Proceedings of the 2007 IEEE
Intelligent Transportation Systems Conference
Seattle, WA, USA, Sept. 30 - Oct. 3, 2007

Vehicle Detection under Various Lighting Conditions by Incorporating Particle Filter

Yi-Ming Chan, Shih-Shinh Huang, *Member, IEEE*,
Li-Chen Fu, *Fellow, IEEE,* and Pei-Yung Hsiao, *Member, IEEE*

***Abstract*—We propose an automatic system to detect preceding vehicles on the highway under various lighting and different weather conditions based on the computer vision technologies. To adapt to different characteristics of vehicle appearance at daytime and nighttime, four cues including underneath, vertical edge, symmetry and taillight are fused for the preceding vehicle detection. By using Particle Filter with four cues through the processes including initial sampling, propagation, observation and cue fusion and evaluation, particle filter accurately generates the vehicle distribution. Thus, the proposed system can successfully detect and track preceding vehicles and be robust to different lighting conditions. Unlike normal particle filter focuses on a single target distribution in a discrete state space, we detect multiple vehicles with particle filter through a high-level tracking strategy using clustering technique called basic sequential algorithmic scheme (BSAS). Finally, experimental results for several videos from different scenes are provided to demonstrate the effectiveness of our proposed system.**

I. Introduction

Owing to the advanced technologies and the well-off living, vehicles are popular transportation now. But, at the same time, the number of deaths each year due to the usages of motor vehicles is increasing. In [1], the number of accidents related to motor vehicles is 6,181,000 at 2004 in the USA. 42,636 people were killed, and about 2,788,000 people were injured. This is the reason why developing a driver assistance system (DAS) can not be over emphasized.

Vision-based DAS approaches [2][3] are recently popular since the power of the environment description and the low price is superior then other sensors. This paper focuses on the vision-based DAS, which is achieved by using Particle Filter with high level tracking technique.

Although the preceding vehicle pattern is specific and regular, it is still hard to recognize them from the natural scene. The main challenges of vehicle detection are:

1) The system should detect any kinds of vehicles such as sedans, vans, wagons, buses, and so on.
2) The system should work well under various conditions such as sunny, cloudy, and moist weather conditions. Furthermore, such system function should stand when the vehicle is driving through a tunnel.
3) The computing cost should be rather economical since this system will be exploited in real time driving environment.

There are numerous approaches proposed so far to perform vehicle detection. The intensity-based symmetry method [4] is often used in the car-following situation, as the rear of most vehicles is typically symmetrical. However, this method is limited to the strict car-following or situations where the object being dealt with displays some degree of symmetry. Besides, the method of symmetry may cause false alarms due to the objects, e.g., fences, with symmetry property on roadway. In ARGO project, Broggi *et al.* [5][6] produced a symmetry map by combining the gray-level and horizontal-edge symmetry information. Then, the position of the vehicle's bottom is found by fitting a template to the edge map.

In the low light condition, there are several approaches using cues other from normal daytime situation. Chem *et al.* [7] proposed a monocular vision system on the highway at night. Wang *et al.* [8] proposed vehicle detection system based on the taillight detection. Because of the taillights are the most significant feature of the vehicle at night, they apply blob template in the night scene for the taillight detection.

In case of one fail of single cue resulting the whole system down, there are some approaches using multiple cues. Huang *et al.* [9] employed vertical edge, underneath and symmetry as the prior knowledge of the vehicle. Tao *et al.* [10] proposed sequential Monte Carlo approaches with line and color based features for the vehicle. Martin *et al.* [11] argued that applying multiple cue integration with democratic integration can improve robustness of visual tracking. They showed the democratic integration combined with CONDENSATION [12] can achieve the goal of multiple hypotheses in visual tracking.

Particle Filter is known as CONDENSATION or SMC (Sequential Monte Carlo) method. It has been widely used in the area of tracking [13][14][15], because it can adequately model the multimodal and no-Gaussian distribution. It is not only easier to track multiple objects but also simpler to integrate multiple vision cues in the particle filtering. We know there are several features commonly adopted for the vehicle candidate, but not all of them always work well under all conditions. In our system, we try to propose a way to integrate these features intelligently and effectively. We realize that

Yi-Ming Chan is with the National Taiwan University, Taiwan (886-2-33661367; e-mail: yimingchan@gmail.com).

Shih-Shinh Huang is with the National Taiwan University, Taiwan (e-mail: poww@gmail.com).

Li-Chen Fu is with the National Taiwan University, Taiwan (e-mail: lichen@ntu.edu.tw).

Pei-Yung Hsiao is with the National University of Kaohsiung, Taiwan (e-mail: pyhsiao@nuk.edu.tw).

some cues will not be taken into account when they fail to contribute to the recognition task positively. By saving computation efforts for executing those cues, the efficiency of our system will be improved apparently.

For the purpose of robustness and multiple target tracking, we need to track multi-modal density distribution. Among probabilistic tracking approaches, Kalman filtering [16] is the most well-known approach which uses Gaussian distribution density to represent the target. Kalman filtering is good for tracking a single target, but inadequate for multimodal situation. One solution of dealing with multiple targets is Particle Filter framework, which can easily integrate cues and be extended to multiple target tracking. To extend the Particle Filter to multi-target tracking version, we cluster particles into several groups by the Basic Sequential Algorithm Scheme (BSAS) [17], and then use Kalman Filter with constant velocity model to accomplish the goal of high-level tracking.

In our system, we propose several techniques to improve the performance of the Particle Filter. We enhance the initial sampling to converge earlier and use multiple cues to improve the robustness. First, the data-driven initial sampling draws particles from high likelihood area instead of random guessing from the whole state space; it is important that fast converging when new target appears. Second, in order to apply this vehicle tracking algorithm under various lighting conditions, we use multiple cues for each candidate. These cues will be integrated by cue fusion procedure to get the final likelihood of the candidate. In order to extend the particle filter into multiple target approaches, we cluster the particles with BSAS. The BSAS in our system is integrated with high-level tracking framework. After an iteration of the particle filters algorithm, the most probable candidate will propagate to the next frame through a high-level tracking technique. The vehicle detection and tracking architecture is shown in Fig. 1.

The rest of the paper is organized as follows. In Section II, we will introduce the idea of the particle filter and the first two processes, the initial sampling and the propagation. In Section III, the observation and evaluation will be described. Section IV shows the experiments. Then, we will conclude the paper at Section V.

II. Particle Filter And Hypothesis Generation

The particle filtering plays the most important role in our approach. These "particles" in particle filters are all possible candidates of a vehicle; the particles are interchangeable with samples through this paper. In the following sections, we will discuss how to use particle filter for vehicle tracking. The first two processes, the initial sampling and the propagation, are for hypothesis generation part in the framework.

A. Data Driven Initial Sampling

Although the lane detection is not the focus of the paper, we use the lane boundaries information [8][9] to generate the region of interest (ROI) as in Fig. 2 before vehicle detection. The ROI not only can reduce complexity in searching for vehicle candidates but also can decrease the false positive detection rate. We will only evaluate vehicle candidates in the region of interest, that is, we only generate particles in the ROI.

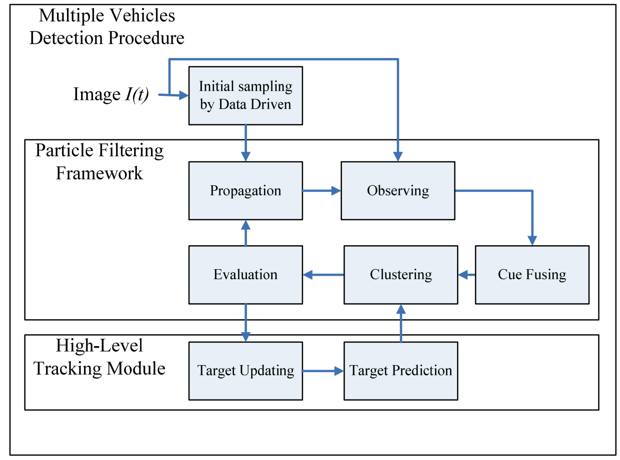

Fig. 1 The architecture of the vehicle detection procedure

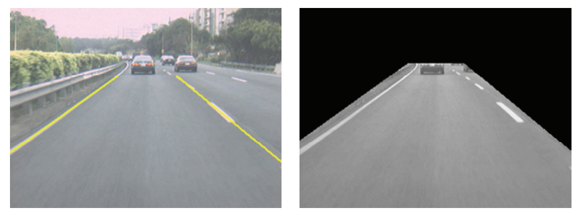

Fig. 2 The lane boundaries and the region of interest

In our approach, the particles in the particle filter framework stand for the possible candidates of a vehicle. Here, we use a vector to represent each vehicle candidate.

$$\mathbf{x}_t^k = (u_t, v_t, w_t, h_t, \Delta u_t, \Delta v_t, \Delta w_t)$$

In other words, the *k-th* vehicle candidate state vector $\mathbf{x}^k$ at time step t is denoted as the following notations:

(u_t, v_t) : (U, V) of the top left corner

(w_t, h_t) : (width, height) of the sample

$(\Delta u_t, \Delta v_t, \Delta w_t) = (u_t - u_{t-1}, v_t - v_{t-1}, w_t - w_{t-1})$

See the following illustration in Fig. 3 for the detail. Note that the element Δh_t is always proportional to the Δw_t in the vehicle cases; the Δh_t is abandoned in our state vector. In the system, we also define the measurement vector at time step $\boldsymbol{t}$ as z_t^k , $z_t^k = (z_t^{im}, z_t^{un}, z_t^{ve}, z_t^{tl})$, which includes image z_t^{im} , vertical edge map z_t^{ve} , underneath map z_t^{un} and taillights map z_t^{tl} .

If the samples are good enough, the samples will converge to the solutions soon. The "initial guess" is very important if we want to converge earlier. The idea goes to the data driven approaches; instead of random guessing, we draw samples according to the information from current image. Isard and Blake [18] proposed an importance sampling approach to get a better solution when new object appears. Similar to Isard and Blake, we use the data-driven technique to provide a better guess of initial sampling. We draw samples with high likelihood of the underneath cue and the vertical cue first. We randomly generate possible candidates after lane detection in one iteration. Only 10% of samples with highest likelihood score are taken into the later iteration, the other 90% randomly samples are abandoned. After we have previous detection results, we keep generating 10% of the samples from *data driven initial sampling*, and 90% of the samples are generated by Particle Filter framework from previous time step. The 10% new samples take charge of newly introduced target while the rest are responsible for tracked target from previous step. The data driven initial sample help the system converges faster then random initial while new target appears.

The number of samples is a constant and can be chosen on the basis of empirical experience. If the number of samples is close to infinity, the final estimation will converge to the true posterior density. In general, the computational complexity of the particle filtering is linear proportional to $\boldsymbol{n}$, *i.e.*, the number of samples. Owing to maintaining efficiency, we prefer not to use infinite samples. From our experiments, it is shown that the targets are sufficiently well tracked when we use the number of sample set $\boldsymbol{n} = 100$.

B. Propagation

The propagation of the samples is important in the hypothesis generation for the tracking system. In the propagation, the samples will propagate to the new position as a new "guess". This stage consists of two steps, namely *deterministic drift* and *randomize diffuse*.

In the deterministic drift step, we use constant velocity model for each sample set. Because of the relative velocity of vehicles on the highway usually changes with constant interval, the constant velocity model is good enough for most of scenes. Thus, we simply add the velocity component to the center coordinate of the sample set to predict the state of samples in next time step. Samples with identical state vector will identically move to the same position. Next, we use randomize drift in each element of the vehicle candidate. This will only slightly move the vehicle candidate randomly, so that the identically sample may not stay at the sample set after the diffuse. This is important since we can not estimate the real dynamic model of the target and random move can give the sample a chance to drift to the correct state.

From now on, the propagation will have new samples from the previous frame. These new samples are ready for the next steps namely, observation and cues fusion. In the next section, we will introduce how we achieve observation and cues fusion.

III. Particle Filter And Hypothesis Verification

According to observations, the appearance of a vehicle in an image has many properties in the image. We refer to these

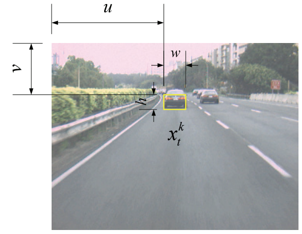

Fig. 3 The illustration of a vehicle candidate vector

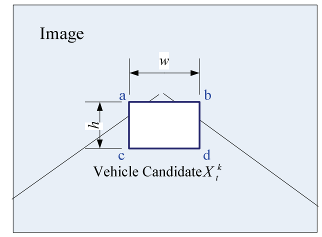

Fig. 4 Vehicle candidate for evaluates likelihood. The rectangle *abdc* denotes a vehicle candidate

properties as "cues" for measuring the observation results for every candidate. In the observation steps, we try to assign a weight for every candidate by determining an appropriate likelihood. In other words, every cue will have likelihood for every measurement. Usually, a single cue will not be robust enough for us to find a good solution due to complex environment. The idea behind the system is to integrate multiple cues so as to obtain more accurate results.

A. Observation

The likelihood of each measurement can be evaluated by the currently observed image with respect to the four cues. These four cues are vertical edge cue, underneath cue, symmetry cue, and taillight cue. We will explain how each cue is evaluated in the following. The notations are defined in a schematic form as shown in Fig. 4.

Vertical Edge Cue

The vehicle candidate will have edge pixels on the vertical boundaries if the bounding box fits the vehicle properly [9]. In the vertical edge cue, it will return a high value if the vertical boundaries contain many edge pixels. We define the likelihood of vertical edge cue as the number of total vertical edge pixels divided by vertical boundaries. η^{v} is the normalized term.

$$g^{\mathrm{v}}(z_t|x_t) = \frac{Vertical\ edge\ pixels\ on\ \overline{\mathrm{ac}}+\overline{\mathrm{bd}}}{\overline{\mathrm{ac}}+\overline{\mathrm{bd}}} \tag{1}$$

$$p^{(\mathrm{v})}\left(z_t|x_t\right) = \frac{1}{\eta^{\mathrm{v}}}\exp(g^{\mathrm{v}}\left(z_t|x_t\right)) \Rightarrow p^{(\mathrm{v})}\left(z_t|x_t\right) \propto \exp(g^{\mathrm{v}}\left(z_t|x_t\right)) \tag{2}$$

Underneath Cue

The shadow under the vehicle is a strong feature for a vehicle; we call it underneath cue [9]. The underneath is a region of dark part in the image. We use horizontal edge detection with low intensity to locate underneath pixels. For the underneath cue, we use the ratio between underneath pixel and the bottom boundary of the vehicle candidate.

$$\begin{aligned} g^{\mathrm{u}}\left(z_t^{un}|x_t\right) &= \frac{Underneath\ pixels\ on\ \overline{\mathrm{cd}}}{\overline{\mathrm{cd}}} \\ p\left(z_t^{un}|x_t\right) &= \frac{1}{\eta^{\mathrm{u}}}\exp(g^{\mathrm{u}}\left(z_t^{un}|x_t\right)) \\ &\Rightarrow p\left(z_t^{un}|x_t\right) \propto \exp(g^{u}\left(z_t^{un}|x_t\right)) \end{aligned} \tag{3}$$

Taillight Cue

In the night scene, taillights are the most significant feature for a vehicle [7][8]. The taillight will show as a blob in the night scene image. For one vehicle, there are two taillights in the vehicle candidate bounding box. We search two taillight spots in the bounding box, and calculate the distance between two farthest taillights. The taillight cue uses the taillight distance in the bounding box to represent the likelihood of the vehicle candidate.

$$g^{\mathrm{tl}}\left(z_t^{tl}\middle|x_t\right)=\frac{\textit{Taillight distance}}{w} \tag{4}$$
$$p\left(z_t^{tl}\middle|x_t\right)=\frac{1}{\eta^{\mathrm{tl}}}\exp(g^{\mathrm{tl}}\left(z_t^{tl}\middle|x_t\right))\Rightarrow p\left(z_t^{tl}\middle|x_t\right)\propto\exp(g^{\mathrm{tl}}\left(z_t\middle|x_t\right))$$

Symmetry Cue

The vehicle usually has symmetry property in the rear part [5][6]. We calculate symmetry function within the bounding box only. The symmetry cue evaluates the ratio between symmetry pixels and the bounding box width and the symmetry likelihood can be defined as

$$g^{s}\left(z_t^{im}\middle|x_t\right)=\frac{\sum_{i=1}^{w/2}\sum_{j=1}^{h}k(i,j)}{\frac{w\cdot h}{2}}$$
$$k(i,j)=\begin{cases}1, & \text{if } \left|\frac{I(i,j)-I(w-i,j)}{I(i,j)}\right|<\theta_{sym}\\ 0, & \text{else}\end{cases}$$
$$p\left(z_t^{im}\middle|x_t\right)=\frac{1}{\eta^{s}}\exp(g^{s}\left(z_t^{im}\middle|x_t\right))\Rightarrow p\left(z_t^{im}\middle|x_t\right)\propto\exp(g^{s}\left(z_t^{im}\middle|x_t\right)) \tag{5}$$

The notation *I(i, j)* here means the intensity of the image at the bounding box coordinate (*i, j*), the *i*-th column and *j*-th row from the origin of the bounding box. The *h* is the bounding box height, and *w* is the bounding box width; both are in terms of pixels. The origin of the bounding box is defined as the top left corner of the bounding box. θ_{sym} is the threshold for a symmetry pair.

These four cues will not always be evaluated in each run because of the efficiency problem. Only cues with higher weight will be assessed in the next frame. The following context will show how the cues are fused.

B. Cue Fusion

We will use only one likelihood density for one vehicle candidate, even though we may have more than one observation. This facilitates us to build a robust driver assistance system since the cues may contain various lighting condition. For example, underneath cue is good at day time whereas taillight cue is good at night time.

To integrate four likelihood functions derived from four cues aforementioned, the natural way is to use a weighted sum to represent the fused likelihood density. Assume that each cue is independent with the other cue; we can evaluate the weighted likelihood $p(z_t^i|x_t^i)$ as multiplications of each cue with a constant η.

$$p\left(z_t^i\middle|x_t^i\right)=\eta\cdot p\left(z_t^{ve}\middle|x_t^i\right)\cdot p\left(z_t^{un}\middle|x_t^i\right)\cdot p\left(z_t^{tl}\middle|x_t^i\right)\cdot p\left(z_t^{im}\middle|x_t^i\right) \tag{6}$$

Rewrite the (6) into the energy form;

$$p\left(z_t^i\middle|x_t^i\right)\propto\exp(W_t^{v}\cdot g^{v}\left(z_t^{ve}\middle|x_t\right)+W_t^{u}\cdot g^{u}\left(z_t^{un}\middle|x_t\right)+W_t^{tl}\cdot g^{tl}\left(z_t^{tl}\middle|x_t\right)+W_t^{s}\cdot g^{s}\left(z_t^{im}\middle|x_t\right))$$

W_t^i, decided by the empirical experience, is the fusion weight of the feature *i* at time step *t* . Thus, we can take the weighted sum of each cue to represent the fused likelihood.

C. Evaluation

After Cue Fusion stage, we have examined the samples in the current time step, and get the likelihood of each sample. How to evaluate the final belief of the state is the topic of later part this section. Isard and Blake [12] choose weighted sum of samples to represent the final belief for single target tracking. It is not adequate for our system, since the target distribution is usually multi-model. In a multi-modal system, *i.e.*, multiple modes appear in the current belief density distribution. A 1D example is shown in Fig. 5. The modes, local maximum, are the most possible solutions in the current time. As we only draw finite number of samples in the state space, we can not just choose the maximum likelihood samples as the best belief. We approximate the belief state by *computing the weighted mean of samples belong to one mode*.

We need to cluster samples first then compute the weighted mean for the final belief. In our case, the number of cluster is unknown; the traditional supervised clustering techniques like k-means are not suitable. Thus, an unsupervised clustering technique named Basic Sequential Algorithm Scheme, BSAS [17], is adopted to cluster samples. The flow chart of traditional BSAS is shown in the Fig. 6. C_d is the threshold of cluster width and $N_{\max}$ is the maximum number of clusters.

As the BSAS algorithm, the clustering results heavily depend on the input sequence change. Therefore, we need to make sure that if we input with meaningful samples before the noise ones. This gives us the idea to improve the solution of the BSAS: to feed the solutions of previous time step first. Instead of giving samples with random input sequence, we feed the previous solutions first. As the distance of these samples are large (or you will not take it as a solution, you will merge it to the closest one) and they are meaningful (possible solutions), the results of the clustering are stable. We maintain the high level solutions by assign each solution a Kalman filter with constant velocity motion model prediction, and then propagate the solution to the BSAS input. Multiple

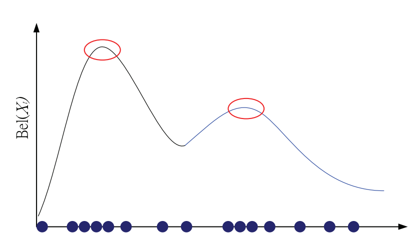

Fig. 5 1D example of a multi-model belief distribution.

objects thus can be tracked at the same time within one particle filter in a very efficient way.

IV. EXPERIMENTS

The hardware architecture of the system is shown in Fig. 7. A CCD camera is used to consecutively grab the images of the road scene. The grabbed images (320×240) are delivered to the mobile computer equipped with Intel® Pentium® M 1.4 GHz processor and 512 MB RAM through IEEE 1394 interface. The camera mounted behind the driving mirror on the vehicle (See Fig. 8) is to continuously extract the traffic scenes at the driving speed between 30 km/h and 100 km/h.

The vehicle recognition results are shown as follows. At first, we generate the region of interest (ROI) from the result of lane detection. Then we only apply sampling techniques in the area of ROI. The details are shown in the Fig. 9.

The observation stage in the proposed particle filtering contains four different of vision cues, namely, vertical edges cue, taillight cue, underneath cue and symmetry. The measurement vectors are shown as follows in the Fig. 10. Fig. 11 shows the multiple target detection results of the system. Fig. 11 (a) is the detection result with high probability. We can see that the truck is also successfully detected although it is quite differ from the sedan. The numbers in the Fig. 11 (b) show the distance of preceding vehicles. We estimated them by inverse perspective transformation. Some special lighting conditions examples are shown in Fig. 12. The overexposure, due to

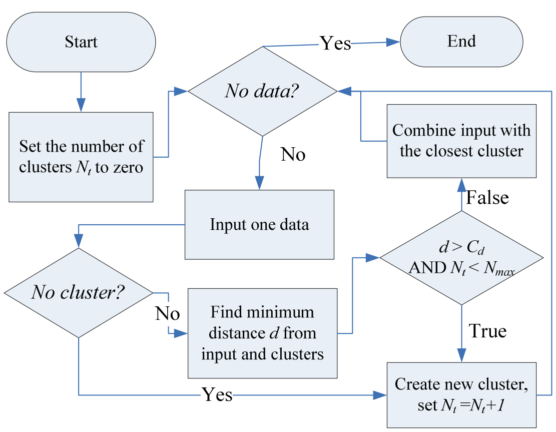

Fig. 6 The flow chart of the Basic Sequential Algorithmic Scheme (BSAS).

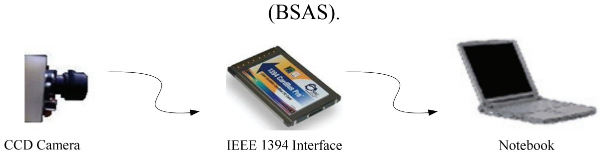

Fig. 7 The image grabber and processing system.

Fig. 8 The setup of the camera

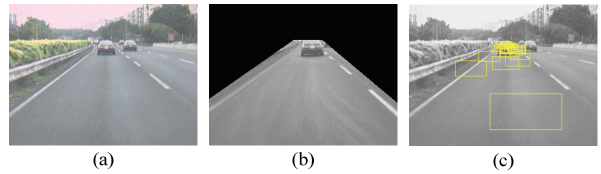

Fig. 9 The region of interest and the initial sampling.
(a) Original image.; (b) The region of interest and (c) The initial sampling in the region of interest using 100 samples.

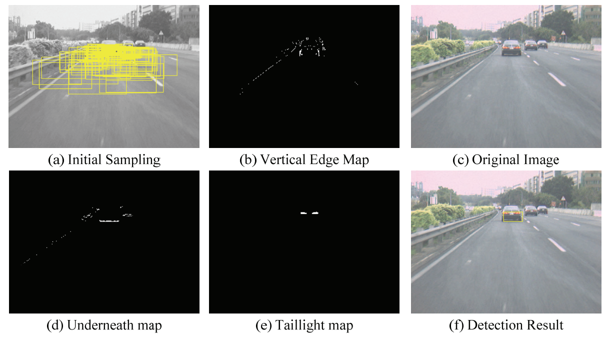

Fig. 10 The measurement vectors and the detection result

Fig. 11 Multiple target detection result

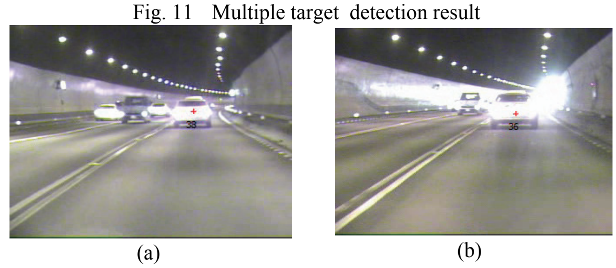

Fig. 12 Special lighting conditions

high contrast, makes the vertical edges blur and disappear; the high level tracking strategy gives us the chance to track vehicle even the features of vehicle are not clear.

The system performance analysis of our system contains two parts, the detection rate and the processing time. In the detection rate analysis, we used more than 28,000 images with more than 10,000 vehicles within 50 meters for the experiment. The details are shown in the Table I. The "*Hit*" means the number of correct detection of vehicles and the "*Miss*" represents the number of false negative. The system has a high detection rate up to 98% at clear scenes; even that the shadow appears on the ground or the host vehicle changes lane. The performance will have slightly deterioration in some cases that the host vehicle drives into tunnel; the blooming effect blurs the vertical edges, and the underneath of preceding vehicles. In the special scene of sunny rain with facing sun, the quality of the images is worse. The rain drop

TABLE I
VEHICLE DETECTION RATE

	Hit	Miss	Detection Rate(%)
Under Overpass	833	58	93.34%
Tunnel	1577	598	72.50%
Left Turn, Changing Lane	622	68	90.14%
Sunset, Lateral	905	33	96.48%
Sunset, Backward	1756	24	98.65%
Night time, Lane Changing, Tunnel	1194	344	77.63%
Sunny Rain, Facing Sun	1868	429	81.32%
Total	8755	1554	84.925%

TABLE II
VEHICLE DETECTION AND TRACKING PROCESSING TIME

Average(ms)	Σ(ms)	Minimum	Maximum
47.03	0.344	46.41	47.48

on the windshield refracts the light of environment, so that, the preceding vehicles are distorted. The sunshine, when facing the sun, lengthens the shadow of vehicles and blurs the vertical edges of preceding vehicle; this is why the performance down in the last scene. The high level tracking in our system gives us a chance to track the preceding vehicle with short period of "missing." The overall system performance is 84.9%. In the near future, the performance will be improved if better vehicle cues for low light conditions are involved.

The time for processing one frame to perform vehicle detection and tracking is less than 48 ms. Since the processing time is closely related to the number of samples, the standard deviation of processing of different scenes is small. The details of the processing time with the number of samples = 100 are shown in the table II.

V. CONCLUSIONS

In this paper, a statistical framework called particle filter is adopted to integrate the four cues including vertical edge, underneath, symmetry and the taillights. By fusing multiple cues for vehicle detection, our system can achieve high detection rate both at daytime and nighttime. Besides four steps including initial sampling, propagation, observation, cure fusion and the evaluation, in the particle filtering framework, a high-level tracking strategy with BSAS clustering technique is chosen to deal with the case that the scene is with blur image for a short period. Finally, seven videos are taken to validate the effectiveness of our system and the detection rate is 84.9% even under some difficult environment and various lighting conditions. And, the frame rate of our proposed system is roughly 20 fps and it is ready for real-time application. However, in some cases, the detection results are unsatisfactorily. In the future, we will introduce more cues for the vehicle detection which can be learned through on-line or off-line mechanisms.

ACKNOWLEDGEMENT

This work is partly supported by the Chung-Shan Institute of Science and Technology, Taiwan, R.O.C.. We also gratefully acknowledge the support by the National Science Council, Taiwan, R.O.C., grant NSC95-2221-E-390-039.

REFERENCES

[1] http://www-nrd.nhtsa.dot.gov/departments/nrd-30/ncsa/
[2] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection Using Evolutionary Gabor Filter Optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, pp. 125-137, 2005.
[3] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, p. 694-711, 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] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo Vision-based Vehicle Detection," *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, pp. 39-44, Oct. 2000.
[6] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi, "Visual Perception of Obstacles and Vehicles for Platooning," *IEEE Transactions on Intelligent Transportation System*, vol. 1, No. 3, pp. 164-176, 2000.
[7] M.-Y. Chern and P.-C. Hou, "The lane recognition and vehicle detection at night for a camera-assisted car on highway," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2110-2115, 2003.
[8] C.-C. Wang, S.-S. Huang, L.-C. Fu and P.-Y. Hsiao, "Driver Assistance System for Lane Detection and Vehicle Recognition with Night Vision," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3314-3319, 2005.
[9] S.-S. Huang, C.-J. Chen, P.-Y. Hsiao, and L.-C. Fu, "On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2456-2461, 2004.
[10] X. Tao and C. Debrunner, "Stochastic car tracking with line- and color-based features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, pp. 324-328, 2004.
[11] S. Martin and S. Bernt, "Towards robust multi-cue integration for visual tracking," *Machine Vision and Applications*, vol. 14, pp. 50-58, 2003.
[12] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, pp. 5-28, 1998.
[13] C. Chang, R. Ansari, and A. Khokhar, "Multiple object tracking with kernel particle filter," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 566-573 vol. 1, 2005.
[14] C. Hue, J. P. Le Cadre, and P. Perez, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Transactions on Signal Processing, [see also IEEE Transactions on Acoustics, Speech, and Signal Processing],* vol. 50, pp. 309-325, 2002.
[15] J. Li, W. Ng, S. Godsill, and J. Vermaak, "Online multitarget detection and tracking using sequential Monte Carlo methods," in *Proc. International Conference on Information Fusion*, vol. 1, pp. 115-121, 2005
[16] R. Kalman, "A New Approach to Linear Filtering and Prediction," *Journal of Basic Engineering*, pp. 35-45, 1960.
[17] S. Theodoridis, K. Koutroumbas, *Pattern Recognition,* Academic Press, p.p.433, 2003.
[18] M. Isard and A. Blake, "ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework," in *Proc. European Conference on Computer Vision*, vol. 1, pp. 893-908, 1998.