BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

RECONSTRUCTION OF 3D INFORMATION ABOUT VEHICLES PASSING IN FRONT OF A SURVEILLANCE CAMERA

REKONSTRUKCE 3D INFORMACÍ O AUTOMOBILECH Z PRŮJEZDŮ PŘED DOHLEDOVOU KAMEROU

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. PETR DOBEŠ**
AUTOR PRÁCE

SUPERVISOR **prof. Ing. ADAM HEROUT, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Dobeš Petr, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Rekonstrukce 3D informací o automobilech z průjezdů před dohledovou kamerou**

Reconstruction of 3D Information about Vehicles Passing in front of a Surveillance Camera

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s problematikou monitorování dopravy kamerou umístěnou u silnice.
2. Nastudujte algoritmy stereo-vidění, 3D rekonstrukce a počítání optického toku.
3. Získejte a pořiďte vhodná data pro experimentování se získáváním prostorových údajů o projíždějících autech.
4. Experimentujte s různými přístupy k získávání prostorových informací o projíždějících autech.
5. Vytvořte nástroje pro vyhodnocování prováděných experimentů.
6. Učiňte závěry z provedených experimentů a zhodnoťte možnosti pořizování prostorových dat z dohledových kamer v dopravě.
7. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 až 5.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herout Adam, prof. Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstract

This master's thesis focuses on 3D reconstruction of vehicles passing in front of a traffic surveillance camera. Calibration process of surveillance camera is first introduced and the relation of automatic calibration with 3D information about observed traffic is described. Furthermore, Structure from Motion, SLAM, and optical flow algorithms are presented. A set of experiments with feature matching and the Structure from Motion algorithm is carried out to examine results on images of passing vehicles. Afterwards, the Structure from Motion pipeline is modified. Instead of using SIFT features, DeepMatching algorithm is utilized to obtain quasi-dense point correspondences for the subsequent reconstruction phase. Afterwards, reconstructed models are refined by applying additional constraints specific to the vehicle reconstruction task. The resultant models are then evaluated. Lastly, observations and acquired information about the process of vehicle reconstruction are utilized to form proposals for prospective design of an entirely custom pipeline that would be specialized for 3D reconstruction of passing vehicles.

Abstrakt

Tato diplomová práce se zabývá 3D rekonstrukcí vozidel projíždějících před dohledovou kamerou. V práci je nejprve představena kalibrace dohledové kamery a souvislost automatické kalibrace s 3D informacemi o sledované dopravě. Dále jsou představeny algoritmy Structure from Motion a SLAM, společně s metodami pro odhad optického toku. Za účelem prozkoumání chování pro snímky projíždějících vozidel jsou provedeny experimenty s výpočtem korespondencí a algoritmem Structure from Motion. Následně je postup algoritmu Structure from Motion upraven. SIFT příznaky jsou nahrazeny algoritmem DeepMatching za účelem získání hustých bodových korespondencí pro následnou fázi rekonstrukce. Rekonstruované modely jsou dále zpřesněny aplikováním dodatečných omezení, která jsou specifická pro rekonstrukci projíždějících vozidel. Získané modely jsou poté vyhodnoceny. Veškeré zjištěné poznatky a informace o rekonstrukci vozidel jsou pak využity k navržení dalších modifikací, které by vedly k vytvoření zcela vlastního rekonstrukčního postupu, specializovaného přímo pro 3D rekonstrukci projíždějících vozidel.

Keywords

3D Reconstruction, Structure from Motion, Traffic Surveillance, Traffic Analysis, Camera Calibration

Klíčová slova

3D rekonstrukce, Structure from Motion, Monitorování dopravy, Analýza dopravy, Kalibrace kamery

Reference

DOBEŠ, Petr. *Reconstruction of 3D Information about Vehicles Passing in front of a Surveillance Camera.* Brno, 2017. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Herout Adam.

Reconstruction of 3D Information about Vehicles Passing in front of a Surveillance Camera

Declaration

I declare that this master's thesis has been composed entirely by myself under the supervision of prof. Ing. Adam Herout, Ph.D. All sources of information used in this thesis are cited and included in the list of references.

. . . .
Petr Dobeš
May 24, 2017

Acknowledgements

I would like to thank my supervisor Adam Herout for his support, as well as numerous valuable recommendations and inspiring ideas. I would also like to thank Jakub Sochor and Lukáš Polok for their help and suggestions.

Contents

Chapter 1

Introduction

Deployment of high-resolution digital cameras in traffic surveillance has increased the need for computer vision algorithms that automatically extract data from captured video streams. When supplemented with computer vision methods, traffic surveillance cameras can serve a wide range of purposes, such as counting of passing vehicles, their classification, finding driving lanes, detecting traffic jams, and discovering drivers in the opposite direction. Moreover, the primary aim of many traffic surveillance systems is to measure the speed of passing vehicles. Nevertheless, many of the tasks cannot be achieved without preceding camera calibration.

This work addresses the problem of reconstruction of 3D information about vehicles passing in front of a surveillance camera. In existing algorithms developed for automatic traffic surveillance, the only obtained 3D data about a passing vehicle is a bounding box constructed using a segmented 2D vehicle blob. This work therefore aims to examine the potential of acquiring more precise 3D representation of a vehicle captured in a video stream. Such information is desired not only for visualization purposes, but may also be utilized to determine the scale of the projected scene, and thus contribute to the camera calibration process.

The primary aim of this thesis is to explore the possibilities for 3D reconstruction of passing vehicles. Available 3D reconstruction tools are first examined and tested to find out whether they could directly be used for the outlined task. Subsequently, a set of experiments carried out with keypoint extraction is described. Moreover, additional constraints applicable to reconstruction of passing vehicles are inspected. The reconstruction pipeline is then modified and the resultant point cloud models are evaluated with respect to the intended use for scale inference in a traffic analysis system. Lastly, observations and acquired information specific to the task of vehicle reconstruction are utilized to form proposals for prospective design of an entirely custom pipeline that would be specialized for 3D reconstruction of passing vehicles.

Chapter 2

Traffic Surveillance using Monocular Camera and its Calibration

Monocular cameras can be utilized in numerous tasks of traffic analysis and surveillance, one of which is speed measurement of passing vehicles. Techniques for visual speed measurement have been developed by various authors [27, 12, 6, 9, 28]. The growing importance of this capability of roadside cameras is also illustrated by a recent comprehensive dataset published by Sochor et al. [28] to address the lack of available data with reliable ground truth, and to enable comparison of various speed measurement methods.

Nevertheless, many of the traffic surveillance tasks, especially accurate speed measurement, require precise calibration of the particular roadside camera. This chapter therefore focuses on the approaches to calibration of monocular camera employed in traffic surveillance.

2.1 Traffic Camera Calibration

Traffic camera calibration can be performed either manually or fully automatically. As standard pattern-based approaches (such as the one developed by Zhang [36]) cannot be used [27], manual calibration requires user input of some information about the scene that is viewed by the camera. Such approach often relies on physical measurements in the scene or on placement of specific markers, and thus involves considerable amount of effort. This renders manual calibration impractical for large-scale deployment of roadside cameras. It is therefore desirable for the calibration to be fully automatic [8, 27].

Standard camera calibration process involves finding its intrinsic parameters (matrix $\mathbf{K}$) and extrinsic parameters (matrix $[\mathbf{R}\,\mathbf{T}]$) that form the projection matrix $\mathbf{P}$:

$$\mathbf{P} = \mathbf{K}\,[\mathbf{R}\,\mathbf{T}] \tag{2.1}$$

However, for the purpose of speed measurement in visual traffic surveillance, it is more convenient to define the problem of camera calibration as finding the intrinsic parameters, determining the road plane, and finding the scale of the road plane. This approach is more suitable, as it enables direct speed measurement of vehicles driving on the road plane. This concept of camera calibration can be considered equivalent and methods exist to convert the obtained parameters to the above mentioned standard camera model [27].

When determining the intrinsic parameters, the camera is expected to exhibit zero pixel skew and to have principal point in the center of the image. This can be safely assumed in case of practically used surveillance cameras [8]. The only remaining intrinsic parameter to determine is therefore the camera's focal length. This parameter can be calculated using two vanishing points. Once vanishing points are determined, the parameters of the road plane (without scale) can also be obtained. The scale of the road plane is thus the last necessary parameter to infer [27].

2.2 Detection of Vanishing Points

Detection of vanishing points is the key part in finding camera's focal length and parameters of the road plane. Some methods, for example those developed by Cathley et al. [6] and He et al. [12], are based on acquiring the information from lane markings. However, this approach usually requires a high number of well-visible markings to be available.

Other methods, such as the one presented by Dubská et al. [8], base the detection of the vanishing points on observed motion of vehicles. Dubská et al. [8] track feature points on passing vehicles to form line segments that are subsequently accumulated in *diamond space* (a scheme utilizing parallel coordinates and Hough transform), using which they obtain the first vanishing point in the direction of the traffic flow. The second vanishing point, which is perpendicular to the first direction, is then acquired by accumulating the edges on the vehicles that do not aim towards the first vanishing point. Authors also point out that in case the optics of the camera is not free from radial distortion, its compensation should precede the camera calibration phase. Furthermore, an algorithm for radial distortion correction using tracked vehicle paths is proposed.

Once positions of two vanishing points in the image space are obtained, focal length of the camera can be calculated. Two vectors from the origin of the camera system can then be constructed from the coordinates of the vanishing points and the focal length. Cross product of these two vectors then yields the normal vector of the road plane. The only remaining parameter is thus the distance of the road plane from the camera which establishes the relation between the image and real-world units, i.e. the scale [27].

2.3 Determining the Scale of the Road Plane

In case of manual calibration, the scale of the road plane can be determined using input of previously measured lengths within the viewed scene. Nevertheless, automatic methods cannot rely on any static physical objects in the scene having precisely given dimensions, and thus they have to infer the scale from observed traffic. Whenever passing vehicles are used as the source of information to obtain the scale of the road plane, camera calibration inevitably becomes closely related to 3D structure of the vehicles. The following subsections summarize two significant approaches used to determine the scale of the road plane from observed traffic flow.

2.3.1 Scale Inference Using 3D Bounding Boxes

Dubská et al. [9] use 3D bounding boxes of passing vehicles and statistical domain adaptation of their dimensions to infer scene scale. They first detect vehicle blobs using background modeling and foreground detection. Furthermore, shadows are removed. Bounding box projection is then created for each vehicle blob. Lines that pass through vanishing points and that are tangent to the vehicle blob are constructed. The first three corner points of the bounding box are obtained from intersections of these lines. Every remaining corner of the bounding box is then found using intersection of two lines, each passing through one of the already obtained corner points and through the corresponding vanishing point. Steps of the bounding box construction are shown in Figure 2.1.

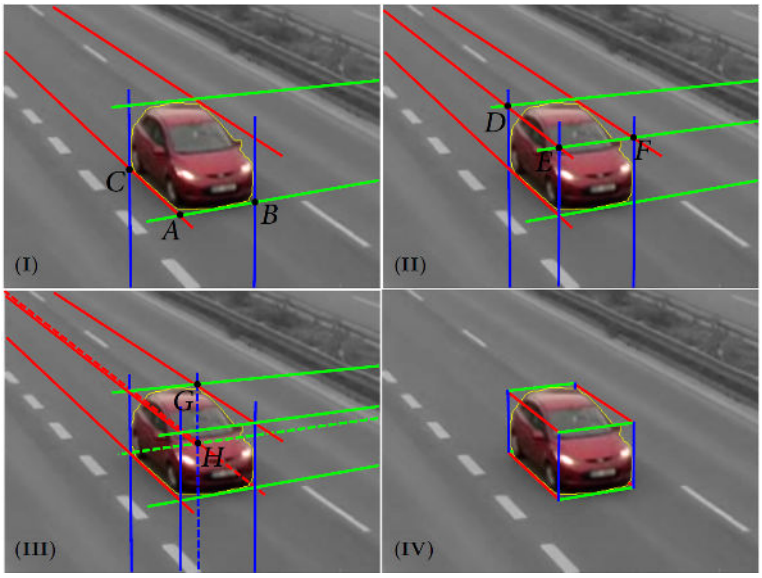

Figure 2.1: Construction of 3D bounding box of vehicle by Dubská et al. [9]. The method is based on lines passing through vanishing points. The first three corner points are obtained using lines that are tangent to the edges of the vehicle blob. The remaining corners are found as intersections of lines passing through vanishing points and already obtained corners.

Once image coordinates of corners of the bounding box are known, it is possible to project the base of the bounding box onto the road plane. As a result, coordinates of the bounding box base in 3D space are acquired. The distance between these coordinates, together with the information about the real-world dimensions of the vehicle, can be used to determine the scale of the road plane. In other words, it is possible to calculate the scale factor λ, which establishes the relation between relative units of the road plane and the real-world units of length (e.g. meters) [9].

In order to determine the scale factor, Dubská et al. [9] collected statistical data about sold cars and their dimensions, and subsequently formed a histogram of their bounding box dimensions. Scene scale was then determined by fitting statistics of known dimensions and the measured data from the observed traffic. The camera calibration process is therefore automatic, and the only necessary input is the statistical information about dimensions of sold cars.

Evaluation of this method in Sochor et al. [28] shows that the mean error in case of speed measurements is 10.89%. Nevertheless, the assessment of the method also suggests that a significant part of the error is probably caused by improper localization of the second vanishing point, and not only by the approach used for scene scale inference.

It is also important to note that when extraction of 3D information about observed traffic is considered, bounding boxes have been so far the only 3D information obtained about passing vehicles. Moreover, edges of the vehicle blobs tend to be bent, and thus constructing the bounding boxes using lines tangent to these edges has negative influence on the overall accuracy.

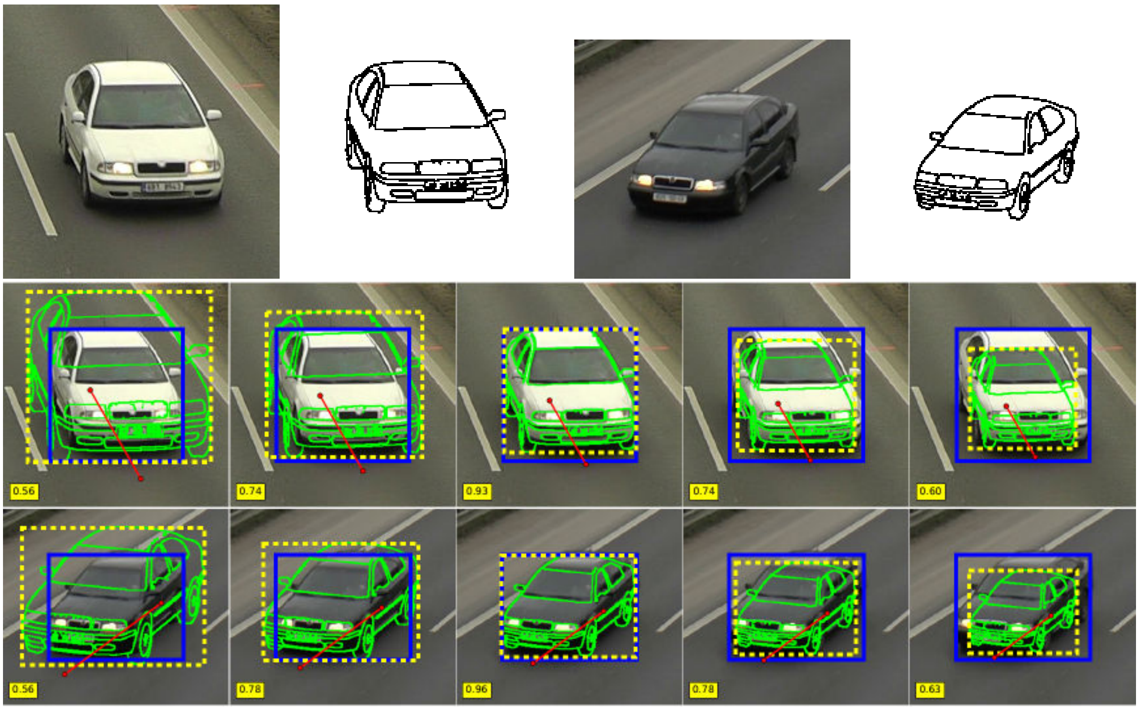

Figure 2.2: Scene scale inference by rendering 3D vehicle models under known viewpoint and aligning their 2D bounding boxes with the bounding boxes of detected vehicles. (Only edges of the rendered models are shown.) Adapted from [27].

2.3.2 Scale Inference by 3D Model Alignment

Sochor et al. [27] infer the scene scale using 3D models of frequently passing cars. They use fine-grained information about vehicle type (i.e. make, model, variant, model year) and obtain 3D models for two vehicle types that are commonly observed. The 3D model with available real-world dimensions is then used to infer the scene scale. An example is shown in Figure 2.2.

The method starts with classification of vehicles in the video stream using a convolutional neural network. When vehicle type with available 3D model is detected, it is further processed. First, the method used by Dubská et al. [9] (described in Subsection 2.3.1) is used to extract the 3D bounding box of the detected vehicle blob, and the center of its base in image coordinates is obtained. Subsequently, the viewpoint vector from the vehicle to the camera is computed. The 3D model of the particular vehicle class is then rendered onto the image under the same viewpoint and at the same position. The only remaining unknown parameter is its size. The image of the 3D model is therefore rendered in multiple different scales and its 2D bounding box is matched with the 2D bounding box of the detected vehicle blob using intersection-over-union metric (IoU).

Once the rendered 3D model is aligned to the detected vehicle in the image, two points representing the front and the rear of the vehicle are projected to the road plane. Knowing the real-world distance of these points from available vehicle dimensions provides sufficient information for the scene scale to be calculated. To obtain the final scene scale, all sizes of the rendered model (with IoU metric above given threshold) are taken into account, and the final scale is computed using kernel density estimation. Evaluation of this approach to scale inference (which also includes improved detection of the second vanishing point) shows that the mean error of speed measurement is 1.39% [27].

2.4 Contribution of This Work to Camera Calibration Process

As this work aims to reconstruct 3D information about vehicles passing in front of a surveillance camera, the extracted 3D data can contribute to further improvement of the camera calibration process. In particular, obtained 3D models could provide additional information for the calibration phase in which scale of the road plane is computed.

Fine-grained classification of detected vehicles could be used to distinguish between various vehicle models. Real-world dimensions would also be stored for each vehicle model. Once particular vehicle with known dimensions is recognized, its detailed 3D reconstruction may be created and utilized to infer the scene scale. Unlike the method where rendered 3D model alignment is used (as described in Subsection 2.3.1), this approach would only require the information about vehicles' dimensions to be available in the traffic surveillance system, and no prior 3D model data would be necessary.

Chapter 3

Utilized Computer Vision Methods

This chapter introduces computer vision algorithms that have been used throughout the work on this thesis. First, Structure from Motion algorithm is described. Secondly, SLAM (Simultaneous Localization and Mapping) is introduced. Thirdly, the concepts of optical flow and DeepMatching are addressed.

3.1 Structure from Motion

Structure from Motion (SfM) is an algorithm used for 3D reconstruction from image collections. Several implementations of this reconstruction strategy exist, such as COLMAP [22], Bundler [25], and VisualSFM [35]. This section introduces and describes individual phases of incremental Structure from Motion algorithm. Provided information is based on Schönberger et al. [22].

The general pipeline of incremental Structure from Motion is shown in Figure 3.1. The input to the SfM is a set of unordered images with projections of a scene that is to be reconstructed. The first stage of the SfM pipeline consists of correspondence search and is followed by the second stage that is represented by an iterative reconstruction component. The output of SfM is sparse 3D reconstruction in the form of a point cloud. An example of resultant SfM output is shown in Figure 3.2.

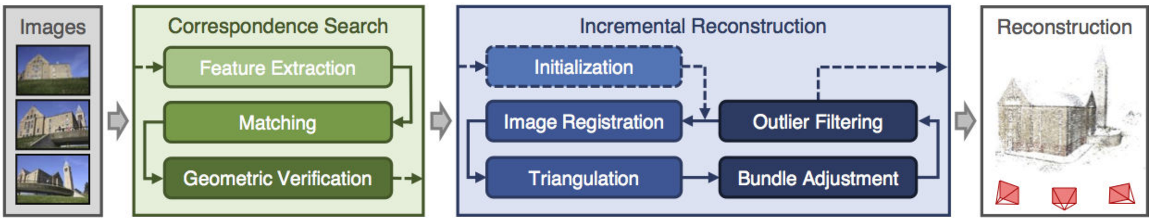

Figure 3.1: General pipeline of incremental Structure from Motion algorithm. Obtained from [22].

3.1.1 Correspondence Search

The first stage of the Structure from Motion pipeline (Fig. 3.1) is correspondence search. This stage involves extraction of local feature points, identification of corresponding pro-

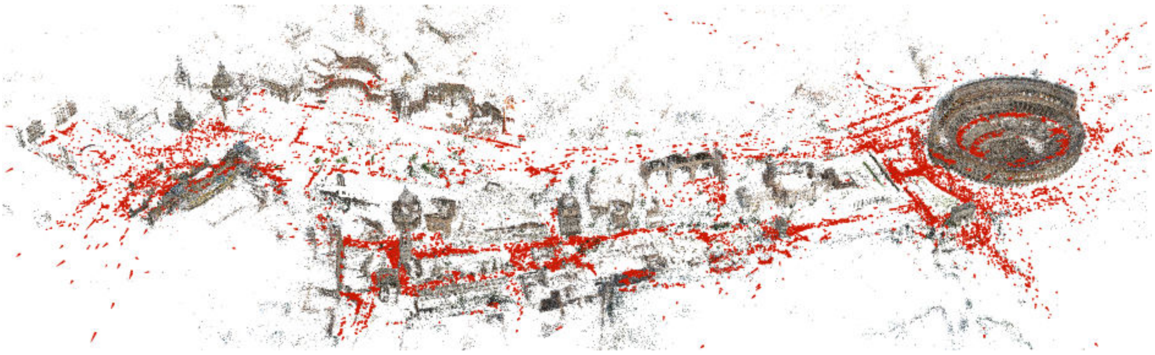

Figure 3.2: Example of a point cloud of central Rome reconstructed using 21K photos by COLMAP. Red dots represent positions of cameras. Obtained from [22].

jections of the same points in overlapping images (matching), and subsequent geometric verification of the found matches.

Feature extraction encompasses detecting coordinates of feature points within each image and representing the points using descriptors. These points need to be distinctive in order to be uniquely recognized in multiple images, and thus SIFT [18] is a common choice in many implementations, including COLMAP. Next, sets of feature points are matched using similarity metric to find corresponding point pairs. Either all feature pairs within every possible image pair can be considered, or approximate approaches can be used for large image collections in order to avoid prohibitive computational complexity of the first approach.

Obtained point correspondences are then geometrically verified. Verification consists of estimating a transformation that maps a sufficient number of corresponding points between images, and the remaining point pairs are filtered out. Since corresponding point pairs are usually contaminated by outlier, estimation of the transformation requires techniques such as RANSAC. The result of this step is a geometrically verified set of image pairs and their associated inlier correspondences.

3.1.2 Incremental Reconstruction

The stage of incremental reconstruction receives the obtained set of image pairs with their point correspondences and performs iterative reconstruction of the scene. Initialization by carefully selected two-view reconstruction, typically from a location with many overlapping cameras, is followed by a cycle in which additional images are registered to the already reconstructed model and new points are triangulated. Furthermore, bundle adjustment is employed to improve the precision of the model.

Image registration phase involves selecting the next image to be added to the reconstruction. Image is registered to the current model by solving the Perspective-n-Point problem [33] using feature correspondences with already existing points in the model (2D-3D correspondences). This task consists of estimating the pose of the camera. As the correspondences often include outliers, methods based on RANSAC are usually utilized with minimal pose solvers.

A newly added image observes existing scene points in the model and can also increase the number of points in the model through triangulation (see Figure 3.3). Once a new scene point is observed from a different angle by at least one more image, its coordinates can be triangulated and the point extends the current model. Triangulation is the key step in SfM as it increases scene coverage and enables registration of more images by providing additional 2D-3D correspondences.

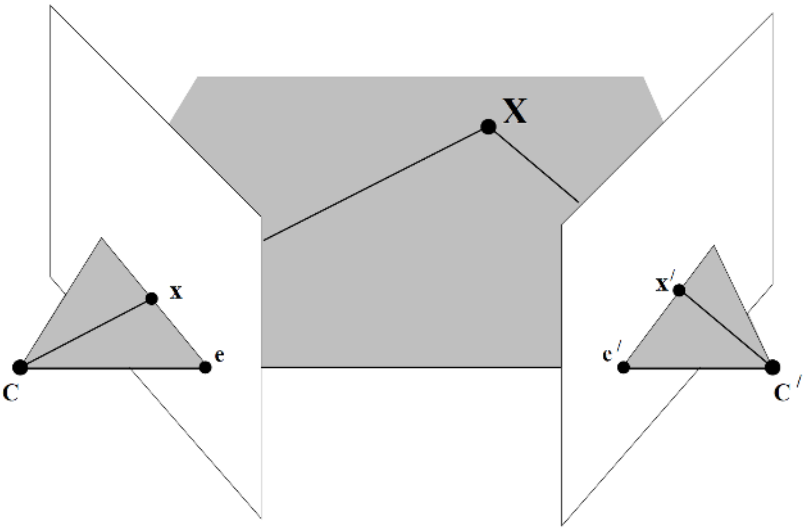

Figure 3.3: Triangulation of 3D point X from two corresponding image points x and x'. Camera centers are denoted by C and C', while e and e' denote the epipipoles (projection of camera center onto other camera's image plane). Adapted from [11].

Imprecisions in estimation of camera pose propagate to triangulated points and vice versa. Therefore, bundle adjustment is used for further refinement. This step is necessary to prevent reconstruction from drifting into a non-recoverable state due to the accumulation of uncertainties in pose estimations and errors in point coordinates. In bundle adjustment, already reconstructed points are projected back into image space of their respective images. The aim of bundle adjustment is then to perform non-linear minimization of the reprojection error, and thus simultaneously refine the camera and point parameters [22].

3.2 SLAM – Simultaneous Localization and Mapping

SLAM (Simultaneous Localization and Mapping) is a problem from robotics and computer vision, which aims to estimate the position of robot moving within unknown environment, while creating a map of the environment at the same time. SLAM tasks are usually expressed using a graph structure. The vertices (nodes) of the graph represent state variables, such as robot poses or locations of observed landmarks. Edges represent the observations between the nodes they connect, i.e. the measurements that form constraints on these nodes [14].

Knowledge about the environment encoded in the graph structure is used to define an error function. The goal of SLAM algorithm is then to refine the state variables (e.g. the positions of the robot and the locations of landmarks) based on the error function. In other words, SLAM can be understood as a graph optimization, which aims to estimate the values of state variables (nodes) that minimize the error determined by the constraints. Optimization of the error function is usually carried out in the non-linear least squares manner, using iterative Gauss-Newton or Levenberg-Marquardt methods [14].

An illustration of SLAM as a graph optimization task is shown in Figure 3.4, where x_i denotes vertices and e_{ij} denotes edges. In general, the structure can also form a hypergraph, since one edge may connect more than two vertices in some cases.

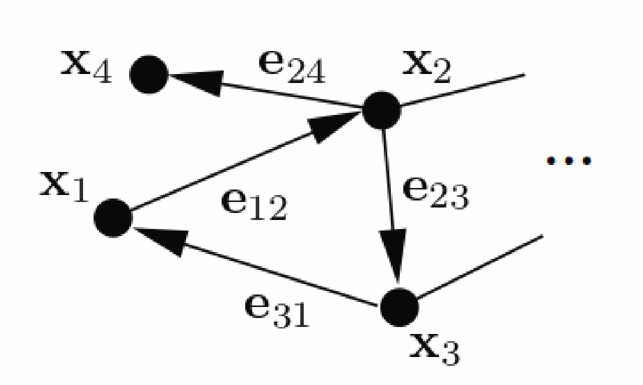

Figure 3.4: SLAM problem expressed as a graph, in which vertices represent state variables to be optimized and edges represent the observations that form constraints on the connected vertices. Adapted from [14].

When used with visual data, i.e. images obtained by a camera mounted on the robot, SLAM becomes related to the task of Structure from Motion in many aspects. Visual data is preprocessed by frontend algorithm to create the graph representation which is then refined by SLAM optimization. State variables (nodes) are the camera poses and the 3D positions of observed points in created point cloud representation of the environment. Graph edges represent which points are seen by which camera pose, and also contain the coordinates of the points in original images. The optimization then simultaneously refines the positions of the 3D points and the locations of camera by reprojecting the observed points and minimising the resultant projection error.

This approach to SLAM is often referred to as bundle adjustment, and its goal is, indeed, exactly the same as the one of the bundle adjustment used in Structure from Motion. Moreover, such point of view is beneficial, since more edges can be easily added to the SLAM optimization graph in order to form additional constraints. Therefore, the refinement process can be customized to fit the particular problem scenario. High level of possible customization is for example offered by SLAM++ library, which is a fast graph optimizer based on efficient usage of sparse block matrices [20], and which also supports GPU acceleration [21].

3.3 Optical Flow

Optical flow belongs to the set of algorithms used for motion estimation between two (or more) images. While other methods exist for simple movements, optical flow is the most general technique. The aim of optical flow is to compute an independent estimate of motion at each pixel [33].

In other words, the task of optical flow is to find a vector for every pixel that defines the displacement of the pixel between two images. A simplified (sparse) illustration of the concept is shown in Figure 3.5. For visualization purposes, color coding is often used (see Figure 3.6). An example of resultant optical flow estimation is presented in Figure 3.7.

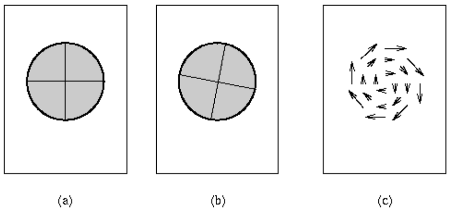

Figure 3.5: The aim of optical flow is to find motion vectors for pixels between two images. The movement between images (a) and (b) is represented by (sparse) vectors shown by (c). Adapted from [30].

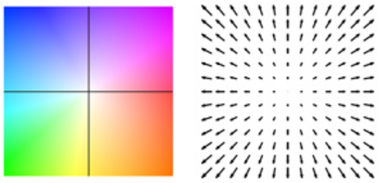

Figure 3.6: Color coding used for optical flow visualization. The central pixel represents zero movement. Every other pixel denotes a vector from the center, where the orientation and the magnitude are represented by hue and saturation. Obtained from [17].

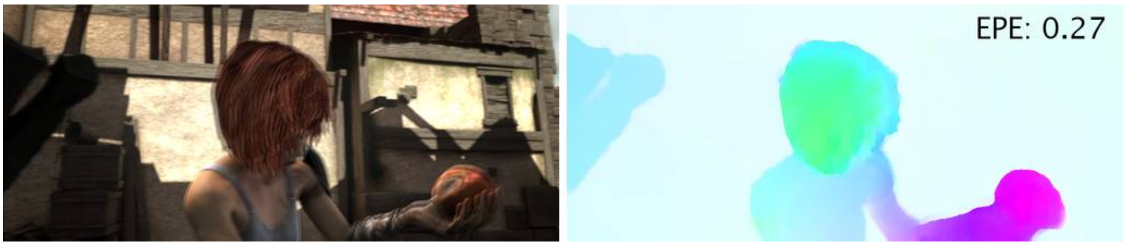

Figure 3.7: An example of optical flow prediction. Overlaid original image pair is shown on the left. The image on the right presents the resultant estimation of optical flow. Adapted from [10].

There are two classic approaches to estimating optical flow. The first is based on calculations using local window (patch-based approach). The second is variational approach using smoothness constraint [33]. Moreover, other methods and improvements have recently been presented.

In their work, Fischer, Dosovitskiy, Ilg et al. [10] approach the problem of optical flow calculation as a supervised learning task and they train a convolutional neural network to perform optical flow estimation. In order to obtain a sufficient amount of training data, they use synthetically generated datasets for training, and subsequently fine-tune the network on more realistic samples. Evaluation presented by the authors shows results that are competitive with other state-of-the-art methods for optical flow.

3.3.1 DeepFlow and DeepMatching

In order to address the problem of large displacements contained within the two input images, some authors incorporate descriptor matching component into the variational approach. The main idea is to guide optical flow estimation by providing correspondences from sparse descriptor matching. Weinzaepfel et al. [34] argue that even though this modification significantly improves the results of optical flow algorithm, standard methods for feature point extraction only produce points for salient image locations. Therefore, in their method for optical flow, named *DeepFlow*, Weinzaepfel et al. [34] enhance the variational approach with a custom descriptor matching algorithm called *DeepMatching*.

The proposed DeepMatching algorithm aims to retrieve quasi-dense point correspondences for later optical flow calculation phase. DeepMatching is strongly inspired by non-rigid 2D warping and deep convolutional networks. SIFT descriptors based on histogram of oriented gradients with 4 × 4 cells are used. However, instead of keeping the fixed 4 × 4 grid, it is divided into 4 quadrants and each of the quadrants is allowed to move independently in order to yield non-rigid matching. This approach is then applied recursively together with max-pooling and convolution [34]. As a result, DeepMatching produces point correspondences with very high density, as can be seen in Figure 3.8.

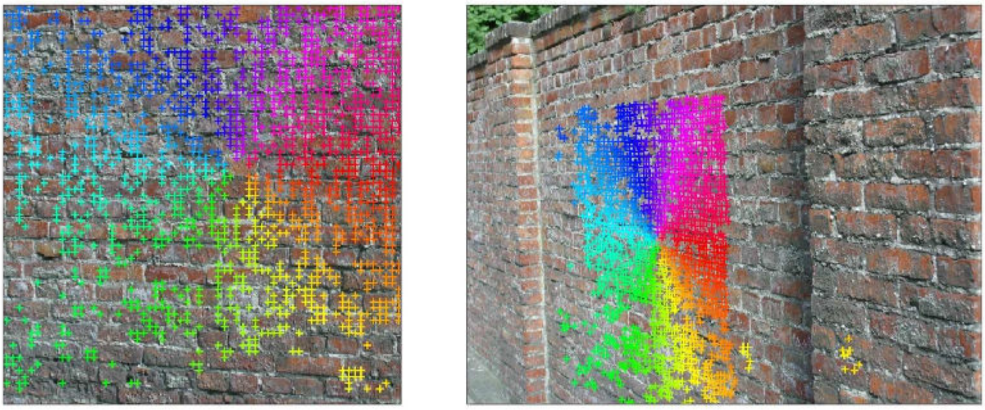

Figure 3.8: Example of quasi-dense point correspondences found by DeepMatching algorithm. Obtained from [34].

Chapter 4

Experiments with SIFT Features and Structure from Motion Tools

Throughout the first part of the work on this thesis, a series of preliminary experiments was carried out in order to evaluate to what extent the current state-of-the-art Structure from Motion algorithms can be used when solving the problem of 3D reconstruction of passing vehicles. For this purpose, two Structure from Motion tools were selected, COLMAP [22] and VisualSFM [35]. However, before examining the performance of SfM tools, one more set of experiments was carried out. Since both of the selected SfM tools base their correspondence search stage on SIFT features [18], experiments were first performed to evaluate the behaviour of SIFT feature extraction and matching on images of vehicles.

In this chapter, data obtained for experimenting are first described. Next, experiments with SIFT features are discussed. Subsequently, the results of Structure from Motion reconstructions are presented.

4.1 Obtained Test Data

Several image sequences of passing vehicles were obtained for experiments presented in this chapter. In order to ensure sufficient quality and resolution, images were captured using stationary reflex camera (Nikon D3200 with Nikon AF-S DX 18-105mm f/3,5-5,6 G ED lens) used in burst mode. Therefore, images in each sequence represent frames that would be extracted from a video at different points in time. Each created sequence contains from 7 to 15 images. For the purpose of experiments, a sample containing sequences of 6 different cars, 2 vans, and 1 truck was selected. Additionally, all images in the selected sequences were cropped to include the vehicle with only a small border containing the background. Examples from two image sequences are shown in Figure 4.1.

Considering the fact that Structure from Motion algorithms expect a static scene and moving photographer, another set of image sequences was obtained using a stationary car with camera moving around. It is therefore possible to compare the results of inputs containing stationary and moving vehicles.

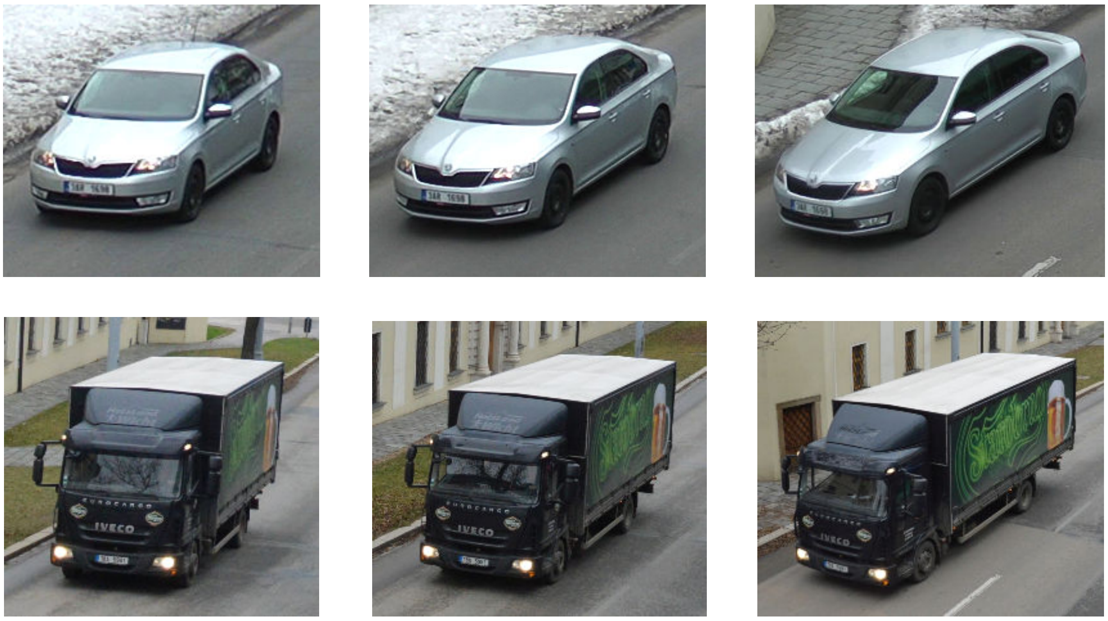

Figure 4.1: Examples of obtained sequences of images with a passing vehicle.

4.2 Experiments with SIFT Feature Extraction and Matching

Characteristics of extracted SIFT keypoints and correspondences were examined on obtained image sequences using SIFT implementation in OpenCV[1] library. First, positions of detected SIFT keypoints were inspected on single images. Secondly, found feature correspondences between pairs of images in each sequence were studied. In this case, various image pairs with different steps between images (i.e. different distances of the images within the sequence) were considered. All experiments were carried out on sequences of both stationary and passing vehicles, with equivalent results.

When SIFT keypoint detection algorithm is applied, vast majority of obtained keypoints is located on the front part of the vehicle (mainly on the grilles and the license plate). The remaining parts of the vehicle are covered very sparsely, as only low numbers of feature points are detected there. Furthermore, when feature point matching is performed, correct correspondences are often found only for small steps between the images in the particular sequence (i.e. small changes in vehicle orientation). Larger steps between images result into significant numbers of incorrectly calculated correspondences, especially for points which are not on the front part of the vehicle (grilles and license plate). An example of computed SIFT correspondences is shown in Figure 4.2.

The results of experiments with SIFT features indicate that algorithms for 3D reconstruction that rely on SIFT in their correspondence search stage are very likely to have only small numbers of feature points for subsequent reconstruction phase. Moreover, the number will probably be further reduced by incorrectly found correspondences.

[1] http://opencv.org/

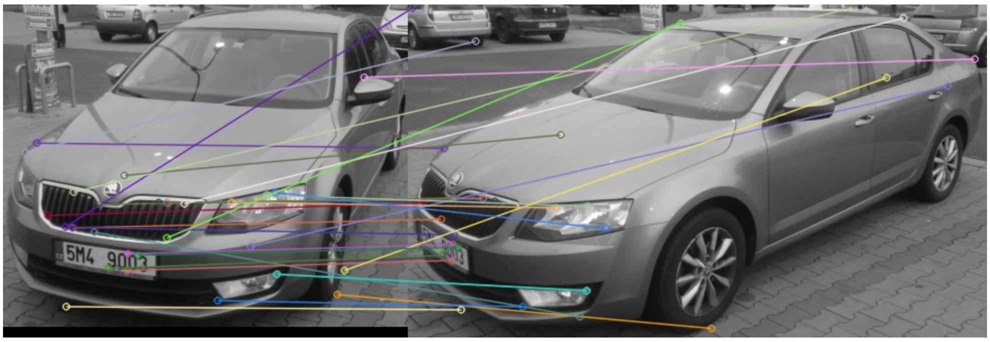

Figure 4.2: Example of found SIFT point correspondences on a static vehicle (30 best matches are shown). Vast majority of feature points is detected on the front part of the vehicle. Moreover, a significant number of incorrect matches can be observed.

4.3 Experiments with Structure from Motion

Experiments with 3D reconstruction were performed using COLMAP tool, which was released in 2016 and is currently the state-of-the-art Structure from Motion implementation [22]. Reconstruction process was tested for all created image sequences of both stationary and passing vehicles.

First, experiments with image sequences of the stationary vehicle were performed. Out of 11 experiments, reconstruction was successfully completed only in six cases. In the remaining cases, the SfM algorithm failed to produce any result at all, reporting that no good initial image pair was found. Only three of the successful reconstructions contained recognizable points that belong to the original vehicle. The best achieved result is presented in Figure 4.3. One of the remaining successful reconstructions shows an attempt of the algorithm to reconstruct the background scene instead of the vehicle, while other two successful reconstructions resulted in a point cloud with no meaningful structure.

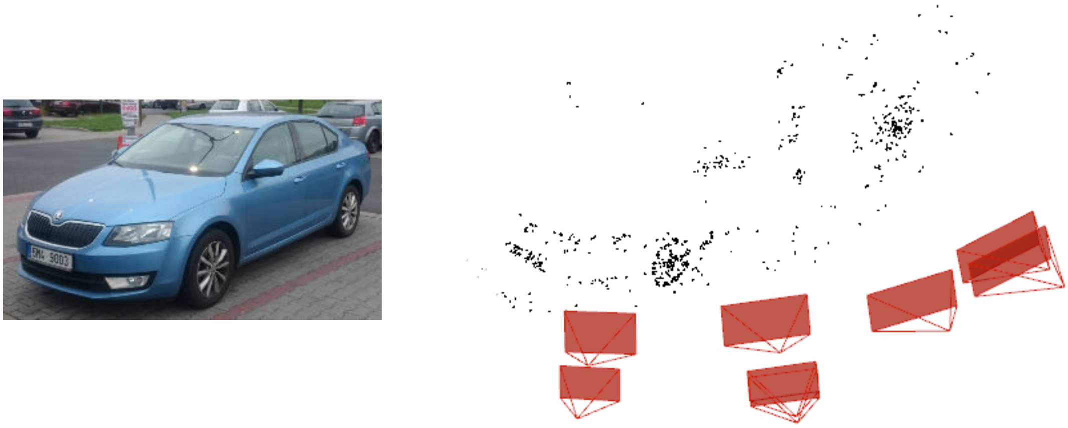

Figure 4.3: The best obtained result using COLMAP Structure from Motion tool for a sequence of images containing a *stationary* vehicle. Point cloud model (on the right) includes partially recognizable front part of the vehicle (especially its license plate) and the front wheel. Remaining parts of the vehicle are not included at all, or reconstructed incorrectly.

Next, COLMAP was used on the image sequences of passing vehicles. Out of six image sequences of cars, only one reconstruction was successfully completed and a point cloud model was produced, whereas all other reconstructions failed (again, the algorithm reported that no good initial image pair was found). As expected, only the front part of the car is partially recognizable in the successfully created model. Reconstruction process also failed in case of image sequences of both vans. Nevertheless, a successful reconstruction was obtained for image sequence of passing truck, where a significant portion of the front part is recognizable. The resultant model is shown in Figure 4.4.

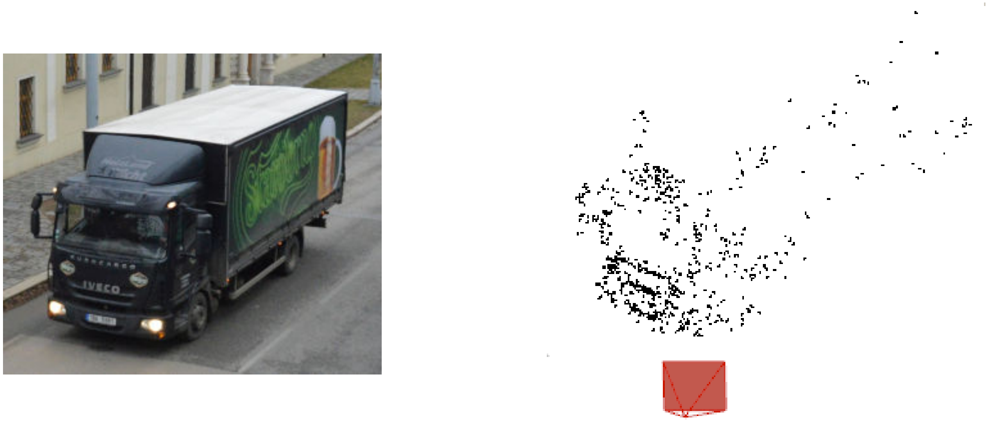

Figure 4.4: The best obtained result using COLMAP Structure from Motion tool for image sequence of *passing* vehicle. Resultant point cloud (on the right) contains recognizable front part of the truck.

The results of the Structure from Motion algorithm confirm the conclusions drawn from the previous experiments with SIFT features. As expected, reconstructed models are often severely incomplete. In a vast majority of cases, the reconstruction process either failed entirely, or the resultant point cloud contained no meaningful structure. Apart from the presented tests using COLMAP, several experiments were also carried out with VisualSFM tool, producing comparable results.

Chapter 5

Improvement of the 3D Reconstruction Process

Based on the experiments described in the previous chapter, two main aspects hindering the 3D reconstruction process can be identified. The first problem is insufficient number of feature correspondences, as standard SIFT features are not a suitable input for reconstruction of passing vehicles. The second significant problem is represented by points and point correspondences located in the image background. In this chapter, changes to the reconstruction process are proposed and applied in order to improve the overall quality of the resultant 3D model.

5.1 Substitution of SIFT Features

In order to increase the number of point correspondences located on vehicle, it is necessary to substitute SIFT features with a different method for keypoint extraction and matching. In particular, a method producing matches with higher density is desirable. One option would be to use the output of an algorithm for optical flow calculation, which would produce a vector that estimates movement of each pixel in an image pair. Nevertheless, in order to address large displacements, optical flow methods often utilize feature matching algorithms, too. It is therefore more beneficial to inspect the feature matching approaches used within optical flow, rather than entire methods for optical flow themselves.

As described in Section 3.3, optical flow algorithm DeepFlow employs a custom feature matching procedure called DeepMatching to calculate quasi-dense point correspondences before smoothing them using variational approach to obtain optical flow estimation. The power of DeepMatching algorithm, even though originally designed for optical flow, could be harnessed to provide a high number of point matches for subsequent 3D reconstruction of passing vehicles. An illustration of point matches found by DeepMatching algorithm is shown in Figure 5.2.

5.2 Filtering of Obtained Correspondences

The second necessary modification of the correspondence extraction procedure is removal of those point matches that belong to the scene background, as these points can be considered outliers, and thus negatively affect the reconstruction process. Obtained correspondences should therefore be filtered using a foreground mask of every individual image, so that only matches located on the vehicles in both images of a particular image pair are taken as an input for reconstruction phase. An example of the original image and its respective foreground mask is shown in Figure 5.1, filtered correspondences are illustrated by Figure 5.2.

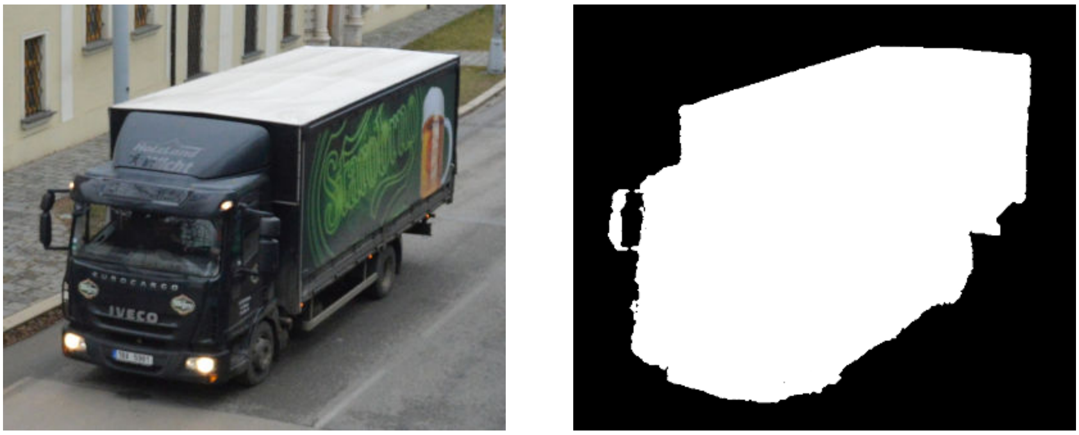

Figure 5.1: Original image of a passing truck and its foreground mask.

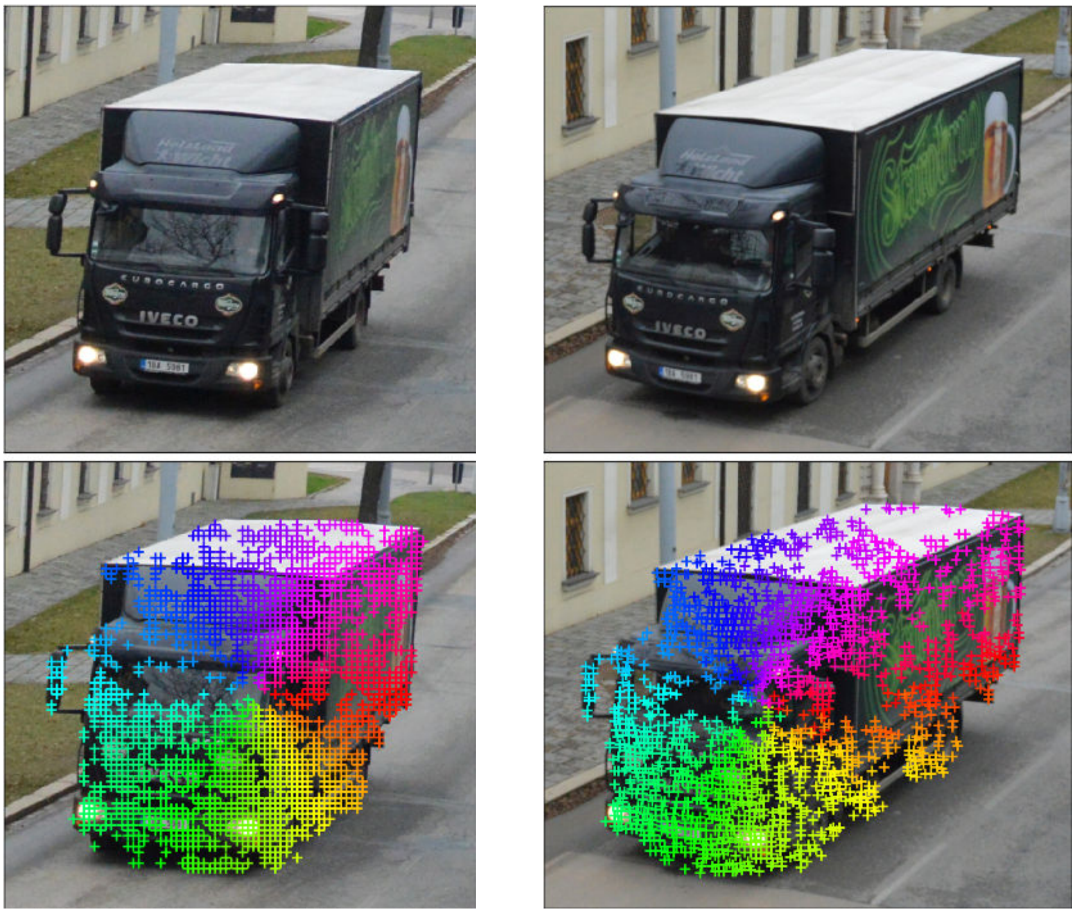

Figure 5.2: Correspondences for two images of passing truck calculated using the DeepMatching algorithm and filtered with foreground masks.

5.3 Implementation of the Proposed Modifications

Implementation of the modifications proposed in Sections 5.1 and 5.2 requires a possibility of defining custom keypoint locations and point correspondences as an input to the following stage of incremental reconstruction. A suitable interface is offered by VisualSFM and application of presented modifications was therefore realized using the VisualSFM tool. The original correspondence search stage provided by VisualSFM is substituted by modified correspondence search, as shown in Figure 5.3.

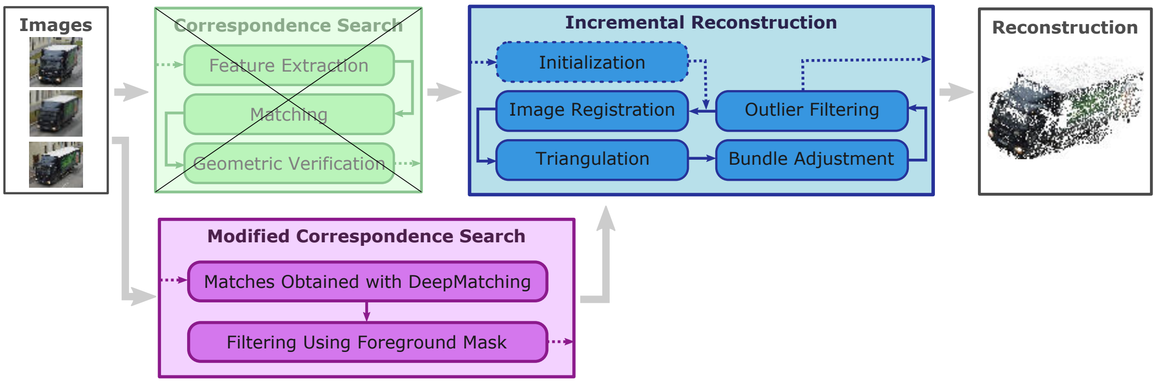

Figure 5.3: Standard pipeline of incremental Structure from Motion algorithm (top) and modified correspondence search stage (bottom), which employs DeepMatching to obtain quasi-dense point correspondences and applies filtering with foreground masks to prepare input for the subsequent reconstruction phase.

Correspondences are first calculated using the DeepMatching algorithm for all possible pairs of images in an image sequence. Next, foreground masks are created and applied to perform filtering of point matches. A file with locations of matched points is then generated for every image. It should be noted that unlike standard SIFT keypoint detection, DeepMatching can obtain slightly different sets of points for one particular image when matching this image with several other images. Therefore, union of the obtained point sets is performed before the output file with keypoint coordinates is created. Furthermore, one file containing information about all found matches is generated. The described procedure replaces the first stage of the SfM pipeline, in which correspondence search is performed.

Information stored in the generated files was then used as the starting point for the 3D reconstruction stage of VisualSFM tool. Examples of successfully created resultant models can be seen in Figure 5.4. When compared to the reconstruction obtained with original SfM algorithm, the results of the proposed modifications significantly improve completeness of the resultant point cloud model.

Nevertheless, even with the applied modifications, it is still not guaranteed that a meaningful 3D reconstruction will be produced. In particular, the reconstruction process has shown to be sensitive not only to the initial image pair selection, but also to wrong estimation of camera pose during the incremental reconstruction stage. It is therefore necessary to study the characteristics of the outlined reconstruction problem and determine further constraints that could be applied to achieve additional improvements.

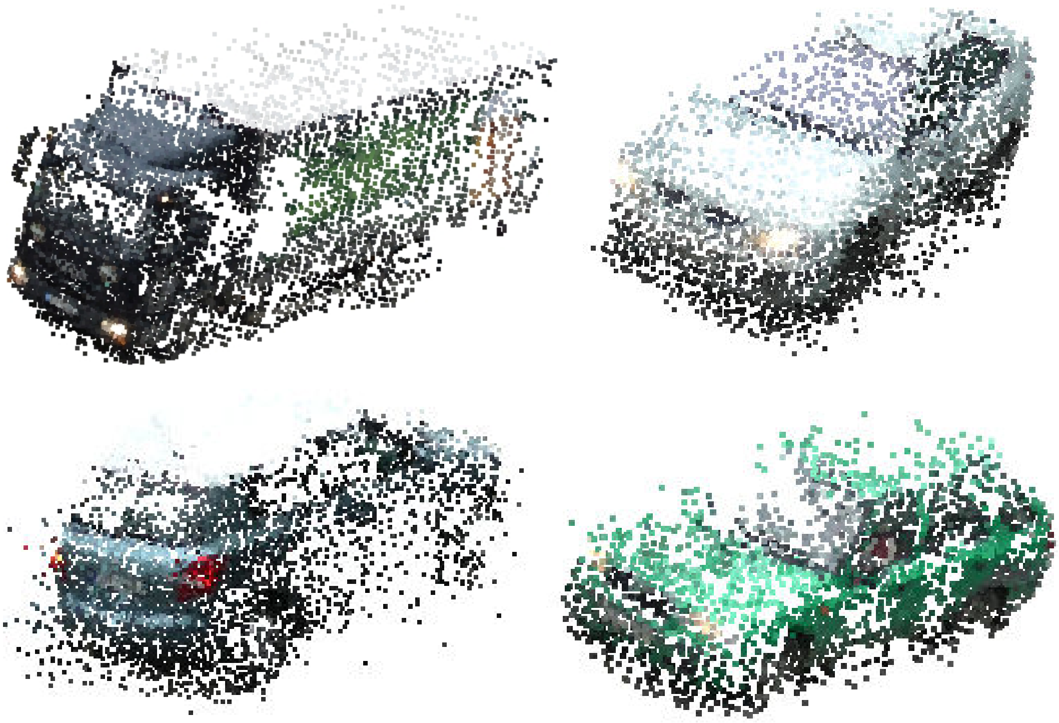

Figure 5.4: Resultant 3D reconstructions obtained when proposed modifications to Structure from Motion pipeline are applied.

5.4 Examination of Additional Constraints

Commonly, the input to a Structure from Motion algorithm is a set of images of a static scene. Cameras are expected to move freely and therefore their position and rotation can be arbitrary. In the most general cases, such as reconstruction from photos taken by tourists, each camera is considered to be independent and has its own intrinsic calibration parameters. Nevertheless, reconstruction of passing vehicles exhibits several specific characteristics that can be used to constrain the level of freedom the reconstruction algorithm is given, and thus prevent wrong estimations and assumptions during the reconstruction process.

The first and most straightforward constraint arises from the fact that all images in a vehicle sequence are taken by the same camera. Therefore, the intrinsic parameters are shared by all estimated cameras in 3D reconstruction. This constraint has already been applied to obtain the results presented in the previous section.

More significant constraints emerge when the characteristics of the observed motion is considered. Unlike the traditional use of Structure from Motion to obtain a reconstruction of a static object by moving camera, the vehicle in our scene is moving while the camera remains fixed. Nevertheless, by applying the foreground masks to obtain only the feature points belonging to the vehicle, we can understand the task as a standard Structure from Motion scenario. In other words, we can consider the vehicle to be a stationary object and assume the camera to be moving while capturing the images. This approach enables us to derive two additional constraints that can be applied in the vehicle reconstruction task.

First, it can be assumed that the camera is observing a straight part of a road and therefore the motion of a vehicle follows a line. Once the vehicle is considered to be stationary, the trajectory of camera movement has to represent the original straight movement of the vehicle. More precisely, estimated positions of camera in 3D space have to belong to a line. This assumption also requires that the vehicle is not changing lanes in case there are multiple road lanes observed (in practical usage, such cases could be detected and filtered out).

Second, the original camera is stationary and the vehicle itself does not perform any kind of rotation. Therefore, the overall rotation of camera during its virtual movement in space should remain constant, i.e. there should be no relative rotation between consecutive camera poses.

Both of the above mentioned assumptions are illustrated by Figure 5.5. Moreover, if the speed of the vehicle stays constant, or its change is only small or gradual, the distance between the camera centers should be approximately the same for three successive camera poses. This could also be used as another constraint.

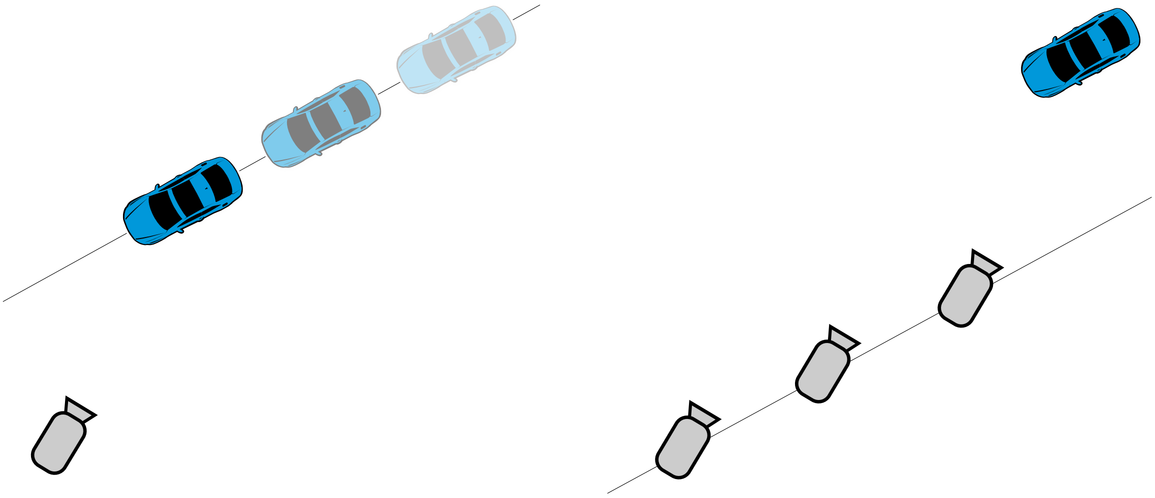

Figure 5.5: Illustration of additional constraints applicable to the outlined task of 3D reconstruction of a passing vehicle. The assumption that the original motion of the vehicle is straight and that the position of the roadside camera is fixed is shown on the left. When the vehicle is considered to be stationary by the Structure from Motion algorithm, it is the camera that should follow the original liner movement, as shown on the right. Furthermore, the rotation of the camera should remain the same for all estimated camera poses.[1]

Close inspections of obtained reconstructions revealed that the aforementioned constraints are often not fulfilled. As a consequence, produced point clouds contain significant numbers of points with incorrect positions. Even though some of these points may originate from outliers generated by the correspondence search stage, most of the incorrectly positioned points are caused by wrong localization of one or more camera poses. This situation is demonstrated by an example in Figure 5.6. Additionally, when the camera rotations are wrongly estimated, the entire model structure degenerates into a rather flat surface, instead of preserving the right angle between the front of the vehicle and its side. This effect is shown by Figure 5.7.

[1] Free vector graphics of the car used in Figure 5.5 was obtained from Vecteezy.com

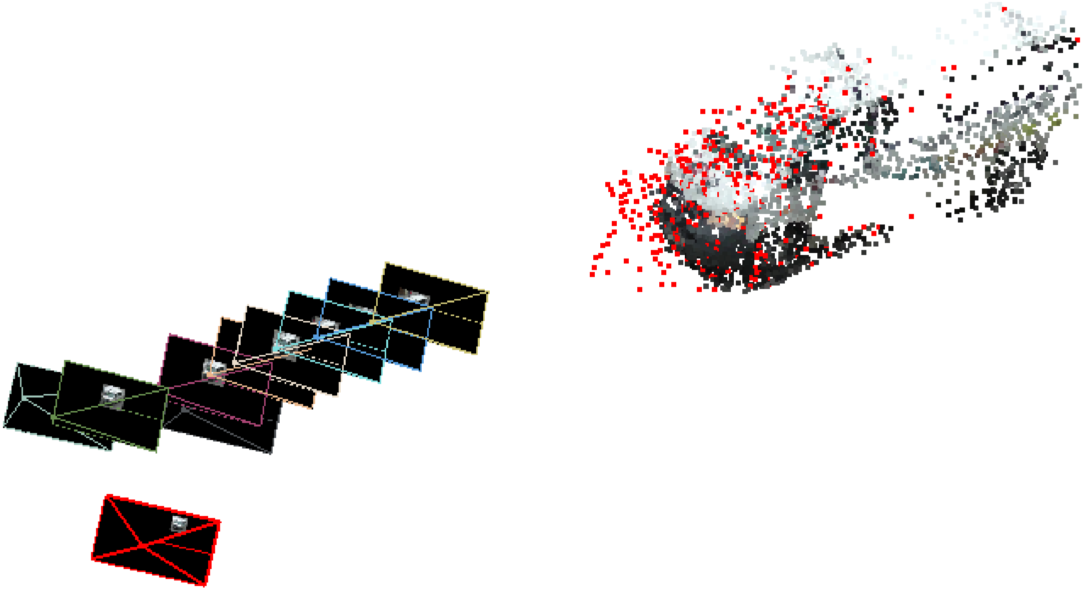

Figure 5.6: Points affected by wrongly estimated camera position, which is not aligned with other camera poses. The points and the camera are marked in red.

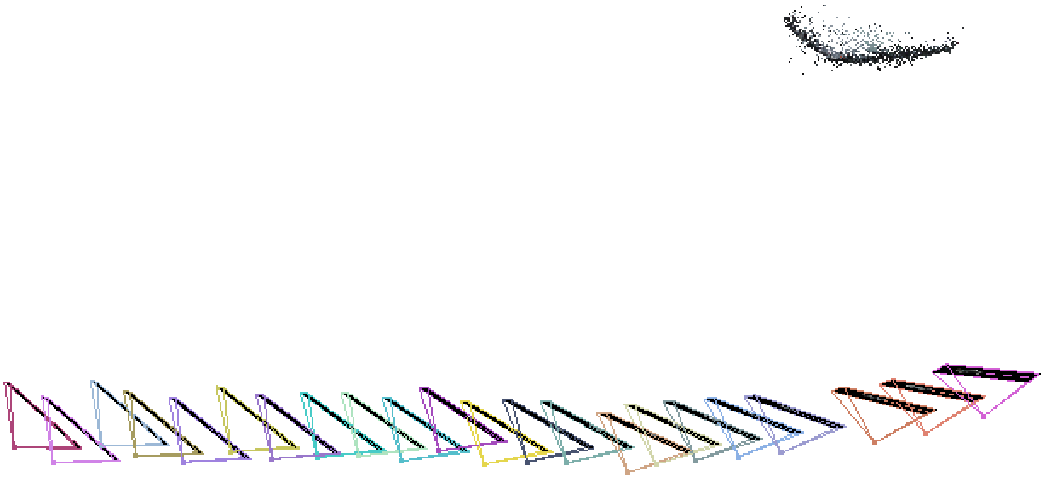

Figure 5.7: Top view of a reconstructed model showing wrongly estimated camera rotations that result into almost flat shape of the model, rather than producing points on the vehicle that would form a right angle.

5.5 Application of Motion Constraints

Violation of the identified constraints has shown to severely affect the quality of reconstructed point clouds. Further refinement is therefore necessary in order to improve precision (in cases where a meaningful 3D model was created). For the purpose of imposing the previously described constraints, customizable bundle adjustment module of SLAM++ library[1] is used.

Reconstruction obtained with modified VisualSFM pipeline is first converted into graph representation, which is suitable for processing by SLAM++, using a converter distributed together with the library. As described in Section 3.2, nodes of the graph represent camera poses and positions of 3D points, while graph edges contain the relations between the points and cameras. In order to apply the motion constraints identified in the previous section, additional edges are added to the graph when being loaded by SLAM++.

Since the constraints apply to camera movement, additional graph edges need to connect the vertices that represent camera poses.[2] More precisely, the three identified constraints described in the previous section are implemented in the following way:

- Connected camera poses are forced to gradually become collinear.
- Camera rotation is pressed towards its average value to ensure zero relative rotation.
- The distances between camera centers are pushed to become equal.

The first part implements the requirement for the camera path to represent the originally straight vehicle trajectory. The second part is related to the fact that no rotational movement takes place, and thus no rotation should occur between camera poses. It should also be noted that the third constraint assumes minimal or no change in vehicle's speed.

Nevertheless, the described constraints are not implemented in the way that one single edge would connect all camera nodes. Instead, one graph edge connects three consecutive camera poses. These three cameras are then forced to fulfil the constraints. Such edges are then added in a sliding-window manner for every three consecutive camera nodes. The constraints (collinearity, etc.) therefore propagate among the entire camera sequence. Apart from being easy to apply, another advantage of this approach is that the initial reconstruction can contain an arbitrary number of camera poses. The only requirement for this principle to work is sorted camera sequence.

The described constraints are all applied together in every iteration of the SLAM optimizer. Edges representing the newly added constraints become part of the error function, and thus each step of SLAM++ performs the corresponding refinement together with the standard bundle adjustment task. In other words, the minimization of error expressed by the new constraints, for example the distance of camera centers from a straight line, is carried out jointly with the refinement of reprojection error, which is given by projecting the 3D points back to the original image coordinates. As a result of the optimization, camera poses should align to respect the original movement of the observed vehicle and their rotation should converge to become approximately the same. Consequently, this should improve the precision of the points in the reconstructed model. The camera alignment effect is demonstrated in Figure 5.8, showing the initial and final state of the refinement process. Figure 5.9 illustrates the improvement of the corresponding point cloud reconstruction.

[1] https://sourceforge.net/projects/slam-plus-plus/

[2] An example code with additional constraints was kindly provided by Lukáš Polok, author of the SLAM++ library.

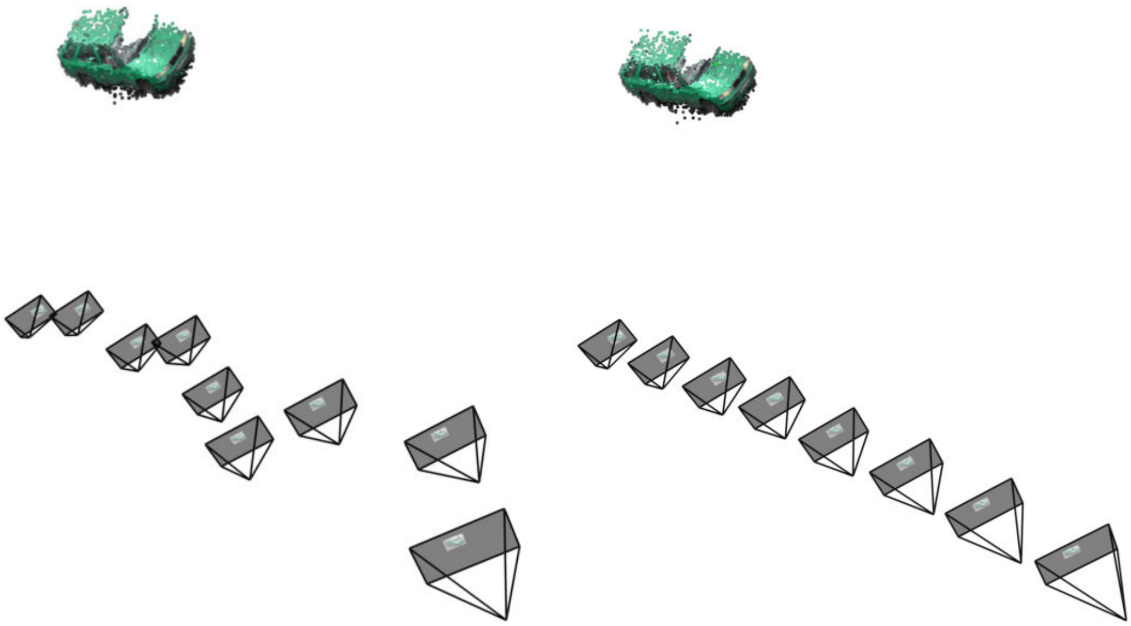

Figure 5.8: Inaccurately estimated camera poses (left), and the resultant alignment (right) achieved by refining the reconstruction with SLAM++ using additional constraints applied to camera motion.

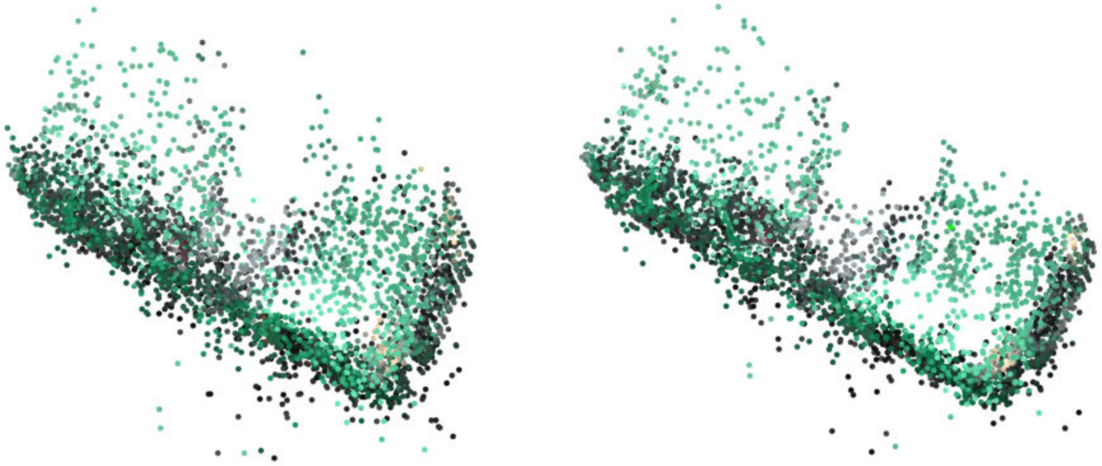

Figure 5.9: Top view of initial point cloud model (left) and improved reconstruction obtained after refinement with SLAM++ (right). Especially noticeable is the rectification of the right angle between the front part and the side of the vehicle.

Since the optimization itself is done in the least-squares manner, it is important to point out that not all constraints have to be entirely fulfilled. Moreover, the refinement process aims to minimize the error in the sense of local optimization, and therefore a satisfactory result is not always guaranteed. The outlined rectification of the point cloud by additional constraints is thus applicable only in cases where camera poses are already estimated to be moderately close to the aligned configuration. Imposing the motion constraints on a reconstruction with camera positions being too far from the expected straight trajectory, or with significantly incorrect rotations, fails to improve the vehicle model, or even degrades its quality. This observation motivates several proposals for future work, which are covered in full detail in Chapter 7.

Before the reconstruction results are evaluated, it should be reminded that only one monocular camera was used for acquiring the vehicle sequences. Therefore, only the front part and one side of the vehicle can be seen. As a result, only these parts of the vehicle are reconstructed in the point cloud, which can be noticed in the presented figures, e.g. Figure 5.9. Nevertheless, the information contained within the point cloud is entirely sufficient for the intended task of traffic camera calibration, as the other side of the vehicle would not contribute any additional information that could be utilized for calibration purposes.

Chapter 6

Evaluation

In this chapter, evaluation of the proposed approach to 3D reconstruction of passing vehicles is performed. The evaluation primarily focuses on distance measurement in the observed scene in order to assess the ability of the method to be utilized for real-world measurements. Subsequently a discussion of general limitations of the method is provided.

6.1 Distance Measurement

Since the prospective usage of the reconstructed model is determining the scale of the road plane in the observed scene, a distance measurement was selected as the most suitable method of evaluation. Therefore, it was necessary to make a real-world measurement of a distinctive object within the road plane. For this purpose, a clearly visible road marking was selected, as shown in Figure 6.1.

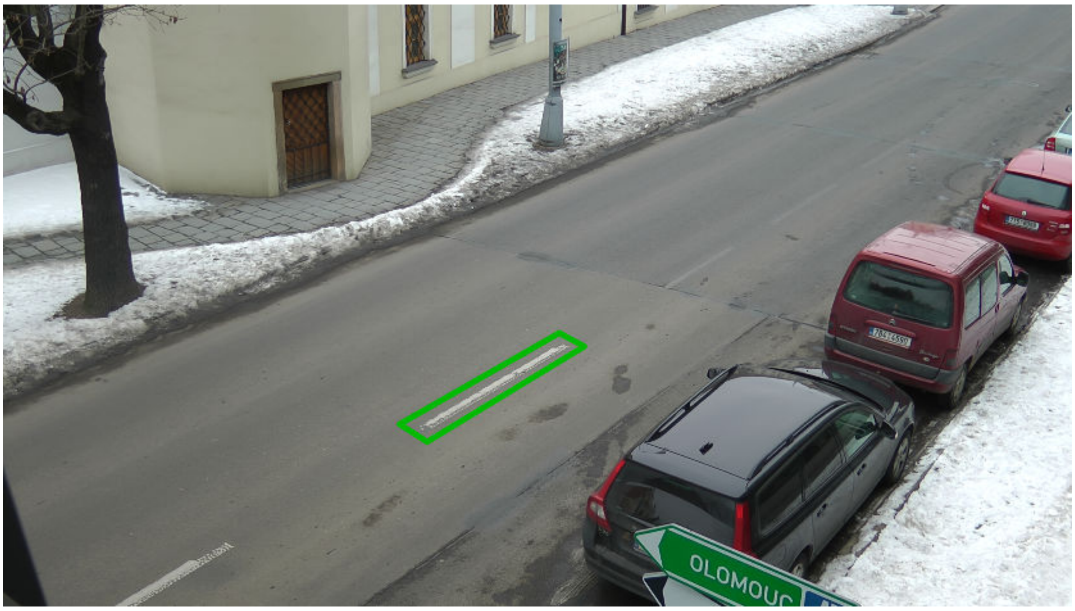

Figure 6.1: Road marking used for measurement evaluation. The ground truth length of the highlighted line is 2.85 meters.

The measurement procedure is performed in the following manner. When reconstruction of a vehicle is completed and refined, it is first necessary to determine the 3D position of the road plane. For this purpose, three points reconstructed at the base of the vehicle are selected. These points uniquely determine a plane in the virtual 3D space of the reconstruction, which represents the real-world road plane. 2D coordinates of the end points of the road marking with known length are then obtained from the original image of the scene. Afterwards, camera poses are retrieved from the 3D reconstruction, including the respective points that define image plane positions in the 3D space. These points are subsequently used to compute the image plane locations for the 2D coordinates of the road marking ends. Once the 3D locations of the road marking ends are available, a ray for each point can be cast from the camera into the scene. More precisely, the ray is cast from camera's center of projection through the 3D location of the corresponding image plane position of the particular road marking end. Two rays are therefore cast, one for each road marking end. Intersections of these rays with the road plane are then calculated. The distance between the two intersection points represents the relative length of the road marking in the virtual scene, as illustrated by Figure 6.2.

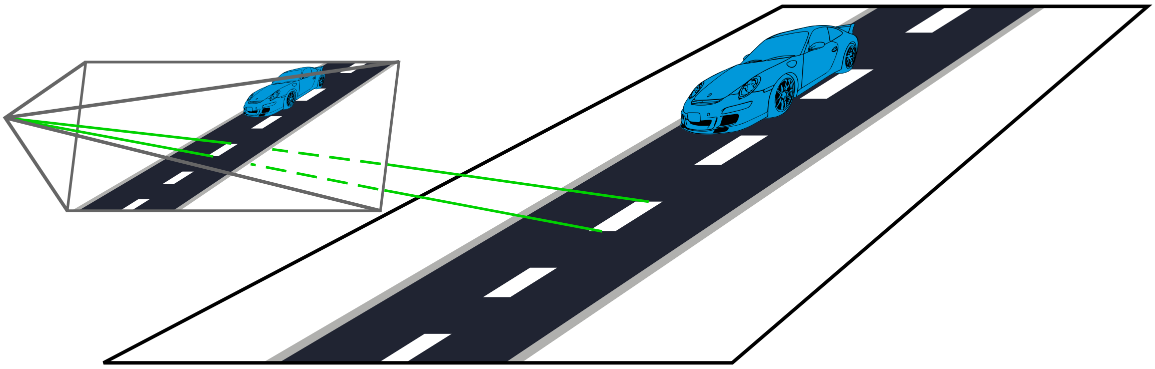

Figure 6.2: Illustration of the distance measurement method used for evaluation of vehicle reconstructions. Rays are cast the from camera center through road marking ends in the image plane. Intersections of these rays with the road plane determine the position of the line in the relative world of the 3D reconstruction. Scale of the road plane is then used to obtain the real-world length of the measured line.[1]

To compute the actual size of the road marking, it is also necessary to determine the scale of the road plane, i.e. the relation of the virtual and real-world units. At this point, the reconstructed model is utilized again. Two points with known dimensions are selected from the point cloud, and their distance is used to obtain the scale. For some models, these points were at the front and the rear part of the vehicle. Nevertheless, in more cases, the wheelbase (the distance between the centers of the front and the rear wheels) was used instead. This was necessary since the rear part of the vehicle was missing in some reconstructions due to imprecise foreground masks. Another reason for using wheelbase is the fact that the exact length of the entire vehicle can sometimes differ depending on the production year of the particular vehicle model due to facelift changes. The wheelbase, however, remains the same. Once the scene scale is obtained, the virtual length of the road marking can easily be converted into measurement in real units of size.

[1] Free vector graphics used in Figure 6.2 was obtained from Vecteezy.com

Table 6.1: Distance measurements of a road line computed using reconstructed vehicle models. Each measurement was calculated as an average value measured by first three camera poses in order to minimize the effect of possible slight inaccuracies in camera pose estimation. Ground truth length of the road line is 285 cm.

Vehicle	**Measurement [cm]**	**Absolute error [cm]**	**Relative error**
1	293.1	8.1	2.8%
2	261.8	23.2	8.1%
3	289.0	4.0	1.4%
4	310.0	25.0	8.8%
5	269.5	15.5	5.4%
6	273.7	11.3	4.0%
7	322.7	37.7	13.2%
8	297.3	12.3	4.3%
9	256.4	28.6	10.0%
Mean	285.9	18.4	6.5%

Measurement of the line shown in Figure 6.1 was performed using vehicle reconstructions obtained from three video sequences (recorded with Full HD camera) using the approach described in the previous chapter. Only reconstructions of vehicles going towards the camera were selected. Moreover, unsuccessful reconstructions which resulted into a meaningless point clouds were excluded. Subsequently, it was necessary to classify the vehicles. Precise information about vehicle make and model was determined by manual examination of vehicle images acquired from the original video. Furthermore, corresponding vehicle dimensions were retrieved, when available. Altogether, nine successful vehicle reconstructions with known real-world dimensions were obtained. Each of these vehicles was then used to calculate an estimate of the road marking length. The measurement results are presented in Table 6.1 together with respective absolute and relative errors. Absolute error is calculated as the absolute value of the difference between the measurement and the ground truth value. Relative error is calculated using Equation (6.1), where the *gt* represents the ground truth measurement and *lenght* represents the calculated estimate of the road marking length.

$$e_{relative} = \frac{|length - gt|}{gt} \tag{6.1}$$

Each measurement in Table 6.1 was calculated as the average value of lengths obtained from the first three camera poses within the particular reconstruction, i.e. the poses closest to the point cloud model. This approach was chosen in order to minimize the possible effect of minor inaccuracies in estimated camera positions and rotations. The presented data shows that the line measurements are above, as well as below the ground truth value. The best achieved error is only 4 cm (1.4%), while the worst length estimate is 37.7 cm (13.2%) away from the real line size. Moreover, mean value of all measurements was computed. It can be seen that aggregation of all estimates in the form of mean value yields a result that is very close to the ground truth distance: 285.9 cm (calculated) vs. 285 cm (ground truth). The mean of the absolute error is 18.4 cm, which represents 6.5% of the ground truth length.

It is also desirable to examine the histogram of all measurements included in the calculation of the resultant mean value. This histogram is shown in Figure 6.3 and contains the original measurements obtained for each of the first three camera poses in all reconstructions used for the evaluation. It can be observed that estimated lengths do not form a clear peak near the ground truth value, but are rather scattered around it. Nevertheless, the data can be expected to create a more distinctive peak in case of larger dataset sizes.

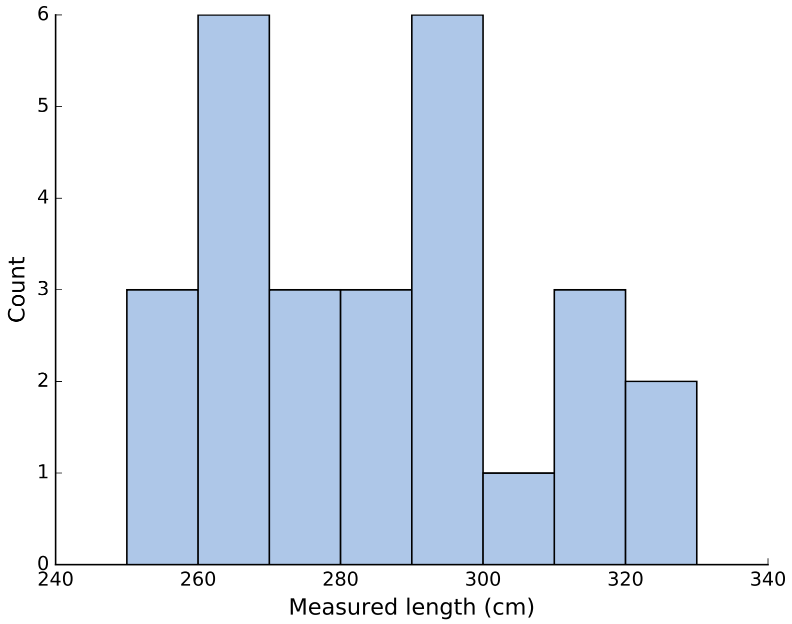

Figure 6.3: Histogram of estimates of the line length obtained from the first three camera poses in each reconstruction.

As described above, the measurement calculation has been obtained by averaging the values obtained for the first three camera poses, which are the closest to the reconstructed point cloud model. Since there are generally more camera poses in every reconstruction, it is also interesting to inspect how the road marking length estimation changes with the camera pose being further away from the point cloud model. The graph with relative change of the measurement from the average value computed from the first three poses is shown in Figure 6.4.

While measurements from some reconstructions show only small relative change in calculated values and remain rather stable even with rising distance of the camera pose from the point cloud, other reconstructions exhibit a more steeper rise of the difference from the road marking length obtained using the first three camera poses. The rise in the measurement change is most likely caused by imprecise road plane position. The road plane in the reconstruction space may be slightly slanted, which will affect the measurements in the more distant camera poses more significantly than the near ones. Another possible cause which can contribute to gradual rise of relative difference is less accurate estimation of more distant poses, since a smaller number of feature points is available in images with lower resolution of the vehicle (when the vehicle is further away from the camera). Imprecise pose estimations also explain the fluctuations in observed trends.

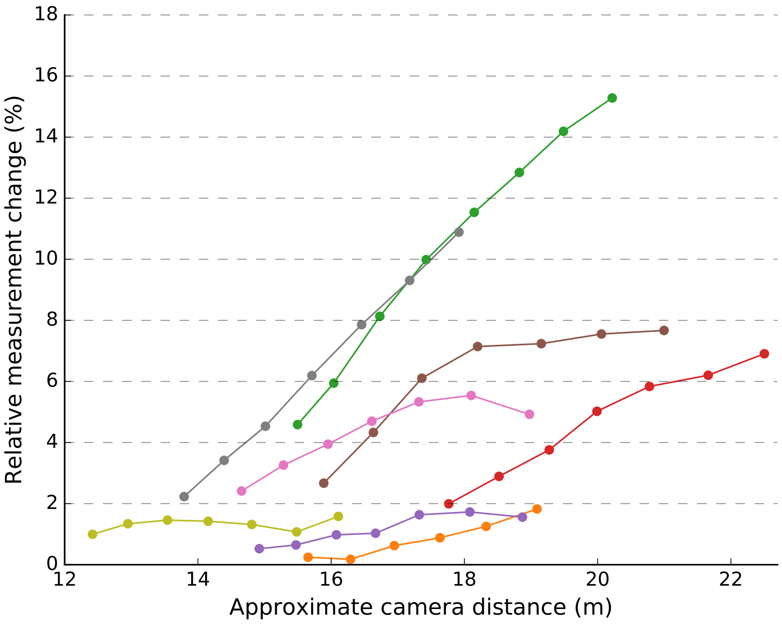

Figure 6.4: Relative difference of the road marking length measured by more distant camera poses from the average value obtained using the first three camera poses. Each colour represents poses from one reconstruction. The graph only shows the more distant camera poses, i.e. the first three poses used for calculation of the average value are not included. For each camera pose, its approximate distance from the reconstructed model is computed. It should also be noted that pose distances can vary between the reconstructions, as image sequences capture vehicles at different positions on the road, and thus at different distances from the camera.

When interpreting the graph in Figure 6.4, it is necessary to consider the original scenario of moving vehicles and static position of the camera. In general, the graph indicates that the measurement precision is likely to gradually decrease with increasing distance of the measured object from the static camera location.

6.2 Comparison with Existing Approaches

To evaluate the performance of the proposed method, obtained results are compared with the results of other existing methods for scene scale inference. As described in Section 2.3, two important approaches exist. The first method presented by Dubská et al. [9] relies on 3D bounding boxes constructed around 2D vehicle blobs and it determines the scene scale using statistical domain adaptation (see Subsection 2.3.1 for details). The second approach proposed by Sochor et al. [27] infers the scene scale by aligning 2D bounding boxes of rendered 3D models with the 2D bounding boxes of observed vehicles. The method utilizes two obtained 3D models with known dimensions (see Subsection 2.3.2). Both of these approaches are thoroughly evaluated by Sochor et al. [27], including the evaluation of distance measurements. This enables comparison with the method explored in this thesis. However, it should be noted that the evaluation of the existing approaches was obtained on a different dataset, and therefore the presented comparison conclusions should be interpreted with caution.

As shown in Table 6.1, the mean relative distance measurement error of the proposed approach based on 3D vehicle reconstruction is 6.5%. The error is therefore lower than the 9.62% mean relative measurement error reported for the first of the existing methods. This suggests that the scene scale inference using reconstructed 3D models should provide superior results than the method based on bounding boxes constructed around 2D vehicle blobs.

On the other hand, the relative error reported for the second existing method is only 2.33%. It can therefore be stated that the presented method based on 3D reconstruction is currently not capable of outperforming this existing approach. Further improvements of the 3D reconstruction process are thus necessary for this method to achieve better results. To this end, specific suggestions are outlined in Chapter 7.

It should also be pointed out that the presented method and the second existing approach both share a similar idea of using the real 3D information about passing vehicles. Nevertheless, the existing approach relies on previously obtained 3D models that are rendered and aligned with observed vehicles. This gives the method presented in this thesis an important advantage in case of prospective practical usage, since the 3D models are reconstructed and therefore do not need to be obtained beforehand. Only the real-world dimensions need to be available.

6.3 General Limitations of the Reconstruction Method

To conclude the evaluation, this section aims to summarize and discuss the general limitations of the proposed method for 3D reconstruction of passing vehicles. As it is usual with camera-based system, results of the algorithm are dependent on the visual quality of the input. The method can therefore be expected to fail in situations where the quality of the supplied input images degrades, such as in case of dark and poorly-lit scenes, and during foggy or rainy weather. Moreover, several other limitations can be identified:

Straight road assumption. The method assumes that the trajectory of observed vehicles is a straight line (see Section 5.4 for more details). In other words, the camera is expected to be observing a straight part of the road. This implies an important restriction on selection of camera placement.

Good quality of input segmentation. The method expects a reasonably good quality of the input vehicle segmentation, i.e. the foreground masks. Supplied masks should only contain the vehicle and should not include any part of the background. On the other hand, significant portions of the vehicle should not be missing when the mask is applied. Problems are also likely to be caused by overlapping vehicles, or vehicles segmented together as one object.

Time consumption. The method is not devised to work in real time. Both the correspondence search and the reconstruction phase are time-costly processes, and the entire reconstruction of one vehicle can take up to several minutes. Moreover, the reconstruction process can start only after all images of the particular vehicle have been obtained.

View orientation. The reconstruction method expects the overall view of the camera to be set so that both the front part and the side of a passing vehicle are captured. Otherwise, a vast portion of 3D structure of the vehicle would not be available for the reconstruction process, and the reconstruction might fail, or produce a severely incomplete model.

However, most of the above mentioned limitations do not pose any major restrictions on practical usage of reconstruction of passing vehicles. Selecting a straight part of the road for camera placement is usually not a significant limitation. Moreover, instead of being a restriction, the last requirement of camera orientation can in fact be perceived as an advantage. It allows the camera to be mounted near the roadside, which considerably lowers the difficulty of its deployment and maintenance.

Furthermore, the aforementioned time aspect of the method should be briefly addressed. Most importantly, it might seem that the calculation not being performed in real time may hinder any practical usage of the method. Nevertheless, the primary aim of examining the topic of vehicle reconstruction is to contribute to automatic camera calibration process, particularly to determining the scene scale. Since camera calibration phase does not have to be carried out in real time, vehicle reconstruction taking up to several minutes is not a serious drawback. Once calibrated, the camera can be used for traffic analysis tasks for a longer period of time, before re-calibration is necessary. Therefore, the fact that the algorithm is not capable of real-time performance does not prevent its practical application.

Chapter 7

Proposals for Further Modifications of the Reconstruction Pipeline

Based on the experiments performed with the 3D reconstruction applied on passing vehicles, this chapter aims at providing detailed proposals and suggestions for further modifications of the reconstruction pipeline. The observations about reconstruction process and outcomes are first summarized. Subsequently, the concepts of possible further improvements are presented.

7.1 Summary of Observations

Throughout the performed experiments with 3D reconstruction of passing vehicles, the standard incremental Structure from Motion pipeline was gradually modified, as described in Chapter 5. In the final reconstruction pipeline, three main parts can be identified. First, the pipeline includes custom correspondence search stage instead of the original one. Secondly, unchanged incremental reconstruction stage is utilized. Thirdly, the reconstructed model is refined using additional motion constraints.

Experiments with this pipeline design have revealed important parts within the reconstruction process that should be given further attention. These are often closely related to the iterative reconstruction phase, which is so far left unmodified, and thus allows a high level of freedom to enable general reconstruction scenarios. When the described reconstruction pipeline was used, the following situations were observed.

Firstly, Structure from Motion algorithms are known to be sensitive to selection of the initial image pair [24]. This has also proved to be a significant factor when reconstructing passing vehicles. Reconstruction outcome has shown to heavily depend on the initial two-view reconstruction selected by the algorithm. Moreover, Necker reversal of the point cloud [24] (the generalised version of the optical illusion where two 3D interpretations of 2D drawing of a cube are possible; see Figure 7.1) could sometimes be observed. This resulted in a vehicle reconstruction where camera poses were all localized below the vehicle, rather than above. An example of a correct and reversed model of the same car are shown in Figure 7.2 and Figure 7.3, respectively.

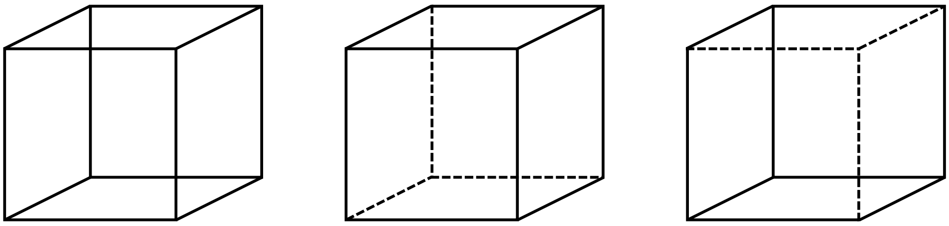

Figure 7.1: Necker cube (left). An optical illusion where 2D drawing of a cube has two possible 3D interpretations, as shown in the center and on the right.

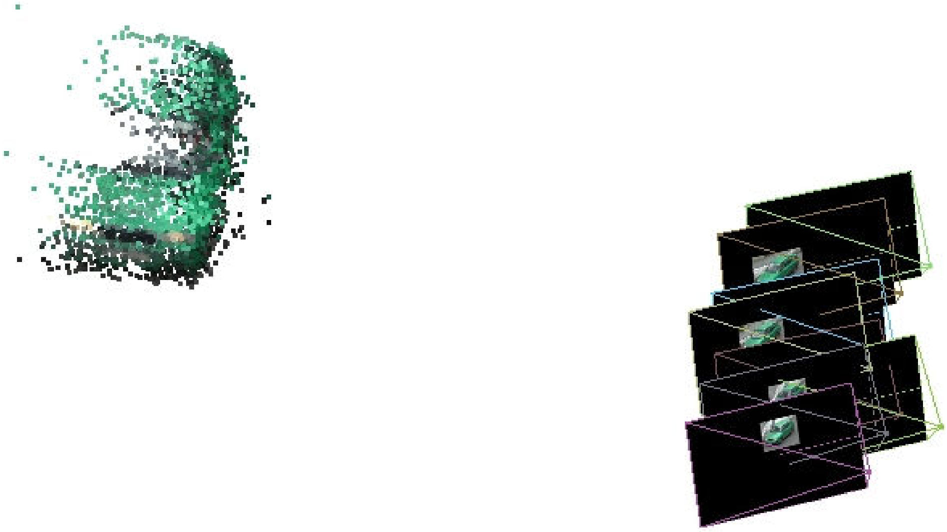

Figure 7.2: An example of a successfully reconstructed model where the reversal did not occur. The camera poses are viewing the vehicle correctly – from above.

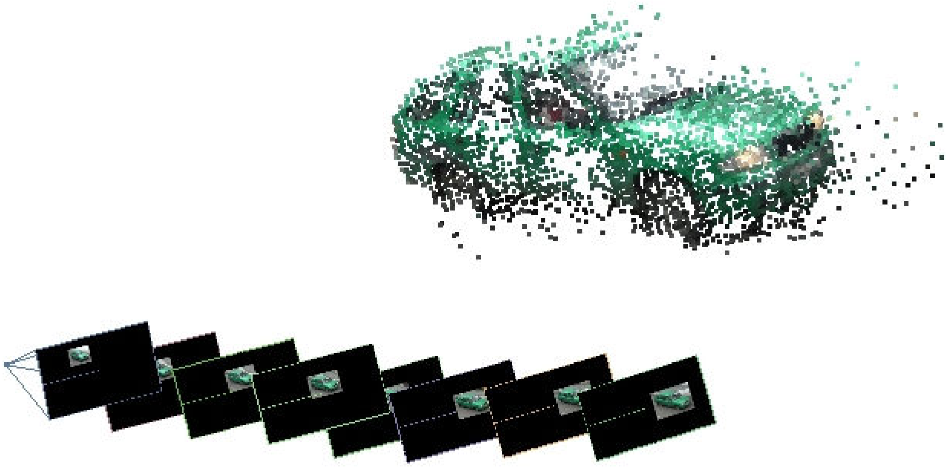

Figure 7.3: An example of Necker reversal. The camera poses are located *below* the model, and the vehicle is reconstructed as is if being viewed from inside.

Secondly, in the vast majority of cases, camera poses were imprecisely estimated. Instead of respecting the original linear vehicle movement (see Section 5.4 for motion constraints discussion), camera poses were often placed away from the expected aligned configuration. Nevertheless, severely incorrect estimation of a camera pose early in the reconstruction process can have a devastating effect on the entire reconstruction result, since misplaced 3D points added to the point cloud by such camera pose are likely to be later used for other pose estimations, and thus prevent them from being localized correctly as well.

Applying the additional motion constraints to enforce the alignment of the camera poses has been carried out by subsequent refinement by the SLAM++ library. However, this has only shown to be applicable in cases where camera poses were already estimated moderately close to the correct positions. Imposing the motion constraints on a reconstruction with camera positions far from the expected straight trajectory did not yield desired improvements. The reconstruction process can, however, generally result into such situations. This observation leads to an important suggestion of incorporating the constraints directly into the incremental reconstruction phase, as described in full detail in the following Section 7.2. This step can be expected to significantly improve the overall stability and success rate of the reconstruction pipeline.

Moreover, in cases where successful reconstruction was obtained and refined, it could be seen that the overall precision of the point cloud could still be improved. This is also supported by the evaluation provided in Chapter 6, which indicates that improvements in the precision of the reconstructed model are desirable in order to achieve better measurement results. Therefore, the proposals provided in the following sections are not only aimed at enhancing the stability of the reconstruction process, but also at upgrading the precision of the resultant point cloud.

7.2 Outline of a Custom Pipeline Structure

The observations presented in the previous section suggest that attention should be focused on the incremental reconstruction phase, which tends to violate the related motion constraints. Therefore, it would be beneficial to design an entirely custom 3D reconstruction pipeline, which would consider and exploit the prior knowledge that can be applied to the problem of passing vehicle reconstruction.

In this specialised pipeline structure, the modified correspondence search stage (as described in Section 5.3) could be used. Subsequently, it would be necessary to deal with the initialization of the reconstruction process. Afterwards, the modified iterative reconstruction together with bundle adjustment can take place. Both the initialization and iterative reconstruction phase are now addressed.

7.2.1 Selection of Initial Image Pair

Initialization of the Structure from Motion algorithm is done by selecting two images for which the corresponding camera poses are estimated from the positions of matched feature points. From these two views, the first 3D points in the model are triangulated. The initialization step is therefore a critical part of the incremental reconstruction stage, as all subsequent pose estimations and point triangulations are directly affected by the quality of the initial two-view reconstruction.

The two images selected for the reconstruction initialization should contain a high number of point correspondences and should also have a large baseline, i.e. the distance between

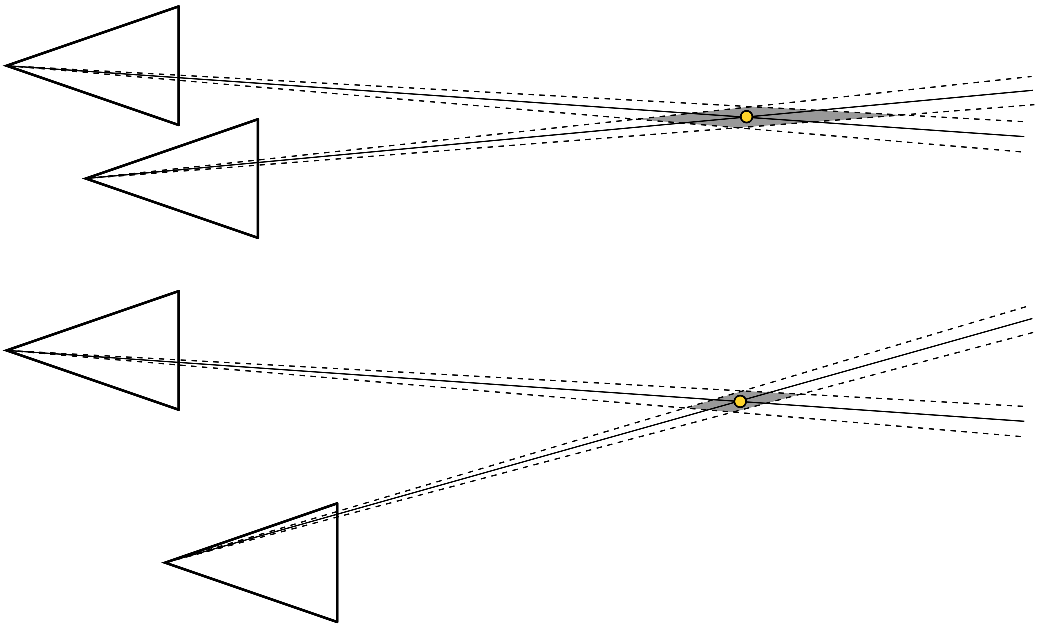

Figure 7.4: The uncertainty in estimation of the 3D point location rises with the decreasing angle between the projected rays, caused by smaller distance between the camera centers. In general, points are less precisely localized when rays become more parallel. The yellow dot marks the original 3D point. However, the point can be triangulated anywhere within the *uncertainty region* (shaded area). This is caused by the fact that the position of the corresponding feature points within the images are affected by noise and their 2D coordinates can therefore be imprecise.

the camera centers [24]. The requirement of a large baseline is crucial, since small distance between camera poses results into rays becoming closer to parallel during the triangulation of 3D points. This enlarges the uncertainty region and thus increases the possibility of imprecise point localization, as demonstrated by Figure 7.4.

It is important to note that the demand for selecting initial images with the highest possible amount of matched points and the need for these images to be far apart are, to some extent, two contradictory requirements in case of reconstruction of passing vehicles. This is due to the fact that the biggest number of corresponding points is likely to be acquired for the last two images of the passing vehicle, where the vehicle is close to the camera, and therefore the images provide a view with the best resolution[1]. However, such image pairs will inevitably have extremely small distance between the centers of the respective camera poses. Selecting such pair for initialization can thus be expected to produce very inaccurate two-view reconstruction.

In order to achieve higher precision of the resultant point clouds, it will be necessary to relax the requirement on the number of matched feature points and instead focus on rejecting images that are too close in the obtained image sequence. For this purpose, the knowledge about the order of images within the sequence can be utilized. Moreover, it is very probable that a reasonable choice for one of the two initial images will be the last

[1] Note that only the images where the entire vehicle is seen are considered. Images with incomplete vehicle (e.g. when the vehicle is leaving the scene and is already partially away from the image) are not included in the sequence.

image of the sequence (the one with the closest view of the vehicle). This image could then be paired with other remaining images and the fitness of each pair for the initialization should be assessed.

To this end, the approach of Beder and Steffen [1] could be integrated. In their work, Beder and Steffen propose a statistically motivated metric for automatic evaluation of the quality of candidate image pairs. The metric is based on expressing the uncertainty region, within which a 3D point can be localized, using an ellipsoid. The shape of the ellipsoid is determined by the point correspondences and respective camera poses, and can therefore be used for assessment of the given image pair. More precisely, the roundness of the ellipsoid (the ratio of its smallest and longest axis) is used as a measure of quality. Higher value of the roundness metric then implies that the particular image pair is more suitable for initializing the reconstruction.

As already suggested, the first image in the vehicle sequence could be fixed to become one of the initial images, and would subsequently be paired with other images to obtain possible candidates for initialization. The approach based on the roundness metric can then be used to evaluate each of these pairs. The image pair with the best score would then be selected to initialize the reconstruction.

7.2.2 Recovering Intrinsic Parameters

The initialization step is also closely related to intrinsic parameters of the camera, more specifically its focal length. (Other intrinsic parameters are expected to be known – the principal point is assumed to be at the center of the image and the pixels are expected to exhibit zero skew). In order to enable 3D reconstruction initialization using the 5-point algorithm, intrinsic parameters need to be available, including the focal length. Nevertheless, the value of the focal length often remains unknown in many practical situations where manual camera calibration cannot be performed. This is also applicable to the task of reconstruction of passing vehicles, since the camera calibration process is desired to be fully automatic.

For cases where the focal length is unknown, approaches based on 6-point algorithm have been developed in order to obtain the focal length value together with the initial two-view reconstruction [2, 16, 4]. These are often referred to as auto-calibration or self-calibration methods. Adding one more point correspondence to the original 5-point solver provides the necessary constraint which allows solving the additional unknown parameter within the initialization step. The 6-point algorithm applies to cases with constant focal length, for example a situation when all images were taken using the same camera without changing the settings. One seemingly possible option would therefore be using an existing 6-point algorithm to simultaneously obtain the initial two-view reconstruction and the unknown focal length.

However, several authors, such as Sturm [32] and Kahl [13], have studied the inherent limitations of self-calibration and identified the conditions under which the retrieval of focal length is impossible or ambiguous. Such conditions can arise from specific camera movements within the scene and are thus called *critical motion sequences*. One of the described critical motion sequences, for which any self-calibration approach will inevitably fail, occurs when optical axes of all camera poses are parallel. In other words, a situation where the camera underwent a purely translational movement. As shown in Section 5.4 (especially Figure 5.5), this is exactly the case with the reconstruction of passing vehicles.

It is therefore evident, that within the task of 3D reconstruction of passing vehicles, no self-calibration algorithm can be utilized to automatically recover the focal length jointly with the reconstruction process. This is an important limitation to be considered when implementing the custom reconstruction pipeline. As a result, it will be necessary to retrieve the focal length before the initialization step of the iterative reconstruction phase. For this purpose, a method based on vanishing points detected in the viewed scene can be used. The most suitable candidate is the approach proposed by Dubská et al. [8] and later improved by Sochor et al. [27], which determines the vanishing points from the observed traffic flow and provides a reliable focal length value.

7.2.3 Coping with Necker Reversal

Necker reversal is another problem associated with the initialization step [24]. If the initial reconstruction from the first two selected images produces a reversed point cloud, the entire reconstruction outcome will be reversed. Since this phenomenon could sometimes be observed in case of vehicle reconstructions, it is necessary to overcome this problem inside the initialization process of a custom pipeline design.

To deal with Necker reversal, Brown and Lowe [3] suggest to try both of the possible 3D interpretations and use the one which minimizes the reprojection error. They first perform regular initialization, run bundle adjustment on the initial two-view reconstruction, and save the result. Afterwords, they swap the camera positions and flip the depth of the reconstructed points to acquire the reversed interpretation. Bundle adjustment applied on the second case converges to a different local minimum than in the first one. The interpretation which minimizes the resultant error is then kept for the subsequent reconstruction process.

7.2.4 Direct Application of Motion Constraints

The key modification in the custom pipeline design should arise from the motion constraints specific to the task of reconstructing passing vehicles, as identified in Section 5.4. Instead of refining the model after the iterative reconstruction to enforce the motion constraints, the constraints should become an inseparable part of the iterative reconstruction phase. More precisely, the fact that the camera undergoes pure translation and no relative rotations occur, should be directly incorporated. This applies to both the estimation of the initial two-view poses, as well as the iterative step where a new camera pose is added. This is now elaborated on in the following subsections.

7.2.5 Determining the Relative Pose of the Initial Image Pair

The relation between the two initial camera poses is determined by epipolar geometry and encoded using fundamental matrix $\mathbf{F}$ (which does not encompass camera calibration) and essential matrix $\mathbf{E}$, which can be formed from the fundamental matrix using the calibration parameters of the corresponding cameras. In general, the relation between the essential and the fundamental matrix, given the matrices $\mathbf{K'}^T$ and $\mathbf{K}$ that contain intrinsic parameters of the two cameras, can be expressed as [11]:

$$\mathbf{E} = \mathbf{K'}^T * \mathbf{F} * \mathbf{K} \tag{7.1}$$

From the essential matrix, the relative translation and rotation between the two camera poses can be extracted. Commonly, five-point algorithms are used, such as the one proposed by Nistér [19] or Stewénius [31], to compute the essential matrix and subsequently perform its decomposition to obtain the camera poses for the initial two-view reconstruction. These standard approaches assume arbitrary orientation of the two cameras.

However, instead of solving the entire non-linear system to obtain the general-case essential matrix, the prior knowledge of the relative rotation between the camera poses being zero can be exploited. As a result, the initial pose estimation will be given less freedom, and it will thus be less prone to contaminating the initialization by incorrect estimate. Moreover, the assumption of no change in rotation between the poses will significantly simplify the original system of equations necessary for obtaining the relative pose.

Firstly, purely translational movement of the camera results into coinciding epipoles in both images (i.e. their location in both images will be the same). This fact, together with the rotation becoming identity matrix, leads to an important simplification of the fundamental matrix, which can in this case be expressed as [11]:

$$\mathbf{F} = [e]_\times \tag{7.2}$$

with e being the epipole, and where the notation $[a]_\times$ denotes a skew-symmetric matrix (matrix representation of vector product), which is defined for vector $a = (a_1, a_2, a_3)$ as:

$$[a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{7.3}$$

At this point, knowing the location of the epipole is enough to build the fundamental matrix, which could then be used to calculate the essential matrix using Equation (7.1). The essential matrix could subsequently be decomposed with the same method as is used for the matrices obtained by regular five-point algorithms to obtain the relative camera poses.

Nevertheless, this approach should not be necessary, since it is possible to simplify the pose estimation even further. This simplification becomes more straightforward when the usual assumptions made during the initialization process are considered.

When initializing the iterative reconstruction phase using the standard five-point algorithms, the following assumptions are commonly made to prevent ambiguities during the decomposition of the essential matrix. The first camera pose is assumed to be positioned at the center of the 3D coordinate space, and its rotation is expressed as an identity matrix. The rotation of the second camera pose is then expressed relative to the first one. The same applies to translation, with one additional assumption. Since no information about the scale of the reconstruction is available, the relative translation vector is chosen to have unit length.

Fixing the first camera pose at the center of the 3D world, in combination with the assumption about the translation vector having unit length, can be utilized to directly obtain the second camera pose from the location of the epipole, without the necessity to compute the essential matrix. As stated, the first camera pose will be positioned at the center of the 3D space. Its known focal length and the coordinates of the epipole in the image plane can then be used to express a 3D direction vector from the camera center towards the epipole. Normalizing the vector to unit length will then result into obtaining the location of the second camera pose. This can also be seen from Figure 7.5.

The key part of the initialization process will therefore be determining the epipole location in the selected image pair. Even though only two point correspondences are enough

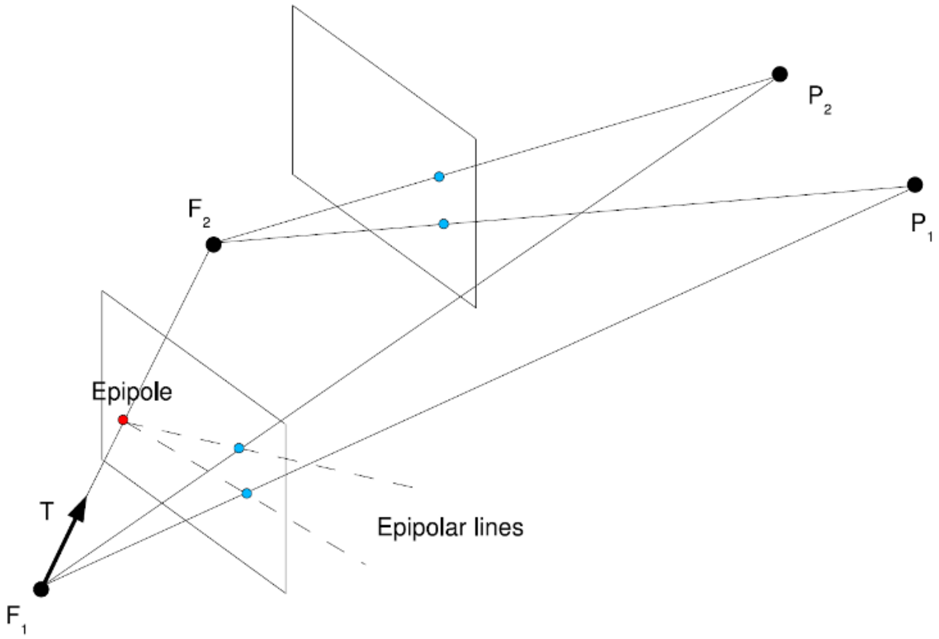

Figure 7.5: When camera undergoes pure translation, the direction vector towards the epipole directly defines the direction of the translational movement. F_1 and F_2 are the camera centers. P_1 and P_2 are the points used to determine the epipole location. The translation direction is shown by vector T. (Note that the epipole can also be positioned within the image plane, but outside of the original image.) Adapted from [5].

to obtain the epipolar lines that uniquely define the epipole, it is desirable to consider more corresponding points to minimize the effect of noise, and use e.g. the method proposed by Chen et al. [7].

However, the epipole location in case of pure translation also coincides with the position of a vanishing point [11]. This knowledge can be readily used to avoid the need of using any additional methods for its retrieval. As described in Subsection 7.2.2, it will be necessary to integrate a method for obtaining focal length of the camera. Since a suitable approach for this task (proposed by Dubská et al. [8] and improved by Sochor et al. [27]) is based on finding the vanishing points, the coordinates of the epipole can be directly obtained by retrieving the location of the respective vanishing point from this method.

7.2.6 Iterative Reconstruction

After the initialization is completed, the iterative reconstruction phase takes place. Similarly to the initial pose estimation, motion constraints should be applied whenever new camera pose is added. Again, exploiting the knowledge of no relative change in camera rotation should significantly simplify the pose estimation process.

Estimation of additional camera pose with known orientation results into a linear problem where only 1.5 point correspondences would suffice (solving only for the unknown position, which has three degrees of freedom, leads to a situation where only three out of four coordinates of two selected corresponding points are necessary, hence 1.5) [15]. However, using only such a small amount of information is not a suitable approach. An outlier-rejection scheme was presented for precisely this case by Larsson et al. [15] to avoid estimating the pose based on spurious matches. The remaining points after filtering can then be used to over-determine the problem and compute the pose robustly.

One of the important parts of the iterative reconstruction is the selection of the next view to be added. Schönberger and Frahm [22] show that high quality of resultant point cloud model is achieved when the feature points that correspond to the already reconstructed 3D points are uniformly distributed within the added image. They propose a method which evaluates candidate images with a score that reflects the number of observed feature points together with the extent to which they are clustered or uniformly distributed.

However, practical usage of this approach in reconstruction of passing vehicles may be limited by the fact that the vehicle is only seen within a small portion of the image, where all detected point correspondences are located, and the remaining parts of the image contain no feature points. Therefore, for the task of 3D reconstruction of passing vehicles, it will be necessary to carry out further experiments to determine the best strategy for the selection of the next image to be added.

An inseparable part of the iterative reconstruction is the bundle adjustment step, which refines the reconstruction by optimizing reprojection error. This step can be run multiple times during the reconstruction process to prevent accumulation of error. However, bundle adjustment is known to be costly in large-scale Structure from Motion tasks and therefore there is a tendency to minimize the number of times this step is applied. Commonly, bundle adjustment is only run after adding several new camera poses or after the size of the point cloud rises by a predefined percentage. This is also true for VisualSFM [35].

Since reconstruction of a vehicle deals with a rather small model (compared, for example, to reconstruction of central Rome from thousands of images), bundle adjustment could be applied after every newly added camera pose, in order to maintain high accuracy of reconstructed points. Moreover, the bundle adjustment step should also include the already mentioned motion constraints to enforce them directly inside the iterative reconstruction. More specifically, only the alignment of the camera poses can be considered for refinement, as the rotation of the poses will already be set to remain the same by the pose estimation step.

For the purpose of bundle adjustment the customizable SLAM++ library can be used, similarly as in Section 5.5. It will also be beneficial to relax the constraint which assumes negligible change in the speed of passing vehicles to enable usage of the reconstruction pipeline in a wider range of more general settings, where vehicles may accelerate or decelerate significantly, for example when the camera is placed next to a crossroad.

7.3 Using the Pipeline with Two Sets of Matches

An additional proposal for experimenting with the reconstruction of passing vehicles is using the pipeline outlined in this chapter with two sets of keypoint matches. This idea is motivated by the fact that the overall quality of 3D reconstruction is often negatively affected by imprecise feature points that can, to some extent, remain within the filtered set even after using methods such as RANSAC. Moreover, most parts of vehicle surface have proven to be challenging for precise localization of keypoints due to the lack of distinctive texture.

It may therefore be interesting to extend the proposed approach based on DeepMatching (see Section 5.3) to include another feature extraction algorithm that would aim at providing a smaller but extremely reliable set of matched points. The entire reconstruction pipeline would then be first used with this set of matches to obtain an estimate of all camera poses with high accuracy. Once all the poses are known, the quasi-dense matches obtained by DeepMatching would be added to the reconstruction to improve the completeness of the point cloud.

In other words, only a very sparse point cloud would firstly be created, with the aim of obtaining a precise estimation of camera poses. The point cloud would then be extended by matches obtained by DeepMatching to produce a model with high density of 3D points. As a result, the overall precision of the reconstruction could be improved, while maintaining high coverage of the final model structure.

It should be noted that once the camera poses are estimated, adding more points to the reconstructed model is simplified to triangulating the new points by casting rays for the particular keypoint locations from respective camera poses and solving for the intersection of the rays to find the 3D positions of the given points. Since the rays will generally not intersect at one precise point in space, the solution should be obtained in a least-squares manner. Subsequently, the points with large residual error (i.e. the points where the rays were rather far from the computed optimal location) could be rejected as outliers and eliminated to prevent contamination of the point cloud with inaccurate points.

The additional set of feature points could be obtained by detecting highly reliable keypoints and by their subsequent tracking in the captured video. In this way, the knowledge that the image sequence is ordered would be directly exploited, and thus the risk of incorrectly matched points would be lowered. For the purpose of tracking, Good Features to Track [23] or similar method could be used. These feature extractors are usually based on corner structures, and therefore should also be applicable in case of passing vehicles, where significant corners are present, for example around windows, doors, lights, etc. Moreover, it might be useful to experiment with tracking in a sequence with reversed order. In other words, to start with the last image of the vehicle (where the vehicle is closest to the camera and therefore has the best resolution), and continue with previous images in the sequence. Additionally, tracking in both directions could be used to check for consistency.

7.4 Prospective Utilization of Vehicle Reconstructions in Traffic Analysis System

Prospective practical usage of 3D reconstruction of passing vehicles is in a camera calibration module of a traffic analysis system. More specifically, the main contribution of the vehicle reconstruction is in determining the scale of the observed scene. Once the scene scale is determined the traffic analysis system can be used for various purposes, including speed measurement of observed vehicles.

The camera calibration module would contain the following parts (as shown in the diagram in Figure 7.6). First, focal length of the camera would be determined. Passing vehicles would then be detected and segmented to obtain image sequence for each vehicle from the video stream. The image sequence would then be passed to the 3D reconstruction pipeline in order to obtain the vehicle model. At the same time, classification of the vehicle would be performed using the original images in the sequence. The result of the classification would be a precise identification of vehicle make and model. This information would then be used to select the corresponding entry from a database of vehicle dimensions. As the last step, the dimensions of the vehicle would be coupled with the reconstructed model to

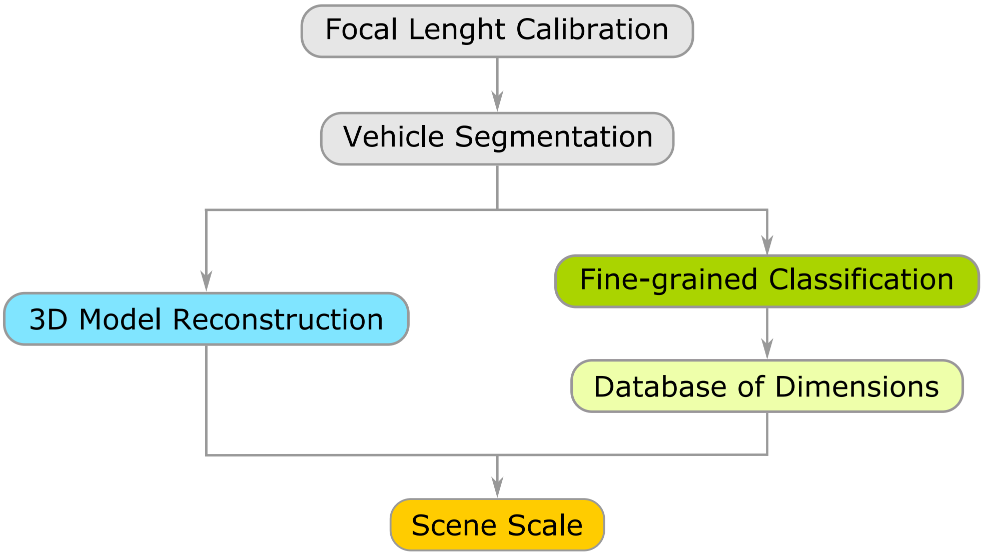

Figure 7.6: Diagram of prospective camera calibration module for traffic analysis system.

determine the scale of the observed scene. Additionally, scale information computed from more vehicle models can be combined to improve precision.

To acquire the focal length, the method proposed by Dubská et al. [8] is a suitable candidate. Background subtraction algorithms can be used for vehicle segmentation. For fine-grained classification of vehicle images, Sochor et al. [26, 29] devise an approach based on convolutional neural networks. The database with vehicle dimensions would then be created using data available from vehicle blueprints. The remaining building block is the reconstruction of vehicle model addressed in this thesis.

Chapter 8

Conclusion

In this thesis, reconstruction of 3D information about vehicles passing in front of a surveillance camera was addressed. The possibilities for reconstruction of passing vehicles were explored and evaluated. In addition, proposals for potential further advancements were summarized.

First, a set of experiments with SIFT feature matching and Structure from Motion algorithm was carried out in order to examine their results on images of passing vehicles. SIFT features were found to be unsuitable for images of vehicles when 3D reconstruction is to be performed. Therefore, modifications to the correspondence search stage of Structure from Motion pipeline were proposed. SIFT features were substituted by DeepMatching in order to obtain quasi-dense correspondences for the subsequent reconstruction phase. Moreover, filtering of the computed correspondences using foreground masks was involved to eliminate points that are not located on the vehicle. Implementation of these modifications significantly improved the overall completeness of the reconstructed point cloud models.

Furthermore, the models were refined by enforcing additional motion constraints that are specific to the task of vehicle reconstruction. The resultant point clouds were then evaluated with respect to the intended use for scene scale inference. Lastly, observations and acquired information about the process of vehicle reconstruction were utilized to form proposals for prospective design of an entirely custom pipeline that would be specialized for 3D reconstruction of passing vehicles.

This thesis was presented at Excel@FIT 2017, a student conference organized at Faculty of Information Technology of Brno University of Technology, and was awarded by one of the industrial partners of the conference. The work was also presented at Central European Seminar on Computer Graphics 2017 (CESCG) and won 2^{nd} best paper award and 1^{st} best presentation award.

Prospective practical application of 3D reconstruction of passing vehicles is within the camera calibration module of a traffic surveillance system. Reconstructed vehicle models can be combined with their real-world dimensions to determine the scale of the observed scene. The surveillance camera can then be utilized for various traffic analysis tasks, including speed measurement.

Bibliography

[1] Beder, C.; Steffen, R.: *Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction from an Image Sequence*. Springer Berlin Heidelberg. 2006. ISBN 978-3-540-44414-5. pp. 657–666. doi:10.1007/11861898_66.

[2] Bocquillon, B.; Bartoli, A.; Gurdjos, P.; et al.: On Constant Focal Length Self-Calibration From Multiple Views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007. ISSN 1063-6919. pp. 1–8. doi:10.1109/CVPR.2007.383066.

[3] Brown, M.; Lowe, D. G.: Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM)*. 2005. ISSN 1550-6185. pp. 56–63. doi:10.1109/3DIM.2005.81.

[4] Bujnak, M.; Kukelova, Z.; Pajdla, T.: Robust Focal Length Estimation by Voting in Multi-view Scene Reconstruction. In *Proceedings of the 9th Asian Conference on Computer Vision (ACCV)*. Springer-Verlag. 2010. ISBN 3-642-12306-6, 978-3-642-12306-1. pp. 13–24. doi:10.1007/978-3-642-12307-8_2.

[5] Burschka, D.; Mair, E.: Direct Pose Estimation with a Monocular Camera. In *Proceedings of the 2nd International Conference on Robot Vision*. 2008. ISBN 3-540-78156-0. pp. 440–453.

[6] Cathey, F. W.; Dailey, D. J.: A Novel Technique to Dynamically Measure Vehicle Speed Using Uncalibrated Roadway Cameras. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005*. 2005. ISSN 1931-0587. pp. 777–782. doi:10.1109/IVS.2005.1505199.

[7] Chen, Z.; Pears, N.; McDermid, J.; et al.: Epipole Estimation under Pure Camera Translation. In *Digital Image Computing: Techniques and Applications (DICTA)*, vol. 3. 2003. pp. 849–858.

[8] Dubská, M.; Herout, A.; Juránek, R.; et al.: Fully Automatic Roadside Camera Calibration for Traffic Surveillance. *IEEE Transactions on Intelligent Transportation Systems*. vol. 16, no. 3. 2015: pp. 1162–1171. ISSN 1524-9050. doi:10.1109/TITS.2014.2352854.

[9] Dubská, M.; Sochor, J.; Herout, A.: Automatic Camera Calibration for Traffic Understanding. In *Proceedings of British Machine Vision Conference (BMVC)*. 2014. pp. 1–10.

[10] Fischer, P.; Dosovitskiy, A.; Ilg, E.; et al.: FlowNet: Learning Optical Flow with Convolutional Networks. *CoRR*. vol. abs/1504.06852. 2015.

[11] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press. 2004. ISBN 0521540518.

[12] He, X. C.; Yung, N. H. C.: A Novel Algorithm for Estimating Vehicle Speed from Two Consecutive Images. In *IEEE Workshop on Applications of Computer Vision (WACV)*. 2007. ISSN 1550-5790. pp. 12–12. doi:10.1109/WACV.2007.7.

[13] Kahl, F.; Triggs, B.: Critical Motions in Euclidean Structure from Motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. 1999. ISSN 1063-6919. doi:10.1109/CVPR.1999.784661.

[14] Kümmerle, R.; Grisetti, G.; Strasdat, H.; et al.: g2o: A General Framework for Graph Optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011. pp. 3607–3613. doi:10.1109/ICRA.2011.5979949.

[15] Larsson, V.; Fredriksson, J.; Toft, C.; et al.: Outlier Rejection for Absolute Pose Estimation with Known Orientation. In *Proceedings of British Machine Vision Conference (BMVC)*. 2016.

[16] Li, H.: *A Simple Solution to the Six-Point Two-View Focal-Length Problem*. Springer Berlin Heidelberg. 2006. ISBN 978-3-540-33839-0. pp. 200–213. doi:10.1007/11744085_16.

[17] Liu, C.; Yuen, J.; Torralba, A.: SIFT Flow: Dense Correspondence across Scenes and its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. vol. 33, no. 5. 2011: pp. 978–994. ISSN 0162-8828. doi:10.1109/TPAMI.2010.147.

[18] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*. vol. 60, no. 2. 2004: pp. 91–110.

[19] Nister, D.: An Efficient Solution to the Five-point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 26, no. 6. 2004: pp. 756–770. ISSN 0162-8828. doi:10.1109/TPAMI.2004.17.

[20] Polok, L.; Šolony, M.; Ila, V.; et al.: Efficient Implementation for Block Matrix Operations for Nonlinear Least Squares Problems in Robotic Applications. In *IEEE International Conference on Robotics and Automation (ICRA)*. 2013. ISSN 1050-4729. pp. 2263–2269. doi:10.1109/ICRA.2013.6630883.

[21] Polok, L.; Smrz, P.: Fast Linear Algebra on GPU. In *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS)*. IEEE. 2012. pp. 439–444.

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

[23] Shi, J.; Tomasi, C.: Good Features to Track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Jun 1994. ISSN 1063-6919. pp. 593–600. doi:10.1109/CVPR.1994.323794.

[24] Snavely, K. N.: *Scene Reconstruction and Visualization from Internet Photo Collections*. PhD. Thesis. University of Washington. 2008.

[25] Snavely, N.; Seitz, S.; Szeliski, R.: Photo Tourism: Exploring Image Collections in 3D. *ACM Transactions on Graphics*. 2006.

[26] Sochor, J.; Herout, A.; Havel, J.: BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[27] Sochor, J.; Juránek, R.; Herout, A.: Traffic Surveillance Camera Calibration by 3D Model Bounding Box Alignment for Accurate Vehicle Speed Measurement. *Computer Vision and Image Understanding (under review)*. arXiv:1702.06451.

[28] Sochor, J.; Juránek, R.; Špaňhel, J.; et al.: BrnoCompSpeed: Review of Traffic Camera Calibration and A Comprehensive Dataset for Monocular Speed Measurement. *IEEE Transactions on Intelligent Transportation Systems (under review)*.

[29] Sochor, J.; Špaňhel, J.; Herout, A.: BoxCars: Improving Vehicle Fine-Grained Recognition using 3D Bounding Boxes in Traffic Surveillance. *International Journal of Computer Vision (under review)*. arXiv:1703.00686.

[30] Šonka, M.; Hlaváč, V.; Boyle, R.: *Image Processing, Analysis and Machine Vision*. PWS. 1999. ISBN 0-534-95393-X.

[31] Stewénius, H.; Engels, C.; Nistér, D.: Recent Developments on Direct relative Orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*. vol. 60. 2006: pp. 284–294. doi:10.1016/j.isprsjprs.2006.03.005.

[32] Sturm, P.: Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun 1997. ISSN 1063-6919. pp. 1100–1105. doi:10.1109/CVPR.1997.609467.

[33] Szeliski, R.: *Computer Vision: Algorithms and Applications*. Springer Science & Business Media. 2010. ISBN 978-1848829343.

[34] Weinzaepfel, P.; Revaud, J.; Harchaoui, Z.; et al.: DeepFlow: Large Displacement Optical Flow with Deep Matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*. 2013.

[35] Wu, C.: Towards Linear-time Incremental Structure from Motion. In *2013 International Conference on 3D Vision (3DV 2013)*. IEEE. 2013. pp. 127–134.

[36] Zhang, Z.: A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. vol. 22, no. 11. 2000: pp. 1330–1334. ISSN 0162-8828. doi:10.1109/34.888718.