Monocular Template-based Vehicle Tracking for Autonomous Convoy Driving

Carsten Fries and Hans-Joachim Wuensche

***Abstract*— This paper presents a vision-based solution for detection and tracking of convoy vehicles. Our approach is able to estimate the 3D vehicle pose, velocity and steering angle of the leader vehicles and needs template images of each vehicle. The improved template-based algorithm refers to a previously publication. First, we present an extension of our dynamic region growing algorithm, which is used to remove unnecessary image information. Thanks to a new preprocessing step, the segmentation algorithm is more stable while finding the vehicle silhouette. Second, we achieve an improved pose estimation by using rotated image features. Third, the extensive comparison of algorithms to train cascade classifiers leads to the best one for vehicle detection. The algorithm was evaluated while driving autonomously in urban and non-urban environments. Experimental validation shows that our approach can detect and track poorly visible vehicles under different weather conditions in real-time.**

Fig. 1: Vehicle tracking for autonomous convoy driving.

I. Introduction

Image-based advanced driver assistance systems have received considerable attention. They are one of the core technological innovations in the area of intelligent transportation systems. Many fields of application exist where driver assistance systems have to detect other vehicles. Well known applications are the collision avoidance system, adaptive light control and adaptive cruise control. All use one or more sensors like ultrasonic sound, RADAR, LiDAR or camera sensors to gain context information about the surrounding environment [1]–[3]. In recent years, the usage of RADAR and LiDAR sensors increased considerably despite the high acquisition costs. Both provide depth information with a detection range up to 200 m (automotive industry). They are more robust to light and weather changes and the angular resolution is lower compared to optical image sensors. Cameras have a higher resolution and the power consumption and acquisition costs are lower. A disadvantage is that the image quality strongly depends on light and weather conditions.

This paper focuses on a low-cost camera-based solution for following a leading vehicle (see fig. 1). Following a vehicle autonomously requires fast and robust estimation of the 3D vehicle position and velocity. Our previous publication [4] mentioned our template-based approach for the first time. The presented algorithm was able to roughly estimate 3D vehicles position and velocity in a fast way. In this publication, we will present an improved version in detail. Our dynamic region growing algorithm is more effective while finding the vehicle silhouette. Second, the pose estimation is enhanced by using rotated image features. Third, classifier training algorithms and their parameters are compared to determine the best one for vehicle detection. These extensions yield a vehicle tracking algorithm which is more robust against changes in weather conditions and achieves a more accurate estimation.

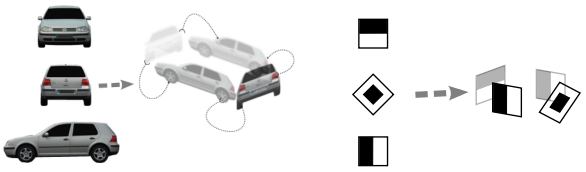
The outline of the paper is as follows: Section I motivates camera-only fields of application. The second section contains the related work about vehicle tracking methods. Our approach needs some preprocessing steps which are explained in section III. Section IV describes our procedure for vehicle detection and tracking. Experimental results collected while driving in urban and non-urban environments are presented in section V. Finally, conclusions and future work are given in the last section.

All authors are with department of Aerospace Engineering, Autonomous Systems Technology (TAS), University of the Bundeswehr Munich, Neubiberg, Germany. Contact author email: carsten.fries@unibw.de

II. Related Work

In recent years, the research to detect road participants with active LiDAR and RADAR sensors has grown continuously. Template-based vehicle detection approaches are used in point clouds or in $2\frac{1}{2}$ D occupancy grids. Some are geometry based, e.g. by fitting L-shapes or bounding boxes [5], [6]. These shape models are fixed, adaptive or flexible. Other publications process the raw point cloud data. E.g. [7] work on vehicle detection with 3D point cloud histograms.

This paper focuses on non-stationary passive sensors. On the one hand, stereo-based techniques are popular. Two cameras are used to estimate a depth map, objects are segmented, detected and tracked in generated 3D point clouds [8]. On the other hand, many publications work with only



(a) (b)

Fig. 2: Model generation: (a) Vehicle-specific. (b) Generic.

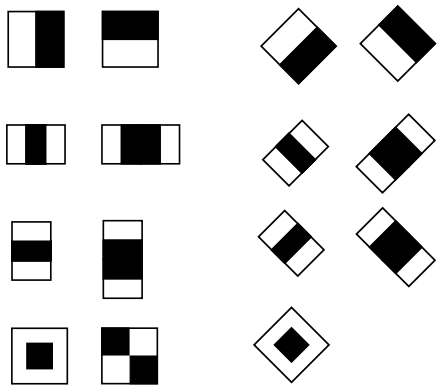
one camera that is mounted behind the vehicle windshield [9]–[11]. They perform single or multi-vehicle tracking. E.g. an autonomous convoy vehicle does single vehicle tracking to follow a specific leading vehicle. The literature shows that a precise information about each convoy vehicle is necessary to achieve a high precision estimation [4], [12]–[14]. In general, the precision for multi-vehicle tracking is not so important. Many of these applications do statistical analysis or provide collision information with the focus on multi-vehicle detection without using any kind of vehicle specific information. [15] do multi-vehicle detection on motorways with well-marked road lines, a uniform surface and low change of the road curvature. Their algorithm is a combination of WaldBoost classifier training with an unsupervised learning method called Tracking Learning Detection (TLD).

A common way is to train a classifier offline, e.g. [9] train a classifier with the fast computable Haar-like features [10]. Both detect vehicle rear sides from highway participants that have the same traveling direction. Moreover, [11] train Support Vector Machine (SVM) classifiers to detect separate parts of a vehicle. They use features called Histogram of Orientated Gradients (HOG) and are able to detect any vehicle side on intersections.

In the early days, when computing power only allowed for edge detection, the shadow cast on the road was used as a primary cue for a vehicle [16]. Statistically the shadow is darker than the road area which is not occupied. [17] propose a rough shadow detection with an adaptive template matching approach. Unfortunately, an important problem is that shadows from other objects like trees or the own shadow might affect the tracking. Instead of searching for a vehicle shadow, [18] focuses on finding a U-shape with a horizontal and vertical edge locator. Also the Hough transformation was applied to find the road boundary whereby the search area and further the computational cost could be reduced.

In addition, filtering is often applied. The multidimensional particle filter has become quite popular. It is used to verify up to thousands of hypotheses for the object of interest. The Kalman filter is another popular filter which is used to estimate state values over time [19].

Contrary to our work, many publications are primarily designed for motorways with clearly visible road markings, a uniform surface or small change in road curvature [20]. Our focus is on- and off-road driving, therefore, we do not require a ground plane assumption.



(a) (b)

Fig. 3: Haar feature set: (a) Upright. (b) Rotated.

III. Preprocessing: Building Template-models

This section mentions the offline preprocessing steps to apply the template-based tracking algorithm. A so-called 3D template model is necessary for each convoy vehicle. Any vehicle-specific model requires three template images for their generation. These reference images must be the rear, left and front view of the vehicle to enable image tracking from all viewing directions. The vehicle-specific model generation is visualized in figure 2(a) and is described in [4] in detail. Otherwise, a generalized 3D template model is also possible. Therefore, movable markers are usable (see fig. 2(b)).

The generation of a 3D template model includes the following three steps.

- Attaching 2D template images to one 3D model
- Searching and adding image features, e.g. red rear lights or license plate
- Training of cascade classifiers

Texture-based classifiers are necessary for each vehicle side. Therefore, the 3D template model, Haar-like [10], Local Binary Pattern (LBP) [21] and HOG [22] features, and a large collection of negative samples are used. We extend the feature set with rotated features by using 45°-rotated integral images [23]. Figure 3 visualizes the edge, line, diagonal and center-surround upright as well as rotated Haar features. White regions have a negative weight, black areas have a positive.

Negative samples are background images that do not contain the object of interest. In contrast, positive samples show the object of interest whose image location is also known. We generate the positive samples automatically, which is not common in the literature. This is only possible due to the 3D template model. An algorithm generates vehicle image poses randomly. In this process, the image location of all vehicle sides is known and defined by an upright rectangle $(u, v, \mathrm{width}, \mathrm{height})$. A brightness correction implies more robustness against illumination changes.

Using the negative and positive samples, three strong classifiers, the $\{$left+right$\}$, front and rear vehicle side, are trained for every feature type. The training algorithm tries

to find the best weights to separate these two classes, the negatives and positives. Some iterative algorithms exist to improve the classification accuracy stage by stage based on a sequence of weak classifiers. Our comparison between the vanilla Discrete Adaptive Boosting (DAB), Real Adaptive Boosting (RAB), Logit Boosting (LB) and the Gentle Adaptive Boosting (GAB) leads us to the best method to train a vehicle classifier [10], [24].

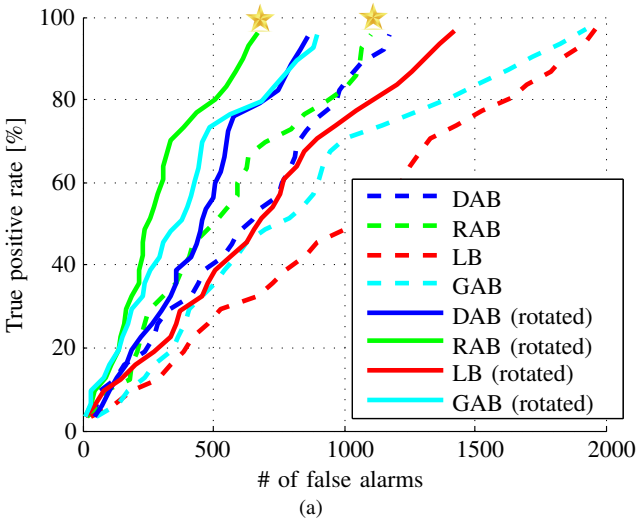
On the one hand, 5,000 negative images were used for evaluation. They show urban and non-urban environments with different weather conditions. On the other hand, 5,000 positive samples were automatically generated. The original image size is 752×480 pixels. The sample size is defined as $(\boldsymbol{s}_{\mathrm{w_{px}}}, \boldsymbol{s}_{\mathrm{h_{px}}}) = \omega \cdot (\boldsymbol{s}_{\mathrm{w_m}}, \boldsymbol{s}_{\mathrm{h_m}})$ pixels where $(\boldsymbol{s}_{\mathrm{w_{px}}} = 36, \boldsymbol{s}_{\mathrm{h_{px}}} = 33)$ and $(\boldsymbol{s}_{\mathrm{w_m}} = 1.8, \boldsymbol{s}_{\mathrm{h_m}} = 1.65)$ are the vehicle's width and the height in pixel and meter and ω represents a statistical evaluated scale factor of 20. We choose a high stage number of $N_{\max} = 10$ stages because of the huge amount of samples. At each stage a classifier is trained to reach the minimum hit rate of $h = 99\%$ and the maximum false alarm rate of $f = 40\%$. The overall hit rate is defined as $h^N \approx 90\%$. Accordingly, the overall false alarm rate per image window is calculated by $f^N \approx 0.0001\%$. Result of the analysis is shown in figure 4 and in the tables I-III as well. The tables show the overall training time (TT) of each training algorithm with upright features only and with rotated features included. Each training algorithm processes up to the maximum stage number $N_{\max}$ unless the algorithm terminates in a lower stage N_{stop}. This occurs in case that the algorithm achieves the stage-dependent and the overall minimum hit rate (HR) and maximum false alarm rate (FAR). HR and FAR are reported in percent with a range from 0.0 to 1.0. Vehicle detection works with a sliding window approach, where the features are scaled based on the original scale with a rescale factor of 1.1. A classifier detection can be correct (hit) or incorrect, which is called false alarm. We distinguish between these two states by considering the Root Mean Square Displacement (RMSD) between the upright rectangle from the classifier response and the ground truth quadrangle:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{4} (\boldsymbol{a}_i - \boldsymbol{b}_i)^2}{4}} \qquad (1)$$

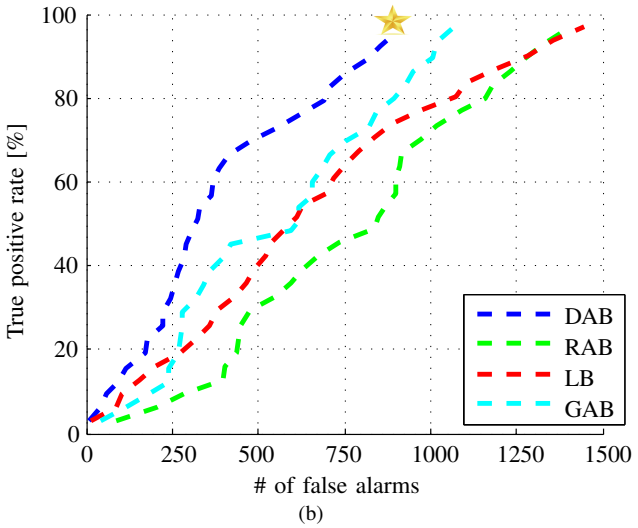
In equation 1, $\boldsymbol{a}_{1\ldots4} = \{(\boldsymbol{a}_{1_u}, \boldsymbol{a}_{1_v}), \ldots, (\boldsymbol{a}_{4_u}, \boldsymbol{a}_{4_v})\}$ represents four two dimensional quadrangle corners as well as $\boldsymbol{b}_{1\ldots4}$. A classifier response is graded as a false alarm, if the displacement overreached a specific threshold ($\mathrm{RMSD} > 10\%$).

Receiver Operating Characteristic (ROC) curves are one possibility to visualize the relationship between the positive detection rate and the number of false alarms. The figures 4(a) to 4(c) show ROC curves of the varying boosting strategies. Axis of abscissae represents the overall hit rate and axis of ordinates stands for the amount of false alarms.

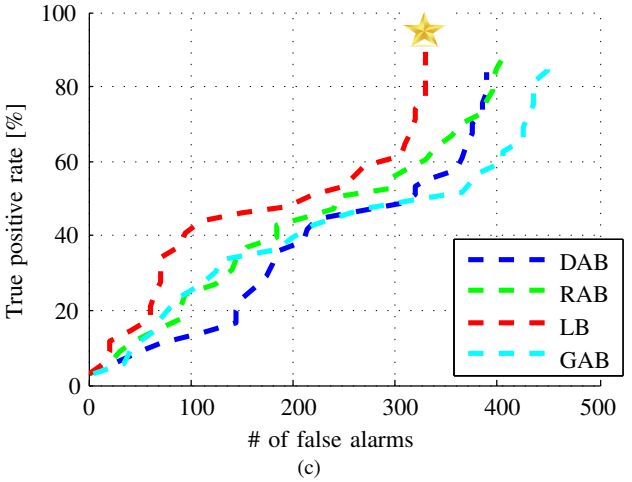
Algorithm analysis is done on a computer which has an Intel Core i7-2600K quad-core processor, 2 GB usable



(a)



(b)



(c)

Fig. 4: Receiver Operating Characteristics of 5,000 images. A star denotes the best training algorithm according to the feature setup: (a) Haar features without (dashed line) and incl. (solid line) rotated features (b) LBP features. (c) HOG features.

Boost algorithm	Incl. rotated feature set	TT[h]	N_{stop}	HR	FAR	P
DAB		5.32	8	0.96	0.24	0.86
	x	6.66	8	0.95	0.17	0.89
RAB		5.44	9	0.96	0.22	0.87
	x	8.62	9	0.97	0.14	0.92
LB		5.15	7	0.97	0.40	0.79
	x	7.24	8	0.97	0.29	0.84
GAB		5.53	8	0.97	0.39	0.79
	x	7.68	8	0.96	0.18	0.89

TABLE I: Analysis of HAAR-based classifier training algorithms. The overall training time (TT) is given in hours. N_{stop} gives information in which stage the algorithm terminates. Fastest and lowest algorithms to achieve the minimum HR and maximum FAR are colored green and red. HR and FAR represent the overall hit rate and false alarms rate. The classifier precision P is estimated with $\frac{\text{HR}+(1-\text{FAR})}{2}$.

Boost algorithm	TT[h]	N_{stop}	HR	FAR	P
DAB	4.79	7	0.96	0.18	0.89
RAB	3.77	8	0.96	0.28	0.84
LB	3.33	8	0.97	0.29	0.84
GAB	3.81	8	0.97	0.22	0.88

TABLE II: Analysis of LBP-based classifier training algorithms.

Boost algorithm	TT[h]	N_{stop}	HR	FAR	P
DAB	4.69	9	0.84	0.08	0.88
RAB	3.56	9	0.88	0.08	0.90
LB	1.80	9	0.89	0.07	0.91
GAB	0.98	9	0.85	0.09	0.88

TABLE III: Analysis of HOG-based classifier training algorithms.

memory and the C++-based program code is written for parallel processing.

The evaluation leads to the following best appropriate training algorithm depending on the application. If the overall training time (TT) is the primary focus, then

- DAB for a rotated Haar feature set
- LB for non-rotated Haar and LBP features
- GAB for HOG features

is advisable for classifier training. Mostly, the focus is not on the training time. Therefore, the classifier precision is more important because classifiers are trained in an offline step. A high hit rate and low false alarm rate define a good classifier precision $\text{P} = \frac{\text{HR}+(1-\text{FAR})}{2}$ whereas the appropriate training algorithms result in

- DAB for a LBP feature set
- RAB for rotated and non-rotated Haar features
- LB for HOG features

. The extended Haar feature set includes all rotated features and is recommendable due to the highest precision enhancement. This is significantly visible in figure 4(a) because of the lower amount of false alarms.

IV. Vehicle Detection and Tracking

This section describes an enhanced vehicle tracking algorithm compared to [4]. The algorithm contains the following three major steps to determine the vehicle information about a (local) convoy leader.

A. SUKF Vehicle Prediction

A Scaled Unscented Kalman Filter (SUKF) [25] is applied to predict the relative vehicle pose $\boldsymbol{P}_{\text{ego}\to\text{obj}} = \{r, \alpha, h, \psi, \theta, \phi\}$ in cylindrical coordinates (see table V), velocity v and steering angle λ. These eight state values $\boldsymbol{x}$ are predictable through a kinematic bicycle model and by consideration of the estimated egomotion [26] from time step $t-1$ to t. The SUKF measurement consists of eight 2D corners $\boldsymbol{y}_{1\ldots8}$ of the 3D vehicle bounding box. They are image coordinates $\{(u_1, v_1), \ldots, (u_8, v_8)\}$ and predicted by projection of the 3D vehicle pose $\boldsymbol{P}_{\text{ego}\to\text{obj}}$ into the image plane.

(a) (b)

Fig. 5: Dynamic region growing inside area of interest: (a) Publication [4]. (b) Improved result.

B. Performing Measurement for SUKF

The SUKF measurement step consist of three stages:

- Remove unnecessary image information
- Pre-detection of each vehicle side with classifiers $\rightarrow$ Upright rectangles
- Refine classifier detections through feature matching $\rightarrow$ Quadrangles $\{\boldsymbol{y}_{i,j,k,l}\} \in \boldsymbol{y}_{1\ldots8}$

The vehicle prediction $\boldsymbol{P}_{\text{ego}\to\text{obj}}$ is projected into the image plane to apply image processing in a smaller image area. Then, the dynamic region growing algorithm [4] tries to remove image areas (pixel blackening, see fig. 5) which do not belong to the tracked vehicle. We add an image denoising preprocessing step so that this image segmentation algorithm is more effective for blackening pixels outside the vehicle silhouette. Therefore, the comparison between the normalized box-, median-, Gaussian- and bilateral filter helps us to improve the region growing procedure by applying a smoothing preprocessing step. The evaluation result is shown in figure 5 and table IV. In proportion to the additional filter time, the median filter is the appropriate choice to enhance the region growing result. Furthermore, in comparison to [4], we reduce the pixel connectivity window from Moore neighborhood to Von Neumann neighborhood in the region growing algorithm. By using the Moore neighborhood, the surrounding eight pixels $(u \pm 1, v \pm 1)$ are compared to a pixel (u, v). The Von Neumann neighborhood considers only the four orthogonal surrounding pixels.

After the region growing algorithm, most of the non-blackened image area should belong to the vehicle, which we want to detect. Therefore, the trained classifiers are used for each vehicle side. A classifier responds with one or more upright rectangles. Each of them can be a false positive. For verification and to achieve a higher precision estimation (upright rectangles $\rightarrow$ quadrangles), we do

	Normalized Box	Gaussian	Median	Bilateral
Additional time [ms]	0.2	0.2	0.8	4.8
% more correct eliminated pixels	26	28	43	17

TABLE IV: Comparison between image filters to increase the elimination of unnecessary image pixels by applying dynamic region growing [4]. Kernel size = 5. Average area of interest size has $(w \times h) \approx (300 \times 160)$ image pixels.

feature matching between each upright rectangle and the 3D template model. For example, features are the red rear lights, front lights, license plate, vehicle silhouette, dark tires, windows or side doors (See [4]). Due to the high amount of eliminated image pixels, we only detect our primary vehicle target and not other vehicles. Result of the performing measurement step is an amount of quadrangles $\{\boldsymbol{y}_{i,j,k,l} = \{(u_i, v_i), (u_j, v_j), (u_k, v_k), (u_l, v_l)\}\} \in \boldsymbol{y}_{1\ldots8}$, where each of them describes four 2D boundary corners of a vehicle.

C. SUKF Sequential Innovation

Each new measurement $\boldsymbol{y}_{i,j,k,l}$ induces a sequential filter innovation which corrects the predicted state values $\boldsymbol{x}^\star$: Relative pose $\boldsymbol{P}_{\text{ego}\to\text{obj}}$, object velocity v and steering angle λ (Ackermann-steering model). The incoming measurement $\boldsymbol{y}_{i,j,k,l}$, its prediction $\boldsymbol{y}^\star_{i,j,k,l}$ and the Kalman gain matrix are used for this correction. A detailed description of the sequential Kalman filter innovation can be found in [27].

V. Experimental Results

This section presents some vehicle tracking visualizations and the tracking accuracy is compared to our prior publication [4].

The tracking system was evaluated while following autonomously in a convoy of two vehicles in total. The autonomous robot car was MuCAR-3 (Munich Cognitive Autonomous Robot Car 3$^{\text{rd}}$ Generation [28]) and the convoy leader was a VW Golf 4 and a VW Tiguan. MuCAR-3 has an on-board computer, an Intel Xeon L5640 Dual CPU Hexa Core. A single CMOS camera was used to get images in a $50\,\text{ms}$ cycle. The camera has a resolution of 752×480 pixels and a lens of $8\,\text{mm}$ focal length. It is mounted behind the windshield on a platform [29] that allows a maximum yaw ψ gaze of $\pm 45^\circ$ and a pitch rotation θ up to $\pm 12^\circ$. This allows extending the field of view to keep objects of interest in the image center. The processing time of our algorithm is $5 - 25\,\text{ms}$ for each image.

The test route took about 15 minutes to drive where the autonomous vehicle MuCAR-3 handled the throttle, brake and steering. We drove the same test route during different weather conditions, in rain, clouds and sunshine. Table V shows the average estimation precision of all three test runs. The real-time capability and high pose estimation accuracy allows a convoy velocity of up to $28\,\frac{\text{m}}{\text{s}}$.

	r[m]	α[°]	h[m]	ψ[°]	θ[°]	ϕ[°]	$v[\frac{\text{m}}{\text{s}}]$
max_{new}	36.1	32.7	1.34	62.84	7.44	5.68	28.09
ϵ_{new}	0.92	1.01	0.14	7.08	2.95	0.51	0.52
ϵ_{old}	1.07	1.12	0.16	7.24	2.96	0.73	0.56

TABLE V: Tracking accuracy compared to ground truth: Evaluated values are the relative position $\{r, \alpha, h\}$ and orientation $\{\psi, \theta, \phi\}$ in cylindrical coordinates and the object velocity v. max_{m} denotes the maximal state values and ϵ_{old} the RMSE of our template-based tracking from [4]. ϵ_{new} denotes the error of the template-based tracking from this paper.

(a) (b) (c) (d)

Fig. 6: Template-based Tracking: (a-c) with vehicle-specific model. (d) based on generic model.

We evaluated the tracking algorithms with ground truth data, generated by an IMU coupled to a RTK-DGPS in each of the vehicles involved. Our INS of type OxTS RT3003 reaches a precision of approximately $5\,\text{cm}$ for ground truth data. The INS estimates the object velocity with an accuracy of $0.2\,\frac{\text{m}}{\text{s}}$. The transformation from the ego to the object position results in the ground truth pose $\boldsymbol{P}_{\text{GT}_{\text{ego}\to\text{obj}}}$. Table V lists our enhanced tracking accuracy. Only the vision-based tracking algorithms were used for continuous estimation of the relative pose $\boldsymbol{P}_{\text{ego}\to\text{obj}}$. Communication between these vehicles was exclusively used for generating the ground truth data.

We compared the ground truth data with the template-based tracking and determined a false positive rate of zero. This can be traced back to the dynamic region growing method for unnecessary pixel elimination and the classifier detection cascaded with a distinctive feature detector.

The system shows its robustness while driving in different weather conditions. Also, it is invariant against partial occlusion (e.g. bushes, high grass or windshield wipers).

Furthermore, figure 6 visualizes the specific and generalized vehicle detection. The test route and some other visual impressions are accessible on Youtube: http://youtu.be/O79CrwhKJ3w

VI. Conclusions and Future Works

A. Conclusions

We presented a fast vision-based solution for detection and tracking of convoy vehicles. The algorithm is able to estimate the 3D vehicle pose, velocity and steering angle. Template images of each vehicle are the only requirement for our approach if a specific vehicle should be tracked. Otherwise, a generic template model is buildable with markers. The core of the tracking algorithm is the elimination of unnecessary image information combined with an analysis of cascaded classifiers in a top down procedure.

This paper introduced an extended dynamic region growing method for image segmentation. A new preprocessing (image smoothing) step results in a more stable segmentation while finding the vehicle silhouette. We also achieved an enhanced pose estimation by using rotated image features. The final novelty is the knowledge about the best appropriate classifier training procedure (boosting algorithm and parameters) for vehicle detection.

Algorithm evaluation is based on real-world data by driving autonomously in urban and unstructured environments. The results show that poorly visible vehicles are traceable in different weather conditions and in real-time.

B. Future Works

The usage of Pan-Tilt-Zoom (PTZ)-cameras reduces the amount of unnecessary image information. However, it is only applicable in case that the intrinsic camera calibration values of each zoom position are correctly defined. This will show an extensive analysis.

Acknowledgments

The authors gratefully acknowledge funding by the Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw).

References

[1] A. Petrovskaya and S. Thrun, "Model Based Vehicle Detection and Tracking for Autonomous Urban Driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[2] M. Buehler, K. Iagnemma, and S. Singh, Eds., *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, George Air Force Base*, ser. Springer Tracts in Advanced Robotics, vol. 56. Springer, 2009.

[3] A. Mueller, M. Manz, M. Himmelsbach, and H.-J. Wuensche, "A Model-Based Object Following System," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, China, 2009.

[4] C. Fries, T. Luettel, and H.-J. Wuensche, "Combining Model- and Template-based Vehicle Tracking for Autonomous Convoy Driving," in *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Australia, June 2013.

[5] D. Ferguson, M. Darms, C. Urmson, and S. Kolski, "Detection, prediction, and avoidance of dynamic obstacles in urban environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Eindhofen, 2008, pp. 1149–1154.

[6] B. Gao and B. Coifman, "Vehicle identification and gps error detection from a lidar equipped probe vehicle," in *Intelligent Transportation Systems Conference, ITSC/IEEE*, 2006, pp. 1537–1542.

[7] M. Himmelsbach, T. Luettel, and H. Wuensche, "Real-time object classification in 3d point clouds using point feature histograms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 994–1000.

[8] A. Barth and U. Franke, "Tracking Oncoming and Turning Vehicles at Intersections," in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, pp. 861–868.

[9] J. Choi, "Realtime On-Road Vehicle Detection with Optical Flows and Haar-like feature detector," University of Illinois at Urbana-Champaign," Computer Science Research and Tech Reports, 2006.

[10] P. Viola and M. Jones, "Robust Real-time Object Detection," in *International Journal of Computer Vision*, 2001.

[11] S. Sivaraman and M. M. Trivedi, "Real-Time Vehicle Detection Using Parts at Intersections," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2012.

[12] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3-D Model-Based Vehicle Tracking," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1561–1569, 2005.

[13] H. Schneiderman, M. Nashman, A. J. Wavering, and R. Lumia, "Vision-based robotic convoy driving," in *Machine Vision and Applications*, vol. 8, no. 6, 1995, pp. 359–364.

[14] M. Manz, T. Luettel, F. von Hundelshausen, and H.-J. Wuensche, "Monocular Model-Based 3D Vehicle Tracking for Autonomous Vehicles in Unstructured Environment," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[15] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2012, pp. 975–982.

[16] E. D. Dickmanns, *Dynamic Vision for Perception and Control of Motion*. Springer Verlag, 2007.

[17] M. Krips, A. Teuner, J. Velten, and A. Kummert, "Camera based vehicle detection and tracking using shadows and adaptive template matching," in *Proceedings of the 2nd WSEAS International Conference on Electronics, Control and Signal Processing (ICECS)*, 2003.

[18] M. Boumediene, A. Ouamri, and M. Keche, "Vehicle detection algorithm based on horizontal/vertical edges," in *7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)*, 2011, pp. 396–399.

[19] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME – Journal of Basic Engineering*, no. 82 (Series D), pp. 35–45, 1960.

[20] J. Arrospide, L. Salgado, and M. Nieto, "Vehicle detection and tracking using homography-based plane rectification and particle filtering," in *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2010, pp. 150–155.

[21] T. Ojala, M. Pietikainen, and D. Harwood, "Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 1 – Conference A: Computer Vision & Image Processing*, 1994, pp. 582–585.

[22] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[23] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing*, vol. 1, 2002, pp. 900–903.

[24] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *The Annals of Statistics*, vol. 38, no. 2, 2000.

[25] S. J. Julier, "The Scaled Unscented Transformation," in *Proceedings of the 2002 American Control Conference*, vol. 6, 2002, pp. 4555–4559.

[26] T. Luettel, M. Himmelsbach, F. von Hundelshausen, M. Manz, A. Mueller, and H.-J. Wuensche, "Autonomous Offroad Navigation Under Poor GPS Conditions," in *Proceedings of 3rd Workshop On Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[27] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[28] M. Himmelsbach, F. von Hundelshausen, T. Luettel, M. Manz, A. Mueller, S. Schneider, and H.-J. Wuensche, "Team MuCAR-3 at C-ELROB 2009," in *Proceedings of 1st Workshop on Field Robotics, Civilian European Land Robot Trial*, 2009.

[29] A. Unterholzner and H.-J. Wuensche, "Hybrid Adaptive Control of a Multi-Focal Vision System," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, USA, 2010, pp. 534–539.