MONOCULAR RECONSTRUCTION OF DYNAMIC VEHICLES ON ARBITRARY ROAD PROFILES FROM A MOVING CAMERA

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

JUNAID AHMED ANSARI
20162149
`junaid.ansari@research.iiit.ac.in`

International Institute of Information Technology
Hyderabad - 500 032, INDIA
SEPTEMBER, 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "MONOCULAR RECONSTRUCTION OF DYNAMIC VEHICLES ON ARBITRARY ROAD PROFILES FROM A MOVING CAMERA" by JUNAID AHMED ANSARI, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. K. Madhava Krishna

To my parents

Acknowledgments

First and foremost, I would like to thank my supervisor, Prof. K. Madhava Krishna, for showing faith in me by offering me an MS by Research and giving me the opportunity to work in the Robotics Research Center (RRC), IIIT Hyderabad. Without his immense support and able guidance, I would not be anywhere close to where I currently stand.

I would also like to extend my thanks to J. Krishna Murthy for being an incredible mentor. I am grateful to Sarthak Sharma, Shashank Srikanth, Karnik Ram, Anshuman Majumdar, and Gokul for their constructive suggestions.

A special thanks to Gourav Kumar, Bharath Gopalakrishnan, Ravi Shankar Prasad, Ayush Gaud, Harit Pandya, Satyajit Tourani, and Sai Bhargav for making my stay at IIIT Hyderabad a pleasant one. I also want to thank Suman Saurabh, one of my really good friends from my undergraduate college, for all his help throughout my stay in Hyderabad, especially when I was new in the city. All the above mentioned people have always been there for me whenever I needed any help. Their company was really incredible and the time spent with them is invaluable. I would also like to thank all the students and interns at RRC, I really had a great time with them at RRC.

I am also extremely grateful to Dr. Ramesh Balasubramanyam, Associate Professor at Raman Research Institute, Bangalore, for giving me a chance to work with him after my undergraduate studies and for being extremely patient in teaching me how to approach research problems. It was the projects and research problems that I worked on with him that steered my research interest towards autonomous robotics and machine vision.

Abstract

Accurate localization of other traffic participants is a vital task in autonomous driving systems. State-of-the-art systems employ a combination of sensing modalities such as RGB cameras and LiDARs for localizing traffic participants. However, with the advent and subsequent commercialization of autonomous driving, there has been an increased interest in monocular object localization (and reconstruction) for urban driving scenarios. While there have been very interesting attempts to tackle this problem using a single monocular camera, most approaches assume that the ego car and the car to be localized share the same road plane. This condition is often termed as the coplanarity assumption.

In this thesis, we relax the aforementioned coplanarity assumption and demonstrate — to the best of our knowledge — the first results for monocular object localization and shape estimation on surfaces that are non-coplanar with the moving ego vehicle mounted with a monocular camera. We approximate road surfaces by local planar patches and use semantic cues from vehicles in the scene to initialize a local bundle-adjustment like procedure that simultaneously estimates the 3D pose and shape of the vehicles, and the orientation of the local ground plane on which the vehicle stands. We also demonstrate that our approach transfers from synthetic to real data, without any hyperparameter-/fine-tuning. We evaluate the proposed approach on the KITTI and SYNTHIA-SF benchmarks, for a variety of road plane configurations. The proposed approach significantly improves the state-of-the-art for monocular object localization on arbitrarily profiled/graded roads.

Contents

Chapter Page

List of Figures

List of Tables

Chapter 1

Introduction

With the advent and subsequent commercialization of autonomous driving, there has been an increased interest in monocular object localization (and reconstruction) for urban driving scenarios. Object localization is a crucial part of the overall autonomous driving ecosystem as it is very important for an autonomous car to be cognizant and keep track of its traffic participants, especially the dynamic ones. Here, by object we mean vehicles, specifically cars.

Reconstructing dynamic vehicles from a monocular camera is a challenging task, owing to several factors, such as dearth of stable feature tracks on moving vehicles, self-occlusion, and is ill-posed if the camera itself is in motion. To overcome these issues with monocular dynamic object localization, discriminative features [32] and shape priors [31, 30, 55] have been used to pose a bundle adjustment like scheme [31] that solves for the shape and pose of a detected vehicle, assuming a prior on the shapes of all instances from a category. Use of shape priors results in a richer representation of reconstructed vehicles (3D wireframes rather than 3D bounding boxes). Detailed reconstruction using shape priors can also aid tracking as they are composed of always existing semantic keypoints and not generic features which, in majority of the cases, are difficult to extract and reliably match owing to the vehicle's monochromatic nature.

While there have been successful attempts [30], [31], [41], [42], [13] to tackle this problem i.e. dynamic object localization using a single moving monocular camera, they are all confined to scenarios where the road is very (or nearly) flat. In other words, such approaches assume that the ego-vehicle (on which the camera is mounted) and the target vehicle i.e. the moving vehicle which is to be localized (and reconstructed) in full 6DoF, share the same road plane. This assumption is referred as the *coplanarity assumption* throughout this thesis. This assumption, although quite valid for majority of the urban driving scenarios, can still be violated in several cities like San Francisco (USA) and Hyderabad (India), where the roads are quite steep and graded.

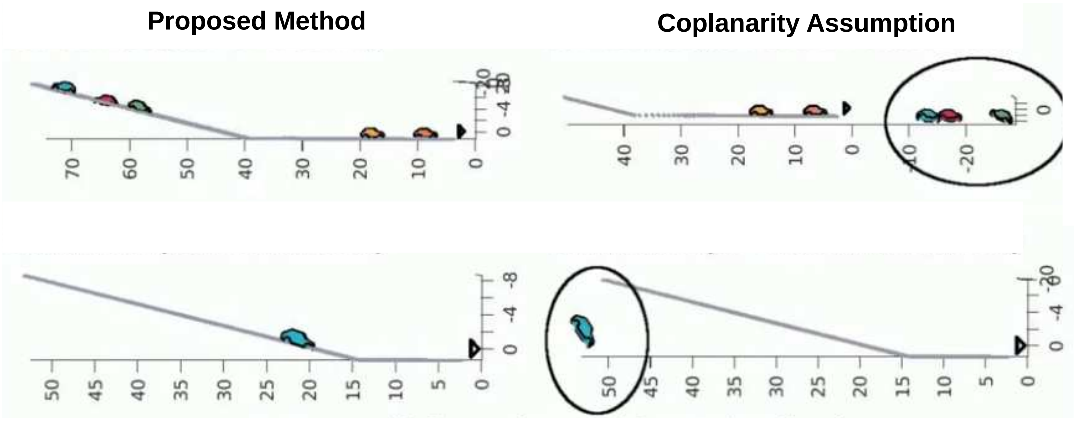

Figure 1.1 Failure of coplanarity assumption for steep roads on SYNTHIA-SF [24] dataset. Notice how the cars that are moving on road which is non-coplanar to the road plane of the ego vehicle are localized very accurately by the proposed method (left column). Where as, if the coplanarity is assumed as in [30, 31], the localization of those cars suffers drastically as shown in the right column (highlighted by black ellipses). The camera mounted on the ego vehicle is shown as black triangle

Contributions

In this thesis, we relax the aforementioned coplanarity assumption and *reconstruct vehicles moving on arbitrary road plane profiles* from a moving monocular camera. Following are the key contributions of this thesis:

- We demonstrate – *for the first time* – accurate localization (pose) and reconstruction (shape) of vehicles on *steep and graded roads* from a single moving monocular camera (see Fig. 1.1, left column). Other methods, for e.g. [30, 31], rely on the coplanarity assumption for monocular localization of vehicles and therefore drastically fail when the cars to be localized are not on the same road plane as that of the ego car(see Fig. 1.1, right column). We outperform our best competitor [31] by a significant margin as discussed briefly in the next section (detailed analysis is reported in Ch. 5). A typical outcome of the proposed work can be seen in Fig. 1.2.

- We propose a novel *joint optimization formulation* for accurate pose (localization) and shape (reconstruction) estimation of cars, predominantly using cues from a single image. The proposed method jointly optimizes over pose/shape and the local road plane parameters of the car. It relies on the fact that cars move on roads and hence their poses are constrained by their local road plane geometry.

- We introduce novel cost functions to narrow down the solution space leading to a more reliable and accurate localization and reconstruction. These cost functions, originating from the intuition that cars move on the road, primarily help in constraining the optimizer to adjust the pose and shape parameters of the car in a way that it always lies close to the road and its orientation closely matches the orientation of its local road plane.

- We propose a simpler method to learn the shape prior that does not require us to annotate the semantic keypoints in 3D. We first render the 3D car models from ShapeNet [8] using Blender [16]. Then we annotate the rendered images and use multi-view geometry to recover the 3D annotations of semantic keypoints. Finally, we use these 3D semantic keypoints of different models of car to learn the shape prior.

Evaluation

To evaluate our approach, we use KITTI [21] and SYNTHIA-SF [24] benchmarks. While sequences from KITTI [21] dataset only have mild-to-moderate slopes and banks, it provides a fair comparison with other baselines [30], [31]. Whereas, SYNTHIA-SF [24] has extremely steep roads and demonstrates the efficacy of the proposed approach in adapting to a wide range of road surfaces.

On KITTI [21], our approach shows a mark improvement over our best competitor [31] with an accuracy of about 3-4 times that of [31]. On SYNTHIA-SF [24], where [31] drastically fails, our method achieves localization with a mean error of less than 1 meter. These results showcase the importance of incorporating the local ground plane of the vehicles for the task of localization and reconstruction.

Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, we extensively survey existing approaches. In Chapter 3, we discuss on the relevant theoretical background and also briefly discuss monocular vehicle pose and shape recovery using the coplanarity assumption. Chapter 4 discusses on how we use synthetic data to learn shape priors i.e. a mathematical model which compactly encodes the shape-space of an entire object category. Finally, in chapter 5, we discuss our methodology and validate our claims by a thorough quantitative and qualitative analysis of the results with the current state-of-the-art monocular localization.

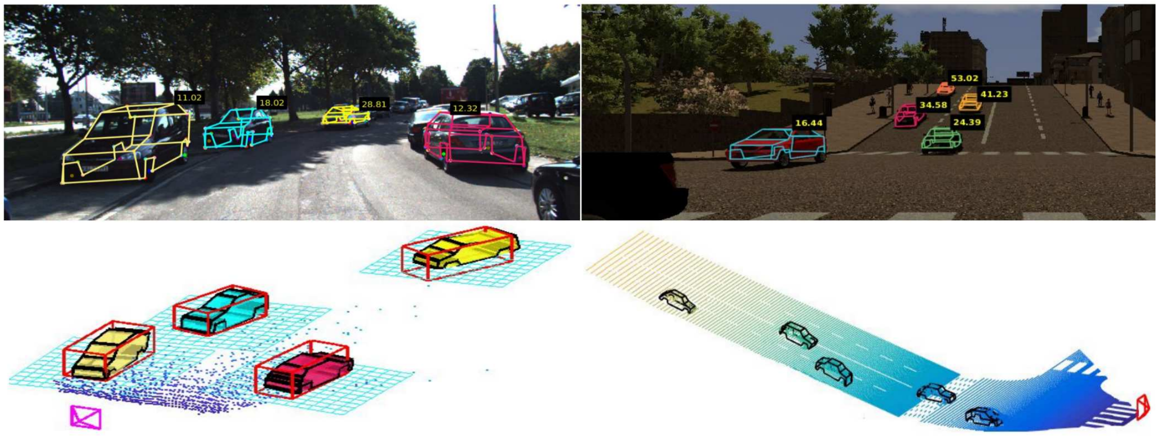

Figure 1.2 Outputs of the proposed monocular object localization system. The system is capable of estimating the shape and pose (without scale-factor ambiguity) of objects located on surfaces that are non-coplanar with the moving ego vehicle. *Top*: Projection of the estimated shapes (wireframes) of cars. Above each car, distance of the car from the camera is shown (in meters). *Bottom*: Estimated wireframe and road points in 3D. For the first image, estimated wireframes are compared with their respective ground truth 3D bounding boxes (in red), highlighting the accurate localization of the objects. In the second scene, we show the estimated cars in 3D, overlaid on a dense ground truth 3D point cloud consisting of road surface and the target vehicles. Notice how even objects over 50 meters away on steep slopes are accurately localized.

Chapter 2

Related Work

For any autonomous vehicle to drive safely, it is very important for it to keep track of other traffic participants, especially the moving ones. In the last decade or so, the robotic vision community has been aggressively targeting this problem of 3D object (here we work with only cars) localization. However, most of the successful efforts rely on a suite of sensors comprising of LiDARs, RADARs, stereo and monocular cameras. Very recently, due to the remarkable advancements in Convolutional Neural Networks (CNNs) in the past demi-decade or so, monocular object localization i.e. reconstructing a moving vehicle from a single moving camera, has become one of the hot topics in computer/machine vision.

In this section, we briefly review the relevant literature and contrast it with the proposed approach.

Shape Priors

The underlying idea of using a shape prior is that the shape of any instance from a category can be represented as a linear combination of the deformations of the category's mean shape along certain directions, called *basis vectors*. This idea was first proposed by Cootes et al. [17], where the shape of a category is represented by a set of 'landmark' points (which we refer as keypoints) and the basis vectors are the dimensions along which the shape is deformed to fit the target model of that category. Ever since then, shape priors have been widely used in a variety of applications.

For e.g. [55, 54, 46, 31, 30] use shape prior to ease the task of object reconstruction from a monocular camera. This linear subspace model was used to formulate a stochastic hill climbing problem in [55, 54] to estimate the shape and pose of a vehicle in a single image. However, this is prohibitively slow to be used in real-time.

On the other hand, these priors have also been explored in the context of human face shape and expression estimation and facial feature tracking. For e.g. [50, 5, 29, 6, 39] estimate the shape while [9] uses shape priors for facial expression analysis. Additionally, [25] uses shape priors for successfully inferring the human body pose.

Keypoint Localization

Recent successes in monocular pose-shape estimation from a single image can be credited to the availability of deep keypoint localization architectures. One of the earlier methods for keypoint detection and localization has been presented in [47]. Keypoint estimates from two different scales are composed in conjunction with a viewpoint prior to generate keypoint likelihoods across the image. However, the response maps from the Convolutional Neural Network were highly multi-modal. As a consequence, accuracy suffered.

In [44, 35], finetuning subnetworks were proposed to refine the estimates from a coarse-grained regressor. In [27], intermediate shape concepts are fed in for a better supervision of the learning process.

Recently, stacked hourglass networks [32] have been proposed for the task of keypoint localization for human pose estimation. These networks are, by construction, multi-scale and possess an iterative refinement nature. One such architecture with spatial constraints among keypoints was used by [31]. The architecture consisted of 8 *stacked* encoder-decoder modules with skip connections across corresponding downsampling and upsampling layers in one stack. We improve upon the architecture proposed in [31] by reducing the number of hourglass stacks and report superior results on a vast validation set consisting more than double the number of keypoints as proposed in [31]. The reduction in number of stacks also results in faster computation at test time.

Monocular Localization in Urban Driving Scenarios

Estimating the 3D shape and pose from a single image has attracted a lot of interest in recent years, supported with the availability of datasets like KITTI [21], Cityscapes [18], NuScenes [7], Robotcar [28] and ShapeNet [8]. In this section we will discuss the prior art on monocular localization of objects in the context of urban driving scenario regardless of the technique used. The approaches discussed below can be broadly classified into two categories – detailed reconstruction and cuboid level localization.

Detailed reconstruction of objects: Approaches in this category follow a 3D-2D pipeline that involves modeling the 3D shape of the object offline and then solving for the 3D deformation in it using the corresponding semantic keypoints that are localized via a CNN. [30, 31, 34, 19, 4, 53] use shape priors to estimate the shape and pose of the object. It has been demonstrated in [31] that having a richer representation for the vehicle (3D wireframe), significantly boosts localization accuracy.

Practically, [30] is the closest to us with one disadvantage that it assumes the vehicle to be reconstructed and the ego vehicle to share the same road plane. The 3D shape of a vehicle is modeled using a shape prior based on a linear subspace model and deformation coefficients are estimated by solving an optimization problem with 2D keypoints, localized using a CNN. [31] is very similar to [30] just that it uses multiple observations of the same vehicle to stabilize the shape estimate. [34] proposes an approach that is agnostic to texture of the object image. [19] proposed a multi-view approach where the images have been captured by multiple cameras with small overlaps. They introduce two novel methods namely, Cross Projection Optimization (CPO) and Hierarchical Wireframe Constraint (HWC). The HWC is used in the iterations of CPO process to produce pose and shape estimates of quality surpassing the ones obtained from existing monocular and stereo methods. However, it is not an on-board approach which is the requirement for reconstructing vehicles in the context of autonomous driving. [4] proposes an end-to-end approach based on keypoints for detection of 2D bounding boxes, keypoints, and orientation, and full 3D pose from a single RGB image. They propose a multi-branch model around 2D semantic keypoints and complement it with simple geometric reasoning. Similar concept was used in [53] for 3D human pose estimation using a sequence of monocular images; here 3D pose is represented as a linear combination of predefined basis poses.

Cuboid level localization: In this section we discuss those approaches which localize objects on a cuboid level. This representation of 3D objects although not very rich in nature, does serve the purpose.

In [41] and [40], the authors develop a real-time monocular SfM system using information from multiple image frames. Mono3D [11] trains a CNN that jointly performs object detection in 2D and 3D space and estimates oriented bounding boxes for vehicles. Although it outperforms stereo competitors, it assumes planar road surfaces. CubeSLAM [52] proposes an approach to estimate high-quality cuboid proposals from 2D bounding boxes and vanishing points sampling. These proposals are then scored and chosen based on how well they align with the image. They also use multiple views for refining the estimates. MonoGRNet [36] proposes a single unified network that relies on geometric reasoning to produce 3D cuboid estimates for the vehicles in the scene. [12] first generates a set of candidate class-specific object proposals, which are further run through a standard CNN pipeline to obtain high-quality object detections. [51] presents an end-to-end CNN based approach for the task in discussion. Segment2Regress [14] exploits a very widely accepted assumption that the vehicles move on road i.e. they stand on the road surface, and propose a two-stage approach consisting of a segment network and a regression network. Just like [11, 51], [4] also proposes an end-to-end approach, just that it is based on keypoints for detection of 2D bounding boxes, keypoints, and orientation, and full 3D pose from a single RGB image. They propose a multi-branch model around 2D semantic keypoints and complement it with simple geometric reasoning. The authors in [49] propose a rather creative solution for 3D localization of the vehicles by lifting the input image to a point cloud representation, which they call pseudoLiDAR point cloud and use LiDAR-based algorithms for 3D vehicle localization.

In Many of the approaches discussed above, the coplanarity assumption plays an important role which we intend to relax. For e.g. [30, 31, 41, 40] rely on the coplanarity assumption for the target vehicle and the ego car. Most of these methods use the approach outlined in [42] to estimate the depth of a vehicle under the coplanarity assumption.

Monocular Road Surface Reconstruction

There is relatively little work on road surface estimation from a monocular camera. In [10], the authors propose a simple road edge prediction framework using edges and lanes detected in earlier frames. No surface level reconstruction is provided. In [20], road width and shape of the drivable area are estimated using a Conditional Random Field (CRF).

In contrast to the above approaches, the proposed approach is independent of the road plane profile of vehicles and is capable of accurately localizing the vehicles. The method outperforms the current best competitor [31] by a significant margin, highlighting how the existing approaches fail to deal with vehicles on arbitrarily oriented road surfaces.

Chapter 3

Background

In this chapter, we discuss on shape priors and its formulation, keypoint localization using convolutional neural networks, monocular single image based pose-shape recovery of moving vehicles using coplanarity assumption, and problem with coplanarity assumption.

3.1 Shape Priors

3.1.1 Motivation

Localizing objects in 3D from a single monocular camera is also not spared from the curse of ill-posedness. This is because a monocular camera is a bearing only sensor, i.e. it can only measure the angle to a point which is being imaged, but not the distance to it. Formally, cameras are projective devices i.e. they produce an image of a point in 3D space onto a 2D image plane following the laws of projective geometry. In projective geometry, a point in the image plane can be considered to be a ray which originates at the center of the camera, C, and passes through the image of the point, p, and the actual 3D point, P, in 3D space. It can be easily seen in Fig. 3.1 that any point on the ray will always project to the same point. Mathematically, we say that any vector p is equivalent to kp where k is a scalar greater than zero.

However, if we had a prior on the 3D shape of the car shown in the image (Fig. 3.1), then it would only require use to perform a 3D-2D based pose recovery of the car w.r.t. the camera. This process is called *resection*, which is schematically shown in Fig. 3.2.

Again, the aforementioned method is only valid if we have a database storing the 3D shapes of all the cars which will be seen when the system is operational. While we can have a database of 3D shapes, it will never be complete. Meaning, it will always require us to keep updating the database whenever a new model comes into existence, making the resection based pose recovery infeasible.

Another solution, which is also used in this thesis, is to model the shape of cars. We know that cars, although unique in shape and size, do follow a common design pattern. If this common design pattern can be modeled then we can represent the entire category by a mathematical model. The following

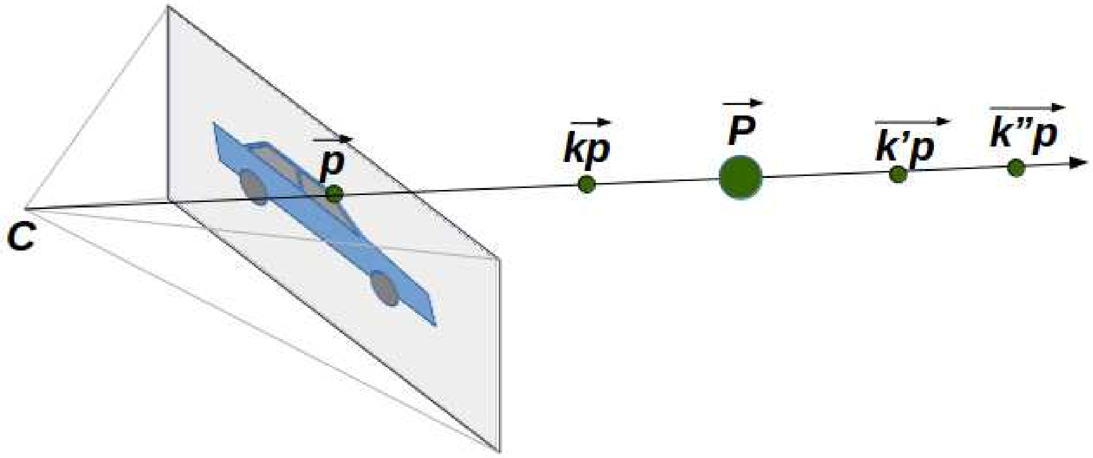

Figure 3.1 This figure schematically depicts the source of scale ambiguity. The vector kp with any value of k (for e.g. k, k' or k'') greater than zero will project to the same location (p) in the image. This is shown in a black line with arrow in the end signifying its length is ambiguous up to a positive scalar.

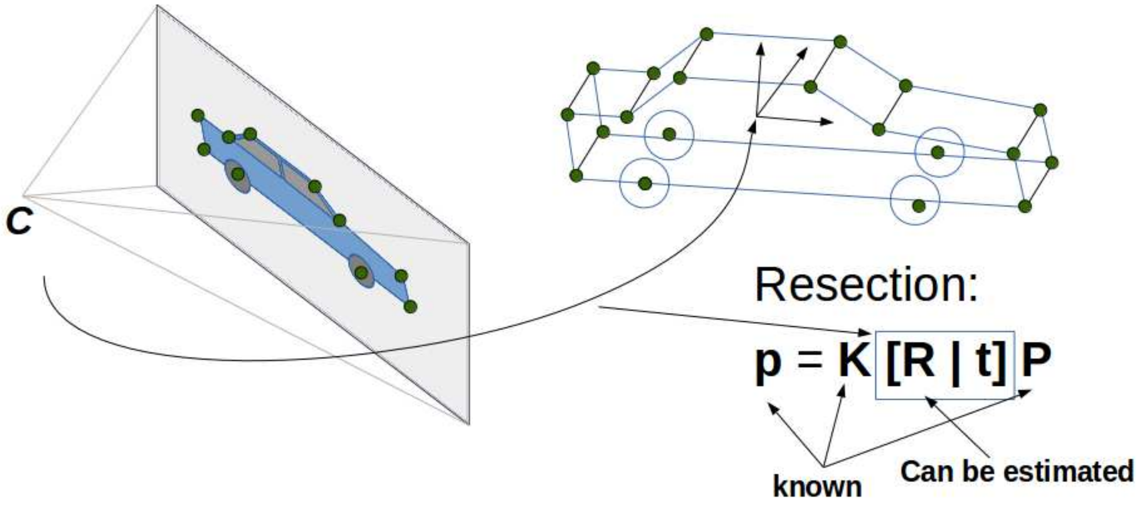

Figure 3.2 In this figure, P is a the 3D point of the known car model represented in the car coordinate frame and p is the corresponding projection on the image plane defined in the image coordinate frame. K is the camera matrix which is assumed to be known here. When we have enough number of such 3D-2D correspondences, a resection process aims to estimate the 6DoF localization (3DoF for rotation and 3DoF for translation) of the camera frame with respect to the vehicle frame.

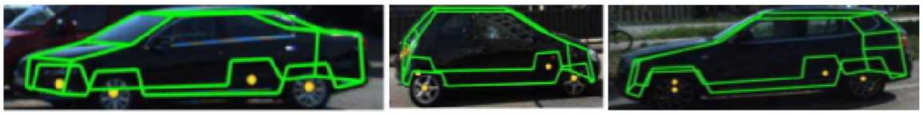

Figure 3.3 In this figure we see the capability of the chosen 36 kepoints to adequately model different class of cars – sedan, hatchback, and an SUV. For a better illustration, the keypoints have been connected based on their order to form a wireframe in green color.

sections will discuss the concept of shape prior in more detail and how it can be used to mathematically model the shape of cars.

3.1.2 Formulation

A shape prior is a mathematical model that represents a manifold on which all possible shapes of an object category lies. As studied by Cootes *et al.* in [17], it has to be a deformable model owing to the fact that objects in a category can be of different shapes and sizes, and hence it has to cope up with the intra-category object shape variations(as an example, see Fig. 3.4, where we show different models of the car class). A rigid model on the other hand would fail to completely represent an object category.

We hypothesize that this manifold of all valid shapes is confined to be a very low-dimensional subspace of the space of all possible shapes of all objects. Let us analyze the aforementioned statement from an intuitive perspective. We know that while the shape of each car could be unique, it bears close resemblance to shapes of other instances in the category. Additionally, we note that shape of an object belonging to one category vary a lot from the shape of an object from another category. For e.g., any model from a car class looks very different from any of the objects from the bicycle class.

In this thesis, similar to [55, 13, 46], we characterize the shape manifold of car category as a linear subspace model [45], which will be discussed in detail later in this chapter.

3.1.3 Semantic Keypoints

To represent shape, one could use a dense point cloud comprising of the 3D locations of all points on the object. However, such a rich representation with thousands of points is often filled with redundancy and is not suitable for real-time pose and shape estimation. Hence, instead of using all points on the object, we choose a very sparse set of points which are common across all objects in the category. We refer to these points as semantic keypoints, or just keypoints for short. These semantic keypoints are very carefully chosen such that any valid configuration of these points should be sufficient to capture the entire shape of the object without exhibiting any redundancy.

In this thesis, for localization and reconstruction of cars, we represent a car using 36 keypoints as shown in Fig. 3.5. These 36 keypoints mark the corners on the outer body of the car and the wheel centers, which are the typical spots for deformation. The generalizing capability of the keypoints is evident from Fig. 3.3 where it can be seen to adequately capture different classes of a car. For the

Figure 3.4 Collage of different instances of cars.

purpose of better illustration, the keypoints have been connected with respect to their order to form a wireframe in green color.

3.1.4 Definition

Formally, we define the shape of an instance as an ordered collection of 3D locations of the keypoints specific to that particular instance. So, a shape is a K–tuple $(X_1, X_2, ..., X_K)$, where each $X_k (k \in 1..K)$ is a 3–vector. The shape X is then a $3K$–vector.

A *shape prior* consists of two components: a *mean shape* and a *deformation basis*. The mean shape refers to the average of all shapes of the category. The deformation basis refers to a set of linearly independent basis vectors that can be used to express any specific instance from the category as a linear combination in conjunction with the mean shape. Formally, a shape prior is the following linear subspace model.

$$X = \bar{X} + V\lambda \tag{3.1}$$

Using notation from [31], we denote the mean shape (or wireframe) for the vehicle category by $\bar{X} \in \mathbb{R}^{3K}$. The basis vectors are stacked into a $3K \times B$ matrix denoted V. The deformation coefficients (also referred to as the shape parameters) $\lambda \in \mathbb{R}^{B}$ uniquely determine the shape of a particular instance. If we assume that the object coordinate frame has a rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$ with respect to the camera center, any instance X can then be parameterized by the shape prior model as shown below in in Eq. 3.2 which is also pictorially illustrated in Fig. 3.6.

$$X = \hat{R}\left(\bar{X} + V\lambda\right) + \hat{t} \tag{3.2}$$

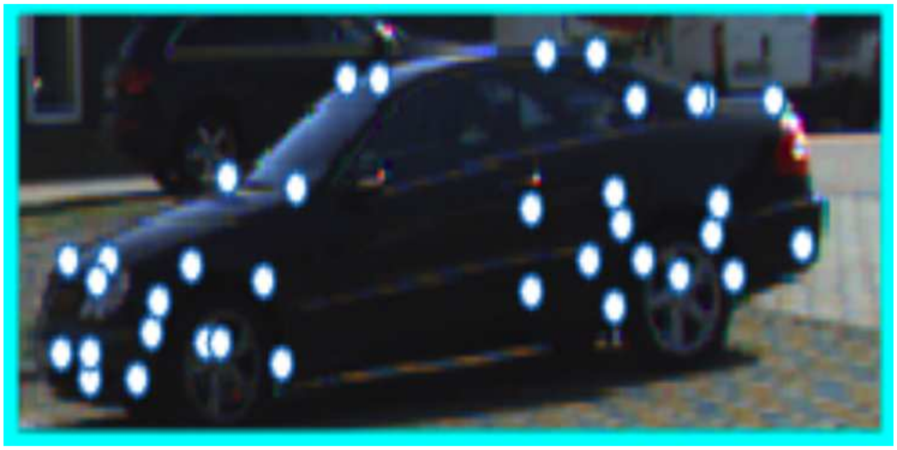

Figure 3.5 This figure shows the 36 keypoints used in our shape representation. The projections of the semantic keypoints are displayed using white circular markers. These semantic keypoints have been carefully chosen to capture adequate variance in the shapes of cars.

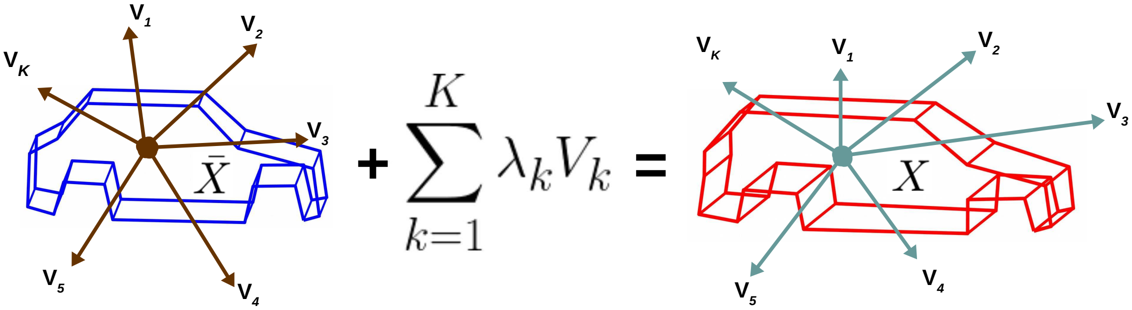

Figure 3.6 Illustrating linear combination of deformations of a mean shape along its basis vectors to produce any other shape in the category

Here, $\hat{R} = diag([R, R, ..., R]) \in \mathbb{R}^{3K \times 3K}$, and $\hat{t} = \left(t^T, t^T, ..., t^T\right)^T \in \mathbb{R}^{3K}$.

$\bar{X} = \left(\bar{X}_1^T, \bar{X}_2^T, ..., \bar{X}_K^T\right)^T$ is an ordered collection of the 3D locations of the keypoints in the mean wireframe.

If we denote the locations of an ordered collection of 2D keypoints by $\hat{x} = \left(\hat{x}_1^T, \hat{x}_2^T, ..., \hat{x}_K^T\right)^T \in \mathbb{R}^{2K}$, the pose (R, t) and shape (λ) of the vehicle can be obtained by minimizing the following objective function in an alternating fashion - once for pose, and once for shape.

$$\min_{R,t,\lambda} \mathcal{L}_r = \|\pi_K \left(\hat{R}\left(\bar{X} + V\lambda\right) + \hat{t}; f_x, f_y, c_x, c_y\right) - \hat{x}\|_2^2 \tag{3.3}$$

$\pi_K()$ is a vectorized version of the perspective projection operator, which takes in K 3D points and computes their image coordinates, given the camera intrinsics $\mu = (f_x, f_y, c_x, c_y)$. Specifically, π_K is the following function.

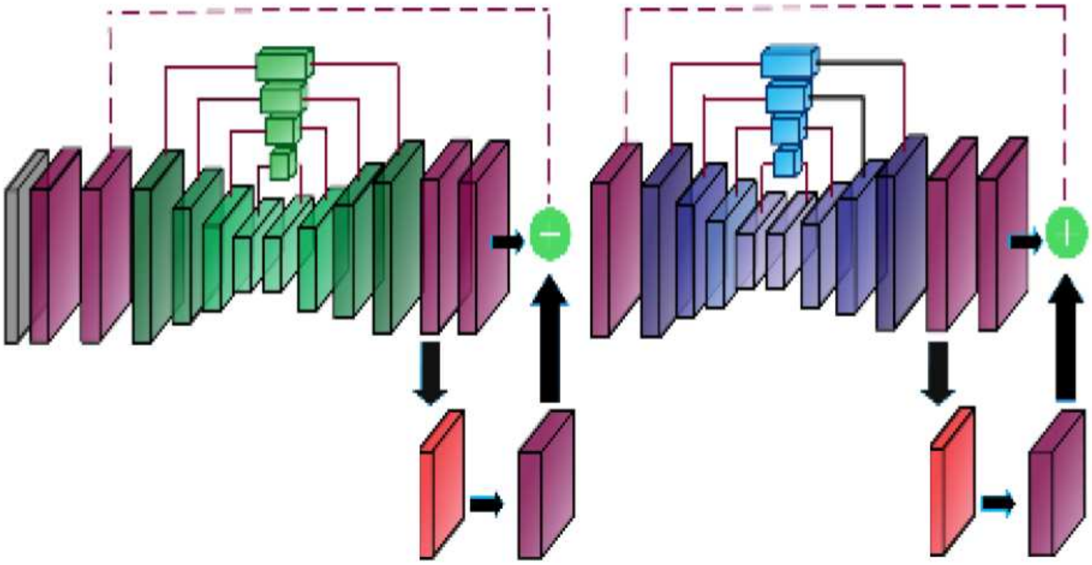

Figure 3.7 Proposed network architecture. The proposed network comprises of 2 stacks of hourglass modules. This reduces the number of parameters significantly with superior performance on a large set of 36 discriminative keypoints.

$$\begin{aligned}\pi\left((X,Y,Z)^T;\mu\right) &= \begin{pmatrix}\frac{f_xX}{Z}+c_x\\ \frac{f_yY}{Z}+c_y\end{pmatrix}\\ \pi_K\left((X_1^T,...,X_K^T)^T;\mu\right) &= \left(\pi(X_1;\mu)^T,...,\pi(X_K;\mu)^T\right)^T\end{aligned} \tag{3.4}$$

3.2 Stacked Hourglass Network for Semantic Keypoint Localization

In recent times there has been a tremendous advancement in deep learning for computer vision tasks like reasoning about object properties, such as viewpoints, shape, etc., from a single image [43, 22]. One such tasks is of localizing discriminative keypoints across various object categories. In [32], human pose estimation (localization of discriminative human body parts) is done using a novel encoder-decoder style architecture. In [31], the authors train a hourglsas network with 8 stacks to predict 14 discriminative set of keypoints on vehicles. Inspired by this, we train a network with 2 stacks of hourglass modules, thereby reducing the number of parameters significantly with superior performance on a large set of 36 discriminative keypoints. Specifically, the network takes as input an RGB detection of a car of 64×64 resolution and regresses on the location of the keypoints by predicting K heatmaps, where $K = 36$ in our case. For each keypoint, the location in the 64×64 output heatmap is chosen as the one with highest activation. The network architecture, as shown in Fig. 3.7, consists of two *hourglass modules* stacked one after another. Each *hourglass module* consists of an encoder-decoder style architecture with

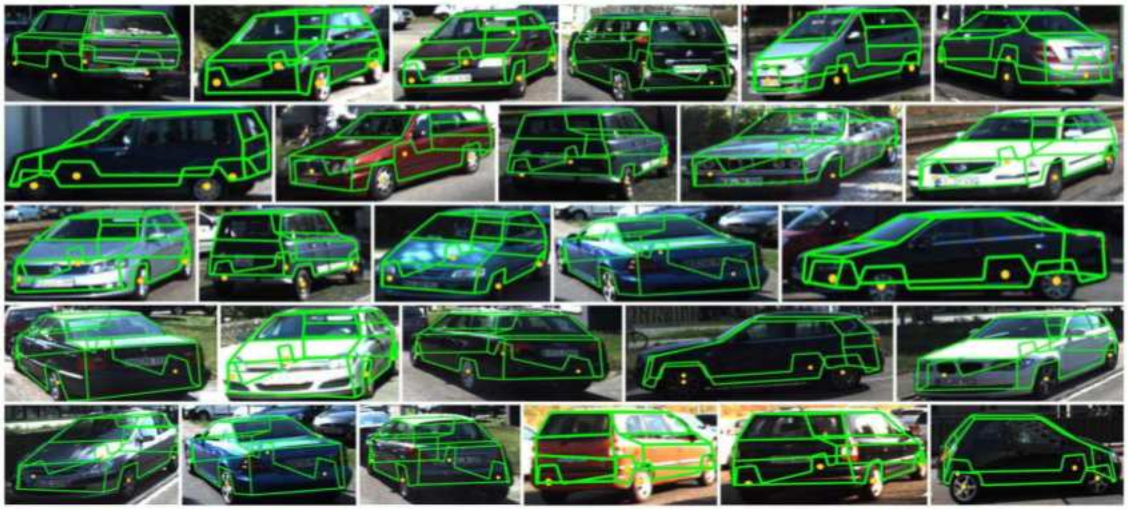

Figure 3.8 A collage of different types of cars with the corresponding wireframe overlaid on them. The wireframes were generated by connecting the detected keypoitns by the proposed stacked hourglass network

skip connections. The network predicts an output of resolution $K \times 64 \times 64$: one for each semantic keypoint. The corresponding ground truth is of same spatial dimension. The loss function used is Mean Squared Error function on the predicted and the label heatmaps, with Adam optimizer [26] and learning rate of $1e - 5$. The network is trained on images obtained from RenderForCNN [43] deployed on ShapeNet [8] CAD models. The accuracy of the proposed 2D keypoint detection network is evaluated using the standard PCK (Percentage of Correct Keypoints) and APK (Average Precision of Keypoints) metrics as in [31]. For the determination of correctness of keypoint estimates, we use a very tight threshold of 2 pixels. Our trained keypoint model achieved a PCK measure of 96.89 at $\alpha = 0.1$ APK on the aforementioned validation set. The network was deployed on KITTI [21] and SYNTHIA-SF [24] datasets. The results of the network on a variety of vehicle models is shown in Fig. 3.8.

3.3 Shape Prior Based Reconstruction of Vehicles

Simultaneous recovery of pose and shape of dynamic vehicles is an ill-posed problem because of the very fact that by the time second image has been taken for reconstruction, the vehicle would have moved, resulting into inaccurate triangulation of the 2D points on the vehicle. This problem is schematically shown in Fig. 3.9.

However, one should note that the problem is no more ill-posed if the shape of the vehicle or the pose of the moving camera is known. But, in our case, none of the two is known. Shape priors have been widely used in many works [30, 31, 34, 19, 4, 53] to tackle the ill-posedness of the problem.

Here we will briefly discuss the work of Murthy et al. [30] as it is the closest approach to ours. For details, the reader is requested to refer [30]. The authors' argument is based on the premise that we

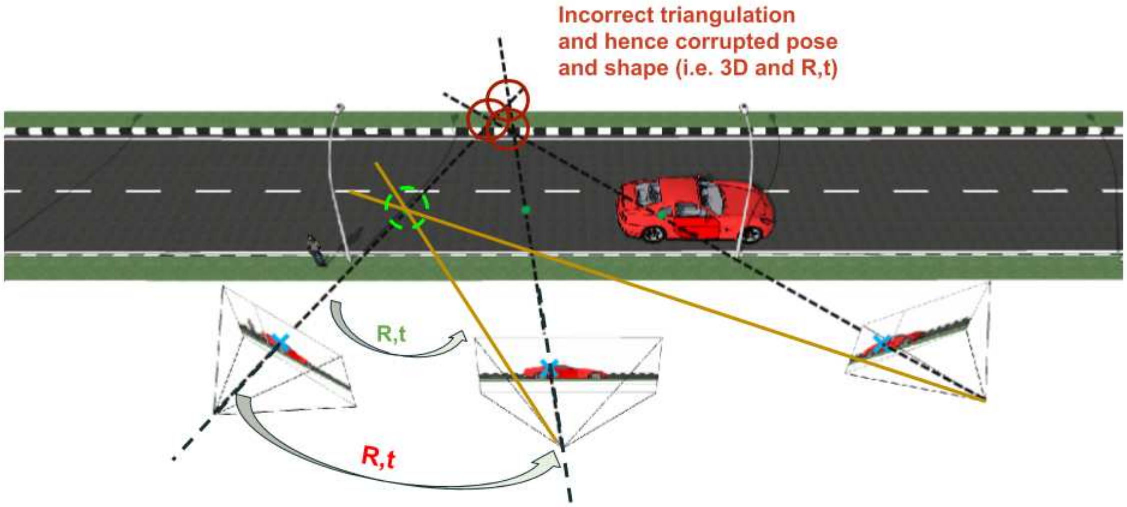

Figure 3.9 This figure shows why conventional triangulation based reconstruction of dynamic vehicles from a moving monocular camera fails. In this example, the moving camera captures three frames. If the object, in this case the car in red, had been stationary then triangulating the interest points would have allowed us to compute the correct pose of the camera and shape of the vehicle. This technique of estimating camera pose and structure of the rigid scene is know as SfM (Structure from Motion). However, as the vehicle is moving, it results in wrong triangulation leading to inaccurate pose of the camera and consequently distorted reconstruction of the vehicle

humans are capable of perceiving the 3D shape of object from a single image. This is possible because humans have seen a large number of such known objects and have an idea how do they look in 2D. Using this prior understanding of shapes and how they look, we can efficiently infer the reverse process.

Murthy et al. [30] first detect 14 semantic keypoints of a detected car using a trained CNN-cascade. The cascade architecture consists of small, keypoint specific subnetworks. Once the semantic keypoints have been robustly localized, a pose adjustment and shape-aware adjustment routines are alternatively performed to estimate accurate pose and shape of the vehicle. The pose and shape adjustment routines are conventional Bundle Adjustment like cost functions that tend to minimize the reprojection error while optimizing for the pose and shape parameters. The two non-linear cost functions have to be optimized in an alternating fashion because as discussed above, simultaneous recovery is ill-posed.

As the cost functions which are being optimized in the pose and shape adjustment phases consist of rotation matrices, they are highly non-linear. This requires the optimization routine to be initialized with a good initial guess for the parameters. During pose adjustment, these parameters are the rotation matrix, R, and the translation vector, t, of the vehicle w.r.t the camera. The vehicle's pose is initialized using the coplanarity assumption and camera height prior.

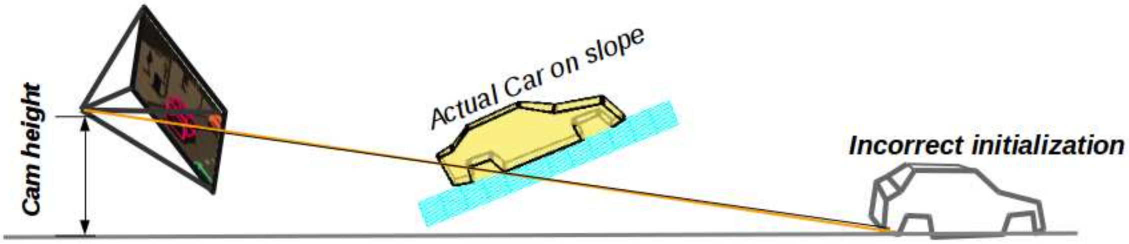

Figure 3.10 Schematic description of the problem arising from vehicle pose initialization using the coplanarity assumption. As the actual car is resting on a road plane which is different from that of the ego vehicle on which the camera is mounted, initialization of the vehicle pose using the coplanarity assumption and camera height prior leads to erroneous pose and shape estimates of the vehicle.

3.4 Initializing Vehicles using Coplanarity Assumption and its Caveats

It is very important to have a good initial guess for the parameters which are being estimated in functions which are highly non-linear in nature, such as the ones which are used to estimate the pose and shape of vehicles from a single image. In these scenarios, coplanarity assumptions serves as an easy way to initialize the pose of vehicles. As the cars are moving on road, the corresponding 2D detection bounding box's lower boundary is supposed to lie on the road plane. Now, given that we have very tight bounding boxes, the center of the lower boundary of the detection box would lie very close to the rear end of the car on the road. This information, in conjunction with the camera height prior, can be used to estimate a fairly accurate initial translation of that vehicle. And the initial rotation estimate can be acquired from a view point network which are capable of recovering the orientation of the vehicle w.r.t. the camera.

However, when the coplanarity assumption is violated, then the initialization could be quite different from the actual pose of the car leading the optimizer to get stuck in false *local minima* and hence resulting in inaccurate pose and shape estimation. This caveat in the coplanarity assumption based pose initialization has been schematically depicted in Fig. 3.10.

Chapter 4

Learning the Shape Prior

In this chapter, we discuss on learning the shape prior of cars using a small set of carefully chosen 3D models from ShapeNet [8]. To learn the shape prior, we need to annotate the semantic keypoints so that we can model the variations. However, we have found that annotating models in 3D is much more difficult than annotating their 2D images. Therefore, we make use of Blender [16] to render the 3D models into multiple images, annotate them in in 2D and then produce the 3D annotations using the process of multi-view reconstruction. Once we have the 3D annotations, we can then capture the statistics of the class to learn the shape prior. Each above mentioned step is further discussed in detail in the following sections.

4.1 Rendering 3D Models into 2D Images for Annotation

Firstly, we carefully select a set of 945 car models from ShapeNet [8], such that they have a good mix of the three car class types, viz. sedan, hatchback, and SUV. These models are in 3D mesh form and annotating them is quite difficult and tiresome. Therefore, we choose to render each model into three images from different, but known, camera positions using Blender [16]. The camera positions are chosen such that the camera always sees the left side of the car. This is done to reduce the annotation effort. As we know that cars are symmetric in nature, reconstructing one side is enough to reconstruct the entire model. Fig. 4.1 shows the three rendered images of a sample from each of the three car class types. The rendering pipeline of Blender [16] also helps us render the models with different colors and varying lighting conditions so that real world situation can be simulated.

Once we have the rendered images, we manually annotate the 18 semantic keypoints (there are 36 keypoints in total) as shown in Fig. 4.2 for each image of all the models. These keypoints are annotated in a designated order and their positions are chosen in a way that maximum deformation happen about those positions. This is done to make sure that the learned model represents the entire category in an unbiased way.

Details on camera positions, coordinate transform, and reconstruction will be discussed in the following section.

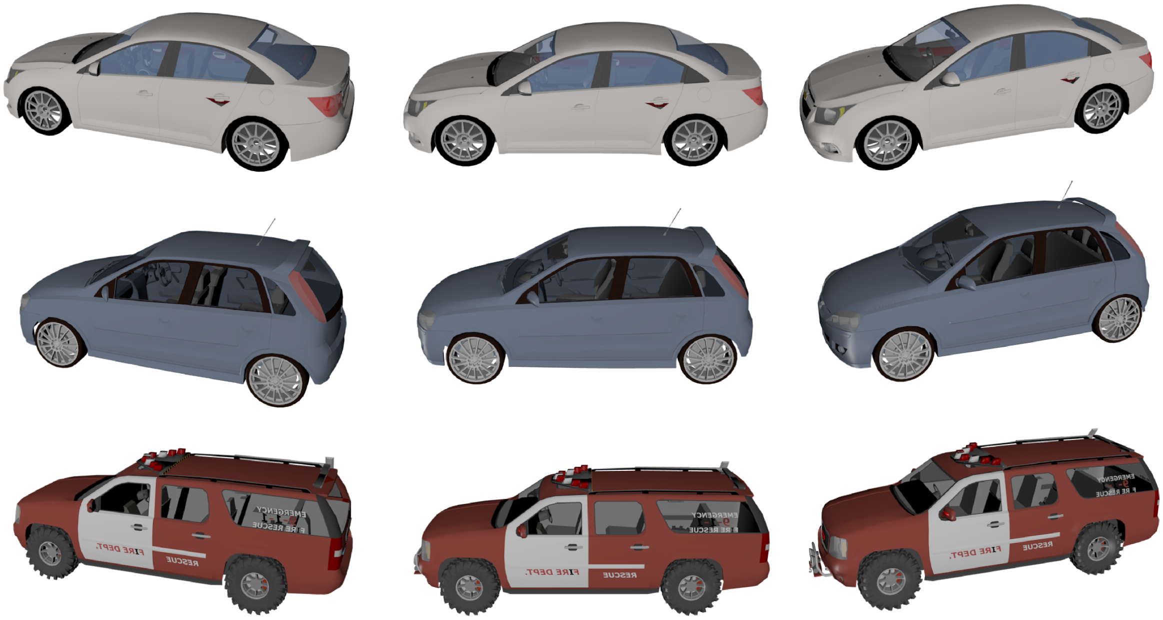

Figure 4.1 This figure shows the three rendered images of a sample from each of the three car class types namely, sedan, hatchback, and SUV (in order from top to bottom).

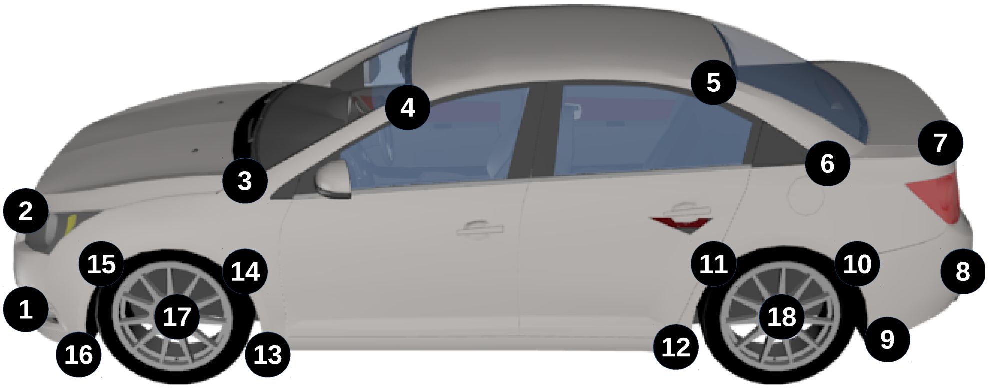

Figure 4.2 Here we show the locations of the semantic keypoints on the left side of the car. As cars are symmetric, annotating and then reconstructing only one side is sufficient to obtain all the 36 keypoints of the car

4.2 Obtaining 3D Annotations by Multi-view Reconstruction

Three different images are taken at three different azimuth angles of $90°$, $70°$ and $110°$ respectively and an elevation of $20°$ is maintained for each camera. The aim is to obtain the appropriate projection matrix for each of these images which will be used to reconstruct the semantic keypoints using triangulation. The obtained 3D locations of the semantic keypoints will be further optimized using bundle adjustment to smooth out the noise occurring from manual annotation.

Before we can reconstruct the models, we will have to align the coordinate frames of the camera and ShapeNet [8]. The initial orientation of the object coordinate system and the canonical camera system is illustrated in Fig. 4.3:

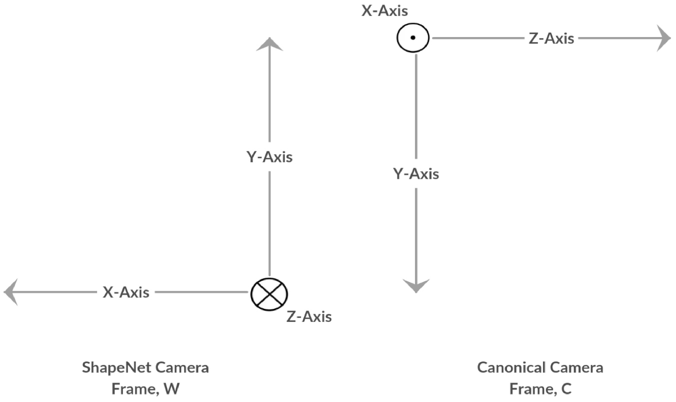

Figure 4.3 Initial axes orientation of the object and canonical camera coordinate system

Given this set of orientation, what we attempt to find now is 'R_c^w' for each image. For this, we must first find the required sequence of rotational transforms that would align the world coordinate system to the canonical camera frame.

We could first try to align the Y-axis of both the systems by rotating the 'W' about its X-axis by $+180°$.

Now, we must apply a rotation of $-90°$ on the above intermediate coordinate system about its current Y-axis to align it with the camera coordinates. Hence, we see that the sequence of rotational transformations applied so far are: $R_x(+180°) * R_y(-90°)$. Moreover, as mentioned initially, in each image, an elevation on $20°$ is maintained. This is obtained by incorporating one more rotational transformation about its current X-axis of $+20°$. We thus obtain the final sequence of rotational transformations as:

$$R_x(+180°) * R_y(-90°) * R_x(+20°)$$

This sequence is common to all the three images. What differs between the three images is the azimuth angle maintained in each of them which are $90°$, $70°$ and $110°$ respectively. To obtain this, an initial rotation about its fixed Y-axis is made by the respective angles before applying the above mentioned sequence of rotations. Hence, we obtain the three cumulative rotational transformations for the three images as follows:

$$R^w_{c1} = R_y(+90^\circ) * R_x(+180^\circ) * R_y(-90^\circ) * R_x(+20^\circ) \tag{4.1}$$

$$R^w_{c2} = R_y(+70^\circ) * R_x(+180^\circ) * R_y(-90^\circ) * R_x(+20^\circ) \tag{4.2}$$

$$R^w_{c3} = R_y(+110^\circ) * R_x(+180^\circ) * R_y(-90^\circ) * R_x(+20^\circ) \tag{4.3}$$

Now, moving on to the required translational transformations. A net distance of 1.5 units is maintained from the object in each image by the camera at an elevation of $+20°$. Hence, this distance is interpreted as $1.5 * cos(+20°)$ and $1.5 * sin(+20°)$ along horizontal and vertical directions respectively with respect to the object plane. This has been schematically depicted in Fig. 4.4

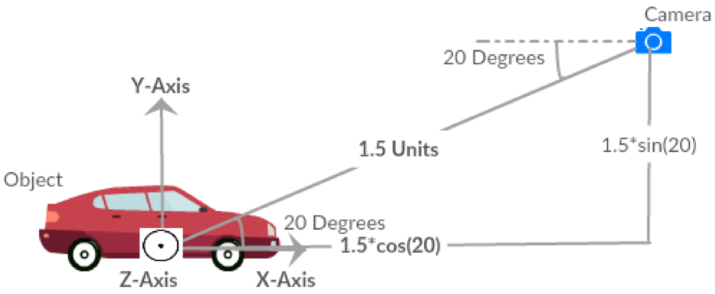

Figure 4.4 Translations about X and Y-Axes with respect to the object frame

Thus, the required translational matrix common to each of the images is obtained as follows:

$$t = \begin{bmatrix} 1.5 * cos(+20^\circ) \\ 1.5 * sin(+20^\circ) \\ 0 \end{bmatrix}_{3X1} \tag{4.4}$$

The azimuth angle has to be taken into consideration for each image resulting in the following translational transformation for each image:

$$t^w_{c1} = R_y(+90^\circ) * t \tag{4.5}$$

$$t^w_{c2} = R_y(+70^\circ) * t \tag{4.6}$$

$$t^w_{c3} = R_y(+110^\circ) * t \tag{4.7}$$

Given the above rotational and translational transformations, we obtain the required homogeneous matrix for each image as:

$$H_{c1}^{w} = \begin{bmatrix} R_{c1}^{w}{}_{3X3} & t_{c1}^{w}{}_{3X1} \\ 0_{1X3} & 1_{1X1} \end{bmatrix}_{4X4} \tag{4.8}$$

$$H_{c2}^{w} = \begin{bmatrix} R_{c2}^{w}{}_{3X3} & t_{c2}^{w}{}_{3X1} \\ 0_{1X3} & 1_{1X1} \end{bmatrix}_{4X4} \tag{4.9}$$

$$H_{c3}^{w} = \begin{bmatrix} R_{c3}^{w}{}_{3X3} & t_{c3}^{w}{}_{3X1} \\ 0_{1X3} & 1_{1X1} \end{bmatrix}_{4X4} \tag{4.10}$$

Now, we move into triangulation process. For this process, we require to obtain the appropriate projection matrix for each image. Here, we take the projection matrix of 'C_1' as a reference and find the projection matrix for 'C_2' and 'C_3' with respect to that of 'C_1'. Here, the transformations are inverted since we are finding the coordinates with respect to the camera. Hence, we obtain the projection matrix for 'C_1' as:

$$P_1 = K * (H_{c1}^{w})^{-1} \tag{4.11}$$

Here, 'K' is the camera intrinsic parameter. Since 'C_1' is taken as reference for other cameras, the above term is taken as '$K * [I|0]$', where 'I' is the identity matrix and '0' is the zero matrix. When we do this, we will have to find the cumulative homogeneous matrix for 'C_2' and 'C_3' with respect to 'C_1' which could be obtained as:

$$H_{c1}^{c2} = (H_{c1}^{w})^{-1} * H_{c2}^{w} \tag{4.12}$$

$$H_{c1}^{c3} = (H_{c1}^{w})^{-1} * H_{c3}^{w} \tag{4.13}$$

We could use the above matrices for finding the projection matrix for 'C_2' and 'C_3' with respect to 'C_1' in the following way:

$$P_2 = K * (H_{c1}^{c2})^{-1} \tag{4.14}$$

$$P_3 = K * (H_{c1}^{c3})^{-1} \tag{4.15}$$

Equations (4.11), (4.14) and (4.15) provide us with the required Projection matrices 'P_1', 'P_2' and 'P_3'. Once we have the projection matrix we use these to triangulate and reconstruct two set of 3D semantic keypoints: one using the image rendered from the view point with 70° azimuth and another take from the view point with 110° azimuth with the image taken from 90° as the reference frame. As the keypoints were annotated manually, they are bound to have noise in the semantic keypoint locations and hence we further refine the reconstructed keypoints using bundle adjustment.

4.3 Capturing the Statistics to Model the Shape

Once we have the 3D locations for all the models from above mentioned process, we intend to learn a model which best explains the category i.e. a model which defines the space in which every point is a valid car. As we have discussed above, we would like to model the space as a linear subspace in which any valid shape of a car can be obtained by the mean shape of the car plus some deformation along the basis vectors.

Mean shape: The mean shape of a car can be obtained by taking the mean of each semantic keypoint over the entire dataset. Given that we have N aligned shapes the mean shape $\bar{X}$ can be defined as a set of the mean of all the keypoints over the entire dataset as follows:

$$\bar{X} = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_{36}\} \tag{4.16}$$

where, $\bar{x}_j$ is given below. The indices i and j belong to keypoint and model, respectively.

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^{N} x_{i,j} \tag{4.17}$$

Basis vectors: As every car shape is represented by an ordered set of 36 semantic keypoints which are each defined in 3D, the dimensionality of the car shape space is 36 $\times$ 3. This means that the car shape resides in 108 dimensions, i.e. we have 108 orthogonal basis vectors. Any linear combination of these 108 basis vectors plus the mean shape gives a valid shape from the car shape manifold. However, looking at different car models it is very evident that cars deform in a specific way and not every keypoint moves substantially. In other words, it means that the deformations happen in a space whose dimensionality is less than 108. This observation can be proved by finding the *eigen vector decomposition* of the data and see which dimensions show good amount of variation. Another alternative to capture the statistics is to do *PCA (Principal Component Analysis)* over the entire dataset. This is the method which we will be following in this work.

PCA gives us the basis vectors, V, and the magnitude of variation – the shape coefficients, λ, in the corresponding directions. After sorting the vectors in the decreasing order of their corresponding magnitude of variation, we select the first few vectors which capture approximately 100% of the variations. In our case, we have chosen 42 vectors using the formula 4.18.

$$S_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{k=1}^{108} \lambda_k} \geq 99.9 \tag{4.18}$$

where, S_i is the index number of the sorted list of shape coefficients. Once we attain approximately 99.9%, we stop at that index number and select all the shape coefficients from 1 to i and their corresponding basis vectors, V. This set of shape coefficients and *basis vectors* constitute our *car shape manifold*. Fig. 4.5 shows the mean shape (right) and a plot of the 42 chosen shape coefficients (λs)

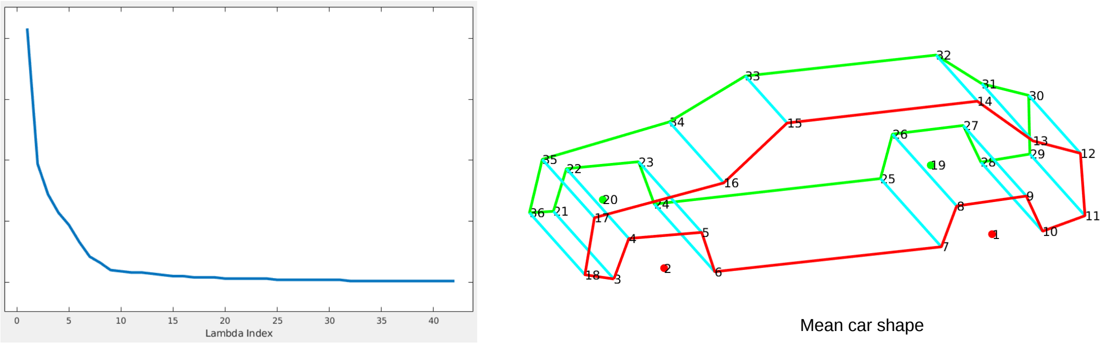

Figure 4.5 $Left$: a plot of the selected 42 the shape coefficients which shows that even though we needed 42 vectors to capture approximately 100% of the variations, majority of the variation happen along very few basis vectors. This shows that deformations happen in a very low dimensional space. $Right$: A 3D plot of the mean shape of the car category with the semantic keypoints labelled according to their order of annotation. The two sides of the car are shown in red and green. The keypoints are connected to form a wireframe for a better illustration.

(left). From the figure it is clear that even after choosing 42 dimensions, substantial variations happen only along very few directions.

Moving on this manifold by taking linear combinations of the basis vectors plus the mean shape will produce different shapes from the car category. And every basis vector in the selected list is responsible for deforming the shape in a specific way. For e.g., Fig. 4.6 shows the independent effect of the first three vectors on the shape of the car when the corresponding coefficient (λ) is varied.

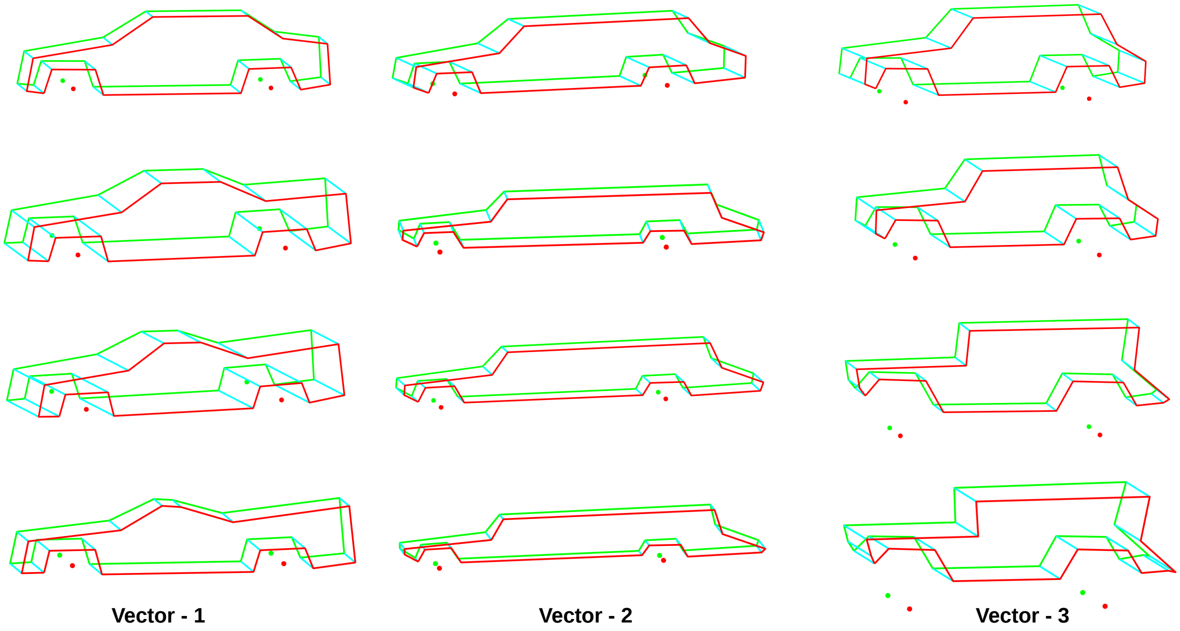

Figure 4.6 Shows the independent effect of the first three basis vectors on the shape of the car when the corresponding shape parameters are varied.

Chapter 5

Reconstructing Dynamic Vehicles on Arbitrary Road Profiles from a Moving Monocular Camera

In this chapter, we discuss on the primary contribution of this thesis which is a joint optimization formulation to recover pose and shape of cars on arbitrary road profiles from a moving monocular camera.

The proposed joint optimization framework jointly optimizes over car's pose-shape and its local road plane geometry. The introduction of road plane geometry in the optimization routine provides multiple constraints such as, the car should be on the road, the orientation of the car should be similar to the orientation of its local road plane, etc.. These constraints, discussed in detail later in this chapter, help in narrowing down of the solution space which results in more stable and accurate localization (and reconstruction). We test our method on KITTI [21] Tracking training sequences as these sequences provide us the ground truth pose of the cars in the scene. However, as KITTI [21] does not have many sequences with steep and graded roads, we also test our method on SYNTHIA-SF [24] dataset, a synthetic dataset simulating steep and graded roads of San Francisco (USA). We compare our results with other relevant methods and show that that the proposed method yields more accurate localization.

5.1 System Setup

We operate on image streams captured by a front-facing monocular (RGB) camera mounted on a car. The height H above the ground at which the camera is assumed to be known a priori (this helps in resolving scale-factor ambiguity in monocular reconstruction).

We assume that, on each incoming image, an object detector [37] runs and detects vehicles in the image (as bounding boxes). We also perform a semantic segmentation of the input image using the SegNet [3] convolutional architecture. The proposed pipeline is illustrated in Fig. 5.1.

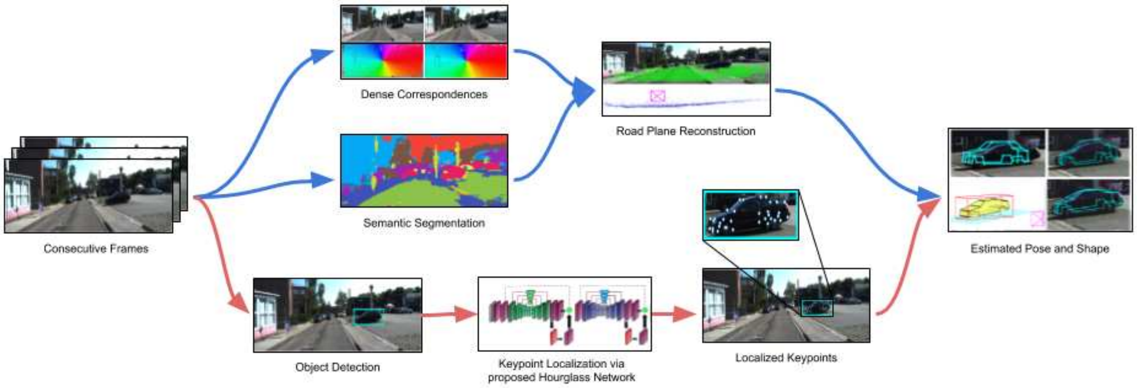

Figure 5.1 Illustration of the proposed pipeline. The system takes three consecutive frames (in case of no lane markers). In the upper half (blue arrows), we illustrate the method for estimating the ground plane i.e. using dense correspondences over the frames and then performing bundle adjustment. In the lower half (red arrows), the detected bounding boxes in each frame are processed using the proposed keypoint localization CNN to obtain 2D locations of a discriminative set of semantic keypoints. The pose and shape of the object are then adjusted by incorporating the estimated ground plane information.

5.2 Reconstruction of Vehicles on Slopes

To formulate a lightweight, yet robust optimization problem for reconstructing vehicles on non-planar road surfaces(i.e. roads with slopes and banks), we assume that the road is locally planar. By this, we mean that the patch of the road that lies exactly beneath a detected vehicle is assumed to be a planar patch. This assumption is corroborated by [41], where allowing each vehicle to have an adaptive local ground plane boosts localization accuracy.

Each detected vehicle v is on a planar patch parameterized by (n_g^{vT}, d_g^v), where n_g^v is a vector that denotes the normal to the planar patch and d_g^v denotes the distance of the planar patch from the origin of the camera coordinate frame.

5.3 Resolution of Scale-Factor Ambiguity

Monocular camera setups inherently suffer from scale-factor ambiguity, i.e., any 3D length estimated from a set of images is accurate up to a positive scalar. But, for the autonomous driving applications, we require that vehicles are localized in *metric scale*, i.e., in real-world units (such as meters, for instance). We resolve scale ambiguity using one of the following two approaches.

5.3.1 Using Dimensions of Detected Lanes

Most roads have lane marking or zebra crossings of standard dimensions that are known to us a priori. We use the method from [38] to detect lane markings, and if we know the height of the camera above the ground and the dimensions of the lane markings, we can retrieve the planar patch comprising the lane marking and the distance to that lane marking (in meters). Such a method estimates the local ground plane (of a lane marking near the vehicle) using information from just a single image.

5.3.2 Using 3-View Reconstruction and Camera Height

The above method can only be employed on roads where there are lane markings and in particular only if a lane marking is detected near a vehicle, which is not true for all scenarios we encounter. In the more general case, we can recover absolute (metric) scale by using the following 3-view reconstruction scheme. Assume we have three consecutive frames f_1, f_2, f_3 with sufficient parallax. We use DeepMatching[48] for establishing dense correspondences between frames f_1 to f_2 (see Fig. 5.4). Then, using a sufficient mix of road and non-road points, we estimate the egomotion between the frames using standard multi-view motion estimation techniques [23]. Using the estimated egomotion, we triangulate points *close*[1] to the car that lie on the road surface (see Fig. 5.5) and add points from frame f_3 to the reconstruction[2]. A local ground plane patch can then be estimated by estimating a dominant plane from the obtained point cloud using a RANSAC-like routine. Once such a plane is obtained, we can scale the reconstruction such that the median of the Y-coordinates of the estimated plane is roughly equal to the height of the camera above the ground (which is assumed to be known during initial setup). Fig. 5.3 schematically shows the above mentioned process.

5.4 Joint Optimization for Ground Plane and Vehicle Pose and Shape Estimation

Equation 3.3 (in Sec. 3.1.4) represents the optimization problem that is solved to estimate the shape and pose of a vehicle from just a single image or from a pair of images whenever available [31]. However, this formulation assumes coplanarity of the ego car and of the object being reconstructed. We illustrate in Fig. 5.2 that drastic errors in localization result when the assumption does not hold and how using the local ground plane circumvents this problem.

We assume that, in the current frame, a set of vehicles $\mathcal{V}$ have been detected by the object detection network [37]. For a particular vehicle $v \in \mathcal{V}$, we let X_i^v denote the coordinates of the i^{th} keypoint of the vehicle in 3D. Also, we parameterize the local ground plane beneath v by its normal vector n_g^v and the distance of the plane from the camera origin d_g^v. Also, we denote by n_c^v the normal of the car. The

[1] We expand the car bounding box by a factor of 1.9 to 2.0, and pick all points from the expanded bounding box that are classified as *road* by SegNet [3].

[2] This is typically done by propagating feature matches from frame f_2 to frame f_3, and running a resection routine to estimate the egomotion between frame f_1 and frame f_3, and then triangulating points from f_3 onto the initial reconstruction [23]

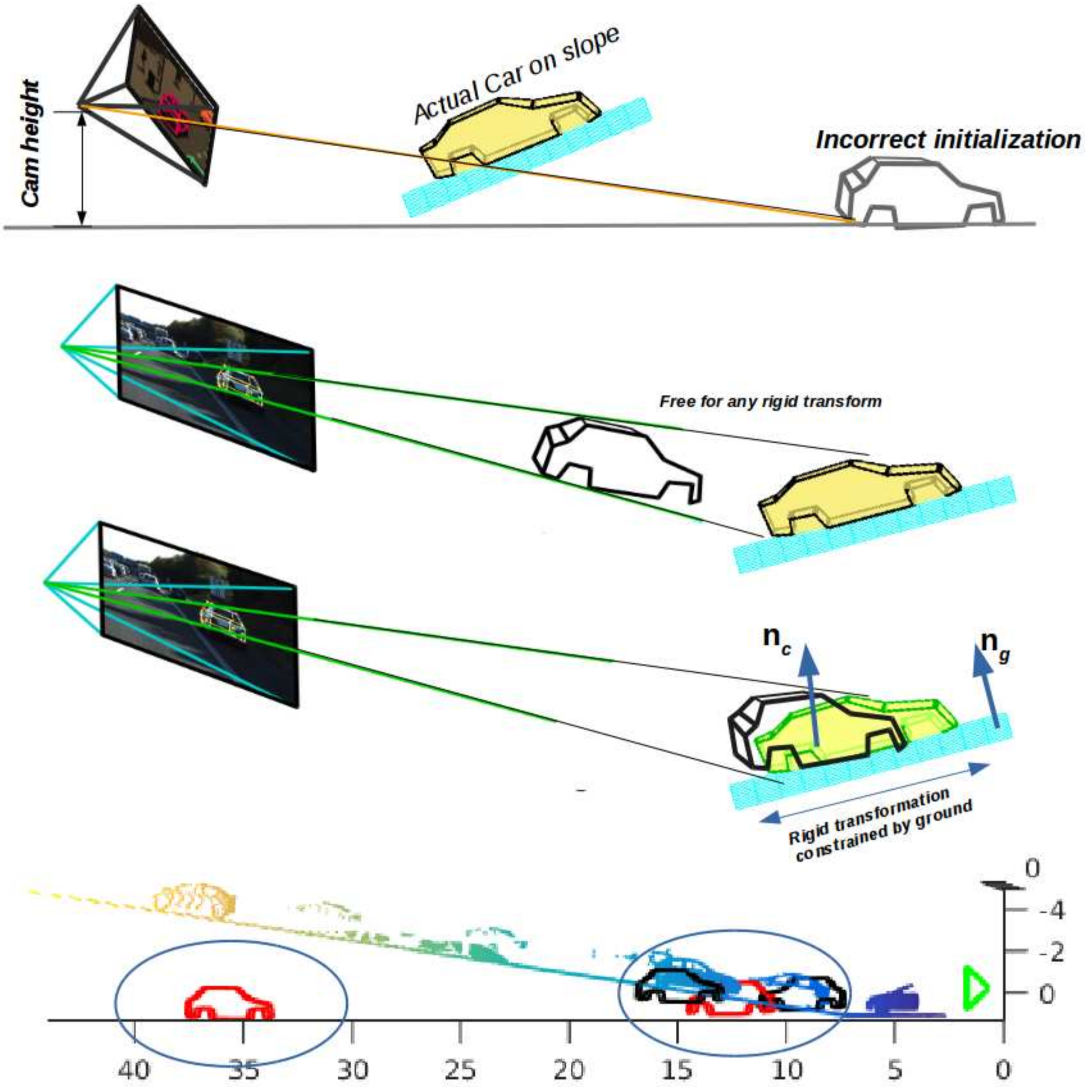

Figure 5.2 *How does ground plane help?* From top to bottom - (i) Illustrating how coplanarity assumption results in incorrect initialization in existing approaches (ii) Relying only on minimizing the reprojection error, leaves the optimizer free to rigidly transform the mean car (iii) Joint optimization constrains the car to be on ground while minimizing the reprojection error, resulting in more accurate reconstruction and localization (n_c and n_g are car base and road plane normals respectively) (iv) Failure of coplanarity assumption for steep roads on SYNTHIA-SF [24]. Notice the incorrect initialization of the car on slopes via method proposed by [31], shown in red. Our method is not bound by this coplanarity assumption and initializes the vehicle correctly, shown in black. We overlay the initialized wireframe on the ground truth 3D points for comparison.

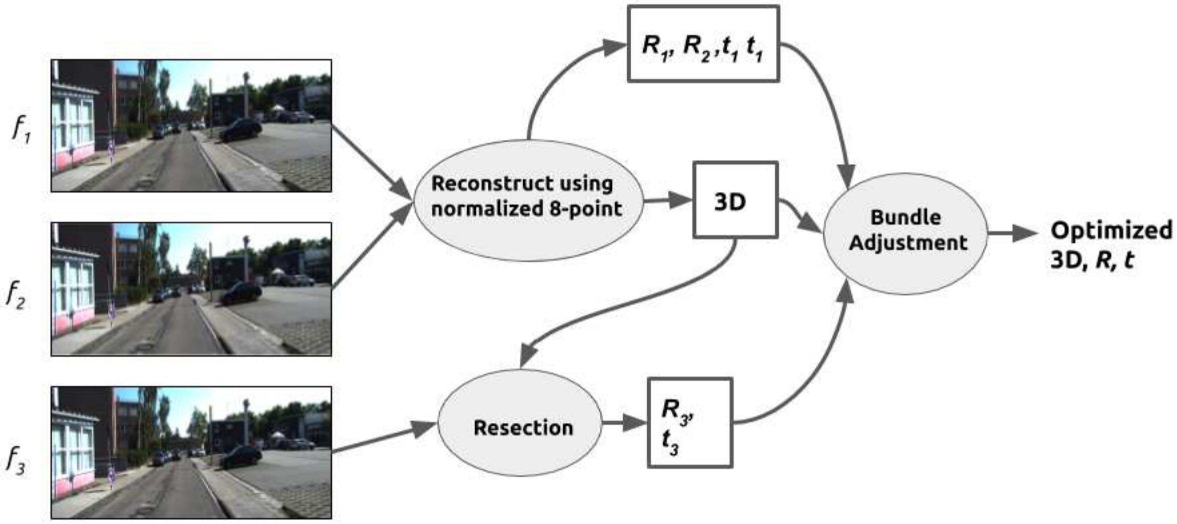

Figure 5.3 Schematic description of 3-view reconstruction. To, start with, the first two views are used to generate the pose of the second camera with respect to the first camera and an initial point cloud using the normalized 8-point algorithm. Then the pose of the third view is computed using resection. Finally, a bundle adjustment routine is run to refine the poses and the reconstruction.

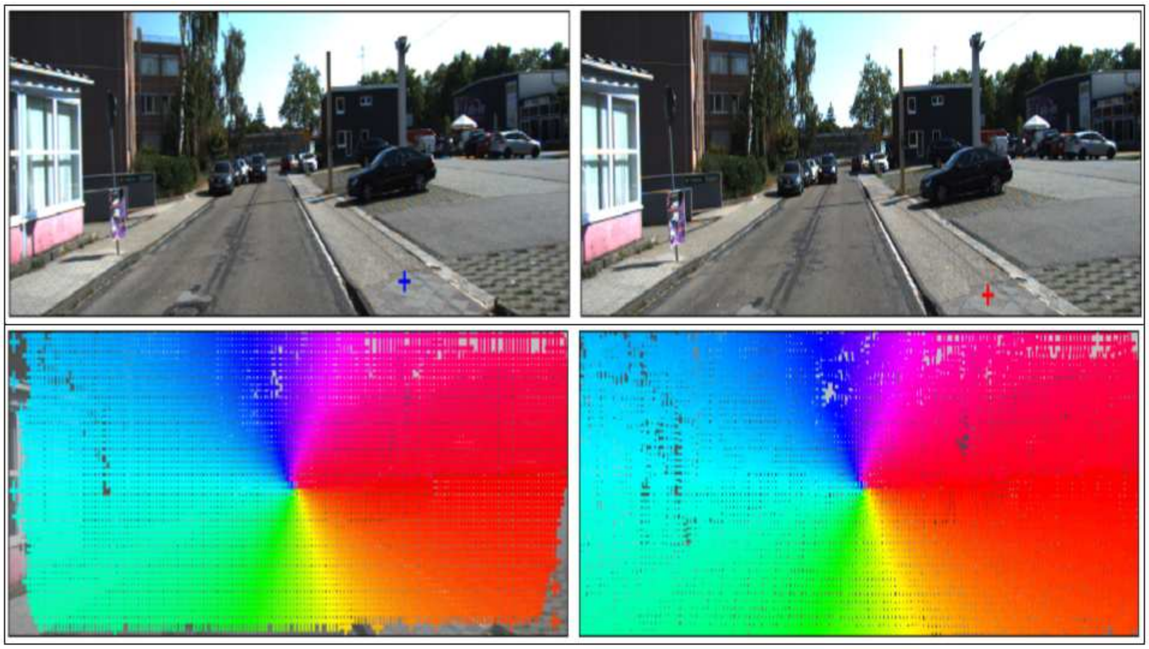

Figure 5.4 The figure shows the result of DeepMatching [48] on two images from KITTI [21] Tracking sequence. The correspondences are shown by same color in the two images in the lower half of the figure.

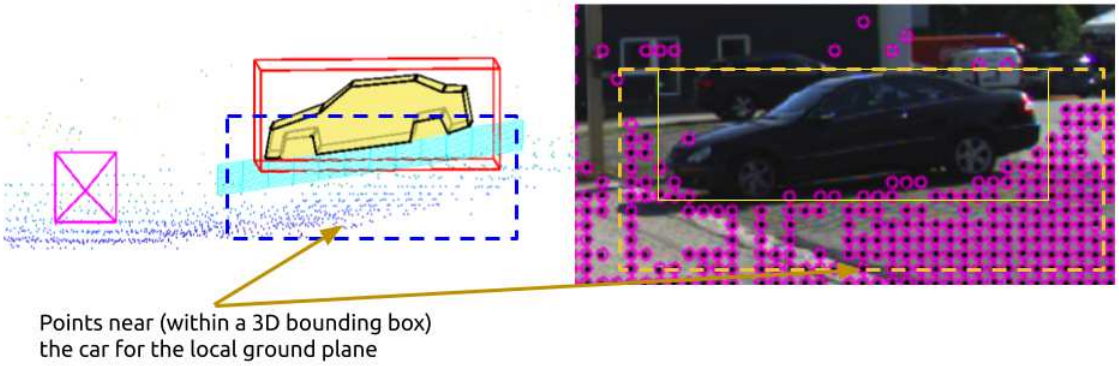
Points near (within a 3D bounding box)
the car for the local ground plane

Figure 5.5 Points from the vicinity of the car's detection bounding box are selected to reconstruct the local road plane on which the corresponding car stands. The vicinity is defined by expanding the detection bounding box towards the lower sides as depicted in the right image by the yellow rectangle drawn in dashed line

normal of the car is defined as the normal of a plane that *best*[3] fits the keypoints corresponding to the wheel centers of the cars.

We now formulate a set of cost functions that relax the coplanarity assumptions in [30, 31] and estimate the vehicle's pose and shape as well as the equation of the ground plane patch beneath it.

5.4.1 Ground Plane Estimation

We define a ground plane estimation loss term, which *encourages* the vehicle to be close to the ground plane. Specifically, we obtain the translation vector t_c^v to the bottom of the vehicle v[4] from the camera center. This, in an ideal setting, represents the position vector of a point on the ground plane, the points of which are denoted as X_g^v. Formally, this term (for all vehicles in the image) can be represented as follows,

$$\mathcal{L}_g = \sum_{v \in \mathcal{V}} \|n_c^v \cdot t_c^v - d_g^v\|^2 \tag{5.1}$$

5.4.2 Normal Alignment

The normal alignment loss term stipulates that the normal of the vehicle (n_c^v) must be encouraged to be parallel to the normal of the estimated ground plane, as shown in Fig. 5.6. An initial guess for the ground plane normal is obtained as described earlier, using either lane markings, or a 3-view

[3] Although, in practice, all 4 wheel centers of a car are coplanar, it may still be numerically hard to determine a plane equation that satisfies all 4 points. So, we fit a plane in the least squares sense to the 4 wheel centers.

[4] We first obtain the rigid-body transform to the origin of the vehicle coordinate frame, and then concatenate to it the rigid-body transformation from the origin of the vehicle coordinate frame to the bottom of the vehicle.

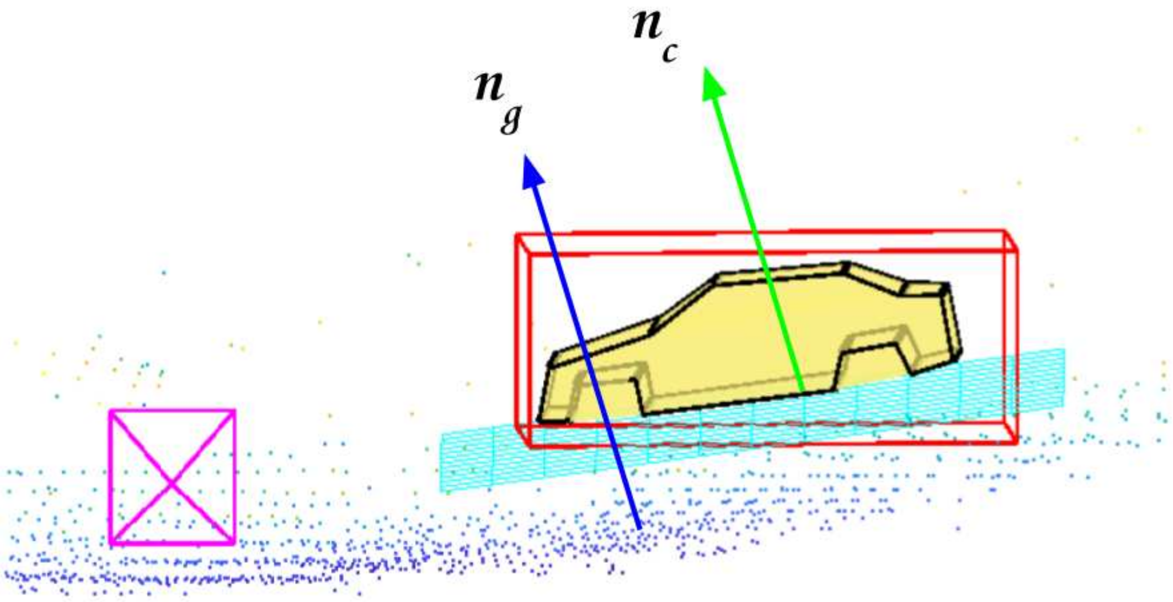

Figure 5.6 This figure shows how the normal of the base plane of the car, made up of all its base points, is parallel to the local road plane normal. This is termed as normal alignment constraint which enforces the the two normals to point in the same direction.

reconstruction. This loss can be denoted as follows. $\times(.,.)$ denotes the vector cross product.

$$\mathcal{L}_n = \sum_{v \in \mathcal{V}} \| \times (n_c^v, n_g^v) \|^2 \tag{5.2}$$

5.4.3 Disambiguation Prior

The above loss term has one drawback in that, it is minimized even when the estimated ground plane and vehicle normals are anti-parallel. To disambiguate such unwarranted solutions, we make use of the fact that even the steepest roads in the world have slopes less than $25\,\mathrm{deg}$ [1]. Whenever multiple solutions are available, we encourage the solution that's *more upright* to have a lower cost. If e_2 denotes the Y-axis of the camera coordinate system (i.e., the axis vertically pointing down), we formulate the disambiguation prior as follows (ϵ is a tiny positive constant that provides numerical stability).

$$\mathcal{L}_d = \sum_{v \in \mathcal{V}} \left\| \frac{-1}{e_2 \cdot n_c^v + \epsilon} \right\|^2 + \left\| \frac{-1}{e_2 \cdot n_g^v + \epsilon} \right\|^2 \tag{5.3}$$

5.4.4 Base Point Priors

We also use a loss term that encourages points along the base of the car (this includes keypoints on the car wheel centers, bumpers, etc) to lie as close to the estimated ground plane as possible (see Fig.

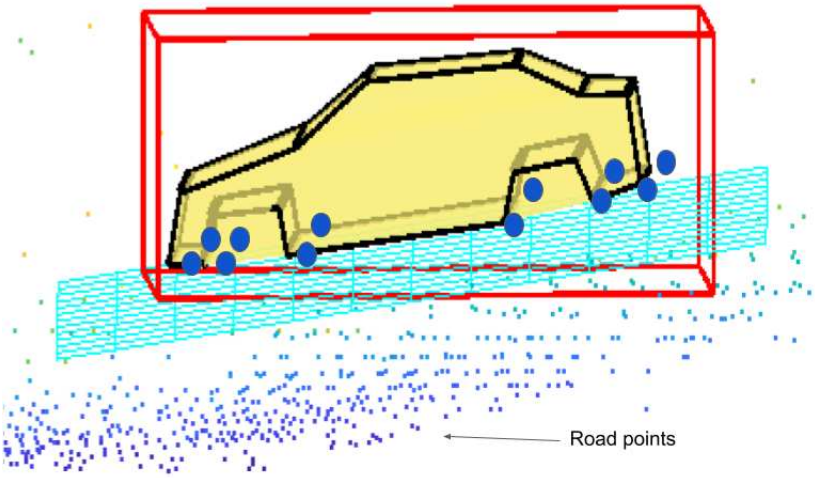

Figure 5.7 This figure shows base keypoints of a car in large blue markers. As it can be seen that they are very close to the road plane, a prior on the closeness can be imposed to constrain the pose/shape of the car leading to an accurate localization.

5.7). If X_b is a keypoint on the car base, and $\mathcal{K}_b$ denotes the set of all keypoints that lie along the base of the car, base point priors are imposed using the following expression.

$$\mathcal{L}_b = \sum_{v \in \mathcal{V}} \sum_{X_b \in \mathcal{K}_b} \| n_c^v \cdot t_c^v - n_c^v \cdot X_b \|^2 \tag{5.4}$$

5.4.5 Global Consistency

Although we assume that each vehicle has its own planar ground patch, it is safe to assume that the road planes of cars which are in close vicinity of each other are not susceptible to abrupt change (see Fig. 5.8). This is encoded into the global consistency loss term, that encourages the planar ground patch of a vehicle to be consistent with that of other vehicles around it. If $\mathcal{V}^n$ denotes the set of all vehicles within a distance d around vehicle v (v is usually chosen to be $5 - 7$ meters), the global consistency loss term is as follows.

$$\mathcal{L}_c = \sum_{v \in \mathcal{V}} \sum_{v^n \in \mathcal{V}^n} \| n_g^v - n_g^{v^n} \|^2 + \| d_g^v - d_g^{v^n} \|^2 \tag{5.5}$$

5.4.6 Dimension Regularizers

We also place priors on dimensions of vehicles that we observe, which provides a well-conditioned problem to work with and leads to better convergence rates. We use regularizers similar to ones proposed in [31], and denote the loss term by $\mathcal{L}_{reg}$.

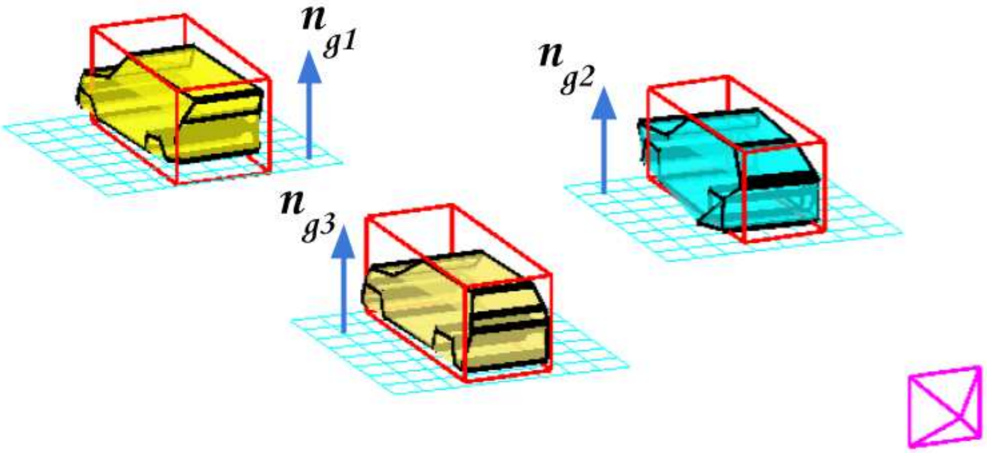

Figure 5.8 This figure shows how cars moving in close vicinity of each other can have similar local road plane parameters and therefore we can impose a global consistency constraint.

5.4.7 Overall Optimization Problem

The overall minimization problem involving all the energy terms can be posed as follows (cf. Eq 3.3 5.1 5.2 5.3 5.5 5.4).

$$\begin{aligned}\min_{R,t,\lambda,n_g^v,d_g^v,n_c^v} \mathcal{L}_{total} &= \eta_r\mathcal{L}_r + \eta_g\mathcal{L}_g + \eta_n\mathcal{L}_n \\ &+ \eta_d\mathcal{L}_d + \eta_b\mathcal{L}_b + \eta_c\mathcal{L}_c + \eta_{reg}\mathcal{L}_{reg}\end{aligned} \tag{5.6}$$

Here, η_r, η_g, η_n, η_d, η_b, η_c, and η_{reg} are weighing factors that control the relative importance of each of the loss terms. In practice, η_r, η_g, η_d, and η_b are more dominant compared to the other terms. The actual values of these weighing factors do not really matter as long as the above terms are properly weighted.

The above problem is minimized using Ceres Solver [2], a nonlinear least squares minimization framework, using a Levenberg-Marquardt optimizer with a Jacobi pre-conditioner. In addition, each term is composed with a Huber loss function, to reduce the effect of outliers on the solution.

5.5 Experiments and Results

We perform a thorough quantitative and qualitative analysis of our approach on challenging sequences from KITTI Tracking [21] and SYNTHIA-SF [24] benchmarks. These sequences are chosen such that they capture a diverse class of road plane profiles viz. uphill, downhill, combinations of them, and even banked road planes. We compare the 3D localization error of the proposed method with the current state-of-the-art monocular competitor [31], and demonstrate significant improvements. Through a series of systematic evaluations, we demonstrate that ground plane estimation is vital for accurate lo-

Table 5.1 Mean Localization Error (Standard Deviation in parenthesis) in meters for the vehicles evaluated using our approach on the KITTI [21] Tracking dataset (Here ($<x\ m$) and ($>x\ m$) denote the set of all cars within a ground-truth depth of x meters and beyond the depth of x meters respectively)

Approach	Overall (m)	$<= 15m$	$<= 30m$	$>30m$
Murthy et. al. [31]	2.61 (±2.23)	1.59 (±0.96)	2.52 (±2.16)	4.30 (±2.83)
Ours (with coplanarity assumption)	1.00 (±0.77)	0.67 (±0.50)	0.94 (±0.69)	2.19 (±1.18)
Ours (joint optimization)	**0.86** (±**0.87**)	**0.55** (±**0.50**)	**0.79** (±**0.79**)	**2.16** (±**1.18**)

Table 5.2 Mean Localization Error (Standard Deviation in parenthesis) in meters for the vehicles with challenging road profiles evaluated using our approach on the KITTI [21] Tracking dataset

Approach	Overall (m)	$<= 15m$	$>15m$
Murthy et. al. [31]	2.55 (±3.16)	2.32 (±2.21)	2.92 (±3.38)
Ours (with coplanarity assumption)	0.95 (±0.89)	0.92 (±0.68)	1.00 (±0.96)
Ours (joint optimization)	**0.67** (±**0.66**)	**0.64** (±**0.60**)	**0.72** (±**0.71**)

calization on roads surfaces with pitch and banks. We also demonstrate that our method is independent of the road plane profile on which vehicles are to be localized and reconstructed. In other words, unlike others (such as [30, 41, 11]) we do not assume that the ego car and the car to be reconstructed are on the same road plane.

Dataset

We use the KITTI Tracking [21] benchmark to evaluate our proposed method. Sequences numbered 1, 3, 7, 8, 9, 10, 11 and 20, which contain a large number of vehicles located on roads with varying plane profiles, were used for evaluating our approach. But, KITTI [21] has only a limited number of steep slopes and banks. So, we also select about 200 vehicles located on challenging plane profiles from sequences numbered 1, 2, 4, 5 and 6 of the SYTHIA-SF [24] dataset. To ensure fair comparison, we evaluate the previous best monocular competitor [31] on the same sequences.

Keypoint Network Training

The proposed network (Sec. 3.2) was trained on the Torch framework [15] with more that 1.2 million images generated synthetically using the modified render pipeline presented in [33]. A train-validation split of $75 - 25$ % was used. The keypoint network was trained for 7 epochs on NVIDIA GTX TITAN X GPUs ($\sim$ 36 hours).

5.5.1 Localization Accuracy

To evaluate localization precision, we compute the mean Absolute Translational Error (ATE) of the vehicles (in meters) of the approaches considered against the available ground truth information. We present these results in Table 5.1, Table 5.2 and Table 5.3. While Table 5.1 captures the overall

Table 5.3 Mean Localization Error (Standard Deviation in parenthesis) in meters for the vehicles (including challenging road profile) evaluated using our approach on the SYNTHIA-SF [24] dataset

Approach	Overall (m)	$<= 15m$	$<= 30m$	$>30m$
Murthy et. al. [31]	76.34 (± 94.03)	54.21 (± 47.93)	66.28 (± 88.74)	86.40 (± 99.32)
Ours (coplanarity)	32.03 (± 45.60)	6.3 (± 19.17)	21.76 (± 65.76)	42.31 (± 25.42)
Ours (joint optim.)	**0.92** ($\pm$**0.93**)	**0.66** ($\pm$**0.49**)	**0.82** ($\pm$**0.76**)	**1.23** ($\pm$**1.11**)

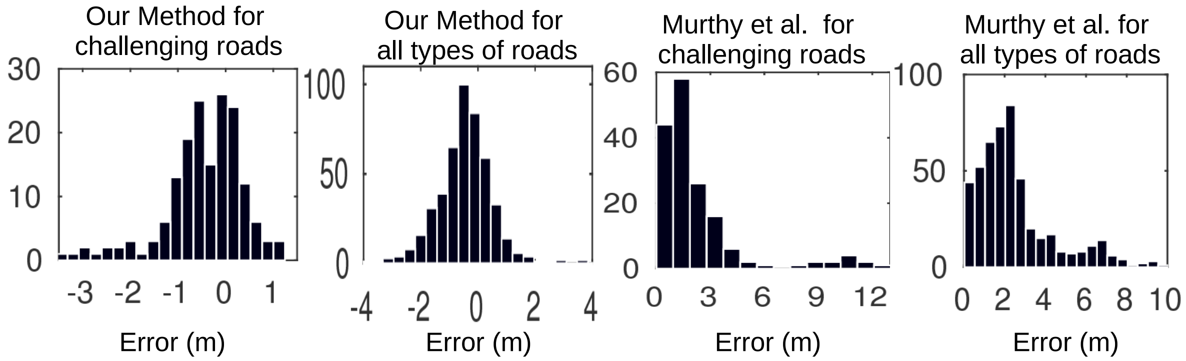

Figure 5.9 Histograms showing distribution of localization errors; challenging roads mean slopes, slanted roads, banked roads, etc.

performance of our approach on KITTI [21] dataset, Table 5.2 presents an analysis of the performance of our approach on KITTI[21] sequences with cars on roads with some pitch or banking angle, or parked on pavements. In Table 5.3, we perform a thorough analysis of our approach on SYNTHIA-SF [24] which has extremely steep roads, and demonstrate the efficacy of the proposed approach in adapting to a wide variety of road plane profiles.

We outperform the current best monocular localization result of [31] on the KITTI benchmark [21] by a significant margin. It is important to note that in [31], the shape priors comprised 14 keypoints per vehicle, whereas we use a different shape prior model comprising 36 keypoints per vehicle. However, to emphasize that this improvement does not stem from more expressive shape prior used in this work, we re-implement the approach in [31] using our learnt shape priors and provide an ablation study to further drive the point home. This highlights the importance of the inclusion of ground plane in localization. As shown in Table 5.1, we achieve a mean localization error of 0.86 meters, as compared to 2.61 meters in [31]. This is a mark improvement stemming from the inclusion of ground plane.

We also address challenging sequences with road slopes on KITTI [21] and provide our localization errors in Table 5.2, and perform an ablation study of our approach to highlight how the inclusion of ground plane reduces the localization error to 0.67 meters, as compared to an error of 2.55 meters given by [31]. The current state-of-the-art [31] relies on the assumption that the plane of the target vehicle and ego vehicle are co-planar. We circumvent this assumption leading to a highly accurate localization

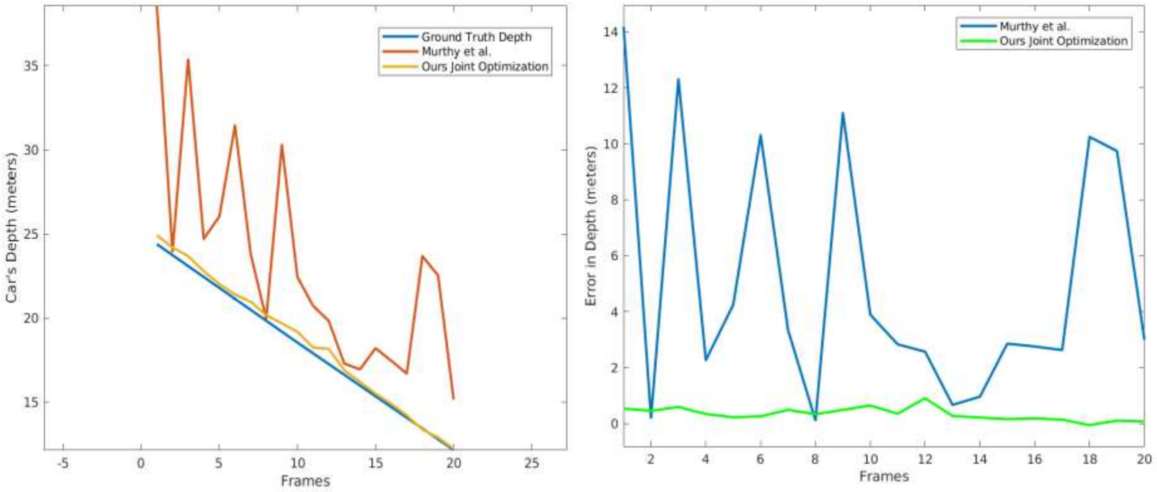

Figure 5.10 *Left*: Estimated depth of a car on a steep slope. We compare our method's localization with [31] against the ground truth. *Right*: Localization error for the same car using the proposed method and the one proposed in [31].

of the target vehicle, in a more diverse set of scenarios. For vehicles that are close to the car, we achieve a high degree of precision (mean error of about 0.67 meters, with a low standard deviation as well).

To further evaluate our approach, we test it on the extremely challenging SYNTHIA-SF [24] dataset which has steep road surfaces with several non-planar profiles. [31] fails completely in the task of accurate shape estimation and localization of objects in such scenarios, due to the coplanarity assumption. Moreover, the method given by [31] fails drastically in non-planar surfaces, giving a mean localization error of 76.34 meters, amplified by the sparse set of keypoints (14 keypoints are used, as opposed to ours, which uses 36) leading to large localization errors. Our system achieves a mean localization error of 0.92 meters, the results of which are shown in Table 5.3. The proposed method generalizes well to different plane profiles and performs significantly well. Once again, we stress the importance of ground plane and exhibit how its inclusion helps us to perform significantly better as compared to the approach of [31], which assumes coplanarity of the vehicles and ego car. Fig. 5.9 shows the error distribution of our approach (first two) and for the approach proposed in [31] (last two); Fig. 5.10 shows the trajectory and localization error for a car in KITTI [21].

5.5.2 Qualitative Results

We showcase the qualitative results of our approach on challenging KITTI [21] and SYNTHIA-SF[24] scenes with moderate to high slopes. For KITTI [21], in figures 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, we overlay the final estimate of the car in 3D along with the ground truth 3D bounding box to show how our approach estimates the vehicle shape and pose accurately. In each figure, on the top we show the image of the cars overlaid with the projection of the estimated reconstruction (wireframe) along with the estimated depth on top it. And in the bottom, we show the localization in 3D from different perspectives. It can be seen in Fig. 5.11 that our method is able to very accurately recover the pose and shape of the vehicle on inclined surfaces.

For SYNTHIA-SF[24], see figures 5.17, 5.18, 5.19, 5.20, 5.21, we overlay the estimate of the car after shape and pose adjustment on the ground truth scene points to highlight the accurate shape and pose estimation of the car. The estimated depth of each car from reconstruction has been shown on the image above the corresponding car. It is clear from the figures that our approach is capable of reconstructing vehicles on very steep roads. Note that, when we say that the vehicles are on steep and inclined roads, we mean that the ego car is not on the same surface.

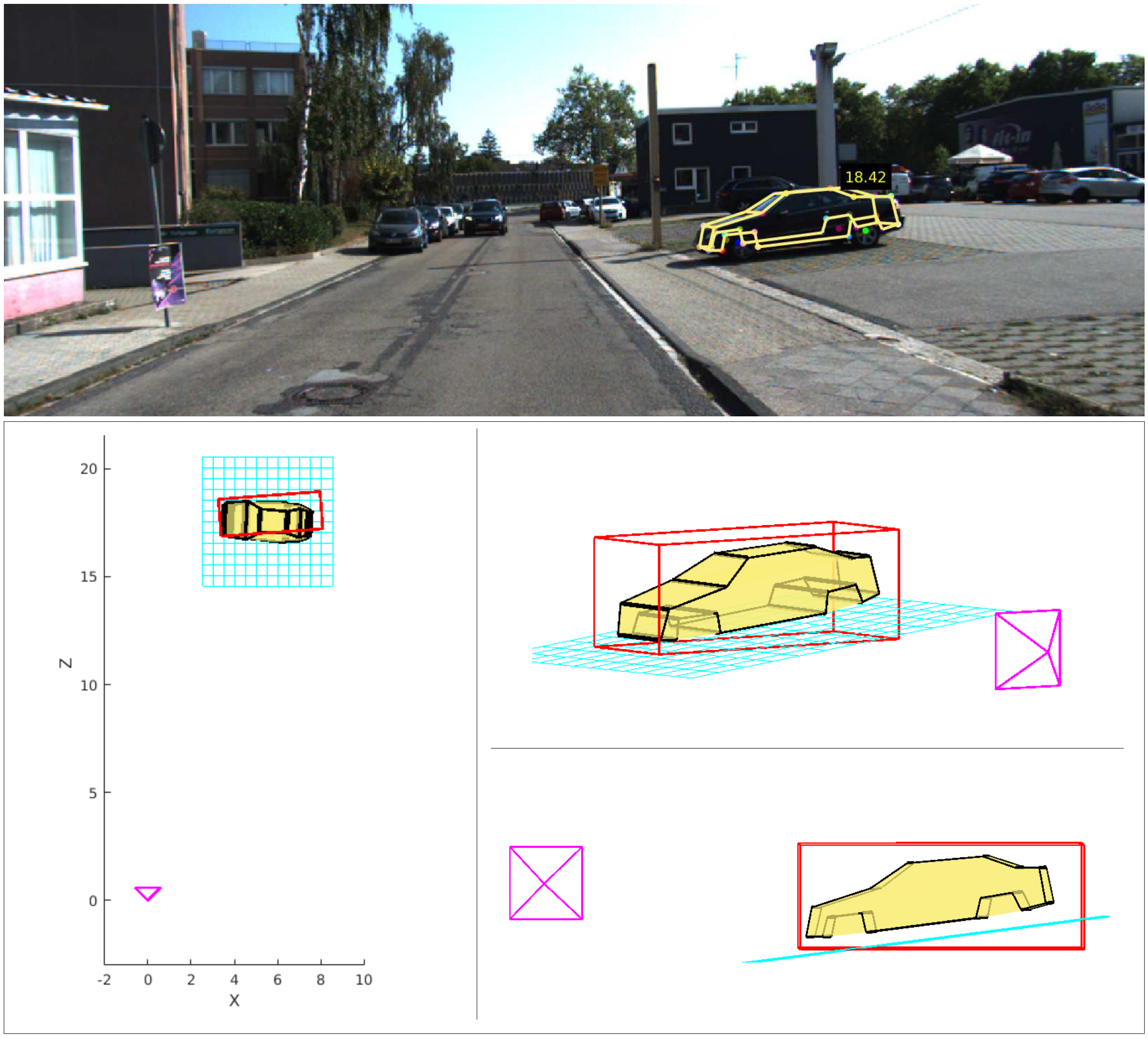

Figure 5.11 Qualitative results on KITTI[21]. *Top:* Estimated 3D wireframe (reconstructed shape) for a car projected on the image, with depth displayed on top of the car. *Bottom:* Bird's eye view of the car overlaid with its respective ground truth bounding boxes (in red). The cyan mesh represents the ground plane on which the car stands

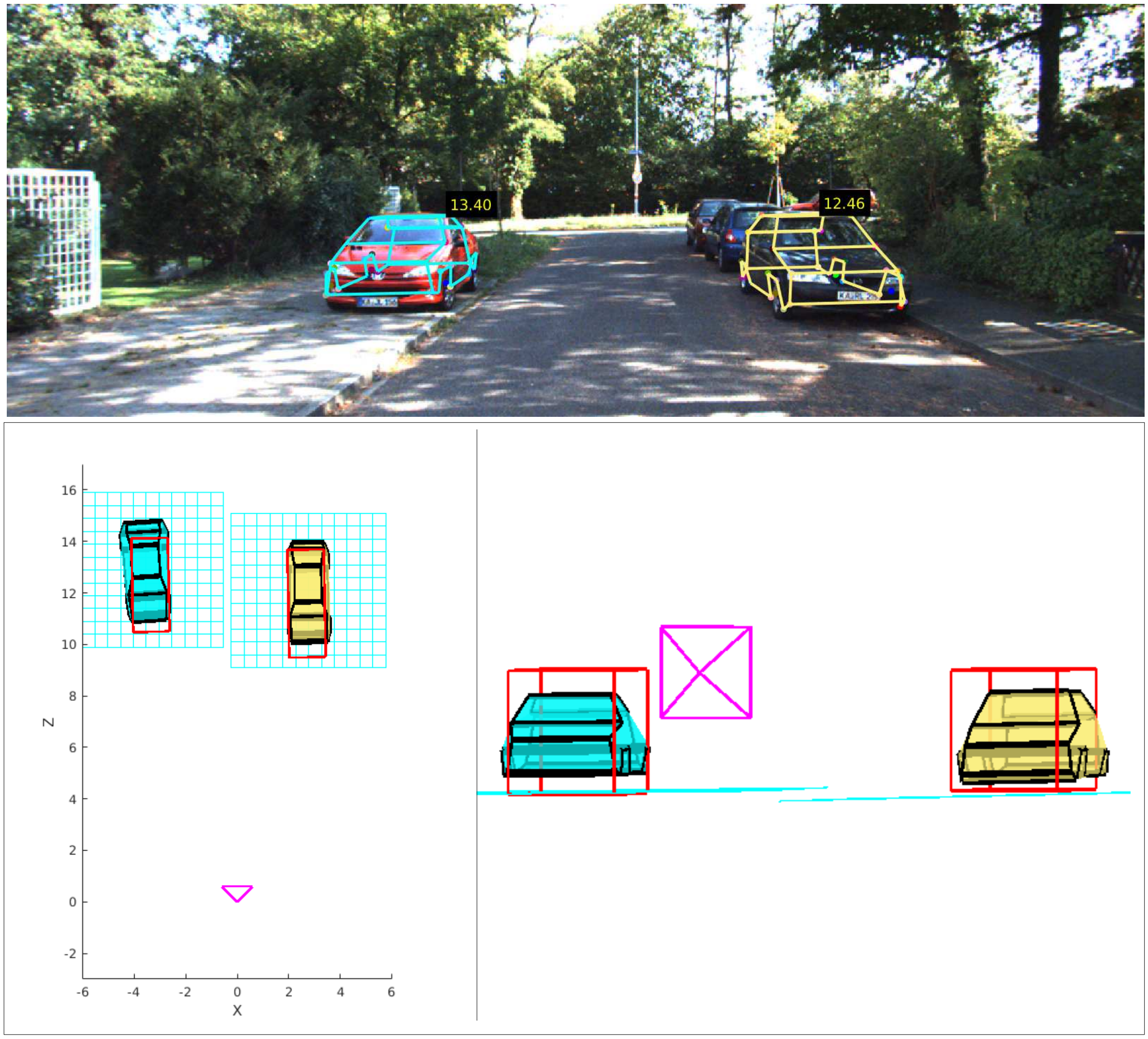

Figure 5.12 Qualitative results on KITTI[21]. *Top:* Estimated 3D wireframes (reconstructed shapes) for selected cars projected on the image, with depth displayed on top of each car. *Bottom:* Bird's eye view of the cars overlaid with their respective ground truth bounding boxes (in red). The cyan mesh represents the ground plane on which the car stands

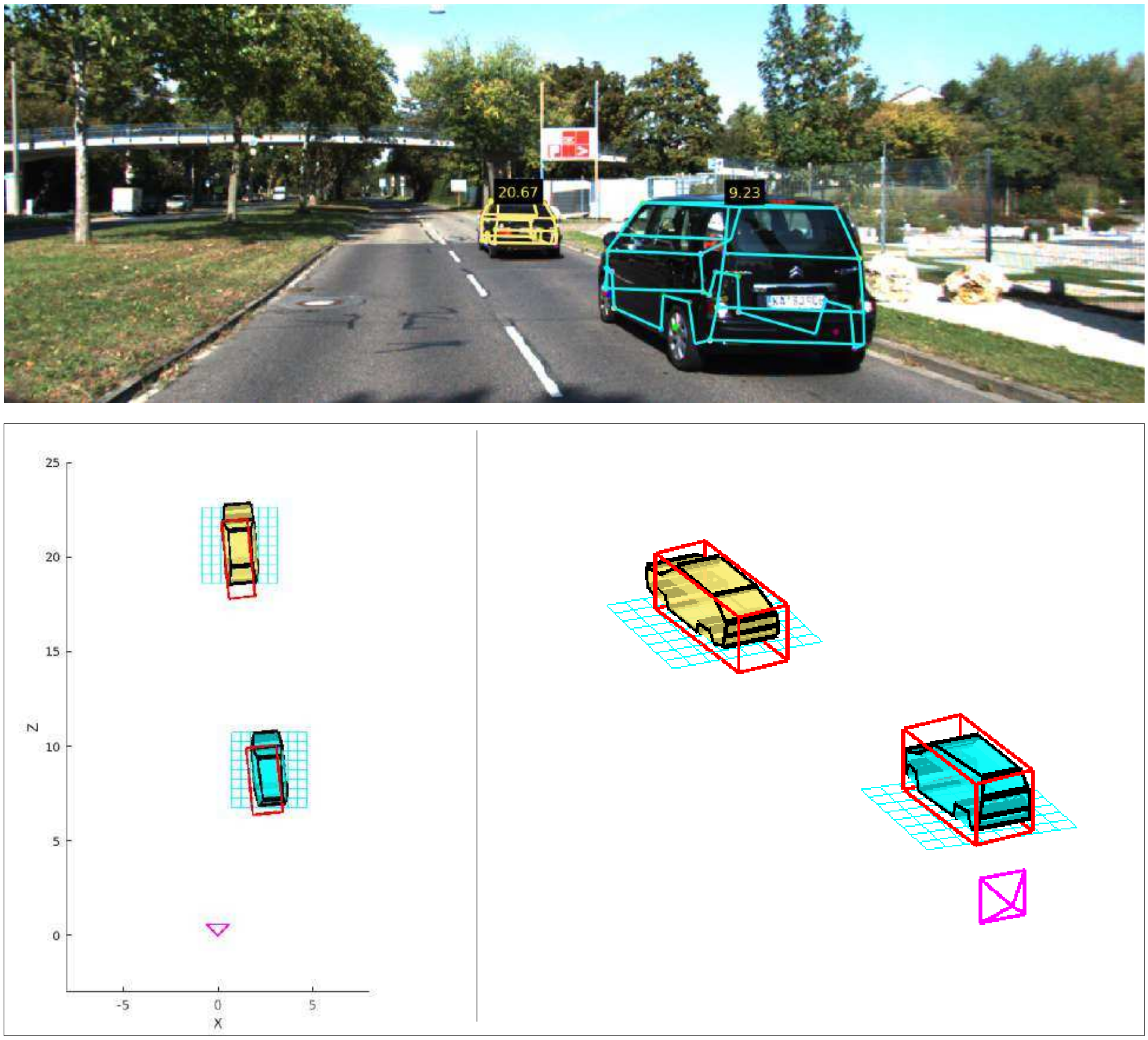

Figure 5.13 Qualitative results on KITTI[21]. *Top:* Estimated 3D wireframes (reconstructed shapes) for selected cars projected on the image, with depth displayed on top of each car. *Bottom:* Bird's eye view of the cars overlaid with their respective ground truth bounding boxes (in red). The cyan mesh represents the ground plane on which the car stands

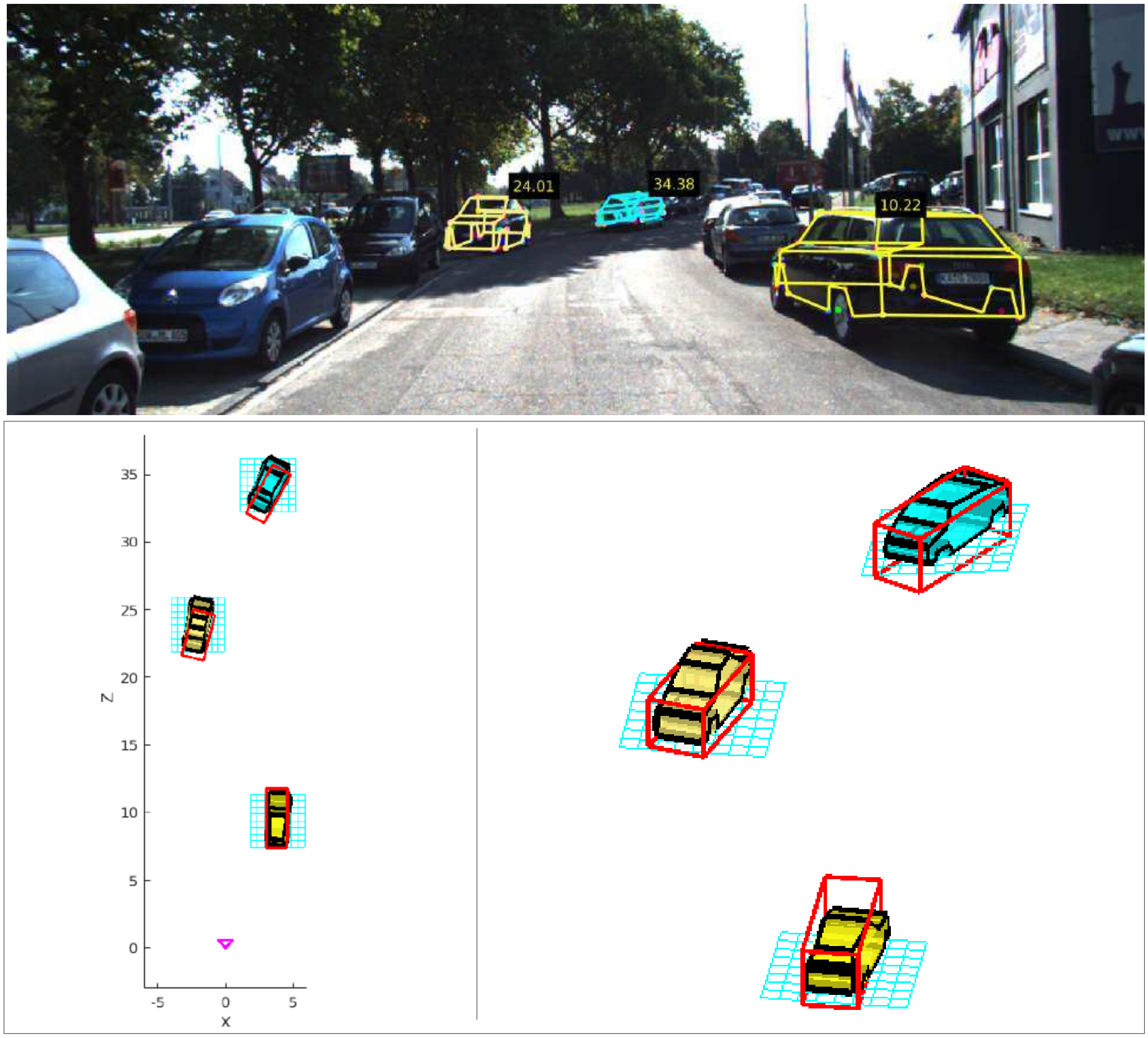

Figure 5.14 Qualitative results on KITTI[21]. *Top:* Estimated 3D wireframes (reconstructed shapes) for selected cars projected on the image, with depth displayed on top of each car. *Bottom:* Bird's eye view of the cars overlaid with their respective ground truth bounding boxes (in red). The cyan mesh represents the ground plane on which the car stands

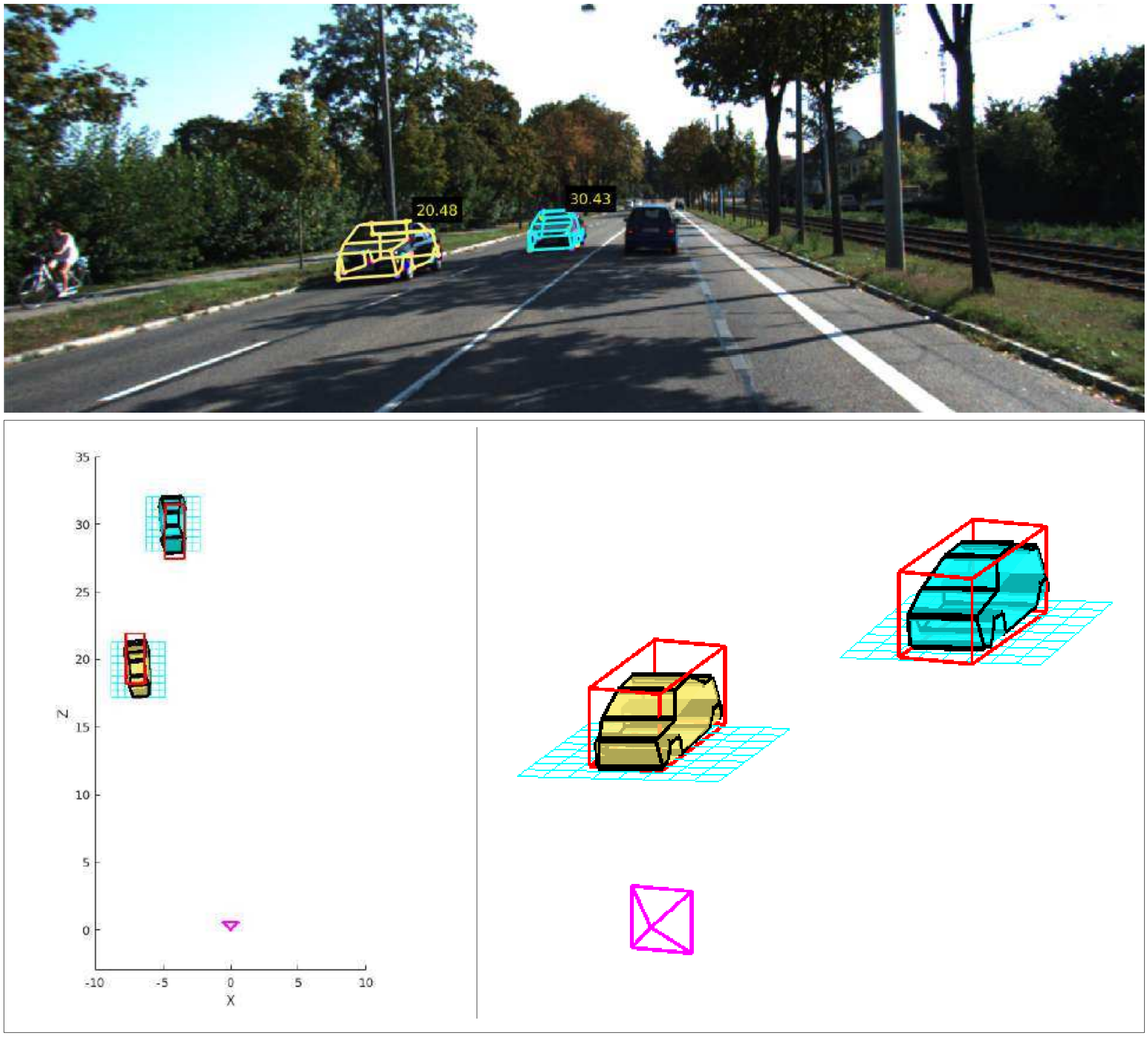

Figure 5.15 Qualitative results on KITTI[21]. *Top:* Estimated 3D wireframes (reconstructed shapes) for selected cars projected on the image, with depth displayed on top of each car. *Bottom:* Bird's eye view of the cars overlaid with their respective ground truth bounding boxes (in red). The cyan mesh represents the ground plane on which the car stands

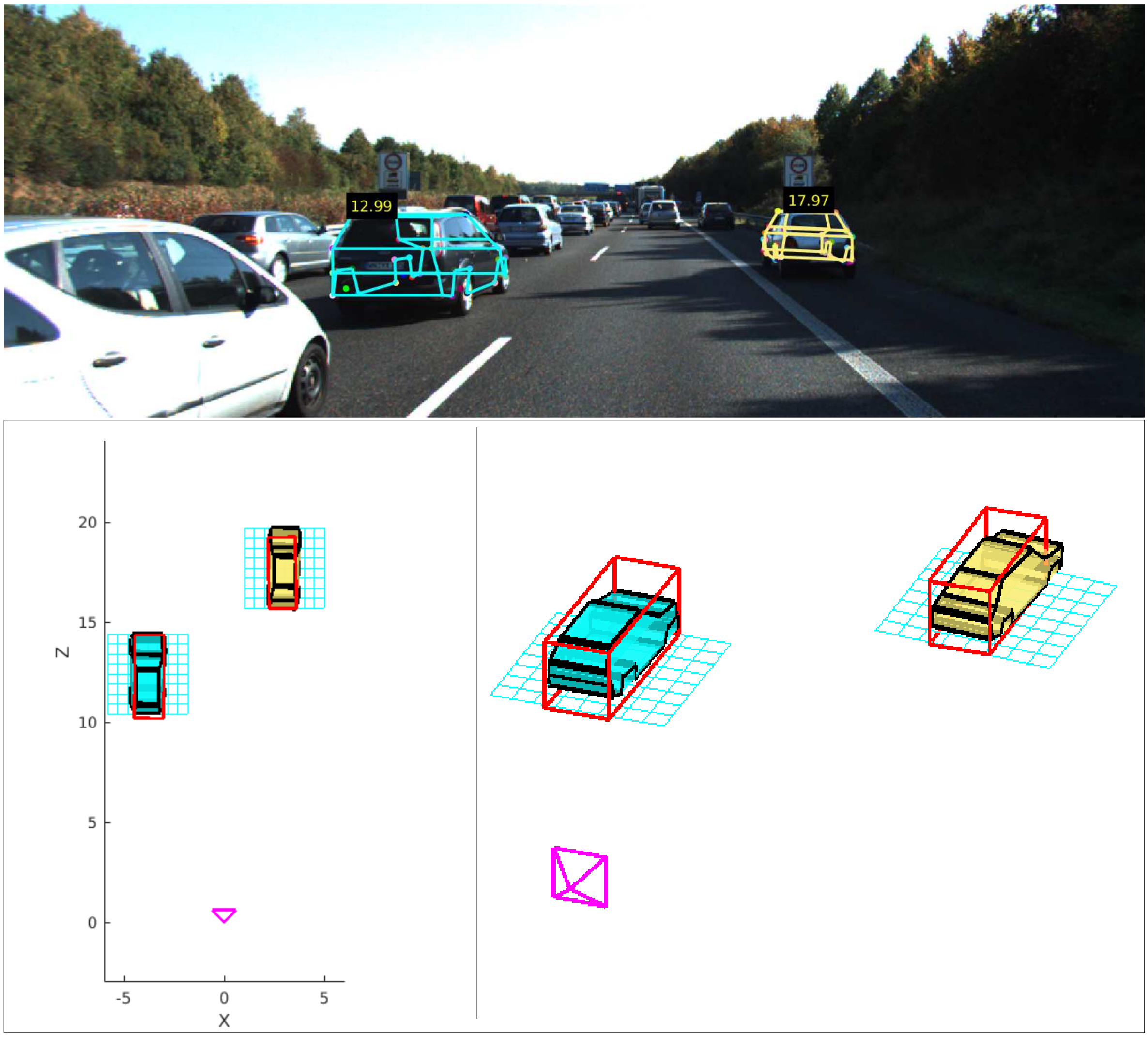

Figure 5.16 Qualitative results on KITTI[21]. *Top:* Estimated 3D wireframes (reconstructed shapes) for selected cars projected on the image, with depth displayed on top of each car. *Bottom:* Bird's eye view of the cars overlaid with their respective ground truth bounding boxes (in red). The cyan mesh represents the ground plane on which the car stands

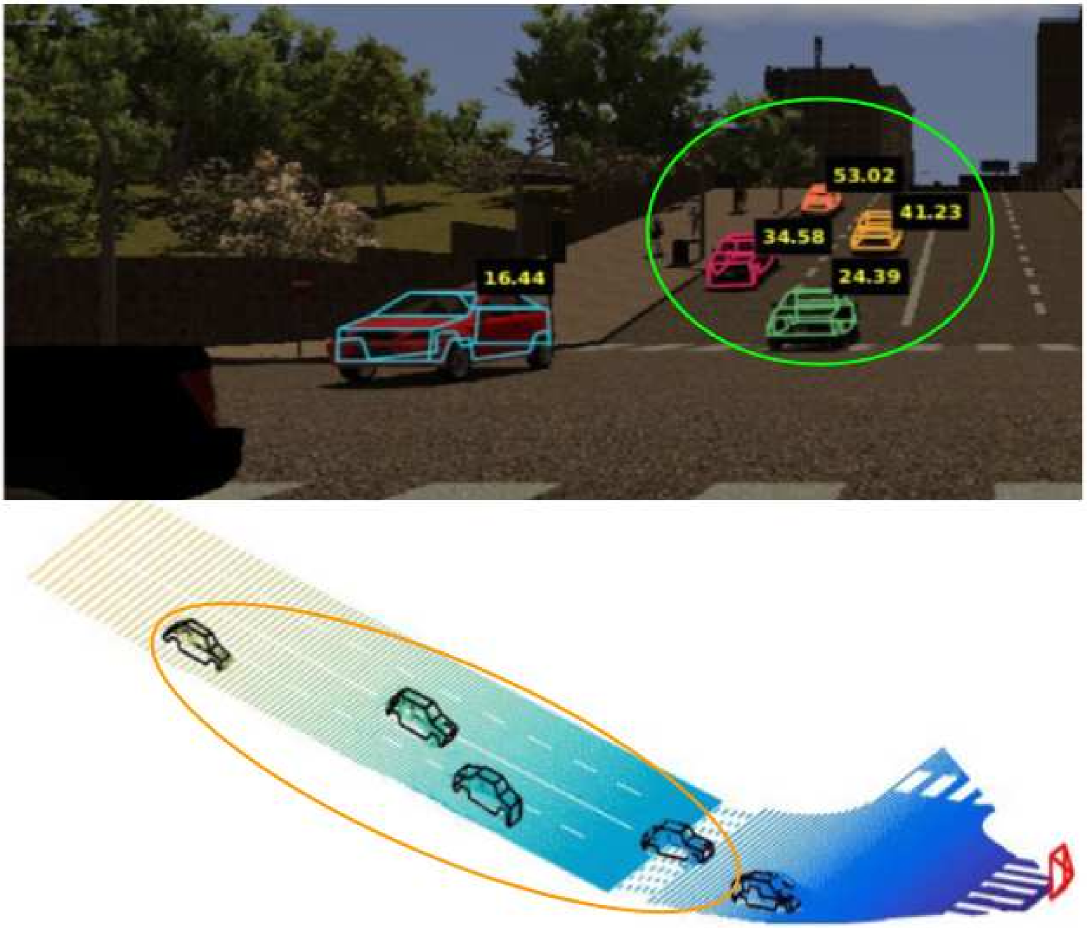

Figure 5.17 Qualitative results on SYNTHIA-SF[24]. *Top:* Estimated 3D wireframes for selected cars (on different road profiles) projected on the image, with depth displayed on top of each car. *Bottom:* visualization of the estimated wireframes in 3D, overlaid on dense ground truth 3D scene points.

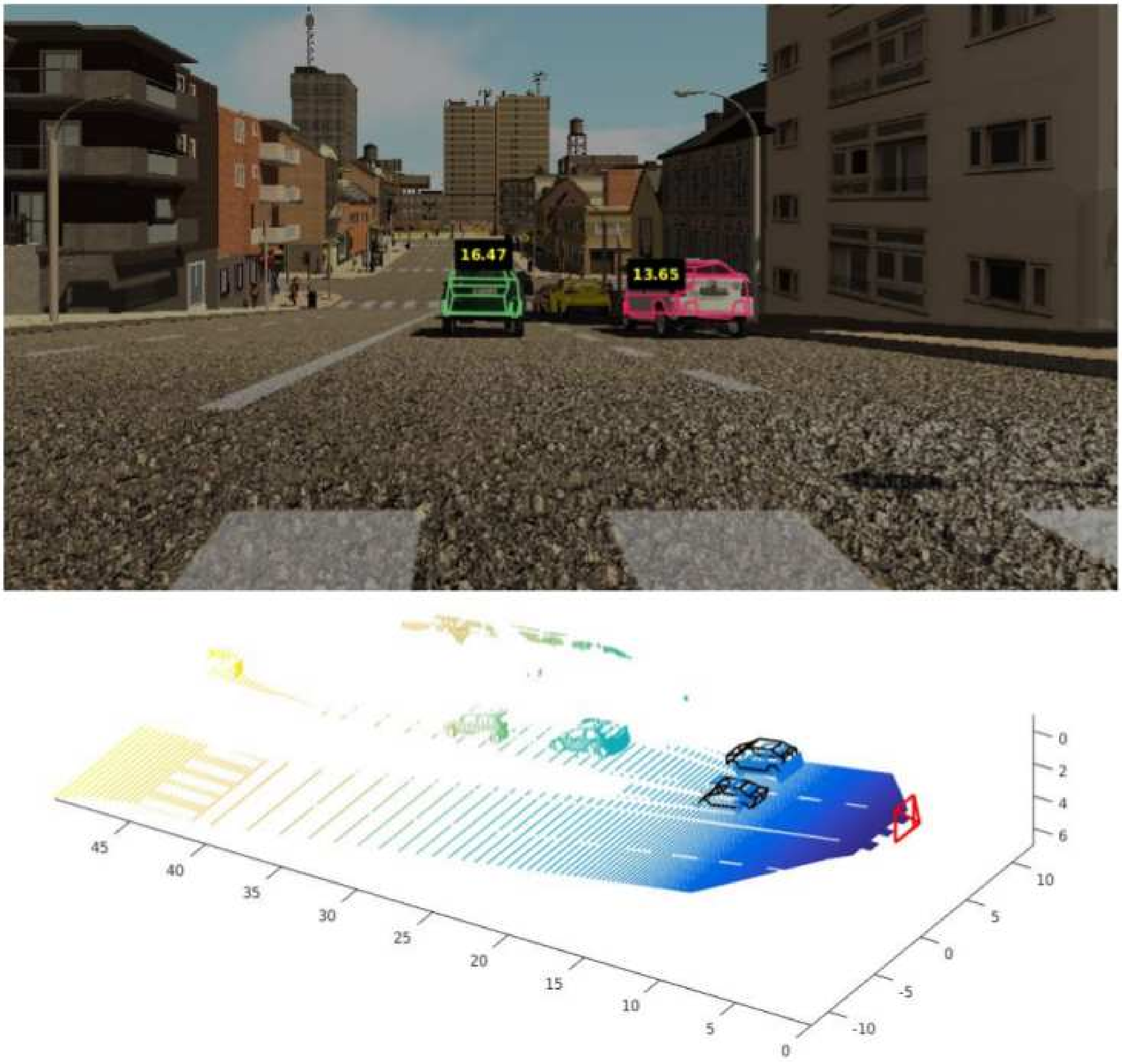

Figure 5.18 Qualitative results on SYNTHIA-SF[24]. *Top:* Estimated 3D wireframes for selected cars (on different road profiles) projected on the image, with depth displayed on top of each car. *Bottom:* visualization of the estimated wireframes in 3D, overlaid on dense ground truth 3D scene points.

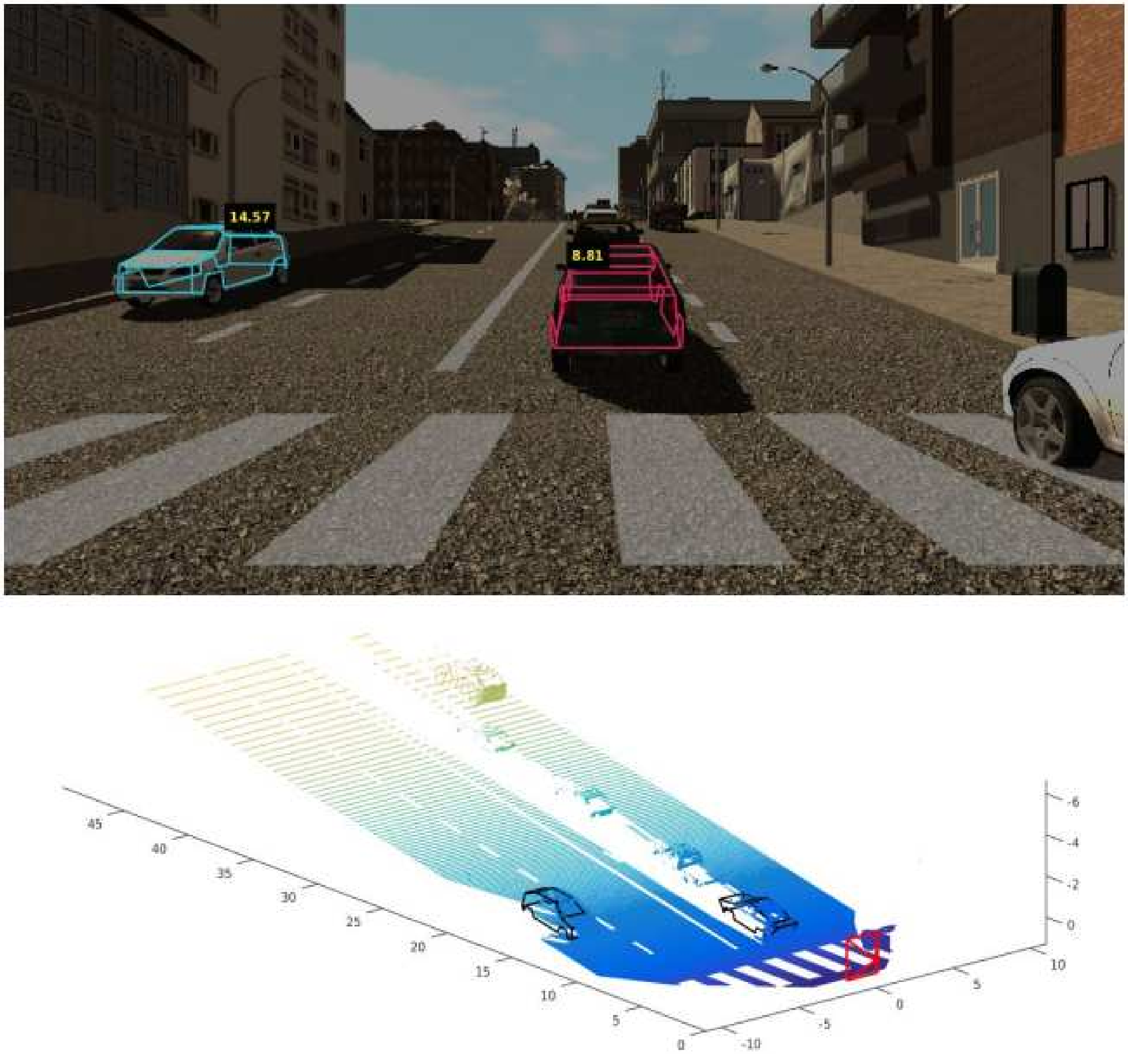

Figure 5.19 Qualitative results on SYNTHIA-SF[24]. *Top:* Estimated 3D wireframes for selected cars (on different road profiles) projected on the image, with depth displayed on top of each car. *Bottom:* visualization of the estimated wireframes in 3D, overlaid on dense ground truth 3D scene points.

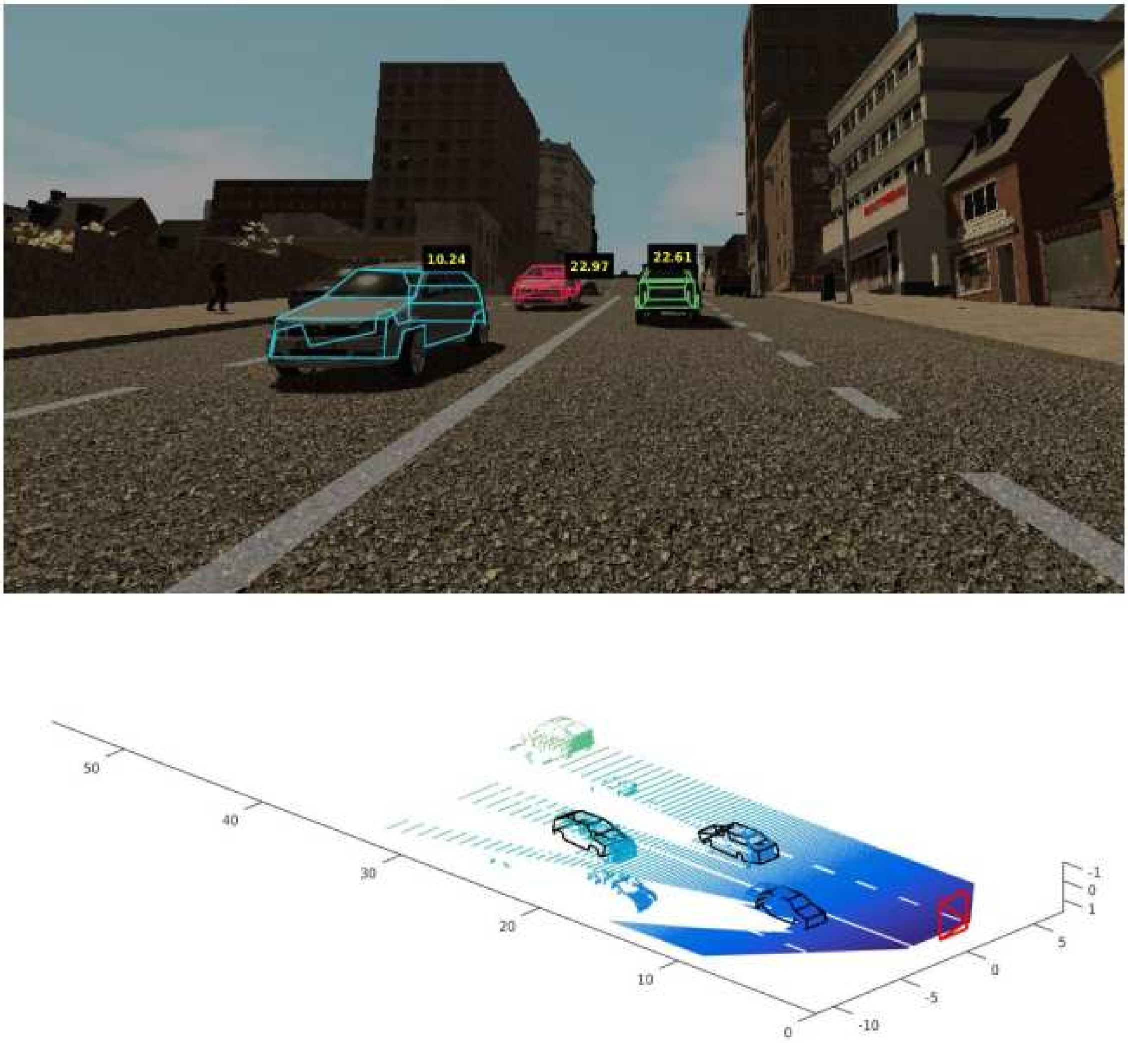

Figure 5.20 Qualitative results on SYNTHIA-SF[24]. *Top:* Estimated 3D wireframes for selected cars (on different road profiles) projected on the image, with depth displayed on top of each car. *Bottom:* visualization of the estimated wireframes in 3D, overlaid on dense ground truth 3D scene points.

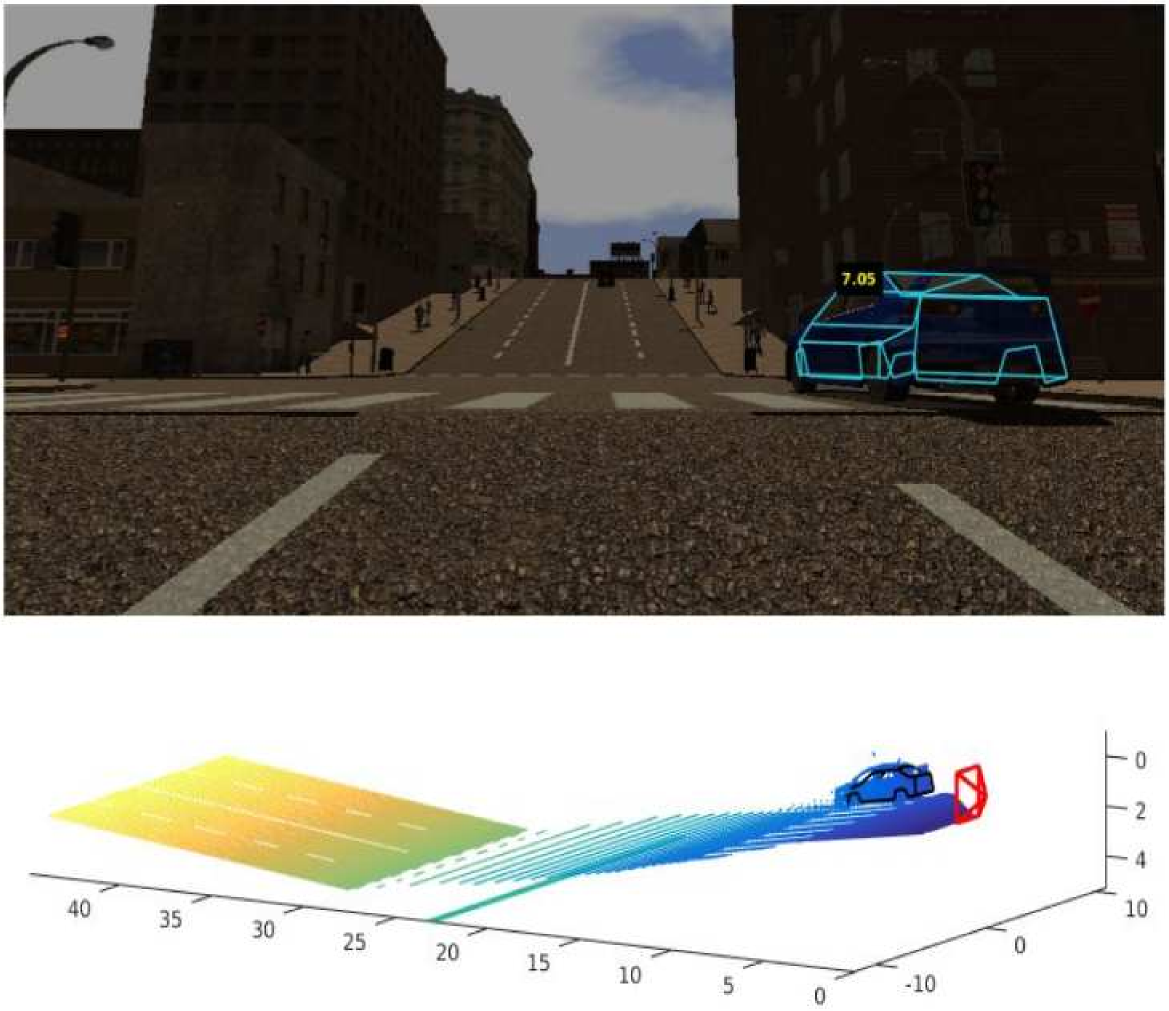

Figure 5.21 Qualitative results on SYNTHIA-SF[24]. *Top:* Estimated 3D wireframe for a car (on a challenging road profile) projected on the image, with depth displayed on top of the car. *Bottom:* visualization of the estimated wireframe in 3D, overlaid on dense ground truth 3D scene points.

5.5.3 Summary of Results

The cornerstone of this effort is to highlight that the presence of non-planar road profiles leads to an unsuccessful pose estimation of cars in urban scenarios by the current state-of-the-art approach, due to the fact that it relies on the coplanarity of the ego vehicle and the car. Our proposed approach is independent of the plane profile on which the car is located. We improve localization accuracy by a large margin through the joint estimation of ground plane in KITTI [21] sequences regardless of whether or not they contain slopes. (cf. Table 5.1 and Table 5.2). The importance of the proposed approach is highlighted in Table 5.2, where we achieve a performance boost of about 4 times in scenes with moderate slopes. For an overall comparison on KITTI [21], we evaluate our approach on scenes with different planar and non-planar road surfaces and show an improvement of about 3 times. We further present the performance of our approach on SYNTHIA-SF [24] which has extremely steep roads, resulting in a catastrophic failure of the current state-of-the-art monocular shape and pose estimation [31]. Our performance is significantly improved in such scenes, irrespective of the road profiles, the results of which are reported in Table 5.3. We also perform an ablation study, reported in Table 5.1, Table 5.2 and Table 5.3, to highlight the importance of our ground plane estimation policy, and show that it provides a significant performance boost over just the utilization of a well-constrained 36 keypoint shape prior.

Chapter 6

Conclusions

We presented an approach for accurate 3D localization and shape estimation of dynamic vehicles on arbitrary road profiles from a moving monocular camera. Where most current monocular localization systems rely on the standard coplanarity assumption which requires the ego car and the car to be localized to share the same road plane surface, we relax this assumption and propose a joint optimization formulation which jointly estimates the car's pose-shape and its local road plane parameters. In this work, we have demonstrated that inclusion of plane brings in many intuitive yet important cues such as the car to be localized should be close to its local ground plane, the 3D orientation of the car should closely match the orientation of its local road plane, etc. These cues help to narrow down the solution space of the highly non-linear function to be optimized, thereby resulting into a more stable optimization process and more reliable localization.

We validate our method by testing it on two publicly available datasets - KITTI [21] and SYNTHIA-SF [24], and demonstrate that our method is able to recover the pose and shape (i.e. localization and reconstruction) of car's on varying road profiles, ranging from coplanar to highly steep road surfaces. We demonstrate a significant improvement over the state-of-the-art monocular localization methods. On KITTI [21], our approach shows a mark improvement over our best competitor [31] with an accuracy of about 3-4 times that of [31]. On SYNTHIA-SF [24], where [31] drastically fails, our method achieves localization with a mean error of less than 1 meter. These results showcase the importance of incorporating the local ground plane of the vehicles for the task of localization and reconstruction.

As future work, a possible extension would be to include observations of cars from multiple frames. These observations can be exploited to bring in more constraints such as temporal consistency in the shape of the vehicle, non-holonomic motion constraint to enforce smooth trajectories of the localized cars, etc. Another important extension would be to address the scenario where not much of the road surface is visible, for e.g. in heavy traffic conditions.

Related Publications

- **Junaid Ahmed Ansari***, Sarthak Sharma*, Anshuman Majumdar, J Krishna Murthy, K Madhava Krishna. *The Earth Aint Flat: Monocular Reconstruction of Vehicles on Steep and Graded Roads from a Moving Camera*. **In IEEE International Conference on Intelligent Robots and Systems (IROS)** 2018. *Published*.

Other Publications

- Sarthak Sharma*, **Junaid Ahmed Ansari***, J Krishna Murthy, K Madhava Krishna. *Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking*. **In IEEE International Conference on Robotics and Automation (ICRA)** 2018. *Published.*
- Shashank Srikanth, **Junaid Ahmed Ansari**, Karnik Ram, Sarthak Sharma, J Krishna Murthy, K Madhava Krishna. *INFER: INtermediate representations for distant FuturE pRediction.* **In IEEE Conference on Intelligent Robots and Systems (IROS)** 2019. *Accepted*.

*(*Equal contribution)*

Bibliography

[1] Kiwi climb: Hoofing up the world's steepest street. http://edition.cnn.com/travel/article/worlds-steepest-street-residents/index.html.

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. `ceres-solver.org`.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *PAMI*, 39(12):2481–2495, 2017.

[4] I. Barabanau, A. Artemov, E. Burnaev, and V. Murashkin. Monocular 3d object detection via geometric reasoning on keypoints. *arXiv preprint arXiv:1905.05618*, 2019.

[5] V. Blanz, T. Vetter, et al. A morphable model for the synthesis of 3d faces. In *Siggraph*, volume 99, pages 187–194, 1999.

[6] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *cvpr*, volume 2, page 2690. Citeseer, 2000.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] Y. Chang, C. Hu, R. Feris, and M. Turk. Manifold based analysis of facial expression. *Image and Vision Computing*, 24(6):605–614, 2006.

[10] F. Chausse, R. Aufrere, and R. Chapuis. Recovering the 3d shape of a road by on-board monocular vision. In *ICPR*, 2000.

[11] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, pages 2147–2156, 2016.

[12] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[13] F. Chhaya, D. Reddy, S. Upadhyay, V. Chari, M. Z. Zia, and K. M. Krishna. Monocular reconstruction of vehicles: Combining slam with shape priors. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5758–5765. IEEE, 2016.

[14] J. Choe, K. Joo, F. Rameau, G. Shim, and I. S. Kweon. Segment2regress: Monocular 3d vehicle localization in two stages.

[15] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[16] B. O. Community. *Blender – a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2017.

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

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] W. Ding, S. Li, G. Zhang, X. Lei, and H. Qian. Vehicle pose and shape estimation through multiple monocular vision. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 709–715. IEEE, 2018.

[20] J. Fritsch, T. Kühnl, and F. Kummert. Monocular road terrain detection by combining visual and spatial information. *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[21] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[22] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016.

[23] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[24] D. Hernandez-Juarez, L. Schneider, A. Espinosa, D. Vazquez, A. M. Lopez, U. Franke, M. Pollefeys, and J. C. Moure. Slanted stixels: Representing san francisco's steepest streets. In *BMVC*, 2017.

[25] C. Jang and K. Jung. Human pose estimation using active shape models. *Proceedings of World Academy of Science: Engineering & Technology*, 46, 2008.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] C. Li, M. Zeeshan Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5465–5474, 2017.

[28] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[29] P. Martins, R. Caseiro, J. F. Henriques, and J. Batista. Let the shape speak-discriminative face alignment using conjugate priors. In *BMVC*, volume 1, page 2. Citeseer, 2012.

[30] J. K. Murthy, G. S. Krishna, F. Chhaya, and K. M. Krishna. Reconstructing vehicles from a single image: Shape priors for road scene understanding. In *ICRA*, 2017.

[31] J. K. Murthy, S. Sharma, and K. M. Krishna. Shape priors for real-time monocular object localization in dynamic environments. In *IROS*, 2017.

[32] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*. Springer, 2016.

[33] J. K. M. K. M. K. Parv Parkhiya, Rishabh Khawad and B. Bhowmick. Constructing category-specific models for monocular object slam. In *ICRA*, 2018.

[34] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2011–2018. IEEE, 2017.

[35] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[36] Z. Qin, J. Wang, and Y. Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. *arXiv preprint arXiv:1811.10247*, 2018.

[37] J. X. J. W. J. QiongYan and Y.-W. LiXu. Accurate single stage detector using recurrent rolling convolution. In *CVPR*, 2017.

[38] R. K. Satzoda and M. M. Trivedi. Vision-based lane analysis: Exploration of issues and approaches for embedded realization. In *CVPR Workshops*, pages 604–609. IEEE, 2013.

[39] T. Shan, B. C. Lovell, and S. Chen. Face recognition robust to head pose from one sample image. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 515–518. IEEE, 2006.

[40] S. Song and M. Chandraker. Robust scale estimation in real-time monocular sfm for autonomous driving. In *CVPR*, pages 1566–1573, 2014.

[41] S. Song and M. Chandraker. Joint sfm and detection cues for monocular 3d localization in road scenes. In *CVPR*, 2015.

[42] G. P. e. a. Stein. Vision-based acc with a single camera: Bounds on range and range rate accuracy. In *Intelligent Vehicles Symposium*. IEEE, 2003.

[43] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, December 2015.

[44] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[45] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892, 2008.

[46] S. Tulsiani, A. Kar, J. Carreira, and J. Malik. Learning category-specific deformable 3d models for object reconstruction. *PAMI*, 2016.

[47] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *CVPR*. IEEE, 2015.

[48] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV*, Sydney, Australia, Dec. 2013.

[49] X. Weng and K. Kitani. Monocular 3d object detection with pseudo-lidar point cloud. *arXiv preprint arXiv:1903.09847*, 2019.

[50] Y. Wu, Z. Wang, and Q. Ji. Facial feature tracking under varying facial expressions and face poses based on restricted boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3452–3459, 2013.

[51] B. Xu and Z. Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2353, 2018.

[52] S. Yang and S. Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 2019.

[53] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *CVPR*, pages 4966–4975, 2016.

[54] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object recognition and modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2608–2623, 2013.

[55] M. Z. Zia, M. Stark, and K. Schindler. Towards scene understanding with detailed 3d object representations. *IJCV*, 2015.