Vehicle Tracking by Simultaneous Detection and Viewpoint Estimation

Ricardo Guerrero-Gómez-Olmedo[1], Roberto López-Sastre[1], Saturnino Maldonado-Bascón[1], and Antonio Fernández-Caballero[2]

[1] GRAM, Department of Signal Theory and Communications, UAH, Alcalá de Henares, Spain.
[2] Department of Computing Systems, UCLM, Albacete, Spain.

Abstract. We address the problem of vehicle detection and tracking for traffic monitoring in Smart City applications. We introduce a novel approach for vehicle tracking by simultaneous detection and viewpoint estimation. An Extended Kalman Filter (EKF) is adapted to describe the vehicle's motion when not only the pose of the object is measured, but also its viewpoint with respect to the camera. Specifically, we enhance the motion model with observations of the vehicle viewpoint jointly extracted by the detection step. The approach is evaluated on a novel and challenging dataset with different video sequences recorded at urban environments, which is released with the paper. Our experimental validation confirms that the integration of an EKF with both detections and viewpoint estimations results beneficial.

Keywords: vehicle tracking, vehicle detection, tracking by detection, viewpoint estimation, Smart City

1 Introduction

Within the context of Smart Cities, there are several relevant applications which need a robust system for detecting and tracking the vehicles in the scene. Some examples are vehicle speed estimation [1] or illegal parking detection [2].

In a vehicle tracking application, a fundamental part of the pipeline is the object detection step. However, we want to argue that it is also beneficial to incorporate to the tracking model the observation of the object's viewpoint. Can we recover this information jointly during the detection step? How can we efficiently integrate these pose observations into the tracking approach? These are just two of the questions we want to answer with this work.

Object class detection and recognition in images and videos has been a very popular research theme over the last years (*e.g.* [3,4,5,6,7]). That is, the objective of all these works has been to estimate the bounding boxes in the images in order to localize object of interest within them. Although it is a much less researched area, some recent works propose to deal with the problem of estimating the viewpoint of the objects (*e.g.* [8,9,10,11,12]). We do believe that this viewpoint observation can be beneficial for a tracking model. For instance, if we humans look at the car shown in Figure 1, we are able to infer its pose, and consequently to predict a *logical* direction for its movement.

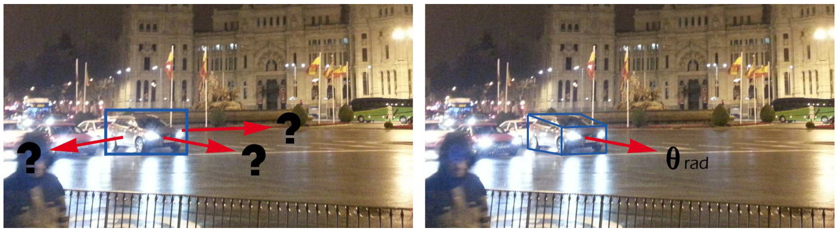


Fig. 1. We humans are able not only to detect an object, but also to estimate its viewpoint or orientation. Furthermore, we are able to use this *semantic* information to estimate a logical direction for the movement of the object of interest. For instance, if a car is observed under a frontal orientation, we will predict that it will move towards the camera position.

In this paper we propose a novel approach for vehicle tracking, using the Extended Kalman Filter (EKF), which is able to simultaneously integrate into the motion model both the position and the viewpoint of the object observed. The approach is evaluated on a novel and challenging dataset with three video sequences recorded at urban environments. We publicly release the dataset, with the ground truth annotations, to provide a common framework for evaluating the performance of vehicle detection and tracking systems within the context of smart city applications.

The paper is structured as follows. In Section 2 we review related work. Section 3 introduces a detailed description of the tracking system. Section 4 presents experimental results and Section 5 concludes the paper.

2 Related work

The tracking-by-detection approach has become very popular recently [13,14,15]. A common problem of most of this type of works is that the bounding boxes are not adequate to constrain the object motion sufficiently. This really complicates the estimation of a robust trajectory. On the other hand, following the tracking-by-detection philosophy one is able to work in complex scenes and to provide automatic reinitialization by continuous application of an object detector.

For the tracker, we use an EKF [16]. Other approaches have been proposed, like the color based Mean-Shift[17] and Cam-Shift[18]. They are lightweight and robust, but in a traffic urban scene, typically crowded with vehicles, they are not the best choice.

Some model-based tracking approaches use a 3D model, of a particular target object, in order to estimate its precise pose [19,20]. However, it is hard to run them in complex outdoor settings where many different objects are present.

This paper builds on state-of-the-art object detection and viewpoint estimation approaches and leverages recent work in this area [11,7]. Specifically, we propose to learn a system for simultaneous detection and viewpoint estimation, following a similar learning strategy to the one introduced in [11], but using the ground-HOG detector [7]. This system is further integrated into the dynamic motion model of an EKF, which is used to track the vehicles in the scene.

3 Tracking by detection and viewpoint estimation

We present a new approach to address the problem of vehicle tracking via simultaneous detection and viewpoint estimation of the target objects. Essentially, ours is a tracking-by-detection approach which incorporates the observations of the viewpoint of the objects into the EKF motion model. This way, by adequately parameterizing the scene, an object detection can be accompanied by a viewpoint, which is subsequently associated to an orientation of the movement (see Figure 1).

For the tracking system, we decided to use the EKF [16]. A simple motion model considering just the position and the speed of the vehicles using a Kalman Filter (KF) is enough in some cases. To enhance its performance, we use the discrete and non-linear version of the KF, *i.e.* the EKF, with the Ackermann steering model [21] for vehicle's non-holonomic motion. One of its main disadvantage is that, as the EKF is a Taylor's linearized version of the KF, it quickly diverges if the process is not perfectly modeled, or if we are not able to get measures in a certain interval. We try to avoid these limitations using the pose recovered from the detector to estimate the orientation of the object movement. In order to track vehicles in crowded scenes, where occlusions are one of the main problems to deal with, the EKF, with an adequate motion model, results very convenient, specially if we compare it with other tracking approaches based in color features (*e.g.* [18,17]).

We start briefly describing the object detection and viewpoint estimation step in Section 3.1. Then, we offer a detailed description of how we integrate these observations into the object tracking pipeline (Section 3.2).

3.1 Vehicle detection and pose observation

The basis for a tracking-by-detection and viewpoint estimation approach are the object *detections*, which are defined as follows,

$$\overline{d}_t^{(i)} = [x_t^{(i)}, y_t^{(i)}, \theta_t^{(i)}]^T \,, \tag{1}$$

where, $(x_t^{(i)}, y_t^{(i)})$ encodes the 2D position of the object, and $\theta_t^{(i)}$ corresponds to the estimation of the viewpoint, for an object i at time stamp t. For the sake of clarity, we shall mostly omit the superscript i in the following.

Inspired by [11], we propose to to learn a set of viewpoint vehicle detectors for four particular viewpoints: frontal, rear, left and right. Instead of using the Deformable Part Model [6], we use the HOG based model described in [7]. Specifically, for learning the set of detectors we follow the next approach. We learn a HOG template for each viewpoint independently. Each template is refined using the hard negatives, as described in [3]. Because the objective is to provide a precise viewpoint estimation, when training for a particular pose (*e.g.* frontal), the negative examples may be extracted from images with the same object class but from the opposite viewpoint (*e.g.* rear). During detection, and in order to combine all the outputs for the different viewpoints into a single detection response, we follow a bounding box based non-maximum-suppression step on the individual outputs. This way, the object detection step is able to feed the tracking motion model with object detections like $\overline{d}_t$.

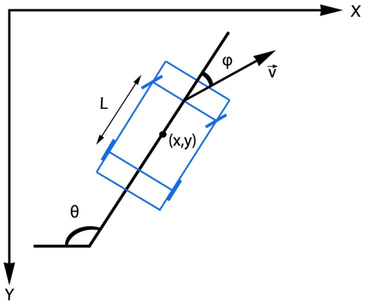


Fig. 2. Ackermann Steering model [21]

3.2 Vehicle Tracking

We define a measurement vector $\overline{z}_t \in \mathbb{R}^3$ as $\overline{z}_t = [x_t, y_t, \theta_t]^T$, and we assume that the tracking process has a state vector $\overline{x}_t \in \mathbb{R}^6$ as $\overline{x}_t = [x_t, y_t, \theta_t, v_t, \phi_t, a_t]^T$, where: x_t and y_t encode the position of the object in the image (*i.e.* the center of the bounding box), θ_t defines the orientation of the movement, ϕ_t is the steering angle, and v_t and a_t are the linear speed and the tangential acceleration, respectively. This formulation corresponds to the Ackermann steering model for cars [21]. See Figure 2 for a graphical representation of the dynamic model.

We use an EKF in combination with this dynamic model to describe the motion of the vehicles. The EKF is a recursive Bayesian filter which iteratively repeats two steps at each frame: first, it estimates the object state $\overline{x}_t^-$ by applying the dynamic model to the previous state $\overline{x}_{t-1}$; second, it updates the resulting state to the corrected state $\overline{x}_t$ for the current frame by fusing it with the new observation $\overline{d}_t$.

According to the dynamic model proposed, we define the following state transition function $f : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ as

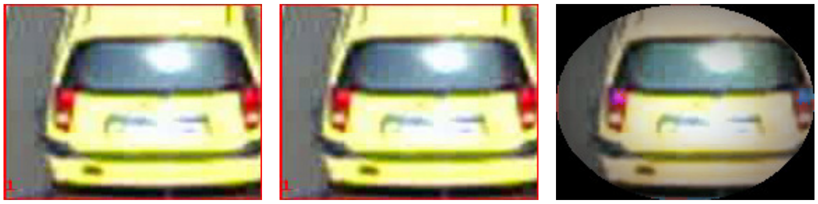
$$\overline{x}_t = f(\overline{x}_{t-1}) = \begin{bmatrix} x_{t-1} + v_{t-1}cos(\theta_{t-1})\Delta t + \frac{1}{2}a_{t-1}cos(\theta_{t-1})\Delta t^2 \\ y_{t-1} + v_{t-1}sin(\theta_{t-1})\Delta t + \frac{1}{2}a_{t-1}sin(\theta_{t-1})\Delta t^2 \\ \theta_{t-1} + \frac{1}{L}v_{t-1}tan(\phi_{t-1})\Delta t \\ v_{t-1} + a_{t-1}\Delta t \\ \phi_{k-1} \\ a_{t-1} \end{bmatrix} . \quad (2)$$

Note that L is the distance between the axles of the car, we fixed it to the value of 3.2 m.

We follow a hypothesize-and-verify framework for the proposed tracker. Each vehicle trajectory hypothesis is defined as $H^{(i)} = [D^{(i)}, A^{(i)}]$, where $D^{(i)}$ denotes its supporting detections, and $A^{(i)}$ is the appearance model for the vehicle. For $A^{(i)}$, we choose an $(8 \times 8 \times 8)$-bin color histogram in HSV space. As in [15], given a bounding box via the detector, we do not directly compute the histogram for all its pixels. Instead, we preprocess the image within this detection window. Rejecting the portion of the bounding box that is not vehicle is extremely important in order to be able to perform a good matching from one frame to another. Also, we must be resistant to small color variations produced by illumination changes. In order to accentuate the pixels located at the center, we process the image inside each bounding box using a Gaussian kernel to weight each pixel at position x and y as follows,

$$\alpha_{x,y} = e^{\frac{(x-x_c)^2}{(\frac{w}{\delta})^2} + \frac{(y-y_c)^2}{(\frac{h}{\delta})^2}} , \tag{3}$$

where w and h are respectively the width and the height of the bounding box where the histogram is computed, x_c and y_c encode the center of the bounding box, and δ is an empirical constant with a value of 2. Finally, we compute the histogram in HSV color space only for those weighted pixels inside an ellipse fitted to the bounding box. This final step allows us to ignore portions of the image located at the corners that are typically asphalt. Figure 3 graphically shows how the preprocessing pipeline works.



(a) (b) (c)

Fig. 3. Preprocessing of the detection window before computing its histogram. a) Original image. b) Image weighted with a Gaussian kernel. c) Image masked with an ellipse.

Every time a new observation $\overline{d}_t^{(i)}$ is added to a trajectory, we update the appearance model as follows, $A^{(i)} = a_t^{(i)}$, where $a_t^{(i)}$ encodes the appearance of the new measure. If there is no measure available, the appearance model is updated using the last estimation. For the data association step, we measure the similarity of an object and its hypothesis using the Bhattacharyya distance [22] between the histograms,

$$d(a_t^{(i)}, A^{(i)}) = \sqrt{1 - \sum_j \frac{a_t^{(i)}(j) \cdot A^{(i)}(j)}{\sqrt{\sum_j a_t^{(i)}(j) \cdot \sum_j A^{(i)}(j)}}} . \tag{4}$$

4 Results

4.1 Experimental setup

We have tested our approach in the novel *GRAM Road-Traffic Monitoring* (GRAM-RTM) dataset. It consists of 3 challenging video sequences, recorded under different conditions and with different platforms. The first video, called `M-30` (7520 frames), has been recorded in a sunny day with a Nikon Coolpix L20 camera, with a resolution of 640×480 @30 fps. The video was scaled to 2000×1200. The second sequence, called `M-30-HD` (9390 frames), has been recorded in the same place but during a cloudy day, and with a high resolution camera: a Nikon DX3100 at 1280×720 @30 fps. In this case, the video was scaled at 2980×1788. The third video sequence, called `Urban1` (23435 frames), has been recorded in a busy intersection with a low-quality traffic camera with a resolution of 480×320 @25fps. This video was also scaled to 2980×1788. Figure 4 shows some examples of the images provided in the dataset. All the vehicles in the GRAM- RTM dataset have been manually annotated using the tool described in [23]. The following categories are provided: `car`, `truck`, `van`, and `big-truck`. The total number of different objects in each sequence is: 256 for `M-30`, 235 for `M-30-HD` and 237 for `Urban1`. Note that we provide a unique identifier for each vehicle. All the annotations included in the GRAM-RTM were created in an XML format PASCAL VOC compatible [24]. The vehicles that appear within the red areas shown in the second row of Figure 4 have not been annotated, hence any detection in these areas must be discarded before the experimental evaluation. We publicly release the GRAM-RTM dataset[3], including the images, the annotations, and a set of tools for accessing and managing the database. Our aim is to establish a new benchmark for evaluating vehicle tracking and road- traffic monitoring algorithms within the context of Smart City applications.

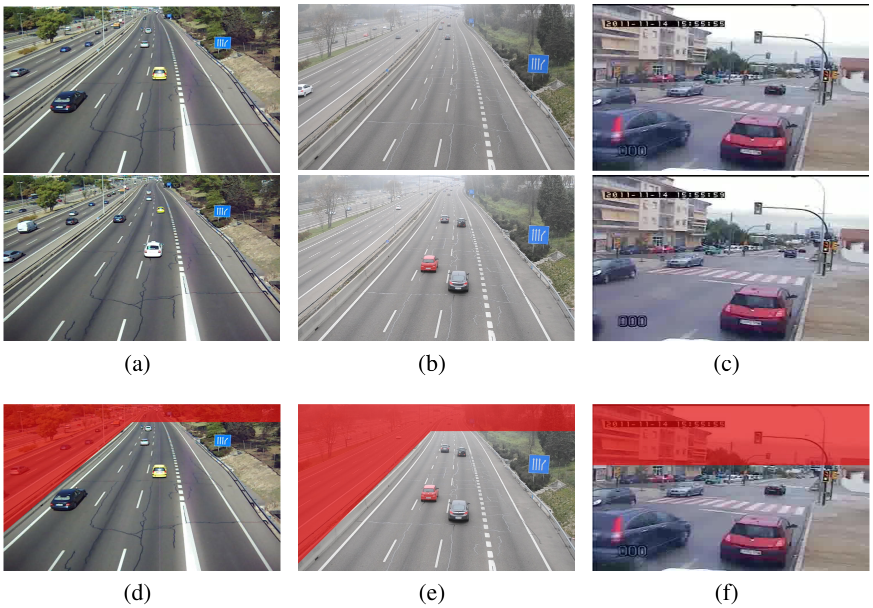
For establishing further fair comparisons, we encourage to use for training the systems any data except the provided sequences. The test data provided with these sequences must be used strictly for reporting of results alone - it must not be used in any way to train or tune systems, for example by running multiple parameter choices and reporting the best results obtained.

We define two evaluation metrics. First, for detection, we propose to use the Average Precision (AP), which is the standard metric used in the object detection competition of the last PASCAL VOC challenges [24]. For the evaluation of the tracking, inspired by [25], we propose to use the AP and precision/recall curves too. That is, for each estimated bounding box given by the tracking ROI_T, we measure its overlap with the ground truth bounding box provided ROI_G. An estimation is considered valid if it is over a threshold τ,

$$\frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)} > \tau \, . \tag{5}$$

Normally, for object detection performance evaluations in the PASCAL VOC challenge, $\tau = 0.5$. In our tracking scenario, we propose to use a less restrictive threshold

[3] `http://agamenon.tsc.uah.es/Personales/rlopez/data/rtm/`



(a) (b) (c)

(d) (e) (f)

Fig. 4. GRAM Road-Traffic Monitoring dataset images. Row 1: Examples of images for the sequences: a) `M-30`, b) `M-30-HD`, c) `Urban1`. Row 2: Exclusion areas (in red). Vehicles within these areas have not been annotated.

of 0.2 (for both the detection and tracking AP computations). This strategy has been also considered within the context of object detection [26].

Note that with this new criterion, when a tracking systems is going to be evaluated, we can consider the four cases represented in Figure 5. Basically, the AP is adequate for measuring the four situations proposed. However, for penalizing the situation drawn in Figure 5(d), we propose to also compare the different methods by providing the number of vehicles counted during the sequence. We do believe that with these three evaluations metrics, AP for detection, AP for tracking, and number of vehicles tracked, we provide a rigorous benchmark.

Technical details. As it was described in Section 3.1, we train a multi-view detector using the groundHOG [7]. We use the PASCAL VOC 2007 dataset [24] as training set to learn four viewpoints for the category car: frontal, left, right and rear. The groundHOG was parameterized with different values of HOG window and HOG descriptor for each viewpoint. See Table 1 for all the details. Depending on the sequence, detections with a score below a threshold are discarded: `M-30` (0.3), `M-30-HD` (0.21) and `Urban1` (0.17).

4.2 Results

Detection results. First, we analyze the results of our detector in order to provide a baseline to establish further comparisons, specially with the tracking results. As it can

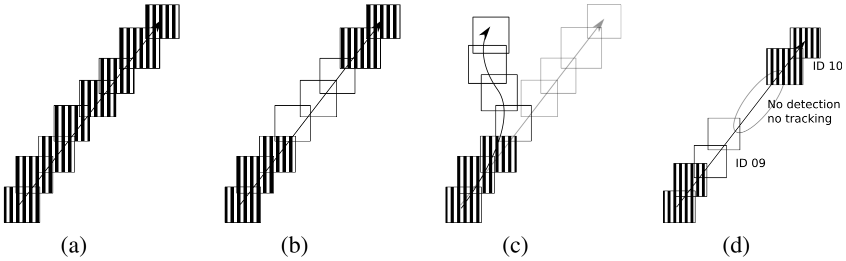


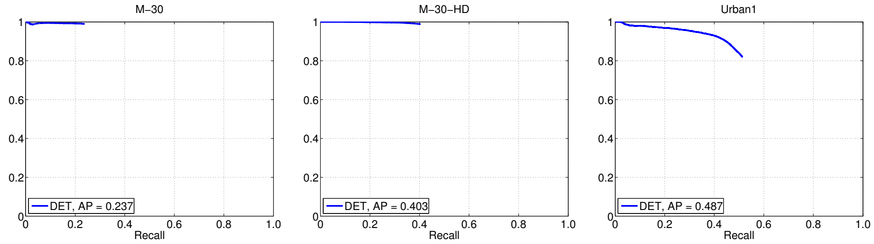
Fig. 5. Tracking Evaluation cases: a) Detector and Tracker are correct. b) Detector fails and Tracker is correct. c) Detector and Tracker fail (only one vehicle is counted). d) Detector and Tracker fail (more than one vehicle is counted)

	Frontal	Left	Rear	Right
HOG window (width, height)	(145,107)	(145,57)	(145,107)	(145,57)
HOG descriptor (width, height)	(16,12)	(16,6)	(16,12)	(16,6)

Table 1. groundHOG's settings

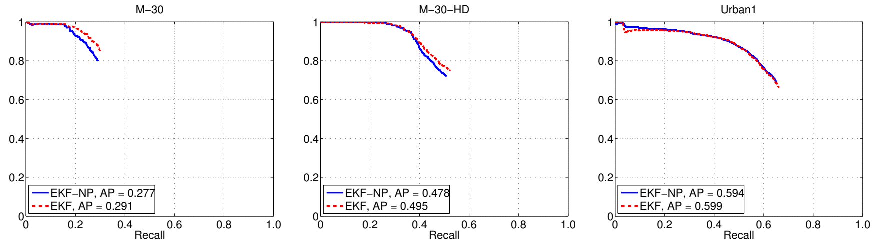
be seen in Figure 6, with the thresholds considered in our detectors, we have a very conservative approach where the number of false positives is under control. The recall is low, and the precision is high, *i.e.* our system only provides very confident detections. Note that within the context of smart cities and surveillance applications, it is important to control the number of false positives. The `Urban1` sequence is more challenging due to the variation in viewpoint and scale of the objects within it. Furthermore, it seems that our detector is able to deal with very different image resolutions and qualities. Surprisingly, the best average precision was obtained in the most difficult sequence: `Urban1`, with a value of 0.4872.

Tracking results. With the introduced benchmark, we can evaluate the tracking precision of the proposed EKF with pose information. As a baseline, we also report the results of a simple EKF, with the same dynamic model, but where the pose of the object is not observed through the detector. We call this second approach EKF-NP (no pose). Figure 7 shows the results obtained. In general, the EKF obtains higher APs than the EKF-NP for all the sequences, which confirms our hypothesis: to observe the orientation of the object during the detection step improves the tracking. This increment is very relevant for both `M-30` sequences. It is also relevant the increment of the recall with respect to the detection curves (see Table 2), which confirms the benefits of using the proposed tracking approaches. The robustness of the tracker relies on the video quality due to the data association step. As resolution and quality decay, histograms of close regions will be more similar and in that way, more incorrect matchings will be performed, increasing the false positive ratio. This explains the false positive obtained for the `Urban 1` sequence. Finally, in Table 3 we show that our approach is the best one counting the cars in the scene.



(a) (b) (c)

Fig. 6. Precision/Recall curves for object detection: a) `M-30`, b) `M-30-HD` c) `Urban1`.



(a) (b) (c)

Fig. 7. Precision/Recall curves for vehicle tracking: a) `M-30`, b) `M-30-HD` c) `Urban1`.

5 Conclusion

We have proposed a novel approach for tracking vehicles. Using the EKF, our architecture is able to simultaneously integrate into the motion model both the detections and the viewpoint estimations of the objects observed. Our experimental evaluation in a novel and challenging dataset confirms that this semantic information is beneficial for the tracking.

Acknowledgements This work was partially supported by projects TIN2010-20845-C03-01, TIN2010-20845-C03-03, IPT-2011-1366-390000 and IPT-2012-0808-370000. We wish to thank Fernando García and Laura Martín for their help with the annotation of the GRAM-RTM dataset.

References

1. Zhu, J., Yuan, L., Zheng, Y., Ewing, R.: Stereo visual tracking within structured environments for measuring vehicle speed. IEEE TCSVT **22** (2012) 1471–1484

	`M-30`	`M-30-HD`	`Urban1`
Detection	0.2384	0.4044	0.5132
EKF-NP	0.2916	0.5074	0.6518
EKF with Pose	**0.3009**	**0.5241**	**0.6616**

Table 2. Maximum recall.

	`M-30`	`M-30-HD`	`Urban1`
Annotated	256	235	237
EKF-NP	300	353	877
EKF with Pose	290	339	789

Table 3. Counted cars

2. Lee, J., Ryoo, M., Riley, M., Aggarwal, J.: Real-time illegal parking detection in outdoor environments using 1-d transformation. IEEE TCSVT **19** (2009) 1014–1024
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
4. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV **77**(1-3) (2008) 259–289
5. Chang, W.C., Cho, C.W.: Online boosting for vehicle detection. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **40** (2010) 892–902
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI **32** (2010) 1627–1645
7. Sudowe, P., Leibe, B.: Efficient use of geometric constraints for sliding-window object detection in video. In: International Conference on Computer Vision Systems. (2011)
8. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Van Gool, L.: Towards multiview object class detection. In: CVPR. Volume 2. (2006) 1589–1596
9. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: ICCV. (2007) 1 – 8
10. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3D object classes. In: CVPR. (2009)
11. Lopez-Sastre, R.J., Tuytelaars, T., Savarese, S.: Deformable part models revisited: A performance evaluation for object category pose estimation. In: ICCV 2011, 1st IEEE Workshop on Challenges and Opportunities in Robot Perception. (2011)
12. Pepik, B., Gehler, P.and Stark, M., Schiele, B.: 3D ˆ 2PM - 3D deformable part models. In: ECCV. (2012)
13. Gavrila, D.M., Munder, S.: Multi-cue pedestrian detection and tracking from a moving vehicle. IJCV **73**(1) (June 2007) 41–59
14. Leibe, B., Schindler, K., Cornelis, N., Van Gool, L.: Coupled object detection and tracking from static cameras and moving vehicles. PAMI **30**(10) (October 2008) 1683–1698
15. Ess, A., Schindler, K., Leibe, B., Van Gool, L.: Object detection and tracking for autonomous navigation in dynamic environments. Int. J. Rob. Res. **29** (2010) 1707–1725
16. Welch, G., Bishop, G.: An introduction to the kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill (2006)
17. Comaniciu, D. & Meer, P.: Mean shift analysis and applications. In: ICCV. (1999)
18. Bradsky, G.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, Q2 1998 (1998)
19. Koller, D., Danilidis, K., Nagel, H.H.: Model-based object tracking in monocular image sequences of road traffic scenes. IJCV **10**(3) (June 1993) 257–281
20. Dellaert, F., Thorpe, C.: Robust car tracking using kalman filtering and bayesian templates. In: Intelligent Transportation Systems. (1997)
21. Cameron, S. & Proberdt, P.: Advanced guided vehicles, aspects of the oxford agv project. World Scientific, Singapore (1994)
22. Bradsky, G., Kaehler, A.: Learning OpenCV. Computer Vision with the OpenCV Library. O'Reilly (2008)
23. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation - a set of best practices for high quality, economical video labeling. IJCV **101**(1) (2013) 184–204
24. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007)
25. Wang, Q., Chen, F., Xu, W., Yang, M.H.: An experimental comparison of online object tracking algorithms. In: SPIE. (2011)
26. Hoiem, D., Chodpathumwan, Y., Qieyun, D.: Diagnosing error in object detectors. In: ECCV. (2012)