Classification and Pose Estimation of Vehicles in Videos by 3D Modeling within Discrete-Continuous Optimization

Michael Hödlmoser[1] Branislav Micusik[2] Ming-Yu Liu[3*] Marc Pollefeys[4] Martin Kampel[1]

[1] CVL, Vienna University of Technology
[hoedl,kampel]@caa.tuwien.ac.at

[2] AIT Austrian Institute of Technology
branislav.micusik@ait.ac.at

[3] University of Maryland College Park
mingyliu@umiacs.umd.edu

[4] Computer Vision and Geometry Lab, ETH Zürich
marc.pollefeys@inf.ethz.ch

Abstract

This paper presents a framework for classification and pose estimation of vehicles in videos by assuming their given 3D models. We rank possible poses and types for each frame and exploit temporal coherence between consecutive frames for refinement. As a novelty, first, we cast the estimation of a vehicle's pose and type as a solution of a continuous optimization problem over space and time. Due to a non-convexity of this problem, good initial starting points are important. We propose to obtain them by a discrete temporal optimization reaching a global optimum on a ranked discrete set of possible types and poses. Second, to guarantee effectiveness of the proposed discrete-continuous optimization, we present a novel way to efficiently reduce the search space of potential 3D model types and poses for each frame for the discrete optimizer. It avoids common expensive evaluation of all possible discretized hypotheses. The key idea towards efficiency lies in a novel combination of detecting the vehicle, rendering the 3D models, matching projected edges to input images, and using a tree structured Markov Random Field to get fast and globally optimal inference and to force the vehicle follow a feasible motion model in the initial phase. Quantitative and qualitative experiments on a variety of videos with vast variation of vehicle types show superior results to state-of-the-art methods.

1. Introduction

Object classification and pose estimation are two important tasks in the application of vehicle surveillance. Solving them enables automatically analyzing a traffic scene (*e.g.* determination of vehicle speed, traffic frequency, driver behavior, or shape and style of the traffic participants). Due to occlusion handling, most of these metrics need a 3D reconstruction of a scene, which is traditionally captured of surveillance cameras as 2D image stream. By representing a 3D scene in 2D, the depth information gets lost and going back to 3D from a single viewpoint is therefore an ill-posed problem. In computer vision, the same object may be captured under varying lighting conditions and different poses. Therefore, traditional object detection algorithms extract features from multiple input images to cover at least a discrete set of variations in order to relax the problem of this so called intra-class variations [2]. These approaches were also extended to view-invariant detection methods by learning a sparse 3D object model from multiple training images [15, 14, 18]. To overcome both of these problems, generic frameworks for object classification based on a discretely synthetically rendered dataset of 3D models are presented in [3, 10, 19, 8, 15]. The generic class of methods has shown its great success in a variety of tasks, however,

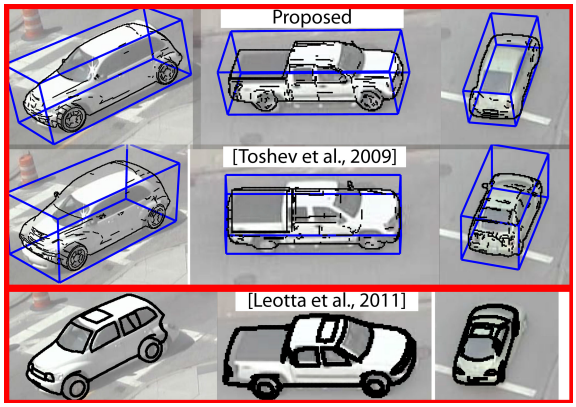


Figure 1. We significantly improve pose estimation over [19] by processing in continuous space (columns 1, 2), reduce wrong classifications due to incorrect scales (column 3) and improve pose estimation over deformable model approaches [8] by using existing 3D models.

*Ming-Yu Liu is now with Mitsubishi Electric Research Laboratories.

the drawback always is their high cost due to extensive evaluation of the hypotheses and inevitable coarse discretization of the model space. This paper tackles both these issues and moves thus forward these class methods towards their applicability. We consider the case of video sequences rather than single images, as such setups are prevalent in surveillance applications.

This paper presents a framework for estimating a vehicle's pose and type by ranking possible poses and types for each frame and exploit temporal coherence between consecutive frames for refinement. The contribution is twofold. First, we propose to define the problem of estimating a vehicle pose and type as a solution over all possible poses and types along a sequence as a continuous optimization in space and time. To solve the problem in continuous space, starting points are required due to the non-convex objective function, which are proposed to be obtained by an initial discrete optimizer reaching a global optimum on a discrete set of ranked types and poses. Hence, the proposed strategy is referred to as a discrete-continuous optimization method. As a second contribution, to cope with computational complexity of the generic class of methods, we present a novel way to efficiently reduce the search space over the vehicle poses and types for the discrete optimizer. We show how a state-of-the-art object detector and Fast Directional Chamfer Matching (FDCM), an OpenGL renderer, the Ackermann principle in the vehicle motion model, and a tree structured Markov Random Field (MRF) to get fast and globally optimal inference in the initial phase can be combined to improve current approaches [19, 8]. As can be seen in Fig. 1, using continuous optimization eliminates two main problems of *e.g.* [19], namely i) discrete pose estimation errors (rows 1,2:columns 1,2) and ii) one scaling of a wrong model may fit better than the best discrete scaling of the correct model (rows 1-2:column 3). By exploiting 3D models with known dimensions, the approach does also improve pose estimation over methods using generic models (row 3).

Different to pose estimation methods working on still images [5], we are not optimizing each frame separately but exploit temporal consistency in the video for temporal inference and refinement. This means it relaxes the computational cost of finding the best solution for each frame by exploiting temporal coherence as a strong prior. We never predict a pose from one frame to a subsequent one as proposed in [8], but evaluate a set of poses for each frame and find the best combination of vehicle types and poses over time. This brings us the advantage of having multiple hypothesized poses available for each frame and a higher accuracy in terms of pose estimation, as can be seen in the experiments.

We first apply a 2D object detector [1] to obtain some initial hypotheses of the vehicle positions in the images. This brings the advantage of not having a noisy localization compared to background subtraction. We then assume to have a calibrated setup and a known ground plane (as others, *e.g.* [8]), to lift the hypotheses to 3D space. This gives us a rough estimate of the vehicle's 3D trajectory. We then hypothesize a variety of poses which are used to render all 3D models and apply the FDCM algorithm [11, 12] to rank them. FDCM, which works superior in terms of speed and accuracy compared to conventional Chamfer matching [11], uses object contours which have been shown to be robust in case of low resolution images or texture-less objects [8, 15]. Exploiting edges for matching the projections of 3D models to 2D images gives us the advantage of not dealing with any noisy foreground segmentation, as in [19]. Finally, we propose a tree-structured MRF model to compute an optimal pose trajectory in the discrete domain which forces the vehicle to only move with feasible and constant motion. This serves as a reliable initial point for solving the final continuous optimization problem. The optimization is done by exploiting a least squares method, where the distances between projected 3D model points and corresponding 2D edge points in the input images are used to improve the pose estimation result.

2. Related Work

Neuroscience studies have pointed out that a human's brain works with 3D representation of objects and somehow (unclear how) stores the 3D information to achieve object detection and recognition at the level we humans have been experiencing. That drove the structural object detection and tracking solutions over the last years to utilize structural nature of objects to reproduce (at least) human ability on machines. Using 3D models as an input for tracking and pose estimation approaches in images and videos has been studied extensively in the last few years. Vehicle classification using synthetic 3D models was first introduced in [7]. One of the first deformable 3D models used for car detection is described in [3]. Complex 3D models were introduced in [10] to localize cars and motorcycles in still images. The training set consists of multiple projections of the 3D object, over which a camera is orbiting. Combining these geometric features with appearance features learned from 2D images is demonstrated in [9, 6, 4]. The approach of [10] was developed further in [19], where videos are used as an extension to still images. In [20], a fitness score and a particle filter are used for tracking. In [13], vehicles are tracked by exploiting existing 3D models and a Kalman filter. The authors of [8] proposed an approach using a combination of a deformable 3D model and a Kalman filter for tracking a vehicle in a video.

In [8], tracking is done by changing parts of a deformable 3D model and predicting the pose by a Kalman filter. We are not using a deformable model since on the one hand, hav-

ing more parameters to optimize to get the shape of the car can fail due to multiple local minima. On the other hand, not having the strong prior knowledge about the vehicle's shape gives worse tracking results, as can be seen in our experiments. To sum up [5], they work on single images, gain an initial pose from meta-data and match 2D images to 2D projection by conventional Chamfer distance. Rendering is done on manually labeled semantic parts. They exploit an MRF for connecting the right semantic parts of the vehicle. Exploiting temporal inference gives us two advantages over [5], namely outliers are rejected and poses are refined over time. We use a state-of-the-art object detector and FDCM for ranking all possible projections for each frame, which works superior to using meta-data as in [5] in terms of speed, accuracy and practicality [11]. The main drawback of the approach presented in [19] is the inevitable perfect foreground segmentation for each frame and the limited use because of only choosing the best pose from a discrete set of possible object poses. We are able to overcome this problem by introducing a continuous set of possible poses and relying on edges for matching instead of background subtraction.

In this paper we follow the aforementioned strategy to detect and classify a vehicle, and to estimate its pose from calibrated 2D input videos utilizing known 3D models. To the best of our knowledge, there is no existing work where classification and pose estimation is done continuously in terms of space and time, which enables higher accuracy compared to state-of-the-art methods.

3. Classification and Pose Estimation

This paper describes an efficient approach for accurately detecting a vehicle's pose and its type in continuous space, given a calibrated video sequence and a set of 3D models containing a large variety of vehicles. As [8], we assume, that given the ground plane, the vehicle's pose is parameterized by $p = (x, y, \alpha)$ as shown in Fig. 2(b). The car's centroid on the ground plane is therefore denoted as $C = (x, y, z = 0)$, its orientation is described by the angle α. The whole pipeline can be summarized in the following steps. We first detect the object in each single frame by using [1], which does not provide any 3D information. By re-projecting the 2D detection in 3D space, we get a rough guess on the pose of the object. By using FDCM, we then obtain the k most similar models by measuring the distance between the projected edges of all 3D models and the edges in the scene, generated by a Canny edge detector. These steps are described in Section 3.1. We then apply small variations on the model's pose, measure the similarity by combining area overlap and FDCM and put all the projections in an MRF. By introducing a simple motion model and solving the MRF using a forward-backwards algorithm, the best fitting projection sequence for a given video sequence is determined. The temporal alignment is explained in Section 3.2. A final pose refinement, described in Section 3.3, is applied to get an optimal result in continuous space. Fig. 2(a) shows the whole workflow of the proposed framework.

3.1. Vehicle Detection, Classification and Orientation Estimation

Matching vehicles in image sequences can be a hard task since these objects provide specular but large texture-less surfaces, different shapes and different colors. The most stable features which can be used are their edges. For comparing a 3D model to an input frame we therefore need to render the model. This equals to detecting visible edges, which can arise from sharp edges between two adjacent faces of the 3D model and from the silhouette of its projection. For speed reasons, the whole rendering pipeline is computed on the GPU. Given a vehicle's pose and a calibrated camera, we exploit the visibility constraint to obtain for each pixel the face which is the closest visible to the camera center. For finding all the sharp edges of a model, we calculate the normal vectors for adjacent faces, $\mathbf{n}_i$ and $\mathbf{n}_j$. A sharp edge is found, when $|\mathbf{n}_i \cdot \mathbf{n}_j| \leq$ threshold, meaning that $\mathbf{n}_i$ and $\mathbf{n}_j$ are pointing in different directions. The threshold is empirically chosen to be 0.95 ($= 20.00°$). A visible edge must therefore pass the visibility and the sharp edge constraint. We first need to roughly locate the vehicle in the image. As can be seen in [19] it is clearly not enough to perform background subtraction to obtain an accurate foreground mask due to highlights and shadows in the scene. We use a state-of-the-art object detector [1], which returns a bounding box for the 2D vehicle location. Afterwards, the bounding box's centroid is re-projected on a horizontal plane at a certain height above the ground plane. As proposed for track initialization in [8], this height is assumed to be one meter. The gathered 3D point is projected on the ground plane and aligned with the 3D model's centroid on the ground plane. We now have a rough estimate of x and y. The procedure is applied to the first subsequent frames. By fitting a spline through these points, we obtain an initial guess for the orientation α.

The goal of the next step is to determine the class of the vehicle as well as the refined pose. Since the type of the target vehicle in the frame is unknown, we would need to apply small variations of the initial pose guess ($\pm 5°$ in our experiments) for all the models in the data set, render all poses and perform a similarity search using FDCM to obtain the matching score s. For additional speed-up, we perform this step only on the k best fitting modes where the k best models are found by ranking the FDCM scores of the models in the initial pose. We empirically found that keeping the hypotheses for the best two models is sufficient in our experiments. For the determination of the matching score using FDCM, let U and E be the edge maps of the rendered model and the

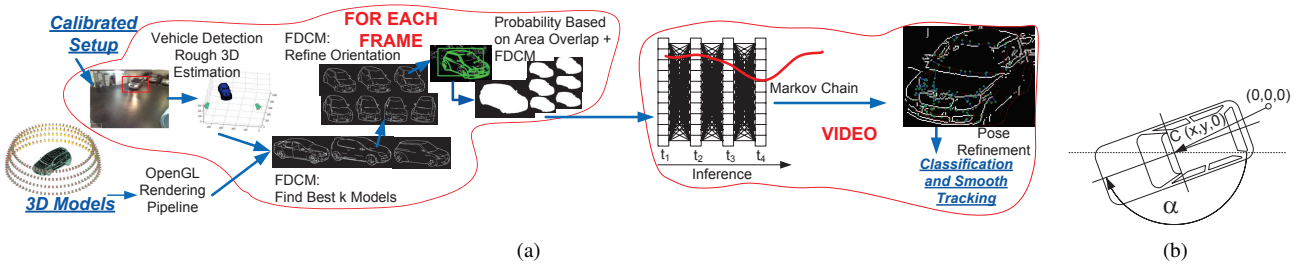


Figure 2. (a) Framework application flow. See text for details. (b) Vehicle, described by orientation α and centroid on the ground plane $C = (x, y, 0)$.

input image respectively. FDCM maps the edge pixels in U and E to an orientation augmented space. The alignment cost between the two edge maps is then given by

$$d(U,E) = \sum_{\mathbf{u}_i \in U} \min_{\mathbf{e}_j \in E} ||\mathbf{u}_i - \mathbf{e}_j||_2 + \lambda |\angle(\mathbf{u}_i) - \angle(\mathbf{e}_j)| \quad (1)$$

where $\angle(\cdot)$ refers to the orientation of the edge pixel and λ is the weighting factor on the orientation term. The best alignment $\hat{d}$ is then the rigid transformation in 2D image space $\mathbf{r} \in SE(2)$ minimizing $\hat{d} = \min_{\mathbf{r} \in SE(2)} d(U_\mathbf{r}, E)$. Here we use $U_\mathbf{r}$ to denote the transformation of the edge map U with the parameter $\mathbf{r}$. In [11], it is shown that via line-segment approximation of the edge maps the FDCM cost can be evaluated efficiently using an integral distance transform structure. Such an approximation is particular beneficial for our case since the structure of vehicles generally follows some straight line pattern. Moreover, the prior that a vehicle lies evenly on the ground further eliminates the need to search for an in-plane rotation and speeds up the matching process. We then update the matching score by setting

$$s_{l,\mathbf{p}} = 1 - \hat{d}_{l,\mathbf{p}}, \quad (2)$$

where $l = 1 \cdots k$, $\hat{d}_{l,\mathbf{p}}$ is the normalized FDCM matching score within $[0, 1]$ for model l, rendered with pose $\mathbf{p}$. FDCM returns the similarity score between the edge image and the rendered 3D model as well as its 2D location. It therefore shifts the projection on the image plane to get the best possible match, which results in a re-projection error due to not caring about projective correctness. To handle this, we render the best k models by aligning the 2D output location of the FDCM with the vehicle's centroid again and use pose variations $\mathbf{q}$ (in our experiments $\mathbf{p} \pm 80$ centimeters and $\pm\, 2°$). Given the shifted but projective wrong model projection $A_l^\mathbf{p}$ and a projective correct model projection area $B_l^\mathbf{q}$, the similarity score for a pose is calculated by combining the output of FDCM and the area overlap by

$$s_l^\mathbf{q} = \frac{s_{l,\mathbf{p}} + \left(\frac{|A_l^\mathbf{p} \cap B_l^\mathbf{q}|}{|A_l^\mathbf{p} \cup B_l^\mathbf{q}|} \right)}{2}. \quad (3)$$

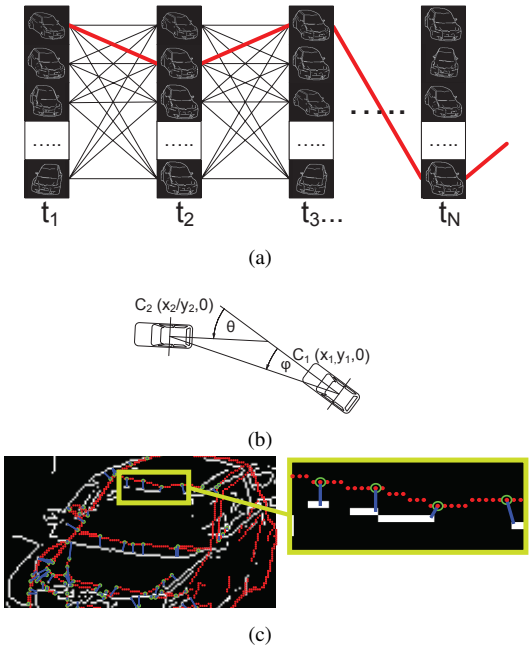


Figure 3. (a) Temporal inference for ranked projections. (b) Ackermann steering principle where $\varphi = \frac{\theta}{2}$. (c) Corresponding points between model's projected edges and edge image.

3.2. Temporal Model Alignment

Consider an input vehicle sequence $\mathcal{V} = \{v_1, ...v_t...v_N\}$ and multiple model sequences $\mathcal{M}_l = \{m_{l,1}...m_{l,t}...m_{l,N}\}$, where l is obtained using the method described in Section 3.1, v_t is the projection of the vehicle in the input video and $m_{l,t}$ is the projection of model l at time t. We want to find the best matching model sequence $\hat{\mathcal{M}}$, which means to find the best fitting model at each time instance t. This implies that i) the whole sequence provides the same model type and the vehicle moves with ii) constant and iii) feasible motion. This problem can be solved by calculating the sequential inference as shown in Fig. 3(a). The inference can be calculated by using an MRF, which is a chain-structured undirected graphical model, where the joint distribution for

$\mathcal{M}_l$ given $\mathcal{V}$ for each 3D model l is determined by

$$P(\mathcal{M}_l|\mathcal{V}) = \frac{1}{Z(\mathcal{V})}\prod_{t=1}^{N} F(m_{l,t}|\mathcal{V})F(m_{l,t}, m_{l,t-1}|\mathcal{V}), \quad (4)$$

where $Z(V)$ is a partition function guaranteeing a probability distribution, $F(m_{l,t}|\mathcal{V})$ is the matching score between a model projection $m_{\mathbf{q}}$, where $\mathbf{q}$ denotes a pose variation, and vehicle at time instance t. $F(m_{l,t}, m_{l,t-1}|\mathcal{V})$ denotes the transition of model l between consecutive frames. The matching score term for each frame is simply determined by

$$F(m_{\mathbf{q}}|v_t) = s_l^{\mathbf{q}}. \quad (5)$$

We assume that a vehicle moves continuous over time. According to [17] and [16], the Ackermann steering principle ensures a feasible vehicle movement by applying different but defined turning radii for the inner and outer wheels of the car. The principle is shown in Fig. 3(b). Combining continuous and feasible motion leads to

$$F(m_{l,t}, m_{l,t-1}|\mathcal{V}) = \exp(-(\|\mathbf{p}_{l,t-1}-\mathbf{p}_{l,t}\|^2+\lambda_2(\varphi-\frac{\theta}{2}))), \quad (6)$$

where λ_2 is a weighting constant guaranteeing an equal impact of both terms on the final outcome. For solving Eq. (4) and finding the best fitting $\mathcal{M}_l$ for a given $\mathcal{V}$, we need to determine the path having the maximum probability through the graph and compute both Eq. (5) for each of the model projections and Eq. (6) for each edge, where an edge should be from each projection of frame at time $t-1$ to each one at time t. Since the model type determined in this step must not change over time, we solve Eq. (4) for each vehicle type l. After establishing the whole graph for each type, the problem becomes a labeling problem, where we need to find the best matching projection sequence $\hat{\mathcal{M}}_l = \text{argmax}_{\mathcal{M}_l} P(\mathcal{M}_l|\mathcal{V})$ for each class l. This inference can exactly be solved by a forward-backwards algorithm. We then find the vehicle type w for the best fitting model sequence by $w = \max_{l=1}^{k}\left(P(\hat{\mathcal{M}}_l|\mathcal{V})\right)$. The optimized discrete sequence is then $\hat{\mathcal{M}} = \hat{\mathcal{M}}_w$.

3.3. Pose Refinement

Using the MRF, for each frame we can only choose the best fitting projection out of discretely rendered 3D models, which means that the results of the previous step depend on the stepsize of the discretization. To refine the pose, we carry out a continuous optimization of the car's parameters in 3D space. We therefore propose to use a least squares method where the distances between projected 3D model points and image edge points are minimized. At time t, we consider Q_t 3D model points $\mathbf{A}_t$ for which their projected 2D points $\mathbf{a}_t$ are part of a rendered edge. For all points $\mathbf{a}_t$, we use the Euclidean distance to find the closest corresponding points $\hat{\mathbf{a}}_t$ in the input edge image. These corresponding points are shown in Fig. 3(c). To assure a smooth movement of the vehicle, we additionally minimize the distance between two consecutive poses over time. Given a camera matrix $\mathtt{K}$, its rotation and translation $\mathtt{R}$ and $\mathbf{t}$, we calculate the error for a set of discretely optimized pose vectors $\hat{\mathcal{P}} = \{p_1...p_t...p_N\}$ for a set of model projections $\hat{\mathcal{M}}$ by

$$\varepsilon(\hat{\mathcal{P}}) = \sum_{t=1}^{N}\frac{\lambda_3}{Q_t}\sum_{i=1}^{Q_t}\|\mathtt{K}[\mathtt{R}\ \mathbf{t}]\mathbf{A}_{t,i}-\hat{\mathbf{a}}_{t,i}\|^2+\sum_{t=2}^{N}\|\mathbf{p}_{t-1}-\mathbf{p}_t\|^2, \quad (7)$$

where λ_3 is a weighting constant. Note that projected 3D points $\mathbf{A}_{t,i}$ have been converted from homogeneous to 2D coordinates. The iterations are performed by updating the poses $\hat{\mathcal{P}}^{j+1} = \hat{\mathcal{P}}^j + \Delta\hat{\mathcal{P}}$. The pose update $\Delta\hat{\mathcal{P}}$ is determined by $\mathtt{J}_\varepsilon^T\mathtt{J}_\varepsilon\Delta\hat{\mathcal{P}} = \mathtt{J}_\varepsilon^T\varepsilon$, where $\mathtt{J}_\varepsilon$ is the Jacobian matrix determined for each ε of Eq. (7). At every iteration, we calculate the points $\mathbf{a}_{t,i}$, $\hat{\mathbf{a}}_{t,i}$ for each time instance t and minimize the squared error between all pairs as well as between consecutive poses. After this step, the final projection set $\hat{\mathcal{M}}$ is updated according to $\hat{\mathcal{P}}$. Fig. 4 shows the results of the pose estimation using simply FDCM (top row), the improved pose estimation result using an MRF (middle row) and the final result using the proposed discrete-continuous optimization (bottom row).

4. Experiments

Experiments are carried out using eight calibrated real world data sequences, where the sequence numbers correspond to rows in Fig. 7(bottom). These show six different types of vehicles, provide between 30-150 frames, have a resolution between 640x480 (sequence 1) and 1280x720 (sequences 2-8, taken from [8]) pixels and are captured from different viewpoints. To obtain a classification rate, we downloaded six corresponding CAD models from the Internet for the cars shown in the videos. The dimensions of the models are taken from the manufacturer's specifications. Fig. 5(c) shows the six models used, which are from left to right: Chevrolet Silverado 2500HD, Chrysler PT Cruiser, VW Beetle, Toyota RAV4, Chevrolet Blazer and Skoda Fabia.

Quantitative experiments: We show the output for each step of our pipeline (using FDCM only (No Opt), applying the MRF (MRF) and applying pose refinement (Opt)) and compare them to [19] [8]. For generating ground truth data, we manually segment the 2D area of the vehicle as foreground for each frame. Since Toshev's approach is not publicly available, we re-implemented it and used the manual segmentation as input. This perfect foreground localization prevents wrong classifications due to a bad segmentation. To assure fairness in our comparison, we re-implemented

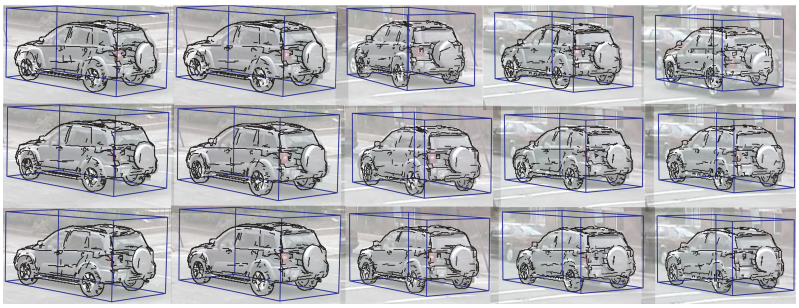
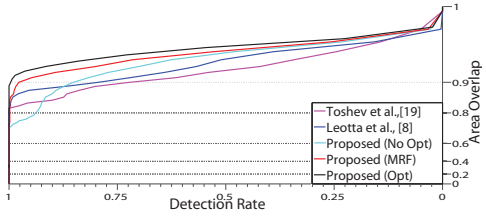


Figure 4. Pose estimation using FDCM only (top row), combining FDCM and MRF (middle row), combining FDCM, MRF and continuous optimization (bottom row).

	Proposed(Opt)	Proposed(MRF)	Proposed(No Opt)
μ(cm)	24.68	25.68	28.63
σ(cm)	11.93	13.24	16.38
max(cm)	66.29	87.08	126.08
	Toshev et al.,[19]	Leotta et al.,[8]	
μ(cm)	35.47	28.75	
σ(cm)	31.49	15.06	
max(cm)	288.45	120.17	

(a)



(b)

(c)

Figure 5. (a) Offsets of the vehicle's center on the ground plane compared to ground truth. (b) Area overlap over all sequences for correctly classified vehicles. (c) Used models.

the method in such a way that we get comparable performance on similar videos as presented in [19]. The dataset is generated with an azimuth, elevation and distance step-size of 5°, 10° and 50 centimeters, respectively. Leotta's approach [8] is publicly available. Due to the blur in sequence 1, this approach cannot be used on it and therefore we excluded this sequence for the quantitative experiments. First, we compare the overlap between the ground truth region and the projected 3D model. As this metric is used for the evaluation of the detected pose, it is only valid when correctly classified frames are used. In our experiments, we obtain a correct classification for all sequences, whereas Toshev's method gets the correct class in sequences 3-6 and 8. Since Leotta is using a deformable model, it cannot be directly used for classification and therefore all frames are labeled as correctly classified. The graph in Fig. 5(b) shows the vehicle detection rate for correctly classified frames at a certain amount of overlap for all the steps of our implementation (No Opt, MRF, Opt) as well as for Toshev's and Leotta's approach. As can be seen, we are able to outperform both Toshev's and Leotta's method. Toshev's approach is using all scales of a discretely rendered model, which are available in the dataset. This can lead to misclassifications, where the wrong, discrete scale of a wrong model may fit better than the next best discrete scale of the correct model. As can be seen in Fig. 5(b), this does not directly influence the overlap between ground truth and the detected model. Therefore we also compare the offset of the vehicle's centroid on the ground plane to the ground truth data (Table 5(a)). As can be seen, we clearly outperform both methods when using our optimized results. Leotta's performance is worse due to the deformable model and Toshev's method is worse since it does not incorporate ground plane estimation but does only classify each single frame based on the area overlap between training data and the input frame. We provide a maximum deviation to the ground truth of 66.29 cm, Leotta 120.17 cm and Toshev even up to 288.45 cm. Fig. 6 shows the trajectory of the detected vehicle's centroid over a whole sequence for each method. The vehicle's starting position is denoted by $\times$, $*$, $\star$, $\bigtriangledown$, $\diamondsuit$, $\circ$ and $+$ for Opt, MRF, No Opt, Leotta, Toshev, ground truth and Opt(model n/a) respectively. The results of sequences 1-8 are shown in columns 1-4, ordered from left to right and top to bottom. As can be seen, there are more jumps in space between consecutive frames using Toshev's method than using ours since we assume the 3D model to be located on the ground plane. The last column shows a comparison between our optimized result with and without having the best matching vehicle in the data set.

Qualitative experiments: Fig. 7 shows qualitative example results using our approach. One row shows one sequence, where the leftmost image of each row presents the first frame of it and our optimized projected track. The following frames of the row provide the refined pose of the detected class, projected onto the image plane. For viewing purposes, we crop out the region around the vehicle. Rows 1-8 show result images for sequences 1-8, where the correct 3D model is within the dataset. We plot the bounding box and the projection of the best fitting 3D model. Note that also blurry images (*e.g.* sequence 1) and occlusions (se-

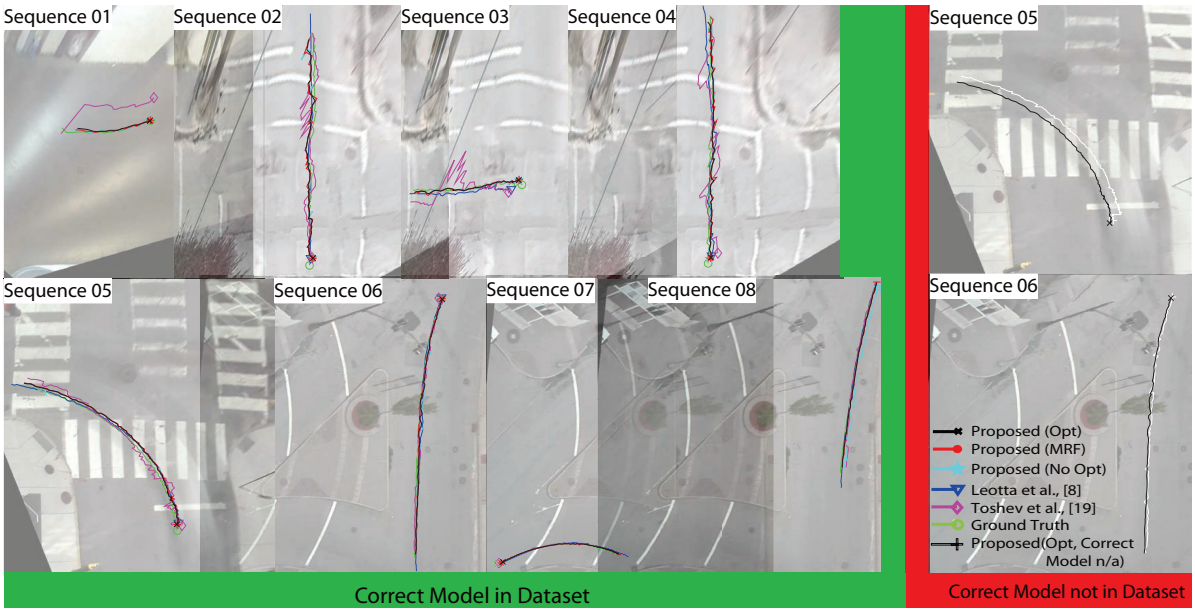


Figure 6. Columns 1-4: 3D tracks of car's centroid on warped top-view for all sequences. Proposed method (No Opt, MRF, Opt), [19], [8], ground truth. Column 5: Optimized output with and without best matching model in dataset. See text for details.

quence 6, frame 3) are handled over time by our framework. Rows 9-10 show results where the correct model is not in the dataset.

5. Conclusion

In this paper we proposed to describe the task of car classification and pose estimation in videos as a continuous optimization problem, which can practically be solved by discrete-continuous optimization. We obtained good starting points, which are essential for the ensuing continuous optimization problem, by a discrete optimization reaching a global optimum on the given discrete set. In both our quantitative and qualitative results we showed that we clearly outperform state-of-the-art methods from both approaches, namely those using deformable models and those using models with known dimensions. It could also be seen that our method handles two drawbacks of purely discrete solutions, namely huge pose estimation errors and preferring a wrong model at a wrong scale over the correct one.

Acknowledgments: The authors would like to thank Sebastian Zambanini for his valuable help, the members from the Computer Vision and Geometry Group / ETH Zurich for fruitful discussions, Matt Leotta for his dataset, code and support, and the reviewers for valuable comments. The work was supported by WWTF project ICT08-030, FFG project 830042 (CAPRI) and CogVis Ltd.

References

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32:1627–1645, 2010. 2, 3

[2] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005. 1

[3] J. Ferryman, A. Worrall, G. Sullivan, and K. Baker. A generic deformable model for vehicle recognition. In *BMVC*, 1995. 1, 2

[4] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011. 2

[5] Y. Guo, C. Rao, S. Samarasekera, J. Kim, R. Kumar, and H. S. Sawhney. Matching vehicles under large pose transformations using approximate 3d models and piecewise mrf model. In *CVPR*, 2008. 2, 3

[6] S. Khan, H. Cheng, D. Matthies, and H. Sawhney. 3d model based vehicle classification in aerial imagery. In *CVPR*, 2010. 2

[7] D. Koller. Moving object recognition and classification based on recursive shape parameter estimation. In *ICAI*, 1993. 2

[8] M. Leotta and J. Mundy. Vehicle surveillance with a generic, adaptive, 3d vehicle model. *PAMI*, 33(7):1457–1469, 2011. 1, 2, 3, 5, 6, 7

[9] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010. 2

[10] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *CVPR*, 2008. 1, 2

[11] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, 2010. 2, 3, 4

[12] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, A. Agrawal, and H. Okuda. Pose estimation in heavy clutter using a multi-flash camera. In *ICRA*, 2010. 2

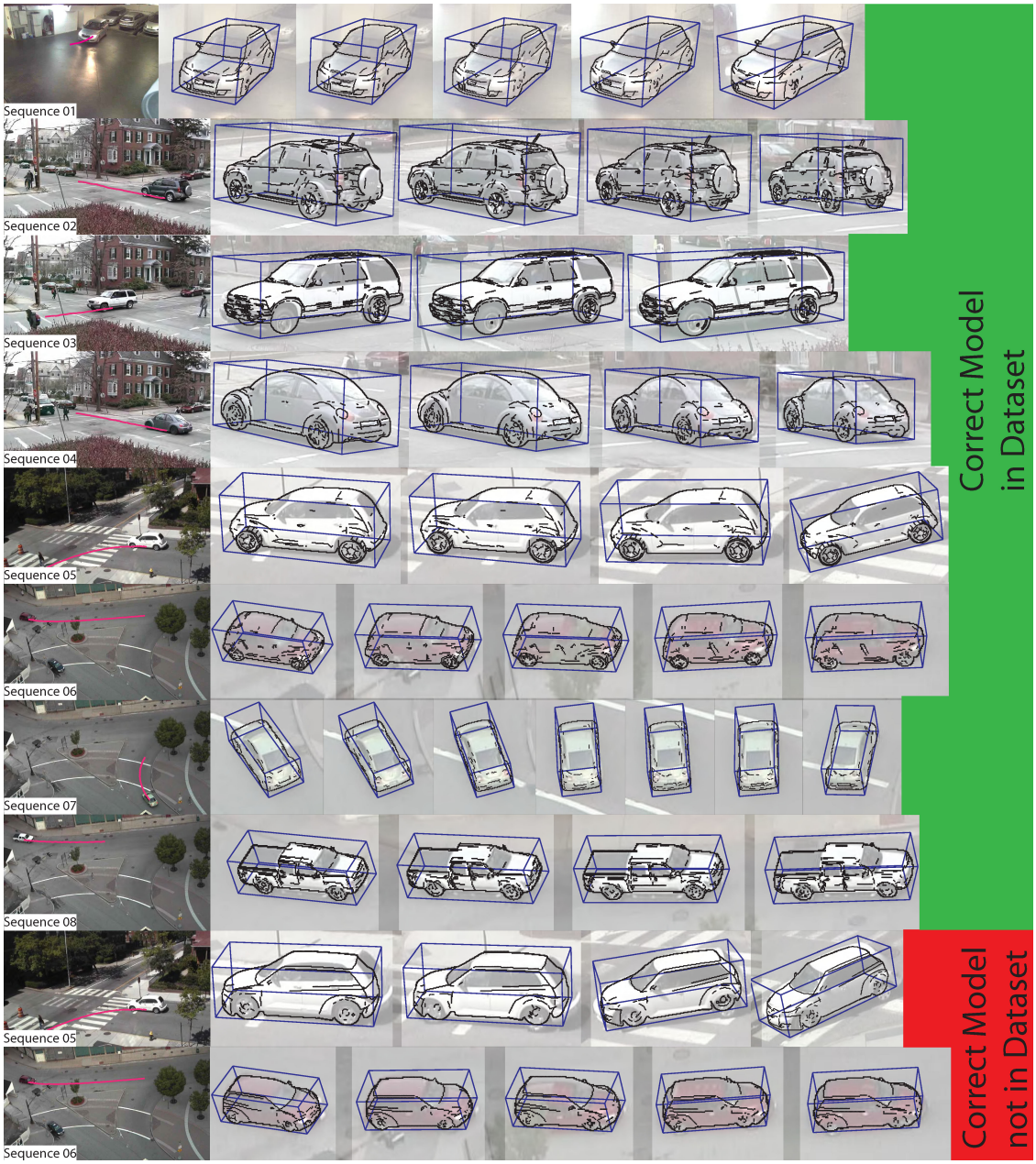


Figure 7. Qualitative results. One row describes one sequence where leftmost image shows first frame of it and our optimized projected track. For rows 1-8, correct model is in dataset, for rows 9-10 it is not. See text for details.

[13] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank. 3-d model-based vehicle tracking. *Image Processing*, 14:1561–1569, 2005. 2

[14] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009. 1

[15] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, 2011. 1, 2

[16] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *ICRA*, 2009. 5

[17] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004. 5

[18] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006. 1

[19] A. Toshev, A. Makadia, and K. Daniilidis. Shape-based object recognition in videos using 3d synthetic object models. In *CVPR*, 2009. 1, 2, 3, 5, 6, 7

[20] Z. Zhang, M. Li, K. Huang, and T. Tan. 3d model based vehicle localization by optimizing local gradient based fitness evaluation. In *ICPR*, 2008. 2