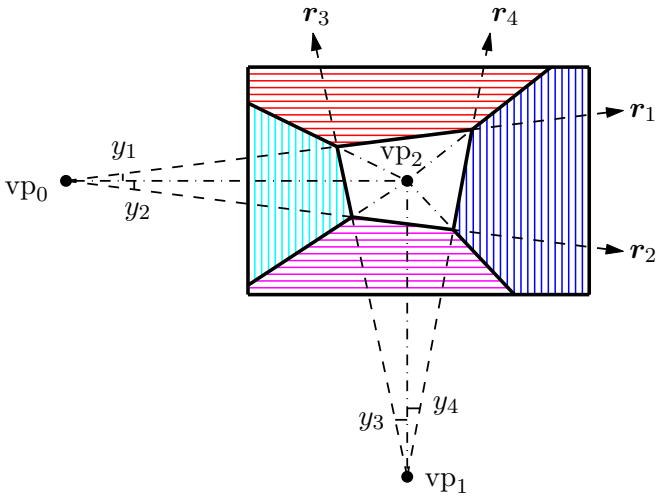
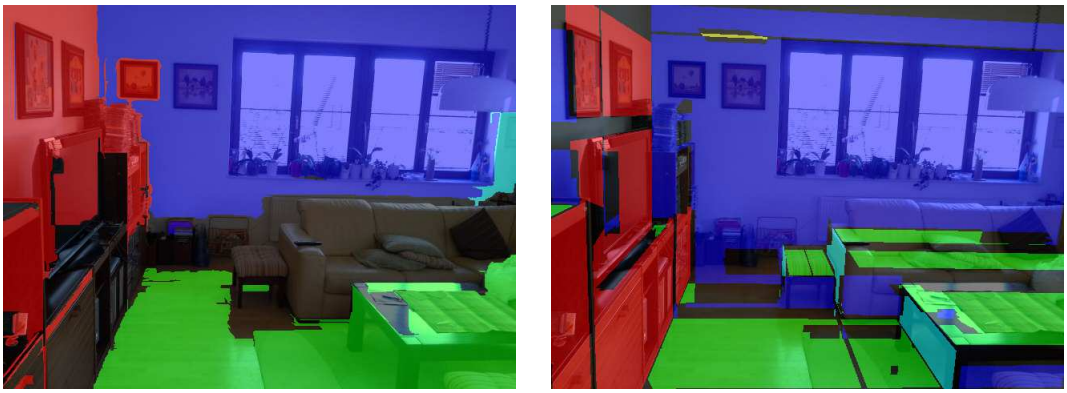


Figure 3.4 Parameterization of 3D layout estimation



(a) geometric context (b) orientation maps

Figure 3.5 A visualization of geometric context (GC) and orientation maps (OM). The colors red, blue, cyan, green and yellow represent the left wall, the front wall, the right wall, the floor and the ceiling, respectively. Uncolored locations symbolize unassigned pixels for OM and objects for GC.

Energy function Having all the possible layouts parameterized, we score a single layout hypothesis y by an energy function $E(y)$. In the following, we discuss the employed energy function. Let the five visible layout faces be subsumed within the set $\mathcal{F} = \{\text{left-wall}, \text{right-wall}, \text{front-wall}, \text{floor}, \text{ceiling}\}$. We design the energy function so that it decomposes into a sum of terms, each depending on a single layout face, *i.e.*,

$$E(y) = \sum_{\alpha \in \mathcal{F}} E_\alpha(y_\alpha). \tag{3.28}$$

Note that the set of variables involved in computing a face energy E_α is a subset of all variables, *i.e.*, $\alpha \subseteq \{1, \ldots, 4\}$ denotes a restriction of $(y_1, \ldots, y_4)$ to y_α.

To define an energy function of a single face, we employ geometric context (GC) [27] and orientation maps (OM) [36] (see Fig. 3.5 for an example), which were shown by many authors [35, 47, 52] as the most promising image cues for indoor scene layout estimation. Hence, we let the face energy function decouple to $E_\alpha = E_{\alpha,\mathrm{GC}} + E_{\alpha,\mathrm{OM}}$.

Figure 3.6 The estimated room layout represented by the color cyan.

Orientation maps are based on sweeping lines to obtain one of five possible wall orientations. In contrast, geometric context is computed using classifiers trained on a data set provided by Hedau *et al.* [25].

Scene model construction Importantly, using *integral geometry*, it was shown by Schwing *et al.* [52] that the energy functions $E_{\alpha,\cdot}$ decouple for every wall face α into a sum of terms with each summand depending on at most two variables. This enables efficient storage. Furthermore, [53] demonstrated that the geometric properties of the parameterization can be exploited in order to utilize branch-and-bound algorithm which retrieves the globally optimal solution $y^* = \arg\min_{y \in \mathcal{Y}} E(y)$ of the initially given optimization problem stated in Eq. (3.27).

The approach proceeds by successively dividing a set of layouts $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}^1 \cup \hat{\mathcal{Y}}^2$ into two disjoint sets $\hat{\mathcal{Y}}^1$, $\hat{\mathcal{Y}}^2$ with $\hat{\mathcal{Y}}^1 \cap \hat{\mathcal{Y}}^2 = \emptyset$. A lower bound on the energy function is computed for each set so that it is ensured for every member of the set to score equally well or worse. The sets are inserted into a priority queue with a score being the lower bound. An iterative procedure retrieves the lowest scoring set to be considered for further partitioning until such a set contains only a single element, *i.e.*, when the procedure stops, we have found the best scoring layout y^*. Even though the algorithm evaluates all the possible layout hypotheses in the worst case, it was shown in [53] that only a few layouts are partitioned in practice. An example of the best layout for one room is visualized in Fig. 3.6.

3.3.2 Image rectification

Finding the optimal 3D layout by solving the problem given in Eq. (3.27) yields a result similar to the one visualized in Fig. 3.6, which enables a rectification of the estimated 3D parametric box (shown in Fig. 3.7). The rectification, which could also be explained as an unwrapping of the 3D box, makes local patches in images look more similar which eventually makes matching them easier.

To unwrap the 3D box into a 2D image we apply a homography to each face separately. Given the three vanishing points and the estimation result being the four angles y_i, the four corners of the front wall are completely specified by intersecting the rays $\boldsymbol{r}_i$. Since we only observe three walls as well as the floor and ceiling, but not the closing wall of the box, the other corners are not specified uniquely. We compute the other corners in

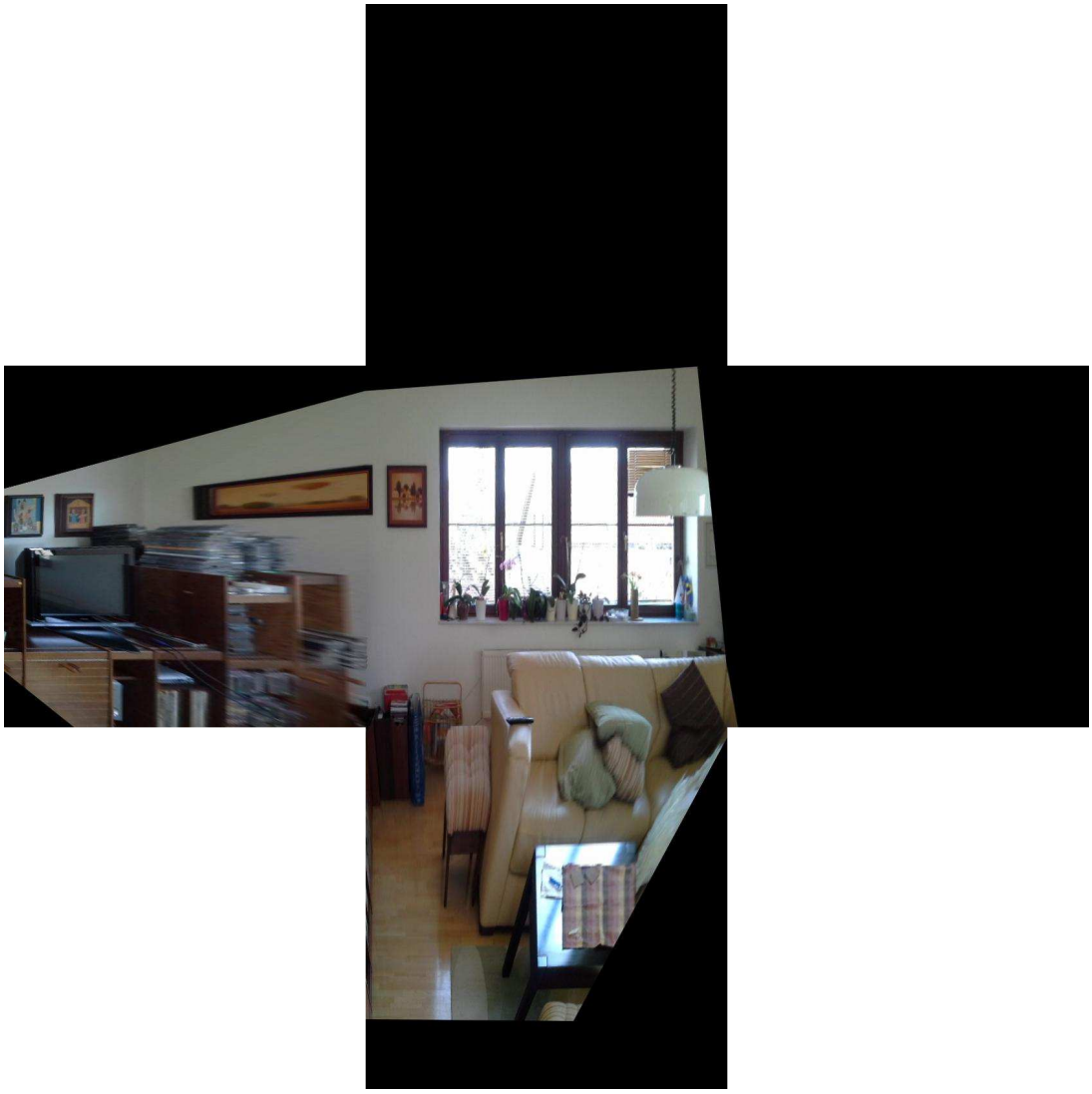


Figure 3.7 A visualization of the unwrapped room, *i.e.*, all rectified faces.

a way that no image region is cropped.

Hence, every wall is given by four points and we compute a projective transformation to warp each quadrilateral into a square-shaped image. The length of a side of the resulting image a is computed in the following way.

$$a = \min\left(\lfloor \frac{h+w}{2} \rfloor, 1600\right) \tag{3.29}$$

where h and w are the height and width of the original image.

To give an example, let the four corners of the quadrilateral describing the front wall be referred to via $x_1, \ldots, x_4 \in \mathbb{R}^2$. We solve a linear system of equations to obtain the transformation matrix $T_{\text{front-wall}}$ which projects $x_1, \ldots, x_4$ to the corners of the square defined above. We then warp the texture of the front-wall to the square by applying the transformation. The resulting rectification upon processing all the walls, ceiling and floor is illustrated in Fig. 3.7.

We address the fact that a camera can move in an indoor scene which makes defining the front, left and right wall ambiguous. We compose all three rectified images of walls

next to each other into one image. That way, when matching one composed walls image to another, we can easily match a wall which is front for one camera and right for the other, for example.

3.3.3 Feature extraction and matching

Next, we describe SIFT features [39] extraction and matching.

Feature extraction Having defined the images of rectified faces defined above, we employ SIFT extraction described in Sec. 3.2.1. We detect standard SIFT on floors and ceilings and more discriminative upright SIFT on walls since we assume that they are aligned with gravity. In cases where the scaling of the SIFT detector extension (see Sec. 3.2.1) is on, we relax the detection of the first octave for the image of walls. In more detail, the walls image is composed from 3 individual images so we adjust the function detecting the first octave by inputting a resolution three times smaller than the image containing the walls has. In addition, we traditionally detect standard SIFT for the original images.

Feature matching Next, we match the detected features utilizing the OpenCV [9] implementation of FLANN (see Sec. 3.2.2). Especially for our indoor approach, we match floors with floors, ceilings with ceilings, walls with walls and original images with original images only.

In the last step, we transform all the detected features from the rectified domain back to the original image domain using inverse mappings for every face, *e.g.*, $T^{-1}_{\text{front-wall}}$. Afterwards, we augment matches obtained by traditional matching of original images with those acquired by matching ceilings, floors and walls. We refer to these combined matches as *tentative scene model matches*.

3.3.4 Verification of matches and Structure from Motion

In this last phase, we verify tentative scene model matches using the algorithm described in Sec. 3.2.3. Subsequently, we run an iterative SfM procedure presented in Sec. 3.2.5.

4 Experiments

In the following, we firstly show that our reconstruction pipeline performs just as well or better than Bundler [55] on standard data sets and secondly we evaluate our approach to indoor scenes in comparison to a standard run of our pipeline without the indoor extension. By Bundler we always mean Bundler v0.4 described in Sec. 3.1.

4.1 Reconstruction pipeline

In this section, we evaluate our pipeline on 8 different data sets. These data sets contain 64, 13, 122, 66, 63, 189, 31 and 50 images. Lets consider labelling of the sets 1, 2,. . . , 8 as in Tab. 4.1. Then, the data set 1 depicts a paper model of Daliborka tower, Prague and neighboring buildings, the number 2 shows a desk with various leaflets and boxes on it, the number 3 displays a fantasy figurine of werewolf, the images in 4 capture an African tribal mask, the number 5 contains a statue of Probost Mikulas Karlach and the number 7 a statue of Saint Prokop, the data set 6 shows a large church Sagrada Familia, Barcelona and finally, the images in 8 capture a rock on sandy ground. We consider these data sets to be an interesting mixture of different types of scenes thus enabling us a good evaluation of our pipeline.

We divide the evaluation into the following subsections. First of all, we show how well our pipeline performs as a whole and what influence has our scaling of the SIFT detector. Secondly, we evaluate efficiency of our PROSAC implementation in contrast to the standard RANSAC. Finally, we compare two different bundle adjustment packages which we have integrated into our pipeline.

4.1.1 The pipeline as a whole and scaling of the SIFT detector

From all the options we have implemented and described in Sec. 3.2, we use one configuration. For SIFT, we utilize VLFeat [61]. For matching, we employ OpenCV [9] FLANN implementation. Matches are subsequently verified using the proposed PROSAC procedure (see Sec. 3.2.3). In SfM phase, we use CMP version of bundle adjustment [4], which is based on Ceres Solver [1].

Also, to test our automatic selection of the octave in SIFT detection described in Sec. 3.2.1, we show results for running our pipeline on resized images without the automatic octave detection (denoted as *Ours-*) and with the automatic octave detection on original images (denoted as *Ours+*). All results can be found in Tab. 4.1 and Tab. 4.2. We point out that Bundler does not implement any such thing as automatic resizing and therefore a problem of how to resize the images arises, because both Bundler and Ours- require resized images; for our purpose, to a size at which less than 20000 keypoints gets detected. Since every data set was taken with a single camera, all images in a single data set have the same resolution. Thus, for every data set we do the following. We let Bundler detect keypoints in all images of a data set and if an average of more than 20000 keypoints is detected, we decrease the size of all the images by a factor of 2 and repeat the process again. A resolution of the original images, of the resized images and additionally an octave which was selected as a starting one by our scaling extension can be found in Tab. 4.2.

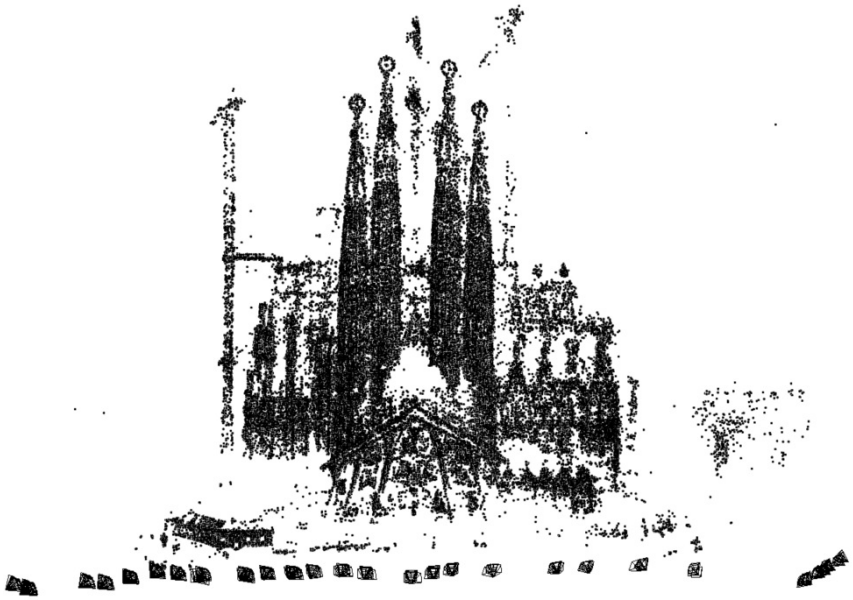
Set	# images	# cameras			Error		
		Bundler	Ours-	Ours+	Bundler	Ours-	Ours+
1	64	**64**	**64**	**64**	0.377	**0.320**	**0.320**
2	13	**13**	**13**	**13**	**0.206**	0.307	0.487
3	122	**122**	**122**	**122**	**0.195**	0.215	0.388
4	66	**21**	**21**	**21**	**0.438**	0.453	0.506
5	63	**63**	**63**	**63**	**0.217**	0.241	0.664
6	189	**189**	**189**	**189**	**0.333**	0.344	0.627
7	31	**31**	**31**	**31**	**0.261**	0.301	0.794
8	50	21	**50**	**50**	1.792	**0.217**	0.322

Table 4.1 The number of images and the number of cameras reconstructed by the standard Bundler procedure and by our proposed approach without and with the scaling of the SIFT extension (denoted by - and + respectively) for eight different data sets.

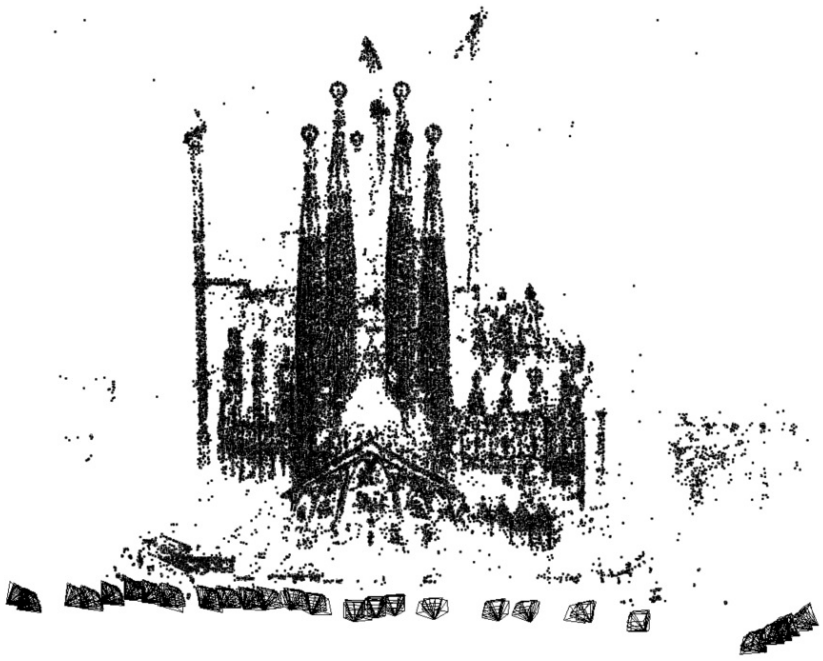
Set	resolution [MPix]		Octave	# keypoints		
	original	resized	Ours+	Bundler	Ours-	Ours+
1	0.94	0.94	−1	981	**1196**	**1196**
2	10.04	2.51	0	8982	9594	**9936**
3	10.04	2.51	0	8530	8339	**8728**
4	5.63	5.63	0	2564	**5137**	2476
5	14.16	1.77	1	18123	**20632**	10608
6	7.08	1.77	0	8670	10900	**11399**
7	14.16	1.77	1	10427	**13473**	7753
8	4.92	1.23	0	**14705**	13020	14270

Table 4.2 The resolution of the original images, of the resized images and the first octave chosen by Ours+. Also, the average number of detected keypoints on the resized images by Bundler and our approach without the scaling extension and on the original images by our approach with the scaling extension.

In Tab. 4.1 we show results for running Bundler, Ours- and Ours+ on the data described earlier. Both Bundler and Ours- have the advantage of manually resized images whereas Ours+ relies on an automatic detection of the first octave by our scaling extension. You can see that we manage to recover the same number of cameras for data sets 1-7. As far as the data set 8 is concerned, our pipeline performs rather well in comparison to Bundler since we are able to recover all of the cameras (50) whereas Bundler only 21. Next, we notice that for all the data sets except for the data set number 4, we recover all of the cameras. Comparing the reprojection error of Bundler and Ours- for the data sets 1-7, we see that even though we are better only for two data sets, the errors are typically very close to those of Bundler. Considering Ours+, we observe higher reprojection error than the other two approaches. Since the difference between Ours- and Ours+ is only the keypoint detection, we hypothesise that detecting different keypoints can result in different matches which can place different constraints on the optimization and that can result in a different reprojection error. Nevertheless, the error is still of the same order and the reconstruction was finished successfully (see Fig. 4.1 and Fig. 4.2).



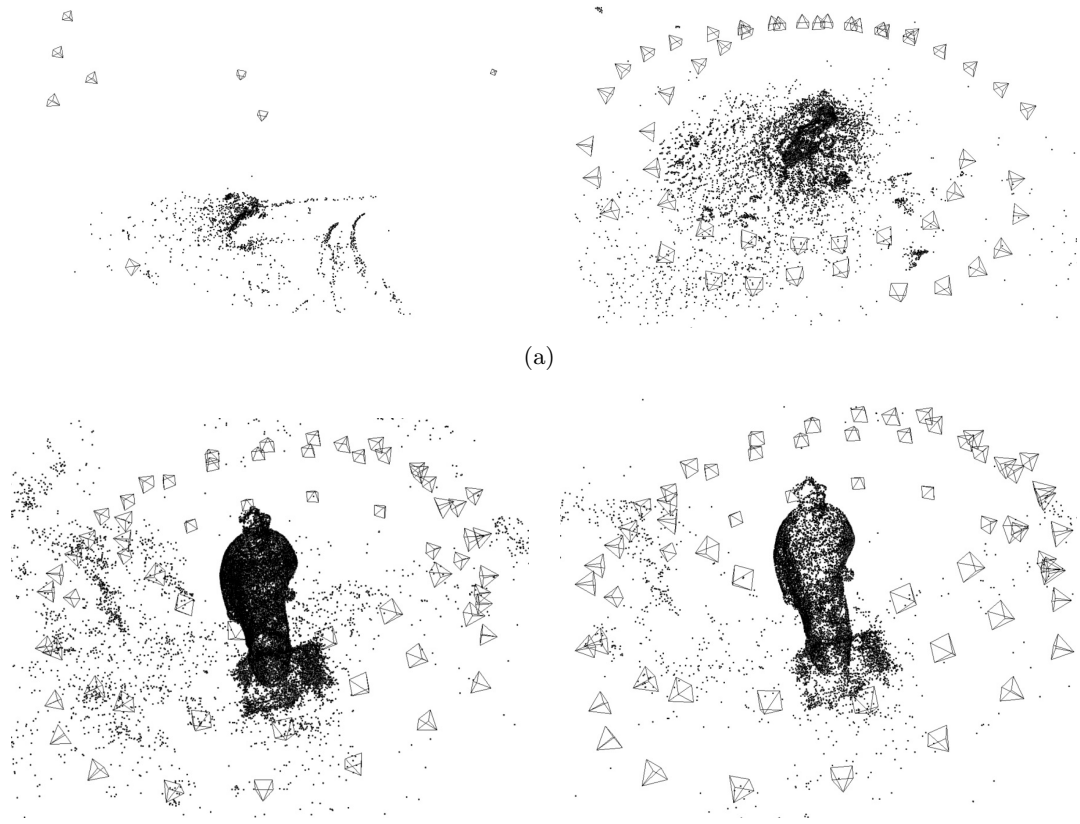
(a)



(b)

Figure 4.1 Visualizations of reconstructions of the data set 6. In (a), we see a result of Bundler and in (b), a result of Ours+ approach. The square pyramids represent cameras. A camera center is in a top of its pyramid and an orientation of the camera is given by a base of its pyramid. Note that the clutter at the bottom are cameras since it is a large church and all photos were taken from the ground. We see how our pipeline performs just as well as Bundler.

Reconstructions of data sets for which we recover the same number of cameras as Bundler are very similar. As a typical example of this, see a visualization of reconstruc-



(a)

(b)

Figure 4.2 Visualizations similar to Fig. 4.1. In (a), results of Bundler are shown on the left and Ours+ on the right and in (b) reconstruction done by Ours- on the left and the one done by Ours+ on the right. (a) displays reconstructions of the data set 8. On the left, you see how Bundler failed due to a lack of information in EXIF tags of images of this set. Whereas on the right, you see how our pipeline recovered all the cameras. (b) demonstrates how detecting less keypoints influences a reconstruction process for the data set 5. Concretely, Ours- reconstructed 49 731 3D points and Ours+ 20 329 3D points.

tions of the data set 6 in Fig. 4.1. You can even spot by eye how the reconstruction done by Bundler and the one done by our pipeline look alike which shows that we perform just as well as Bundler. Next, we visualize reconstructions of the data set 8, for which we outperform Bundler, in Fig. 4.2(a). We have investigated why we are able to return so much better result in this particular case. It is caused by a lack of information in EXIF tags of images in 8 which are used for estimating initial focal lengths of cameras. Therefore we directly see how our improvement for data sets containing images with no EXIF tags detailed in Sec. 3.2.5 really makes a difference in the final result.

In Tab. 4.2, we compare the number of detected keypoints by Bundler and Ours- on the resized images and by Ours+ on the original images. Comparing results of Bundler and Ours- procedures, we see that for most of the data sets our approach yields in average more keypoints. Even though we detect less keypoints on data sets 3 and 8, we argue that the difference between our results and Bundler's is not so large to fatally influence the reconstruction as a whole as you can see in Tab. 4.1. To evaluate our scaling of the SIFT detector extension we compare Ours+ approach to Ours- and Bundler. Firstly, consider Tab. 4.2 which compares the numbers of detected keypoints

by the approaches. We point out how the scaling extension utilized by Ours+ algorithm automatically chooses the first octave such that in average less than 20000 keypoints is successfully detected for every data set. Next, we see that Ours+ approach detects in average more keypoints than Bundler and Ours- procedure for half of the data sets and less keypoints for the other half. A special case is the data set 4, for which Ours- detected twice more keypoints than Ours+. Ours+ has automatically chosen the octave 0 as a starting one which is equivalent to resizing the original images to half of their sizes. On the other hand, Ours- utilized original images for the SIFT detection.

We hypothesise that as a result of detecting different numbers of keypoints, the numbers of matches could be different which could possibly result in better or worse reconstruction. Especially, less 3D points is typically reconstructed when less matches is provided but note that that does not yet mean worse resulting parameters of recovered cameras. We demonstrate this on the data set 5, where Ours- procedure detects almost twice more keypoints in average than Ours+. The resulting reconstruction by Ours- and Ours+ is visualized in Fig. 4.2(b). What we see is a sparser density of 3D points returned by Ours+ but still well estimated cameras. Since Ours+ and Ours- are the same except keypoint detection, we see the direct impact of detecting less keypoints.

4.1.2 Verification via PROSAC

In this subsection, we evaluate our implementation of Progressive Sample Consensus (PROSAC) which we use for estimating a fundamental matrix in the verification of matching phase (see Sec. 3.2.3). We first run our algorithm on benchmark data of Raguram *et al.* [48] and compare results achieved by our PROSAC implementation with [48]. Subsequently, we extensively evaluate the PROSAC on the real data used for 3D reconstruction in the previous chapter.

Our RANSAC randomly samples matches used to generate a hypothesis and subsequently generates it. Then it counts the number of inliers and if it is the highest number so far, it remembers the hypothesis. If a hypothesis supported by more than 95% of matches is found, the procedure is terminated. A total of 2048 hypothesis is tested if the algorithm is not terminated prematurely. The PROSAC implementation is described in detail in Sec. 3.2.3. Both of the algorithms use the threshold of 5 pixels to determine inliers.

Data of Raguram et al. In Tab. 4.3, we give results for the RANSAC and the PROSAC of [48] on their data designated to be used for fundamental matrix estimation. Additionally we provide results of our RANSAC and our PROSAC implementation on the same data. We note that [48] employs seven-point algorithm [24] whereas our implementation exploits eight-point algorithm [24]. Also note that we do not know the inlier threshold applied in [48]. The data consists of five matched image pairs. We denote them the same way as in [48], *i.e.*, A–E. The image pairs show a house, a part of a church, a street, a forest and two books.

We point out that the RANSAC of [48] has an adaptive stopping criterion. On the other hand, ours tests a predefined number of models. To give comparable results, we set the number of models to test by our RANSAC to the number of models that the RANSAC of [48] needed, *i.e.*, also the maximum number of tested models by our PROSAC is set accordingly. Nevertheless, note that to get our results and those of [48], a different computer was used. Therefore the time of our approaches and those of [48] is not comparable. The time is averaged over 500 runs of the algorithms.

Set		Raguram *et al.* [48]		Ours	
		RANSAC	PROSAC	RANSAC	PROSAC
A	# inliers	1412	645	894	1478
	# models	3499	10	3499	348
	time	255.90	2.01	259.73	53.72
	# matches	3154			
B	# inliers	315	115	156	227
	# models	941	17	941	99
	time	14.98	1.35	28.23	7.47
	# matches	575			
C	# inliers	381	295	361	371
	# models	19578	9	19578	1956
	time	546.53	5.41	664.63	78.82
	# matches	1088			
D	# inliers	324	313	321	319
	# models	661141	16	661141	66113
	time	23892.20	1.77	26189.60	2724.00
	# matches	1516			
E	# inliers	685	588	602	613
	# models	34	3	34	33
	time	0.85	0.12	76.52	17.22
	# matches	786			

Table 4.3 The number of found inliers, the number of tested models, the total time in milliseconds and the number of initial matches. Every data set (A–E) is a pair of matched images. All the data is the courtesy of Raguram *et al.* [48]. The table compares the results achieved by [48] and those achieved by our implementations (see Sec. 3.2.3). Note that to get our results and those of Raguram *et al.*, a different computer was used. All of the results are averaged over 500 runs of the algorithms.

To evaluate a PROSAC, we compare its results with those of its respective RANSAC (see Tab. 4.3). Considering results of [48], we see that their PROSAC procedure is 7–127 times faster than their RANSAC for 4 data sets and 13 498 times faster for the data set D. On the other hand, our PROSAC implementation estimates a fundamental matrix 3.5–9.5 times faster than our RANSAC. Even though our PROSAC introduces a smaller speedup than the PROSAC of [48], we note that our PROSAC always finds more inliers. For the data sets A and B even twice more which is a significant number.

Our data We use our pipeline with a configuration utilizing the scaling of the SIFT detector extension (see Sec. 3.2.1), VLFeat keypoint detector [61] and OpenCV [9] FLANN for matching. We let our pipeline detect keypoints and match them to get tentative (not verified) matches. Then, we compare a verification of the matches using epipolar geometry in a RANSAC and a PROSAC procedure.

Results for running both of the algorithms on the data presented earlier in previous sections (see Sec. 4.1.1) can be seen in Tab. 4.4. To provide better results for these procedures utilizing randomness, we average the results over 25 runs of the algorithms. We observe how PROSAC speeds up the process of verification. In some cases, the PROSAC requires only half of the time needed by RANSAC and it is never slower. Note that we include time needed for sorting matches in the PROSAC time shown in

Set		RANSAC	PROSAC	Set		RANSAC	PROSAC
1	# inliers	65	64	2	# inliers	392	376
	# pairs	842	839		# pairs	61	60
	# models	1878	1109		# models	2048	1256
	time	27.18	18.50		time	6.38	3.06
	# matches	66			# matches	418	
	# pairs	954			# pairs	78	
3	# inliers	551	514	4	# inliers	128	116
	# pairs	1454	1450		# pairs	316	320
	# models	1757	1047		# models	2048	1109
	time	158.41	117.31		time	14.10	7.85
	# matches	512			# matches	137	
	# pairs	1639			# pairs	394	
5	# inliers	218	183	6	# inliers	576	531
	# pairs	528	528		# pairs	7052	7050
	# models	2038	961		# models	1973	723
	time	24.24	12.40		time	720.14	421.82
	# matches	269			# matches	616	
	# pairs	580			# pairs	8253	
7	# inliers	481	450	8	# inliers	328	299
	# pairs	305	303		# pairs	235	234
	# models	2048	626		# models	1872	632
	time	27.90	12.30		time	17.53	9.98
	# matches	676			# matches	364	
	# pairs	321			# pairs	247	

Table 4.4 The average number of inliers found for an image pair and the total number of matched image pairs after verification. The average number of tested models for an image pair and time in seconds needed to verify all of the initially matched image pairs. The average number of initial matches for one image pair and the total number of initially matched image pairs. The results are provided for our RANSAC and for our PROSAC described in Sec. 3.2.3. All of the results are averaged over 25 runs of the algorithms.

the table. Related to the required time, we see that the PROSAC always tests less models (hypotheses) in average than the RANSAC which is the main reason for being faster. Unfortunately, the speed up has a cost in a form of finding less inliers in average. This sometimes results in finding less image pairs. We hypothesise that such a pair must have been weakly connected since a small decline in the number of inliers decreased the number of matches below the acceptable minimum.

We notice that our PROSAC speeded up our RANSAC more for the data of [48] than for our data. To explain this, note that the data of [48] are more contaminated by outliers than our data, which has ultimately effect on the speedup.

4.1.3 Bundle adjustment

This section describes evaluation of two bundle adjustment packages incorporated into our pipeline. First of them is the sparse bundle adjustment library (sba) of Lourakis and Argyros [38] which is also utilized by Bundler [55]. The other one is the CMP version of bundle adjustment [4] based on Ceres [1]. For evaluation, consider our pipeline

Set	# cams		# points		Error		Time	
	sba	Ceres	sba	Ceres	sba	Ceres	sba	Ceres
1	**64**	**64**	**6303**	6284	0.359	**0.320**	70.76	**11.16**
2	**13**	**13**	**11833**	11413	**0.325**	0.487	39.33	**4.45**
3	**122**	**122**	**127929**	127822	0.419	**0.388**	3670.10	**611.05**
4	**21**	**21**	**6373**	6326	**0.500**	0.506	23.25	**3.53**
5	**63**	**63**	**21521**	20329	0.670	**0.664**	291.30	**36.29**
6	**189**	**189**	**92177**	83551	**0.598**	0.627	15132.70	**1421.63**
7	**31**	**31**	**24294**	21402	**0.787**	0.794	221.69	**46.43**
8	48	**50**	17836	**23922**	1.081	**0.322**	998.13	**33.19**

Table 4.5 The results of running our pipeline with two different bundle adjustment packages. In more detail, the number of recovered cameras, the number of reconstructed 3D points, the average reprojection error and the time in seconds required by a bundle adjustment package alone.

in a configuration using the scaling of the SIFT extension (see Sec. 3.2.1), the VLFeat keypoint detector [61], the OpenCV [9] FLANN for matching and verification exploiting our PROSAC (see Sec. 3.2.3). That is basically the configuration defined as *Ours+* earlier in this part of the thesis, only without the restriction on a bundle adjustment package.

We let our pipeline detect keypoints, match them, verify them and then we run Structure from Motion (SfM) with different bundle adjustment modules. That is done for all 8 data sets described in the beginning of this chapter. We have implemented the SfM so that we can change a bundle adjustment module with a simple switch. That way we can directly see the impact of using different bundle adjustment packages. The results are shown in Tab. 4.5. As far as the data sets 1-7 are concerned, we see that not considering the time, the packages perform similarly well. The number of recovered cameras is the same for both of the modules, the number of points reconstructed by sba is always better but not significantly and the reprojection error is sometimes better for one package and sometimes for the other but the differences are negligible. The 8th data set is an exception since sba was able to connect only 48 cameras whereas Ceres-based CMP package was able to reconstruct all 50 cameras. Relating to that, sba reconstructed less 3D points and its reprojection error is one order higher than the one of CMP module. Finally, considering the time issue, the Ceres-based bundle adjustment is always more effective. It is four to ten times faster for the data sets 1-7 and thirty times faster for the 8th data set.

4.2 Indoor reconstruction

In this section, we evaluate our improvement for indoor scenes on 10 different data sets. These data sets contain 101, 75, 116, 129, 79, 79, 98, 70, 57 and 492 images. Lets consider labelling of the sets 10, 11,. . . , 19 as in Tab. 4.1. Then, the data sets visualize: living rooms in 10 and 14, a bathroom with a large mirror on a wall in 11, kitchens in 12 and 17, general rooms in 13 and 15, a library in 16 and an empty room with white walls in 18. Finally, the number 19 is a gallery data set of Furukawa *et al.* [21] consisting of multiple rooms. Some of these data sets are very challenging as you will subsequently see.

Set	# images	# cams			
		sba		Ceres	
		Std	Indoor	Std	Indoor
10	101	**100**	**100**	**100**	**100**
11	75	29	**31**	30	**38**
12	116	102	**107**	**104**	59
13	129	**89**	83	**103**	99
14	79	75	**77**	72	**76**
15	79	9	**55**	9	**10**
16	98	**95**	**95**	**94**	**94**
17	70	**70**	**70**	**70**	**70**
18	57	9	**20**	9	**20**
19	492	290	**373**	**484**	414

Table 4.6 The number of images and the number of recovered cameras for two bundle adjustment packages.

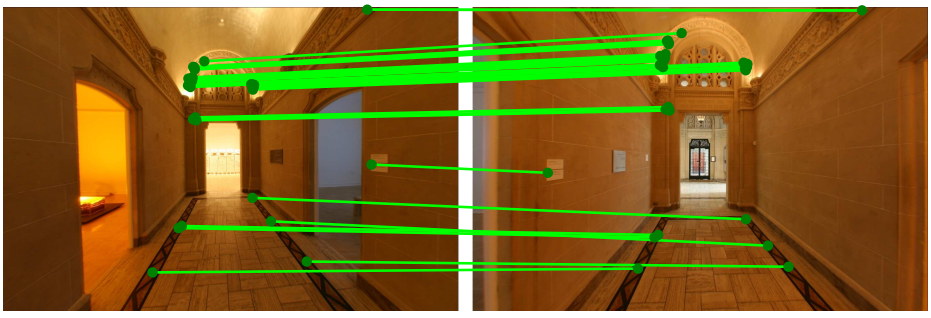
Set	# points				Error			
	sba		Ceres		sba		Ceres	
	Std	Indoor	Std	Indoor	Std	Indoor	Std	Indoor
10	15741	**34002**	15772	**33563**	**0.425**	0.494	**0.434**	0.451
11	7230	**9614**	7354	**10358**	**0.435**	0.498	**0.467**	0.491
12	25037	**35297**	**21539**	16487	0.978	**0.618**	1.044	**0.675**
13	**29302**	25193	34618	**39030**	**0.705**	3.442	**0.564**	0.623
14	20847	**27888**	21518	**30547**	**0.534**	0.614	**0.540**	0.569
15	668	**12135**	588	**2801**	0.594	**0.438**	0.576	**0.450**
16	34691	**63126**	33414	**63383**	**0.714**	0.737	**0.703**	0.717
17	14338	**23683**	14321	**23804**	**0.719**	0.812	**0.727**	0.799
18	867	**2084**	867	**2022**	**0.417**	0.858	**0.410**	0.801
19	35270	**59820**	69677	**73013**	**0.924**	1.162	**0.392**	0.656

Table 4.7 The number of reconstructed 3D points and the reprojection error for two bundle adjustment packages.

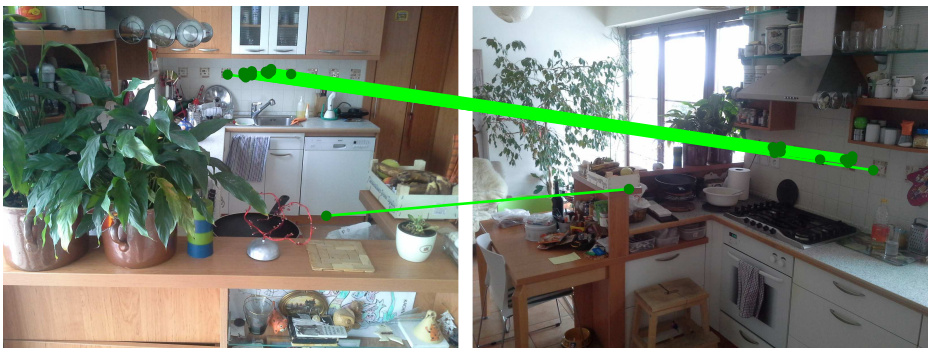
For the following subsections, consider our pipeline with 2 configurations. Both of them utilize the scaling of the SIFT extension (see Sec. 3.2.1), the VLFeat keypoint detector [61], the OpenCV [9] FLANN for matching and verification exploiting our PROSAC (see Sec. 3.2.3). The only difference being that one of them exploits the sparse bundle adjustment library (sba) of Lourakis and Argyros [38] and the other one is the CMP version of bundle adjustment [4] based on Ceres [1]. Next, we denote standard runs of our pipeline by *Std* and runs utilizing the indoor extension (see Sec. 3.3) by *Indoor*. We first provide quantitative results before subsequently showing qualitative results on the aforementioned data.

4.2.1 Quantitative evaluation

In Tab. 4.6 and Tab. 4.7, we compare results of the reconstruction phase. We provide the number of recovered cameras, the number of reconstructed 3D points and the aver-



(a)



(b)

Figure 4.3 Typical examples of image pairs mismatched due to the occurrence of repetitive structures. The green lines represent verified matches and the dots respective keypoints. (a) shows two different ends of the same hallway and (b) visualizes two different walls of the same kitchen incorrectly matched via tiles.

age reprojection error achieved by either of the methods. Additionally, we show results for both of the incorporated bundle adjustment packages since they perform quite differently in some cases. Out of 20 cases, we are able to recover more cameras with the indoor extension in 10, the same number in 6 and surprisingly, less in 4 cases. Except for two failures, we always reconstruct more 3D points and, considering the reprojection error as a metric, we perform worse in 16 cases and better in 4 cases.

We have investigated the cases in which our extension produced a lower number of recovered cameras and found that the failures have 2 causes. Firstly, since our indoor extension detects keypoints located on planes, it is possible that all (or most of) matches of an image pair are located on a single plane. That can cause the verification phase which utilizes the eight-point algorithm [24] and does not calibrate cameras to estimate a degenerate epipolar geometry which does not reject some mismatches. Secondly, if a scene generates by its nature mismatches caused by repetitive structures, our approach can simply reinforce the mismatches. The mismatches can then influence the SfM. See Fig. 4.3 for typical examples of image pairs mismatched due to the occurrence of repetitive structures. To explain the influence on the SfM, we emphasize that we utilize an incremental SfM algorithm, *i.e.*, the procedure iterates over cameras and always adds only a subset of them having enough matches to already reconstructed 3D points. Therefore, if a camera connected by mismatches is added to the reconstruction, the whole process might irreversibly deteriorate. To fully exploit the benefit of our contribution in SfM pipelines, the problem of dealing with repetitive structures has

Set	# matches				# pairs			
	Tentative		Verified		Tentative		Verified	
	Std	Indoor	Std	Indoor	Std	Indoor	Std	Indoor
10	102	**166**	100	**168**	890	**1120**	850	**1010**
11	121	**130**	115	**137**	405	**510**	314	**338**
12	**144**	139	137	**195**	972	**1709**	766	**840**
13	**178**	**178**	206	**243**	1238	**1641**	818	**878**
14	197	**199**	184	**214**	550	**760**	464	**522**
15	94	**121**	88	**113**	279	**313**	232	**249**
16	272	**361**	316	**481**	1846	**2276**	1282	**1312**
17	188	**284**	168	**245**	1183	**1278**	1003	**1062**
18	80	**128**	64	**89**	195	**222**	138	**156**
18	95	**137**	92	**129**	6415	**6968**	5336	**5920**

Table 4.8 The average number of tentative and verified matches and the total number of matched image pairs before and after verification. The results are shown for a standard run of our pipeline described in Sec. 3.2 and a run utilizing the indoor extension from Sec. 3.3.

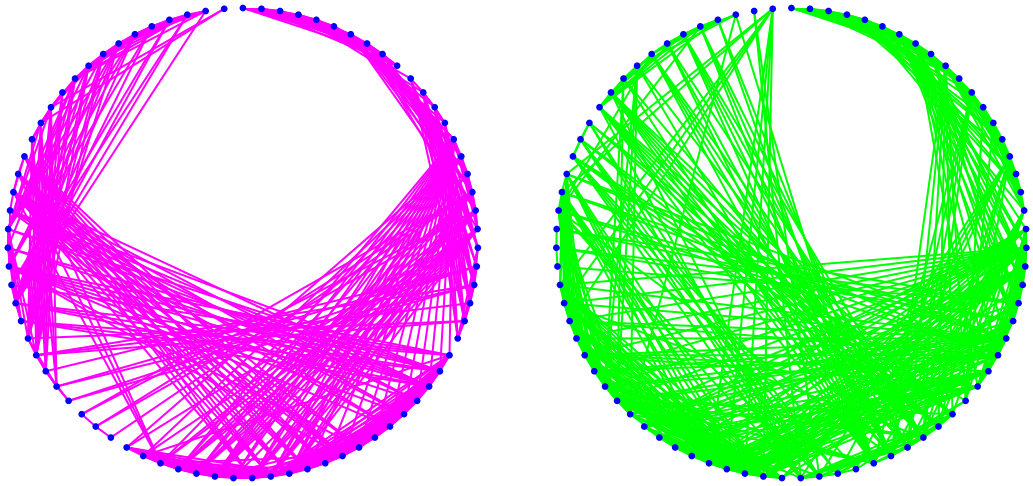
Set	Component size	Graph diameter	
		Std	Indoor
10	101	7	**5**
11	75	12	**10**
12	115	**7**	**7**
13	123	8	**7**
14	79	6	**5**
15	(72 + 1)	(9 + 0)	**6**
16	98	**6**	**6**
17	70	5	**4**
18	10, (23 + 13), 4, 4	**4**, (5 + 6), **2**, **2**	**4**, **10**, **2**, **2**
18	486	21	**19**

Table 4.9 The size and the graph diameter of all connected components of size larger than 2. Individual components are separated by ','. If a component was disconnected utilizing the standard approach, we show the sizes and the diameters of all sub-components that got connected in parenthesis, separated by '+'. The graphs are constructed from matched image pairs that were accepted by the epipolar geometry verification.

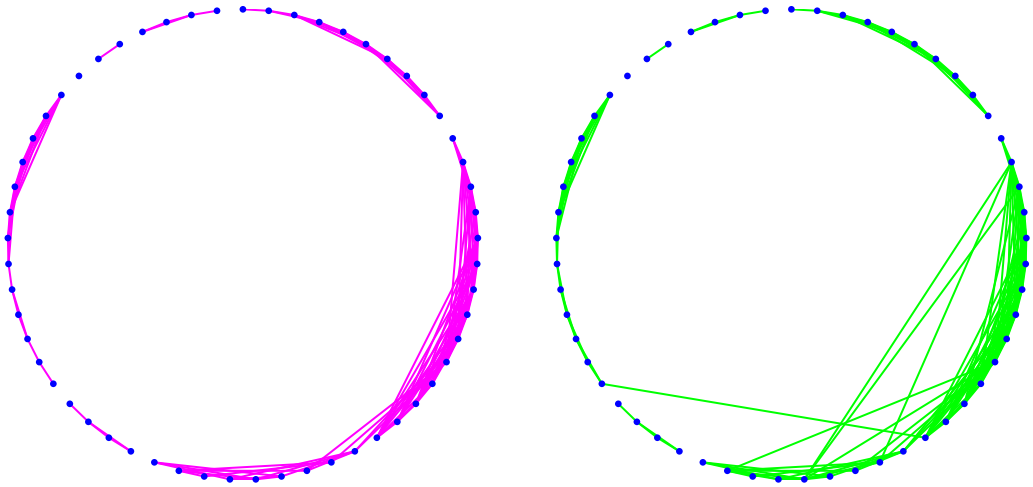
to be addressed. Also, matches have to be verified by an algorithm which does not estimate a degenerate epipolar geometry from matches located on a single plane. Such an algorithm should employ camera calibration.

To explain the worse reprojection error for successfully reconstructed data sets, we hypothesise that one cause is adding new cameras and a second one is adding new matches which can result in reconstructing more 3D points. In both situations new constraints to the bundle adjustment are added and that can make it harder to get lower reprojection error. We point out that higher reprojection error does not automatically mean worse reconstruction. Unfortunately, we do not have ground truth data for the data sets and therefore we are not able to compute the true reconstruction error.

To improve the reconstruction process, we want connections between images to be



(a)



(b)

Figure 4.4 Nodes depicted as the blue dots represent images. The magenta lines represent matched and verified image pairs of the standard method and the green lines those of our approach. In (a) we show the data set 14. The nodes are ordered in a way to nicely illustrate the connectedness of the graphs. In (b) we visualize the data set 18. The nodes in both graphs have the same ordering which allows observing the individual connected components and how two standardly disconnected components got connected by our approach.

strong, *i.e.* we want to have as many matches as possible. In Tab. 4.8, we show the number of tentative and verified matches averaged over all image pairs. We note that our approach utilizes matches found by the indoor extension as well as standard matches. Therefore the tentative number of matches shows how much our indoor extension reinforces the standard matches. To explain, why our method finds less tentative matches for one data set and the same number of matches for another data set, we point out that our approach connected many more image pairs than the standard procedure, hence the number of matches is averaged over more image pairs. Considering the number of verified matches, we observe that our approach performs better than the standard method for all the data sets. That means that matches found by the indoor extension are not spurious since they overcome the verification phase.

Additionally, it is desired that images are connected to as many other images as possible. Consider a graph where nodes represent images and an edge exists if an image pair was matched and accepted by the epipolar geometry verification; we want the graph to be connected as tightly as possible. In Tab. 4.8 we show the total number of matched image pairs and we see that our proposed approach outperforms the standard one. In Fig. 4.4 we observe how the graphs based on our matches are more connected than those of the standard approach. For capturing the connectivity of a graph, we propose to use the graph diameter, *i.e.* the maximum length of all shortest paths. Intuitively, a graph's diameter is the largest number of vertices that must be traversed in order to travel from one node to another. We provide the diameter in Tab. 4.9 for all connected components of size larger than four. A diameter of connected components is measured as if the component were regular graphs. According to the diameters, we conclude that graphs created from our matches are more connected or have the same level of connectedness than those of the standard approach.

4.2.2 Qualitative evaluation

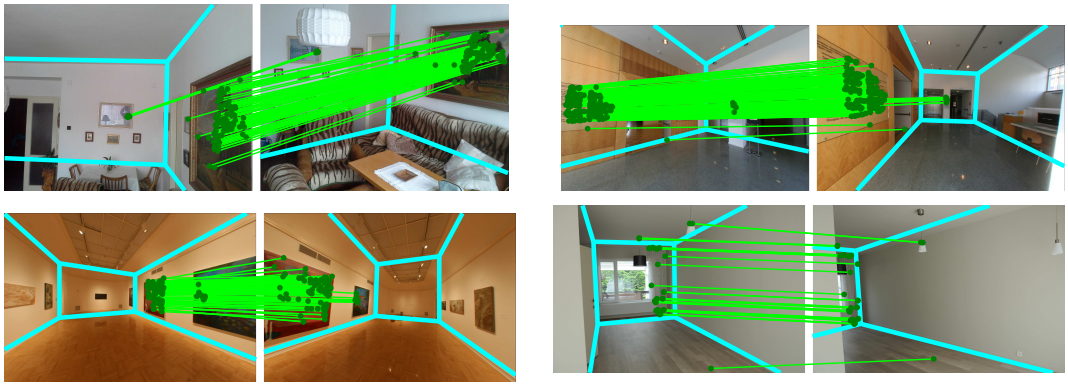
Consider again a graph where nodes represent images and an edge exists if an image pair was matched and accepted by the epipolar geometry verification. It is impossible to reconstruct a whole data set if its graph contains multiple connected components instead of a completely connected graph. Hence, our goal is to reduce the number of connected components to a minimum. Our proposed approach is able to address this problem. See Fig. 4.4(b) for a graph which contains 7 connected components when matched by the standard procedure and 6 when utilizing our method.

Having shown that our method outperforms the standard approach, we next show some typical examples.

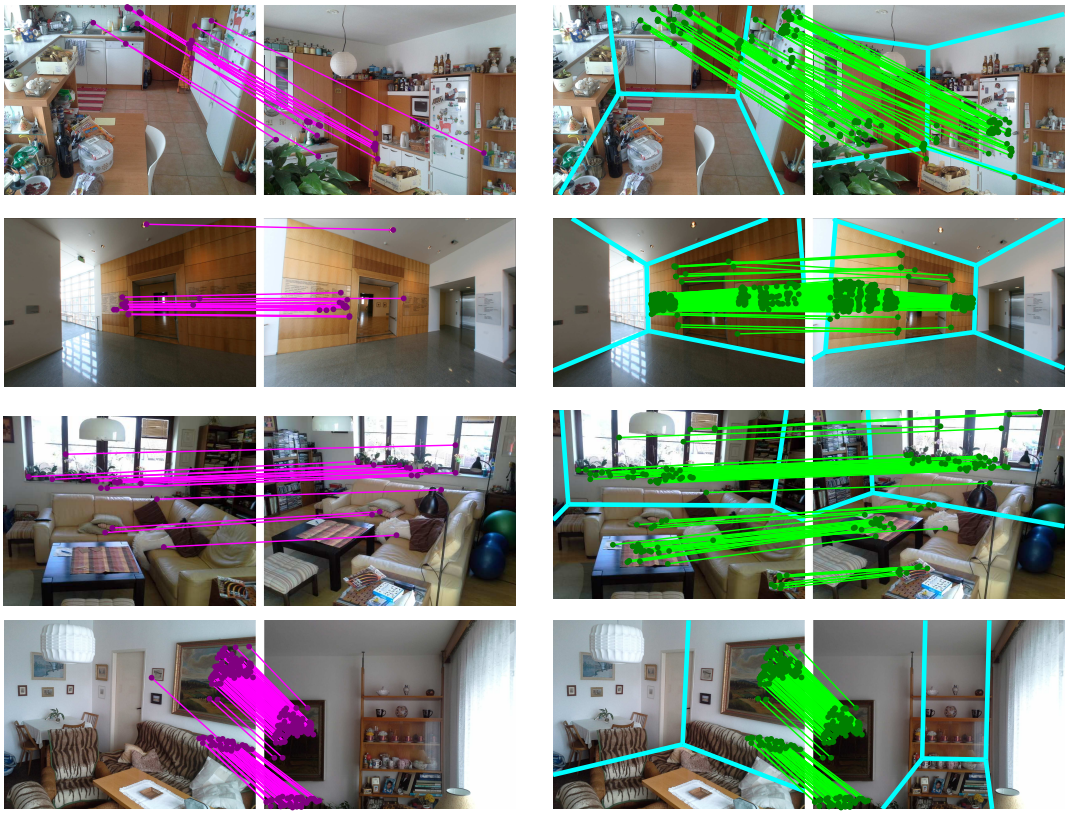
In Fig. 4.5(a) we show image pairs with verified matches obtained by our proposed approach. The standard procedure could not find matches good enough so they would not be rejected by the verification process. This is not surprising considering the large transformations between the images. For completeness we enhance the figure with the estimated room layouts.

In Fig. 4.5(b) we visualize images connected by verified matches found by the standard procedure and compare them to our method improving the indoor matching. We see that even when the scene estimation failed, our proposed approach utilizes matches originated from the standard matching and thus does not fail.

In Fig. 4.6 we show reconstructions of the data set 15 done by the standard and our approach, both utilizing sba bundle adjustment package. We see how the standard method recovered only 9 cameras all of which viewing one wall whereas our approach reconstructed 55 cameras situated all around the scene. Next, we present the data set 12 which contains repetitive structures. That is why some of the additional matches provided by the indoor extension are mismatched. Nevertheless, in this case, sba deals with the mismatches and provides much better reconstruction with the extra matches. This is shown in Fig. 4.7. The top view nicely visualizes how the shape of the reconstructed room looks like. On the other hand, in Fig. 4.8 we see how the Ceres based bundle adjustment was affected by the mismatches and provided worse reconstruction than without the additional matches.

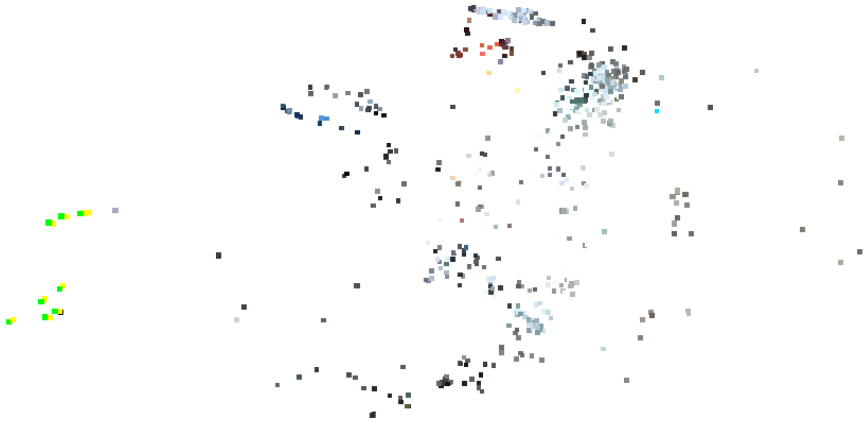


(a)

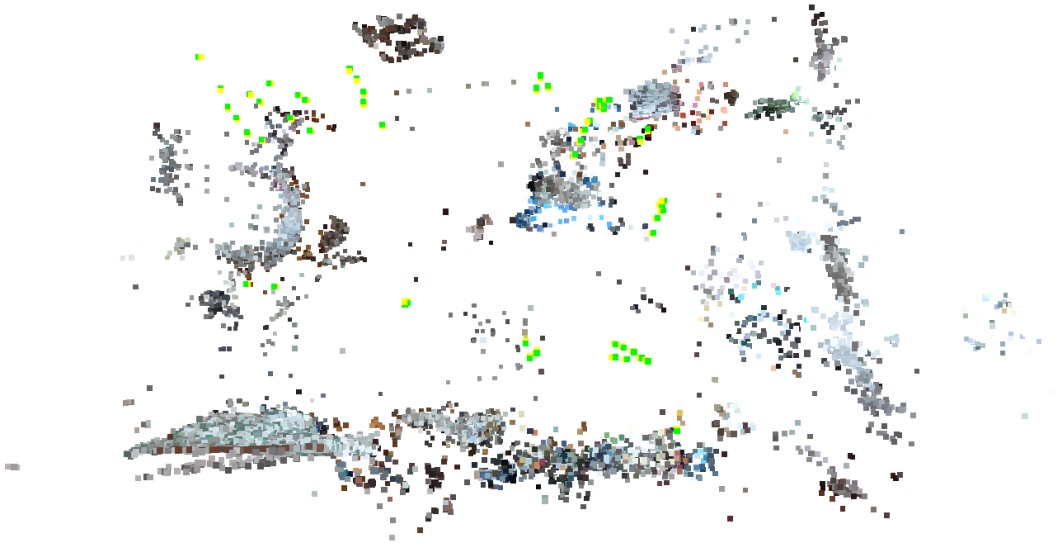


(b)

Figure 4.5 The visualizations above show matched image pairs. The bright green lines depict verified matches generated by our approach, the bright magenta lines symbolize verified matches originated from the standard method and the dark green and magenta dots represent locations of related keypoints. The cyan color illustrates detected scene layout. All of the image pairs in (a) could not be matched by the standard approach at all whereas we have found 94, 190, 91 and 38 matches respectively. The image pairs in (b) located on the same row show the same image pair, only matched by different methods. We observe how our approach exploits knowledge of the scene if it is estimated correctly and retrieves more matches (the standard approach found 19, 17, 17 and 159 matches and we obtained 100, 344, 78 and 174 respectively).

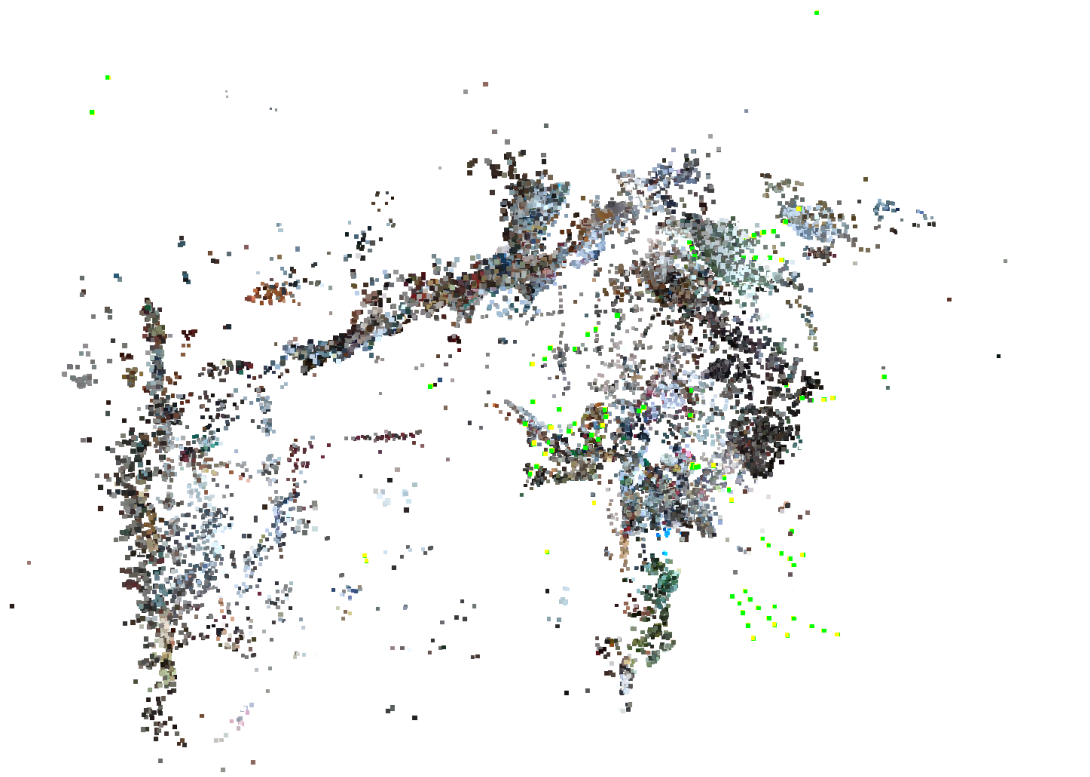


(a)

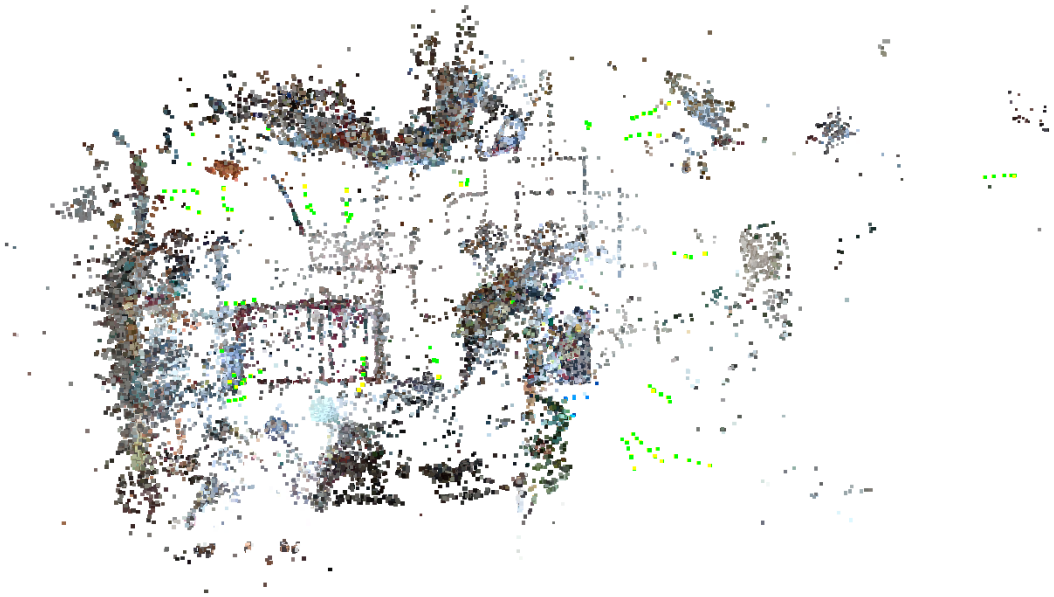


(b)

Figure 4.6 The top view of reconstructed indoor scenes. (a) visualizes reconstruction done by the standard method and (b) the one done by our proposed approach. The package sba bundle adjustment was utilized to get both of the results. What we see are reconstructed 3D points colored according to the input images. Additionally, we visualize recovered cameras. A camera is represented by a pair of green and yellow dots. We show only the top view because the sparse reconstructions of these indoor scenes do not give a visually appealing model when zoomed in. The top view illustrates how well was a room reconstructed. For instance, perpendicular walls indicate a good model. This particular example shows the reconstruction of the data set 15.

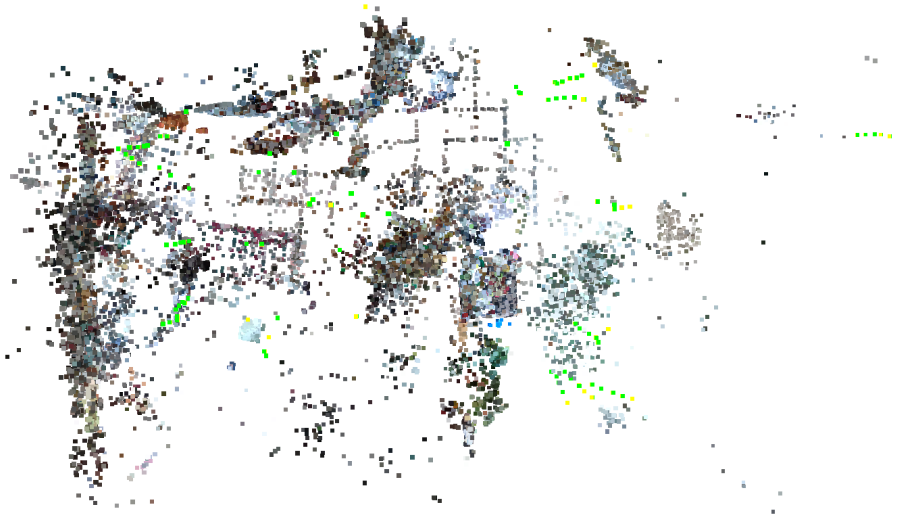


(a)

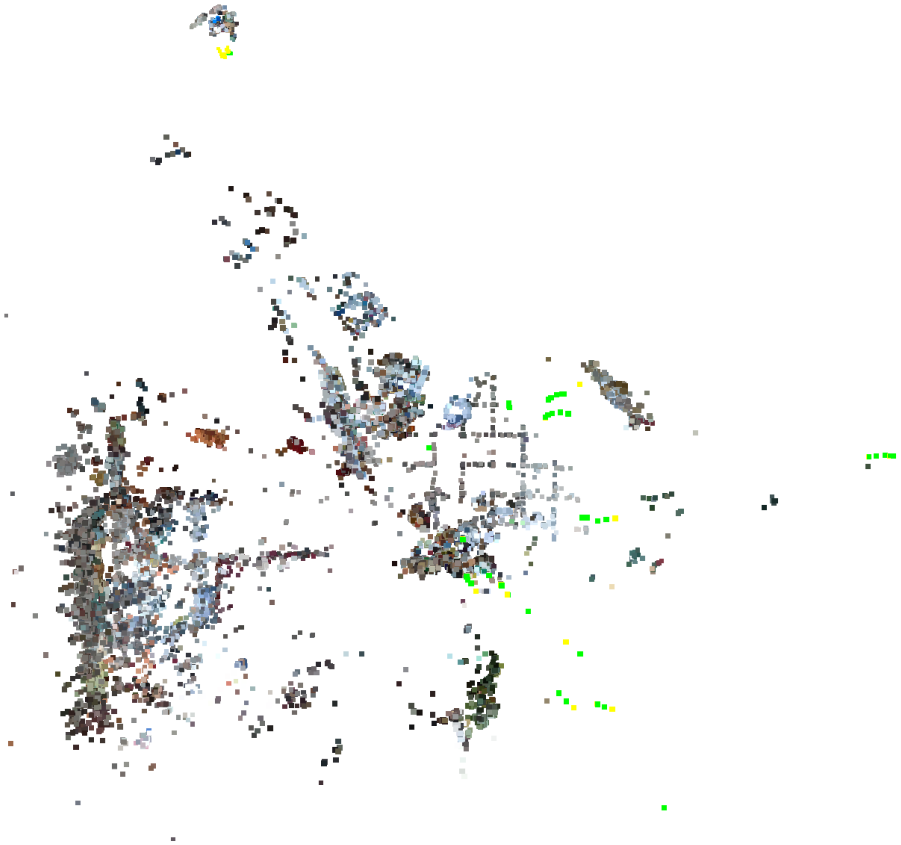


(b)

Figure 4.7 Visualizations of reconstructed indoor scenes similar to Fig. 4.6. The visualizations are reconstructions of the data set 12.



(a)



(b)

Figure 4.8 Reconstructions of indoor scenes similar to Fig. 4.6 with the only difference being that the Ceres based bundle adjustment was used to obtain the results. This figure shows reconstructions of the data set 12.

5 Conclusion

This work describes a well-known 3D reconstruction pipeline and builds upon this explanation to detail the implementation of our own pipeline which uses different modules and introduces three notable improvements to the reference pipeline. Firstly, we presented an approach that automatically detects a reasonable number of keypoints in an image based on a resolution of the image. Next, we introduced our implementation of Progressive Sample Consensus (PROSAC) which is employed instead of the standard Random Sample Consensus (RANSAC) for estimating an epipolar geometry of an image pair. Finally, we proposed an improvement for sets of images for which a focal length cannot be estimated for any image of a set. Then, we showed that our pipeline performs just as well as the reference pipeline, and in some cases even better.

Furthermore, this work details, as the title suggests, an improvement of 3D reconstruction for indoor scenes. We have proposed employing scene understanding to improve image matching. Importantly, we followed a human-like approach of firstly reconstructing a global scene representation before matching details which are commonly represented via features that focus on image transformations. We demonstrated that our method outperforms the standard matching procedure on challenging indoor data sets. Nevertheless, we note that two problems have to be addressed to fully exploit the benefit of our pipeline since our method can reinforce mismatches or match only keypoints located on a single plane. First, matches have to be verified by an algorithm which does not estimate a degenerate epipolar geometry from matches located on a single plane. Sendond, the Structure from Motion (SfM) algorithm has to address the problem of dealing with repetitive structures.

Bibliography

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. https://code.google.com/p/ceres-solver/. 5, 18, 25, 31, 33

[2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, pages 72–79. IEEE Computer Society, 2009. 5

[3] Amir Akbarzadeh, Jan-Michael Frahm, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, Paul Merrell, M. Phelps, Sudipta N. Sinha, B. Talton, Liang Wang 0002, Qingxiong Yang, Henrik Stewénius, Ruigang Yang, Greg Welch, Herman Towles, David Nistér, and Marc Pollefeys. Towards urban 3d reconstruction from video. In *3DPVT*, pages 1–8. IEEE Computer Society, 2006. 5

[4] Cenek Albl and Tomas Pajdla. Global camera parameterization for bundle adjustment. In *VISAPP*, volume 3, pages 555–561, 2014. 5, 18, 25, 31, 33

[5] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998. 5, 8, 15

[6] Adam Baumberg. Reliable feature matching across widely separated views. In *CVPR*, pages 1774–1781. IEEE Computer Society, 2000. 6

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *CVIU*, 110:346–359, 2008. 6

[8] Alexander C. Berg, Tamara L. Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, volume 1, pages 26–33. IEEE Computer Society, 2005. 6

[9] Gary R. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 5, 13, 15, 24, 25, 30, 32, 33

[10] Matthew Brown and David G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *3DIM*, pages 56–63. IEEE Computer Society, 2005. 5

[11] C3 Technologies. Nokia Maps 3D. http://maps.nokia.com/, 2011. 2

[12] Gustavo Carneiro and Allan D. Jepson. Multi-scale phase-based local features. In *CVPR*, volume 1, pages 736–743. IEEE Computer Society, 2003. 6

[13] Ondrej Chum and Jiri Matas. Matching with prosac - progressive sample consensus. In *CVPR*, volume 1, pages 220–226. IEEE Computer Society, 2005. 2, 15

[14] Roberto Cipolla, Duncan Robertson, and Ben Tordoff. Image-based localisation. In *VSMM*, 2004. 7

[15] Andrea Cohen, Christopher Zach, Sudipta N. Sinha, and Marc Pollefeys. Discovering and exploiting 3d symmetries in structure from motion. In *CVPR*, pages 1514–1521. IEEE Computer Society, 2012. 6

[16] Nico Cornelis, Kurt Cornelis, and Luc J. Van Gool. Fast compact city modeling for navigation pre-visualization. In *CVPR*, volume 2, pages 1339–1344. IEEE Computer Society, 2006. 5

[17] David J. Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, pages 3001–3008. IEEE Computer Society, 2011. 5

[18] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE Computer Society, 2005. 2, 6

[19] M. Fischler and R. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. 1987. 2, 8, 11, 12, 15

[20] Jan-Michael Frahm, Pierre Fite Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, and Svetlana Lazebnik. Building rome on a cloudless day. In *ECCV*, volume 6314 of *Lecture Notes in Computer Science*, pages 368–381. Springer, 2010. 5

[21] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *ICCV*, pages 80–87. IEEE Computer Society, 2009. http://grail.cs.washington.edu/projects/interior/. 4, 7, 32

[22] Google. Google Earth. http://earth.google.com/, 2004. 2

[23] Google. Photo Tours in Google Maps. http://maps.google.com/phototours/, 2012. 2

[24] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2nd edition, 2003. 2, 5, 8, 12, 29, 34

[25] Varsha Hedau, Derek Hoiem, and David A. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, pages 1849–1856. IEEE Computer Society, 2009. 3, 7, 19, 22

[26] Varsha Hedau, Derek Hoiem, and David A. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, number 6 in Lecture Notes in Computer Science, pages 224–237. Springer, 2010. 7

[27] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, 2007. 7, 19, 21

[28] Derek Hoiem, Rahul Sukthankar, Henry Schneiderman, and Larry Huston. Object-based image retrieval using the statistical structure of images. In *CVPR*, volume 2, pages 490–497, 2004. 7

[29] Nianjuan Jiang, Ping Tan, and Loong Fah Cheong. Multi-view repetitive structure detection. In *ICCV*, pages 535–542. IEEE Computer Society, 2011. 6

[30] Nianjuan Jiang, Ping Tan, and Loong Fah Cheong. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In *CVPR*, pages 1458–1465. IEEE Computer Society, 2012. 6

[31] Manfred Klopschitz, Christopher Zach, Arnold Irschara, and Dieter Schmalstieg. Generalized detection and merging of loop closures for video sequences. In *3DPVT*, 2008. 5

[32] Kevin Koser, Christopher Zach, and Marc Pollefeys. Dense 3d reconstruction of symmetric scenes from a single image. In *DAGM-Symposium*, volume 6835 of *Lecture Notes in Computer Science*, pages 266–275. Springer, 2011. 6

[33] Avanish Kushal, Ben Self, Yasutaka Furukawa, David Gallup, Carlos Hernandez, Brian Curless, and Steven M. Seitz. Photo tours. In *3DIMPVT*, pages 57–64. IEEE Computer Society, 2012. 2

[34] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178. IEEE Computer Society, 2006. 6

[35] David C. Lee, Abhinav Gupta, Martial Hebert, and Takeo Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, pages 1288–1296. Curran Associates, Inc., 2010. 7, 19, 21

[36] David C. Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, pages 2136–2143. IEEE Computer Society, 2009. 7, 19, 21

[37] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, volume 5302 of *Lecture Notes in Computer Science*, pages 427–440. Springer, 2008. 5

[38] Manolis I. A. Lourakis and Antonis A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Softw.*, 36(1), 2009. 5, 12, 18, 31, 33

[39] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2, 5, 6, 8, 13, 19, 24

[40] Microsoft. Photosynth. `http://photosynth.net/`, 2008. 2

[41] Branislav Micusík and Jana Kosecka. Piecewise planar city 3d modeling from street view panoramic sequences. In *CVPR*, pages 2906–2912. IEEE Computer Society, 2009. 5

[42] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005. 6

[43] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, volume 1, pages 331–340. INSTICC Press, 2009. 15

[44] David Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):756–777, 2004. 5, 11, 18

[45] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999. 9

[46] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research*, 155, 2006. 3, 6

[47] Luca del Pero, Jinyan Guan, Ernesto Brau, Joseph Schlecht, and Kobus Barnard. Sampling bedrooms. In *CVPR*, pages 2009–2016. IEEE Computer Society, 2011. 7, 19, 21

[48] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: A universal framework for random sample consensus. *PAMI*, 35(8):2022–2038, 2013. http://www.cs.unc.edu/~rraguram/usac/. 15, 29, 30, 31

[49] Roberts Richard, Sudipta N. Sinha, Richard Szeliski, and Drew Steedly. Structure from motion for scenes with large duplicate structures. In *CVPR*, pages 3137–3144. IEEE Computer Society, 2011. 6

[50] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV*, volume 2350 of *Lecture Notes in Computer Science*, pages 414–431. Springer, 2002. 5

[51] Grant Schindler, Panchapagesan Krishnamurthy, Roberto Lublinerman, Yanxi Liu, and Frank Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*. IEEE Computer Society, 2008. 6

[52] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *CVPR*, pages 2815–2822. IEEE Computer Society, 2012. 7, 19, 21, 22

[53] Alexander G. Schwing and Raquel Urtasun. Efficient exact inference for 3d indoor scene understanding. In *ECCV*, volume 7577 of *Lecture Notes in Computer Science*, pages 299–313. Springer, 2012. 7, 19, 22

[54] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Trans. Graph.*, 30(6):154, 2011. 7

[55] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, pages 835–846, New York, NY, USA, 2006. ACM Press. vii, viii, 2, 5, 8, 10, 13, 14, 25, 31

[56] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008. 10

[57] Jean-Philippe Tardif, Yanis Pavlidis, and Kostas Daniilidis. Monocular visual odometry in urban environments using an omdirectional camera. In *IROS*, 2008. 5

[58] Kinh Tieu and Paul A. Viola. Boosting image retrieval. *IJCV*, 56(1-2):17–36, 2004. 7

[59] Akihiko Torii, Josef Sivic, Tomás Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *CVPR*, pages 883–890. IEEE Computer Society, 2013. 6

[60] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV, pages 298–372. Springer-Verlag, 2000. 5, 9

[61] Andrea Vedaldi and Brian Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms. `http://www.vlfeat.org/`, 2008. 13, 15, 25, 30, 32, 33

[62] Maarten Vergauwen and Luc J. Van Gool. Web-based 3d reconstruction service. *Mach. Vis. Appl.*, 17(6):411–426, 2006. 5

[63] M. Wandel. jhead: Exif Jpeg header manipulation tool. `http://www.sentex.net/~mwandel/jhead/`, 2001. 9

[64] Huayan Wang, Stephen Gould, and Daphne Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, volume 6312 of *Lecture Notes in Computer Science*, pages 435–449. Springer, 2010. 3, 7, 19

[65] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, pages 127–134. IEEE Computer Society, 2013. 5

[66] Changchang Wu, Jan-Michael Frahm, and Marc Pollefeys. Repetition-based dense single-view reconstruction. In *CVPR*, pages 3113–3120. IEEE Computer Society, 2011. 6

[67] Yahoo! Flickr: Online photo management and photo sharing application. `http://flickr.com/`, 2005. 5

[68] Christopher Zach, Arnold Irschara, and Horst Bischof. What can missing correspondences tell us about 3d structure and motion? In *CVPR*. IEEE Computer Society, 2008. 6