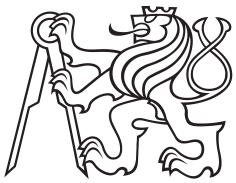
Bachelor's Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Investigating Visual Localisation Based on Semi-Generalized Camera Pose Estimation

Alena Smutná

Supervisor: RNDr. Zuzana Kúkelová, Ph. D.
Study Program: Cybernetics and Robotics
May 2023

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Smutná Alena** Personal ID number: **468845**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Investigating Visual Localization Based on Semi-Generalized Camera Pose Estimation

Bachelor’s thesis title in Czech:

Zkoumání vizuální lokalizace založené na semi-generalizovaném odhadu pozice kamery

Guidelines:

Visual localization is the problem of estimating the precise position and orientation from which a given image was taken. State-of-the-art localization algorithms rely on 3D models, typically constructed from a database of images, and estimate the camera pose by establishing matches between pixels in the query image and 3D points in the model. While this leads to highly accurate poses, building and maintaining a 3D model are complex tasks themselves. A lightweight alternative represents the scene as a set of database images with known camera poses and intrinsics. The pose of the query image is then estimated from 2D-2D correspondences between the query and two or more database images. Yet, in our experience, the resulting poses are less accurate than those obtained from 2D-3D matches. The goal of the thesis is to better understand this behavior and to investigate potential approaches to improve pose accuracy from 2D-2D matches.
In detail, the goals of the thesis are:
• To familiarize yourself with the problem of camera pose estimation from 2D-2D matches, with a focus on approaches for semi-generalized relative pose estimation.
• To investigate, through practical experiments, how the accuracy of the query pose depends on the configuration poses of the database images.
• If possible, use the results of this investigation to improve the accuracy of 2D-2D matching-based localization algorithms.

Bibliography / sources:

[1] E. Zheng, Ch. Wu. Structure from Motion Using Structure-Less Resection, IEEE International Conference on Computer Vision (ICCV), 2015
[2] S. Bhayani, T. Sattler, D. Barath, P. Beliansky, J. Heikkila, Z. Kukelova. Calibrated and Partially Calibrated Semi-Generalized Homographies, International Conference on Computer Vision (ICCV), 2021

Name and workplace of bachelor’s thesis supervisor:

RNDr. Zuzana Kúkelová, Ph.D. Visual Recognition Group FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **16.02.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

RNDr. Zuzana Kúkelová, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Acknowledgements

I would like to thank Zuzana Kúkelová and Torsten Sattler for guidance, helpfull advice and never-ending patience. Many thanks also belong to my family members who (sometimes unwillingly) served as debugging rubber ducks.

Declaration

I declare that presented work was developed independently and that I have listed all sources of information used within it in accordance with methodical instructions for observing the ethical principles in preparation of university thesis.

Prague, 26. May 2023

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

V Praze, 26. května 2023

Abstract

The goal of this thesis was to study the behaviour of E_{5+1} solver for the semi-generalized camera pose estimation problem. The thesis develops and presents experiments studying how the camera pose estimation accuracy is affected by the distance between the query and database cameras and by the distances between used database cameras. The results show that the pose estimation accuracy is mostly not dependent on the distance between query and database cameras, but the accuracy decreases when a small number of database cameras is used. The experimental results suggest that for images capturing distant objects, the database image used should not be too close to the query one to achieve reasonable pose estimate precision.

Keywords: 3D computer vision, visual localization, perspective camera, pose estimation, generalized camera, semi-generalized camera, minimal solvers

Supervisor: RNDr. Zuzana Kúkelová, Ph. D.
Visual Recognition Group, FEE CTU

Abstrakt

Cílem této práce bylo prozkoumat chování řešiče E_{5+1} pro problém semi-generalizovaného odhadu pozice kamery. Byly navrženy a provedeny experimenty zkoumající, jak je přesnost odhadu pozice ovlivněna vzdáleností odhadované kamery od databázových kamer a vzdálenosti použitých databázových kamer. Výsledky ukazují, že přesnost odhadu pozice kamery převážně nezávisí na vzdálenosti odhadované kamery od databázových, ale přesnost odhadu klesá, pokud je použit malý počet databázových kamer. Výsledky experimentů naznačují, že při odhadu pozice kamery snímající vzdálené objekty vede použití databázových kamer příliš blízko u sebe k nepřesnému odhadu.

Klíčová slova: trojrozměrné počítačové vidění, vizuální lokalizace, perspektivní kamera, odhad pozice, generalizovaná kamera, semi-generalizovaná kamera, minimální řešiče

Překlad názvu: Zkoumání vizuální lokalizace založené na semi-generalizovaném odhadu pozice kamery

Contents

Figures

Tables

Chapter 1

Introduction

In many applications, such as robot or self-driving car localization, cameras are still one of the preferred sensors from which the position and orientation of the camera (and consequently of the object to which the camera is attached) can be estimated. The task of estimation of camera *pose* (position and orientation) based on the image taken by the camera is known as visual localization. Visual localization is one of the classical problems of computer vision, due to its wide range of applications. Camera pose estimation task is present in applications such as augmented and virtual reality or indoor and outdoor navigation. Navigation itself includes various applications, from people navigation using mobile phone cameras to autonomous robot or self-driving car navigation, sometimes combined with the usage of depth cameras or LIDARs.

In navigation applications, estimation of absolute camera pose – that means position (translation) with scale and orientation (rotation) – is needed. The terms position and orientation mean pose in the (previously established) world coordinate systems, while the terms translation and rotation are more likely used when emphasizing relation to some other camera. The absolute pose estimation problem can be solved using the geometric properties of cameras and point-to-image projection and methods of algebraic geometry.

State-of-the-art methods of visual localization are based on the 2D-3D correspondences – they use correspondences between pixels (2D points) in the image from query camera[1] and 3D points from existing model. This approach requires the existence and maintenance of a scene 3D model along with correspondences between 3D points and 2D points in the query camera image. In this case Perspective-n-Point algorithms (PnP) [1] are widely used. The most

[1] cameras for which the pose is being estimated will be throughout the text referred to as *query camera* and cameras with known pose and intrinsics used for estimation will be referred to as *database cameras*

common solver used for camera localization is the well-known calibrated P3P solver [2].

Although PnP algorithms generally have very high precision, the usage of 3D models has several disadvantages. Creation and maintenance of 3D models can be both time- and space-consuming. For triangulation of 3D points, such points must be visible in at least two database cameras. Therefore, points visible in the query camera but only in one database camera cannot be used for pose estimation.

As an alternative, the scene can be represented with a set of images with known extrinsic (pose) and intrinsic parameters (focal length, principal point coordinates, pixel aspect ratio, skew angle etc.). Then, for the pose estimation, 2D-2D correspondences are used between the query image and database images. If there are used 2D-2D correspondences between query image and a single database image, only relative pose of the query camera can be estimated (rotation and translation without scale). For translation scale estimation (so-called absolute camera pose estimation problem), correspondences with multiple database cameras need to be used. Algorithmically, these multiple database images are represented as a generalized camera, i.e. a camera with multiple centres of projection. Consequently, the problem of estimating the pose of a single perspective camera w.r.t. a generalized camera is also known as semi-generalized camera pose estimation problem. Algorithms that use this approach are proposed by Zheng and Wu in [3] and by Bhayani et al. in [4]. A combination of 2D-2D and 2D-3D correspondences is used for semi-generalized pose estimation by Bhayani et al. in [5].

However, the semi-generalized pose estimation methods are not applicable as generally as the absolute pose methods that use only 2D-3D correspondences. The main reasons are that semi-generalized solvers either consider only special cases of a scene (planar or close to planar in case of [4]), they are much slower than methods using 2D-3D correspondences (case of E_{4+2} and Ef_{5+2} solvers from [3]), or the accuracy of the resulting pose is worse in comparison to methods using 2D-3D correspondences (as in case of E_{5+1} and Ef_{6+1} solvers from [3] compared to the P3P solver).

The aim of this work is to investigate further how the accuracy of semi-generalized pose estimation algorithms, particularly the E_{5+1} algorithm from [3], depends on camera configuration and whether the accuracy can be improved by establishing constraints on choosing the used database cameras. These constraints can be the distance between used database cameras or the distance between the query camera and database cameras.

1.1 Thesis Structure

First, Chapter 2 describes the problems of visual localization and camera pose estimation, along with a description of a localization pipeline and details on generalized and semi-generalized cameras, and the semi-generalized camera pose estimation problem formulation. In addition, the chapter raises a set of research questions regarding the dependency of semi-generalized camera pose estimation accuracy on the distance relations between query and database images. The Chapter 3 contains detailed information about the experiments performed and aims to answer the questions asked in the previous chapter. In Section 3.1, implementation details and the used software libraries are described. The Section 3.2 provides information about the used dataset and error measures. These are then followed by sections describing the experiments themselves – used setup and results. The last chapter 4 aims to conclude the obtained results and suggest future work.

There is no separate section on the current State-of-the-Art. Rather, these works (along with related work) are sufficiently described in Chapter 2, particularly in Sections 2.1, 2.2, and 2.3.

1.2 Thesis Contribution

The thesis contributes understanding of the performance of the E_{5+1} solver used for semi-generalized camera pose estimation through an experimental analysis. The goal is to study the solver behaviour when the query and database cameras are in different configurations. Particularly, the effect of the distance between query camera and database cameras and the distances between database cameras is examined. The discovered dependencies could help to improve the E_{5+1} solver accuracy by choosing appropriate database cameras for pose estimation during the camera pose estimation stage of visual localization.

Chapter 2

Visual Localization Based on Semi-Generalized Camera Pose Estimation

2.1 Visual Localization

As outlined in Chapter 1, visual localization approaches try to estimate the pose of a query camera with respect to a given scene. Visual localization algorithms are part of interesting real-world applications, such as self-driving cars and other autonomous robots [6], and augmented and virtual reality applications [7, 8].

Localization algorithms can be divided into groups based on the representation of the scene they use: Traditionally, explicit representations, i.e. storing a set of 3D points [9, 10] or a set of database images with known poses [3, 11], are used. With rising popularity of deep learning methods, more implicit representations are proposed [12, 13, 14, 15]. These approaches represent the scene through the weights of machine learning models (mostly neural networks). They are trained to directly regress the pose (either absolutely with respect to the scene as in [13] or relatively, w.r.t. a given set of database images as in [16]) or to predict a set of 2D-3D matches for PnP-based pose estimation (scene coordinate regression, in [17, 14, 15, 12]). Pose regressors are currently significantly less accurate than methods based on the explicit scene representations [18]. It is currently unclear whether scene coordinate regressors or explicit representations are better [19]. This thesis focuses on explicit representations. More specifically, it aims to investigate the performance of methods that are based on storing a set of database images with their poses (and intrinsic calibrations) and that estimate the query pose w.r.t these database images.

The following text first reviews a localization pipeline commonly used in the literature that stores a 3D point cloud and uses PnP algorithms for

pose estimation. Section 2.3 reviews work on (semi-)generalized camera pose estimation, derives the E_{5+1} solver used in this thesis starting from a more general definition of semi-generalized pose estimation, and explains how the localization pipeline from the previous section can be adapted to use semi-generalized pose estimation. Finally, Section 2.4 raises research questions which this thesis aims to address.

2.2 A Common Localization Pipeline

Probably the most popular traditional approach to visual localization is to represent the scene through a (sparse) 3D point cloud [9, 10]. This 3D model is computed in an offline pre-processing stage: Given the database images, features [20] are extracted from each database image and matched between the database images. This results in a set of 2D-2D correspondences. If the intrinsic calibration and camera poses of the database images are known, the 3D points corresponding to these 2D-2D matches can be obtained by triangulation. If these parameters are not known, they can be estimated using Structure-from-Motion [21]. In both cases, the result is a 3D point cloud, where each 3D point was triangulated from features found in multiple images. As such, each 3D point can be associated with the corresponding feature information, in particular a feature descriptor [20].

During online operation, the pipeline tries to estimate the absolute pose of a given query image w.r.t the 3D point cloud. To achieve this, it established matches between features extracted from the query image and 3D points in the scene, which is possible by comparing the descriptors of the 3D query features with the descriptors associated with the 3D points. This results in a set of 2D-3D correspondences that can be used for pose estimation.

Pose estimation is done using a PnP solver. The solver itself only needs a small number of correspondences. However, not all correspondences will be correct, i.e. one cannot just take a small subset of the matches for pose estimation and ignore all others. Thus, the solver itself is run in a RANSAC (RAndom SAmple Consensus [1]) loop. In every iteration of the RANSAC loop, the appropriate number of random correspondences from random images (RANdom SAmple) is selected and used for pose estimation by the solver. The resulting pose is then evaluated on all correspondences. For each correspondence, it measures the reprojection error between the 3D point and the corresponding 3D feature. If the error is lower than some chosen threshold, the correspondence is counted as inlier – 'correct correspondence' – that is the Consensus part. The loop has either fixed number of iterations or it is stopped on some criterion, e.g. when the probability of missing a better model falls below a chosen threshold. The pose with the most inliers is then presented as the algorithm result (it is the beast pose from the tested ones,

but it could happen not to be precisely the global optimum and it usually is not).

There exist some modifications of the standard RANSAC algorithm: MLESAC (Maximum Likelihood Estimation SAmple Consensus, [22]) does not count the number of inliers, but the impact of every inlier is weighted based on its distance from the reprojection error threshold (outliers are given a constant weight). This amount to optimizing a robust cost function, which has been shown to lead to better results. Another modification is LO-RANSAC (Locally Optimized RANSAC, [23]) algorithm, which uses local optimization of each newly found best model to reduce the impact of noise on the pose estimated, resulting in more accurate poses. Both modifications are recommended in practice [24].

Comparing the descriptor of a query feature against the descriptors of the 3D points can be time-consuming, especially in large scenes. a common approach is thus to use an intermediate image retrieval step [25, 26] that identifies a small subset of database images that are visually similar to the query image. The features in the query image are then matched against the features extracted from the retrieved database images, resulting in 2D-2D matches. From the pre-processing stage, it is known which database features correspond to 3D points. With this information, the 2D-2D matches are lifted to 2D-3D matches for pose estimation.

2.3 Semi-Generalized Camera Pose Estimation

The following first reviews state-of-the-art algorithms for (semi-)generalized camera pose estimation, then derives the semi-generalized pose estimation problem and the E_{5+1} solver, and finally explains how the localization pipeline from the previous section can be adapted to use a solver for semi-generalized pose estimation.

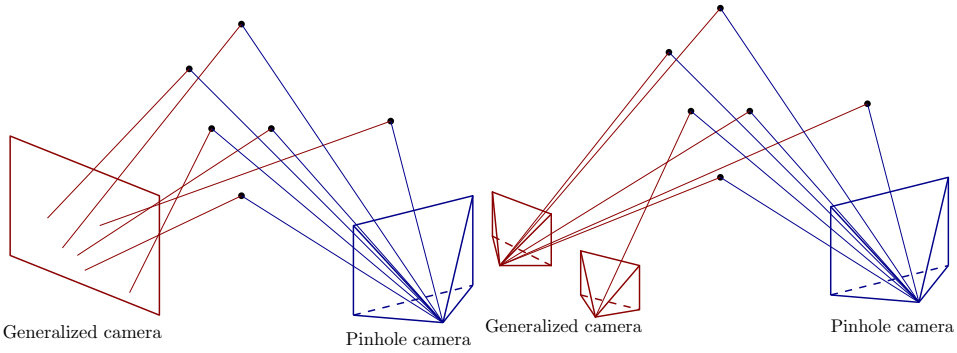
2.3.1 State-of-the-Art

When estimating the relative pose (orientation and translation without scale) of one image with respect to another, only the translation direction, but not the magnitude of the translation, can be estimated. In the context of visual localization, the magnitude is required. Thus, the pose of a query image needs to be estimated relative to multiple database images. To simplify the modelling of such a situation, the concept of generalized camera ([27], [28]) was established. A generalized camera is a camera with multiple projection

centres (or even more generally a set of common rays without the same principal point).

There exist solvers for relative pose estimation of two generalized cameras (from 6 correspondences estimating the orientation and translation with scale), e.g., the 6pt solver for relative generalized pose problem [29]. But this is generally a very hard problem and state-of-the-art algorithms are too slow for practical usage in real applications. For the problem of a single perspective camera pose estimation w.r.t a generalized camera, they are not even needed, because the substitution of one generalized camera with a single perspective camera, the model is simplified and so are the equations that need to be solved. The problem of absolute pose estimation of a single perspective camera w.r.t. generalized camera is called semi-generalized pose estimation.

In [3], several solvers for the semi-generalized pose estimation problem are proposed. The problem can be divided into two cases based on the distribution of correspondences between the cameras in the generalized camera (see figure 2.1).



(a) : Semi-generalized camera model.

(b) : Correspondences distribution used in E_{5+1} solver.

Figure 2.1: General model of semi-generalized camera and configuration used in E_{5+1} solver. Inspired by [3].

The absolute camera pose problem has 6 degrees of freedom (DOFs), and 7 in the case of an uncalibrated camera (where the focal length is also estimated), therefore 6 (resp. 7) correspondences need to be used. First one is a case, where there are five correspondences with one camera and one remaining correspondence with another camera (or Ef_{6+1} in the case of an uncalibrated camera). In the case of the E_{5+1} solver, one can use the standard 5pt essential matrix estimation algorithm [30] on the image with five correspondences. The essential matrix is then decomposed into relative rotation and translation and from the additional image correspondence, the scale of translation is computed. The Ef_{6+1} solver analogously uses the 6pt algorithm (the original problem was described by Stewenius et al. in [31], [3] uses a more robust formulation and implementation from Bujnak et al. [32]).

The other case is where 4 or less correspondences are with one camera. For this case, new minimal solvers E_{4+2} and Ef_{5+2} were developed in [3].

In visual localization pipelines, cameras are usually assumed to be calibrated, therefore the solvers for uncalibrated cases will not be examined in this thesis. The E_{5+1} solver is much faster than the E_{4+2} solver, it only runs a few microseconds (because it is based on the well-known efficient 5p relative pose solver by Nister [30]) and therefore can be used for real applications. Thus, only the E_{5+1} solver and not the E_{4+2} solver will be further examined in this thesis.

2.3.2 Semi-Generalize Camera Pose Estimation

In semi-generalized camera pose estimation, the goal is to estimate the position and the orientation of a perspective camera, denoted as $\mathcal{P}$, w.r.t. a generalized camera, denoted as $\mathcal{G}$. In this thesis, we study the problem where the generalized camera $\mathcal{G}$ consists of a set of k fully calibrated perspective cameras $\{\mathcal{G}_1, \dots \mathcal{G}_k\}$ with known poses and the internal calibration of $\mathcal{P}$ is known as well.

In general, 2D points detected in the generalized camera $\mathcal{G}$ can come from different perspective cameras $\mathcal{G}_j$. Therefore, the standard way how to represent a 2D image measurement in the generalized camera $\mathcal{G}$ is using a 3D line L defined in the coordinate system of the generalized camera. Such a 3D line can be represented using Plücker coordinates as $\mathbf{L} = [\mathbf{q}^T, \mathbf{q}'^T]^T \in \mathbb{R}^6$. In this case, the vector $\mathbf{q} \in \mathbb{R}^3$ is the unit direction of the line $\mathbf{L}$, and $\mathbf{q}' \in \mathbb{R}^3$ is a vector such that $\mathbf{q}' = \mathbf{q} \times \mathbf{p}$ for any point $\mathbf{p}$ on the line $\mathbf{L}$. With this representation, any point $\mathbf{p}(\lambda)$ on the line $\mathbf{L}$ can be represented as

$$\mathbf{p}(\lambda) = \mathbf{q}' \times \mathbf{q} + \lambda \mathbf{q} \tag{2.1}$$

for $\lambda \in \mathbb{R}$ and two lines $\mathbf{L_1} = [\mathbf{q}_1^T, \mathbf{q}_1'^T]^T$ and $\mathbf{L_2} = [\mathbf{q}_2^T, \mathbf{q}_2'^T]^T$ intersect in space if and only if

$$\mathbf{q}_1 \mathbf{q}_2' + \mathbf{q}_1' \mathbf{q}_2 = 0. \tag{2.2}$$

Without loss of generality, let us assume that the global coordinate system corresponds to the coordinate system of the generalized camera $\mathcal{G}$. The goal is to estimate the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ that transform the coordinate system of the query camera $\mathcal{P}$ to the global coordinate system. For this task, we can use 2D image point correspondences detected in the perspective query camera $\mathcal{P}$ and the generalized camera $\mathcal{G}$, respectively, in one of its perspective cameras $\mathcal{G}_j$. These 2D image point correspondences are represented by 3D rays from $\mathcal{P}$ and $\mathcal{G}_j$.

For the i^{th} 2D point correspondence, let the Plücker coordinates of the 3D ray from $\mathcal{P}$, in the coordinate system of $\mathcal{P}$, be denoted as $\mathbf{L}_i = [\mathbf{q}_i^T, \mathbf{q}_i'^T]^T$,

and the Plücker coordinates of the 3D ray from $\mathcal{G}_j$, in the coordinate system of $\mathcal{G}$, be denoted as $\mathbf{L}_{ij} = [\mathbf{q}_{ij}^T, \mathbf{q}_{ij}'^T]^T$. After transforming these two 3D lines in the same coordinate system, i.e., the coordinate system of $\mathcal{G}$, these lines should intersect. Using equation (2.2), this results in the semi-generalized epipolar constraint

$$\mathbf{q}_i^T \mathbf{R} \mathbf{q}_{ij}' + \mathbf{q}_i'^T \mathbf{R} \mathbf{q}_{ij} - \mathbf{q}_i^T [\mathbf{t}]_\times \mathbf{R} \mathbf{q}_{ij} = 0, \tag{2.3}$$

where $[\mathbf{t}]_\times$ represents the skew-symmetric cross-product matrix.

The constraint (2.3) can be further simplified by assuming that the camera centre of $\mathcal{P}$ is the origin of its local coordinate system. In this case $\mathbf{q}_i' = 0$ and the equation (2.3) has the form

$$\mathbf{q}_i^T \mathbf{R} \mathbf{q}_{ij}' - \mathbf{q}_i^T [\mathbf{t}]_\times \mathbf{R} \mathbf{q}_{ij} = 0. \tag{2.4}$$

If less than 5 points are detected in the same camera $\mathbf{G}_j$ of the generalized camera $\mathbf{G}$, the semi-generalized relative pose estimation problem results in a quite complex system of equations. Such a system can be solved from the minimum number of six point correspondences using algebraic methods, e.g. Gröbner bases. However, the resulting solvers are large and slow for practical applications [31, 3].

On the other hand, if five point correspondences are detected by the same camera, e.g., $\mathcal{G}_1$ of the generalized camera $\mathcal{G}$, the situation is significantly simpler. In this case, without loss of generality, we can assume that the camera centre of $\mathcal{G}_1$ is the origin of the coordinate system of $\mathcal{G}$. This means that $\mathbf{q}_{i1}' = 0$ and the semi-generalized constraint (2.4) for the points detected in $\mathcal{G}_1$ has the form

$$\mathbf{q}_i^T [\mathbf{t}]_\times \mathbf{R} \mathbf{q}_{ij} = 0. \tag{2.5}$$

This is a well-known standard epipolar constraint with the unknown essential matrix $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$.

Five point correspondences detected between the query camera $\mathcal{P}$ and the database camera $\mathcal{G}_1$ give us five equations of the form (2.5). Note that here these equations are homogeneous and they can be used to estimate the unknown translation only up to scale. The five equations of the form (2.5) can be highly efficiently solved using the well-known 5pt relative pose solver [30]. Then the scale of the translation can be computed using sixth point correspondence coming from some other camera $\mathcal{G}_j, j \neq 1$.

2.3.3 Structure-less Visual Localization

The visualization pipeline described above can be adapted for E_{5+1} solver. When using the 2D-2D correspondences, the offline pre-processing stage,

where 3D model is created is not needed. In first step of the pipeline, 2D-2D feature matches between query and database cameras are established and they are directly used as 2D-2D correspondences for pose estimation. Image retrieval can be used here as well. Pose estimation in this case is done using E_{5+1} solver and it is run in the RANSAC loop, in the same manner as the PnP solver in the structure-based pipeline above.

2.4 Motivation of Experimental Analysis

Experiments performed on multiple datasets have shown that the localization pipeline (outline above) based on the semi-generalized pose estimation leads to less accurate pose estimates when compared to the standard pipeline based on PnP solver. These observations, made outside of the scope of this thesis, have served as the motivation for this thesis. Particularly, the aim of this thesis is to understand the factors that affect the pose accuracy of localization pipelines based on semi-generalized pose estimation.

To this end, the following is investigated in the thesis:

- How is the pose accuracy affected by the distance between the pose of the query image and the poses of the database images? Does the pose accuracy drop if the query image was taken too close or too far away from the database images? Understanding this relation allows practitioners to judge whether localization systems based on semi-generalized pose estimation are suitable for their problem setting.
- In some settings, e.g. when the scene is represented by a dense 3D model [33], it is possible to render synthetic views of the scene from novel viewpoints. This can be exploited in the context of localization by generating additional views around an initial pose estimate of the query image. To this end, the thesis aims to understand how different ways how to generate these views (that is, how to select their orientation and distance to the query pose) impact pose accuracy.
- How does the distance between the database images impact the pose accuracy? Is it helpful to select database images that have a certain minimal distance? Or are clusters of database images (with similar poses) not an issue in practice?

Chapter 3 addresses these questions through experiments on both real and synthesized data.

Chapter 3

Experiments

This chapter is structured as follows: Section 3.1 provides implementation details, including which libraries were used. Section 3.2 describes the datasets used for experimental evaluation. Section 3.2.1 details the error measures used for evaluation. All subsequent parts aim to address the research questions posed in Section 2.4.

3.1 Implementation

All experiment code was written in Python version 3.9.2 with libraries PoseLib [34] and pycolmap [35] (which is library providing Python bindings for C++ library COLMAP [36, 21]). For rotation manipulation is used `Rotation` object from `scipy.spatial.transform` library [37].

All codes, along with measured data are available on `https://gitlab.fel.cvut.cz/smutnale/experiments-on-e5-1-solver`.

3.1.1 PoseLib

The solver implementation used in this thesis is from the *PoseLib* [34], library of minimal solvers for camera pose estimation from V. Larsson. This library contains state-of-the-art solvers and their implementation is fast and robust. Besides the solvers themselves, this library provides the implementations for many parts of a full visual localization pipeline, including data normalization, LO-RANSAC loop and post-RANSAC non-linear refinement of the estimated pose on the inlier set. The library is written in C++, but it does have also Python bindings, which were used in this thesis as all code is in Python. In

this thesis, the tested solver is the semi-generalized relative pose estimation solver that uses $5 + 1$ points (correspondences). The method from PosLib, that is used, is `estimate_generalized_relative_pose()`, which applies the E_{5+1} solver inside a LO-RANSAC loop, followed by non-linear refinement of the estimated pose on the predicted inlier set.

3.1.2 COLMAP

The dataset used in this thesis is in format compatible with COLMAP library [21, 36]. This library provides general-purpose Structure-from-Motion and Multi-View Stereo pipelines with graphical and command-line interfaces. In this thesis the reconstruction tools provided in this library were not used, only the method for data (cameras, images and 3D points) manipulation. Using the dataset format readable by COLMAP library, the library interface for work with cameras, images and 3D points can be used. This allows to access the camera intrinsic calibration, image and 3D points poses and also methods for 3D points projection and coordinate systems transformation. The library is written mostly in C++, but in this thesis was used Python library pycolmap [35], which exposes part of the COLMAP library to Python.

Throughout these thesis, the distance between cameras (images) means the distance between their projection centres. Camera projection centre position can be computed using COLMAP method, but the camera object does not store this camera parameter directly, only camera rotation and translation is stored. Therefore when the camera projection centre position needs to be changed, from the changed values of projection centre and rotation matrix the camera translation vector needs to be computed and saved.

3.2 Cambridge Landmarks dataset

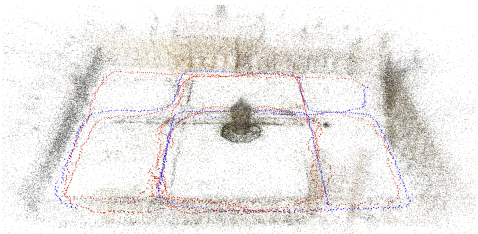
The Cambridge Landmarks dataset [38] is widely used to benchmark visual localization techniques, e.g. in [39] or [40]. This dataset consists of six outdoor scenes (Great Court, King's College, Old Hospital, Shop Facade, Street and St. Mary's church), with different number of captured images and partial overlap between the scenes (e.g., the Street scene contains areas covered in the King's College and St. Mary's Church scenes). Visualizations of scenes are in figure 3.1. The scenes differ extensively, e.g. in size – compare Shop Facade and Great Court – or shape – Old Hospital scene is close-to-planar, Shop Facade scene consists of two planar parts perpendicular to each other, Kings College is a very indented building and St. Mary's Church or Great Court scenes are depicting very not planar landmarks. The scenes, as they are outdoor, also contain the 3D points reconstructed from moving

pedestrians and vehicles, which add to the point cloud points with depth (distance from camera) different from the the 3D points that are on the buildings or landmarks.

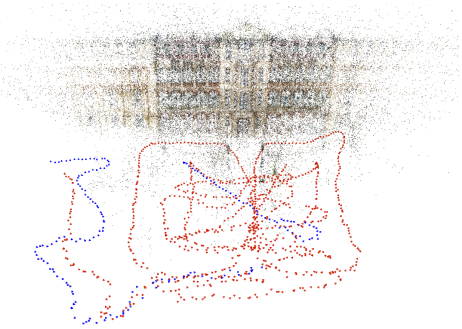
Every scene data consists of a set of video recordings of the landmark, captured by pedestrian walking around the landmark in different illumination and weather conditions. From this high definition video, images were produced by subsampling the video. One image was captured every 2 s, which is equivalent to approximately 1 m distance between two subsequent camera positions. From these images was generated 3D point cloud (in `.nvm` format) using the VisualSFM Structure-from-Motion method from Wu [41]. Each scene has in addition also text files with selection of train and test images, along with ground truth camera poses.

In this thesis, the dataset information are transformed from VisualSFM's file format into the format in which the COLMAP library stores its representation of the dataset. To this end, COLMAP was used to extract and match SIFT [20] features. The known poses and intrinsics of the images, together with the matches, are then used to triangulate the 3D point cloud. The results are three binary files and one text file with list of query cameras, which is the same as the test set from original dataset, but it does not include the ground truth positions (since they are present in the binary files). In the first binary file, there are 3D points positions with information about the images containing particular point. In the second one, there are information about cameras – their intrinsic parameters, their model, etc. In third one, there are information about images – by which camera it was taken, the camera pose from which the image was taken and how many and which points visible in the image are triangulated (and therefore present in the 3D point cloud). Although the COLMAP terminology distinguishes between cameras and images, where cameras include information only about camera model and intrinsic and images include the information about name of the camera that captured the image (through this it can be connected to information from camera object), camera pose and captured 3D points, further in this thesis there will be used both the terms camera and image with same meaning, joining the information about camera intrinsics, pose and 3D points in particular image.

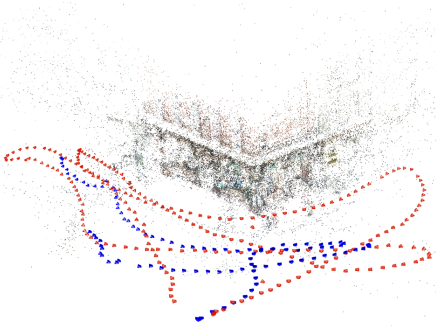
The scenes have different number of triangulated 3D points, database cameras and query cameras, particular values are shown in table 3.1. Notice that in this thesis, the Street scene is not used. The underlying 3D geometry is not fully correctly estimated, leading to the same physical structure being duplicated multiple times in the model. It is this common to not use the Street scene for evaluation.



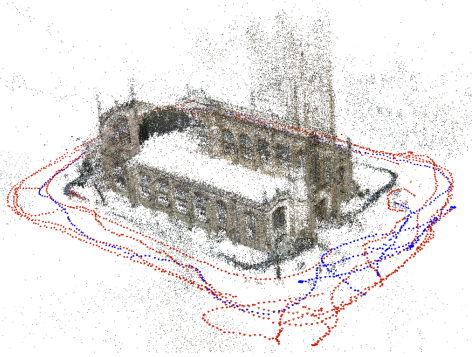
(a) : Great Court.



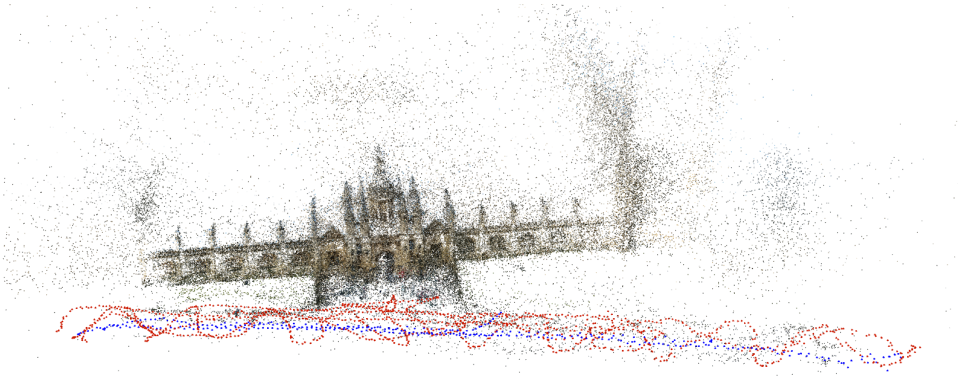
(b) : Old Hospital.



(c) : Shop Facade.



(d) : St. Mary's Church.



(e) : King's College.

Figure 3.1: Cambridge landmarks dataset scenes. Query cameras are depicted in blue and database cameras are red.

scene	# query cameras	total # cameras	# 3D points
Great Court	760	2292	211 902
King's College	343	1563	161 339
Old Hospital	182	1077	136 157
Shop Facade	103	334	43 933
St. Mary's Church	530	2017	274 012

Table 3.1: Parameters of Cambridge Landmarks dataset. # sign means 'number of'.

3.2.1 Error meassures

The examined task in this thesis is camera pose estimation. A pose consists of two distinct parts, translation and orientation, for which the errors need to be computed separately. The position error ϵ_{pos} was computed as the Euclidean norm of difference between computed and ground truth position of query camera projection centres

$$\epsilon_{\text{pos}} = \|C_{\text{gt}} - C\|_2 \ , \tag{3.1}$$

where C_{gt} is the ground truth projection centre position vector and C is estimated projection centre position vector (both in the world coordinate system of 3D model) and it is reported in meters. Orientation error ϵ_{or} was computed as

$$\epsilon_{\text{or}} = \arccos\left(\text{trace}\left(R_{\text{gt}}^T R\right) - 1\right) \ , \tag{3.2}$$

where R_{gt} is ground truth rotation matrix and R is estimated rotation matrix. The error is reported in degrees. As is common [13], median position and orientation errors over the query images are reported per scene.

3.3 Experiments with distance between query and database cameras

The first experiment aims to answer question 1 raised in Section 2.4, i.e., whether the distance between the query and database images affects pose accuracy. Notice that for further experiments, relevant database images per query are determined by shared 3D points in the 3D models, not by using actual image retrieval as to avoid introducing a potential error source. Similarly, matches are obtained via the 3D points, not by running a separate matching algorithm, for the same reason.

3.3.1 Experiment setup

The experimental process on one scene and one query camera-database cameras distance was following:

1. From all 3D points in the 3D model, 3D points that are visible by query camera and database cameras are picked and saved in a structure preserving the correspondence information.
2. For every query camera:
    1. The set of all database cameras is restricted to set of the ones in given distance from query camera and with at least 10 correspondences with query camera. This means that both cameras used in E_{5+1} solver are within a given distance.
    2. The list of database cameras is sorted in descending order according to the number of correspondences and only first 1000 is used further.
    3. All visible 3D points are projected into the query camera and corresponding database cameras using funcionality provided by COLMAP.
    4. To every projected point is added random noise, generated from uniform distribution on a circle with radius 1 px.
    5. These points are saved as correspondences.
    6. Correspondences, database cameras pose and intrinsics, and query camera intrinsics are given as input to the PoseLib semi-generalized pose estimation method.
    7. The pose obtained from the estimation method is compared with the ground truth and position and orientation errors are computed (as described in Section 3.2.1) and saved.

The described process is run for a set of multiple minimum and maximum distances between query and database cameras. The minimum distance changes from 0 m to 15 m with 1 m step and the maximum distance changes from 1 m to 15 m also with 1 m step. In the last run, maximum distance is set to 10 000 m, which means that all database cameras farther than 15 m from query camera are considered (since there are used real datasets, bound 10 000 m is high enough to not cut any camera out). The distance between database cameras is not restricted in this experiment.

If for any query camera there are not enough cameras in a given range of distances to run E_{5+1} solver (at least 2 database cameras are needed), that query camera is not considered in the computation of error.

3.3.2 Results

Median position errors for all scenes are shown in figure 3.2. Median errors in

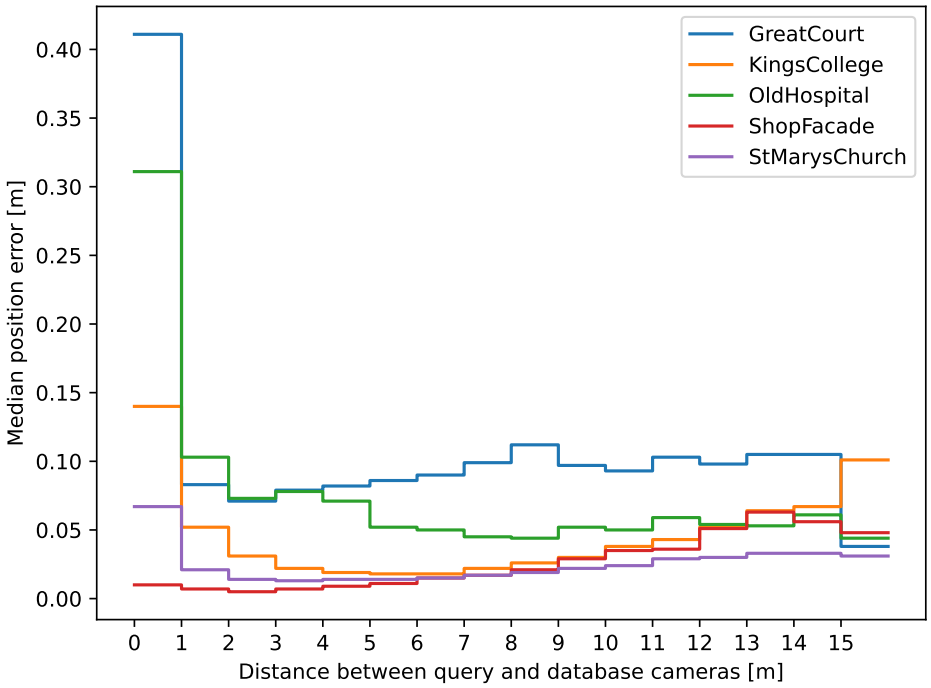


Figure 3.2: Dependency of the median position error on the distance between query and database cameras.

the Shop Facade scene show different trend than the errors in the other scenes. The median position error for the closest distance range is significantly higher than for the rest of distances for all scenes except the Shop Facade. This can be caused by insufficient number of successfully (with at least two database cameras with enough correspondences shared) estimated camera poses (as shown in figure 3.3). In the case of the closest range, the Shop Facade scene has highest ratio of estimated camera poses from all scenes, while with increasing distance (from distance of 11 m up) the ratio falls rapidly. Also the number of database cameras with enough correspondences shared with query camera is low for close ranges for all scenes and also for far ranges for the Shop Facade scene (see figure 3.4).

In terms of absolute values, the highest median position errors were measured on the Great Court scene. From figure 3.4, it is visible that the Great Court scene has low average number of usable database cameras in all distance ranges. This is probably due to large spatial extent of this scene – both cameras and 3D point are distributed on large area, therefore the distance scale used in this experiment is a bit inappropriate. If applying a more appropriate scaling of the distance ranges used, it is possible that the errors

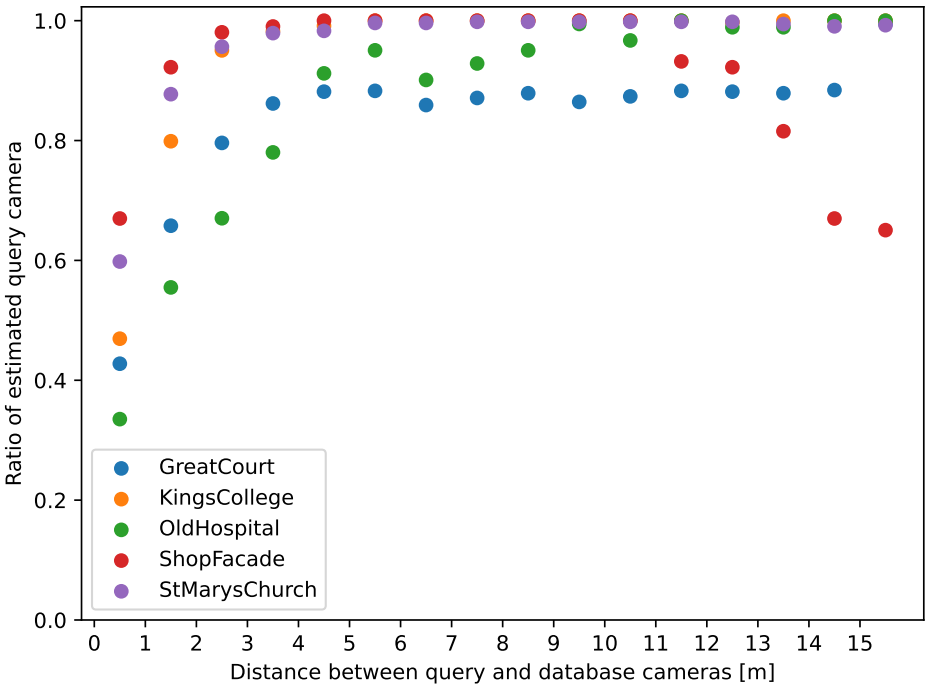


Figure 3.3: Ratio of query cameras for which a pose could be estimated.

would be comparable to other scenes. But the inappropriate scaling allows to see one possible influence on the pose estimation errors, which is the low number of usable database cameras. This dependency is confirmed also by median position errors on other scenes in particular distance ranges. It is particularly visible on distant ranges in the Kings' College scene – decrease in number of database cameras leads to an increase in position error – and the Old Hospital scene – increase in number of database cameras leads to decrease in position error.

Median position errors are reasonably low for St. Mary's Church, King's College and Shop Facade scenes. But a low median error does not need to mean that all measured errors are low. The graphs of position errors measured on these scenes are in figure 3.5. The difference between median and mean values is smallest for the St. Mary's Church scene, for other two scenes, it has some exceeding values. The general trend of the third quartile is similar to the median one, showing that the estimation algorithm behaves best in the medium distance ranges, not closer than 1 m or 2 m and not farther than 12 m or 13 m.

Median orientation errors are shown in figure 3.6. The orientation errors are comparable for most scenes, except the Shop Facade and Old Hospital scenes.

The orientation error for the closest range is higher, which corresponds

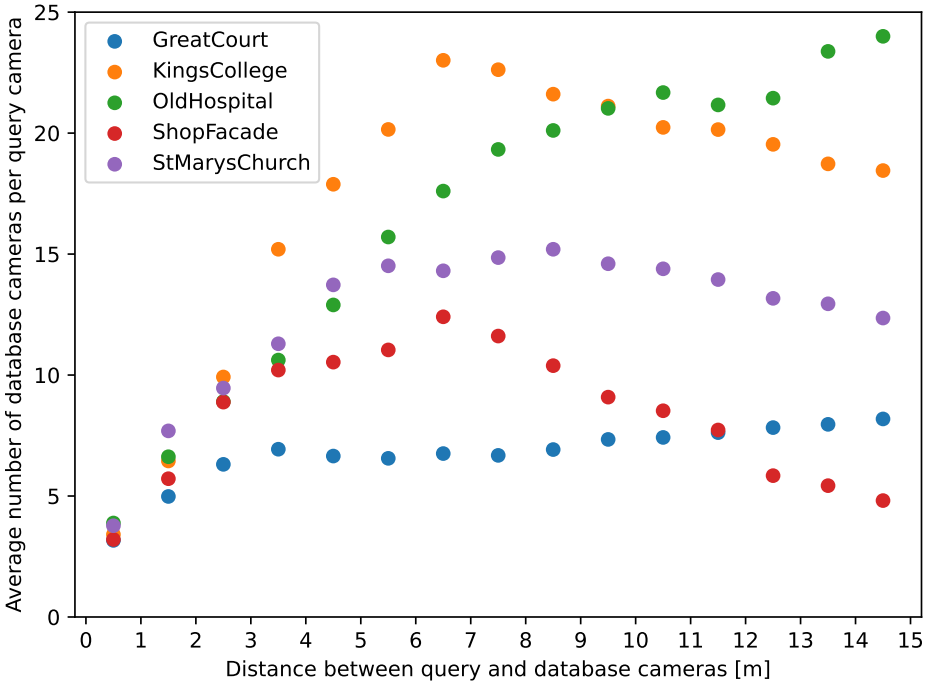


Figure 3.4: Average number of database cameras with shared correspondences with query images as a function of the distance between the query and the database images.

with the measurements of median position errors. But the differences for the rest of the distance ranges are (if Shop Facade and Old Hospital scenes are omitted) small. Even the error for the Great Court scene is comparable with (or even lower than) errors on other scenes. This could be caused by the nature of the scene. The farther distance of 3D points from cameras can cause higher position errors, while it can be beneficial when estimating the orientation. This corresponds to the human experience – when you are looking at some distant object and walk some small distance, the position of distant object on the 'projection plane' of the human eye changes only a little (if at all), while when you rotate a little, the position of distant object in the 'projection plane' changes a lot.

In the case of the Shop Facade scene, the median orientation errors are low for close ranges and increasing for far ranges, which corresponds to the trend of the median position error and it is probably caused by the same reason, i.e., the small number of usable database cameras and also the smaller number of estimated poses. Therefore, the statistics can be non-descriptive regarding the general behaviour of the algorithm. In the case of the Old Hospital scene, which has also slightly larger median position errors than the rest (except for Great Court scene), the median orientation error is multiple times higher than the median orientation error on the King's College and Great Court scenes. The Old Hospital scene does have a sufficient number

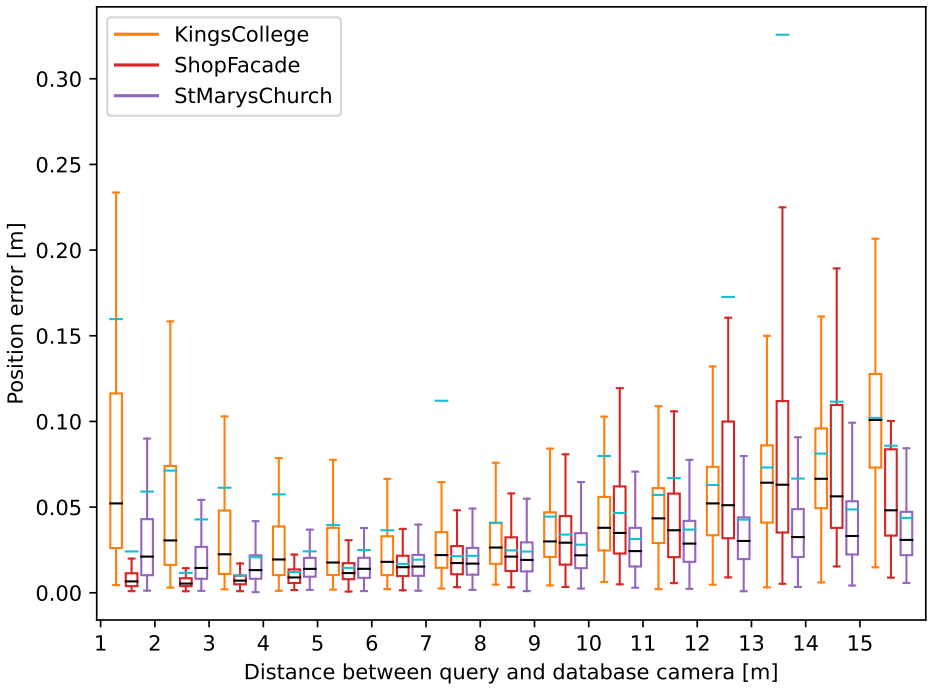


Figure 3.5: Position errors, black lines are medians, cyan lines are mean values. Errors for closest distance range were omitted since they worsen the readability and they are irrelevant for examination of error distribution of ranges with low median error. Outliers are not present.

of usable database cameras, therefore the reason of the higher error needs to be elsewhere. Another possible explanation could be that in the Old Hospital scene, cameras are close to the nearest 3D points (the cameras do look perpendicular to the facade direction, therefore the nearest 3D points are seen by the cameras) and the same explanation as in the case of Great Court can be applicable (reversely). But the median position error is in comparison with other scenes higher as well. From the collected data, it is not obvious, what is the cause of higher errors measured on Old Hospital scene.

3.3.3 Conclusion

Both position and orientation median error is high for the closest range (up to 1 m). This could be caused by an insufficient number of database cameras sharing correspondences with the query camera. But this could not be the only cause. Based on the performed experiments, no conclusion regarding the suitability of database cameras in different distances from query camera can be made. It needs to be further investigated the algorithmic behaviour for database cameras close to the query camera and for an equal number of cameras in each distance range.

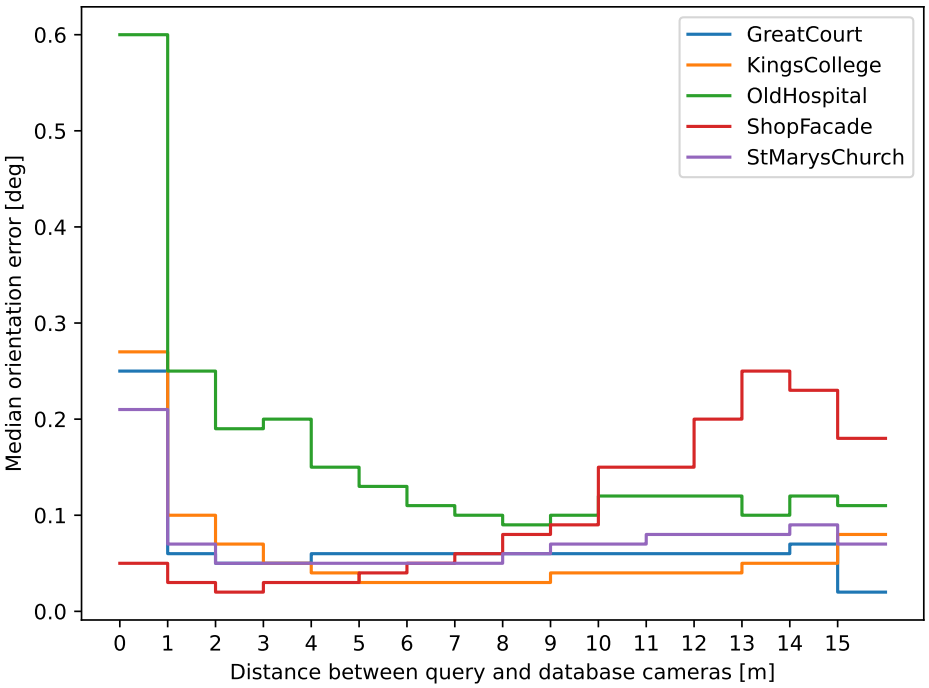


Figure 3.6: Dependency of median orientation error on the distance between query and database cameras.

Because one of the main cause of higher errors seems to be the lack of database cameras sharing the correspondences with the query camera. Therefore, I would suggest not to restrict the used cameras in any way which could result in using only a small number of database cameras as input to the RANSAC loop.

The median position errors for scenes and distance ranges, where the number of database cameras and correspondences was sufficient (St. Mary's Church, King's College and Shop Facade scenes with database cameras between 1 m and 13 m far from the query camera) are smaller than 5 cm, the third quartile of position errors on mentioned scenes and distances (see figure 3.5) is smaller than 10 cm.

The median orientation errors are for all scenes and all distance ranges except the closest one (up to 1 m) smaller than 0.3°, for scenes with lower errors, it is even smaller than 0.1°. This is an error, which ensures sufficient precision for most of the applications.

3.4 Experiments for close distances between query and database cameras

For closer distance ranges (up to 2 m) the errors for all scenes except the Shop Facade are much higher than for the farther distances. The number of database cameras in this distance range from query cameras (average number of database cameras in particular distance range from query camera is shown in figure 3.4) in particular scenes could be one of the causes. But the error is a little bit higher for close ranges than for more distant ones also in the case of Shop Facade dataset, therefore it could have also other causes, e.g. the E_{5+1} algorithm produces poses with higher errors if query and database cameras are close to each other.

To further investigate this behaviour, an experiment with distance ranges from 0 m up to 2 m, width of each range was 0.1 m, was performed. To ensure sufficiently many database images, synthetic views were generated, as to eliminate issues caused by an insufficient number of database images. At the same time, this experiment will provide guiding information for question 2 from Section 2.4.

3.4.1 Experiment setup

The experiment pipeline is the same as in the previous experiment (see section 3.3.1), with two differences. Firstly, database images are synthetically generated and not real images. Synthetic images, place close to the query images, are used, as the real database images are typically not available in the close range (as discussed above and shown in the dataset visualizations in figure 3.1). Secondly, as the database cameras (their poses) are synthetically generated, there is no information about shared correspondences. Since the cameras are generated in such way that their field of view is more less the same as the field of view of the query camera, as correspondences shared between the query and database cameras are used all 3D points visible by the query camera.

Synthetically generated cameras

In the previous experiment, scenes with low errors on middle distance ranges, e.g., the St. Mary's Church or King's College scenes, had more than 10 usable database cameras per query. But the sufficient number of database cameras with shared correspondences is eliminating only one case of possible problems – the database cameras that are in the wrong configuration with

query camera. The second part is sufficient number of correspondences to eliminate the effect of point outliers, usually wrong correspondences. In the used experiment setup, wrong correspondences are not present, as there is one 3D point projected into two cameras. But what can happen and affect the correspondence precision is, e.g., when the projected 3D point is really far away from the cameras and when adding 1 px noise, it can, when computing, e.g., the reprojection error (or applying other loss function), behave as an outlier from the point of view of RANSAC. Therefore not only a sufficient number of generated cameras was required in this experiment, but also a sufficient number of correspondences. The average number of correspondences per query camera for the previous experiment is shown in figure 3.7. From that, it can be seen that except for King's College scene, the number of correspondences is smaller than 4 000. It is reasonable to assume that with 4 000 correspondences the effect of insufficient number of correspondences should be eliminated.

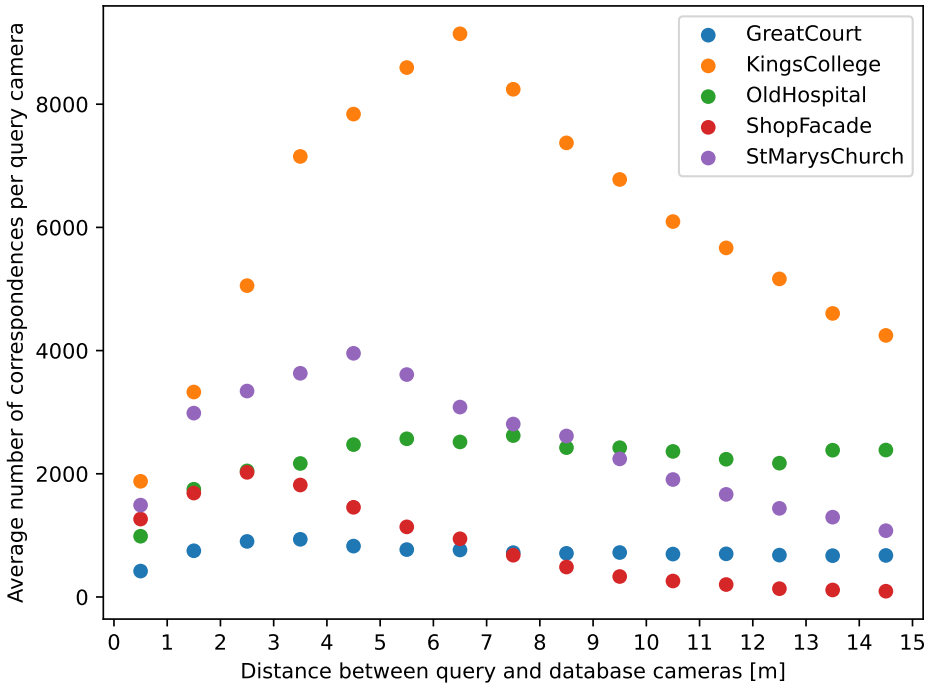


Figure 3.7: Average number of shared correspondences per query camera.

Therefore the number n_{dbcam} of database cameras generated for every query camera was set as

$$n_{\text{dbcam}} = \max\left(10, \left\lfloor \frac{4\,000}{n_{\text{vp}}} \right\rfloor\right) , \tag{3.3}$$

where n_{vp} is number of 3D point visible in the query camera. This is because in this experiment, all 3D point visible by the query camera are projected both in the query and all database cameras. It can be done so because

database cameras are generated in such way (details are described below) that it is reasonable to assume that generated cameras will see the 3D points visible by the query camera.

Synthetic database camera for one query camera in one distance range experiment was generated as follows:

1. Query camera projection centre was saved.
2. Random translation of unit length is generated from uniform distribution.
3. Translation vector is multiplied by random number (again from uniform distribution) in range given by examined distance range.
4. Generated camera projection centre is computed as sum of query camera projection centre and generated translation vector.
5. Generated camera orientation is generated, used methods are described below.
6. From projection centre C and orientation in form of rotation matrix R of generated camera is computed generated camera translation t as

$$t = -RC \ . \tag{3.4}$$

7. Generated translation and rotation (in the form of rotation quaternion) along with the query camera intrinsics were saved as a new synthetic camera.

Orientation generation

Three different methods how to generate orientation for the new camera were used in this thesis.

The first method is to simply copy the orientation of the query camera. Because only small translations of projection centre (up to 2 m) were applied, the database camera looking in the same direction as the query camera will still see the same 3D points as the query camera. This is not true generally, for cameras too close to 3D points it would be affected by field of view of the cameras. But this is not the case for datasets used in this thesis, where cameras are far enough from the captured 3D points. Camera generation can be simplified for this case – it is not needed to translate the projection centre and from that compute the translation vector again. It is possible to translate the translation vector itself and the projection center is translated accordingly. Projection centres distance is, since the rotation matrix

of both cameras is the same and rotation matrix does not change the norm of multiplied vector, equal to

$$\|C_{\mathrm{q}} - C_{\mathrm{db}}\| = \left\|-R_{\mathrm{q}}^T t_{\mathrm{q}} + R_{\mathrm{db}}^T t_{\mathrm{db}}\right\| = \left\|R^T\left(-t_{\mathrm{q}} + t_{\mathrm{db}}\right)\right\| = \|t_{\mathrm{q}} - t_{\mathrm{db}}, \| \quad . \tag{3.5}$$

The second method tries to simulate real image capturing. The orientation is generated in such a way that the resulting camera looks in direction given by a 3D point selected randomly from the 3D points visible by the query camera. This is done by changing the base vectors of camera coordinate system and compound rotation matrix from them. As can be seen in figure 3.8, depicting the pinhole camera model, direction in which the camera

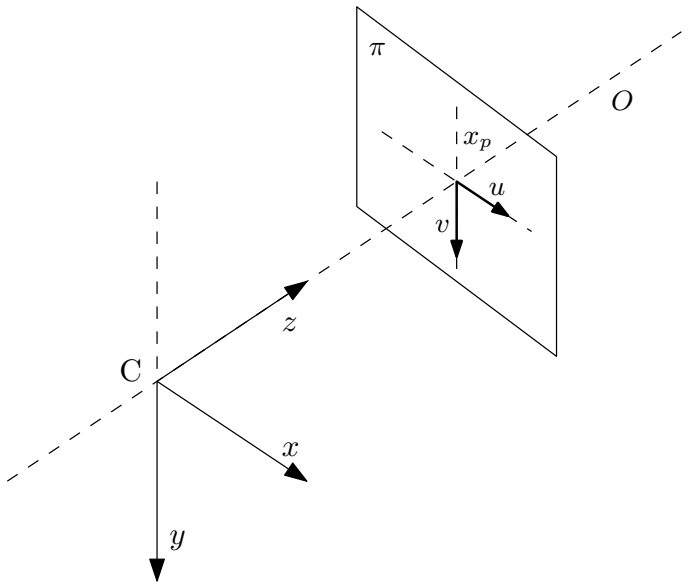


Figure 3.8: Pinhole camera model with camera and image coordinate systems. Camera coordinate system has center in projection centre C and consist of x, y and z vectors. Image coordinate system has centre at principal point x_p and consist of u and v vectors. O is optical axis and π is image plane.

looks is the direction of z-axis. Therefore the desired direction, computed as normalized difference between selected 3D point and generated camera (already translated) projection centre, is written in third column of new rotation matrix. Because rotation matrix must be orthogonal, also the other two columns (camera coordinate system basis vectors) need to be computed. First, it is computed new y-axis (second column of new rotation matrix), as vector (cross) product of the new z-axis and old x-axis (in this order), normalized to unit vector. New x-axis (first column of new rotation matrix) is computed as normalized vector product of new y-axis and new z-axis. The new rotation matrix is transformed into rotation quaternion and saved as newly generated database camera orientation. The difference between query camera orientation and database camera orientation generated using this

method depends on the position of randomly selected 3D point and can be larger than in the case of small rotation from third method. But this method generates orientation changes scalable with distance of 3D points from cameras. It operates with content of the projection plane (projected points) and this operation has similar results for all types of scenes, as opposed to the fixed (limited from above) rotation value from the third method.

The third method is to rotate the query camera randomly. This is done using rotation axis-angle representation. A random rotation vector is generated (from uniform distribution) and normalized to unit length. A random angle is drawn uniformly from the range -0.01 rad to 0.01 rad, which equals approximately to $0.57°$ and that is equivalent as if the 3D point in a distance of 15 m on which the camera looks shifts 30 cm. The generated vector is multiplied by generated angle and this angle-vector representation is transformed into a rotation matrix. Rotation matrix of query camera is then multiplied (from left) by generated small rotation and saved as newly generated database camera orientation.

3.4.2 Results

Median and mean values of visible points per query camera for all scenes are in table 3.2. When compared to the previous experiment, the number

scene	v. p. mean	v. p. median	gen. cam. mean
Great Court	1080	1162	12
King's College	1712	1765	10
Old Hospital	1312	1396	10
Shop Facade	1150	1192	10
St. Mary's Church	1507	1478	10

Table 3.2: Visible points and generated cameras statistics per query camera for all scenes, rounded to integers.

of database cameras is smaller than in case of real data, but the number of correspondences will be comparable (or even higher) to the real data, because all 3D points visible by query camera are used and not only subset shared with database cameras. Therefore the errors could be generally smaller in absolute values (in comparison to the previous experiment), because there is sufficient number of correspondences to choose from inside the RANSAC loop.

Median position errors measured on the Shop Facade scene are in figure 3.9. In absolute values, the median position errors measured for the other methods than same orientation one are comparable with the values for closest ranges measured on real data. The median position errors measured on St.

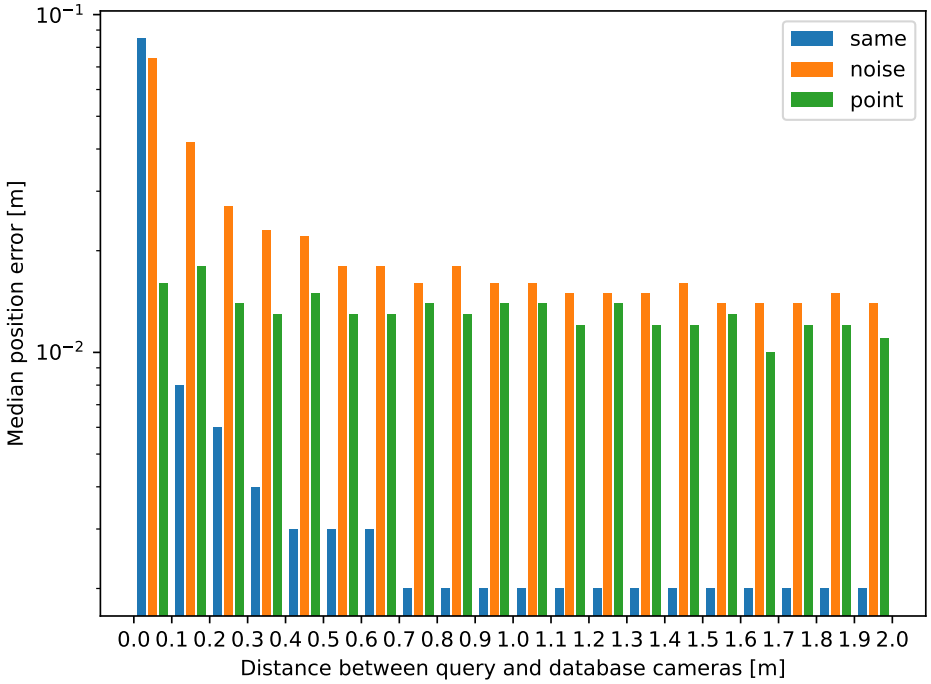


Figure 3.9: Dependency of median position error on distance between query and database cameras measured on the Shop Facade scene.

Mary's scene are very similar to the ones measured on the Shop Facade scene. These two scenes have similar also median orientation errors (orientation error measurement on St. Mary's Church scene shown in figure 3.11.

The median position errors for the method, where orientation is computed to point to random 3D point from the points visible in query mage, do not seem to depend on distance between query and database images (for all scenes). In comparison with other camera orientation generation methods, for some scenes it is higher – that is the case for the Great Court and King's College scenes – and for some lower (in comparison to the random noise method, same orientation method has the lowest error for all scenes, at least in distances larger than 0.5 m) – that applies for Old Hospital scene – or comparable – in case of Shop Facade and St. Mary's Church.

Median position errors measured for same orientation method (see figure 3.10) show dependency of median position error on query and database cameras distance. The values are by order higher for close distance ranges (up to approximately 0.3 m, depending on the scene). In this experiment, the error can no longer be affected by the low number of usable database cameras and shared correspondences or by some scene property, because this effect is visible on all scenes. Therefore it is reasonable to assume, that the higher errors (on close distances between query and database images, with similar orientation) are caused by the properties of E_{5+1} solver. The most likely

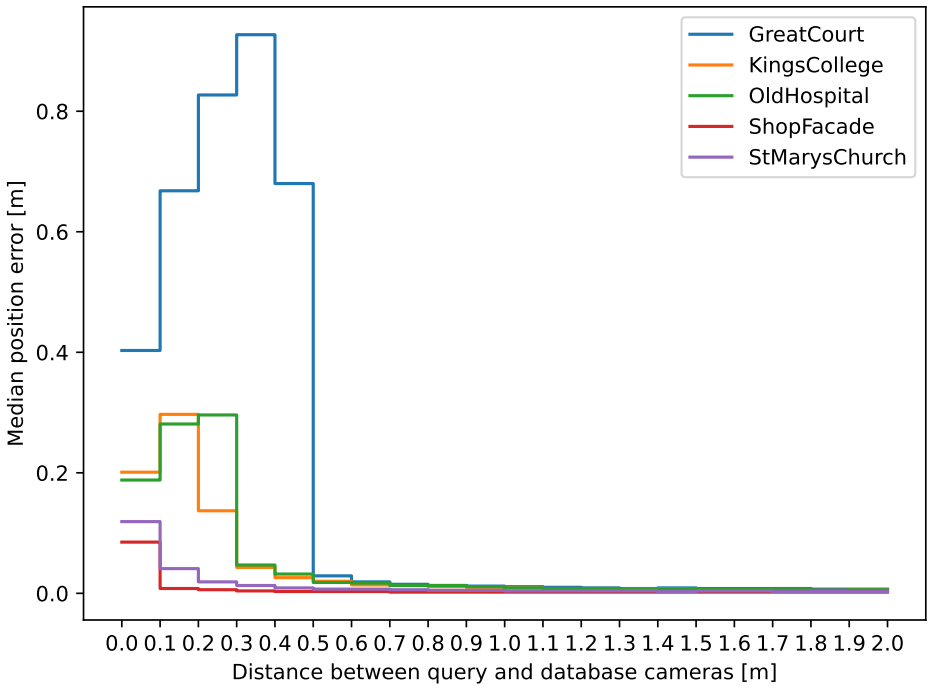


Figure 3.10: Dependency of median position error on distance between query and database camera for database cameras generated with the same orientation as the query camera.

explanation is that the E_{5+1} solver does not generate accurate translation estimated if the query image is too close to the query image. The errors on distant ranges are multiple times lower than the ones measured on real data in distance up to 2 m. This could mean that the precision for database cameras that are at least 0.5 m from query image, but has very similar rotation, increases.

Measured median position errors are in absolute values comparable or better than the errors measured in previous experiment for more distant query and database images and on real data. This can be caused by the small orientation difference between query image and generated images, whereas in case of real data, the database images orientation has probably wider distribution.

The median orientation errors of all three methods compared on St. Mary's Church scene are in figure 3.11. Although the values of errors measured for random noise method and direction towards random point method are not the same for all scenes (and it also varies which of the method produces higher errors), for all scenes hold that they are comparable (or even same) throughout all distance ranges. These methods do not show any dependency of orientation error on distance. The comparison of median orientation errors measured on all scenes with database cameras generated by the same

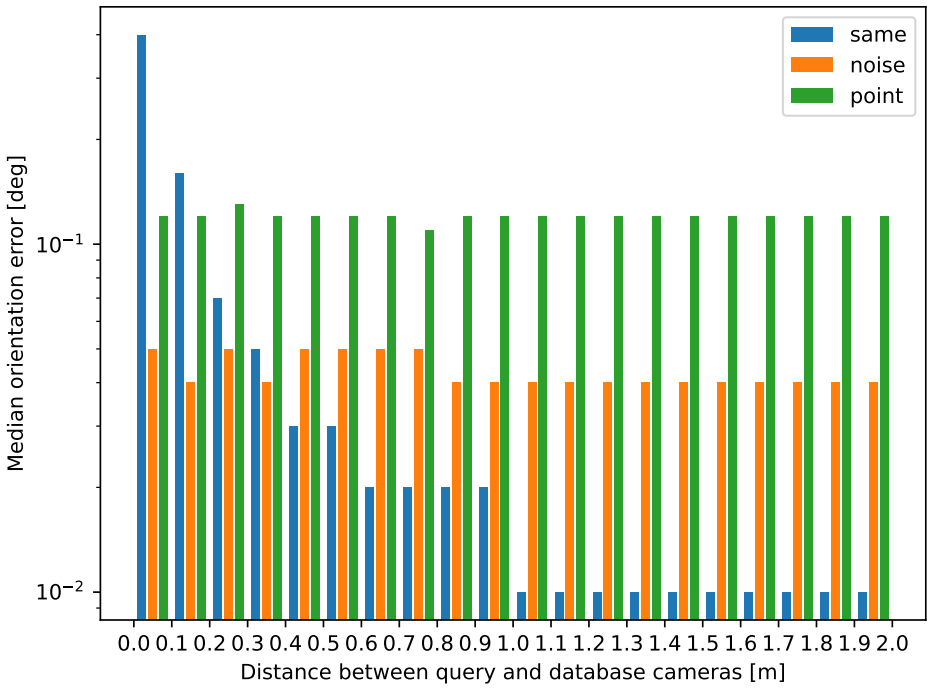


Figure 3.11: Dependency of median orientation error on distance between query and database camera measured on the St. Mary's Church scene.

orientation method is in figure 3.12. The orientation errors for this method are multiple times higher for close ranges (up to 0.1 to 0.5 m, depending on the scene. Same behaviour was observed on the median position errors for the same orientation database camera generation method. That supports the hypothesis that the high errors for close distance and similar query and database cameras orientations are caused by E_{5+1} solver.

3.4.3 Conclusion

The fact, that there exists camera generation method for which the median position and orientation errors are not dependent on the distance between query and database cameras suggests, that the accuracy of $E + 5 + 1$ solver does not depend solely on this distance. The errors measured with database images generated with same orientation as the query image suggest, that problematic is combination of close distance and similar orientation. Based on the results, I would suggest avoiding usage of database images that are close to the query image and have similar orientation with it. For simplicity of implementation, only the distance threshold (based on the experiment the suitable value appears to be 0.5 m) can be used and the orientation similarity does not need to be examined.

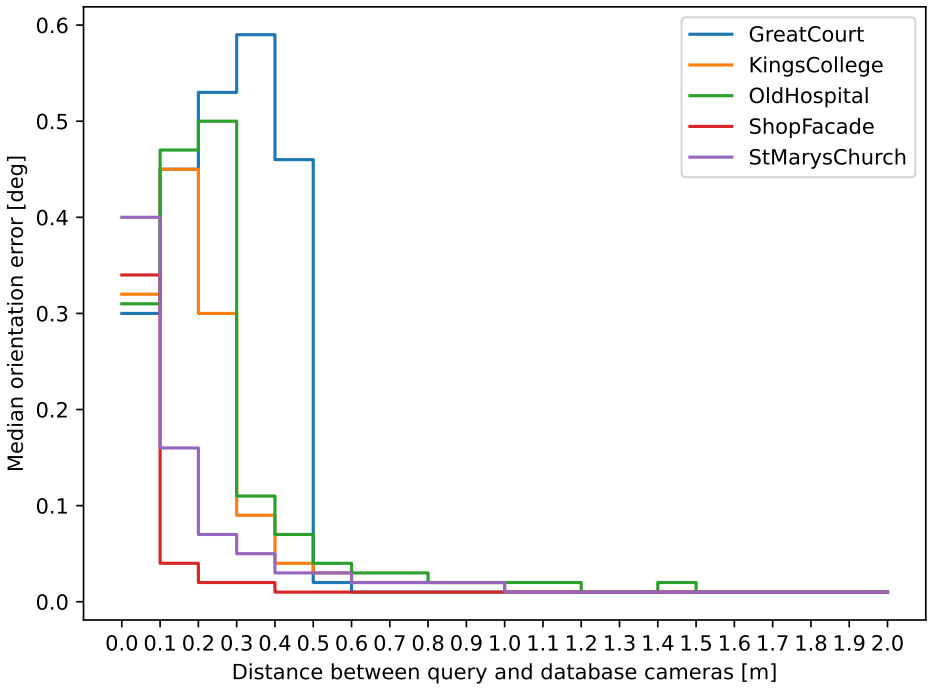


Figure 3.12: Dependency of median orientation error on distance between query and database camera, using same orientation camera generation method.

3.5 Experiments with database camera distance

Another factor, that can impact the query camera pose estimation accuracy is the distance between database cameras. The hypothesis is that the small distance between database cameras leads to worse accuracy (see question 3 in Section 2.4). Therefore the impact of minimum distance between every pair of database cameras on the pose estimate accuracy was tested.

3.5.1 Experiment setup

The experiment pipeline is similar to the one used in previous experiments except for the step where database cameras that satisfy the distance constraints are selected. In this experiment, distance between database cameras and query cameras was not restricted. Because there would not be sufficient number of database cameras for which the distance of each pair is in particular distance, only minimum distance between every pair of selected database cameras was restricted. That means that the distance between database cameras is constrained only from below. The selection process of suitable database cameras was following:

1. The database cameras with at least 10 shared correspondences were sorted in descending order according to the number of correspondences shared with query image.

2. Database number with highest number of shared correspondences is added as first to the list of selected database cameras.

3. The sorted list of database cameras is iterated. For every camera C_{db1} it is iterated over the list of already selected cameras.

4. For every camera C_{s} from the selected cameras list the distance between C_{db} and C_{s} is computed.

5. If this distance does not satisfy the distance constraint, next database camera C_{db} is examined.

6. If the distance satisfies the distance constraint for every C_{s} camera, it is added to the list of selected cameras.

This selection process creates the list of selected database cameras, where all cameras share at least 10 correspondences with the query camera and the distance between every pair of cameras satisfies the distance constraint. This is needed because in the E_{5+1} solver, two cameras are used and one of the simple ways how to ensure that the two database cameras selected by the RANSAC loop satisfy the distance constraint, is to select them from the pool where every two cameras satisfy the distance constraint.

3.5.2 Results

Median position errors are shown in figure 3.13 and median orientation errors are shown in figure 3.14. The general trend of both median position and orientation error is increasing with increasing minimum distance between database cameras. This holds for all scenes, although in absolute values both median position errors and median orientation errors for different scene varies significantly. The number of database cameras (and consequently the number of correspondences) decreases with increasing minimum distance between database cameras, which can be the cause of increasing errors. The increase in error can also be caused by the properties of E_{5+1} solver, but from this experiment, the results are not sufficiently proving any dependency. To get results not affected by the decreasing number of database cameras, experiment with same number of cameras for each minimum distance text would need to be performed.

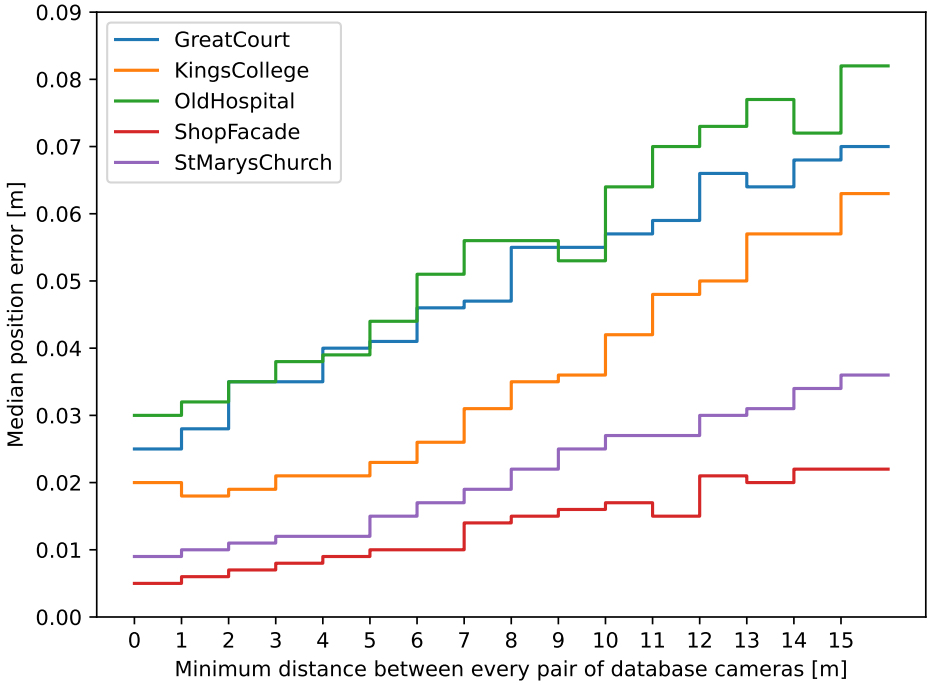


Figure 3.13: Dependency of median orientation error on minimum distance between database cameras.

3.5.3 Conclusion

The results of this experiment are not sufficiently convincing, therefore no conclusion regarding the selection of database cameras can be made, based on the results of this experiment.

3.5.4 Experiment with at most 10 database cameras per tested minimum distance

In this experiment, the setup was same as in the previous one, but the number of used database cameras was constrained from above to at most ten. The number of database cameras used should be the same as in experiment on closer distances between query and database cameras, but it could happen that the number of database cameras with large minimum distance will be smaller and therefore the results will say only slightly more than the previous experiment. Because the used database cameras are selected from top of the list sorted in descending order by number of shared correspondences and the number of shared correspondences is probable to be higher for closer images, in this experiment, although only minimum distance of database cameras is restricted, there will be preference on closer cameras (cameras close to the minimum distance limit).

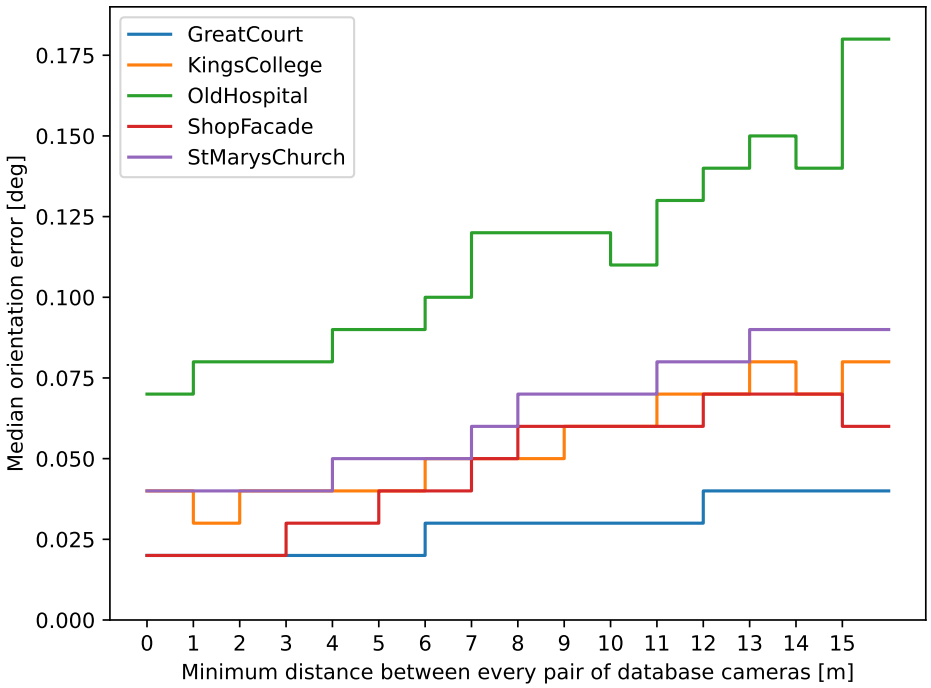


Figure 3.14: Dependency of median orientation error on minimum distance between database cameras.

3.5.5 Results

The dependency of average number of database cameras used on the minimum distance between database cameras is shown in figure 3.15. It can be seen that when increasing the minimum distance between database images, the number of usable database cameras drops and therefore only the results for minimum distance between database cameras up to 5 m can be compared, and the results for larger ranges can only be analyzed for the Great Court, Kings College and Old Hospital scenes.

Median position errors are shown in figure 3.16. The median position errors decrease (or not grow in case of the Shop Facade scene) for minimum distance between database cameras smaller than 2 m and for larger minimum distances position errors increase. Because for minimum distances up to 5 m the number of database images used is the same, this can not be caused by insufficient number of cameras (when compared to the results of previous experiment, the errors are in absolute value higher for the case with limited number of database cameras, but the number is limited for all minimum distances, therefore comparison between different minimum distances of database cameras is valid). As mentioned before, in this experiment, the database cameras in distances close to the minimum distance limit are implicitly preferred. Thus, it cannot be concluded, that the pose estimation

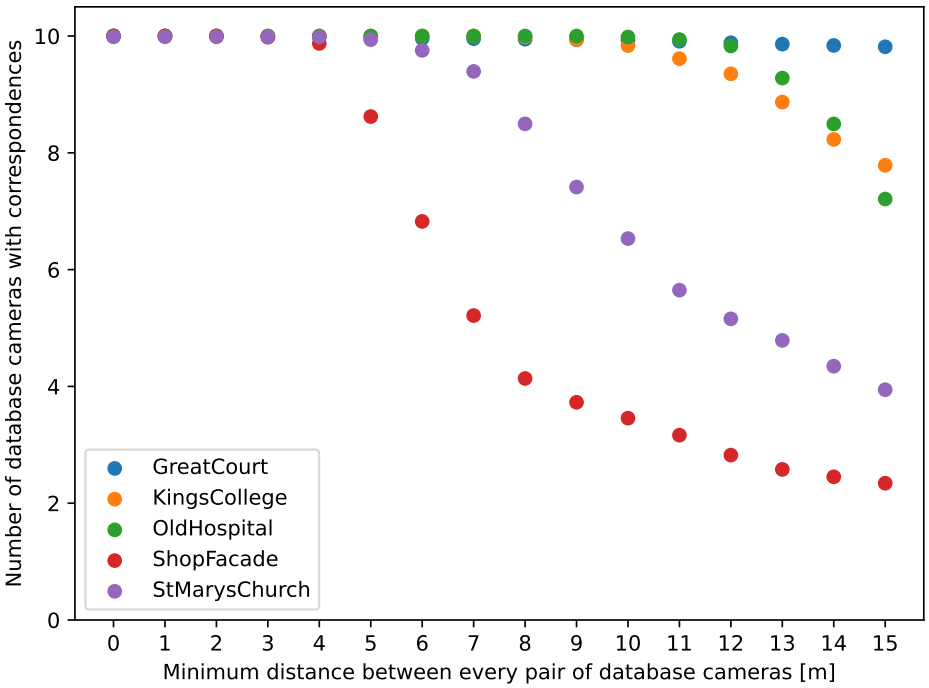


Figure 3.15: Average number of database cameras with shared correspondences with query images as a function of the distance between each pair of database images.

accuracy increases for minimum distance of database cameras up to 2 m because the constraint of minimum distance is not the only factor here, also the actual small distance between used database cameras could have an impact.

To distinguish between the impact of minimum database camera distance and impact of the limited number of cameras with most shared correspondences (and consequently the probable small distance between database cameras), it would be good to measure the distribution of the database cameras' distance for the database cameras that were used for the estimation (the ones chosen by RANSAC loop). But this is problematic in experiment setup used in this thesis, where the solver with RANSAC loop is used as black box from external library and it could require the modification of the PoseLib implementation of E_{5+1} solver pipeline. From the accessible data, there can be computed only distribution of distances between all pairs of database cameras selected as input for the E_{5+1} solver black box. But this distribution can have no relevance to the result (but it can have and it is not possible to decide, which case it is).

Same trend can be seen also on graph of median orientation errors in figure 3.17.

From the comparison of trends of median position and orientation errors

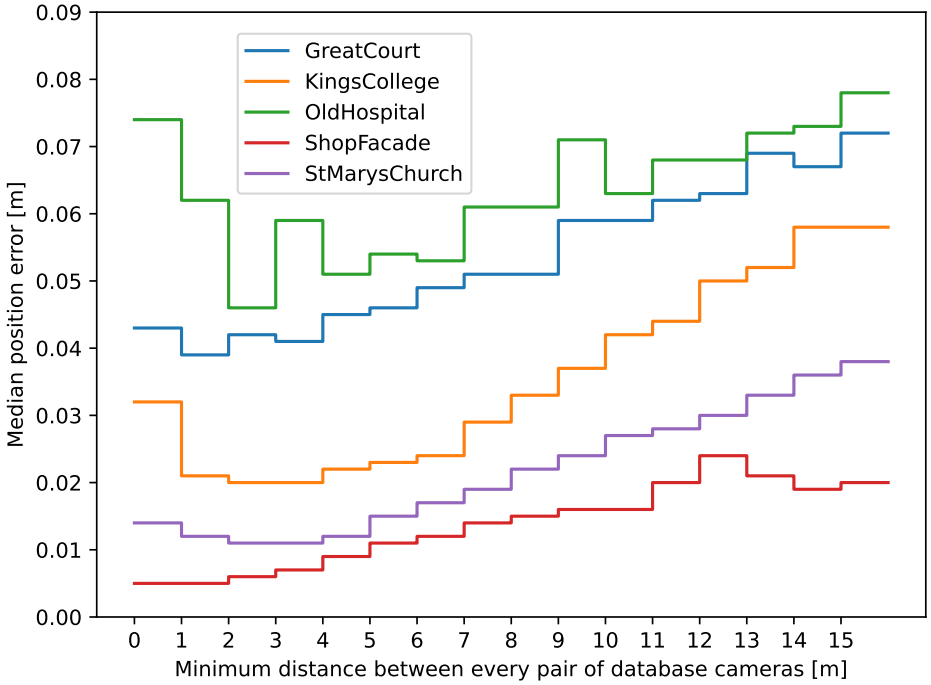


Figure 3.16: Dependency of median orientation error on distance between database cameras, if at most 10 database cameras are used.

measured on the Great Court, Kings College and Old Hospital scenes (the ones with not rapidly decreasing average number of database cameras) – which is increasing with increasing minimum database cameras distance – can be concluded that if the minimum database image distance is set to large distance, the accuracy of pose estimation decrease. From comparison of the absolute value of the median position error with the lowest errors measured in experiment with query and database cameras distance, it is reasonable to not set larger values than 7 m. This means that the inclusion of database cameras closer to each other can be beneficial.

The case of the Shop Facade scene is different. The median errors increase only with increasing minimum distance, the initial high errors and following drop is absent. There are two possible interpretations. First one is, that there is no dependency of the pose estimate accuracy on small minimum distance between database cameras (and the conclusions made based on the results on the rest of scenes are wrong). The second possibility is, that these results are caused by the nature of the scene. This scene is smaller than the rest of them – there are two perpendicular walls (the corner) of the building captured from the adjacent street. This means that most of the database images which see the same points are within a few meters from each other. And the database images far from each other are most likely to capture different walls. In this scene, as opposed to the scenes, where the 3D points are farther from the images, the angle between the projection rays of two database cameras close

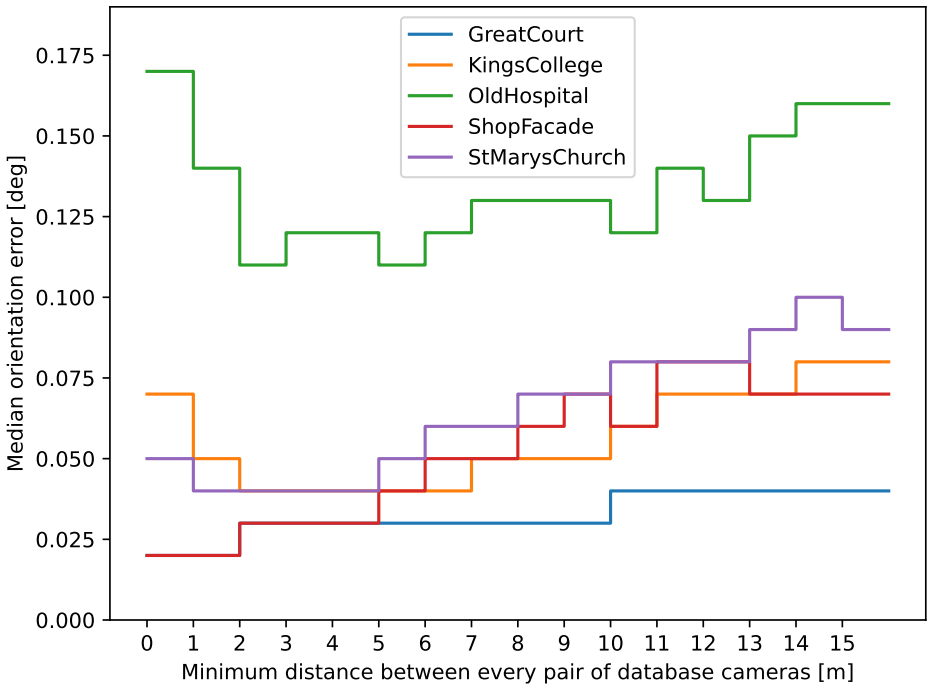


Figure 3.17: Dependency of median orientation error on distance between database cameras, if at most 10 database cameras are used.

to each other to the same 3D point can be large enough to obtain precise pose estimate. Thus, for the smaller scenes, it seems that the usage of database cameras close to each other is not problematic.

3.5.6 Conclusion

To conclude, for scenes, where 3D points are far from the cameras, the usage of database cameras too close to each other (less than 2 m) can decrease the accuracy of pose estimate. In case of scenes with 3D points close to the cameras, this is not happening.

Chapter 4

Conclusion

The aim of this work was to investigate how the accuracy of semi-generalized pose estimation algorithms, particularly the E_{5+1} algorithm from [3], depends on camera configuration and whether the accuracy can be improved by establishing constraints on choosing the used database cameras. Particularly, the constraints examined were the distance between used database cameras and the distance between the query camera and database cameras.

The pose accuracy is generally not affected by the distance between the pose of the query image and the poses of the database image. But this is not true for database cameras with similar orientation as the query camera. If these are used, then for close ranges (up to 0.5 m) the pose accuracy drops significantly. For spatially small scenes where images are captured with small camera rotation, the visual localization based on the semi-generalized pose estimation can be unsuitable.

In settings, when it is possible to render synthetic view of the scene from novel viewpoints, the results suggest to generate cameras with orientation similar to the query camera, but in distance from the query camera larger than 0.5 m.

The experiments on the impact of the distance between the database images on the pose accuracy do not show any significant results applicable generally. However, there are results suggesting, that in case of scenes, where the cameras are distant enough from the objects captured, use of database cameras too close to each other (not closer than 2 m) can be problematic. On the contrary in scenes, where the 3D points (the objects captured) are close to the cameras, using camera clusters is not an issue.

Besides the distance dependencies, it was discovered that the pose estimate accuracy is dependent on number of database cameras that share correspondences with query camera and also on the number of established correspon-

dences. Low number of database cameras and shared correspondences can increase the pose estimation accuracy significantly.

4.1 Future work

The results of the experiment on closer distances between query and database cameras suggest, that not only distance but also the orientation similarity between query and database camera does have an impact. Therefore this is suggested to be further examined.

In this thesis, the localization pipeline using E_{5+1} solver was tested only on outdoor (and therefore spatially large) scenes. The proposed improvement tricks could be tested also on smaller scenes (rooms or even smaller ones).

The E_{5+1} solver uses correspondences from two database cameras. One database camera is used to estimate the essential matrix and from the other one, the translation scale is extracted. Interesting question if the pose estimate accuracy does depend on pose of one of them differently than on the other one – that is, if, for example, the camera used for essential matrix estimation should be close and the one used for scale estimation far, or vice versa.

Bibliography

1. FISCHLER, Martin A.; BOLLES, Robert C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*. 1981, vol. 24, no. 6, pp. 381–395. ISSN 0001-0782. Available from DOI: 10.1145/358669.358692.
2. PERSSON, Mikael; NORDBERG, Klas. Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV. In: 2018, pp. 334–349. ISBN 978-3-030-01224-3. Available from DOI: 10.1007/978-3-030-01225-0_20.
3. ZHENG, Enliang; WU, Changchang. Structure from Motion Using Structure-Less Resection. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 2075–2083. Available from DOI: 10.1109/ICCV.2015.240.
4. BHAYANI, Snehal; SATTLER, Torsten; BARATH, Daniel; BELIANSKY, Patrik; HEIKKILÄ, Janne; KUKELOVA, Zuzana. Calibrated and Partially Calibrated Semi-Generalized Homographies. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 5916–5925. Available from DOI: 10.1109/ICCV48922.2021.00588.
5. BHAYANI, Snehal; SATTLER, Torsten; LARSSON, Viktor; HEIKKILÄ, Janne; KUKELOVA, Zuzana. Partially calibrated semi-generalized pose from hybrid point correspondences. In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 2881–2890. Available from DOI: 10.1109/WACV56688.2023.00290.
6. HENG, Lionel; CHOI, Benjamin; CUI, Zhaopeng; GEPPERT, Marcel; HU, Sixing; KUAN, Benson; LIU, Peidong; NGUYEN, Rang; YEO, Ye; GEIGER, Andreas; LEE, Gim; POLLEFEYS, Marc; SATTLER, Torsten. Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System. In: 2019, pp. 4695–4702. Available from DOI: 10.1109/ICRA.2019.8793949.

7. LIM, Hyon; SINHA, Sudipta N.; COHEN, Michael F.; UYTTENDAELE, Matthew. Real-time image-based 6-DOF localization in large-scale environments. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1043–1050. Available from DOI: 10.1109/CVPR.2012.6247782.
8. LYNEN, Simon; SATTLER, Torsten; BOSSE, Michael; HESCH, Joel; POLLEFEYS, Marc; SIEGWART, Roland. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In: 2015. Available from DOI: 10.15607/RSS.2015.XI.037.
9. LI, Yunpeng; SNAVELY, Noah; HUTTENLOCHER, Dan P. Location Recognition using Prioritized Feature Matching. In: *ECCV*. 2010.
10. SATTLER, Torsten; LEIBE, Bastian; KOBBELT, Leif. Fast Image-Based Localization using Direct 2D-to-3D Matching. In: *ICCV*. 2011.
11. ZHANG, Wei; KOSECKA, Jana. Image based Localization in Urban Environments. In: *International Symposium on 3D Data Processing, Visualization, and Transmission*. 2006.
12. BRACHMANN, Eric; KRULL, Alexander; NOWOZIN, Sebastian; SHOTTON, Jamie; MICHEL, Frank; GUMHOLD, Stefan; ROTHER, Carsten. DSAC — Differentiable RANSAC for Camera Localization. In: 2017, pp. 2492–2500. Available from DOI: 10.1109/CVPR.2017.267.
13. KENDALL, Alex; GRIMES, Matthew; CIPOLLA, Roberto. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: 2015, pp. 2938–2946. Available from DOI: 10.1109/ICCV.2015.336.
14. BRACHMANN, Eric; ROTHER, Carsten. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In: 2018, pp. 4654–4662. Available from DOI: 10.1109/CVPR.2018.00489.
15. CAVALLARI, Tommaso; BERTINETTO, Luca; MUKHOTI, Jishnu; TORR, Philip; GOLODETZ, Stuart. Let's Take This Online: Adapting Scene Coordinate Regression Network Predictions for Online RGB-D Camera Relocalisation. In: 2019, pp. 564–573. Available from DOI: 10.1109/3DV.2019.00068.
16. BALNTAS, Vassileios; LI, Shuda; PRISACARIU, Victor. RelocNet: Continuous Metric Learning Relocalisation using Neural Nets. In: *The European Conference on Computer Vision (ECCV)*. 2018.
17. SHOTTON, Jamie; GLOCKER, Ben; ZACH, Christopher; IZADI, Shahram; CRIMINISI, Antonio; FITZGIBBON, Andrew. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2930–2937. Available from DOI: 10.1109/CVPR.2013.377.
18. SATTLER, Torsten; ZHOU, Qunjie; POLLEFEYS, Marc; LEAL-TAIXÉ, Laura. Understanding the Limitations of CNN-Based Absolute Camera Pose Regression. In: 2019, pp. 3297–3307. Available from DOI: 10.1109/CVPR.2019.00342.

19. BRACHMANN, Eric; HUMENBERGER, Martin; ROTHER, Carsten; SATTLER, Torsten. On the Limits of Pseudo Ground Truth in Visual Camera Re-localisation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6218–6228.

20. LOWE, David G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* 2004, vol. 60, no. 2, pp. 91–110. Available from DOI: 10.1023/B:VISI.0000029664.99615.94.

21. SCHÖNBERGER, Johannes Lutz; FRAHM, Jan-Michael. Structure-from-Motion Revisited. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

22. TORR, Philip; ZISSERMAN, A. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding*. 2000, vol. 78, pp. 138–156. Available from DOI: 10.1006/cviu.1999.0832.

23. CHUM, Ondřej; MATAS, Jiří; KITTLER, Josef. Locally Optimized RANSAC. In: MICHAELIS, Bernd; KRELL, Gerald (eds.). *Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 236–243. ISBN 978-3-540-45243-0.

24. MARTÍNEZ-OTZETA, José María; RODRÍGUEZ-MORENO, Itsaso; MENDIALDUA, Iñigo; SIERRA, Basilio. RANSAC for Robotic Applications: A Survey. *Sensors*. 2023, vol. 23, no. 1. ISSN 1424-8220. Available from DOI: 10.3390/s23010327.

25. PHILBIN, James; CHUM, Ondrej; ISARD, Michael; SIVIC, Josef; ZISSERMAN, Andrew. Object retrieval with large vocabularies and fast spatial matching. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8. Available from DOI: 10.1109/CVPR.2007.383172.

26. ARANDJELOVIĆ, Relja; GRONAT, Petr; TORII, Akihiko; PAJDLA, Tomas; SIVIC, Josef. NetVLAD: CNN architecture for weakly supervised place recognition. In: *CVPR*. 2016.

27. NAYAR, Shree K.; GROSSBERG, Michael D. A General Imaging Model and a Method for Finding its Parameters. In: *Computer Vision, IEEE International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, 2001, vol. 1, p. 108. Available from DOI: 10.1109/ICCV.2001.937611.

28. PLESS, Robert. Using Many Cameras as One. In: *CVPR 2003: Computer Vision and Pattern Recognition Conference*. Los Alamitos, CA, USA: IEEE Computer Society, 2003, vol. 3, p. 587. ISSN 1063-6919. Available from DOI: 10.1109/CVPR.2003.1211520.

29. STEWÉNIUS, Henrik; NISTÉR, David; OSKARSSON, Magnus; ÅSTRÖM, Kalle. Solutions to minimal generalized relative pose problems. *Workshop on omnidirectional vision*. 2005.

30. NISTÉR, David. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2004, vol. 26, no. 6, pp. 756–770. Available from DOI: 10.1109/TPAMI.2004.17.

31. STEWENIUS, Henrik; NISTER, David; KAHL, Fredrik; SCHAFFALITZKY, Frederik. A minimal solution for relative pose with unknown focal length. In: 2005, vol. 2, 789–794 vol. 2. ISBN 0-7695-2372-2. Available from DOI: 10.1109/CVPR.2005.36.

32. BUJNAK, Martin; KUKELOVA, Zuzana; PAJDLA, Tomas. 3D reconstruction from image collections with a single known focal length. In: *2009 IEEE 12th International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 1803–1810. ISSN 1550-5499. Available from DOI: 10.1109/ICCV.2009.5459402.

33. PANEK, Vojtech; KUKELOVA, Zuzana; SATTLER, Torsten. MeshLoc: Mesh-Based Visual Localization. In: AVIDAN, Shai; BROSTOW, Gabriel; CISSÉ, Moustapha; FARINELLA, Giovanni Maria; HASSNER, Tal (eds.). *Computer Vision – ECCV 2022*. Cham: Springer Nature Switzerland, 2022, pp. 589–609. ISBN 978-3-031-20047-2.

34. LARSSON, Viktor. *PoseLib – Miniminimal Solvers for Camera Pose Estimation* [online]. 2020 [visited on 2023-03-29]. Available from: https://github.com/vlarsson/PoseLib.

35. DUSMANU, Mihai; LINDENBERGER, Philipp; LAMBERT, John; SARLIN, Paul-Edouard, et al. *pycolmap – Python bidnings for COLMAP* [online]. 2023 [visited on 2023-03-15]. Available from: https://github.com/colmap/pycolmap.

36. SCHÖNBERGER, Johannes Lutz; ZHENG, Enliang; POLLEFEYS, Marc; FRAHM, Jan-Michael. Pixelwise View Selection for Unstructured Multi-View Stereo. In: *European Conference on Computer Vision (ECCV)*. 2016.

37. VIRTANEN Pauli, et al. *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python* [online]. 2020 [visited on 2023-05-08]. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.transform.Rotation.html.

38. KENDALL, Alex; GRIMES, Matthew; CIPOLLA, Roberto. Convolutional networks for real-time 6-DOF camera relocalization. *CoRR*. 2015, vol. abs/1505.07427. Available from arXiv: 1505.07427.

39. ZHANG, Zichao; SATTLER, Torsten; SCARAMUZZA, Davide. Reference Pose Generation for Long-term Visual Localization via Learned Features and View Synthesis. *International Journal of Computer Vision*. 2021, vol. 129, pp. 1–24. Available from DOI: 10.1007/s11263-020-01399-8.

40. YU, Hailin; FENG, Youji; YE, Weicai; JIANG, Mingxuan; BAO, Hujun; ZHANG, Guofeng. *Improving Feature-based Visual Localization by Geometry-Aided Matching*. 2023. Available from arXiv: 2211.08712 [cs.CV].

41. WU, Changchang. Towards Linear-Time Incremental Structure from Motion. In: *2013 International Conference on 3D Vision - 3DV 2013*. 2013, pp. 127–134. Available from DOI: 10.1109/3DV.2013.25.

42. ŠÁRA, Radim. 3D Computer Vision lecture slides [online]. 2022 [visited on 2023-05-05]. Available from: http://cmp.felk.cvut.cz/cmp/courses/TDV/2022W/lectures/tdv-2022-all.pdf.