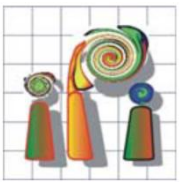
Leibniz University Hannover

Institute of Photogrammetry and Geoinformation



Master thesis

Tracking and Pose Estimation of Vehicles from Stereo Image Sequences

Bowen Zhou B.Sc.

Matr.-Nr.: 3201270

Supervisors: apl. Prof. Dr. techn. Franz Rottensteiner
M.Sc. Max Coenen

Hannover, September 2017

Institut für Photogrammetrie
und GeoInformation

Leibniz Universität Hannover

Institut für Photogrammetrie und GeoInformation,
Nienburger Straße 1, 30167 Hannover

Fakultät für Bauingenieurwesen
und Geodäsie

Institut für Photogrammetrie
und GeoInformation

Prof. Dr.-habil. Christian Heipke

M.Sc. Max Coenen

Tel. +49 511 762-2488
E-Mail: coenen@ipi.uni-hannover.de

20.03.2017

Topic of the master thesis of B.Sc. Bowen Zhou

Tracking and Pose Estimation of Vehicles from Stereo Image Sequences (Working title)

In many future sensor network systems, especially in automated or autonomous driving environments, collaborative positioning will play an important role. To that end, the detection, tracking and pose estimation of vehicles constitute one of the main contributions. Concentrating on real world street scenarios, this leads to a complex vehicle recognition problem and beyond to the need of suitable techniques for precise 3D object reconstruction which enables the derivation of relative poses with respect to the ego position. The problem of vehicle pose estimation should be tackled in this master thesis on the basis of stereo image sequences acquired by vehicle mounted stereo cameras.

The goal of this master thesis is the detection, tracking and pose estimation of vehicles from stereo image sequences. As current developments show, the usage and integration of 3D shape priors in form of deformable object models are able to enhance the object pose estimation. This is why a 3D Active Shape Model (ASM), which is learned from 3D vehicle CAD data, is supposed to be used in this master thesis to optimize the trajectory parameters of a vehicle throughout its track. For this purpose, a 3D stereo reconstruction has to be conducted in a preceding step using a dense matching technique. Also, an existing vehicle detection technique has to be applied, to deliver the initial vehicle detections. In the theoretical part of this master thesis, Mr. Zhou has to develop a concept to associate the detected vehicles into tracks and to fit a common vehicle model into each tracked object to refine the respective pose estimation of the individual objects in each frame.

Concerning the practical implementation, the developed concept of Mr. Zhou is supposed to be built on methods that already exist at the Institute of Photogrammetry and GeoInformation (IPI) e.g. for dense matching, 3D vehicle detection and 3D ASM. These programs can be adapted by Mr. Zhou if needed. The experimental evaluation of the developed method should be conducted on sequences of the KITTI benchmark data set which contains reference detections, tracks and poses for all visible vehicles. This reference information should be compared to the results of the developed method in the experimental evaluation.

M.Sc. Max Coenen / apl. Prof. PD. Dr. techn. Franz Rottensteiner

Visitors adress:
Nienburger Straße 1
30167 Hannover
www.ipi.uni-hannover.de

Declaration of Authorship

I declare that this thesis and the work presented in it are my own. I confirm that:

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Abstract

In the highly dynamic street environments, individually moving traffic participants, in particular vehicles are challenges for the automated and autonomous driving systems. In order to discover the behaviours of other moving vehicles, tracking and pose estimation of these vehicles are essential tasks. Consequently, the automated and autonomous driving systems can understand the traffic environments and anticipate early on based on these information.

In this master thesis, we tackle the tracking and pose estimation of vehicles problem on the basis of street level stereo image sequences, which are acquired by static stereo cameras. For tracking, we follow the tracking-by-detection strategy and try to associate the corresponding detections over time. The association is formulated as an energy minimization problem, where the energy function consists of data association and motion term. Based on tracking results, we try to reconstruct the tracked vehicles in 3D by applying a model-based approach, making use of a deformable 3D vehicle model which is learned from CAD models of vehicles. By fitting one model to each tracked vehicle, the optimal pose and shape parameters of a tracked vehicle can be determined.

The evaluation of the developed approach is performed experimentally based on the object tracking benchmark from the KITTI Vision Benchmark Suite (Geiger et al., 2012). For all vehicles, all of the tracked vehicles are correct and 74% vehicles in the reference can be tracked successfully. For fully visible tracked vehicles, the completeness owns 96%. More than 82% positions and 71% orientations of the tracked vehicles are estimated correctly. In the future study, our approach will be extended to the stereo image sequences, which are acquired by the stereo cameras with ego-motion.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, automated and autonomous driving systems are coming of age and developing quickly. The automated and autonomous driving systems are equipped with multiple sensors to understand highly dynamic environments and deal with all kinds of situations in the complex scenes. In real world street scenarios, interaction with other traffic participants, in particular with other vehicles are challenges for automated and autonomous driving systems (Janai et al., 2017). To this purpose, tracking and estimating poses of these vehicles are essential components for the interaction.

Tracking of other traffic participants is a very important task in understanding the complex street scenes. For instance, in the case of collision with other vehicles, the autonomous driving system needs to react early enough, because the braking distance of a vehicle increases quadratically with its speed. The trajectory of other vehicles allows to predict the future location and brake early to avoid the collision. Additionally, tracking of other vehicles can be useful to the automatic distance control and other operations, such as overtake, turning, of the autonomous driving systems (Janai et al., 2017). However, different types of traffic participants appear in the street scenes simultaneously and tracking a certain type of objects needs the detection firstly. To that end, the tracking-by-detection strategy, only considering the relevant objects, is helpful for tracking the certain type of objects in a complex dynamic street scenes (Ošep et al., 2016). Pose

(position and orientation) estimation of other traffic participants is another important aspect in realizing the autonomous driving system. The relative positions to the location of the autonomous driving system are useful for the self-driving cars to take any driving operations timely and properly. The orientations of these vehicles can help to predict the moving directions and the possible routes of them. The autonomous driving systems can discover the future behaviors of other vehicles based on the poses of them and anticipate early on.

Concentrating on real world street scenarios, the reconstruction of the 3D street scenes is quite useful to accomplish tracking and estimating poses of other vehicles. Frequently, stereo cameras and laser scanners are used as the sensors to acquire high density points of a scene. Compared to laser scanners, stereo images can deliver dense 3D point clouds and additional color information less expensive. Therefore, stereo cameras are assembled into vehicles widely.

1.2 Goal of this thesis

In this master thesis, the goal is to develop a method of tracking and estimating poses of vehicles using stereo image sequences, which are acquired by stereo cameras mounted on a static vehicle. For tracking, we follow the tracking-by-detection strategy, which builds on the vehicle detection approach of Coenen et al. (2017). Consequently, we try to associate the corresponding detections over time and estimate trajectories of each vehicle in one stereo image sequence. Based on tracking results, we attempt to reconstruct the tracked vehicles in 3D by applying a model-based approach, making use of a deformable 3D vehicle model which is learned from CAD models of vehicles. By fitting one model to each tracked vehicle, the optimal pose and shape parameters of a tracked vehicle can be determined.

In this thesis, we try to to solve the problems in (Coenen et al., 2017) by developing our methods. On the one hand, there are initialization problems for vehicle orientation in (Coenen et al., 2017), which only use the direction of the semi-major axis of the bounding box for a detection as initialization of orientation. In our approach, the prior knowledge

about movement direction for each vehicle is used as initialization for orientation, which is closer to the orientation direction of the vehicle. Furthermore, as the 3D modelling method introduced by (Coenen et al., 2017) needs 5 parameters for each detected vehicle, it requires 10 parameters for modelling the same vehicle in two continuous time steps. However, in this approach, only 7 parameters are needed for modelling the corresponding tracked vehicles in two continuous epochs.

1.3 Structure of this thesis

This master thesis is organized as following: in Chapter 2, the related methods and works are reviewed. The theoretical backgrounds, which are necessary to understand the methods, are introduced in Chapter 3. The detailed descriptions of the developed methods in this thesis, about tracking and pose estimation of vehicles using stereo image sequence from static stereo cameras, are given in Chapter 4. The evaluation of the experiments based on the developed methods and the corresponding analysis and interpretations are presented in Chapter 5. Chapter 6 contains a conclusion and an outlook, which discusses the possibilities of optimizing the developed methods in the future.

Chapter 2

Related work

The goal of this master thesis is to detect, track and estimate poses of vehicles based on the stereo image sequences from static cameras. As such, it is related to vehicle detection, tracking-by-detection, vehicle 3D modelling and pose estimation, each of which is reviewed in the following sections.

2.1 Vehicle detection

Vehicle detection from images of urban areas is difficult due to the wide variety of object appearances, changing camera viewpoints, illumination changes, vehicles occluded by other objects and different types and shapes of vehicles, which affect the vehicle detection performance a lot. To handle these challenges, part-based detectors ((Felzenszwalb et al., 2010),(Leibe et al., 2006)) are applied widely. The idea of part-based detectors is to split objects into several simple parts and train a detector for each part and the entire object. The implicit shape model proposed by Leibe et al. (2006), learns flexible representations of image patches for an object. Through extracting local features around interest points and performing clustering of similar patches, a code book is built up in training. Training images from different viewpoints are required to make the detection work for objects under different viewing directions. In order to detect objects, image patches are extracted and matched to the code book for detecting objects such as vehicles. The deformable part model (Felzenszwalb et al., 2010) breaks down complex appearance of objects into representative parts and uses latent support vector machine for training. The model

contains a root filter plus a set of part filters, which are learned from the relative parts of an object, using histogram of oriented gradients (HOG) features. The deformable part model can detect one object with a high complexness, but it may lead to a high rate of false detections. These two part-based detectors deliver 2D bounding boxes as output. In this thesis, we aim to detect and track vehicles in 3D space for further 3D modelling and precise pose estimation.

In order to acquire 3D object information, several approaches (Zia et al., 2013, Pepik et al., 2015) are established based on geometric 3D representations of objects. Zia et al. (2013) realized a method to detect and estimate the pose of objects based on high-quality 3D CAD models. A set of 3D wireframe models is defined manually to derive a shape prior. On vertices of predefined models, a part-detector is trained and detect objects. Using existing CAD models and simulated edges in images, poses of objects can be determined. Pepik et al. (2015) proposed a method of combination of 3D geometric representation with a 2D detector in operation on image, namely the deformable part model (Felzenszwalb et al., 2010). Using this method, 3D CAD information of the objects can help to deliver 3D geometry information additionally to a 2D appearance model. However, the detector has to be trained from different viewing directions. However, these approaches only transform the learned 3D information into images without using 3D information explicitly.

Chen et al. (2015) formulated the object detection as an energy minimization problem. The energy combines an object size prior, ground plane as well as several depth features, point cloud densities and distance to the ground. High-quality 3D object proposals are generated as outputs, a coarse pose for each object is estimated. However, arbitrary types of objects are detected. Ošep et al. (2016) exploited a method with 3D information obtained from stereo images and some prior information of objects to detect different objects. They use stereo images to extract the ground plane and project all 3D points onto the ground plane. The objects in street scenes can be regarded as clusters on a ground plane density map. As output, it delivers a 3D bounding box and a coarse pose estimation for each object. Coenen et al. (2017) adapts the approach of Ošep et al. (2016) to detect vehicles and incorporates geometric assumptions on vehicle inherent properties

into the generic 3D object detection approach. By combining the generic 3D object detections with the deformable part model (Felzenszwalb et al., 2010) detector, visible vehicles can be detected with high-quality.

2.2 Tracking-by-detection

Recently, the problem of tracking objects has been received much attention. Tracking can be regarded as an estimation of the state of one or multiple objects over time (Janai et al., 2017). In complex street scenes, tracking systems face several challenges, such as occlusion, similarity of different objects and different types of traffic participants. Many tracking approaches follow a tracking-by-detection strategy. Using this strategy the specific objects are detected in each frame individually, and afterwards, corresponding detections from different frames are associated to form trajectories (Ošep et al., 2016). Trajectory estimation is formulated as an energy minimization problem or using recursive filters. For the purpose of robustness, the tracking-by-detection approaches combine different complementary cues as data association to determine the corresponding detections from different frames.

Several tracking-by-detection approaches ((Yoon et al., 2015), (Leibe et al., 2008), (Ošep et al., 2016)) use recursive filters to estimate trajectories. A set of features as appearance model and locations of detections in trajectories are considered in tracking process. Leibe et al. (2008) used a non-Markov hypothesis selection framework to solve the tracking problem. This method restricts the possible locations by using Extended Kalman Filter. These locations and color histogram information are formulated in data association. Yoon et al. (2015) used Kalman filter to predict and update state vectors of targets and combine a motion context for multiple objects. Tracking is formulated in a Bayesian filter using this motion context, additionally size and color information as appearance model. Ošep et al. (2016) used Kalman filter to generate trajectory hypotheses and use color histogram information as appearance model. Using an energy optimization approach consisting both terms, the corresponding detections can be tracked.

Several approaches have been proposed to track detections and estimate trajectories

simultaneously as an energy minimisation problem, such as ((Andriyenko and Schindler, 2011), (Shi et al., 2014), (Milan et al., 2013), (Choi, 2015), (Engelmann et al., 2017)). Andriyenko and Schindler (2011) formulated the trajectories estimation as a global energy optimization problem with dynamic model, target persistence and mutual exclusion, which delivers better estimated trajectories than using Extended Kalman Filter. Multiple targets can be tracked using the energy function with models in trajectory estimation and an additional appearance model using the HOG features. Shi et al. (2014) established an energy function consisting of an observation model, an appearance model and a trajectory persistence model to estimate each trajectory and track each detection. Continuous and smooth trajectories are derived using continuous energy minimization. Milan et al. (2013) proposed a mixed discrete-continuous conditional random field model which takes into account constraints in appearance model and trajectory estimation. Each detection is assigned to one target using geometric shape as appearance model while in the trajectory estimation avoids co-occurrence of trajectories. Xiang et al. (2015) used Markov decision processes to solve the tracking problem. In this approach, the Markov decision process uses reinforcement learning in the ground truth trajectories, represented by a similarity function for data association. Choi (2015) proposed a tracking as a global data association problem. The Aggregated Local Flow Descriptor encodes relative motion patterns, which represents trajectories of interest points from detections and these interest points can be associated as tracks in this approach. Engelmann et al. (2017) used 3D shape and motion priors to regularize the trajectory estimation and track vehicles using an energy optimization approach.

Unlike other approaches, Zhang et al. (2008) interpreted each object trajectory hypothesis by connecting each detections as a flow path and solve tracking using a min-cost flow network. The color histogram information is used as data association and an occlusion model is combined into network to handle long-term object occlusions. In order to solve expensive computation as in (Zhang et al., 2008), Lenz et al. (2015) proposed a dynamic min-cost flow solution to adapt computational and memory boundaries.

2.3 Vehicle 3D modelling and pose estimation

Due to different shapes of vehicles, a 3D deformable model is useful to cope with the intra-class variation of vehicles.

Active shape models (ASM) (Cootes et al., 2000) are applied frequently, such as ((Zia et al., 2013), (Zia et al., 2015), (Menze et al., 2015), Lin et al. (2014), Xiao et al. (2016)). The active shape models are able to represent a large variety of shapes of a class of objects based on applying principal component analysis to a set of training examples. Zia et al. (2013) determined a set of vertices from 3D CAD models for vehicles manually and used ASM to represent vehicles. By variation of a few shape parameters, different vehicles can be modelled precisely by matching to detected parts of objects. Menze et al. (2015) used 3D active shape models for reconstructing detected vehicles obtained from stereo image pairs and object scene flow estimation. The inference of this fitting procedure used a conditional random field. However, object detection with scene flow and shape reconstruction cost a lot of time. Lin et al. (2014) used the deformable part model (Felzenszwalb et al., 2010) to detect each part of an object and estimate landmarks in image. Through fitting predefined 3D active shape model to landmarks, shape parameters of an object can be refined. Xiao et al. (2016) also used 3D active shape models to reconstruct detected vehicles. They tried to seek an optimal model to 3D object points through a set of model variations.

Besides active shape models, there are several models to reconstruct the shape of objects. Yingze Bao et al. (2013) defined a prior comprised of a mean shape and a set of weighted anchor points from 3D scans and images of objects from various viewpoints. By matching anchor points to objects in images, shape of an object is modelled as warped mean model in prior. However, it needs a large amount of training examples from different viewpoints. Engelmann et al. (2016) transformed 3D CAD vehicle models into volumetric Truncated Signed Distance Function (TSDF) grids. The TSDF of one 3D point is the the truncated signed distance of this point to the object surface. However, predefined voxel-grid-size limits the level of detail for reconstructing using TSDF.

Instead of being constrained by a certain number of viewpoint classes and imprecise pose parameters calculated directly from detections, more fine-grained pose in 3D space

estimations are able to derived throughout 3D modelling (Coenen et al., 2017).

Zia et al. (2013) derived a pose estimation by reconstructing shapes of objects. Pose parameters are included in one object recognition hypothesis. Optimal recognition hypothesis is determined by matching these hypotheses to detected parts of an object in image. However, this approach depends on quality of pose initialisations. Engelmann et al. (2016) generated a set of models with different pose parameters. Through minimizing the distances of 3D points to the surface of models, pose parameters is able to be optimized. Engelmann et al. (2017) used refined trajectories generated by tracking to provide initial pose parameters for models. Pose parameters are estimated with the same method. Xiao et al. (2016) generated a set of models with different poses and determine the optimal pose parameters through fitting the models to 3D points. Coenen et al. (2017) estimated pose parameters by fitting models for detected vehicles. Pose parameters are optimized by iteratively translating and rotating models to fit the 3D points. In contrast to Xiao et al. (2016), the vehicles stand on the extracted ground plane, which makes the translating of models on a 2D plane.

2.4 Discussion

In this thesis, we aim to track vehicles, model and estimate pose parameters in 3D space using stereo image sequences, which are acquired by static cameras.

This master thesis builds on the vehicle detection approach of Coenen et al. (2017). Unlike the approaches using learned 3D geometric model to detect objects in image, as (Zia et al., 2013, Pepik et al., 2015), this approach uses 3D information explicitly. Inspired by Ošep et al. (2016), the ground plane is extracted using 3D information and a generic 3D object detector is applied to generate object hypotheses. In order to remove the false detections, the deformable part model (Felzenszwalb et al., 2010) is used for vehicle hypothesis verification. In a stereo image sequence, acquired by static cameras, each detection in each frame is acquired in the same 3D coordinate system.

In this thesis, based on the detections in 3D space, a tracking-by-detection strategy is implemented to associate the corresponding vehicles from different frames. Approaches,

such as ((Andriyenko and Schindler, 2011), (Shi et al., 2014), (Milan et al., 2013)), solve tracking-by-detections and trajectory estimations simultaneously as an energy minimisation problem. However, it needs complex models to describe different motion situations to recover trajectories and use a set of features to represent similarity of detections as appearance models for tracking. For instance, Engelmann et al. (2017) needs three different models to describe the motion of objects. Andriyenko and Schindler (2011) used complex dynamic model and target persistence models to recover trajectories and additional appearance model for tacking. Zhang et al. (2008) used min-cost flow network to solve trajectory estimation and track detections, which regards each connection of objects from different frames as a possible trajectory and needs expensive computations. Compared to these methods, trajectories estimation using recursive filters is much easier to implemented, as ((Yoon et al., 2015), (Ošep et al., 2016)). Meanwhile, we make usage of the extracted ground plane to reduce computation. Based on possible locations defined by trajectory estimations and selected features as appearance model, the tracking problem becomes an energy minimization problem, similar to Ošep et al. (2016).

As in Chapter 2.3, the active shape models are applied widely to cope with the intra-class variation of vehicles than other models as, e.g. in Engelmann et al. (2016) who used volumetric Truncated Signed Distance Function (TSDF) grids. In this thesis, we follow the active shape models defined by Coenen et al. (2017). Zia et al. (2013) and Lin et al. (2014) match models to detected parts in image and depends on good pose initialisation. In this thesis, we fit the models directly to the 3D points, similar to Xiao et al. (2016). A set of models with different shapes are generated and through minimize the distance of 3D points to the model surfaces. Meanwhile, fine-grained poses in 3D space is able to be estimated by 3D shape reconstruction, similar to Coenen et al. (2017). In contrast to Coenen et al. (2017), we do not fit models into the single detections individually, but into vehicle tracks, i.e. into several vehicles simultaneously. By using a motion model, this leads to a reduction the number of required parameters. Variations of pose parameters are combined into generated models and pose parameters is able to be optimized with shape parameters simultaneously. In this thesis, we make usage of ground plane and tracked vehicles. Shape and pose parameters are able to be estimated with fewer number

of parameters, compared to Coenen et al. (2017).

Chapter 3

Theoretical background

This chapter presents the theoretical basics of the methods used in this thesis. The basic theories about 3D reconstruction will be introduced in Chapter 3.1, because 3D information is fundamental for the whole approach. Chapter 3.2 describes background about the quick shift clustering and the theory about the deformable part model is introduced in Chapter 3.3. The following Chapter 3.4 gives the fundamentals of the linear Kalman filter strategy. Chapter 3.5 discusses the active shape models.

3.1 3D reconstruction from stereo image pair

A single image can only provide 2D image information without 3D information about the original objects. If two rays from two images acquired from different viewpoints have an intersection in a point of an object, the 3D object point can be reconstructed. This is possible using stereo cameras, which consist of two cameras capturing the same scene simultaneously and are settled parallel in viewing direction within a certain baseline from each other. The baseline is the distance of the projection centers of two cameras.

In order to reconstruct 3D object points from stereo image pairs, stereo rectification and image triangulation are fundamental. In this thesis, we call the images in one stereo image pair as the left image and the right image.

3.1.1 Stereo rectification

The epipolar images are two images with exactly parallel image coordinate systems. Original stereo image pairs are not epipolar images and the corresponding epipolar images can be generated by stereo rectification. The rectification is to project two images onto a common plane. Figure 3.1 shows image planes and image coordinates before and after the rectification.

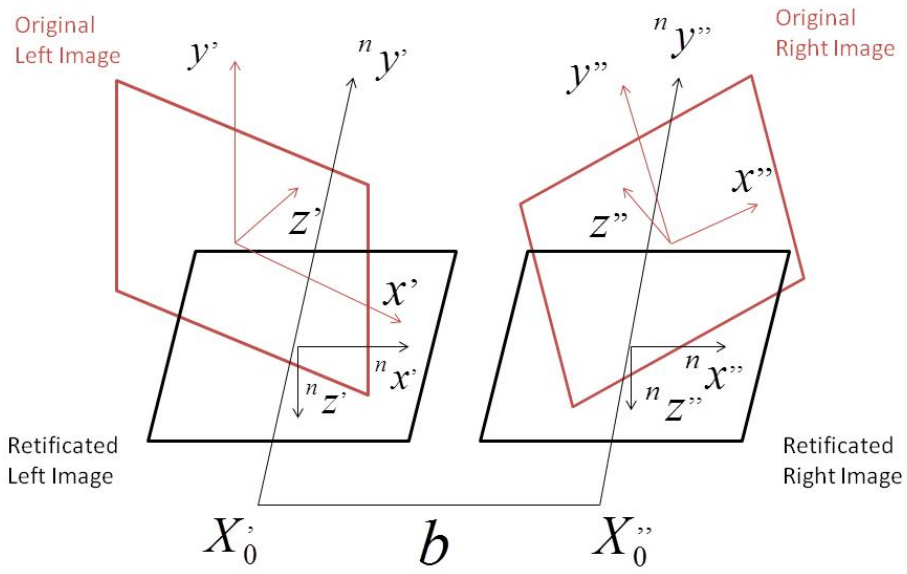


Figure 3.1: Rectified image planes (black) and original image planes (red)

In original images, the x' and y' axes consists the image coordinates of the left image and the x'' and y'' axes consists the image coordinates of the right image, where the z' and z'' axes are oriented in viewing direction for two cameras. After rectification, the ${}^n y'$ and ${}^n y''$ axes in the image coordinates of the left and the right images are parallel orthogonal to the baseline b, that b is the connection of the projection centers $\boldsymbol{X}_0'$ and $\boldsymbol{X}_0''$ for two images. The ${}^n x'$ and ${}^n x''$ axes in the image coordinates of the left and the right images are coincided. The ${}^n z'$ axis in the left rectified image is orthogonal to the ${}^n x'$ and ${}^n y'$ axes, the same definition as the ${}^n z''$ axis in the right rectified image.

Stereo rectification can be implemented by defining homographies H' and H'' for both

images (Hartley and Zisserman, 2000).

$$\begin{aligned} \boldsymbol{x}_i^{'} &= H^{'} \cdot {}^{n}\boldsymbol{x}_i^{'} \\ \boldsymbol{x}_i^{''} &= H^{''} \cdot {}^{n}\boldsymbol{x}_i^{''} \end{aligned} \tag{3.1}$$

In Eq. 3.1, $\boldsymbol{x}_i^{'}$ is one image point in the original left image and ${}^{n}\boldsymbol{x}_i^{'}$ is the corresponding image point in the rectified left image. $\boldsymbol{x}_i^{''}$ and ${}^{n}\boldsymbol{x}_i^{''}$ in the right image are defined as the same as the left image. In order to define homography, the known calibration matrix $K^{'}$ and $K^{''}$ and rotation matrix of the exterior orientations $R^{'}$ and $R^{''}$ of original stereo cameras are prerequisite. The calibration matrix compromise the interior orientation parameters, which are image coordinates of the principal point, camera constant, skewness and scale of the y axis of the image coordinate.

The homography $H^{'}$ of the left image can be defined

$$H^{'} = K^{'} \cdot R^{'T} \cdot R_{rect} \cdot K_{rect}^{-1} \tag{3.2}$$

In Eq. 3.2, $K^{'}$ is the calibration matrix and $R^{'}$ is the rotation matrix of the original left image. K_{rect} is defined for the image coordinates centred at the principal point after rectification. R_{rect} is defined to rotate the original left image to position of the left rectified image.

$$\begin{aligned} &K_{rect} = diag(-c, -c, 0) \\ &R_{rect} = (r_1^T, r_2^T, r_3^T) \\ &where \quad r_1 = \frac{b}{\parallel b \parallel}, \quad r_2 = \frac{b \times d}{\parallel b \times d \parallel}, \\ &d = \sqrt{r_3^{'} + r_3^{''}}, \quad r_3 = r_1 \times r_2 \end{aligned} \tag{3.3}$$

In Eq. 3.3, r_1 is to make the ${}^{n}x^{'}$ axis after rectification along the direction of the baseline. r_2 rotates ${}^{n}y^{'}$ axis after rectification orthogonal to the ${}^{n}x^{'}$ axis and makes the original $z^{'}$ axis along the original viewing directions. r_3 rotates ${}^{n}z^{'}$ axis after rectification orthogonal to the baseline and the ${}^{n}y^{'}$ axis. $r_3^{'}$ and $r_3^{''}$ are the third column of the rotation matrix of the left and right image.

The homography H'' of the right image is defined using the same method as the left image. By determination of homorgraphies H' and H'' for the left and right images, image points in original images can be projected to the rectified images, using Eq. 3.1.

3.1.2 Image triangulation

After stereo rectification, the rectified images in one stereo image pair become epipolar images. The corresponding points locate on the epipolar lines parallel to the ${}^{n}x'$ and ${}^{n}x''$ axes of two images and have identical coordinates in the ${}^{n}y'$ and ${}^{n}y''$ axes. The search for corresponding points in the epipolar images reduces into 1-D space by determination of disparity.

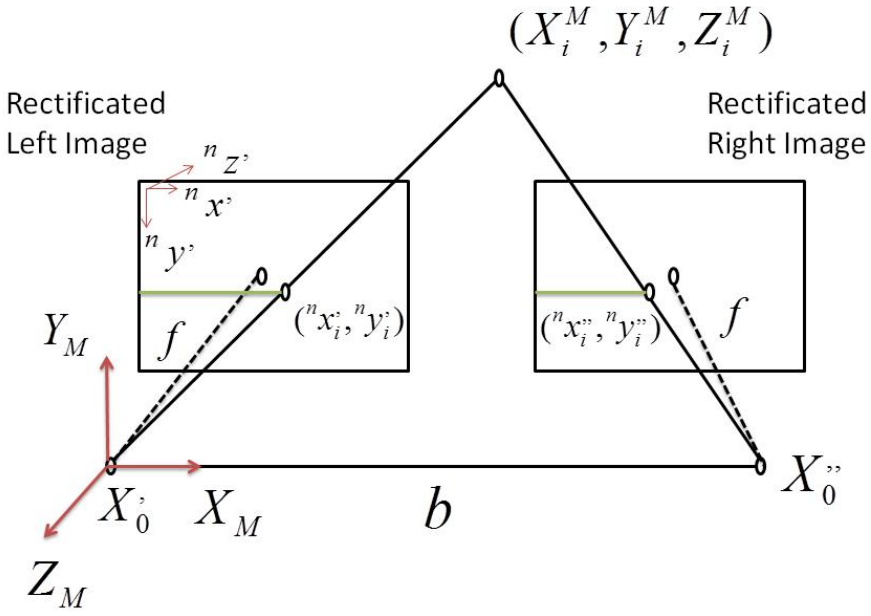


Figure 3.2: Geometry of images after stereo rectification

As in Figure 3.2, $({}^{n}x_i', {}^{n}y_i')$ in the left image and $({}^{n}x_i'', {}^{n}y_i'')$ in right image are two corresponding points with ${}^{n}y_i' = {}^{n}y_i''$. The disparity d_i is defined as shift of these two points in ${}^{n}x$ axis.

$$d_i = ({}^{n}x_i'' - {}^{n}x_i') \tag{3.4}$$

These two image points intersect in a 3D object point $\boldsymbol{X}_i^M$. The coordinate of this 3D object point in the model coordinate system (X_i^M, Y_i^M, Z_i^M) can be defined through triangulation for image rays, with f as focal length of two cameras and b as length of baseline. As in Figure 3.2, the baseline is parallel to ${}^n x$ axis of images after rectification.

$$X_i^M = \frac{{}^n x_i^{,} * b}{d_i}, Y = -\frac{{}^n y_i^{,} * b}{d_i}, Z_i^M = -\frac{f * b}{d_i} \tag{3.5}$$

The 3D object point $\boldsymbol{X}_i^M$ is defined in a model coordinate system (X^M, Y^M, Z^M), as in Figure 3.2. The origin of the model coordinate system locates at the projection center $\boldsymbol{X}_0^{,}$ of the left image, with X^M axis parallel to ${}^n x^{,}$ axis in the image coordinate and Y^M axis as negative direction of ${}^n y^{,}$ axis in the image coordinate of the left image. The Z^M axis is the negative view direction from projection center of the left image to the 3D object point.

The coordinate of 3D object point $\boldsymbol{X}_i^M$ in the model coordinate can transform into the global coordinate ${}^G\boldsymbol{X}_i$ given the exterior orientation parameters, which compromise the coordinate of the projection center ${}^G\boldsymbol{X}_0$ in the global coordinate system $({}^G X_0^{,}, {}^G Y_0^{,}, {}^G Z_0^{,})$ and the rotation matrix of the left image $R^{,}$, by using

$$\begin{bmatrix} {}^G X_i \\ {}^G Y_i \\ {}^G Z_i \end{bmatrix} = \begin{bmatrix} {}^G X_0^{,} \\ {}^G Y_0^{,} \\ {}^G Z_0^{,} \end{bmatrix} + R^{,} \cdot M \begin{bmatrix} X_i^M \\ Y_i^M \\ Z_i^M \end{bmatrix} \tag{3.6}$$

In Eq. 3.6, M is the matrix to transform the axes in the model coordinate to the axes in the global coordinate.

3.1.3 Efficient large-scale stereo matching

A dense matching approach can be used to determine the corresponding points in stereo image pairs and determine disparities. In this thesis, Efficient large-scale stereo matching (ELAS) (Geiger et al., 2010) is used as the dense matching approach. ELAS is a fast Bayesian approach to implement stereo matching without global optimization.

In image matching, plenty of stereo correspondences are ambiguous. However, at the

same time, some pixels can be matched robustly, which can be regarded as support points. These support points can provide prior information about disparities for those ambiguous stereo correspondences. Each support point s_m can provide information about its image coordinate (u_m, v_m) and disparity d_m, i.e. $s_m = (u_m, v_m, d_m)^T$, and $S = \{s_1, \ldots, s_m\}$ is a set of support points. These support points can be used to interpolate disparities for other points throughout Delaunay triangulation, as sampling process shown in Figure 3.3.

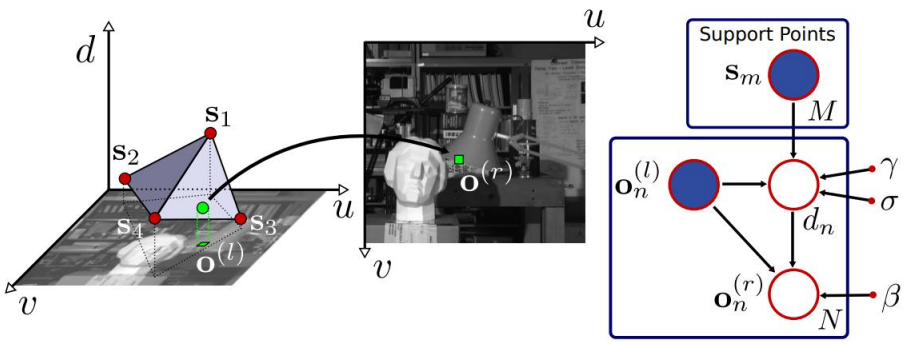


Figure 3.3: Sampling process (Geiger et al., 2010)

The interpolated disparity d_n for a point x_n can be used as prior $p(d_n|S, o_n^{(l)})$ in the disparity calculation, where $o_n^{(l)} = (u_n, v_n, f_n)^T$ is this point in the left image. (u_n, v_n) are image coordinates and f_n is a feature vector. Given the feature vector and the interpolated disparity in the left image, it can draw samples in right image using $p(o_n^{(r)}|o_n^{(l)}, d_n)$ and be regarded as likelihood, where $o_n^{(r)}$ is the point in the right image. At inference, the disparities can be computed by maximum a-posterior estimation (Geiger et al., 2010)

$$d_n^* = argmax \quad p(d_n|o_n^{(l)} o_1^{(r)}, \ldots o_N^{(r)}, S) \tag{3.7}$$

$o_1^{(r)}, \dots o_N^{(r)}$ are all observation in the right image located on the epipolar line of $o_n^{(l)}$.

The posterior can be factorized as

$$p(d_n|o_n^{(l)}o_1^{(r)}, \dots o_N^{(r)}, S) \propto p(d_n|S, o_n^{(l)})p(o_1^{(r)}, \dots o_N^{(r)}|o_n^{(l)}, d_n) \tag{3.8}$$

The distribution of all the observations along the epipolar line in the right image are modeled as

$$p(o_1^{(r)}, \dots o_N^{(r)}|o_n^{(l)}, d_n) \propto \sum_{i=1}^{N} p(o_i^{(r)}|o_n^{(l)}, d_n) \tag{3.9}$$

Eq.3.7 is optimized by each $p(o_n^{(r)}|o_n^{(l)}, d_n)$ independently.

The result of ELAS obtains the dense disparity map of the right image based on the left image as reference. In order to eliminate the inaccurate disparities in occluded and homogeneous regions, the approach is applied to both images practically to perform a left/right consistency check (Geiger et al., 2010).

3.2 Quick-shift clustering

In 3D space, different objects are represented by different clusters of 3D points, and detecting these clusters is a state-of-art method to detect objects Janai et al. (2017). However, calculations in 3D space are quite complex. If we project all 3D object points lower than a certain height onto a plane, all objects lower than that height correspond to different clusters in the 2D plane (Ošep et al., 2016). Thus, 3D object detection transforms into a problem of finding for different clusters in 2D space, and calculation becomes much easier. In this thesis, the Quick-shift (Vedaldi and Soatto, 2008) mode seeking algorithm is used to identify different clusters in this 2D plane.

Given N data points, $x_1, \dots, x_N$ in a 2D space, a mode is determined by computing a probability density function:

$$P(x) = \frac{1}{N} \sum_{i=1}^{N} k(x, x_i), x \in \mathbb{R}^2 \tag{3.10}$$

In Eq.3.10, $k(x)$ can be a Gaussian or other kernel function. In clustering, we search

for the nearest mode of $P(x)$ for each point x_i using gradient ascent. A cluster is formed as all the points which converge to the same mode.

In order to seek the modes, there are plenty of different algorithms. Mean shift is a well-known mode seeking algorithm. It is an iterative method to move each data point uphill towards the mode, approximately following the gradient of the probability density function according to Eq.3.10. In Figure 3.4(left), the black dots represent individual data points and these points move along blue lines. The intensity of the image is proportional to the probability density function $P(x)$ and blue lines follow the direction of gradients of the $P(x)$.

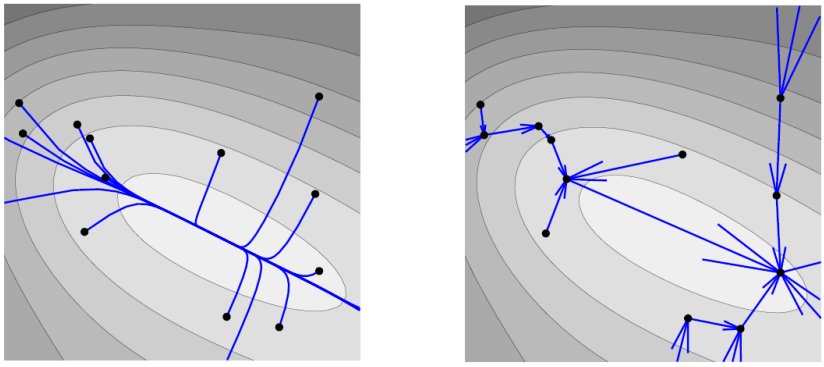


Figure 3.4: Mean shift (Left) and quick shift (right)(Vedaldi and Soatto, 2008)

Quick shift is different from mean shift. In particular, it does not need gradients. It moves each data point x_i to its nearest neighbor x_j, where there is the max local increase of the probability density P_x. In formulas,

$$\begin{aligned} y_i(1) &= argmin\ D(x_i, x_j), \quad j : P_{x_j} > P_{x_i} \\ P_{x_i} &= \frac{1}{N}\sum_{j=1}^{N} k(D(x_i, x_j)), \quad D(x_i, x_j) < \tau \end{aligned} \tag{3.11}$$

In Eq.3.11, $y_i(1)$ is the trajectory where one data point x_i moves to data point x_j. The kernel function $k(D(x_i, x_j))$ measures the distance $D(x_i, x_j)$ of these two points and

compute the probability density function. The distance of two points x_i and x_j must be lower than a predefined threshold τ.

In Figure 3.4(right), the data points (black dots) moves directly to the nearest neighbors with a higher probability density, as directions indicated by the arrows and probability density represented by intensity of the image. All the data points are connected as a tree and modes are recovered by breaking the branches of the tree that are longer than the distance threshold τ.

Quick shift is much simpler and faster than mean shift. The complexity of mean shift is $O(dN^2T)$, however the complexity of quick shift is only $O(dN^2)$, where d dimensionality of data space, N is the number of points and T is the number of iterations. At the same time, based on the predefined distance threshold τ, quick shift can avoid under- or over-fragmentation of modes, which frequently occurs with mean shift(Vedaldi and Soatto, 2008).

3.3 Deformable part model

The deformable part model (Felzenszwalb et al., 2010) is one type of sliding window detector to locate different kinds of objects in 2D images. The basic idea of this model is that an object consists of different parts and train a detector for the entire object and the corresponding parts.

The model consists a coarse root filter to cover an entire object and higher resolution part filters to cover smaller parts of the object. A Histogram of Oriented Gradients (HOG) feature map is computed from each level of an image pyramid and feature maps in different levels consist a feature pyramid. Figure 3.5 illustrates an instance of a model in a feature pyramid. The location of root filter (blue) defines a detection window. The part filters (yellow) are located λ levels down in the pyramid.

Mathematically, a model for an object is defined by $(F_0, P_1, ..., P_n, b)$, where F_0 is a root filter, P_i is the part filter for part i, n is the number of part filters and b is a bias term. Each P_i is represented by (F_i, v_i, d_i), where F_i is the HOG features matrix of the ith part filter, v_i defines the relative position to the root filter as anchor position, and d_i

represents the deformation cost for the part filter to the root filter (Felzenszwalb et al., 2010).

In order to detect objects in the image, an object hypothesis determines the locations of both root P_0 and n part filters P_n in the model in the feature pyramid $z = (P_0, ..., P_n)$. Where $P_i = (x_i, y_i, l_i)$ defines location (x_i, y_i) and level l_i of the each part filter, as shown in Figure 3.6.

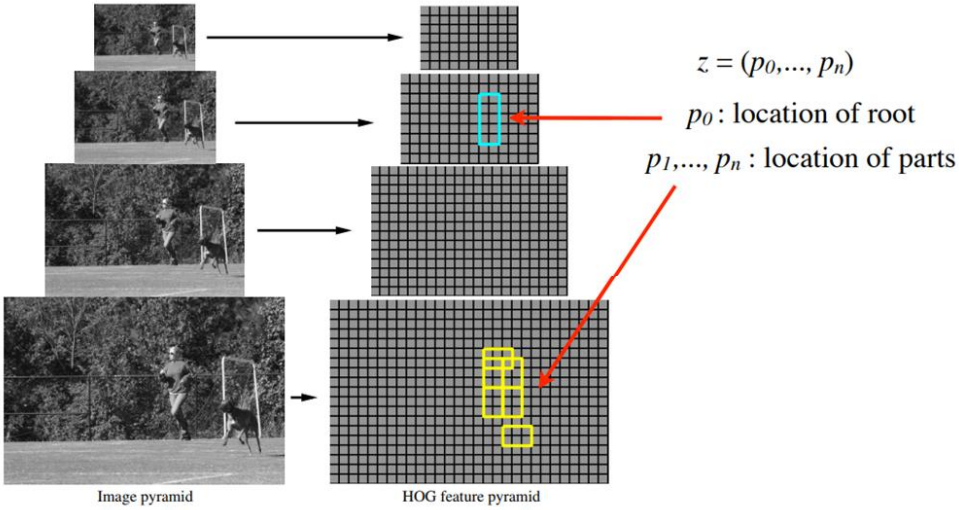


Figure 3.5: Feature pyramid and definition of an object hypothesis (Felzenszwalb et al., 2010)

The score of an object hypotheses is a criterion to detect objects in image. This score is given by the scores of the filters at their locations minus a deformation cost that depends on the relative position of each part with respect to the root filter, plus the bias.

$$score(P_0, ..., P_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, P_i) - \sum_{i=0}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b \tag{3.12}$$

In Eq. 3.12, $F_i \cdot \phi(H, P_i)$ is the dot product of HOG feature matrix for each filter F_i and calculated feature vector at the position and scale of each filter $\phi(H, P_i)$, which represents the scores of each filter at their respective position. d_i is the deformation cost for the part filter to the root filter and $\phi_d(dx_i, dy_i)$ is the displacement of each part

filter relative to anchor position defined by v_i. The dot product of these two terms $d_i \cdot \phi_d(dx_i, dy_i)$ determines the deformation cost which depends on the relative position for each part filter to the root filter, as spatial prior. b is the corresponding bias term of each model. To detect an object in image, an overall score based on best placement of parts is computed for each possible root location. High scoring root locations and corresponding part filters define detections of the objects and predict a bounding box for each object with the size of root filter.

3.4 Linear Kalman filter

The Kalman Filter (Kalman et al., 1960) is used widely in optimal estimation using a linear model, e.g. trajectory estimation. In a 2D spcae, the state vector of one object is described by $X_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)$, which consists of position (x_t, y_t) and its first derivative (velocity $(\dot{x}_t, \dot{y}_t)$). In the state vector, t represents the current epoch.

The Kalman Filter works in a two-step process, prediction and correction. In prediction, a linear motion model is used to generate prediction for the state vector of epoch t from the state vector of previous time to current time.

$$\bar{X}_t = A\hat{X}_{t-1} \tag{3.13}$$

In Eq. 3.13 $\bar{X}_t$ is the predicted state vector of epoch t and $\hat{X}_{t-1}$ is the corrected state vector of epoch $t-1$. A is the linear state transition matrix to describe the motion model.

The motion model is

$$\begin{aligned} \bar{x}_t &= \hat{x}_{t-1} + \hat{\dot{x}}_{t-1} \cdot \Delta t \\ \bar{y}_t &= \hat{y}_{t-1} + \hat{\dot{y}}_{t-1} \cdot \Delta t \end{aligned} \tag{3.14}$$

where Δt is the time step between the epochs t and $t-1$.

The covariance $\bar{P}_t$ of the predicted state vector $\bar{X}_t$ in current time depends on the covariance $\hat{P}_{t-1}$ of the previous corrected state vector $\hat{X}_{t-1}$ and an additional covariance

matrix of system noise Q_t.

$$\bar{P}_t = A\hat{P}_{t-1}A^T + Q_t \tag{3.15}$$

In Eq. 3.15, the Q_t is the covariance matrix of system noise, which includes the noise of the velocity $(\sigma_{\dot{x}_t}, \sigma_{\dot{y}_t})$.

The correction step in the Kalmnan Filter uses a linear measurement model to correct the predicted state and estimate optimal the current state vector based on prediction and the current measurements. The measurement model is

$$\begin{aligned} x_t^M &= \bar{x}_t + \sigma_x \\ y_t^M &= \bar{y}_t + \sigma_y \\ \dot{x}_t^M &= \bar{\dot{x}}_t + \sigma_{\dot{x}_t} \\ \dot{y}_t^M &= \bar{\dot{y}}_t + \sigma_{\dot{y}_t} \end{aligned} \tag{3.16}$$

where $(\sigma_x, \sigma_y, \sigma_{\dot{x}_t}, \sigma_{\dot{y}_t})$ represents the measurement noise, (x_t^M, y_t^M) is the current measured position and $\dot{x}_t^M, \dot{y}_t^M)$ is the current measured velocity.

In this process, the Kalman gain K_t is an important parameter, which is calculated from the covariance matrix of the predicted state.

$$K_t = \bar{P}_t C_t^T (R_t + C_t \bar{P}_t C_t^T)^{-1} \tag{3.17}$$

In Eq. 3.17, R_t is the covariance matrix of the measurement noise and C_t is the output matrix to describe the measurement model. The covariance $\hat{P}_t$ of the current corrected state vector can be calculate based on the Kalman gain K_t and the covariance $\bar{P}_t$ of predicated state vector.

$$\hat{P}_t = (I - K_t C_t)\bar{P}_t \tag{3.18}$$

where I is an identity matrix.

The optimal state vector $\hat{X}_t$ is determined based on Kalman gain K_t and predicted

state vector $\bar{X}_t$ of the current epoch.

$$\hat{X}_t = \bar{X}_t + K_t(X_t - C_t\bar{X}_t) \tag{3.19}$$

In Eq. 3.19, $X_t = (x_t^M, y_t^M, \dot{x}_t, \dot{y}_t)$ is the measured state vector of current epoch and $\hat{X}_t$ is the corrected state vector of current epoch. The estimated state vector $\hat{X}_t$ can be used to predict the state vector $\bar{X}_{t+1}$ for further epoch and form the optimal trajectories for each object.

If the state transition and measurement models require non-linear forms, the Extended Kalman Filter is able to deliver the optimal estimations.

3.5 Active Shape Model

In order to deal with the large variations of vehicle shapes, 3D active shape models as a deformable 3D model learned from a set of CAD vehicle models is applied widely. Active shape models were firstly introduced by Cootes et al. (1995) and extended to 3D using sets of ordered vertex points in 3D space to represent objects, such as vehicles, bicycles and faces. These models can be deformed iteratively to fit to the objects having different shapes detected in 3D space.

As in (Zia et al., 2013), a certain number of annotated vertices is defined for a set of CAD training vehicle models manually, residing in 3D space. This set of training vehicles includes different types of vehicles, such as compact car, sports-car, SUV and so on. These vertices are chosen from 3D CAD models, usually located at wheels, corner of windows, doors and so on, to cover the entire vehicle. The definition and training set in this thesis refer to Chapter 4.5.1.

The mean values for all vertex positions are calculated as mean model of all input training models. Directions of most dominant deformations can be determined by applying a principal component analysis of covariance matrix of vertices. Active shape models of any vehicles can be obtained through a linear combination with few parameters based on the mean model and eigenvalues λ_i of the eigenvectors e_i (Coenen et al., 2017).

$$v\left(\gamma_k\right) = m + \sum_i \gamma^{(i)} \lambda_i e_i \tag{3.20}$$

In Eq. 3.20 m is the mean model and e_i is the i principal component eigenvector of the covariance matrix of the vertices. The eigenvectors e_i are weighted by their corresponding eigenvalues λ_i and scaled by shape parameters $\gamma_k^{(i)}$, where i is number of eigenvalues, eigenvectors and shape parameters.

In this thesis, the number of eigenvectors and eigenvalues in the active shape model is defined as $i \in \{1, 2\}$, for a balance of complexity and quality of the model. Figure 3.6 shows active shape models as representations for different vehicles. These vehicles are represented by 40 vertices and only using 2 shape parameters, i.e. $(-1, -1)$, $(1, -1)$, $(-0.5, -0.5)$, $(-0.3, -0.8)$ and mean model as $(0, 0)$.

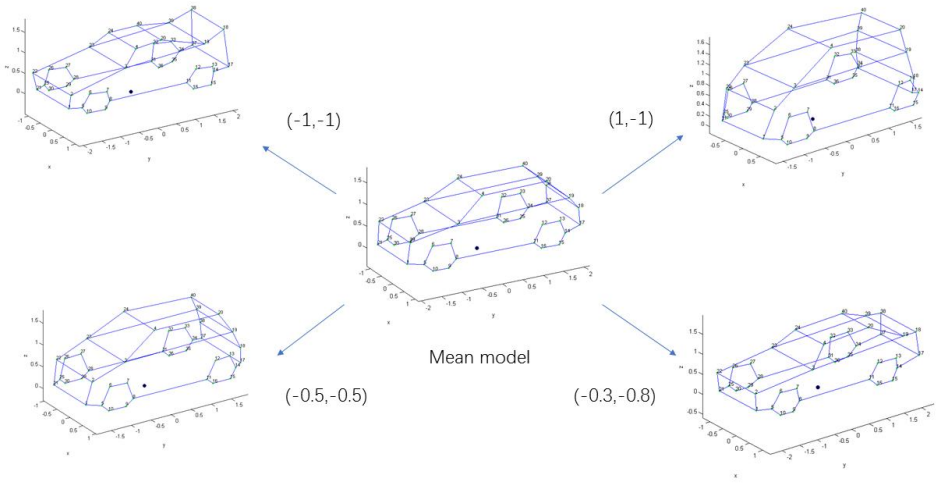


Figure 3.6: Active shape models for representing vehicles by variation of two shape parameters

Chapter 4

Methodology

The purpose of this thesis is to detect and track vehicles from stereo image sequences and to derive pose and shape parameters for each vehicle. This approach builds on several methods for 3D reconstruction from images, ground plane extraction, object detection, tracking-by-detection and 3D modelling for objects.

In this thesis, the stereo image sequences acquired by static stereo cameras are used as data source. The observing vehicle equipped multiple sensors has no ego motion, and as a consequence, model coordinate systems for all stereo image pairs in a sequence are identical. In the future, visual odometry e.g. (Kitt et al., 2010) can be introduced to take into account the ego movement of sensors platform and stereo cameras. The stereo cameras are calibrated so that interior and relative orientation parameters of the two cameras are known for all epochs.

The basic work flow of this approach starts with 3D reconstruction from stereo image pairs and detecting vehicles frame by frame. Based on detections, we try to track vehicles over two continuous frames to determine each vehicle track, which means that two vehicles from detections are associated to form one track. In the end, 3D modelling and pose estimation is implemented for each vehicle track.

This thesis builds on the approach of Coenen et al. (2017) and tries to solve the following problems. On the one hand, there are initialization problems for vehicle orientation in (Coenen et al., 2017), which only use information about the bounding box in a single frame to determine orientation for vehicle heading. In our approach, the prior knowl-

edge about movement direction for each vehicle is used as initialization for orientation, which is closer to the orientation direction of vehicle. Furthermore, as the 3D modelling method introduced by (Coenen et al., 2017) needs 5 parameters for each detected vehicle, it requires 10 parameters for modelling the same vehicle in two time steps. However, in this approach, it only needs 7 parameters for each vehicle track with the help of motion model.

Figure 4.1 shows the whole framework. In the beginning, the 3D object points are generated by stereo image rectification, dense matching and triangulation. After 3D reconstruction as first step, the following procedure is to detect vehicles. Vehicle detection consists of two steps, the generic 3D object detection and the deformable part model as verification in the image. In the vehicle tracking step, based on reliable detected vehicle hypotheses, a tracking-by-detection strategy is implemented and identifies each vehicle track. An energy function is introduced that consists of a data association term and a motion model term. In the last step, an active shape model is fitted to each vehicle track in two time steps simultaneously. At the same time, pose of two vehicles in each vehicle track are optimized. More details of every step will be explained in the subsequent sections.

4.1 Problem statement

Stereo image pairs from static stereo cameras can help to reconstruct a 3D scene. The goal of this thesis is to describe a dynamic street scene, observed by static stereo cameras, by a ground plane and a set of vehicle tracks and shapes described by a few parameters. Given the static stereo image sequences showing a dynamic street scene, all visible vehicles are detected frame by frame firstly. Further more, we try to associate detections of the same vehicle in images from subsequent time steps. For each time step, pose (position and orientation on the ground plane) is estimated and the shape is reconstructed for each vehicle.

In detection, we focus on finding sets of reconstructed 3D object points x_k and corresponding color vector c_k for each point in each stereo pair individually for each vehicle k.

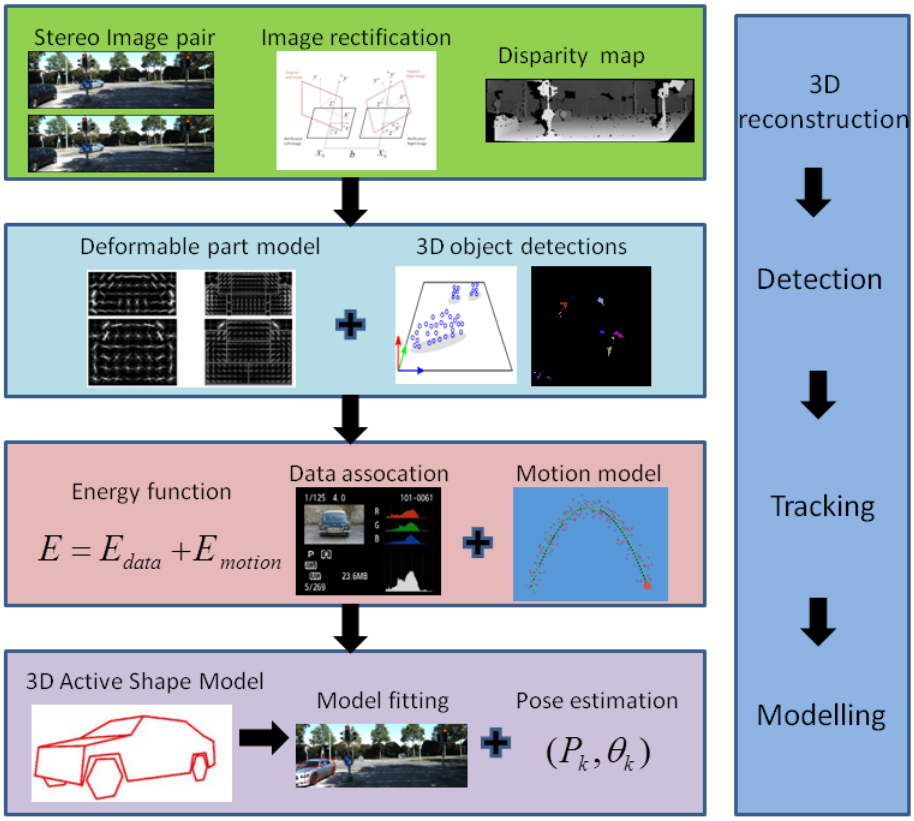


Figure 4.1: Flow-chart of the whole approach

Based on detections, we try to associate the vehicles. The vehicle track is defined as the corresponding vehicles k_{t-1} and k_t in two continuous frames, the previous and current frame. Fitting a 3D deformable model is used as 3D shape reconstruction for each vehicle track, and the vehicle poses (position and orientation on the ground plane) can be estimated at the same time. Each vehicle track can be described by a vector of unknowns $S_{k,t-1} = (p_{t-1}, \theta_{t-1}, \gamma_{t-1}, v_{t-1})$ for the vehicle in the previous frame and another vector of unknowns $S_{k,t} = (p_t, \theta_t, \gamma_t, v_t)$ for the vehicle in the current frame. p_{t-1} and p_t are the positions of vehicles on the ground plane. θ_{t-1} and θ_t are the rotation angles about the axis vertical to the ground plane for each vehicle track, which are regarded as orientations of each vehicle track. v_{t-1} and v_t are the translational velocities of each vehicle track with respect to the ground plane. γ_{t-1} and γ_t are shape parameters to reconstruct the shapes of each vehicle track. As the vehicle shape does not change with time, the shape parameters of each vehicle track are identical. The fitting procedure is done for each vehicle track simultaneously by estimating the vector of unknowns $S_{k,t}$. By making

use of the motion model, the vector of unknowns $S_{k,t}$ results directly from the estimated vector $S_{k,t-1}$.

4.2 3D reconstruction

In the beginning, individual stereo image pairs and corresponding interior and relative orientation parameters are inputed. Stereo image pairs can be rectified based on interior parameters and relative orientation parameters of two cameras from calibration. In each rectified image pair, the left image is defined as reference image. The dense disparity map can be generated for every rectified image pair using the ELAS matcher (Geiger et al., 2010) as described in Chapter 3.1.2.

The accuracy of depth value σ_Z of one 3D point is able to be determined using error propagation of depth calculation in Eq. 3.5.

$$\sigma_Z = \sqrt{\frac{f^2 * b^2 * \sigma_d^2}{d^4}} = \frac{f * b}{d^2} * \sigma_d \tag{4.1}$$

with the length of baseline b, the focal length f, the disparity value d and the accuracy of disparity σ_d. Eq. 4.1 shows that, the inaccuracy of depth value σ_Z is increasing with lower disparity values d. As a consequence, a maximum acceptable accuracy of depth value $\sigma_{z_{max}}$ can be defined and the minimum valid disparity d_{min} can be calculated by

$$d_{min} = \sqrt{\frac{f * b * \sigma_d^2}{\sigma_{z_{max}}}} \tag{4.2}$$

In Eq. 4.2, the maximum acceptable accuracy of depth value $\sigma_{z_{max}}$ and the accuracy of disparity σ_d are defined by user. For each pixel in the disparity map, only the pixel in the left rectified image with disparity $d > d_{min}$ is regarded as valid 2D image point. By the definition of minimum valid disparity d_{min}, the maximum depth value z_{max} of generated 3D point cloud can be defined using Eq. 3.5. The coordinates of the 3D points in the model coordinate system corresponding to the valid 2D image points are calculated using Eq.3.5. Figure 3.2 shows the definition of the model coordinate system containing these 3D points. The color information of the valid 2D image points in the left rectified image are assigned to the corresponding 3D object points. The generated 3D point clouds

with additional color information are essential for the further procedures.

4.3 Vehicle detection

In this thesis, vehicle detection follows (Coenen et al., 2017). Generic 3D object detection is implemented for generating vehicle hypotheses and these hypotheses are verified by the deformable part model in the left rectified image to achieve high-quality vehicle detections. Stereo image pairs and generated 3D object points clouds are necessary inputs and this detection approach is performed individually for each frame. At the same time, some heuristic assumptions are taken into account as prior knowledge for detections. In the end, a 2D bounding box in image space and a set of 3D object points for each detected vehicle in each frame are delivered.

4.3.1 Vehicle detection assumptions

The stereo cameras are fixed on or near the top of a car with a certain height. When the car does not move, the coordinate system of the image and generated 3D object points are identical in the whole stereo image sequence. Furthermore, we make the assumptions to constrain the vehicle detection procedure:

1. Vehicles are located on the street, which corresponds to the ground plane
2. Vehicles are surrounded by free space
3. Vehicles are always lower than a certain max height h_{max}
4. The area covered by a vehicle on the ground plane is within a known range

4.3.2 Ground plane extraction

Ground plane extraction is essential for the following procedures. The ground plane can be extracted based on 3D points, generated as shown in Chapter 4.2. Only if the camera capture from approximately horizontal direction to the ground plane, the reconstructed 3D objects points belonging to ground plane own low values in the axis vertical to ground

plane. As shown in the coordinate system for reconstructed 3D point cloud in Figure 4.2, the points belonging to ground plane have similar height values vertical to ground plane, which is Y_M axis in the coordinate system. Figure 4.3 shows corresponding stereo image and it is captured from approximately horizontal direction to the ground plane.

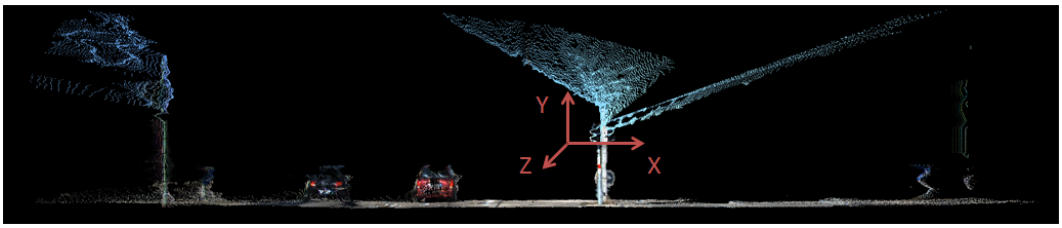


Figure 4.2: Model coordinate system definition in the reconstructed 3D point cloud

Figure 4.3: Corresponding stereo image

In this thesis, the ground plane is detected based on the reconstructed 3D object points. The Random sample consensus (RANSAC) method is applied to determine plane parameters in Hesse normal form. The plane equation for a plane Ω can be formulated as

$$\Omega : ax + by + cz + d = 0 \tag{4.3}$$

where $n = [a, b, c]^T$ is normal vector with $||n|| = 1$ and d is the distance of the plane from the origin. In order to reduce the number of points, the generated 3D points are sorted firstly. The N lowest 3D object points according to their Y_M^N coordinate (which is almost vertical) are used for RANSAC. In each iteration of random sample consensus, 3 candidates points are selected randomly to construct a plane. The parameters of a plane are determined by principal component analysis on the covariance matrix M of the points p_i with $i \in 1, N$

$$M = \sum_{i=1}^{i} (p_i - \bar{p})(p_i - \bar{p})^T \tag{4.4}$$

where $\bar{p}$ is the center of gravity from p_i. The normal vector n can be determined as eigenvector corresponding to the smallest eigenvalue and d can be determined by

$$d = -n^T \bar{p} \tag{4.5}$$

In each iteration, the number of other 3D points located near the generated ground plane inside the threshold d_{thres} are recorded. After a certain times of iterations n_i, the plane with the highest number of 3D points located inside the threshold is determined to be the ground plane.

4.3.3 Generic 3D object detection

In the two-step approach for vehicle detection by Coenen et al. (2017), the first step is to generate vehicle hypotheses in 3D object space using a clustering method.

Based on the ground plane extracted in the way as described in Chapter 4.3.2, the height of each point can be defined as its distance from the ground plane. Based on assumption (3) in Chapter 4.3.1, all the points lower than the predefined max height h_{max} are accepted as possible points, except the points belonging to the ground plane. These 3D points are projected to the ground plane and a ground plane density map of these projected 3D points is computed. A grid with certain width Δ_D is defined in the ground plane to represent the density in a raster map.

According to assumption (2) in Chapter 4.2.1, each vehicle can be represented by a cluster of projected points in the ground plane density map. Quick-shift clustering (Vedaldi and Soatto, 2008) is used for clustering here (cf. Chapter 3.2). After clustering, a vehicle hypothesis is represented by 3D projected points in a cluster across plenty of cells. However, these clusters may not only represent a vehicle, but also objects such as pedestrians, traffic lights and trees. Assumption (4) in Chapter 4.2.1 contributes to remove these false hypotheses. Clusters covering an area smaller than a threshold A_{min} and larger than a another threshold A_{max} on the ground plane, where the area

is defined by minimum bounding rectangle (Shekhar and Xiong, 2007), are regarded as false hypotheses and rejected. The remaining clusters represent vehicle hypotheses for the further process. The corresponding 3D object points and 2D image points are also stored as vehicle hypothesis information in the second detection step.

4.3.4 Verification using the deformable part model

Vehicle hypotheses generated by 3D object detection are not reliable enough. On the one hand, there is still a number of false detections after rejecting plenty of them only by the area assumption, which cannot differentiate vehicles from objects having a similar size. On the other hand, many vehicles are split into multiple clusters, each corresponding to part of the vehicle. This can be caused by occlusion. In the original images, some parts are occluded or cannot be matched, which leads to 3D projected points in ground plane density map clustered into multiple clusters.

To overcome these problems, the deformable part model (Felzenszwalb et al., 2010) is used to verify vehicle hypotheses in 2D images (cf. Chapter 3.3). The deformable part model delivers 2D bounding box BB^{DPM} in a 2D image for each 2D vehicle hypothesis.

Every 3D object hypothesis is composed of a set of 3D points, and the corresponding 2D points. Based on these 2D points, the 2D bounding box in image space is derived as BB^{hyp} . The Jaccard Index (Everingham et al., 2010) corresponds to the intersection over union in image space index for BB^{DPM} and BB^{hyp}.

$$\begin{aligned} J_I\left(BB^{DPM}, BB^{hyp}\right) &= \frac{|BB^{DPM} \cap BB^{hyp}|}{|BB^{DPM} \cup BB^{hyp}|} \\ &= \frac{|BB^{DPM} \cap BB^{hyp}|}{|BB^{DPM}| + |BB^{hyp}| - |BB^{DPM} \cap BB^{hyp}|} \end{aligned} \tag{4.6}$$

In order to increase the accuracy of detections, when $J_I\left(BB^{DPM}, BB^{hyp}\right)$ is larger than a threshold θ_J, a 3D object hypotheses will be kept as a final detection. The threshold θ_J is defined by user.

If several object hypotheses BB^{hyp} have the $J_I\left(BB^{DPM}, BB^{hyp}\right)$ larger than the threshold θ_J to a same 2D vehicle hypothesis BB^{DPM} detected by the deformable part

model, these object hypotheses correspond to one entire vehicle and will be merged.

The final detections after verification and merging are reliable and used in the following tracking and 3D modelling procedures. The results of the detection step consist of a set of 3D object points and the corresponding 2D bounding box in image space for each detected vehicle. Based on the 3D projected points in the ground plane density map corresponding to each detection, the minimum bounding rectangle can be used to define as 2D bounding box in object space. The position of a detection hypothesis is defined as the center of the bounding box. At the same time, the other information (position in image, color information of each points belonging to detection) is stored for further processing.

4.4 Vehicle tracking

The method described in Chapter 4.3 can deliver reliable results in vehicle detection. Here, a tracking method is implemented. Two subsequent stereo pairs and the detected vehicles with additional information as described previously are used as input. In this step, individual vehicles in two continuous time steps are associated.

4.4.1 Kalman filter

In this thesis, the Kalman filter is used to estimated trajectories in this thesis (cf. Chapter 3.4).

According to the assumptions in Chapter 4.3.1, vehicles only move on the ground plane, so that movement of vehicles can be described in 2D space instead of 3D. As described in Chapter 4.3.2, the ground plane has a certain height in y axis of the model coordinate system, so that the 2D ground plane for vehicle moving consists of x and z axes. As a consequence, the position of one vehicle in the current epoch t can be described as (p_t^x, p_t^z), which is the center of the bounding box on the ground plane. (v_t^x, v_t^z) is the translational velocity in x and z axes of the ground plane.

In epoch $t > 0$, the velocity of vehicle in the current epoch can be measured with the relative position of two vehicles in previous and current frame

$$
\begin{aligned}
v_t^x &= (p_t^x - p_{t-1}^x)/\Delta t \\
v_t^z &= (p_t^z - p_{t-1}^z)/\Delta t
\end{aligned}
\tag{4.7}
$$

where Δt is time step of two continuous epoches, which is known from the stereo image pairs acquisition setup. As in Chapter 3.4, the measured state X_t can be described as $X_t = (p_t^x, p_t^z, v_t^x, v_t^z)$. $\bar{X}_t$ is the predicated state vector of the current epoch and $\hat{X}_t$ is the corrected state vector, with the same form as the measured state X_t.

Based on the motion model, described in Chapter 3.4, the transition matrix A in the Kalman filter is defined as

$$
A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}
$$

The output matrix C in the Kalman filter to describe the measurement model is

$$
C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.9}
$$

In implementation, the Kalman filter needs the initialization and determination of a set of parameters. In epoch $t = 0$, if there is any detected vehicles, the initial state vectors compromise the position of these detected vehicles in epoch $t = 0$ and the initial translational velocity (v_0^x, v_0^z) defined by user, additionally, the initial covariance matrix of initial state vector P_0. The noise of translational velocity $(\sigma_{v^x}, \sigma_{v^z})$ and the noise of measured position $(\sigma_{p^x}, \sigma_{p^z}$ are predefined. Consequently, the system noise covariance matrix Q_t and the measurement noise covariance matrix R_t are defined. If there is no detected vehicles in one epoch $t = a$ of one stereo image sequence, the Kalman filter is initialized again in epoch $t = a + 1$ using the same initialization method.

In epoch $t > 0$, the state vectors of each pair of detected vehicles in two continuous frames, including one vehicle in the previous frame V_i^{t-1} and one vehicle in the current frame V_j^t, are estimated using Kalman filter. The Kalman filter can estimate a corrected state vector for each detected vehicle in epoch $t > 0$. A predicted state vector of the vehicle in the current frame $\bar{X}_t$ can be delivered based on the corrected state vector of the vehicle in the previous frame $\hat{X}_{t-1}$ using Eq. 3.13 and it is recorded for tracking process, additionally, the corresponding measure state X_t.

4.4.2 Tracking

Tracking is carried out in object space and usually formulated as an energy minimization problem, where the energy function consist of a data association term and motion term. Often, the combination of different complementary cues is used to improve the robustness of tracking systems (Janai et al., 2017), such as knowledge about geometry, radiometry, topology and semantics of an object.

In this thesis, based on detected vehicles in each frame, the purpose of tracking is to associate the same vehicles in two continuous frames using a set of reasonable cues. According to the assumptions in Chapter 4.3.1, vehicles only move on the ground plane. In this thesis, the vehicles are tracked on the ground plane, in a 2D projection of object space. One detected vehicle from previous frame V_i^{t-1} and one detected vehicle from current frame V_j^t constitute a pair of detected vehicles. Parameters of each pair of detected vehicles are calculated in an energy function

$$E(C_t, C_{t-1}, X_t, \bar{X}_t) = E_{data}\left(C_t, C_{t-1}\right) + E_{motion}(X_t, \bar{X}_t) \tag{4.10}$$

In Eq. 4.10, t is the epoch of the detected vehicle V_j^t and $t-1$ is the epoch of the detected vehicle V_i^{t-1}.

$E_{data}\left(C_t, C_{t-1}\right)$ is the data association term including the feature, namely, the difference of mean color (mean RGB value) for two detected vehicles in previous and current frame. Where C_t is the mean value of RGB values for each 3D point of the detected vehicle from current frame V_j^t and C_{t-1} is the same parameter of the detected vehicle

from previous frame V_i^{t-1}.

In motion term $E_{motion}(X_t, \bar{X}_t)$, X_t is measured state of the detected vehicle from current frame V_j^t and $\bar{X}_t$ is the predicted state vector of V_j^t, estimated from the previous corrected state vector $\hat{X}_{t-1}$ of the vehicle V_i^{t-1} using the Kalman filter (cf. Chapter 4.4.1).

The tracking starts from epoch $t = 1$ and compute the energy function frame by frame. It needs initialization for the Kalman filter in $t = 0$. If there is no detected vehicles in one epoch $t = a$ of one stereo image sequence, the Kalman filter needs initialization in epoch $t = a + 1$ and tracking starts again from epoch $t = a + 2$.

In this approach, the energy function is computed for each pair of detected vehicles, V_i^{t-1} and V_j^t. Only if the energy of one pair of detected vehicles is lower than a threshold E_{thres}, the corresponding vehicle in previous V_i^{t-1} and current frame V_j^t can be regarded as a potential vehicle track (V_p^{t-1}, V_p^t). When more than one vehicle in previous frame may be associated to the same vehicle in the current frame with an energy lower than the threshold E_{thres}, the pair having the minimum energy is selected. The threshold E_{thres} of energy function is selected experimentally.

By analogy, the current frame t becomes the previous frame corresponding to the epoch $t + 1$. The energy function is computed for each pair of detected vehicles in the current frame V_j^t and the epoch V_k^{t+1} and determine the potential vehicle track (V_p^t, V_p^{t+1}) using the same method. If two potential vehicle tracks (V_p^{t-1}, V_p^t) and (V_p^t, V_p^{t+1}) corresponds the same vehicle V_p^t in the epoch t, the corresponding vehicle in the epoch $t - 1$, t and $t + 1$ is associated as potential vehicle tracks.

Data association term

The stereo image pairs are captured with a high frame rate. The illumination, therefore is quite similar for two continuous. Furthermore, vehicles are usually covered mainly by one color, so that we can assume corresponding vehicles show similar mean color (mean RGB value) of all 3D object points in two consecutive frame. So in this approach, low difference of the mean colors of detected vehicles in the previous and the current frames is selected as the feature in data association term, which can be formulated

$$
\begin{aligned}
E_{data}\left(C_t, C_{t-1}\right) &= \frac{||C_t - C_{t-1}||}{256} \\
C_t &= \frac{\sum_{n=1}^{n}(R_n + G_n + B_n)}{n} \\
C_{t-1} &= \frac{\sum_{m=1}^{m}(R_m + G_m + B_m)}{m}
\end{aligned}
\tag{4.11}
$$

In Eq. 4.11, n represents each 3D point belonging to the detected vehicle from previous frame V_i^{t-1} and m represents each 3D point of the detected vehicle from current frame V_j^t. R,G and B represent values in red, green and blue channels. C_t is the mean value of RGB values for each 3D point of the detected vehicle from current frame V_j^t and C_{t-1} is of the detected vehicle from previous frame V_i^{t-1}. The absolute difference of mean color for vehicle in the current and previous frame $||C_t - C_{t-1}||$ is normalised by 256, which is the number of 8-bit grayscale in each RGB channel of a color image.

Based on assumptions described previously, the same vehicles in current and previous frame may be associated, if the data association term $E_{data}\left(C_t, C_{t-1}\right)$ in the energy function is low. Using $E_{data}\left(C_t, C_{t-1}\right)$ term is not sufficient in the situation, where multiple objects of similar appearance are presented. Thus, in order to deal with much more complicated situations, the motion of vehicles has to be modelled and motion model term in energy function is necessary.

Motion term

The motion model term in the energy function is defined as the difference between the measured state X_t and the predicted state vector $\bar{X}_t$ of the detected vehicle in the current frame V_j^t. The predicted state vector $\bar{X}_t$ is estimated from the previous corrected state vector $\hat{X}_{t-1}$ of the detected vehicle in the previous frame V_i^{t-1} using the Kalman Filter (cf. Chapter 4.4.1).

$$
E_{motion}(X_t, \bar{X}_t) = ||X_t - \bar{X}_t|| \tag{4.12}
$$

If two vehicles are the corresponding vehicles in previous and current frame, this motion energy term should be small.

In order to deliver suitable normalised number in the energy function, the motion term in the energy function is defined as norm form in this thesis.

$$E_{motion}(X_t, \bar{X}_t) = ||X_t - \bar{X}_t||^2_\Sigma \tag{4.13}$$

where the norm of $||X_t - \bar{X}_t||^2_\Sigma$ is defined similar to Engelmann et al. (2017):

$$||X_t - \bar{X}_t||^2_\Sigma = ||X_t - \bar{X}_t||^T \Sigma^{-1} ||X_t - \bar{X}_t|| \tag{4.14}$$

The covariance matrix Σ in Eq. 4.14 is approximated using first-order error propagation of velocity noise through the motion model and adding a constant factor on covariances for translational velocity.

$$\begin{aligned} \Sigma &= J \Sigma_{(v^x, v^z)} J^T \\ \Sigma_{(v^x, v^z)} &= \begin{bmatrix} \sigma^2_{v^x} & 0 \\ 0 & \sigma^2_{v^z} \end{bmatrix} \end{aligned} \tag{4.15}$$

In Eq. 4.15, $\Sigma_{(v^x, v^z)}$ is the covariance matrix of translational velocity. J is Jacobian matrix $\nabla_{(v^x, v^z)} g|_{(v^x, v^z)}$ evaluated the state vector X_t at (v^x, v^z)

$$J = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.16}$$

The motion term in the energy function is essential for determination of high quality vehicle tracks.

4.5 3D modelling and pose estimation

The purpose of 3D modelling is to determine a specific 3D representation for each tracking vehicle. Due to the different shapes for each vehicle, a 3D deformable model is necessary

for modelling. In this approach, the active shape model as in Zia et al. (2013) is applied (cf. Chapter 3.5). At the same time, pose parameters belonging to each tracked vehicle can be determined by a model fitting process. In the end, 3D model and precise pose (position and orientation) are delivered as outputs for each tracked vehicle.

4.5.1 Active shape model

In this approach, 40 vertices are chosen to represent a vehicle. These vertices are distributed on the wheels, corners and windows. Figure 4.4 shows the vertex distribution for a vehicles.

Figure 4.4: Vertices definition for an active shape model

The first step of generating different active shape models is to determine the mean model of vehicle. In this approach, a set of different 3D CAD models for different types of vehicles (sports-car, compact car, estate-car, SUV and sedan), seen in Figure 4.5, are used to define mean model, which consists of the mean values for all vertices positions. The mean model of vehicle is shown in Figure 4.6.

Applying principal component analysis directions of deformations can be define to

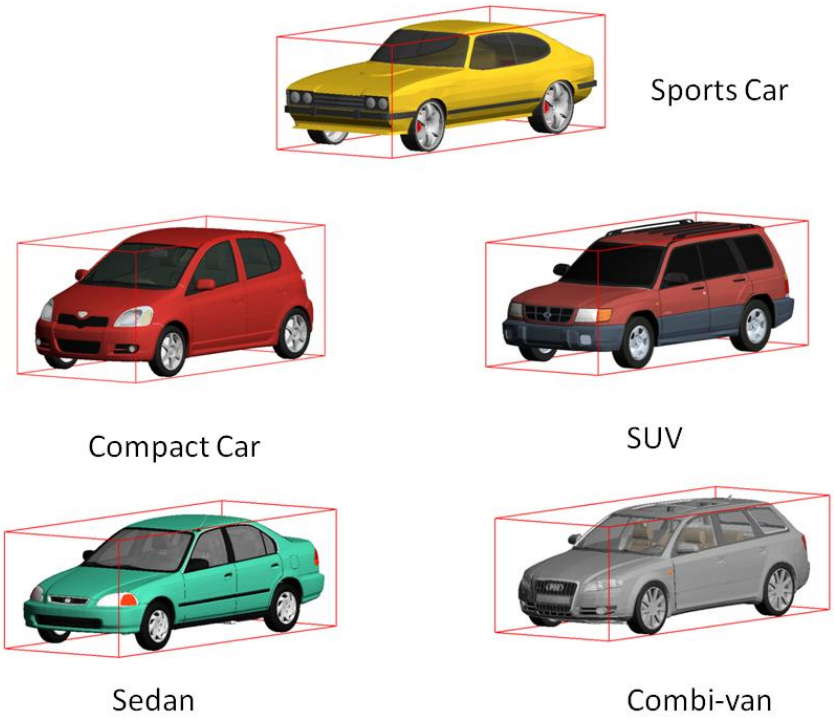


Figure 4.5: 3D CAD models in the training set

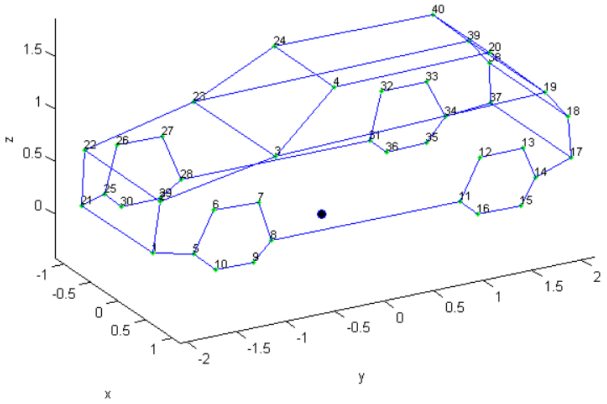


Figure 4.6: Mean Model

deform the mean model so as to deliver different shapes. By a linear combination of eigenvectors in Eq. 3.20, different vehicle shapes can be generated by variation of shape parameters γ_k. Based on an empirical evaluation, two shape parameters ($i \in \{1, 2\}$) were found to be sufficient to represent all vehicles in our test data set.

4.5.2 Initialization for model fitting

In our approach, two vehicles in each vehicle track are modelled simultaneously and starts from epoch $t = 1$. In order to fit models for each vehicle track, several parameters have to be determined. The vectors of unknowns, described in Chapter 4.1, has to be initialized.

The initialised position of the vehicle in the previous frame is defined as $({}^0p^x_{t-1}, {}^0p^z_{t-1})$, the center of bounding box on the ground plane. Using Eq. 4.7, the initialised position of the corresponding vehicle in the current frame can be determined by translation velocity $({}^0v^x_t, {}^0v^z_t)$.

In order to overcome the initialisation problems for heading in (Coenen et al., 2017), the prior knowledge about moving direction in a vehicle track is used for initialing heading instead of using the direction of the semi-major axis of the bounding box. Based on the relative position of two vehicles in previous and current frame, the orientation of previous frame can be initialised as

$$\theta_{t-1} = atan(\frac{v^x_t}{v^z_t}) \tag{4.17}$$

We assume that, the heading direction of a vehicle does not change much in a short time step. As a consequence, the initialised orientation of the vehicle in current frame is assumed the same as the initialised orientation ${}^0\theta_{t-1}$ of the vehicle in previous frame belonging to current vehicle.

As initialization of the vehicle shape, the mean model (cf. Chapter 4.5.1) is used which results in the initial parameters $({}^0\gamma^1_{t-1}, {}^0\gamma^2_{t-1})$ are as $(0, 0)$ vector. Two vehicles in one vehicle track are the same vehicle. We assume that, the shape parameters of a vehicle is identical in two continuous epochs.

The initial parameters for modelling one vehicle track is defined as $({}^0p^x_{t-1}, {}^0p^z_{t-1}, {}^0\theta_{t-1}, {}^0\gamma^1_{t-1}, {}^0\gamma^2_{t-1}, {}^0v^x_t, {}^0v^z_t)$.

4.5.3 Model fitting model and pose estimation

Active shape models consist of sets of 3D vertices. Model fitting is based on minimizing the distances between 3D object points belonging to a vehicle and the models surfaces

according to the active shape models. In order to fit active shape model to a 3D point cloud, a triangle mesh is defined for active shape model vertices. The model surface is represented by these triangles. In our approach, 40 vertices determine 58 triangles for each vehicle.

In active shape model, the shape parameters variate to determine the best shape model for each vehicle track. By translation and rotation of the model on the detected ground plane, the best fitting positions can be found.

Model fitting starts from epoch $t = 1$ and two vehicles in one vehicle track are modelled simultaneously. For each vehicle track, the model for the vehicle in the previous frame M_{t-1} is described by position, orientation and shape parameters, as $(p^x_{t-1}, p^z_{t-1}, \theta_{t-1}, \gamma^1_{t-1}, \gamma^2_{t-1})$. At the same time, the generated model M_{t-1} can provide the model for the corresponding vehicle in the current frame M_t by translational movement, as $M_t(M_{t-1}, v^x_t, v^z_t)$. Consequently, the vehicle poses of two frames in one vehicle track can be modelled simultaneously with the help of 7 parameters in this approach. In contrast to 10 parameters for modelling in Coenen et al. (2017), this approach makes the calculation much less.

Model fitting can be defined as minimizing an energy function for the generated model of a vehicle in two frames with respect to the corresponding 3D object points.

$$E\left(x_{t-1}, x_t, M_{t-1}, M_t\right) = E_{t-1}\left(x_{t-1}, M_{t-1}\right) + E_t\left(x_t, M_t\right) \tag{4.18}$$

where the 3D object points x_{t-1} for vehicle in the previous frame and the 3D object points x_t for vehicle in the current frame are observed for the same vehicle in two different time steps.

The energy of the model in each frame t can be defined as

$$E_t(x_t, M_t) = \frac{1}{P}\sum_{p=1}^{P} d(x^p_t, M_t) \tag{4.19}$$

The energy term corresponds to the mean distance of the P object points $x^p_t \in x_t$ to the triangulated model surface, similar to Xiao et al. (2016). This mean distance is defined as a score for each model. Model fitting for two vehicles in one vehicle track is based on minimizing energy in Eq. 4.18 and try to minimize the scores for both models

M_{t-1} and M_t.

For determining active shape model for each vehicle track, an iterative fitting method is implemented in this approach. In each iteration, we generate i model particles M_{t-1}^i for the vehicle in the previous frame. For each generated model particle M_{t-1}^i, j model particles M_t^j are generated for the corresponding vehicle in the current frame.

In the beginning, the model M_{t-1}^0 and the model M_{t-1}^0 generated by the initial parameters, described in Chapter 4.5.2, are used to calculate an initial score by Eq. 4.18. In the first iteration, we draw i particles to create i different M_{t-1}^i models. For each model M_{t-1}^i, the parameters $p_{t-1}^x, p_{t-1}^z, \theta_{t-1}, \gamma_{t-1}^1, \gamma_{t-1}^2$ are sampled from a uniform distribution centered at the initial parameters. The interval boundaries of the uniform distributions of each parameter are defined individually. For each model particle M_{t-1}^i, we generate j particles of the model M_t^j. For each model M_t^j, the translational velocity parameters v_t^x, v_t^z are sampled from a uniform distribution centered at the initial parameters and the interval boundaries of the uniform distributions of translational velocity parameters are predefined. By using Eq. 4.18 and Eq. 4.19, the scores of each model M_{t-1}^i and the scores of the corresponding models M_t^j are calculated. The M_{t-1}^i model with the minimum score and the M_t^j model corresponding to the M_{t-1}^i model with the minimum score are selected as best scoring models in the first iteration.

For the further iterations, the procedure of particles generation and scoring calculation are the same. However, in the subsequent iterations, the parameters from the best scoring models in last iteration are used as the centers of the uniform distribution and the respective interval ranges of each parameters are reduced with a certain descending factor τ. After n_p number of iterations, the model M_{t-1}^B with minimum score and the corresponding model M_t^B to the M_{t-1}^B model with minimum score from all iterations is selected as final models for one vehicle track.

The final models are generated as wireframes in 3D object space. The position and orientation parameters in the final models are recorded as outputs of pose estimation for each vehicle track.

Chapter 5

Experiments and evaluation

5.1 Test data and setup

In this thesis, the dataset in the object tracking benchmark from the KITTI Vision Benchmark Suite (Geiger et al., 2012) is used in experiments and evaluation. The Kitti Vision Benchmark Suite is a project of Karlsruhe Institute of Technology and Toyota Technology Institute at Chicago. The data of the KITTI Vision Benchmark Suite are captured by the autonomous driving platform Annieway, which is a VW Passat B6 wagon equipped by multiple sensors as shown in 5.1, in rural areas and on highways around Karlsruhe.

Besides a Velodyne laser scanner and a GPS localization system, four cameras are equipped on the wagon, namely two high-resolution color and two grayscale video cameras, the type of *Point Grey Flea*2 1.4 *Megapixels*. The two color cameras are mountained on the top of the wagon with 54 [*cm*] length of the baseline between each other, the same setup of the two grayscale cameras. The cameras are triggered at 10 frames per second and the images are cropped to a size of 1382 $\times$ 512 pixels (Geiger et al., 2012).

The object tracking benchmark dataset in the KITTI Vision Benchmark Suite consists of 21 training sequences with labeled objects and 29 test sequences without labelling. In the object tracking benchmark, the labeled objects in the training sequences can help to evaluate the approach presented in this thesis. These labels include different types of objects, based on the left image of stereo image pairs. In this approach, we only

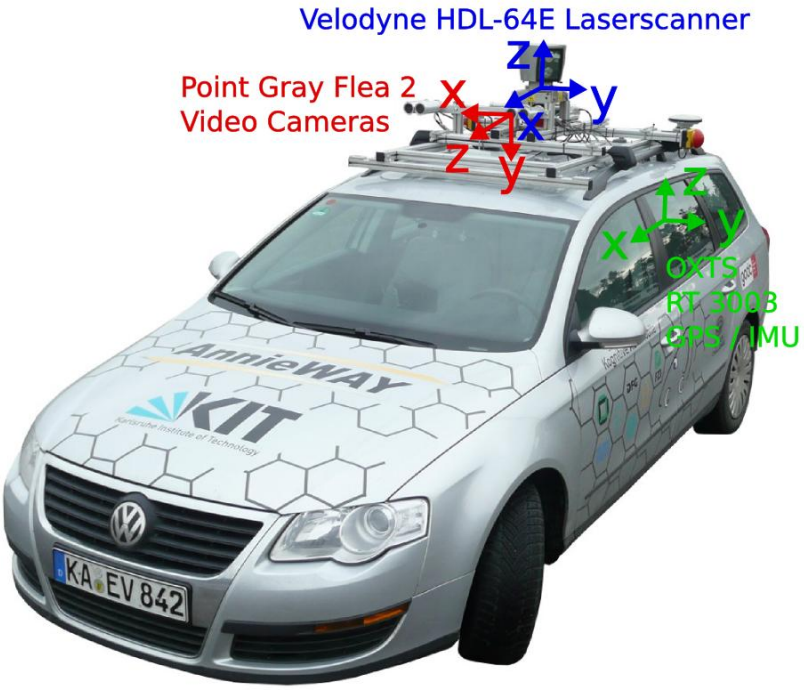


Figure 5.1: Multiple sensors setup on the autonomous driving platform "Annieway" [http://www.cvlibs.net/datasets/kitti/, 09.09.2017]

consider cars. For each tracking object in every frame, the benchmark provides frame number, tracking ID, 2D image bounding box and pose information. Poses of tracked vehicles are represented by 3D object location in object space and orientation, which is the rotation angle about the vertical axis in object space. Vehicles near image borders may be truncated. In the reference, the degree is recorded, too, with 0 = not truncated, 1 = partly truncated and 2 = largely truncated. In some case, vehicles are located behind other objects from the viewpoint of the camera. Since occlusion values are also provided for each object tracking reference, with 0 = fully visible, 1 = partly occluded, 2 = largely occluded and 3 = unknown. Due to different truncation and occlusion situations, three levels of difficulties are determined, as shown in Table 5.1.

Table 5.1: Definition of different reference mode

	Easy	Moderate	Hard
Maximum truncation	0	1	2
Maximum occlusion	0	1	2

In this thesis, stereo image sequences acquired by static cameras are used as data

source. From the object tracking benchmark dataset, two stereo image sequences from training sequences with no camera motion and the corresponding labelled objects are selected for experiments and evaluation. One of these sequence (Sequence 1) consists of 81 frames, where the vehicle with cameras was standing at a traffic light. This scene is an open street with no obvious objects between the cameras and passing vehicles, cf. Figure 5.2. The number of occluded and truncated vehicles is low. The other sequence (Sequence 2) consists of 22 frames. The vehicle with cameras is standing in a street and a traffic light occludes the moving vehicles, cf. Figure 5.3. The number of occluded and truncated vehicles is high. The following evaluation is based on these two sequences, in order to test this approach under different situations.

Figure 5.2: Image from Sequence 1

Figure 5.3: Image from Sequence 2

5.1.1 Evaluation strategy

For the evaluation of vehicle detection, we calculated the intersection of union score S_{IOU} of the detected 2D bounding box BB_{res} and the reference bounding box BB_{ref}, also known as the Jaccard Index (Everingham et al., 2010).

$$
\begin{aligned}
S_{IOU} &= 100 \cdot \frac{|BB_{res} \cap BB_{ref}|}{|BB_{res} \cup BB_{ref}|} \\
&= \frac{|BB_{res} \cap BB_{ref}|}{|BB_{res}| + |BB_{ref}| - |BB_{res} \cap BB_{ref}|}
\end{aligned}
\tag{5.1}
$$

As defined in (Geiger et al., 2012), the detection with the intersection of union score S_{IOU} of one detection higher than 50% and the height of the detected 2D bounding box BB_{res} larger than 25 [*pixel*], is regarded as a correct detection, namely, true positive (TP). When other objects, besides vehicles, are detected as vehicles, the detections are regarded as false positive (FP). False negative (FN) refers to, the vehicles in the reference can not be detected in the experiments. If multiple detections occurs for the same vehicle, we count only one detection as true positive and the further detections are seen as false positives. Furthermore, we calculate the values for (Heipke et al., 1997)

$$
Completness[\%] = 100 \cdot \frac{TP}{TP + FN} \tag{5.2}
$$

$$
Correctness[\%] = 100 \cdot \frac{TP}{TP + FP} \tag{5.3}
$$

$$
Completness[\%] = 100 \cdot \frac{TP}{TP + FN + FP} \tag{5.4}
$$

The completeness measures the percentage of the reference objects that can be detected and the correctness represents the percentage of the detected objects having a corresponding in the reference. Quality is used for ranking as an overall accuracy.

For the evaluation of vehicle tracking, we use the similar method as the evaluation of vehicle detection, by calculating the intersection of union score S_{IOU} of the tracked 2D bounding box and the reference bounding box, using Eq. 5.1. We reqiure the S_{IOU} higher than 30% and the tracked 2D bounding box BB_{res} larger than 25 [*pixel*]. Additionally, we compare the tracking ID of the tracked vehicle to the reference. The tracking ID represents the unique ID of one object in a sequence and represent the unique trajectory of an object. True positive (TP) refers to the tracked vehicle having the same tracking

ID as the reference. False positive (FP) means that, the tracked vehicle is assigned to the wrong trajectory, with a different tracking ID as in the reference. False negative (FN) represents that, a vehicle has not been tracked in one frame of a sequence. The completeness, correctness and quality are calculated using Eq. 5.2, Eq. 5.3and Eq. 5.4. The completeness is the tracked rate, the percentage of the reference objects that can be tracked. The correctness represents the ratio of the correctly tracked objects, with the same tracking ID to the reference. Quality, as overall accuracy, is used for ranking.

For evaluating pose estimation, the position p_t on the ground plane and the orientation θ_t from the model fitting procedures are compared to the references. Only the poses belonging to correctly tracked vehicles (TP) are considered. Because the falsely tracked vehicle (FP) and missed tracked vehicle (FN) cannot estimate poses in the first place. If the difference of position to reference is lower than 0.75 $[m]$ and difference of orientation to reference is lower than 22.5°, the estimation is considered to be correct.

5.1.2 Parameter settings in experiments

Our vehicle tracking and pose estimation approach requires the definitions of several parameters. In each step, the determination takes place in different ways and the individual parameter will be described separately in the following.

Parameters for 3D reconstruction

In 3D reconstruction (cf. Chapter 4.2) the minimum valid disparity needs to be defined. The necessary parameters, the maximal value $\sigma_{z_{max}}$ for precision of the depth values and σd as precision of the disparity, are defined in Table 5.2. According to the sensors setup in the KITTI Vision Benchmark Suite, the focal length of two color cameras is 738.4 $[pixel]$ and the length of baseline between two cameras is 54 $[cm]$. Using Eq.4.2 the minimum valid disparity can be defined. As a consequence, the maximum depth value z_{max} of a reconstructed 3D point cloud is 24.3 $[m]$.

Table 5.2: Parameters to determine minimum valid disparity

Parameter	Value	Description
$\sigma_{z_{max}}$	1.5 $[m]$	maximum valid precision of depth value
σd	1.0 $[pixel]$	precision of disparity
d_{min}	16.1 $[pixel]$	minimum valid disparity

Parameters for vehicle detection

The first step of vehicle detection is to determine the ground plane (cf. Chapter 4.3.1). In ground plane extraction, the random sampling consensus algorithm is used and the definitions of the corresponding parameters in the experiment refer to Table 5.3.

Table 5.3: Parameters for ground plane extraction

Parameter	Value	Description
N	1500	number of points to detect the ground plane
d_{thres}	20 $[cm]$	distance threshold to determine the inliers
n_i	50	number of iterations in RANSAC

In vehicle detection, a few parameters are needed to define the vehicle detection assumptions (cf. Chapter 4.3.2) and these parameters are shown in Table 5.4.

Table 5.4: Parameters in the vehicle detection assumptions

Parameter	Value	Description
h_{max}	2.5 $[m]$	maximum height of a vehicle
A_{min}	1 $[m^2]$	minimum area covered by a vehicle
A_{max}	15 $[m^2]$	maximum area covered by a vehicle

In verification with deformable part model (cf. Chapter 4.3.4), the threshold θ_J for Jaccard Index is defined experimentally and is set to $\theta_J = 0.3$.

Parameters for tracking

In vehicle tracking, several parameters in the Kalman filter (cf. Chapter 4.4.1), which is used for trajectories estimation, need to be determined as in Table 5.5.

Table 5.5: Parameters in the Kalman filter

Parameter	Value	Description
v_0^x, v_0^z	15 $[m/s]$	initial velocity of the vehicle in x and z axes on the ground plane
${}_0P_{p^x}, {}_0P_{p^z}$	$5[m]$	initial covariance of the position
${}_0P_{v^x}, {}_0P_{v^z}$	$2[m/s]$	initial covariance of the velocity
$\sigma_{p^x}, \sigma_{p^z}$	$0.5[m]$	precision of the measured positions in x and z axes on the ground plane
$\sigma_{v^x}, \sigma_{v^z}$	$0.2[m/s]$	precision of the velocity in x and z axes on the ground plane

The threshold for energy function E_{thres} used in tracking (cf. Chapter 4.4.2) is defined based on experiments and is set to $E_{thres} = 0.8$.

Parameters for model fitting

In model fitting procedure, an iteration approach is used(cf. Chapter 4.5.3). Table 5.6 performs the definition of the related parameters in model fitting approach.

5.2 Evaluation

5.2.1 Vehicle detection

Vehicle detection results are essential for the following tracking and pose estimation processes. If the vehicles cannot be detected correctly, vehicles cannot be tracked successfully. In the evaluation of detection, we only focus on the vehicles that can be tracked.

Table 5.7 shows the results of evaluation for detected vehicles in Sequence 1.

In Sequence 1, the quality and completeness of results decrease with increasing level of difficulty. No matter in which difficulty mode, there is a few false positive detections and correctness show good state.

In Table 5.7, there are 5 false positive detections. In this approach, we use minimum bounding rectangle to cover the vehicle parts visible in both stereo images, however, in reference, bounding boxes also contain the non-visible parts. As a consequence, the intersection of union score S_{IOU} of two bounding boxes for these 5 detections become

Table 5.6: Parameters for model fitting

Parameter	Value	Description
n_p	10	number of iterations
i	80	number of generated model M_{t-1} in each iteration
j	10	number of generated model M_t corresponding to each different model M_{t-1}
$R_{p^x_{t-1}}, R_{p^x_{t-1}}$	$\pm 1.5\ [m]$	the interval boundary of the uniform distribution for sampling different positions parameters
$R_{\theta_{t-1}}$	$\pm 15°$	the interval boundary of the uniform distribution for sampling different orientations parameters
$R_{\gamma^1_{t-1}}, R_{\gamma^2_{t-1}}$	± 1	the interval boundary of the uniform distribution for sampling different shape parameters
$R_{v^x_{t-1}}, R_{v^x_{t-1}}$	$\pm 15\ [m/s]$	the interval boundary of the uniform distribution for sampling different velocities parameters
τ	0.9	descending factor of reducing the interval boundary of sampling different parameters in uniform distribution in each subsequent iterations

Table 5.7: Detection evaluation in Sequence 1

	Easy	Moderate	Hard
Ref	118	155	173
TP	113	139	140
FP	5	5	5
FN	5	16	33
Completeness	95.8%	89.7%	80.9%
Correctness	95.8%	96.5%	96.6%
Quality	91.9%	86.9%	78.7%

less than 50% (cf. Chapter 5.1.1) and these 5 detections are regarded as false positive detections. For instance, as in Figure 5.4, the detected vehicle with the orange bounding box and the blue bounding box is one of these 5 detections, with S_{IOU} less than 50%.

The other two vehicles, by green bounding boxes (reference) and blue bounding boxes (detection) covering, have the S_{IOU} larger than 50%. The vehicles without bounding box are not considered in this approach, because of the maximum depth value z_{max} of the generated 3D point cloud (cf. Chapter 4.2).

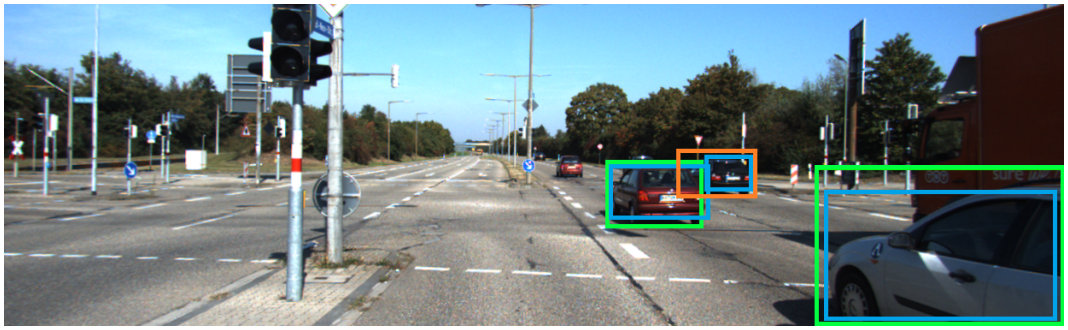


Figure 5.4: False positive detection in Sequence 1

For the easy difficulty level, only few vehicles are missed, and our method can achieve over 95% in completeness and correctness. However, the numbers only consider the vehicles without occlusion and truncation. In the moderate level, there are 37 vehicles in reference and 11 of them missing detection more than the easy level. The completeness decrease 6% and correctness is almost the same. In the hard level, there are 18 vehicles, which are largely truncated or occluded, and 17 of them can not be detected at all. The completeness in the hard level is almost 10% lower than in the moderate level and 15% lower than in the easy level. However, the number of false detections remains the same.

In this detection approach, our vehicle assumptions limits the detection of vehicles partly visible. The 3D points cannot be reconstructed in occluded situations. This leads to plenty of missing vehicles in the moderate and hard levels. For instance, in Figure 5.5, the vehicle in the lower right corner of image and the one occluded by another vehicle cannot be detected successfully. In Figure 5.5, red bounding boxes are missing detections, the true positive detection is covered green bounding box (reference) and blue bounding box (detection) simultaneously.

Table 5.8 shows the evaluation of vehicle detection in Sequence 2.

In Sequence 2, the evaluation results show a similar tendency as for Sequence 1. However, a traffic light always occludes the passing vehicles, so that most detected vehicles belong to the moderate and hard type in reference. There are only 3 fully visible vehicles,

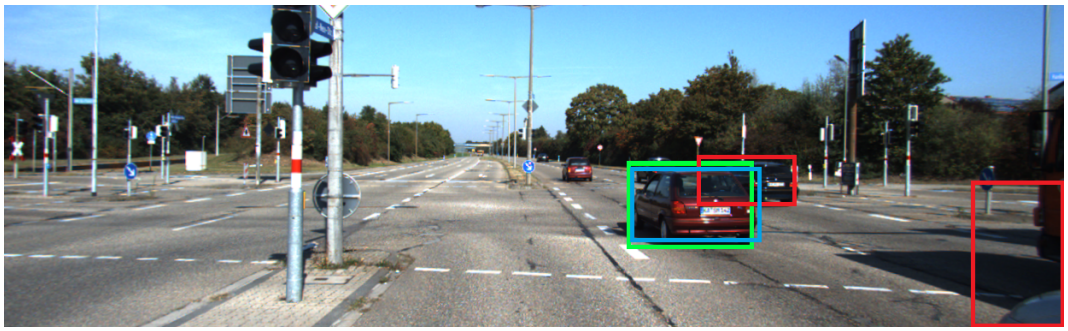


Figure 5.5: Missing detection in Sequence 1 in the hard mode evaluation

Table 5.8: Detection evaluation in Sequence 2

	Easy	Moderate	Hard
Ref	3	25	26
TP	3	16	16
FP	0	0	0
FN	0	9	10
Completeness	100.0%	64.0%	61.5%
Correctness	100.0%	100.0%	100.0%
Quality	100.0%	64.0%	61.53%

and all of them can be detected correctly. With evaluation difficulty increasing, more and more vehicles cannot be detected. More than 30% of the vehicles are missing in the moderate and hard level. However, there is no false detections in any frame from this sequence. Comparing hard and moderate modes, there is only one new vehicle and missing in detection with this approach.

Figure 5.6: Missing detection in Sequence 2 in the hard mode evaluation

Figure 5.6 shows a vehicle which is partly occluded behind the traffic light. Another vehicle is truncated heavily. It is hard to detect both of them, and most frames in this

sequence have similar characteristics.

In Sequence 1, no matter in which mode, over 80% vehicles of the vehicles from the reference can be detected and over 95% are correct. Most vehicles are reliably detected, which is important for the following tracking procedure. In Sequence 2, due to the high number of missing detections, tracking becomes hard. Most vehicles, with largely occluded and truncated as in hard mode, are hard to be detected. This may lead to problems in tracking.

5.2.2 Vehicle tracking

Table 5.9 shows each evaluation result of tracking in Sequence 1 and the definitions of parameters in tracking evaluation refer to Chapter 5.1.1.

Table 5.9: Tracking evaluation in Sequence 1

	Easy	Moderate	Hard
Ref	123	160	178
TP	116	132	132
FP	0	0	0
FN	7	28	46
Completeness	94.3%	82.5%	74.2%
Correctness	100.0%	100.0%	100.0%
Quality	94.3%	82.5%	74.2%

In Sequence 1, the evaluation results show similar potential to the detection evaluation results. However, due to undetected vehicles, there are lots of missed tracked vehicles and completeness is lower compared to Table 5.9. In easy level, a satisfactory result is achieved nevertheless. About 95% of the vehicles are tracked successfully compared to reference and all tracked vehicles are assigned to the right trajectory (tracking ID) as in the reference, with 100% as correctness. The quality in easy level is as high as 91%.

In moderate and hard mode, the number of falsely tracking vehicles is the same, so that correctness remains at a high level. However, more and more tracking vehicles are missed with increasing difficulty, resulting about 82% completeness in moderate mode and 74% in hard mode. Due to the decreased completeness, quality becomes lower, too.

In the hard mode, the quality is only 74%. No new objects are tracked correctly in hard mode compared to moderate mode, which is the consequence of the results in Table 5.9.

Figure 5.7: Missing tracked in Sequence 1, 23rd,24th,25th frames in the hard mode evaluation

Figure 5.7 shows missing tracked vehicles of 23rd, 24th and 25th frames. Due to occlusion by other vehicles, a vehicle has not been detected successfully in the 23rd and 24th frames. As the vehicles are tracked based on detections in two continuous frames (previous and current frame). If a vehicle has not been detected successfully in two continuous frames, a vehicle cannot be tracked at least in 3 continuous frames, which is shown in the figure. The two vehicles in Figure 5.14 without bounding boxes are vehicles not considered in our approach, because of the maximum depth value z_{max} of the reconstructed 3D point cloud (cf. Chapter 4.2). The vehicle covered by red bounding boxes is missing tracked vehicle and the other vehicle with green (reference) and blue

(tracked) bounding boxes is tracked correctly.

Table 5.10 shows tracking evaluation in Sequence 2.

Table 5.10: Tracking evaluation in Sequence 2

	Easy	Moderate	Hard
Ref	3	25	26
TP	3	15	15
FP	0	0	0
FN	0	10	11
Completeness	100.0%	60.0%	57.7%
Correctness	100.0%	100.0%	100.0%
Quality	100.0%	60.0%	57.7%

There are only 3 vehicles in the easy level, and all of them can be tracked successfully, with 100% in completeness and correctness. However, in moderate mode and hard mode, the ratio of missing tracked vehicles is quite high, although there are no false positive tracked vehicles. Completeness and quality are only about 60% in the moderate and hard modes. There are 22 tracked objects in the moderate mode than in the easy mode. Only 12 of them can be tracked correctly and the other 10 are false negative tracked. No new objects are tracked correctly, in hard mode compared to moderate mode, the only new vehicle is missing due to more occlusion and truncation.

As the same lack of detections situation discussed before, there is no successfully detections in 4,5 and 6 frames, as Figure 5.8. Red bounding boxes are missed detection, green bounding boxes are references and blue bounding boxes are detections. Tracking-by-detection strategy cannot work without detections in the first place.

In easy evaluation mode, the method described before can achieve satisfactory results in tracking, with correctness higher than 90% and 100% in completeness. Occluded and truncated vehicles are hard to detect with our approach and show low ratio of successful detection. This leads to a lower tracking quality in the moderate mode than in the easy mode, however, tracking quality in the hard mode is not satisfactory. With increasing occlusion and truncation, completeness and quality become low. The result of detections highly influence tracking and missing detections cause missing tracked vehicles. There

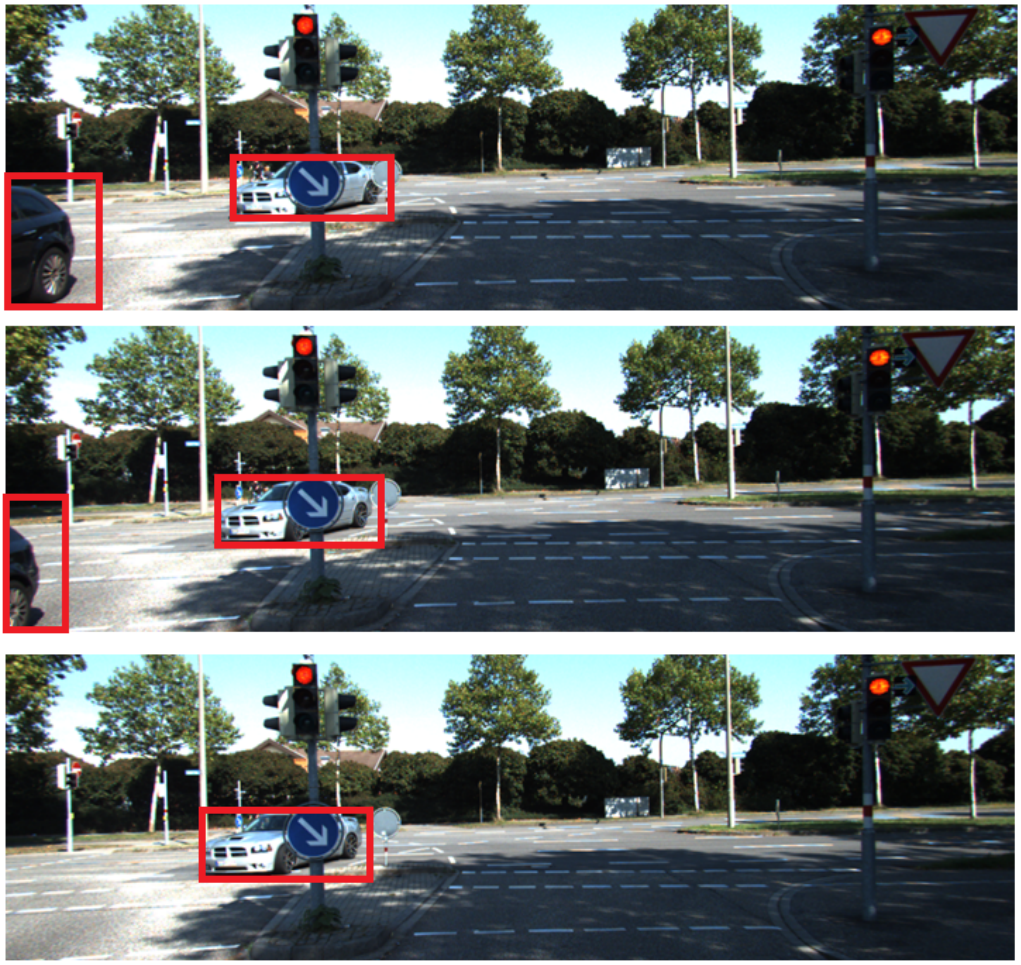


Figure 5.8: Missing tracked in Sequence 2, 4th,5th,6th frames in the hard mode evaluation

is no false positive, which the vehicle is tracked to other trajectory, in each evaluation mode.

5.2.3 Pose estimation

Table 5.11 shows the evaluation of the results of pose estimation for Sequence 1.

In Sequence 1, there is no difference between the hard and moderate mode, because the number of tracked vehicle is identical (cf. Chapter 5.2.2).

In easy mode, in which the vehicles are fully visible, almost all positions estimated are correct, about 88% correct. With increasing difficulty, the ratio of correct position estimations is decreasing a little compared to the easy mode and about 82.6% of the positions are correct for tracked vehicles. However, there are 16 more vehicles in the moderate mode tracked correctly compared to the easy mode and only 7 positions are

Table 5.11: Pose estimation evaluation in Sequence 1

	Easy	Moderate	Hard
Correct tracking	116	132	132
Correct position	102	109	109
Correct orientation	86	94	94
Correctness (position)	87.9%	82.6%	82.6%
Correctness (orientation)	74.1%	71.2%	71.2%

estimated correctly. In the easy mode, about 74% orientations are estimated correctly and 3% lower in the moderate and the hard mode.

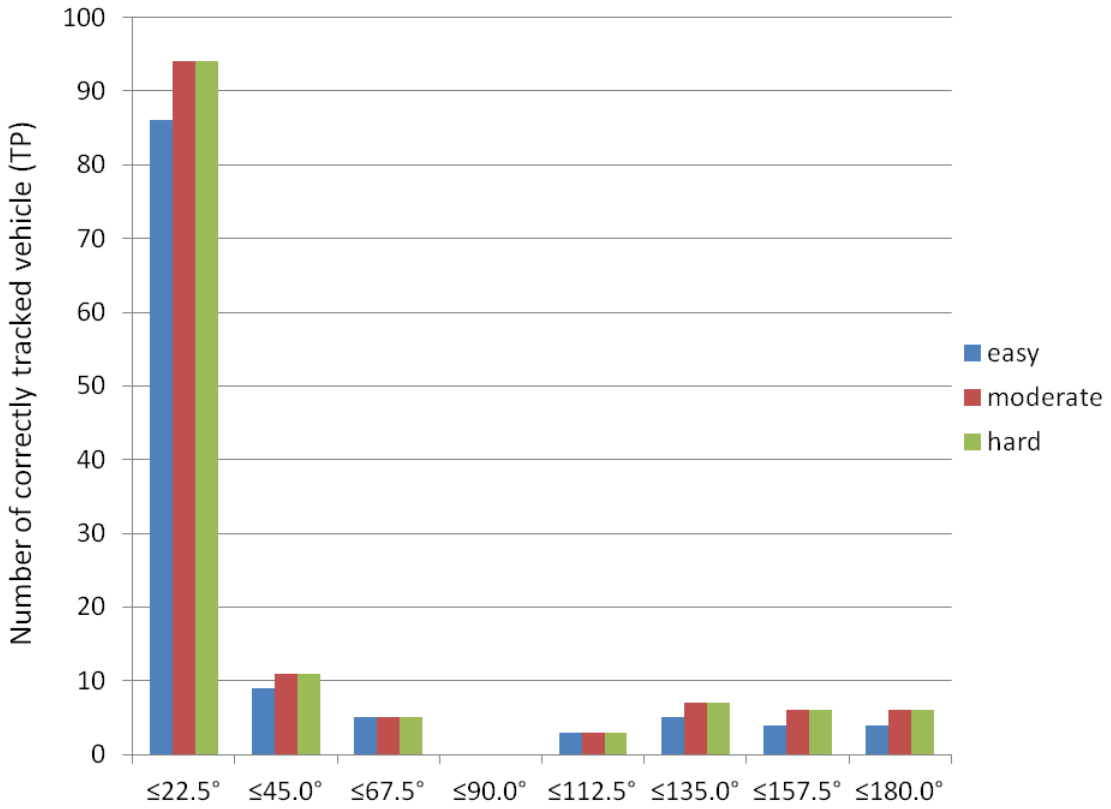


Figure 5.9: Histogram of absolute differences between estimated orientation and reference orientation in Sequence 1

Figure 5.9 shows the histogram of absolute differences between estimated orientation and reference orientation in Sequence 1. From this histogram it is apparent that the absolute differences of estimated orientation and reference orientation are distributed between 22.5° and 180° averagely. In Sequence 1, a lot of vehicles appear from the boundaries of

images. In the boundaries of images, the stereo image pairs cannot be matched well and leads to the low number of 3D object points. Consequently, the positions are imprecise and calculate a false initial orientation for the vehicles belonging to one track, where the initial orientation are calculated based on the relative positions of corresponding vehicles in two continuous frames (cf. Chapter 4.5.2). In model fitting procedure, the particles of orientation parameters are generated beyond a uniform distribution in a certain range $\pm 15°$. As a consequence, if the initial orientation is extremely imprecise, i.e. absolute difference of initial orientation and reference orientation larger than $90°$, it cannot estimate a precise orientation at all.

Table 5.12: Pose estimation evaluation in Sequence 2

	Easy	Moderate	Hard
Correct tracking	3	15	15
Correct Position	3	5	5
Orientation	3	11	11
Correctness (position)	100%	33.33%	33.33%
Correctness (orientation)	100%	73.3%	73.3%

In Sequence 2, the potential is different from Sequence 1, as conclusion in Table 5.12. In easy mode, all position and orientations are correct. However, due to high occlusion, 3D object points and constructed bounding boxes on the ground plane can only cover a part of the tracked vehicles. The position is defined as the center of 3D bounding box, which is determined from 3D object points projected to the ground plane. When occlusion and truncation, 3D object points cannot be reconstructed very well and only a part of a vehicle can be detected. This causes the position to be different from the position that would be calculated from the entire vehicle. However, in the reference, 3D bounding boxes are defined manually from the whole vehicles, and positions are calculated from the whole vehicles.

Figure 5.10 shows an example with an obvious difference of tracked bounding box and the reference bounding box is obvious. In our approach, we only cover the vehicle parts visible and use visible parts to calculate the positions, which leads to imprecise position estimation. Figure 5.11 shows the histogram of absolute differences between estimated

Figure 5.10: Difference of tracked bounding box (blue) and reference bounding box (green)

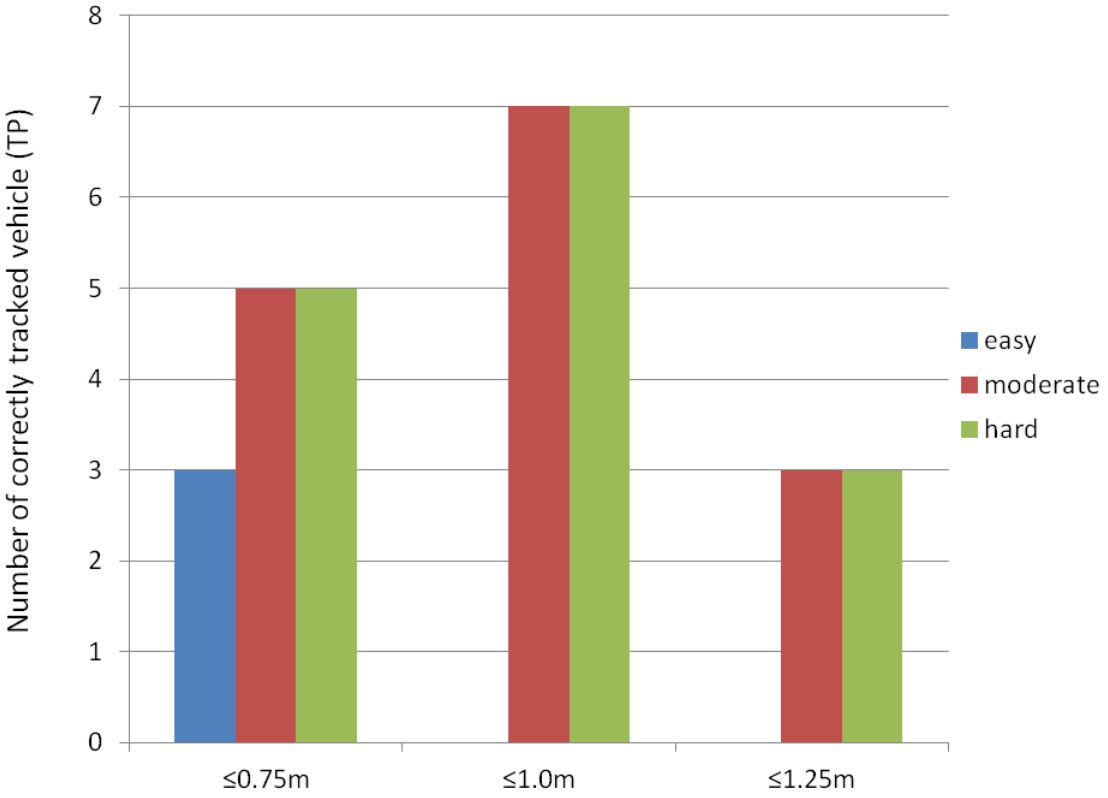


Figure 5.11: Histogram of absolute differences between estimated position and reference position in Sequence 2

position and reference position in Sequence 2. From this histogram it is apparent that the most absolute differences of estimated position and reference position are located between $0.75[m]$ to $1[m]$, which is caused by these position are estimated by the visible parts of vehicles.

In Sequence 2, although the estimated positions are not precise, the estimated orientations achieved a better correctness, cf. Table 5.12 and Figure 5.12. In this approach,

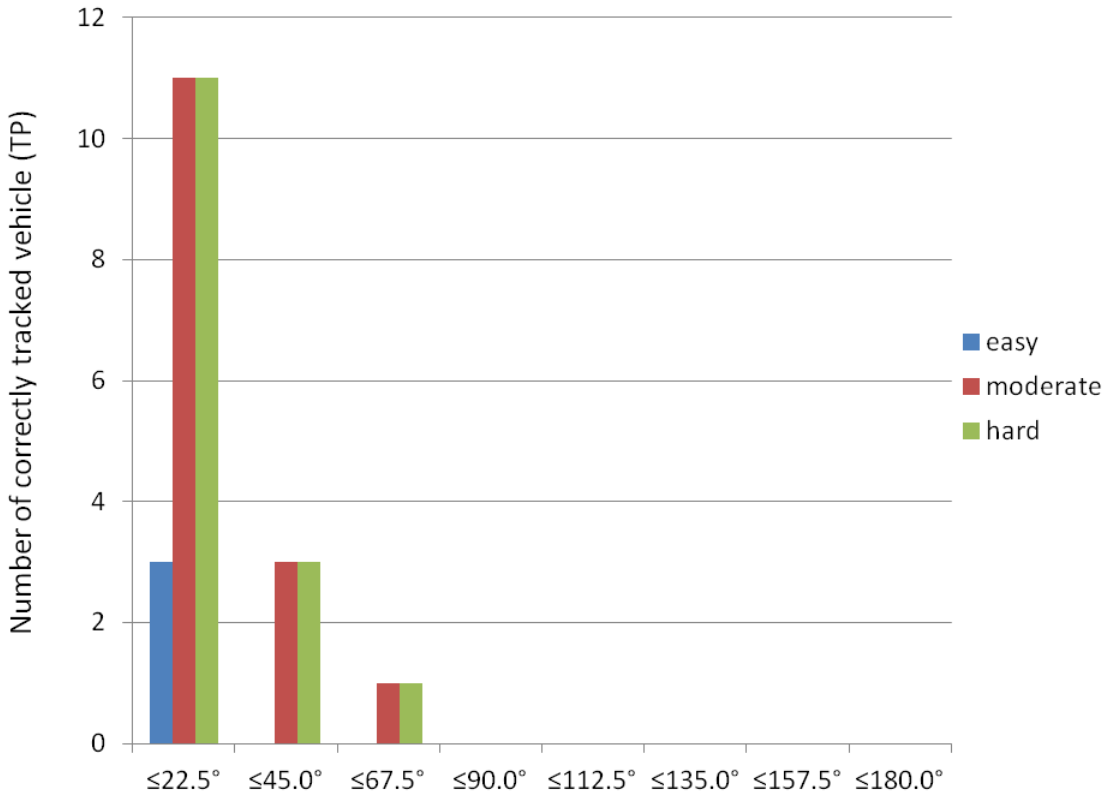


Figure 5.12: Histogram of absolute differences between estimated orientation and reference orientation in Sequence 2

relationships between positions of tracked vehicles in two continuous frames are used to initialize the orientation. The most estimated positions having a difference to the reference within $1[m]$, lead to the initialization of orientations reasonable. Meanwhile, the orientations can be optimized by 3D reconstruction of vehicles.

In Sequence 1, pose estimation of tracked vehicles is satisfactory. However, occlusion and truncation, which make vehicles only part visible, affect the correctness of pose estimation a lot. In Sequence 2, the positions estimation is not satisfactory, due to partly visible of the vehicles.

5.3 3D Modelling results

In this approach, an active shape model is used to model the 3D shapes of the tracked vehicles. Figure 5.13 and Figure 5.14 shows the results by backprojecting the resultant wireframes to the reference image. While the Figure 5.13 show the successful model

fitting example, the Figure 5.14 show some examples of errors.

Figure 5.13: Positive examples in 3D modelling

The successful model fitting example refers to that, the positions of predefined wire-frame vertices (cf. Figure 4.4) are located at the corresponding positions of the objects in the image, i.e. on the wheels of the vehicles, corners of the window and doors.

As in 5.14 first two rows, in this approach, the vehicle located in the boundaries of images usually cannot fit the model well with wrong size. In the boundaries of images, the stereo image pairs cannot be matched well and leads to the low number of 3D object

Figure 5.14: Typical errors in 3D modelling (First two rows: wrong size, third row: wrong orientation, last row: wrong position)

points. In our approach, we try to minimize the mean distance of the 3D points to the surface of the models. As a consequence, low number of 3D points cannot deliver a reliable mean distance, which leads to the size of generated models can not fit the objects. Additionally, the above discussed orientation errors in Sequence 1 (cf. Chapter 5.2.3) become apparent (cf. Figure 5.14 third row). The estimated orientation has almost 180° difference to the reference orientation. The occlusion effect the generated models a lot in Sequence 2 (cf. Figure 5.14 last row). The model can only fit to the visible part of

a vehicle and leads to large difference in position between the model and the object.

Chapter 6

Conclusion and Outlook

In this master thesis, an approach to track and estimate the pose of vehicles from stereo image sequences acquired by static cameras is developed, based on the vehicle detection approach by Coenen et al. (2017). Vehicle detection consists of two steps, the generic 3D object detection and the deformable part model as verification in the image. Based on detected vehicles, we try to associate the corresponding vehicles into trajectories using a tracking-by-detection strategy. We formulate the association as an energy minimization problem, where the energy function consists of data association term and motion term. In data association term, we used mean color difference for two vehicles from two continuous frames as the feature. In motion term we tried to minimize the difference of current measured state and the predicated state of the vehicle in the current epoch. The predicated state is estimated from the previous corrected state using the Kalman filter. In the last step, we make usage of a 3D active shape model to reconstruct the tracked vehicles in 3D. By fitting the model to the 3D point cloud of them, it can deliver the shape and pose parameters of each tracked vehicle.

Generally, this approach achieved satisfactory results. In detection, for fully visible vehicles, detection can achieve both completeness and correctness more than 95%. The number are lower, if any level of occlusion or truncation are considered, the correctness is 96% and completeness is 80%. However, vehicles with large occlusion or truncation cannot be detected quite well. As for based on reliable vehicle detections, all fully visible vehicles can be tracked correctly and only 6% of the detected vehicles are lost tracking. If any

level of occlusion or truncation are considered, the correctness is 100% and completeness is 74%. In our approach, vehicles have not been tracked to the wrong trajectories, namely 100% in correctness. However, due to poor detection results of partly visible vehicles, tracking is more problematic. 3D Active Shape Model is used as deformable vehicle model to represent 3D shapes for tracking vehicles. In model fitting model procedures, pose (position and orientation) of tracked vehicles can be optimized. More than 92% positions and 71% orientations of tracking vehicles can be calculated correctly.

The main problem of this approach is related to optimization position and orientation of tracked vehicles in the model fitting. The initial values of positions are defined as centers of 3D bounding boxes of tracked vehicles. If the vehicles are partly visible, the 3D bounding boxes only cover a part of vehicles, which leads to imprecise positions. As a consequence, the initial value of orientation is not reliable, because the orientation is defined by relative positions of two corresponding vehicles in two continuous frames. In the model fitting procedure, particles of positions and orientations are sampled from an assumed uniform distribution in a certain interval boundary. We want to enhance the model fitting approach in the future. One could generate particles for parameters, e.g. positions and orientations, beyond on a bimodal or multi-modal distribution.

For partly visible vehicles, tracking and 3D reconstruction cannot achieve a good result. In the future, our approach will be extended to handle with stereo image sequences, which are acquired from stereo cameras with ego-motion. Tracking vehicles will not only focus on two continuous frames. More frames will be considered jointly, and predictive models, e.g. (Klinger et al., 2017) to recover the trajectories for the occluded objects should be implemented. At the same time, shape parameters may be considered in the tracking process jointly. The model fitting procedure is only based on 3D points. In the future, features and information in 2D image space can be added into this procedure, i.e. the alignment of image edges and model edges. This would lead to more efficient computation and more precise models for the vehicles.

Bibliography

Andriyenko, A., Schindler, K., 2011. Multi-target tracking by continuous energy minimization, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1265–1272.

Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R., 2015. 3d object proposals for accurate object class detection, in: Advances in Neural Information Processing Systems, Vol. 28, pp. 424–432.

Choi, W., 2015. Near-online multi-target tracking with aggregated local flow descriptor, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3029–3037.

Coenen, M., Rottensteiner, F., Heipke, C., 2017. Detection and 3d modelling of vehciles from terrestrial stereo image pairs, in: International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. XLII-1/W1, pp. 505–512.

Cootes, T., Baldock, E., Graham, J., 2000. An introduction to active shape models, in: Image processing and analysis, pp. 223–248.

Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J., 1995. Active shape models-their training and application, in: Computer vision and image understanding, Vol. 61(1), pp. 38–59.

Engelmann, F., Stückler, J., Leibe, B., 2016. Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors, in: Pattern Recognition, Lecture Notes in Computer Science, Vol. 9796, pp. 219–230.

Engelmann, F., Stückler, J., Leibe, B., 2017. Samp: Shape and motion priors for 4d

vehicle reconstruction, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 400–408.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge, in: International Journal of Computer Vision 88(2), pp. 303–338.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models, in: IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9), pp. 1627–1645.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361.

Geiger, A., Roser, M., Urtasun, R., 2010. Efficient large-scale stereo matching, in: Computer Vision - ACCV 2010, Lecture Notes in Computer Science, Vol.6492, pp. 25–38.

Hartley, R., Zisserman, A., 2000. Multiple View Geometry in Computer Vision. Cambridge University Press, UK.

Heipke, C., Mayer, H., Wiedemann, C., Jamet, O., 1997. Evaluation of automatic road extraction, in: International Archives of Photogrammetry and Remote Sensing, Vol. 32, pp. 151–160.

Janai, J., Güney, F., Behl, A., Geiger, A., 2017. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art, in: Arxiv: 1704.05519.

Kalman, R.E., et al., 1960. A new approach to linear filtering and prediction problems, in: Journal of basic Engineering, Vol. 82(1), pp. 35–45.

Kitt, B., Geiger, A., Lategahn, H., 2010. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme, in: 2010 IEEE Conference on Intelligent Vehicles Symposium (IV), pp. 486–492.

Klinger, T., Rottensteiner, F., Heipke, C., 2017. Probabilistic multi-person localisation and tracking in image sequences, in: ISPRS Journal of Photogrammetry & Remote Sensing, Vol. 127, pp. 73–88.

Leibe, B., Leonardis, A., Schiele, B., 2006. An implicit shape model for combined object categorization and segmentation, in: Toward category-level object recognition, Lecture Notes in Computer Science, Vol. 4170, pp. 508–524.

Leibe, B., Schindler, K., Cornelis, N., Van Gool, L., 2008. Coupled object detection and tracking from static cameras and moving vehicles, in: IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10), pp. 1683–1698.

Lenz, P., Geiger, A., Urtasun, R., 2015. Followme: Efficient online min-cost flow tracking with bounded memory and computation, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4364–4372.

Lin, Y.L., Morariu, V.I., Hsu, W., Davis, L.S., 2014. Jointly optimizing 3d model fitting and fine-grained classification, in: European Conference on Computer Vision (ECCV), pp. 466–480.

Menze, M., Heipke, C., Geiger, A., 2015. Joint 3d estimation of vehicles and scene flow, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-3/W5, pp. 427–434.

Milan, A., Schindler, K., Roth, S., 2013. Detection-and trajectory-level exclusion in multiple object tracking, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3682–3689.

Ošep, A., Hermans, A., Engelmann, F., Klostermann, D., Mathias, M., Leibe, B., 2016. Multi-scale object candidates for generic object tracking in street scenes, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3180–3187.

Pepik, B., Stark, M., Gehler, P., Schiele, B., 2015. Multi-view and 3d deformable part models, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 2232–2245.

Shekhar, S., Xiong, H., 2007. Encyclopedia of GIS. Springer Science & Business Media, Germany.

Shi, Z., Zhu, S., Sun, W., Wang, B., 2014. Continuous energy minimization based multi-target tracking, in: Chinese Conference on Pattern Recognition, pp. 464–473.

Vedaldi, A., Soatto, S., 2008. Quick shift and kernel methods for mode seeking, in: Computer vision-ECCV 2008, Lecture Notes in Computer Science, Vol.5305, pp. 705–718.

Xiang, Y., Alahi, A., Savarese, S., 2015. Learning to track: Online multi-object tracking by decision making, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4705–4713.

Xiao, W., Vallet, B., Schindler, K., Paparoditis, N., 2016. Street-side vehicle detection, classification and change detection using mobile laser scanning data, in: ISPRS Journal of Photogrammetry and Remote Sensing 114, pp. 166–178.

Yingze Bao, S., Chandraker, M., Lin, Y., Savarese, S., 2013. Dense object reconstruction with semantic priors, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1264–1271.

Yoon, J.H., Yang, M.H., Lim, J., Yoon, K.J., 2015. Bayesian multi-object tracking using motion context from multiple objects, in: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 33–40.

Zhang, L., Li, Y., Nevatia, R., 2008. Global data association for multi-object tracking using network flows, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp. 1–8.

Zia, M.Z., Stark, M., Schiele, B., Schindler, K., 2013. Detailed 3d representations for object recognition and modeling, in: IEEE Transactions on Pattern Analysis and Machine Intelligence 35(11), pp. 2608–2623.

Zia, M.Z., Stark, M., Schindler, K., 2015. Towards scene understanding with detailed

3d object representations, in: International Journal of Computer Vision 112(2), pp. 188–203.