Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Rear Vehicle Detection and Tracking for Lane Change Assist

Wei Liu, XueZhi Wen, Bobo Duan, Huai Yuan and Nan Wang, *Member, IEEE*

***Abstract*—A monocular vision based rear vehicle detection and tracking system is presented for Lane Change Assist (LCA), which does not need road boundary and lane information. Our algorithm extracts regions of interest (ROI) using the shadow underneath a vehicle, and accurately localizes vehicle regions in ROI by vehicle features such as symmetry, edge and shadow underneath vehicles. The algorithm realizes vehicle verification by combining knowledge-based and learning-based methods. During vehicle tracking, templates are dynamically created on-line, tracking window is adaptively adjusted with motion estimation, and confidence is determined for tracked vehicle. The algorithm was tested under various traffic scenes at different daytime, the result illustrated good performance.**

I. Introduction

In recent years, developing intelligent driver assistance systems(DAS) aiming to alert drivers about driving environments, and possible collision with other vehicles is becoming an area of active research. In these systems, reliable object detection and tracking are required. This requirement could be accomplished by one or multiple sensors such as vision and radar sensor, etc. It is a basic technology for DAS to detect and track vehicles using a vision sensor. Monocular vision based vehicle detection systems are particularly interesting for their low cost and for the high-fidelity information they give about the driver environment.

There is a great deal of previous research on vehicle detection and tracking using a vision sensor. According to the relative position of moving vehicles and ego- vehicle, it can be classified into front, rear and overtaking vehicle detection, separately used for ACC,LCA,BSD application, etc. T.Zielke et al.[1] developed a system named CARTRACK to detect and track front vehicles as well as measure the distance from a forward looking vehicle mounted camera. Du et al.[2] took advantage of the stable contour symmetry to detect and track front vehicles instead of the intensity symmetry which is noisy and prone to influences of lighting condition changes. Betke et al.[3] exploited a system for detecting and tracking front vehicles on a highway. The system used a combination of color, edge, and motion information to recognize and track the road boundaries, lane markings and other vehicles. Hoffmann et al.[4] used shadow as well as symmetry features for front vehicles detection and IMM method for tracking. Khammari et al.[5] presented a real-time vision-based vehicle's rear detection system, in which the detection algorithm consist of two main steps: gradient driven hypothesis generation and appearance based hypothesis verification. Adaboost classifier was used for verifying vehicle's hypothesis and an approach similar to SVT (Support Vector Tracking) was for tracking. Kim et al.[6] proposed a real-time detection and tracking system that can detect and track vehicles in various environments as well as under various illumination conditions**.** They used the shadow region to extract ROI and accurately localized vehicle regions in ROI by symmetry feature and then adopted template matching method to track vehicles. In [7], front and rear vehicle detection algorithm was developed. It took advantage of vehicles' horizontal features such as plate, rear window and bumper for detecting and tracking. In order to improve the real-time performance, they also exploited the appropriative image processor ViscontiTM. Sholin et al.[8] proposed a vehicle detecting and tracking system for highway scenes of both dry and wet weather conditions taken from a forward-looking vehicle mounted camera and used IMAP-VISION image processing board to realize real-time processing.

Most of the previous vehicle detection and tracking methods used lane or determined free driving space, but, if the lane does not exist or due to an intersection, etc., it is difficult to acquire such information, hereby, a robust algorithm must be developed to handle the problem.

In this paper, we present monocular vision based rear vehicle detection and tracking system for Lane Change Assist, it doesn't need road boundary and lane information. This paper will be organized as follows: In Section II, regions of interest (ROI) extraction and accurate vehicle location algorithms are proposed. In Section III, we present a vehicle verification algorithm that combines knowledge-based and learning-based methods. In Section IV, a vehicle tracking algorithm is described. Finally, we present the experimental results using the test videos that consist of various road conditions in Section V and conclusion is presented in Section VI.

Manuscript received January 12, 2007.This work was supported by the international science technology cooperation important item under grant No.2005DFA10260.

W. Liu is working in postdoctoral science research workstation at the department of computer science and technology from Northeastern University. His research interests include computer vision, pattern recognition and image processing .Moreover, he is also with Advanced Automotive Technology Research Center, Neusoft Park, Shenyang 110179, China (phone: 86-024-83661072; e-mail:lwei@neusoft.com).

X. Wen and B. Duan are with Software Center, Northeastern University, Shenyang 110179, China (e-mail: {wenxz; duanbb}@neusoft.com).

H. Yuan and N. Wang are with Advanced Automotive Technology Research Center, Neusoft Park, Shenyang 110179, China (e-mail: {yuanh ; wangnan}@neusoft.com).

II. VEHICLE CANDIDATE EXTRACTION AND LOCATION

A. Vehicle Candidate Extraction

The shadows underneath vehicles are distinct features for initial object hypotheses. Based on the fact that the area underneath a vehicle is distinctly darker than any other part on the road, we look in the image for vertical transitions from brighter gray values to darker ones (as scanning the image bottom-up). The edges with vertical transitions are defined as potential bottom edges of ROI for further analysis. The differences from [9] lie in two points: First, instead of computing the mean of road pixels, we consider pixels with negative vertical gradient values as local darker regions, as is shown in Fig.1.a. The mean and deviation of different regions in a road may be different. So the proposed method avoids the problem that the shadow underneath a vehicle can't be detected in casting shadows from trees or buildings with the uniform mean and standard deviation. Meanwhile, it's faster for unnecessary to compute the mean of the road pixels. Second, instead of computing vertical difference, we use change rate to extract the edges of shadows underneath vehicles, which is based on the idea that the change rate of gray value of adjacent pixels can be used to determine which pixels are the interesting edge points. Moreover, compared to difference method, ours is more effective.

After accomplish an AND operation of horizontal edges and dark regions masking, we use perspective constraints to filter out some horizontal edges not belonging to shadows underneath vehicles, then regard the lefts as bottom edges and generate ROI based on fixed aspect ratio, as is shown in Fig.1.b.

(a) (b)

Fig. 1. (a) Gradient image of shadow underneath vehicles (b) ROI

The above method is able to accomplish vehicle candidate extraction under different conditions, such as normal light, shadows casting on the road, vehicles in tunnel and in different illumination conditions, etc.

B. Vehicle's Bounding Box Location

We obtained the rough location of the candidate vehicle in the above step, but it needs further improvement. Because shadows underneath vehicles are not always located under the vehicles, it depends on daytime and corresponding sun position. To deal with this, we need to accurately locate the vehicles in ROI by other features.

Symmetry is the vehicle's significant characteristic, it has been often used for vehicle detection. Images of vehicles observed from rear or frontal views are in general symmetrical in the horizontal and vertical directions. Various kinds of symmetry are used in a lot of literature such as contour symmetry, grayscale symmetry, horizontal symmetry and vertical symmetry, etc. However, each of them has its own advantages and disadvantages. For example, the advantage of the binary contour symmetry is that the illumination variety has little influence on it, but the background, especially the symmetrical background, easily affects it, and when the vehicle is partly occluded, its binary contour symmetry will be destroyed. The advantage of the grayscale symmetry is that the various backgrounds hardly affect it, but it is prone to influences of illumination conditions. The HSV space is expressed by the hue (H), saturation (S) and value (V), which is suitable for the human vision characteristic. The S component in HSV space is related to the material property. Therefore, we use the three methods to calculate the vehicle's symmetry axes in ROI: binary contour symmetry, grayscale symmetry and the S component symmetry in HSV space. Then the final vehicle axis is synthesized by integrating the three symmetry axes.

We adopt the algorithm to detect the binary contour symmetry axis as described in [2]. The calculation of gray symmetry axis is the same as that of S component symmetry axis, it's shown as follows [12]:

1) Extract the shadow underneath a vehicle; calculate the bottom edge of the shadow Y_{bc} as well as the left and right terminal X_l , X_r ;
2) Compute the width of the shadow: $W = X_r - X_l$;
3) Measure the symmetry axis $S(j)$ with the following formula:

$$S(j) = \sum_{Y_{bc}}^{Y_{bc}+H} \sum_{\Delta x=1}^{W/2} \sum_{j=X_l-\Delta k}^{X_r+\Delta k} \left| P(j+\Delta x, i) - P(j-\Delta x, i) \right| \quad (1)$$

$$j_{sym} = \arg\min_{j} S(j) \quad (2)$$

Where $H = 0.9W$ and $P(x, y)$ is the gray value or the S-component value. j_{sym} is the symmetry axis we want to get.

The calculation of symmetry axes is shown in Fig. 2 and Fig.3.

Fig. 2. Computing the three kinds of symmetry axes

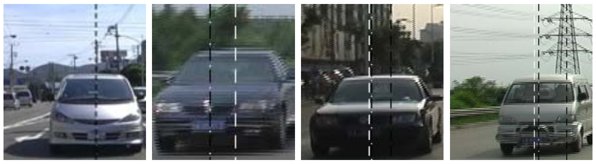


Fig. 3. Examples for the symmetry axes. White dashed represents gray symmetry axis; Black dashed represents binary value symmetry axis; Dash-dot represents S component symmetry axis

As is shown in Fig 2,the calculating results can be classified into three cases: 1) the symmetry axes obtained with the three above methods completely (or approximately) overlap, 2) two of them completely(or approximately) overlap, 3) all of them never overlap with each other(It hardly takes place). In order to overcome the influence of illumination conditions and backgrounds, we adopt the three methods to calculate the vehicle's symmetry axes and synthesize them so that the calculated axis is more accurate. The synthetic strategy is as follows:

1) Calculate the three symmetry axes x_1, x_2, x_3 ;
2) Calculate the synthetic symmetry axis x_{sym} .Three cases should be considered respectively:

Case 1:

If $\max\{x_i\} - \min\{x_j\} \le \Delta$, then

$x_{sym} = x_k$

Where $\min\{x_i\} \le x_k \le \max\{x_j\}$ $\quad i, j, k = 1,2,3$ $i \ne j$;

Case 2:

If $|x_i - x_j| \le \Delta$ and $\min\{|x_i - x_k|, |x_j - x_k|\} > \Delta$ then

$x_{sym} = \left\lceil \frac{x_i + x_j}{2} \right\rceil$, $i, j, k = 1,2,3$, $i \ne j \ne k$;

Case 3:

Otherwise, $x_{sym} = x_1$

Where Δ is a threshold to evaluate the similarity of the symmetry axes and its size is related to the width of ROI.

After obtain the symmetry axis, we calculate the vehicle's left, right and top edge as described in [2]. Its bottom edge is replaced with the edge of the shadow calculated.

III. Vehicle Verification

A. Knowledge-based Vehicle Verification

Our algorithm filtered out ROIs which contain no potential vehicles based on knowledge-based method. The following clues are used for vehicle verification [13]: texture, symmetry, color, shadow underneath vehicles, horizontal edges and vertical edges, etc. Sun et al. [14] discuss the advantages and disadvantages of these clues as well as how to use them for vehicle verification. S. Kyo [15] pointed out that the clues should be used differently depending on the spatial relationship between other vehicles and the ego-vehicle.

We use the clues such as the shadow underneath vehicles, symmetry, and color for vehicle verification. The advantage of shadow is that potential vehicles can be detected and located using it. Meanwhile, we can also use it for vehicle verification since the located potential vehicle should have a shadow proper to its real width. If the shadow is too wide or narrow, then the corresponding potential vehicle is rejected. Yet because bridges and other buildings also have shadows, we also consider symmetry and color to eliminate false verification. The ROIs lack of symmetry will be filtered out. Color information is also a very useful cue for vehicle verification, in particular, the vehicle color will distinctly change from top to bottom in the vertical direction.

In practical application, although we can get rid of about two thirds ROIs in which no vehicles exist using above clues, some backgrounds may also have such features for the complexity of environments. So we use machine learning method to validate the left ROIs.

B. Learning-based Vehicle Verification

Verifying a hypothesis is a two-class pattern classification problem: vehicle versus non-vehicle. At present, SVM and Adaboost classifier are most common methods to solve the problem and show satisfactory results. The advantage of these methods is the automatically generating classified Criteria without the need of human interference. So we use the Haar wavelet transform for feature extraction and SVM for classification.

The images used for off-line training were taken on different daytime, as well as on different scenes: highways, urban roads and so on. The training samples are first automatically generated with the extraction and location algorithms described in previous sections, and then are manually selected.

The training dataset consisted of 17,454 instances roughly split equally between vehicle and non-vehicle samples. After training, we will obtain a classifier to recognize the left ROIs filtered by knowledge-based method.

IV. Vehicle Tracking

In order to accurately track each detected vehicle, we start up a tracker for each vehicle and initialize the parameters of the tracker according to the sizes and positions of detected vehicles. Each tracker tracks only one object. Trackers create tracking windows and adaptively adjust the size and position of the tracking window based on motion estimation. Feature-based and template-based methods are adopted for vehicle tracking. The former are used to track normal vehicles, the latter are used to track vehicles at the image boundary or occluded.

A. On-line Templates Creation

We create on-line gray template for vehicle tracking. For the vehicles at the image boundary or occluded, they only have partial feature information, so using features to track will lead to failure. As a result, we use syncopate template, that is, use templates of the visual part of the vehicle for tracking. For normal vehicles, templates are also used to evaluate and revise the results of feature tracking. In order to accelerate the matching speed, pyramid accelerating algorithm is used. The number of pyramid layers is different according to the size of tracked vehicle images. We adopt SSDA algorithm to reduce the complexity of matching during searching process in each layer.

The template is updated as follows:

--For vehicles at images boundary or occluded, the templates are updated at every frame;

--For normal vehicles, the templates are updated conditionally.

As for normal vehicles, if the online template is updated at every frame, the drift problem may occur when the tracking region gradually moves out of the correct region until a total tracking failure. By computing the tracking results' similarity of current and previous frame, we determine whether or not to update the template. The width, height, and aspect ration of the object are used to measure similarity. The template is updated if

$$S = \sum_i \omega_i E(f_i) > \theta \tag{3}$$

Where θ is a fixed threshold, ω_i is the ith feature weight, $E(f_i)$ is the similarity evaluation function of the ith feature f_i at t and $t-1$ frame.

$$E(f_i) = \frac{\min\{f_i(t), f_i(t-1)\}}{\max\{f_i(t), f_i(t-1)\}} \tag{4}$$

B. Adaptive Tracking Window Adjustment

For tracked vehicles, an adaptive tracking window adjustment is necessary because it can reduce the searching regions of vehicles at the next frame as well as decrease the influence of backgrounds during tracking process. We divide the objects into new and old ones according to their tracked times. Because the two kinds of objects have different history information, we adopt different modes to adjust tracking windows. For the new objects lack of history information of tracking, we adjust the tracking windows by appropriately enlarging them. For the old ones with history information, by computing the center displacement of the bottom edge, we confirm the motion direction and speed to adjust the tracking window at the next frame. The adjusting mode is as follows:

$$W_t(t+1) = W_t(t) - f(h) \times h + \mathrm{sgn}(v_y) \times v_y \tag{5}$$

$$W_b(t+1) = W_b(t) + f(h) \times h + \mathrm{sgn}(v_y) \times v_y \tag{6}$$

$$W_l(t+1) = W_l(t) - f(w) \times w + \mathrm{sgn}(v_x) \times v_x \tag{7}$$

$$W_r(t+1) = W_r(t) + f(w) \times w + \mathrm{sgn}(v_x) \times v_x \tag{8}$$

Where, $W_t(t)$, $W_b(t)$, $W_l(t)$ and $W_r(t)$ determine the estimated tracking window at t frame, h and w are the height and width of the tracked vehicle at t frame; $f(h)$, $f(w)$ are the subsection functions of height and width respectively, and the range is [0.1,0.2]; v_x, v_y are the speed at x-direction and y-direction in the image coordinate respectively. (The unit is pixel); $\mathrm{sgn}(v_x)$, $\mathrm{sgn}(v_y)$ are sign functions, $\mathrm{sgn}(v)$ is defined as follows:

$$\mathrm{sgn}(v) = \begin{cases} -1 & v < -thresh \\ 0 & |v| \le thresh \\ 1 & v > thresh \end{cases} \tag{9}$$

Since the algorithm can adaptively adjust the size and position of the tracking window, it is suitable to such situations as maneuver, swerve, overtaking and reverse of tracked vehicles.

C. Confidence Tracking

Each detected vehicle has its own ID and confidence value during tracking process. The confidence value depends on the features of its aspect ratio, size, horizontal edge and shadow underneath the vehicle, etc. The confidence value is set like below:

If the tracked object is continuously detected and tracked with the same ID, $V_c = V_c + 1$;

If the tracked object is not detected at the current frame,

$$V_c = \begin{cases} V_c - V_p & if\ V_{c\max} \le N_f \\ V_c - round(V_{c\max}/N_f) & else \end{cases} \tag{10}$$

Where, $V_{c\max}$ represents the largest confidence value of the tracking object up to the current frame. N_f represents the frames number of a detection and tracking cycle. V_p is a fixed penalty threshold.

Once $V_c < 0$, object tracking is terminated. The advantage of setting confidence value is to avoid mistakenly terminating tracking when one or several frame detection is failing due to sudden changes of circumstances.

V. Experimental Results

In order to evaluate the performance of our system, different videos were taken from a backward-looking camera mounted on a vehicle, as well as on different scenes, including highway, urban road, sparse road and so on at different times of day. Sometimes roads are covered with japanning, smear, snow and so on. The performance of our system is shown in table 1.

TABLE I
PERFORMANCE OF SYSTEM DETECTION AND TRACKING

Video	*NRR*	*DR*	*FAR*	*RA1*	*RA2*	*TC*
Urban road	1193	99.5%	3.98%	92.5%	90.8%	97.9%
Narrow road	1125	96.4%	0.48%	94.8%	86.4%	93.6%
High way	2294	100%	0%	93.7%	90.3%	100%
Mountain road	1113	95.5%	3.08%	89.1%	88.1%	80.5%

The meaning of each metrics is as follows:

NRR (Number of Reference Regions) represents the reference number of vehicles appearing in the evaluation set. DR denotes the detection rate. FAR denotes the false alarm rate. $RA1$ and $RA2$ are used to evaluate the accuracy of vehicle location.

$$RA1 = \sum \frac{Area(TP)}{Area(RR)}, RA2 = \sum \frac{Area(TP)}{Area(DRR)}$$

Where RR (Reference Region) represents ideal detection regions of vehicles; DRR (Detection Result Region)represents the detection regions obtained by the algorithm. TP denotes the intersection of the above two. As is shown in Figure 4, the larger $RA1$ and $RA2$ are, the more precise the location is. TC (Tracking Continuity) is used to describe the continuity of tracking. It is the ratio of the largest number of continuous tracking frame with the same ID and that of all frames with vehicle appearing. The larger TC, the more stable the tracking. Detecting and tracking results under different road conditions are shown in Fig.5.

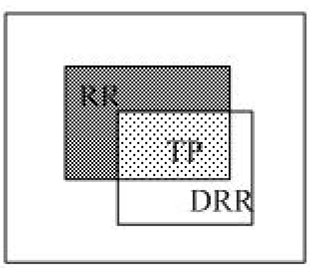


Fig. 4. Sketch map of computing RA1 and RA2

We exploit the automatic evaluation analyzer (Ev Analyer) to reduce workload of evaluation. The analyzer automatically extracts evaluation frames from videos, then RR of the vehicles that should be detected in the images of evaluation frames are manually marked using rectangle box and then the analyzer automatically generates data files recording sizes and positions of RR and compares them with the results obtained by the algorithm. Finally it gives the evaluation results precisely and rapidly.

Currently we are able to achieve a frame rate of approximately 25 frames per second using standard PC machine (Pentium IV 2.8GHz and 512M of RAM) without performing specific software optimizations.

VI. Conclusion And Future Work

In this paper, we propose a monocular vision based rear vehicle detection and tracking system. Experimental results show that high detection rate and low false alarm rate can be obtained under different conditions, even in complicated urban environment. The system can evaluate rear vehicle distance, velocity and TTC for lane change assist.

For future works, we will continue to optimize the system and improve the real-time performance of it. At the same time, we are developing a fusion framework to combine radar and vision in order to enhance the robustness of our system.

References

[1] T. Zielke, M. Brauckmann, and W. von Seelen,"CARTRACK: Computer Vision-Based Car-Following," *Proc. of the First IEEE' Workshop on Applications of Computer Vision*, pp. 156-163, alm Springs, CA, 1992..

[2] Y. Du and N. Papanikolopoulos,"Real-Time Vehicle Following through a Novel Symmetry-Based Approach," *IEEE International Conference on Robotics & Automation. Albuquerque*, New Mexico-April.pp.3160-3165, 1997.

[3] M. Betke, E. Haritaoglu and Larry S. Davis,"Real-time Multiple Vehicle Detection and Tracking from a Moving Vehicle," *Machine Vision and Applications*, vol. 12, no. 2, pp. 69-83, 2000.

[4] C. Hoffmann, T. Dang and C. Stiller, "Vehicle Detection Fusing 2D Visual Features," *IEEE Intelligent Vehicles Symposium*, University of Parma ,Parma, Italy, June 14-17, pp. 280-285, 2004.

[5] A. Khammari, F. Nashashibi, Y. Abramson and C. Laurgeau, "Vehicle Detection Combining Gradient Analysis and Adaboost Classification," *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, September 13-16, pp. 1084-1089, 2005.

[6] S.Y. Kim, S.Y. Oh, J.K. Kang and Y.W. Ryu,"Front and Rear Vehicle Detection and Tracking in the Day and Night Times Using Vision and Sonar Sensor Fusion," *International Conference on Intelligent Robots and Systems*, 02-06 Aug. pp. 2306-2311, 2005.

[7] K.Sasaki and S.Kawamoto,"Visconti Image Processing System Specific to Automotive Domain". *Toshiba Review*, 59(4):28-31, 2004.

[8] S. Kyo, T. Koga, et al, "A Robust Vehicle Detecting and Tracking System for Wet Weather Conditions Using the IMAP-VISION Image Processing Board," *IEEE International Conference on Intelligent Transportation Systems*, pp. 423-428, 1999.

[9] C. Tzomakas and W. Seelen, "Vehicle Detection in Traffic using shadows ," Institution of Neuroinformatik, Ruhr University Bochum in Germany, 1998.

[10] S.S. Huang, C.-Jen Chen, P.Y. Hsiao and L.C. Fu, "On-board Vision System for Lane Recognition and Front-vehicle Detection to Enhance Driver's Awareness," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation* New Orieans, LA-April 2004.

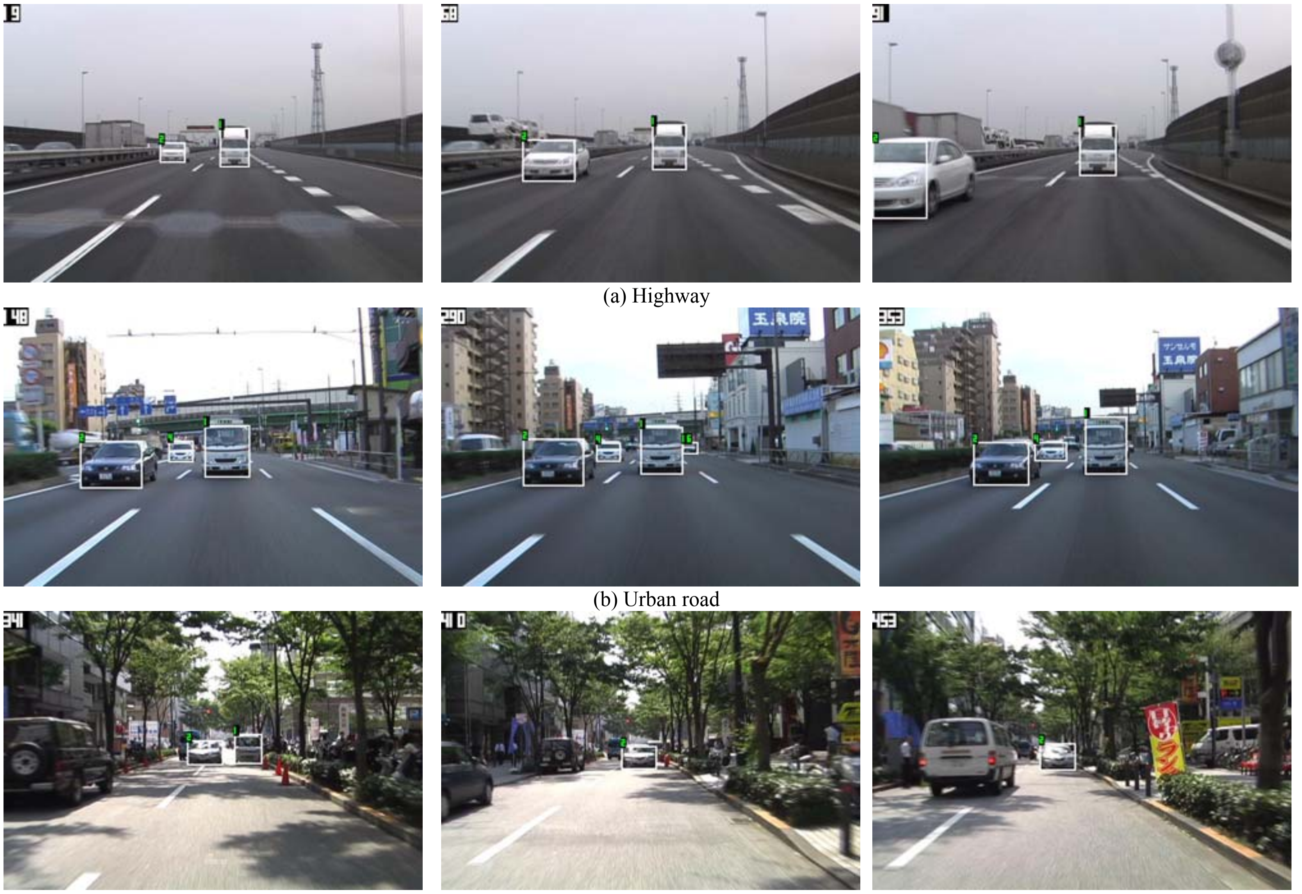
[11] Andreas K, "Symmetry-based recognition of vehicle rears," *pattern recognition letters*,1991,12(4):249-258.

[12] H. Chang, C. Fu, C. Huang, "Real-time vision-based preceding vehicle tracking and recognition," *2005 IEEE Intelligent Vehicles Symposium*,.6-8 June Page(s):514 – 519.

[13] M.B.Van Leeuwen,"Vehicle detection with a mobile camera," Technical report. Computer Science Institute, University of Amsterdam, The Netherlands, October 2001.

[14] Z. Sun, G. Bebis and R. Miller, "On-road vehicle detection using optical sensors a review," *IEEE International Conference on Intelligent Transportation Systems*, Washington, DC, October 3-6, 585-590. 2004.

[15] U. Handmann, T. Kalinkea, C. Tzomakas, M. Werner, and W. Seelen, "Computer Vision for Driver Assistance Systems," In Proceedings of SPIE Vol. 3364, Orlando, 1998. SPIE. Session Enhanced and Synthetic Vision 1998.



(a) Highway

(b) Urban road

(c) Urban narrow road

Fig. 5. Results of different image sequences under typical scenes. The white rectangles mark responding vehicle regions within which vehicles are detected and tracked. The vehicle IDs are shown in the topper left corner of these rectangles. (a) Highway scene, (b) Urban road scene, (c) Urban narrow road scene