International Journal of Control, Automation, and Systems (2013) 11(3):577-585
DOI 10.1007/s12555-012-0353-1
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

A Novel Particle Filter Implementation for a Multiple-Vehicle Detection and Tracking System using Tail Light Segmentation

Ming Qing and Kang-Hyun Jo*

Abstract: This paper proposes a vision-based multiple vehicle automatic detection and tracking system which can be applied in different environments. To detect vehicles, tail light position is utilized for fast vehicle candidate localization. A back propagation neural network (BPNN) trained by a Gabor feature set is used. BPNN verifies vehicle candidates and ensures detection system robustness. In the vehicle tracking step, to overcome multiple vehicle tracking challenges, partial vehicle occlusion and temporarily missing vehicle problems, this paper propose a novel method implementing a particle filter. The color probability distribution function (CPDF) of detected vehicles is used twice in the vehicle tracking sub-system. Firstly, CPDF is adopted to seek potential target vehicle locations; secondly, CPDF is used to measure the similarity of each particle for target vehicle position estimation. Because of various illuminations or target vehicle distances, the same vehicle will generate different CPDFs; the initial CPDF cannot guarantee long-term different scale vehicle tracking. To overcome these problems, an accurate tracking result, which is chosen by a trained BPNN, is used to update target vehicle CPDF. In our experiments, the proposed algorithm showed 84% accuracy in vehicle detection. Videos collected from highways, urban roads and campuses are tested in our system. The system performance makes it appropriate for real applications.

Keywords: Multiple vehicle detection, particle filter, temporarily missing, vehicle particle occlusion.

1. INTRODUCTION

More vehicles on the road result in more fatal vehicle crashes [1]. To improve safety, in recent years, many new technologies have been developed to mitigate or avoid vehicle accidents by sensing danger using intelligent transportation system (ITS) and driver assistance systems (DAS) [2,3].

Environmental information, such as surrounding vehicles, road lane information, nearby pedestrian location, etc. is very important when driving. Vision-based driver assistance systems have become popular, due to the abundant environmental information provided by vision sensors. This study focuses on multiple vehicle detection and tracking system development. Car video is collected by a forward-looking CCD camera which is mounted on the head of a host vehicle. The system automatically detects nearby vehicles then analyzes their position and the trajectory of the detected vehicle through the tracking process.

Manuscript received August 10, 2012; revised December 29, 2012; accepted February 24, 2013. Recommended by Editorial Board member Dong-Joong Kang under the direction of Editor Hyouk Ryeol Choi.

This work has been financially supported by the research grant of University of Ulsan since 2010.

Ming Qing and Kang-Hyun Jo are with the School of Electrical Engineering, University of Ulsan, Daehak Rd. 93, Namgu, Ulsan 680749, Korea (e-mails: qingmingwisher@gmail.com, acejo@ulsan.ac.kr).

* Corresponding author.

Research on vision-based multiple vehicle detection and tracking algorithms has been proposed previously. In a vehicle detection sub-system, numerous features can be selected. In the front vehicle detection case, vehicle rear symmetry appearance is often used. Reference [4] uses an intensity-based symmetry method to determine vehicle positions. Broggi *et al.* [5] generated a symmetry map by combining horizontal-edge and gray-level symmetry information. However, a symmetrical feature-based vehicle detection algorithm requires the target vehicle to be strictly located right in front of the host vehicle. When the target vehicle has been rotated by some angle, such as in a vehicle cornering situation, the rear appearance of the vehicle is not strictly symmetrical. The shadow between the vehicle and the road is another feature which is widely used in vehicle detection research [6]. Shadow information is very sensitive to illumination. Shadow position is unstable because of changing illumination directions. Sun *et al.* [7] generated multiple hypotheses based on vehicle edge information. Goerick *et al.* [8] used a local orientation coding (LOC) technique to build edge histograms in a vehicle detection system. Vehicle edge information shows the vehicle profile, which is a peculiar feature of the vehicle. However, in a complex environment, for example, multiple vehicles on an urban road, it is quite difficult to accurately generate an edge histogram.

In a multiple vehicle tracking sub-system, the vehicle body could be partially covered by another vehicle. The entire target vehicle body could even be temporarily occluded and appear a few seconds later. These two issues are called vehicle partial occlusion and the temporarily missing problem, respectively.

There are numerous algorithms that have been developed to perform vehicle tracking. The mean shift algorithm is a nonparametric statistical method for seeking the nearest mode of a point sample distribution which has been proven to be effective for object tracking [9]. The kernel window bandwidth is essential for the mean shift tracking algorithm. It determined the sampling quantity of the mean shift iteration. Furthermore, it is related to the tracking window size. In [10], a kernel bandwidth increment for scale adjustment is proposed, and this approach process uses one frame with three different bandwidth kernels, which will increase system runtime. References [11,21] use color model under dynamic illumination change and its tail light pairs which depend on tracking window adjustments. Color histogram-based mean shift vehicle tracking is effective in real applications; however, the mean shift algorithm cannot overcome the vehicle temporarily missing problem. The mean shift vector starts from an initial position which is the target vehicle position in the previous frame; however, in the vehicle temporarily missing case, the entire occluded vehicle is unpredictable in the image plane. Particle filter is another famous technique used in vision-based object tracking research areas [12,13]. A particle filter is a filtering method based on a sequential Monte Carlo method; it can adequately model multi-modal and non-Gaussian distributions. Chan *et al.* [14,15] and Lee *et al.* [20] used a data-driven initial sampling method to initialize particles. These particles in the particle filter are all possible candidates for a moving target such as vehicles. The authors combine vertical edge cue, underlying shadow cue, tail light cue and symmetry cue as observations for hypothesis verification. Then, the likelihood of each candidate is updated through a Haar-like feature [16] classifier. However, the vehicle temporarily missing problem is not mentioned in their system. This paper proposes a novel method to implement a particle filter which can overcome vehicle partial occlusion and the temporarily missing problem. Related work on this system have been previously published [11,17].

The rest of this paper is organized as follows. Section 2 shows the overview of the system. In Section 3 and Section 4, a vehicle detection sub-system and vehicle tracking sub-system will be introduced. Experiments are shown in Section 5. The conclusion and future work are discussed in Section 6.

2. SYSTEM OVERVIEW

This paper proposes a system for detection and tracking of multiple vehicles. Fig. 1 shows the rough architecture of the entire system. Car video was used to generate the image sequence. The vehicle detection unit process in each image frame detects target vehicles, then the detection results are sent to the vehicle tracking unit. Each detected target vehicle is tracked by particle filters in parallel. The tracking unit analyzes vehicle trajectory and stores the information in internal storage. Then, the system sends the final tracking result to the vehicle detection unit to update the target color information. A detailed description of the vehicle detection unit and tracking system will be presented in the next sections.

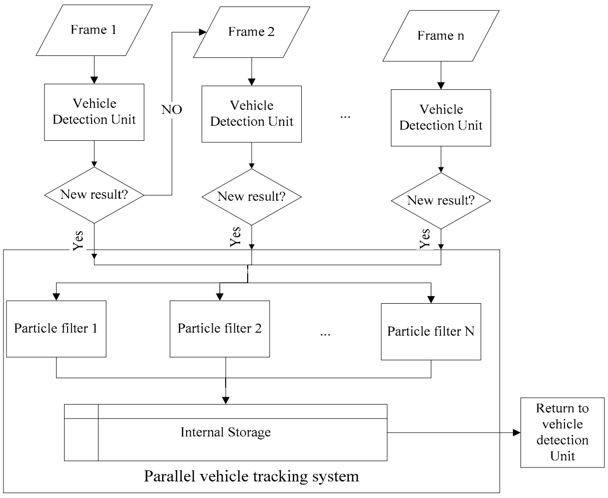


Fig. 1. System architecture.

3. VEHICLE DETECTION

In vision-based systems, vehicle detection generally involves two steps: vehicle candidate generation and vehicle candidate verification. Tail light information is employed in this study to generate vehicle candidates. In the vehicle candidate verification step, a feature set created by Gabor filters using eight directions and five scales is used to train a back propagation neural network (BPNN). The main architecture of the vehicle detection unit is described in Fig. 2.

3.1. HSV color model-based light segmentation

HSV stands for hue, saturation and value and is also often called HSB (B stands for brightness); these are often used by humans for color object description. Based on the HSV color model, a red light can be easily detected with an appropriate threshold. The threshold was determined using 68 vehicle tail light images along with the HSV statistical value. After that, the morphological method was employed to remove small points produced

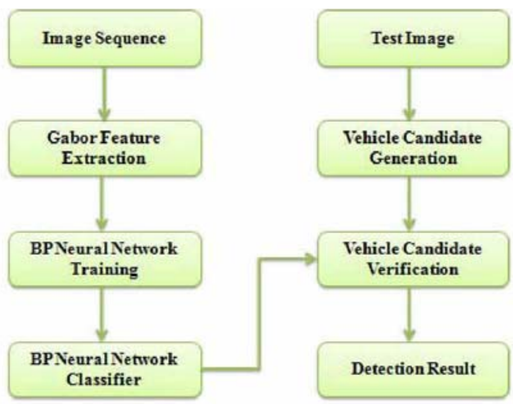
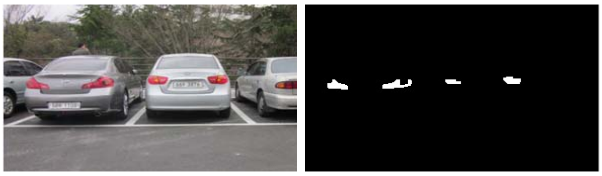


Fig. 2. Vehicle detection architecture.

(a) Single vehicle.



(b) Multiple vehicles.

Fig. 3. Vehicle rear light segmentation.

by color segmentation. Fig. 3 shows vehicle tail light segmentation results for single (a) and multiple vehicles (b).

3.2. Light pair candidate generation

The main goal of this step is to identify the corresponding light pairs. Vehicle tail light pair candidates can be extracted by the following conditions. As shown in Fig. 4, the left and right tail lights are denoted as c_1 and c_2, respectively.

1) The distance between c_1 and c_2 should be limited to a range.

$$w_{\min} \le w_{c_1c_2} \le w_{\max}, \tag{1}$$

where w_{c1c2} represents the width between the two vehicle lights in a pair. In this paper, w_{min} and w_{max} equal 5 pixel lengths and 100 pixel lengths, respectively. The threshold selection depends on the image size.

2) The position of the corresponding light pair approximates the same horizontal line.

$$\left|h_{c_1} - h_{c_2}\right| \le d, \tag{2}$$

where h_{c1} and h_{c2} represent the height of c_1 and c_2 respectively, and d is a constant which depends on the image resolution.

Due to the parameters which are mentioned in the above step, this algorithm is able to extract vehicle candidates based on the tail light distribution. However, in some cases where cars are next to each other, as shown in Fig. 2(b), this process will extract non-paired tail lights. As shown in Fig. 5, each rectangle represents one light pair. Because the same types of vehicle will have the same tail light distances, vehicle candidates can be extracted based on locations of pairs of lights and distance information, as shown in Fig. 5.

Fig. 4. Vehicle tail light pair.

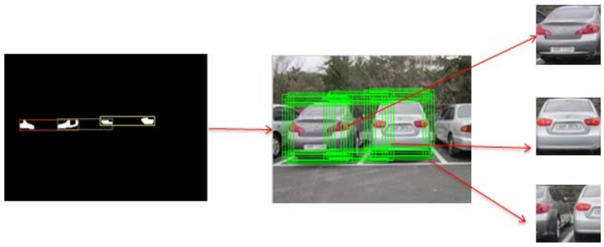


Fig. 5. Vehicle candidate generation.

3.3. Gabor feature extraction and training process

To obtain vehicle features for the BPNN training process, the Gabor feature is investigated. A Gabor feature can effectively describe local features of images with different directions and scales. The Gabor transform [18] can be formulated by the following equation:

$$G(x,y,\omega,\sigma) = \frac{1}{2\pi\sigma_x\sigma_y}\exp^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)}\exp^{-j\omega(x+y)}. \tag{3}$$

This paper uses eight orientations and five scales for Gabor filters to construct the database.

In the vehicle verification step, a back propagation neural network is used. BPNN is applied extensively in practical application due to its superiority in model recognition [19]. It can identify actual vehicles from a pool of vehicle candidates, and non-vehicle candidates are omitted. In this case, the robustness of vehicle detection is guaranteed by a strong BPNN classifier.

4. MULTIPLE VEHICLE TRACKING

A multiple vehicle tracking system is required to achieve the following two goals:

1) Multiple vehicles automatically tracked with tracking windows that scale dynamically and adjust as the target vehicle distance changes.

2) System should track partial occlusion and temporarily missing vehicles.

This paper proposes a novel implementation method of a particle filter. Color histogram detected vehicles

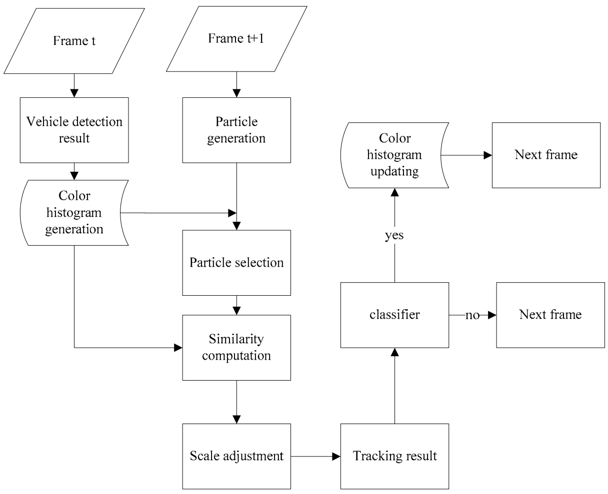


Fig. 6. Vehicle tracking system flow chart.

color probability distribution function is used twice in the vehicle tracking sub-system to weigh particles. The distance between corresponding light pairs is investigated for tracking window scale adjustment. The BPNN classifier is also used for updating the color histogram. Multiple vehicles are tracked by particle filters in parallel. Fig. 6 shows the flow chart of the vehicle tracking system.

4.1. Color feature space representation

In the current frame, the detected vehicle can be represented based on the HSV color model as in Fig. 7 (b). We assume the target color model can be split into several uniform histogram bins; to enhance the importance of the center position of the target vehicle, an Epanechnikov kernel function is utilized. Consequently, the target vehicle can be represented in HSV color space by the color probability density function q, which is calculated by the following equation. This paper used 16x16x16 cubic bins, and the result is shown in Fig. 7(c).

$$q_u = C\sum_{i=1}^{n} g\left(\left\|\frac{x_i}{h}\right\|^2\right)\delta_{ui}, \tag{4}$$

where g is the kernel function, C is the normalization factor, x_i represents the target location, and δ is the Kronecker delta function, u =1…m.

4.2. Particle initialization

Particle filtering is also known as the sequential Monte Carlo method. Several tracking systems have been developed based on particle filtering. In a multiple vehicle tracking system, a color histogram is used as an observation model to weigh each particle. In this system, at the particle filter initialization stage, a large number of particles are generated at the position of the target vehicle in the previous frame, which is described by

$$P_{t+1} = C_t + N(0,\sigma^2), \tag{5}$$

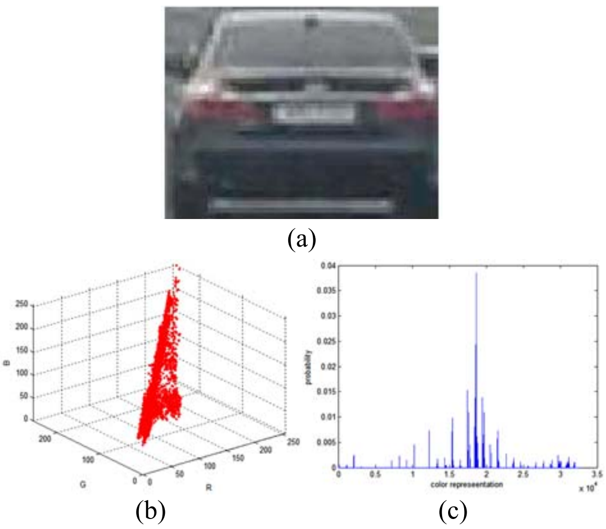


Fig. 7. (a) Target vehicle, (b) HSV color space, (c) color PDF.

where P_{t+1} indicates particle position at frame $t+1$, C_t is the detected (or tracked) target vehicle center at frame t, N is a zero mean Gaussian random value, and σ is Gaussian variance, which depends on tracking window size. Moreover, the number of particles cannot be a constant because of different scale target vehicles; it should be proportional to the tracking window size, which is defined by the following equation.

$$number = \eta h_i w_i, \tag{6}$$

where h_i and w_i indicate the height and width of tracking window I, respectively, η is a scale factor, and 0.1 is chosen in this implementation.

4.3. Particle evaluation

Particle evaluation means using an observation model to evaluate the importance of each particle. In our system, a color model is utilized. Each particle is weighted based on a color probability distribution function. At the first stage, each particle is considered as one pixel. It contains three H, S, and V values in the HSV color space. The weights are computed by the following equation:

$$w_i = CF(p_i), \tag{7}$$

where w_i indicates the weight value, C is a normalized parameter which guarantees the weight values sums to 1, and F is the probability function.

The threshold-based method is used to select some high confidence particles. The number of particles is not sufficient to describe the target vehicle, so the target vehicle will be considered as missing by the system. Otherwise, CPDF can be used to measure the similarity between the selected particles and the target vehicle. In the second stage, each particle is considered a candidate target vehicle. Using the same color space, color histograms of each candidate are computed. To define this similarity, the Bhattacharyya coefficient [20] is used in this paper. It is an approximate measurement of the amount of overlap between two statistical samples. This coefficient can be used to describe the similarity of two discrete, normalized distributions, as shown in (8). Then, the weight value of each particle candidate is computed by (9).

$$\rho_i[p^i, q] = \sum_{u=1}^{m}\sqrt{p^i{}_u q_u}, \tag{8}$$

$$w_i = C\rho_i, \tag{9}$$

where q is the color histogram of the target vehicle, p^i indicates the color histogram of candidate i, ρ_i is the similarity value between target vehicle q and candidate i, C is the normalization parameter, and w_i represents the weight value of particle i.

Finally, the position of the target vehicle in the current frame is estimated using the mean value of those weighted particles, as shown in

$$S = \sum_{i=1}^{m} w_i p^i. \tag{10}$$

4.4. Scale adjustment

The size of the target vehicle varies when the distance between the target vehicle and the camera changes. Consequently, window scale adjustment is essential. In order to obtain a dynamic tracking window, the distance between corresponding light pairs is investigated. Light pairs are segmented in the vehicle candidate generation step and are finally determined during the vehicle verification step using trained BPNN. After applying the vehicle tracking process, the best candidate is obtained. Based on this candidate, the window scale is enlarged by 10%, and a light is detected by color segmentation on the HSV space, which was introduced in Sections 2 A and B. In order to determine the corresponding light pair, the following constraints are proposed: c_1 and c_2 represent left and right lights identified before the tracking process, and c_1' and c_2' represent left and right lights which are detected after the tracking process, respectively.

1) The areas of the corresponding lights are approximately proportional.

$$\frac{S(c_1)}{S(c_1^{'})} \approx \frac{S(c_2)}{S(c_2^{'})} \tag{11}$$

2) The shifts of the corresponding light positions are almost the same.

$$P(c_1) - P(c_1^{'}) \approx p(c_2) - P(c_2^{'}) \tag{12}$$

After corresponding light pairs are identified, the distance between lights after the mean shift iteration, which is determined by the vehicle detection step to be d_0, is defined as d_t, so the size of the tracking window at frame t can be obtained by the following equation:

$$L_t = \frac{d_t}{d_0} L_0, \tag{13}$$

where L_0 is the initial window size, and L_t represents the window size at frame t.

4.5. Vehicle temporarily missing

In the multiple vehicle movement video, the target vehicle may be obscured by another vehicle. Because the observation model of our system is based on color distribution, a particle filter will track the target vehicle until the particle confidence is small, which means the color information of the target vehicle in the current frame is not sufficient to describe the target. Therefore, a color-based particle filter is not sensitive in the partial occlusion case. When the entire body of the target vehicle is blocked by another vehicle, the particle filter will lose the target. However, the system will save the missing vehicle color information for a short period. During this time, particles are generated near the covering vehicle tracking window. Based on these particles, the system selects some effective particles in each frame. When enough effective particles are searched, the missing vehicle can be reidentified. Fig. 8 shows one example of partial occlusion and the temporarily missing problem.

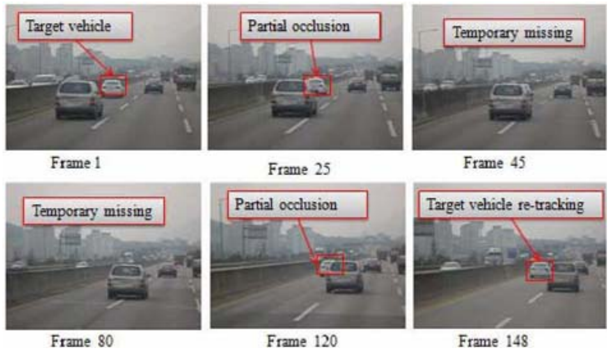
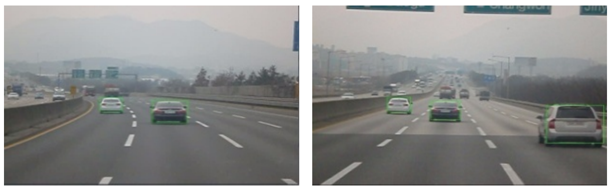


Fig. 8. Vehicle partial occlusion and temporarily missing example.

5. EXPERIMENT

In this section, the experimental results obtained for the proposed method are laid out. The algorithm is implemented by MATLab 7.8.0 (R2009a). The computer processor is an Intel(R) Core(TM) 2 Quad CPU 2.66 GHz, RAM 3.00 GB. One hundred four images are tested for vehicle detection; multiple vehicle movement videos are collected from highways, urban roads and campuses using a normal CCD camera and an HD camera. Video from the normal CCD camera contains images with 480×640 pixels for each frame. Images from the HD camera have been resized to 540×960 in our experiment.

Fig. 9 shows a vehicle detection example on a highway (first column), urban road (second column) and campus (third column). The result shows this approach has good performance for vehicle detection, especially when the target vehicle is nearby. Table 1 shows multiple vehicle detection performance in different environments, with a correct detection rate of around 80 percent.



(a) High way vehicle detection.

(b) Urban road vehicle detection.

(c) Campus vehicle detection.

Fig. 9. Vehicle detection result.

Table 1. Vehicle detection results.

Image origin	Highway	Urban road	Campus
Total number of vehicles	83	106	73
Number of correct vehicle detections	65	87	65
Number of failed vehicle detections	18	19	8
Detection rate (%)	78.3	82.1	89.1

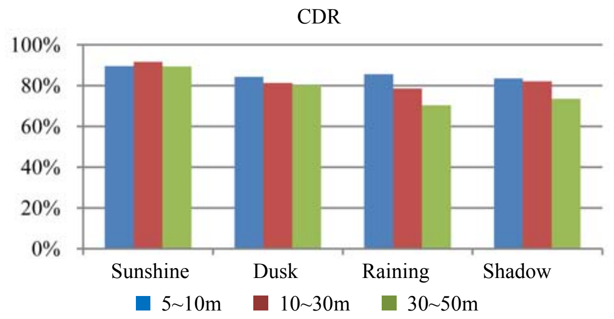


Fig. 10. Vehicle detection under different condition.

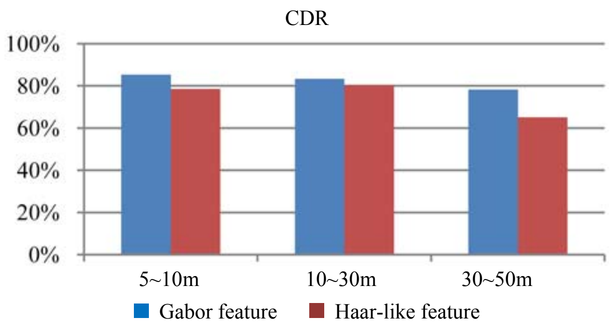


Fig. 11. Vehicle detection performance using Gabor feature/Haar-like feature.

Vehicle tail light segmentation and the Gabor feature extraction technique were used in the vehicle detection sub-system. Tail light segmentation was influenced by lighting conditions. When the target vehicle is very far from the camera, the resulting detection performance is not good, because the color information influences illumination, and the tail light feature is not strong enough. Fig. 10 shows the correct vehicle detection detection rate (CDR) at different distances and different lighting conditions (Sunshine, Dusk, Raining, Shadow).

A Gabor feature set was utilized here for vehicle representation. Fig. 11 is the vehicle detection performance for the Gabor feature compared with the Haar-like feature. A vehicle detection system using the Gabor feature is more stable, because the vehicle can be represented by the Gabor feature at different scales and with suitable orientations.

In our multiple vehicle tracking system, a particle filter tracking algorithm is implemented based on a color probability distribution function. Unfortunately, the color histogram of the target vehicle always changes because of differences in illumination, differences in distance or other unpredictable situations, as shown in Fig. 12. Thus, the BPNN classifier used in the vehicle detection step is

(a) (b)

Fig. 12. (a) Vehicle in different frame (b) Color histogram of detected vehicle.

also applied for updating the color histogram. Fig. 13 shows the tracking result using a different method.

In Fig. 13, the first row shows the particle filter tracking result without color histogram updating. The system yields good tracking performance during the early frames, but the tracking window diverged after frame 100. When the target vehicle is far away, the real CPDF of the target vehicle will change, since the initial CPDF of the target vehicle can only guarantee a few frames of tracking. The second row shows particle filter tracking results with color histogram updating. The tracking window always converged around the real position of the target vehicle.

Fig. 14 shows the tracking results for multiple vehicles, which includes vehicle partial occlusion and the temporarily missing problem. At frame 30, the vehicle still had to be tracked when it was partially occluded. From frame 43 to frame 117, the vehicle was totally obscured by another vehicle. The system saved the color information of the missing vehicle during these frames and reidentified the missing vehicle in frame 118 when it reappears partially. This result shows the system tracks target objects robustly even when they temporally missing.

Fig. 15 shows multiple vehicle detection and the tracking result. Two vehicles were detected in the first frame, and two particle filters tracked them in parallel. The system detected a new vehicle at frame 280. After that, one particle filter was initialized for tracking. The position and tracking window size information are displayed at the top of the video. Fig. 16 shows the exact tracking trajectory of each vehicle in the horizontal, vertical and image planes.

Table 2 shows multiple vehicle tracking results. Usually, most failed detection and tracking cases are caused by weak tail light features. For example, some vehicles, such as trucks, have small red tail lights. The system has difficulty detecting tail light information in these cases. Also, as the distance between the target vehicle and host vehicle increases, tail light information is more and more sensitive to outside illumination, so successful detection rates will decrease.

Frame 1 Frame 55 Frame 62 Frame 115

Frame 1 Frame 55 Frame 62 Frame 115

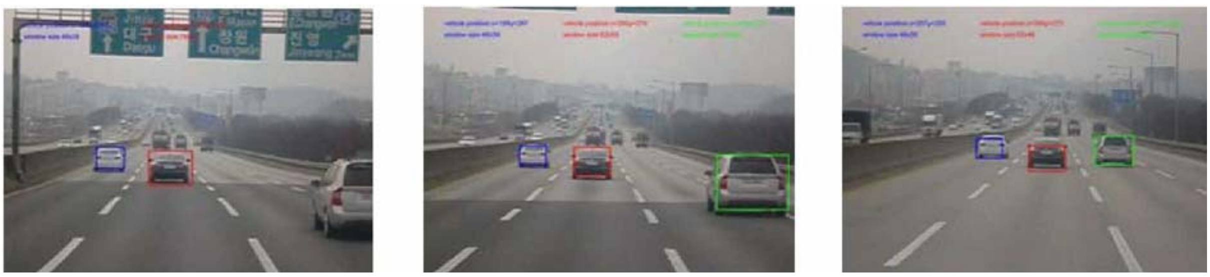
Fig. 13. Tracking comparison results: with(bottom) and without(top) color histogram updating.

Frame 2 Frame 30 Frame 43

Frame 88 Frame 118 Frame 148

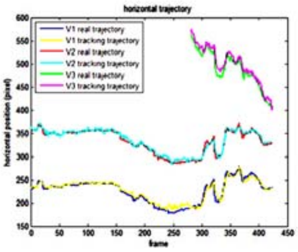
Fig. 14. Vehicle partial occlusion and temporary missing.

Frame 2 Frame 80 Frame 180

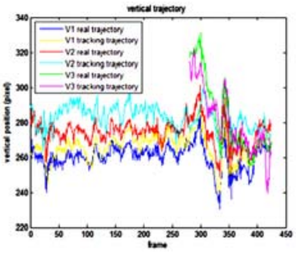


Frame 260 Frame 280 Frame 390

Fig. 15. Multiple vehicle detection and tracking.

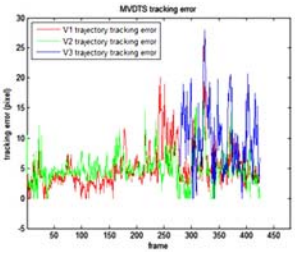


(a) Horizontal trajectory.



(b) Vertical trajectory.

(c) Tracking trajectory in image plane.



(d) Tracking error.

Fig. 16. Vehicle trajectory in horizontal.

Table 2. Multiple vehicle tracking result.

Video (original)	Highway (CCD)	Urban Road (HD)	Campus (CCD)
Total number of frames	5254	952	1326
Image size	480×640	1080×1920	480×640
Tracking window size (in pixels)	(40×32)~(130×104)	(60×48)~(320×256)	(32×25)~(130×104)
Avg. tracking trajectory error (in pixels)	10	23	7
Total number of moving vehicles	12	5	6
Number of missed vehicles	3	1	0

6. CONCLUSION

An effective system for multiple vehicle detection and tracking is proposed in this paper. Vehicle tail light detection is used in the proposed algorithm not only for a vehicle candidate generation, but also for vehicle tracking window adjustment. A back propagation neural network is trained in eight orientations and with five scales A Gabor feature set is used in the vehicle candidate verification step. In the tracking sub-system, multiple vehicle occlusions and temporary missing cases are investigated. This paper presented a novel implementation method using a particle filter to overcome these problems.

The results show that this system has good performance for vehicle detection and tracking, especially when a target vehicle is nearby. Fig. 17 shows an example of failed detection. In the tracking unit, only color information is used for the observation model, which is robust with partial occlusion. However, when the vehicle is partially occluded by the same color vehicle, as shown in Fig. 18, a color-based tracking system cannot accurately track both vehicles. For future work, it is necessary to utilize other observation models or reconstruct a suitable tracking system structure for same-color vehicle occlusion.

Fig. 17. Example of incorrect detection result.

Fig. 18. Same color vehicle occlusion.

REFERENCES

[1] Z. Sun, G. Bebis, R. Miller, "On-road vehicle direction: a review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694-711, May 2006.

[2] Z. Sun, G. Bebis, and R. Miller, "Monocular precash vehicle detection: features and classifiers," *IEEE Trans. on Image Process*, vol. 15, no. 7, pp. 2019-2034, 2006.

[3] H. Cheng, N. Zheng, and C. Sum, "Boosted Gabor features applied to vehicle detection," *Proc. of Int. Conf. on Pattern Recognition*, Hong Kong, vol. 1, pp. 662-666, 2006.

[4] T. Zielke, M. Brauckmann, and W. Von Seelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understanding*, vol. 58, no. 2, pp. 177-190, 1993.

[5] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi, "Visual perception of obstacles and vehicles for platooning," *IEEE Trans. on Intelligent Transportation System*, vol. 1, no. 3, pp. 164-176, 2000.

[6] S. Han, Y. Han, and H. Hahn, "Vehicle detection method using Haar-like feature on real time system," *World Academy of Science, Engineering and Technology*, vol. 59, 2009.

[7] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," *Proc. of the 6th IEEE Workshop on Applications of Computer Vision*, December 03-04, Orlando, Florida,

2002.
[8] C. Goerick, D. Noll, and M. Werner, "Artificial neural networks in real-time car detection and tracking applications," *Pattern Recognition Letters*, vol. 17, pp. 335-343, 1991.
[9] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, August 1995.
[10] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-575, May 2003.
[11] M. Qing and K.-H. Jo, "Vehicle detection and scale-adaptive tracking using tail light segmentation," *Proc. of Image and Vision Computing New Zealand*, pp. 115-119, 2011.
[12] C. Chang, R. Ansari, and A. Khokhar, "Multiple object tracking with kernel particle filter," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 566-573, vol. 1, 2005.
[13] J. Li, W. Ng, S. Godsill, and J. Vermaak, "Online multi-target detection and tracking using sequential Monte Carlo methods," *Proc. International Conference on Information Fusion*, vol. 1, pp. 115-121, 2005.
[14] Y.-M. Chan and S.-S. Huang, "Vehicle detection under various lighting conditions using a particle filter," *IET Intelligent Transp. Syst.*, vol. 6, no. 1, pp. 1-8, March 2012.
[15] Y.-M. Chan and S.-S. Huang, "Vehicle detection under various lighting conditions by Incorporating particle filter," *Proc. IEEE Intelligent Transportation System Conference*, vol. 1, pp. 539-534, 2007.
[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.
[17] Q. Ming and K.-H. Jo, "Vehicle detection using tail light segmentation," *International Forum on Strategic Technology*, August 22-24, Harbin, 2011.
[18] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959-971, October 1996.
[19] C. Goerick, D. Noll, and M. Wener, "Artificial neural networks in real-time car detection and tracking applications," *Machine Vision and Applications*, vol. 12, pp. 69-83, 2000.
[20] J. Lee, M.-H. Jeong, J. Lee, K. G. Kim, and B.-J. You, "3D pose tracking using particle filter with back projection-based sampling," *International Journal of Control, Automation, and Systems*, vol. 10, no. 6, pp. 1232-1239, 2012.
[21] H. Kang, S. H. Lee, and J. Lee, "HCI (Hue-Chroma-Intensity) color model: a robust color representation under illumination changes," *International Journal of Control, Automation, and Systems*, vol. 10, no. 5, pp. 963-971, 2012.

Ming Qing received his M.S. degree from University of Ulsan, Korea in 2012. Currently, he is working at Tsinghua University-Nuctech Joint Research Institute, China. His research interests cover in security equipment using X-rays, image processing and software development.

Kang-Hyun Jo received his Ph.D. degree from Osaka University, Japan, in 1997. He then joined the School of Electrical Eng., University of Ulsan right after having one year experience at ETRI as a post-doc research fellow. Dr. Jo has been active to serve for the societies for many years as directors of ICROS (Institute of Control, Robotics and Systems) and SICE (Society of Instrumentation and Control Engineers, Japan) as well as IEEE IES TC-HF Chair. He is currently contributing himself as an editorial member for a few renowned international journals, such as IJCAS (International Journal of Control, Automation and Systems), TCCI (Transactions on Computational Collective Intelligence, Springer) and IteN (IES Technical News, online publication of IEEE). He had involved in organizing many international conferences such as ICCAS, FCV, IFOST, ICIC and IECON. He had visited for performing his research activity to Kyushu University, KIST and University of California Riverside. His research interest covers in a wide area where focuses on computer vision, robotics, and ambient intelligence.