RECOVERING THE 3D POSE AND SHAPE OF VEHICLES FROM STEREO IMAGES

Max Coenen*, Franz Rottensteiner, Christian Heipke

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
(coenen, rottensteiner, heipke)@ipi.uni-hannover.de

Commission II, WG II/5

KEY WORDS: Object Detection, Stereo Images, Pose Estimation, 3D Reconstruction, 3D Modelling, Active Shape Model

ABSTRACT:

The precise reconstruction and pose estimation of vehicles plays an important role, e.g. for autonomous driving. We tackle this problem on the basis of street level stereo images obtained from a moving vehicle. Starting from initial vehicle detections, we use a deformable vehicle shape prior learned from CAD vehicle data to fully reconstruct the vehicles in 3D and to recover their 3D pose and shape. To fit a deformable vehicle model to each detection by inferring the optimal parameters for pose and shape, we define an energy function leveraging reconstructed 3D data, image information, the vehicle model and derived scene knowledge. To minimise the energy function, we apply a robust model fitting procedure based on iterative Monte Carlo model particle sampling. We evaluate our approach using the object detection and orientation estimation benchmark of the KITTI dataset (Geiger et al., 2012). Our approach can deal with very coarse pose initialisations and we achieve encouraging results with up to 82 % correct pose estimations. Moreover, we are able to deliver very precise orientation estimation results with an average absolute error smaller than 4°.

1. INTRODUCTION

Autonomous driving comes with the need to deal with highly dynamic environments. To ensure safe navigation and to enable the interaction with other objects, 3D scene reconstruction and the identification and reconstruction of moving objects, especially vehicles, are fundamental tasks. Additionally, research for collaborative vehicle positioning requires knowledge about the relative poses between cars for them to be used as vehicle-to-vehicle observations (Knuth and Barooah, 2009). This leads to the need of techniques for precise 3D object reconstruction to derive the poses of other vehicles relative to the position of the observing vehicle. In this context, stereo cameras provide a cost-effective solution for sensing a vehicle's surroundings. Most of the existing techniques for vehicle detection and pose estimation are restricted to a coarse estimation of the viewpoint in 2D, whereas the precise determination of vehicle pose, especially of the orientation[1], and vehicle shape is still an open problem. Consequently, the goal of this paper is to propose a method for precise 3D reconstruction of vehicles from street level stereo images. We make use of 3D vehicle reconstructions to reason about the relative vehicle poses in 3D, i.e. the position and rotation of the vehicles with respect to the observing vehicle. To reconstruct the vehicles in 3D, we apply a model-based approach making use of a deformable 3D vehicle model learned from CAD vehicle models. We formulate an energy minimisation problem leveraging both, 3D and 2D image information, and apply an iterative particle based approach to fit one model to each detected vehicle, thus determining the vehicle's precise pose and shape.

2. RELATED WORK

This section provides a brief overview of related work for vehicle pose estimation, vehicle reconstruction and vehicle modelling. A coarse estimation of the vehicle orientation is delivered already by a number of vehicle detection approaches, though mostly in 2D. As the visual appearance of vehicles in image data has a large variety, e.g. due to changing camera viewpoints, often viewpoint specific detectors (Payet and Todorovic 2011, Ozuysal et al. 2009, Villamizar et al. 2011) are being applied. The resulting detections of these approaches are already associated with a coarse estimation of the orientation of the vehicles. However, viewpoint-specific detectors usually have to be trained using a large number of training examples under different viewpoints. Typically, the viewing directions are divided into a discrete number of pose-bins and a classifier is trained for each bin so that a compromise between the detector complexity and the level of detail of the pose estimation is found. This usually leads to a coarse orientation estimation only. Another strategy frequently used for vehicle detection is given by part based approaches (e.g. Felzenszwalb et al. 2010 and Leibe et al. 2006), which divide the objects into several distinctive parts and learn a detector for each part, thus achieving robustness against occlusions. Usually a global model considering the topology of the individual parts is applied for the detection of the entire object. All the methods mentioned so far are solely 2D appearance based and typically only deliver 2D bounding boxes and coarse viewpoint estimations as output. We aim to obtain vehicle detections as well as precise pose estimations, including the vehicle positions and orientations, in 3D space.

A step towards capturing 3D object information from images is done by approaches which internally enrich a part-based detector by linking 3D object knowledge to the parts and transferring this information to the objects after detection. To that end, often the increasing amount of freely available CAD data is exploited. For instance, Liebelt and Schmid (2010) treat appearance and geometry as separate learning tasks. They train an appearance part model from real images and link each part of the training data with 3D geometry from synthetic models, which allows an approximate estimation of 3D pose. Similarly, Pepik et al. (2012) adapt the deformable part model (DPM) of Felzenszwalb et al.

*Corresponding author

[1] In this work, the term *orientation* of a vehicle refers to the rotation angle about the vertical vehicle axis.

https://doi.org/10.5194/isprs-annals-IV-2-73-2018

(2010). They add 3D information from CAD models to the deformable parts and incorporate 3D constraints to enforce part correspondences. Thomas et al. (2007) enrich the Implicit Shape Model (ISM) of Leibe et al. (2006) by adding depth information from training images to the ISM and transfer the 3D information to the test images, which allows the estimation of coarse 3D pose information. Still, the mentioned approaches only use the 3D information implicitly by transferring the learned 3D information to the detected objects.

Alternatively, 3D model information can be used explicitly by deriving cues from the model representation and using these cues actively for vehicle detection, reconstruction and/or to infer pose information. A commonly applied procedure is to use an arbitrary object detector to initialise or instantiate the model, followed by fine-grain model fitting or optimisation. For example, Bao et al. (2013), Dame et al. (2013) and Güney and Geiger (2015) follow this procedure for 3D scene reconstruction by initially detecting vehicles and subsequently integrating vehicle models into their 3D reconstruction algorithm. Bao et al. (2013) calculate a mean model from laser scans of different vehicle instances and adapt it to newly observed instances. Güney and Geiger (2015) integrate disparity patches sampled from a huge set of CAD vehicle models into a disparity map estimated from stereo images. However, that sampling technique is computationally expensive and object instances occurring in the images but not being present in the CAD data set can not be recovered correctly. Dame et al. (2013) use a Signed Distance Function (SDF) for model representation and optimise initial pose and shape parameters from an object detector in a monocular SLAM system.

A SDF is also used by Engelmann et al. (2016) for pose and shape estimation of vehicles detected in stereo images. They fit the SDF to detected vehicles by minimising the distance of reconstructed 3D vehicle points to the SDF. However, a SDF is a rather complex object representation and its level of detail depends on the applied voxel grid size. Active Shape Models (ASM) (Cootes et al., 2000) provide a less complex method to represent the geometry of an object class while being able to cover object deformations due to the intra-class variability. 3D ASM have already been used in the context of vehicle detection and pose estimation. For instance, based on 3D points from mobile laserscanning data, Xiao et al. (2016) use a 3D vehicle ASM to fit it to detected and segmented generic street scene objects. Coenen et al. (2017) fit ASM representing vehicles to 3D points from stereo images associated to vehicle detections. However, the latter three approaches do not use image information at all or only for the initial vehicle detection, but disregard image cues for model fitting. In contrast, Zia et al. (2013) and Zia et al. (2015) only use single images and incorporate a 3D ASM into their detection approach, using the model also to derive precise object pose estimates. For this purpose, they apply a model-keypoint based multi-class classifier. However, the results of Zia et al. (2013) show that their approach heavily depends on a good pose initialisation. Similarly, Lin et al. (2014) recover the 3D vehicle geometry by fitting the 3D ASM to estimated 2D landmark locations resulting from a DPM detector. Their approach also suffers from wrongly estimated part locations resulting from the DPM. A 3D ASM is also used by Menze et al. (2015) to be fitted to detections of vehicles obtained from stereo image pairs and object scene flow. However, using scene flow for object detection is computationally expensive.

In this work we want to reconstruct vehicles from street level stereo images and fully recover their 3D pose and shape. For this purpose we make use of a shape prior by learning an active shape model from CAD vehicle models. Based on initial 3D vehicle detections, we make the following contributions in this paper: (1) We incorporate different types of features and observations derived from the vehicle model, reconstructed 3D data, scene knowledge, and image information into one common energy function to infer the optimal target parameters; (2) we can work without good pose initialisations by defining a robust model initialisation and model fitting procedure based on an iterative Monte Carlo model particle sampling technique which can also handle local minima in the energy domain; (3) we go beyond common pose estimation methods which are restricted to a small number of orientation bins, delivering fine-grain pose parameters and inferring vehicle shape, instead.

3. METHOD

Our aim is to determine the pose and shape of vehicles detected from street level stereo images acquired from a moving platform with an approximately horizontal viewing direction. To derive the target pose and shape parameters we want to represent each vehicle by a proper 3D vehicle model. For this purpose we use a parametrized deformable model which we try to fit to the detected vehicles based on information derived from the stereo images.

Our framework is depicted in Fig. 1. After a **preprocessing** step, the proposed procedure is divided into the **detection** step, which delivers 3D vehicle detections, and the **modelling** step, in which a deformable vehicle model is fitted to the detected objects. For vehicle detection we use the method described in (Coenen et al., 2017) which we will only recapitulate briefly in this paper. The main focus is on the description of the vehicle model representation and the model fitting strategy for the 3D vehicle reconstruction. The input to our method are calibrated street level stereo images with known interior and relative orientation parameters. Currently, the stereo image pairs are processed individually. We define the left stereo partner to be the reference image and apply dense matching to make use of 3D information in the subsequent

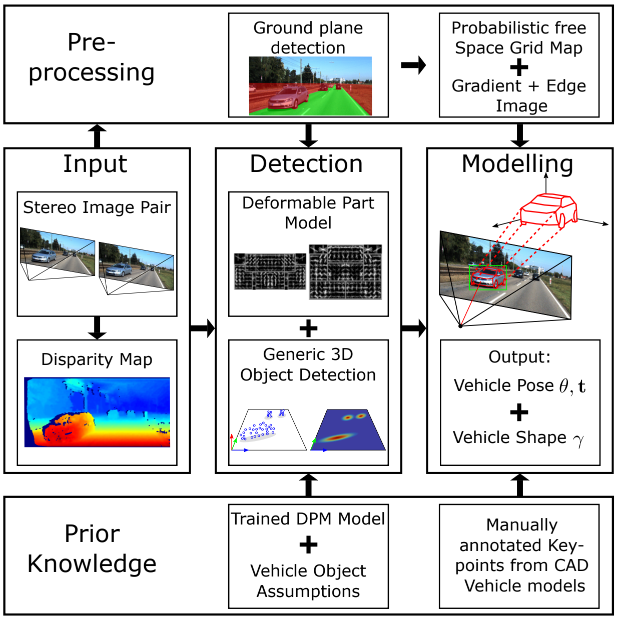


Figure 1. Overview of our framework.

steps. A dense disparity map is calculated for every stereo image pair using the Efficient Large-Scale Stereo Matching (ELAS) method (Geiger et al., 2011). The disparity images are used to reconstruct a 3D point cloud ${}^{M}\mathbf{X}$ in the 3D model coordinate system ${}^{M}CS$ for every pixel of the reference image via triangulation. The origin of the model coordinate system is defined in the projection centre of the left camera. Its x-y plane is parallel to the image plane and its z-axis points in the viewing direction. We discard points further away from the stereo camera than a threshold δ_d. This threshold is determined on the basis of a user-defined maximum allowable threshold for the depth precision δ_{σ_Z}. The dense disparity map and the 3D point cloud serve as the basis for further processing.

3.1 Formal problem definition

Our goal is to describe each stereo scene by a ground plane $\Omega \in \mathbb{R}^3$ and a set of vehicle objects $\mathcal{O}$ that are visible in the stereo images. We want to associate each vehicle object $\mathbf{o}_k \in \mathcal{O}$ with its state vector $(\mathbf{t}_k, \theta_k, \gamma_k)$, where $\mathbf{t}_k$ and θ_k determine the vehicle pose, with its position $\mathbf{t}_k$ represented by 2D coordinates on the ground plane and θ_k being the rotation angle about an axis that is perpendicular to the ground plane (heading); γ_k is a vector of shape parameters determining the shape of a 3D deformable vehicle model representing each object. In this context, we use a 3D active shape model (ASM) (Zia et al., 2013). More details on the vehicle model can be found in Section 3.4.1.

3.2 Preprocessing

Using a stereo image pair and the reconstructed point cloud as input, we detect and extract the ground plane and derive low level features such as gradients and image edges to apply them as additional information and observations in model fitting.

3.2.1 Ground plane extraction: Given our acquisition setup, the 3D points belonging to the ground plane will belong to the set of 3D points with the smallest vertical coordinate (y). We filter the overall point cloud ${}^{M}\mathbf{X}$ by extracting a user-defined percentage p_{gp} of points exhibiting the smallest y-coordinate values. We apply RANSAC to this set of points to find the plane Ω of maximum support, which we assume to correspond to the ground plane. All inliers of the final RANSAC consensus set are stored as ground points ${}^{M}\mathbf{X}_{\Omega} \subset {}^{M}\mathbf{X}$. Additionally to the model coordinate system ${}^{M}CS$ we define a ground plane coordinate system ${}^{\Omega}CS$. We define the origin of the system ${}^{\Omega}CS$ as the orthogonal projection of the origin of the model coordinate system to the ground plane. The y-axis is defined in the direction of the plane normal vector and the x/z-plane lies in the ground plane. We determine the rotation matrix and the translation vector as rigid transformation parameters between the systems ${}^{M}CS$ and ${}^{\Omega}CS$. Using these parameters, any point ${}^{M}\mathbf{x}$ in the model coordinate system can be transformed to a point ${}^{\Omega}\mathbf{x}$ in the ground plane coordinate system.

3.2.2 Region of interest: Assuming that vehicles are always located on the ground plane and do not exceed a maximum height h_{max}, a set of interest points can be extracted from the point cloud by filtering all points not belonging to the ground plane and having a distance from the ground plane smaller than h_{max}. The filtered interest points are stored as ${}^{M}\mathbf{X}_{Int} \subset {}^{M}\mathbf{X}$ with ${}^{M}\mathbf{X}_{\Omega} \cup {}^{M}\mathbf{X}_{Int} = \emptyset$. In addition, the assumption made above allows us to reduce the problem of pose estimation to the 2D problem as described in Section 3.1. For the subsequent procedure we thus transform the previously determined ground plane points ${}^{M}\mathbf{X}_{\Omega}$ and the interest points ${}^{M}\mathbf{X}_{Int}$ to the ground plane system, resulting in ${}^{\Omega}\mathbf{X}_{\Omega}$ and ${}^{\Omega}\mathbf{X}_{Int}$. The proposed methods for vehicle detection and modelling are applied in this domain.

3.2.3 Probabilistic free-space grid map: Based on the points ${}^{\Omega}\mathbf{X}_{\Omega}$ in the ground plane and the extracted interest points ${}^{\Omega}\mathbf{X}_{Int}$ it is possible to reason about free space in the observed scene. We want to represent free space, i.e. areas on the ground plane which are not occupied by any 3D object, by a probabilistic free space grid map Φ delivering a probability for each raster cell of being free space. For this purpose, we create a grid in the ground plane consisting of square cells with a side length l_{Φ}. For each grid cell Φ_g with $g = 1...G$ we count the number of ground points n^g_{Ω} and the number of interest points n^g_{Int} whose vertical projection is within the respective cell. We define the probability ρ_g of each cell to be free space as the ratio of both numbers with

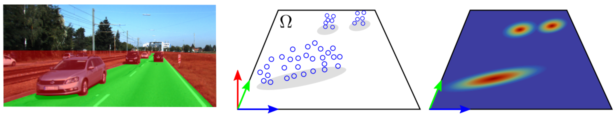
$$\rho_g = \frac{n^g_{\Omega}}{n^g_{\Omega} + n^g_{Int}}. \tag{1}$$

Grid cells without projected points are marked as *unknown*.

3.2.4 Gradient and edge images: We calculate a gradient magnitude image I_{grad} of the reference image using the Sobel operator. Based on I_{grad}, we compute a binary edge image I_{edge} by thresholding the gradient image using the Canny edge detector (Canny, 1986). The gradient and edge images are used as additional data sources for model fitting (cf. Section 3.4.2).

3.3 Vehicle Detection

The goal of this step is to detect all visible vehicles $\mathbf{o}_k$ in the stereo pair by finding their corresponding 3D object points ${}^{\Omega}\mathbf{X}_k$. For vehicle detection we apply the approach described in (Coenen et al., 2017). That method uses both, the 3D points and the image data by fusing a generic 3D object detector with a state-of-the-art vehicle detector in image space, which is expected to result in reliable vehicle detections. The 3D points ${}^{\Omega}\mathbf{X}_{Int}$ inside the region of interest for vehicle detection are projected to the ground plane to obtain a ground plane density map of the 3D points. Assuming that vehicles are surrounded by a band of free space, each vehicle corresponds to a 2D cluster of projected 3D points in the ground plane density map (cf. Figure 2). Quick-Shift Clustering (Vedaldi and Soatto, 2008) is applied to identify the different clusters. This results in generic object proposals, each containing a set of 3D points ${}^{\Omega}\mathbf{X}_k$. A 2D bounding box enclosing the image pixels corresponding to the respective set of 3D points is derived for each object proposal. To reject non-vehicle objects, the DPM (Felzenszwalb et al., 2010) is applied to the reference image. The DPM delivers 2D bounding box detections which are used to verify the vehicle hypotheses resulting from the generic 3D object detection technique by thresholding the intersection over union index of the respective bounding boxes. For more details we refer the reader to (Coenen et al., 2017).



Ground plane (green) and region of interest (red)

3D points of interest ${}^{\Omega}\mathbf{X}_{Int}$

Ground plane density map

Figure 2. Scheme of the generic object detection.

3.4 Pose and shape estimation

Based on the initial vehicle detections we want to reconstruct the vehicles in 3D to recover the pose and shape of each vehicle. For this purpose we make use of vehicle shape priors in the form of a deformable 3D vehicle model representation. We want to fit a vehicle model to each detection, which is achieved by minimising an energy function based on different types of observations using a model-based Monte Carlo Sampling technique.

3.4.1 Model Representation: Like Zia et al. (2013), we learn a 3D active shape model (ASM) as a shape prior for vehicles by applying principal component analysis (PCA) to a set of manually annotated characteristic keypoints of 3D CAD vehicle models. By using vehicles of different types (here: compact car, sedan, estate car, SUV and sports car) in the training set, the PCA results in mean values for all vertex (keypoint) positions as well as the directions of the most dominant vertex deformations. A deformed vehicle ASM is defined by the deformed vertex positions $\mathbf{v}(\gamma)$, which can be obtained by the linear combination

$$\mathbf{v}(\gamma) = \mathbf{m} + \sum_i \gamma^{(i)} \lambda_i \, \mathbf{e}_i \tag{2}$$

of the mean model $\mathbf{m}$ and the eigenvectors $\mathbf{e}_i$, weighted by their corresponding eigenvalues λ_i and scaled by the object specific shape parameters $\gamma^{(i)}$. The variation of the low dimensional shape vector γ thus allows the generation of different vehicle shapes. Figure 3 shows the mean model and two deformed model using a different set of shape parameters. Note how the shape parameters enable the generation of model shapes describing vehicles of different types. For the number of the eigenvalues and eigenvectors to be considered in the ASM we choose $i \in \{1, 2\}$, which we found to be a proper tradeoff between the complexity of the model and the quality of the model approximation. A fully parametrised instance of a 3D vehicle ASM in the ground plane coordinate system, denoted by $M(\mathbf{t}, \theta, \gamma)$, can be created by computing the deformed keypoints using the shape vector γ and subsequently shifting and rotating the whole model on the ground plane according to the translation vector $\mathbf{t}$ and a rotation matrix $R_y(\theta)$ derived from the heading angle θ:

$$M_l(\mathbf{t}, \theta, \gamma) = R_y(\theta) \cdot \mathbf{v}_l(\gamma) + \mathbf{t}, \tag{3}$$

where l is an index for the keypoints. To represent the model surface we define a triangular mesh M_{Tri} for the ASM vertices. To represent the wireframe M_{WF} of the vehicle model, we define *wireframe edges* between selected keypoints. We choose silhouette edges that describe the outline of the vehicle and edges describing distinctive part boundaries, i.e. the transition between semantically different vehicle parts, as wireframe edges. The selected wireframe edges are depicted in Figure 3.

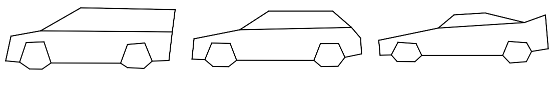


Figure 3. 3D Active Shape Models. Centre: mean shape, $\gamma = (0, 0)$, left: $\gamma_l = (1.0, 0.8)$, right: $\gamma_r = (-1.0, -0.8)$.

3.4.2 Energy function: Given the initial vehicle detections, our aim is to fit a vehicle model $M(\mathbf{t}, \theta, \gamma)$ to each detection by finding optimal values for the variables $\mathbf{t}$, θ and γ by minimising an energy function $E(\mathbf{t}, \theta, \gamma)$:

$$E(\mathbf{t}, \theta, \gamma) = \omega_0 \cdot \bar{E}_{3D} + \omega_1 \cdot \bar{E}_{free} + \omega_2 \cdot \bar{E}_{img}. \tag{4}$$

The function consists of three normalised energy terms $\bar{E}_{(\cdot)}$, each weighted by a weight factor $\omega_{0...2}$. More details on the normalisation of the energy terms is given in Section 3.4.3. The unnormalised energy terms E_{3D}, E_{free}, and E_{img} are based on information obtained in the way described in Section 3.2.

3D energy: The 3D-Energy term E_{3D} is based on the observed 3D vehicle points ${}^{\Omega}\mathbf{X}_k$. It is a score for the Model $M(\mathbf{t}, \theta, \gamma)$ that is determined as the mean distance of the 3D vehicle points from the model surface M_{Tri}:

$$E_{3D} = \frac{1}{P} \cdot \sum_{p=1}^{P} d(\mathbf{x}_p, M_{Tri}). \tag{5}$$

In eq. 5, P is the number of 3D vehicle points and $d(\cdot, \cdot)$ is a function that returns the distance of an individual 3D vehicle point $\mathbf{x}_p \in {}^{\Omega}\mathbf{X}_k$ from its nearest triangle of the model surface. This term tries to fit the 3D ASM to the 3D vehicle point cloud.

Free-space energy: The free space energy term E_{free} takes the probabilistic free space grid map Φ as input data source. In this term, the model $M(\mathbf{t}, \theta, \gamma)$ is evaluated based on the amount of overlap between its 2D ground plane bounding box M_{BB} and the free space grid map cells Φ_g weighted by their probability ρ_g of being free space:

$$E_{free} = \frac{1}{A_{M_{BB}}} \cdot \sum_{g=1}^{G} \rho_g \cdot o(M_{BB}, \Phi_g). \tag{6}$$

In eq. 6, $A_{M_{BB}}$ is the area of the model bounding box. The function $o(\cdot, \cdot)$ calculates the amount of overlap between the model bounding box and a grid cell using the *surveyor's area formula* (Braden, 1986). Thus, this energy term penalises models that are partly or fully located in areas which are actually observed as free space. It acts as substitute information for missing 3D information on the vehicle sides that are invisible to the camera.

Image energy: Additionally to the 3D information considered in the energy terms described so far, image information can also be used directly in the energy function within the *image energy* term E_{img} to evaluate the quality of the correspondence between a model and the observed data. We propose two variants of the image energy term: the *gradient energy* and the *edge energy*.

Gradient energy: In the energy term E_{grad}, the gradient information I_{grad} and the wireframe M_{WF} are considered to obtain a score for the model $M(\mathbf{t}, \theta, \gamma)$. Starting from the assumption that the two types of vehicle edges chosen to define the wireframe correspond to large image gradients, the magnitude of gradients along the backprojected edges of the model wireframe is used as a model score. For this purpose, we backproject the visible parts of the model wireframe to the image, resulting in a binary image I_{WF_b} with entries of 1 at pixels that are crossed by a wireframe edge and 0 everywhere else. We consider differences between the real image gradient positions and the model wireframe caused by generalisation effects of our vehicle model representation by blurring the binary wireframe image using a Gaussian filter, thus transforming I_{WF_b} into a non-binary image I_{WF}. The gradient energy is calculated according to

$$E_{grad} = 1 - \frac{1}{E_{WF}} \cdot \sum_{w=1}^{W} (I_{grad}^{w} \cdot I_{WF}^{w}), \tag{7}$$

where W is the overall number of pixels and $I_{(\cdot)}^{w}$ returns the value of image $I_{(\cdot)}$ at pixel w. For eq. 7, we assume the gradient im-

https://doi.org/10.5194/isprs-annals-IV-2-73-2018 | © Authors 2018. CC BY 4.0 License.

age to be normalised such that both, I^w_{WF} and $I^w_{grad} \in [0, 1]$. E_{WF} is the sum over all grey values in I_{WF} and is used to scale the energy. This energy term becomes small when the backprojected wireframe corresponds well with large image gradients.

Edge energy: In this energy term, the binary backprojected wireframe image I_{WF_b} and the edge image I_{edge} are used to score the Model $M(\mathbf{t}, \theta, \gamma)$ based on the average distance of the backprojected model wireframe edges to image edges. For this purpose, we search the closest non-zero edge pixel in I_{edge} for each wireframe edge pixel in I_{WF_b} along the direction of the respective wireframe edge normal. We define a threshold d_{px} and only consider the number V of pairs of pixels whose distance $u^v_\perp$, with $v = 1...V$, is smaller than d_{px}. The edge energy is calculated by

$$E_{edge} = \frac{1}{V} \cdot \sum_{v=1}^{V} u^v_\perp. \tag{8}$$

This energy term takes small values if the backprojected wireframe is well aligned with the observed image edges.

3.4.3 Energy minimisation: The energy function of eq. 4 is minimized to find the optimal pose and shape parameters for each detected vehicle. As this function is non-convex and the model parameters are continuous, we apply iterative Monte Carlo sampling to approximate the parameter set for which the energy function becomes minimal. To this end we discretise the target parameters by generating model particles for the vehicle ASM. Starting from one or more initial parameter sets, we generate a number of particles n_p in each iteration $j \in [0, n_{it}]$ by jointly sampling the pose and shape parameters from a uniform distribution centered at the preceding parameter values. For the resampling step, we calculate the energy for every particle and introduce the best scoring particles as initial seed particles for the next iteration. In each iteration, the size of the interval from which the parameters are sampled is reduced. In the following paragraphs, more details on the initialisation and the resampling steps are given.

Initialisation: In contrast to (Coenen et al., 2017), where only one initial particle was created, in this work we propose to introduce four initial model particles ${}^0M^i_k({}^0\mathbf{t}_k, {}^0\theta^i_k, {}^0\gamma_k)$ with $i \in [1, 4]$ for every vehicle detection $\mathbf{o}_k$. To initialise the parameters of the particles we create the minimum 2D bounding box enclosing the 2D projections of the 3D vehicle points ${}^\Omega\mathbf{X}_k$ on the ground plane (cf. Figure 4). We define the initial translation vector ${}^0\mathbf{t}_k$ as the centre of the bounding box. The orientations ${}^0\theta^i_k$ of the particles are set to the four orientations of the bounding box semi-axes. By introducing four initial particles with different orientations we expect to be more robust against incorrect orientation estimates compared to only using one initial orientation as in (Coenen et al., 2017). The initial shape parameter vector ${}^0\gamma_k$ is defined as zero vector and, thus, the initial particles correspond to the mean vehicle model.

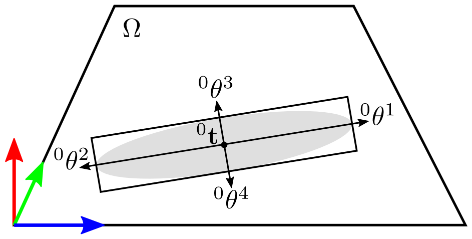


Figure 4. Initialisation of the model particles.

Resampling: In each iteration j we want to find the n_{seed} best scoring particles according to the particle energy in eq. 4. Eq. 4 requires the individual energy terms $E_{(\cdot)}$ to be normalised due to their different units and ranges of values, in order to be able to combine them in one single expression. For this purpose we determine the minimum and maximum energy $E^{min}_{(\cdot)}$ and $E^{max}_{(\cdot)}$ of each energy term resulting from the current particle set and normalise the energy terms of every particle by

$$\bar{E}_{(\cdot)} = \frac{E_{(\cdot)} - E^{min}_{(\cdot)}}{E^{max}_{(\cdot)} - E^{min}_{(\cdot)}} \tag{9}$$

before computing the overall energy. For resampling, we define a number n_{seed} of particles with the lowest energy and forward them to the next iteration as seed particles. By forwarding multiple particles instead of only one particle as in (Coenen et al., 2017) we expect to be able to deal with multi-modal energy distributions and local energy minima in a better way.

Final result: The final values for the target parameters of pose and shape are defined in the last iteration and are set to the parameters of the particle achieving the lowest energy within the particle set of the final iteration.

4. EVALUATION

4.1 Test data and test setup

For the evaluation of our method we use stereo sequences of the KITTI Vision Benchmark Suite (Geiger et al., 2012). The data were captured by a mobile platform in an urban area. We use the object detection and object orientation estimation benchmark, which consists of 7481 stereo images with labelled objects. In our evaluation we consider all objects labelled as *car*. For every object, the benchmark provides 2D image bounding boxes, the 3D object location in model coordinates as well as the rotation angle about the vertical axis in model coordinates. Furthermore, information about the level of object *truncation* and object *occlusion* are available. The values for *truncation* refer to the objects leaving image boundaries and are given as continuous values from 0 (non-truncated) to 1 (truncated). The occlusion state indicates the vehicle's occlusion due to other objects with 0 = fully visible, 1 = partly occluded, 2 = largely occluded and 3 = unknown. We briefly discuss the results for the **vehicle detection** as these results are the input for the proposed pose and shape estimation approach. However, the main focus of the evaluation is on the results for **pose and shape estimation** to analyse the quality of our model fitting approach. For the evaluation, similarly to (Geiger et al., 2012), we define three levels of difficulty as shown in Table 1: *easy*, *moderate* and *hard*, each considering different objects for the evaluation, depending on their level of visibility.

	easy	**moderate**	**hard**
min. bounding box height [Px]	40	25	25
max. occlusion level	0	1	2
max. truncation	0.15	0.30	0.50

Table 1. Levels of difficulty for the evaluation.

We require an overlap of at least 50% between the detected 2D bounding box and the reference bounding box for an object to be counted as a correct detection. In the case of multiple detections for the same vehicle, we count one detection as a true positive, whereas further detections are counted as false positives. For the evaluation of the pose estimation we consider all correctly detected vehicles and compare the 3D object locations $\mathbf{t}_k$ and the orientation angles θ_k of our fitted models to the reference positions and orientations. We consider a model to be correct in position and/or orientation if its distance from the reference position is smaller than $0.75\ m$ and the difference in orientation is less than $22.5°$, respectively.

4.2 Parameter settings

For the 3D reconstruction of the stereo images the maximum value δ_{σ_Z} for the standard deviation of the depth values is defined as 1.5 m. For the specific stereo setup used for the acquisition of the data (cf. Section 4.1), this leads to maximum valid distance of the 3D points from the camera of approximately 24 m.

In all experiments for model fitting, we conduct $n_{it} = 12$ iterations, drawing 140 particles per iteration from $n_{seed} = 8$ seed particles. As initial interval boundaries of the uniform distributions from which we randomly draw the particle parameters, we choose ± 1.5 m for the location parameter $\mathbf{t}_k$, ± 2.5 for the shape parameter vector γ_k and $\pm 45°$ for the orientation θ_k. By choosing $\pm 45°$ as range for the orientation angle of the four initial seed particles we allow particles to take the whole range of possible orientations in the first iteration to be able to deal with incorrect initialisations. In each iteration j the size of the interval boundaries is decreased by a factor 0.85^j. With n_{it} = 12, this leads to a reduction of the final interval range to 14% of the initial width.

To assess the impact of the individual components in the model fitting procedure, we define five different variants with different settings for the generation of particles and for the calculation of the energy terms. The variant **Base** uses a setting that is comparable to the method used by (Coenen et al., 2017). This is achieved by setting $\omega_0 = 1$ and $\omega_{1,2} = 0$ to only consider the 3D energy term for the model fitting. Instead of four initial particles we only create one single particle with an initial orientation ${}^0\theta = {}^0\theta^1$, using an initial interval width of $\pm 180°$ for the orientation. For the particle generation we change n_{seed} to 1. Variant **3D** also only considers the 3D energy term in the model fitting procedure; it differs from *Base* by the settings for the particle generation, i.e. by considering four initial particles with different initial orientations and by increasing the number of seed particles from 1 to 8. In variant **3D+Free**, we add the free-space energy term to the energy function and choose $\omega_0 = 0.8$ and $\omega_1 = 0.2$. To evaluate the full energy function for the model fitting we set $\omega_0 = 0.7$, $\omega_1 = 0.2$ and $\omega_2 = 0.1$. We distinguish between **Full**$_e$ and **Full**$_g$ in which E_{img} is substituted by E_{edge} and E_{grad}, respectively. The values for the weight factors were found empirically. In the last setting, referred to as **Refine**, we apply an adaptive model fitting strategy by using the *Base+Free* setting for coarse initial pose estimation and the *Full*$_g$ setting for a subsequent refinement. For this purpose we vary the weight factors in the iterations. In the first $n_{it} - 1$ iterations we set ω_2 to zero and thus only consider the terms E_{3D} and E_{free} for model fitting. In the last iteration we also include E_{img} by using the *Full*$_g$ parameter setting to leverage the image information for final pose and shape refinement.

4.3 Vehicle detection results

Table 2 shows the values for completeness (the percentage of reference vehicles that were detected), correctness (the percentage of detections that actually are vehicles) and quality (a trade-off parameter combining completeness and correctness) (Heipke et al., 1997) resulting from the vehicle detection approach. We consider these results to be very satisfactory. Compared to Coenen et al. (2017), there is a considerable improvement in all quality indices (up to 9%) due to a better ground plane estimation.

	easy	**moderate**	**hard**
Completeness [%]	94.3	86.4	71.4
Correctness [%]	88.6	92.3	93.2
Quality [%]	84.0	80.6	67.9

Table 2. Vehicle detection results.

4.4 Pose estimation results

Table 3 shows the results of the comparison between the resulting pose parameters from the fitted 3D vehicle models and the reference data for location and orientation of the vehicles. The table contains the percentage of the correctly estimated positions $\mathbf{t}$ and orientations θ and the mean absolute errors for position $\hat{\varepsilon}_t$ and for orientation $\hat{\varepsilon}_\theta$ of the correctly determined models in [cm] and [°], respectively. Comparing the results for different levels of difficulty, we can see a similar pattern of performance for all variants. That is, all variants perform best for the *easy* level and worst for the *hard* level. Independently from the level of difficulty, the percentage of vehicles for which a correct position is determined only differs by about 4% between the different approaches. It may seem counter-intuitive that the positional errors grow with the number of energy terms that are considered (the best values are achieved for *3D*), but these differerences are very small (a few cm, about 10% of the magnitude of the errors). As there are larger differences in the orientation estimation results, we focus on an analysis of the results of the orientations in the following paragraphs. Figure 5 shows a histogram of differences between the vehicle orientations derived by our methods and the reference orientations for all correct detections from the *easy* level.

		Base	3D	3D+Free	Full$_g$	Full$_e$	Refine
easy	$\mathbf{t}$ [%]	71.1	73.2	73.6	72.6	72.6	73.2
	$\hat{\varepsilon}_t$ [cm]	37.9	37.6	38.6	39.7	39.4	39.0
	θ [%]	58.4	74.8	82.4	80.7	80.5	81.2
	$\hat{\varepsilon}_\theta$ [°]	4.3	3.8	3.6	3.4	3.4	3.4
moderate	$\mathbf{t}$ [%]	69.3	72.0	72.0	71.3	71.2	71.6
	$\hat{\varepsilon}_t$ [cm]	37.6	37.2	38.3	39.5	39.0	38.9
	θ [%]	54.9	72.0	76.6	75.8	74.8	75.2
	$\hat{\varepsilon}_\theta$ [°]	4.4	3.9	3.7	3.5	3.5	3.6
hard	$\mathbf{t}$ [%]	66.7	69.6	69.9	68.7	69.1	69.5
	$\hat{\varepsilon}_t$ [cm]	37.7	37.1	38.2	39.3	38.9	38.7
	θ [%]	52.6	68.7	73.2	71.9	71.3	71.8
	$\hat{\varepsilon}_\theta$ [°]	4.4	3.9	3.8	3.6	3.6	3.7

Table 3. Pose estimation results.

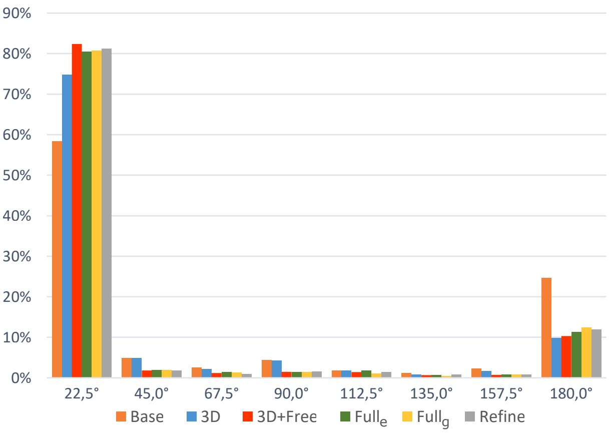


Figure 5. Histogram of absolute differences between estimated and reference orientations (correct detections of the *easy* level).

Base: Applying this setting, equivalent to (Coenen et al., 2017), only leads to correct orientation estimations in between 52% and 58% of the cases, depending on the difficulty level (Table 3), and a mean orientation error of up to 4.4°. Figure 5 shows that a large amount of false orientation estimations (about 25%) are in the last orientation bin, indicating an error of 180°. This effect was already observed in (Coenen et al., 2017) and was found to be caused by incorrect orientation initialisations and/or the almost symmetric 3D shape of vehicles w.r.t. their minor half axis.

3D: This variant differs from variant *Base* by an enhanced particle fitting procedure. Table 3 shows that this leads to a distinct increase of correct orientation estimations by up to 17% and to a decrease of the mean orientation error of about 0.5°. Figure 5 shows clearly that the desired effect of the advanced particle fitting strategy was achieved, resulting in a rigorous reduction of the incorrect orientation estimations in the last orientation bin.

3D+Free: In this approach we incorporated information about observed free space in the scene to the model fitting process. According to Table 3, this leads to an improvement of the number of correct orientation estimates of up to 7.6% and to a reduction of $\hat{\varepsilon}_\theta$ of about 0.2°. Using this approach we achieve the largest number of correct orientation estimates with more than 82% correctly determined vehicle orientations for the *easy* level. Figure 5 shows that this improvement is caused by a reduction of the false orientation estimations in the intermediate orientation bins while the amount of incorrect orientation estimations of the last bin remains unchanged compared to *3D*. We consider this as a natural effect of the free space energy term as it is not able to distinguish between two vehicles with opposite viewing directions.

Full$_g$ and **Full$_e$:** Here we incorporate image data in the model fitting process in the form of gradient and edge information, respectively. As Table 3 shows, *Full$_g$* and *Full$_e$* achieve numerically very similar results, with *Full$_g$* performing slightly better. However, the Full energy settings lead to a decrease in the number of correct orientation estimations of up to 1.7% compared to the *3D+Free* setting. The reason for that can be that non-vehicle gradients and non-vehicle edges or edges resulting from reflections on the vehicle or from shadows can distort the image energy terms due to incorrect gradient and edge associations with the model wireframe. Besides, due to the generalisation of the ASM, the wireframe of the model could possibly differ too much from some of the real world vehicle shapes and thus the image energy terms are not able to support the fitting procedure. Nevertheless, the mean orientation error of the correctly determined vehicle orientations decreases, too. This effect also becomes apparent in Figure 6, which shows a cumulative histogram of the absolute differences between the estimated orientations and the reference using a bin width of 1°. The histogram covers correctly determined orientations with differences smaller than 10°. The cumulative percentage of correct orientation estimations for the *Full$_e$* and *Full$_g$* settings are always better than for the *3D+Free* approach. That is, when the estimated orientation is within the first orientation bin and, thus, the image-based energy terms do not lead to deviations of the orientation values for the reasons just described, the gradient and edge energy terms are able to improve and to refine the orientation result.

Refine: To investigate this effect further, in this variant we only consider the image energy terms in the last iteration of the particle fitting procedure. As a result, the amount of correct orientation estimations increases again (up to 81.2% as in Table 3) while the refining effect of the image energy terms being apparent from the improved mean orientation error and the better behavior of the cumulative histogram in Figure 6 is maintained. Moreover, while vehicles with an estimated orientation offset to the reference in a range of 22.5° are already considered as correct estimations, Figure 6 shows that more than 90% of the correct orientation estimations are even correct within a range of 8° or smaller using the *Full* and *Refine* approaches, but the latter is obviously more robust against divergence in the early stages of the model fitting process, leading to a correct solution in more cases.

The quality of shape estimation is not quantitatively evaluated in this work. However, Figure 7 shows two representative examples underlining the benefit of including the image energy terms for model fitting as the shape and/or the orientation estimation is distinctly improved using the *Refine* setting compared to *3D+Free*.

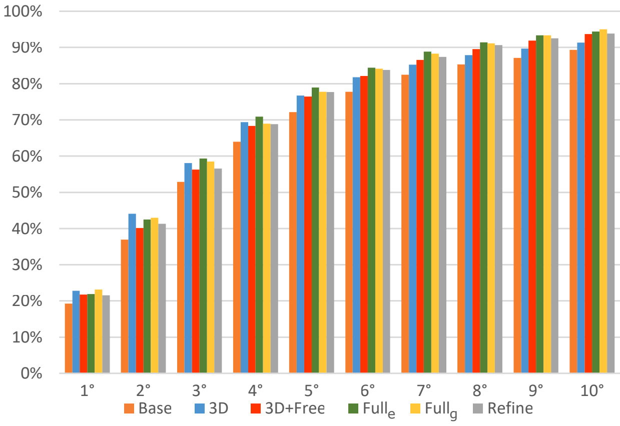


Figure 6. Cumulative histogram of absolute differences between estimated and reference orientation (correct orientation estimations of the different approaches, *easy* level).

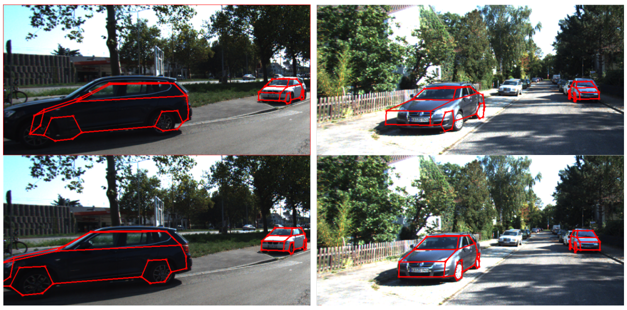


Figure 7. Results of *3D+Free* (first row) and *Refine* (second row)

5. CONCLUSION

We have developed an approach to estimate the pose and shape of vehicles from stereo image pairs. For this purpose we developed a robust Monte Carlo vehicle model fitting technique using an active shape model as shape prior to recover the vehicles in 3D and to infer their position and orientation. To this end, we defined an energy function incorporating reconstructed 3D data, recovered 3D scene knowledge, low-level image information and vehicle model features. Our results show that the advanced particle fitting technique as well as the incorporation of observed free space to the model fitting procedure improves the pose estimation results, especially the results for the vehicle orientation, significantly. Considering gradient or edge information in the energy function could refine the correct orientation estimations. However, non-vehicle gradient and edge data can distort the energy function, leading to slightly fewer correct orientation estimates. Furthermore, the generalisation of the ASM can lead to incorrect associations between image gradients or edges and the model wireframe. To overcome this problem, a more detailed and fine-grained vehicle model can be applied in the future by adding more keypoints to the ASM and its wireframe. Besides, the energy function gives room for extensions. On the one hand, the free space energy term can be extended from 2D to 3D by incorporating free space voxels instead of the free space grid into the model fitting process. Furthermore, the energy function can be extended by computing the gradient and edge energy terms not only in the reference image but in both stereo images to consider additional observations from a different viewpoint. Further, the current state of our work does not comprise occlusion awareness, which will be an essential extension in the future. Another possibility to incorporate image information more robustly into model fitting can be achieved by using a keypoint classifier trained for the individual ASM keypoints. Its classification output for the particle model keypoints can be incorporated using an additional energy term in the model fitting. Also, until now the parameters and weights for the particle model fitting are found empirically. These parameters can be learned, e.g. in a Monte Carlo simulation. Finally, in the future we will make use of the shape estimations results to reason about vehicle categories, the vehicle type or even to recognize individual vehicles.

ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) as a part of the Research Training Group i.c.sens [GRK2159].

REFERENCES

Bao, S. Y., Chandraker, M., Lin, Y. and Savarese, S., 2013. Dense Object Reconstruction with Semantic Priors. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1264–1271.

Braden, B., 1986. The Surveyor’s Area Formula. *The College Mathematics Journal* 17(4), pp. 326–337.

Canny, J., 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6), pp. 679–698.

Coenen, M., Rottensteiner, F. and Heipke, C., 2017. Detection and 3D Modelling of Vehicles from terrestrial Stereo Image Pairs. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-1/W1, pp. 505–512.

Cootes, T., Baldock, E. and Graham, J., 2000. An Introduction to Active Shape Models. In: *Image Processing and Analysis*, pp. 223–248.

Dame, A., Prisacariu, V. A., Ren, C. Y. and Reid, I., 2013. Dense Reconstruction using 3D Object Shape Priors. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1288–1295.

Engelmann, F., Stückler, J. and Leibe, B., 2016. Joint Object Pose Estimation and Shape Reconstruction in urban Street Scenes using 3D Shape Priors. In: *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 9796, pp. 219–230.

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

Geiger, A., Lenz, P. and Urtasun, R., 2012. Are we ready for autonomous Driving? The KITTI Vision Benchmark Suite. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361.

Geiger, A., Roser, M. and Urtasun, R., 2011. Efficient Large-Scale Stereo Matching. In: *Computer Vision – ACCV 2010*, Lecture Notes in Computer Science, Vol. 6492, pp. 25–38.

Güney, F. and Geiger, A., 2015. Displets: Resolving stereo Ambiguities using Object Knowledge. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4165–4175.

Heipke, C., Mayer, H., Wiedemann, C. and & Jamet, O., 1997. Evaluation of automatic Road Extraction. In: *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXII, pp. 151–160.

Knuth, J. and Barooah, P., 2009. Distributed collaborative Localization of multiple Vehicles from relative Pose Measurements. In: *Conference on Communication, Control, and Computing*, pp. 314–321.

Leibe, B., Leonardis, A. and Schiele, B., 2006. An Implicit Shape Model for combined Object Categorization and Segmentation. In: *Toward Category-Level Object Recognition*, Lecture Notes in Computer Science, Vol. 4170, pp. 508–524.

Liebelt, J. and Schmid, C., 2010. Multi-view Object Class Detection with a 3D geometric Model. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1688–1695.

Lin, Y.-L., Morariu, V. I., Hsu, W. and Davis, L. S., 2014. Jointly Optimizing 3D Model Fitting and Fine-Grained Classification. In: *European Conference on Computer Vision (ECCV)*, pp. 466–480.

Menze, M., Heipke, C. and Geiger, A., 2015. Joint 3d Estimation of Vehicles and Scene Flow. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. II-3, pp. 427–434.

Ozuysal, M., Lepetit, V. and Fua, P., 2009. Pose Estimation for Category specific multiview Object Localization. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 778–785.

Payet, N. and Todorovic, S., 2011. From Contours to 3D Object Detection and Pose Estimation. In: *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 983–990.

Pepik, B., Stark, M., Gehler, P. and Schiele, B., 2012. Teaching 3D Geometry to deformable Part Models. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3362–3369.

Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T. and van Gool, L., 2007. Depth-From-Recognition: Inferring Meta-data by Cognitive Feedback. In: *2007 IEEE 11th International Conference on Computer Vision (ICCV)*, pp. 1–8.

Vedaldi, A. and Soatto, S., 2008. Quick Shift and Kernel Methods for Mode Seeking. In: *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, Vol. 5305, pp. 705–718.

Villamizar, M., Grabner, H., Moreno-Noguer, F., Andrade-Cetto, J., van Gool, L. and Sanfeliu, A., 2011. Efficient 3D Object Detection using multiple Pose-Specific Classifiers. In: *British Machine Vision Conference*, pp. 20.1–20.10.

Xiao, W., Vallet, B., Schindler, K. and Paparoditis, N., 2016. Street-Side Vehicle Detection, Classification and Change Detection using mobile Laser Scanning Data. *ISPRS Journal of Photogrammetry and Remote Sensing* 114, pp. 166–178.

Zia, M. Z., Stark, M. and Schindler, K., 2015. Towards Scene Understanding with detailed 3D Object Representations. *International Journal of Computer Vision* 112(2), pp. 188–203.

Zia, M. Z., Stark, M., Schiele, B. and Schindler, K., 2013. Detailed 3D Representations for Object Recognition and Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(11), pp. 2608–2623.