DOI: 10.3901/CJME.2014.0707.118, available online at www.springerlink.com; www.cjmenet.com; www.cjmenet.com.cn

Simultaneous Multi-vehicle Detection and Tracking Framework with Pavement Constraints Based on Machine Learning and Particle Filter Algorithm

WANG Ke[1, 2], HUANG Zhi[1, *], and ZHONG Zhihua[1]

1 *State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, China*
2 *College of Automotive Engineering, Chongqing University, Chongqing 400044, China*

Received April 23, 2013; revised May 9, 2014; accepted July 7, 2014

Abstract: Due to the large variations of environment with ever-changing background and vehicles with different shapes, colors and appearances, to implement a real-time on-board vehicle recognition system with high adaptability, efficiency and robustness in complicated environments, remains challenging. This paper introduces a simultaneous detection and tracking framework for robust on-board vehicle recognition based on monocular vision technology. The framework utilizes a novel layered machine learning and particle filter to build a multi-vehicle detection and tracking system. In the vehicle detection stage, a layered machine learning method is presented, which combines coarse-search and fine-search to obtain the target using the AdaBoost-based training algorithm. The pavement segmentation method based on characteristic similarity is proposed to estimate the most likely pavement area. Efficiency and accuracy are enhanced by restricting vehicle detection within the downsized area of pavement. In vehicle tracking stage, a multi-objective tracking algorithm based on target state management and particle filter is proposed. The proposed system is evaluated by roadway video captured in a variety of traffics, illumination, and weather conditions. The evaluating results show that, under conditions of proper illumination and clear vehicle appearance, the proposed system achieves 91.2% detection rate and 2.6% false detection rate. Experiments compared to typical algorithms show that, the presented algorithm reduces the false detection rate nearly by half at the cost of decreasing 2.7%–8.6% detection rate. This paper proposes a multi-vehicle detection and tracking system, which is promising for implementation in an on-board vehicle recognition system with high precision, strong robustness and low computational cost.

Keywords: simultaneous detection and tracking, pavement segmentation, layered machine learning, particle filter

1 Introduction

With continuously growing of vehicle ownership over the past few decades, traffic accidents have become the important cause of fatalities. Traffic accidents, approximately, injured 20–50 million people worldwide each year, and caused at least 1.2 million yearly deaths[1]. Therefore, in recent years, vision-based driver assistant system has become an active research area[2].

It is widely recognized that robust detection and tracking of adjacent vehicles in highway and urban traffic merely using a monocular camera system remains a challenging task. The main difficulties are as follows: firstly, with the camera installed on a mobile vehicle, the vehicle detection algorithm confronts large variations of environment with ever-changing background and illumination[3]. Secondly, there are large variations of vehicles in shape, color and appearance, which are too large to be modeled[4]. Thirdly, the ego vehicle and other vehicles on the road are generally in motion, and therefore, the sizes and locations of vehicles in the image space are diverse[4].

Various vehicle detection approaches have been reported in literatures. The statistical approach, such as principal component analysis(PCA) and independent component analysis(ICA) were used[5]. This method showed high accuracy but the efficiency is low. A symmetric property and shape-based models were adopted[6], which is usually affected by brightness variation. Support-vector-machine (SVM) approach was used[7], where Sun et al. built multiple detectors using Gabor filters, with the conclusions that the method could achieve robust detection at the cost of high computing power. To implement on-board vehicle recognition, the following methods were widely used: 1) knowledge-based method; 2) feature matching method; 3) sensor fusion method. The knowledge-based method utilizes characteristics to recognize objects and assumes the potential vehicle locations. Typical characteristic information is classified into following categories: shadows[8], corners[9], edges[10], headlights[11], textures[12] and symmetry[6]. This method occasionally suffers from

* Corresponding author. E-mail: huangzhi@hnu.edu.cn
Supported by Open Research Fund of State Key Laboratory of Advanced Technology for Vehicle Body Design & Manufacture of China (Grant No. 61075002) and Hunan Provincial Natural Science Foundation of China (Grant No. 13JJ4033)

unsteadiness due to the changing backgrounds[4]. Feature matching approach mainly involves extraction and classification of features, and training of classifiers using a set of positive and negative images. The most frequently used features are as follows: SIFT[13], PCA[14], SURF[15], HAAR-LIKE[16] and HOG[17]. Classifiers commonly used in feature matching are NN[18], SVM[13] and AdaBoost[19].

Although there have been numerous literatures on vehicle detection and tracking or combination of them, few research used simultaneous detection and tracking method to build an on-board multi-vehicle recognition system. In this paper, a novel framework for simultaneous detection and tracking of vehicles is introduced, which has the advantages of global optimum and robustness by fusion of information from detection and tracking modules.

2 Framework of Simultaneous Multi-vehicle Detection and Tracking

It is well known that, vehicle detection method, which essentially searches for the global-optimal-solution in the whole image, may lead to heavy computation load. Global searching can obtain the desired results, but occasionally targets may be missed, and false alarms may occur. Whereas, vehicle tracking method commits themselves to searching for the local-optimal-solution within the expected image area, with the information of priori-knowledge and spatial-temporal continuity. Unlike conventional approach of sequential structure by isolating these two stages, this paper proposes a simultaneous detection and tracking approach, which takes the advantages of both the global optimal solution of detection module and robustness of tracking module, as shown in Fig. 1.

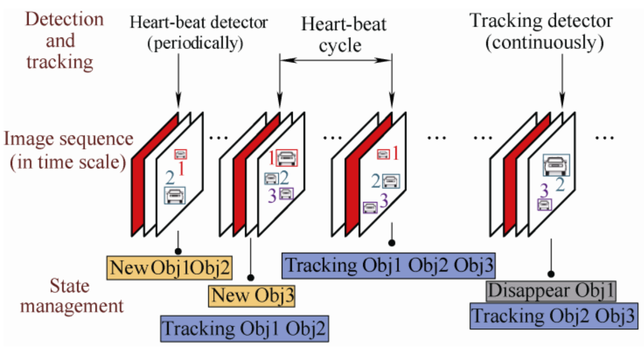


Fig. 1. Simultaneous multi-vehicle detection and tracking framework in time scale

2.1 Framework description in time scale

To achieve robustness and high accuracy, both the spatial feature and temporal information are integrated simultaneously to construct a vehicle recognition system. However, as more modules are employed, the computational load increases significantly. Then, as presented in Fig. 1, a tradeoff framework is presented for rapid and robust vehicle detection and tracking.

The detection module is responsible for entry-exit state management and the tracking module is in charge of accurately and efficiently tracking of targets, which are confirmed in the last detection cycles.

2.2 Modules description

According to the aforementioned strategy, vehicle recognition system is composed of three basic modules: detection module, tracking module and state management module. In the schematic diagram, as shown in Fig. 2, the basic modules and procedures at different operation level are demonstrated.

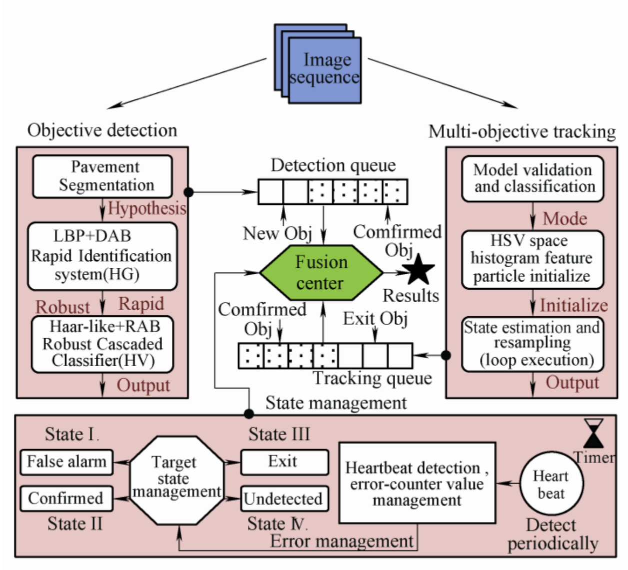


Fig. 2. Schematic diagram of the vehicle recognition system

To search for multiple vehicles within a single video frame, a layered machine learning approach is proposed. The detector employs DAB learning method trained by LBP features, in the early layer, for rapid hypotheses generation. During the coarse-search process, candidates are obtained, which include not only vehicle objectives but also falsealarm. Therefore, in later layer, the detector trained by RAB with Haar-like features is introduced for verification. During fine-search step, each hypothesis is verified once again with the robust boosted cascades classifier.

As is well known that, vehicles just appear at the vanishing edge of pavement area since they always run on the surface of pavement. Considering this fact, a pavement segmentation method is introduced in vehicle detection stage, which separates the road from the non-road region to generate potential vehicle positions.

In tracking module, a particle filter is adopted to track vehicles in successive video frames. Unlike normally particle filter focusing on single target distribution, this paper concentrates on multi-vehicle tracking by introducing the target states of the tracked vehicles.

3 Pavement Segmentation

In the segmentation process, small fragments of the road region are extracted. Then the extracted fragment samples are used as seeds growing based on flood fill algorithm. The pavement region will be separated as a result of the growing process.

3.1 Extraction of characteristic window

Small fragment windows, as the seeds, are obtained by characteristic similarity method. Seeds' extraction process is divided into four steps.

Firstly, three candidate windows with size of 10×10 pixels are randomly extracted at the lower edge of image, W_1, W_2, W_3, as shown in Fig. 3. Secondly, candidate windows move along the direction of x-axis continuously, and the grayscale variance of each window is calculated at each moving step. The search process stops until all the variances are less than the threshold σ_{th}^2 . If search fails, a translation along the positive direction of y-axis is taken, and then search continues until succeed. During the translation process, window collision is allowed while overlaying is forbidden.

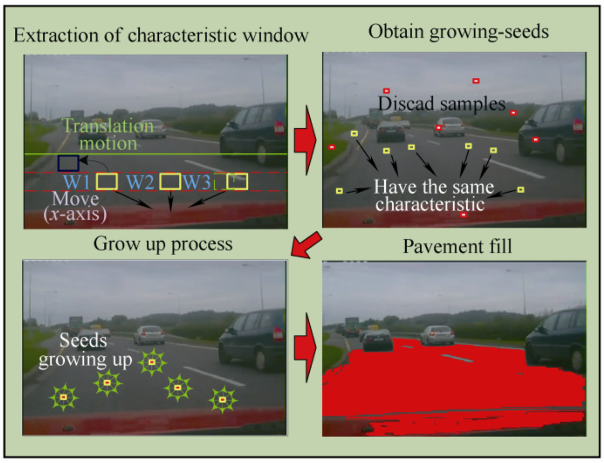


Fig. 3. Schematic diagram of the pavement segmentation

The variance of candidate windows is defined by[22]:

$$\sigma^2 = \left(\frac{1}{N}\sum_{i=0}^{N-1} x_i^2\right) - \left(\frac{1}{N}\sum_{i=0}^{N-1} x_i\right)^2, \tag{1}$$

where x_i is the grayscale of each pixel within the candidate window, N is the number of pixels, and σ^2 is the variance of the candidate window.

Finally, when all the grayscale variances of candidate windows satisfy the predefined threshold, the average grayscale of each window is then calculated. The candidate window with the smallest variance is singled out and verified whether it is less than the grayscale threshold, based on the following equation:

$$\overline{x}_w = \min(\overline{x}_1, \overline{x}_2, \overline{x}_3) \leqslant \overline{x}_{\text{th}}, \tag{2}$$

where $\overline{x}_w$ is the average grayscale of candidate window with the smallest variance, $\overline{x}_{\text{th}}$ is the grayscale threshold, $\overline{x}_1, \overline{x}_2, \overline{x}_3$ is the candidates. If the equation is satisfied, candidate window with the smallest variance is adopted as the characteristic window of pavement, and the parameter $(\overline{x}_w, \sigma_w^2, H_w)$ of reference sample is obtained, H_w is the histogram of the reference sample window. If the equation does not meet, the extraction process returns to the first step and search characteristic windows again.

3.2 Pavement filling

A set of seeds are generated by means of characteristic similarity method. Fragment samples are picked up randomly and characteristic parameters are compared to the reference sample. Fragment windows with similar characteristics are selected as seeds, otherwise discarded.

Feature similarity coefficient(FSC) is defined by

$$d_{\text{correl}}(H_s, H_w) = \frac{\sum_i H_s'(i) \times H_w'(i)}{\sqrt{\sum_i H_s'^2(i) \times H_w'^2(i)}},$$

$$H_k'(i) = H_k(i) - \frac{\sum_j H_k(j)}{N}, \quad k = s, w, \tag{3}$$

where N is the quantity of histogram bin, i and j are the indexes of bin, H_w is the histogram of reference sample, H_s is the histogram of the candidate.

In growing process, the seeds progressively merge the adjacent fragments whose FSC is greater than the merge-threshold. As a consequence of absorbing the adjacent pixels, pavement regions are marked by a specified color, and then the color of pavement area is changed to the specified color. The specified color should be uncommonly used in actual roads(red is used here as shown in Fig. 3). Growing process continues until adjacent fragment windows with similar characteristics cannot be found.

4 Vehicle Detection Method

Contrary to traditional approaches to guess the most suitable vehicle features in advance, the layered vehicle detectors based on boosted cascade classifiers are employed.

4.1 The earlier candidate generation classifier

In rapid candidate generation stage, LBP is used as the feature operator. The feature identification has been used successfully in face recognition, vehicle detection and other object identification area[20–21].

A 9×9-neighborhood scaled LBP operator was employed for vehicle feature description. The comparison-operator for single pixels is calculated using the average grayscale of sub-regions. By this operation, a 9×9-neighbourhood operator is converted to 3×3-neighbourhood one, as described in Fig. 4. Taking the grayscale of center pixel as the threshold, the operator is converted to a binary array. Using the scaled LBP operator, the entire image is represented by a binary string. The histogram of binary string is calculated as a texture descriptor, which is defined by equation:

$$H_i = \sum_{x,y} I\{f_l(x,y)=i\}, \quad i=0,\cdots,n-1, \tag{4}$$

where H is the value of histogram, n is the number of labels, $f_l(x,y)$ is the labeled image and $I\{\Lambda\}$ is defined as follows

$$I\{\Lambda\} = \begin{cases} 1, & \Lambda \text{ is true,} \\ 0, & \Lambda \text{ is false.} \end{cases} \tag{5}$$

The histogram of binary string contains local micro-pattern features, such as edges, texture, spots and flat areas.

LBP detector provides a feature extraction method to train classifiers. A DAB learning method is employed, as described in Table 1, to construct the early vehicle detection classifier.

Table 1. Discrete AdaBoost(DAB)

Input: sequence of N labeled examples $\{(x_1,y_1),\cdots,(x_N,y_N)\}$
distribution D over N examples
weak learning algorithm ***WeakLearn***
integer T specifying number of iterations
Initialize: the weight vector: $w_i^1 = D(i)$ for $i=1,\cdots,N$.
Do for t=1 to T
1. Set
$$p^t = \frac{w^t}{\sum_{i=1}^N w_i^t}$$
2. Train a weak classifier for each feature $h_t : X \to [0,1]$, providing it with distribution of p^t
3. Calculate the error of
$$h_t : \varepsilon_t = \sum_{i=1}^N p_i^t \left| h_t(x_i - y_i) \right|$$
4. Set $\beta_t = \varepsilon_t / (1-\varepsilon_t)$
5. Set the new weight vector to be $w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$

Outputs the hypothesis
$$h_f(x) = \begin{cases} 1, & \text{if } \sum_{t=1}^T (\log\frac{1}{\beta_t}) h_t(x) \geqslant \frac{1}{2}\sum_{t=1}^T \log\frac{1}{\beta_t}, \\ 0, & \text{otherwise.} \end{cases}$$

Most of the redundant information is removed and the most effective features are reserved, by the DAB training algorithm, to construct the cascade weak classifiers for rapid vehicle detection. In order to reduce the missing rate and improve the efficiency, in the stage of candidate generation, weak classifiers are restricted to 4 layers, as descript in Fig. 4. Approximately, 20 candidates remain after the early procedure.

4.2 The later candidate verification classifier

In the previous stage, positions of candidate vehicles are hypothesized. A further verification is needed to eliminate negatives in candidates. Then, two kinds of Haar-like features(non-rotated and rotated Haar-like feature) are used for candidate verification.

Non-rotated features values are calculated by the following function:

$$f(I) = \sum_{m=1,\cdots,N} \varsigma_m \bullet Sum(A_m), \tag{6}$$

and the rotated features values are calculated by:

$$f(I) = \sum_{m=1,\cdots,N} \varsigma_m \bullet RSum(A_m), \tag{7}$$

where N is the quantity of Haar-like features of the image I, m is the index, A_m is a rectangular area, and ς_m is either 0 for black rectangles or 1 for white rectangles, R is the rotation coefficient.

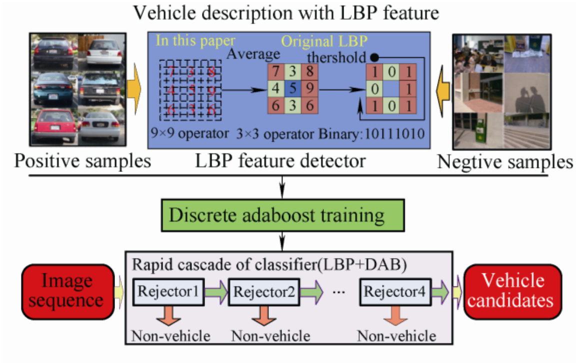


Fig. 4. Classifier of rapid detection trained by DAB learning method of LBP features

Training samples used to construct the robust classifier are images with the size of 32×32 pixels containing large amount of Haar-like features in grayscale space. However, only a few of these are accounting for the vehicle appearances. To this end, it seems proper that, RAB learning method is employed to pick out the efficient features for candidate verification process.

RAB training method, shown in Table 2, is used to construct the robust classifier. An appealing characteristic of this method is that, by using an integral image, the classifier can work both fast and efficiently during object detection procedure.

Table 2. Real AdaBoost(RAB)

Input: $(x_1,y_1),\cdots,(x_M,y_M)$
where $x_i \in X, y_i \in Y = \{-1,+1\}$
Initialize: weight vector: $D(i)=1/m$
Do for t=1 to T
1. Train base learner using distribution D_t
2. Get base classifier $h_t : X \to R$
3. Choose $\alpha_t \in R$
4. Update
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$
Where Z_t is a normalization factor

Outputs the final classifier
$$H(x) = \mathrm{sgn}(\sum_{t=1}^T a_t h_t(x))$$

The training process eventually selects n_f ($n_f \ll N$) subset features from feature space, each of which is associated with a weak classifier. When combined properly, weak classifiers become a strong classifier and can classify the vehicles effectively. The final classifier is expressed as:

$$R(F) = \sum_{i=1}^{n_f} r_i(f_i), \tag{8}$$

where $R(F)$ is the final strong classifier, $r_i(f_i)$ is a decision function on classifier f_i, n_f is the number of weak classifier, it returns a positive (v_i^+) or negative (v_i^-) value according to classification decision, as follows:

$$r_i(f_i) = \begin{cases} v_i^+, & \text{if } f_i \leqslant (\text{or} \geqslant) \text{ threshold}, \\ v_i^-, & \text{otherwise}. \end{cases} \tag{9}$$

To verify candidates, a strong classifier was constructed using a set of weak classifiers with increasing difficulty. Following this strategy, a cascade of classifiers of 8 layers has been developed, as shown in Fig. 5.

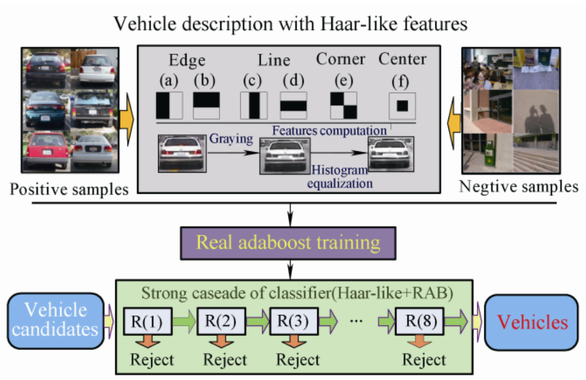


Fig. 5. Verification classifier trained by Haar-like features and RAB

4.3 Implementation

Firstly, pavement area information is used to estimate the potential vehicles' positions. Then, a layered machine learning classifier is applied to detect vehicle positions, as shown in Fig. 6. Through rapid candidate generation and verification, the stable detection results will be obtained.

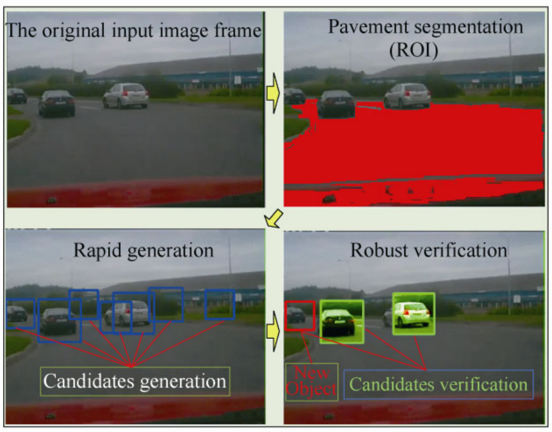


Fig. 6. Sketches details of vehicle detection module

5 Multi-vehicle Tracking Method

Particle filter is employed in tracking module to prevent performance degradation caused by detection errors. The original particle filter is designed to track only one object at each time, thus cannot handle the ever-changing number of objects in tracking procedure. To reach this goal, a multi-vehicle tracking method with target state management method is introduced.

5.1 Target state management

Maintaining tracking of multiple vehicles is carried out by a state management of the tracked targets. In this implementation, there are four states of object: falsealarm, confirmed, undetected and exit, which are managed by credibility counter driven by detection module, as shown in Fig. 7.

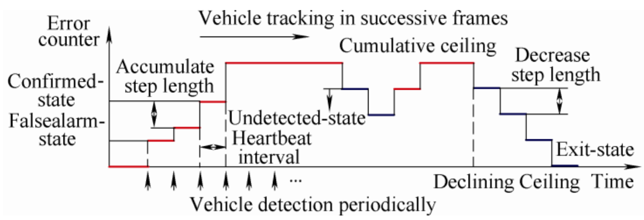


Fig. 7. Credibility counter of target state management

Particle filter algorithm is applied to estimate the probability distribution of the vehicles with confirmed-state and undetected-state. Meanwhile, vehicle detection module inspects the entire pavement area periodically to search vehicles, and then drives the credibility counter.

When a vehicle appears in the field of view, firstly, it is recognized as falsealarm-state, because of the probability of erroneous identification. In the following process, detection module detects it continuously and adjusts the corresponding credibility counter in each heartbeat cycle. Specifically, in the adjustment period, the decrease step is given by a fixed value D_v, while the increase step is determined by the confidence deduced from detection and tracking fusion results, as given by equation:

$$S_t^{(i)} = D^{(i)}(x_t) \bullet P^{(i)}(x_t) \bullet F^{(i)}(x_t), \tag{10}$$

where x_t is the vehicle state at time t and i is the index of tracked vehicles in the current frame. $D^{(i)}(x_t)$ is the vehicle confidence obtained from detection module, $P^{(i)}(x_t)$ is the probability density drawn from tracking module and $F^{(i)}(x_t)$ is the consistency between detection result and tracking result.

Fig. 7 shows the switch process of state management. When a vehicle is in falsealarm-state, the tracking process does not work until the credibility counter is greater than the threshold of confirmed-state. The credibility counter keeps increasing until it reaches the maximum limit.

However, if the previous object cannot be found in the current heartbeat detection cycle, the corresponding credibility counter will be reduced by D_v and switch to undetected-state. Afterwards, vehicle in undetected-state will be tracked until the credibility counter is less than the declining ceiling.

5.2 Algorithm for vehicle individual tracking

A particle filter is introduced for tracking of single vehicle, to solve the general problems of non-linear and non-Gaussian estimation under Markov assumptions. In the proposed approach, color based distribution in HSV space is used as object model to calculate the likelihood. A randomly generated particle sets with scaling size are employed to represent the probability density of vehicle individuals. The method consists of the following steps.

(1) Initialization of particles

The particle of vehicle at time t is constructed by a rectangular bounding box defined by the dynamic model $x_{t,i}^m = [p_{t,i}^m, s_{t,i}^m, h_{t,i}^m]$. Where $m \in \{1, \cdots, M\}$ and M is the quantity of vehicles in the tracking queue, i is the index of the tracked object, $p_{t,i}^n$ determines the center of target position in the image plane, $s_{t,i}^n$ is the scaling coefficient and $h_{t,i}^n$ is color based histogram calculated in HSV color space.

In initialization stage, a random particle set of hypotheses with the quantity of N is generated. Although more promising results can be obtained if the number N is big enough, the computation load would be unacceptable. In a framework of simultaneous detection and tracking, the quantity of particles can be small, since the references are periodically calibrated by the latest results.

(2) Particle transition

Vehicles are assumed to drive on a planar ground. Thus, the range of vehicles in image area can be estimated and restricted by the transition model. A second-order autoregressive dynamics model is adopted, which uses historical data to predict the current status[26]. The transition function is given as follows:

$$X_t = A_1 X_{t-1} + A_2 X_{t-2} + B \bullet N(0, \Lambda), \tag{11}$$

where $X = \{x_{t,i}^n, y_{t,i}^n, s_{t,i}^n\}$ denotes the particle status of image position and scaling, $x_{t,i}^n$ is coordinate of x-axis, $y_{t,i}^n$ is coordinate of y-axis, $s_{t,i}^n$ is particle scaling factor. $\{A_1, A_2, B\}$ is the Autoregressive coefficients, and taking $A_1 = 2.0$, $A_2 = -1.0$, $B = 1.0$. $N(0, \Lambda)$ denotes the Gaussian distribution with zero mean and covariance $\Lambda = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_s^2)$. Here, $\sigma_x^2 = 1.0$, $\sigma_y^2 = 0.5$, and $\sigma_s^2 = 0.01$.

(3) Color based likelihood calculation

Color based histogram distribution in HSV color space is used as object models, since they can achieve robustness against rotation, scaling, illumination changing and partial occlusion. To model colors, a cumulative normalized histogram with three independent channels of hue, saturation and value is used.

The histograms are typically represented in HSV space by $N_H \times N_S + N_V$ ($N_H = N_S = N_V = 9$) bins. The resultant histogram is composed of $N_B = 90$ bins. Pixels with saturation and value greater than V_{th}^S and V_{th}^V fill the first $N_H \times N_S$ bins. Other pixels with colorless feature fill the last N_V value-only bins. Where, $V_{\text{th}}^S = 0.18$, $V_{\text{th}}^V = 0.25$. Color histogram is expressed by:

$$q(n; x_t) = q(x_t),\ n = 1, \cdots, N, \sum_{n=1}^{N} q(n; x_t) = 1, \tag{12}$$

where N is the number of histogram bin, q is the normalized histogram.

In order to calculate the likelihood of particles, reference color histogram is also needed. The HSV color based histogram of the reference vehicle is given by:

$$q^*(n; x_0) = q^*, n = 1, \cdots, N, \sum_{n=1}^{N} q^*(n; x_0) = 1, \tag{13}$$

where q^* is calculated and calibrated in each heartbeat detection cycle using image histogram solving method. The reference image R_t is derived from detection and tracking modules, as given by:

$$R_t = 0.4 D_t + 0.4 T_t + 0.2 R_{t-1}, \tag{14}$$

where D_t is the detection result, T_t is the tracking result, and R_{t-1} is historical information of last heartbeat cycle.

Then, Bhattacharyya coefficient is used to calculate the likelihood between two histograms

$$d^2(x_t, x_0) = 1 - \sum_{n=1}^{N_B} \sqrt{q(n; x_t) * q^*(n; x_0)}, \tag{15}$$

where x_t denotes the image of particles, x_0 denotes the reference image. The weight of the particles is calculated using color based histogram likelihood, as is described by function:

$$p(z_t \mid x_t) \propto \exp(-\lambda d^2(x_t, x_0)), \tag{16}$$

where λ is the distance coefficient, and is set to experimental value of 18.

The more similarity between the tracked vehicle and reference, the greater the weight could be. In this manner, the promising tracked object with best posterior probability density of particle is obtained.

(4) Re-sampling method

The basic idea of resampling is to remove particles with small weight and reproduce great weight ones. Firstly, particles are sorted according to the weights, and then a new set of particles are re-sampled according to the rule of discrete probability distribution $P(\tilde{x}_k^j = x_k^i) = w_k^i$. Where $\{x_k^i\}_{i=0,\cdots,N}$ and $\{\tilde{x}_k^i\}_{i=0,\cdots,N}$ are the particle sequences before and after re-sampling respectively. The newly generated particles are given the equal initialized weights. To maintain the diversity of particles, Gaussian noise is added to the re-sampling process.

6 Experimental Evaluation

In this section, Experiments are conducted to evaluate the accuracy and efficiency of the proposed vehicle identification system.

6.1 Test environment setting

Real road videos were captured with resolution of 640×480 pixels and frame rate of 25 frames per second, in different traffic environments such as expressways, urban roads, suburban roads and rural roads. The image size was further downsized to 320×240 pixels to reduce computation load. The proposed method was evaluated on a windows platform with Intel Pentium(R) dual-core CPU (E5200 frequency 2.5 GHz) and 2 Gb RAM installed.

6.2 Video database

The recorded video databases are mainly composed of two parts: training sets and test sets. The training sets consist of positive subsets and negative subsets. The positive training subsets contain 1276 distinct static images with image size of 32×32 pixels, and include mainly rear-facing samples and a few side-view samples, as shown in Fig. 8. The negative training subsets contain 2234 images of background scene without any vehicles in the image, as shown in Fig. 9.

Fig. 8. Examples of positive training subsets

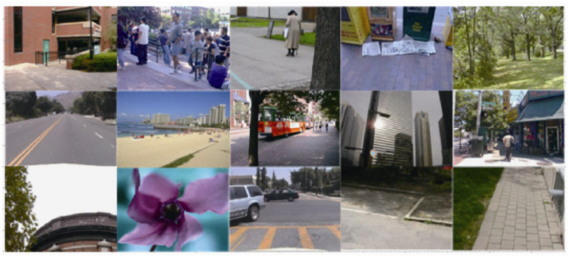


Fig. 9. Examples of negative training subsets

The presented method was evaluated by using test sets of eight video shots lasting 50 min and 75 082 frames in total. The test sets were captured both on the urban road and highway, and the weather conditions are both cloudy and sunny. More than 150 different vehicles were available in test sets.

6.3 Parameter setting

In training stage, each positive image was scaled to the size of 32×32 pixels, which were used as the appearance based training features for LBP feature and Haar-like feature learning. In negative samples, image size was arbitrary.

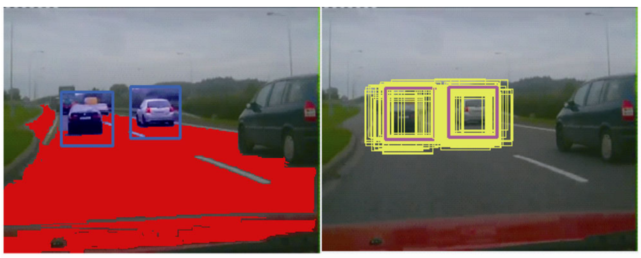
In detection stage, the clustering threshold of similarity measurement was set to 0.8 for the appearance based feature. And the centroid-distance between two clustered vehicle positions was less than 15 pixels.

In tracking stage, 50 particles were used for each tracked vehicle, and the rectangular size of each particle was between 18×15 pixels and 200×180 pixels.

6.4 Performance

Real video test sequences were captured from a CMOS front-mounted camera. Proposed system achieves 91.2% detection rate on a fine day with proper illumination and clear vehicle appearance. However, the recognition rate declines to 85%–90% on a cloudy day due to the smudges and dirt on the windshield.

In the heartbeat detection cycle, as shown in Fig. 10, vehicle detection and tracking methods were implemented in the same video frame. At this moment, with the result fusion of two modules, more credible estimation of vehicle position was obtained.



(a) Detection results (b) Tracking results

Fig. 10. Simultaneous vehicle detection and tracking process for the same image frame

A complementary operation between detection module and tracking module is demonstrated in Fig. 11. The image in the left shows that two vehicles have been successfully detected, and one was missed due to far distance and smudges on the windshield on a cloudy day. However, at the same time, the missed vehicle was still tracked by a particle tracker despite being missed in detection module.

Fig. 12 shows the performance of the presented system at rush hour in the urban traffic environment under different weather condition. With good illumination, all of the front vehicles were detected as shown in Fig. 12(a). However, on a cloudy day, three vehicles were detected and two were missed due to long-distance and poor illumination.

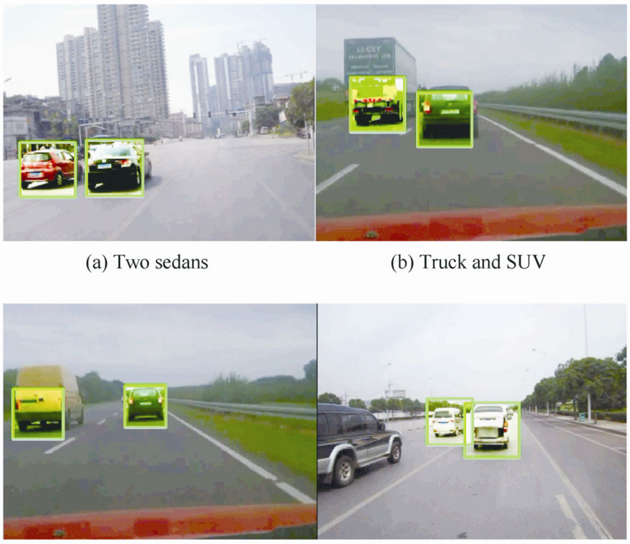
(a) One of vehicles is missed (b) Tracking is still maintained

Fig. 11. A complementary operation between detection module and tracking module

(a) Sunny day at rush hour (b) Cloudy day at rush hour

Fig. 12. System performance under different weather conditions

In these experiments, the presented system exhibited stable adaptability to variety of vehicles, as shown in Fig. 13.



(a) Two sedans (b) Truck and SUV

(c) Special vehicle and SUV (d) Microbuses

Fig. 13. Varieties of vehicles tested by the proposed system

The proposed framework has made full use of the information from different color space, as shown in Table 3, thus improved the performance of the system even if the target vehicles had different color, size, velocity and types.

Table 3. Analysis of the average processing time

Step	Color space	Average time t/ms
Segmentation of pavement area	RGB	2.04
Vehicle detection module	Gray-Level	25.41
Vehicle tracking module	HSV	22.55
Total	–	47.98

6.5 Comparative analysis

The performance criteria include precision, robustness and efficiency. Precision and robustness can be quantified by the true positive rate and false detection rata.

The true positive rate (TPR) is the percentage of vehicles in camera's view that are detected correctly. It is defined as follows:

$$TPR = \frac{\text{true detected vehicles}}{\text{total number of vehicles}}. \tag{17}$$

The false detection rate (FDR) is the proportion of false detected vehicles to totally detected vehicles. FDR is a measure of precision and robustness. FDR is defined as follows:

$$FDR = \frac{\text{false positives}}{\text{true detected vehicles+false positives}}. \tag{18}$$

The proposed algorithms were compared with two typically algorithms as shown in Table 4 using the same test database. As a feature matching method, the active learning based vehicle recognition and tracking system[1] (ALVeRT) is constructed using query and archiving interface with conventional supervised learning method for active learning. While maximum posteriori estimation based vehicle detection method[22](Map-based-method), as a knowledge-based method, employs shadows, textures, symmetry and other characteristic information to recognize vehicle objects.

Table 4. Performance comparison study

Algorithm	Testing database	*TPR* (original)	*FDR* (original)	*TPR* (this paper)	*FDR* (this paper)
ALVeRT[1]	LISA-Q FrontFOV Urban	99.8%	8.5%	91.2%	2.6%
Map-based-method[22]	PETS 2001	95.4%	5.2%	92.7%	3.1%

As shown in Table 4, compared to the other two methods, TPR of the proposed method drops by about 2.7%–8.6%. Using the active learning approach, the ALVeRT system achieves higher detection rate, but it brings cumulative error to classifiers and results in high FDR. The map-based method occasionally suffers from unsteadiness due to dynamic backgrounds, which contribute a high FDR.

The FDR of the proposed system is reduced nearly by half compared to the other two systems. The proposed method is more immune to false positives because the background noise is reduced by pavement segmentation, and moreover, candidates' generation and verification mechanism with a layered machine learning method can ensure reliable detection.

7 Conclusions

A simultaneous multi-vehicle detection and tracking framework using monocular vision technology was proposed to improve efficiency, robustness and accuracy of the on-road vehicle recognition system.

(1) Benefiting from the proposed pavement segmentation method, the unnecessary process time is eliminated, and background noise is reduced.

(2) A novel layered vehicle detection method based on AdaBoost learning algorithm is proposed that combines coarse-search and fine-search together to obtain the optimal target in the heartbeat detection cycle. At the same time, the detector also calibrates the tracking reference for better tracking performance.

(3) An object tracking algorithm based on target state management and particle filter is proposed to successfully maintain multiple vehicles tracking and enter-exit management.

References

[1] SAYANAN S, MOHAN M T. A general active-learning framework for on-road vehicle recognition and tracking[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2010, 11(2): 267–276.

[2] SUN Z, BEBIS G, MILLER R. On-road vehicle detection using evolutionary Gabor filter optimization[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2005, 6(2): 125–137.

[3] TRIVEDI M M, GANDHI T, MCCALL J. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2007, 8(1): 108–120.

[4] JAZAYERI A, CAI Hongyuan, ZHENG Jiangyu. Vehicle detection and tracking in car video based on motion model[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2011, 12(2): 583–595.

[5] WANG C C R, LIEN J J J. Automatic vehicle detection using local features—A statistical approach[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2008, 9(1): 83–96.

[6] WENDER S, DIETMAYER K. 3d vehicle detection using a laser scanner and a video camera[J]. *Intelligent Transport Systems, IET*, 2008, 2(2): 105–112.

[7] SUN Z, BEBIS G, MILLER R. Monocular precrash vehicle detection: features and classifiers[J]. *IEEE Transactions on Image Processing*, 2006, 15(7): 2019–2034.

[8] MO Guoliang, ZHANG Yan, ZHANG Sanyuan, et al. A method of vehicle detection based on SIFT features and boosting classifier[J]. *Journal of Convergence Information Technology*, 2012, 7(12): 328–334.

[9] BAY H, TUYTELAARS T, VAN GOOL L. *Surf: Speeded up robust features*[M]. Computer Vision–ECCV 2006, Graz, Austria, 2006: 404–417.

[10] VIOLA P, JONES M. Robust real-time object detection[J]. *International Journal of Computer Vision*, 2001, 4: 34–47.

[11] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]//*Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, San Diego, USA, 2005, 1: 886–893.

[12] BRADSKI G, KAEHLER A. *Learning OpenCV: Computer vision with the OpenCV library*[M]. O'Reilly Media, Sebastopol, USA, 2008.

[13] KAEMPCHEN N., DIETMAYER K. Fusion of Laserscanner and video for advanced driver assistance systems[C]//*Proc. 11th World Congress on Intelligent Transportation Systems*, Nagoya, Japan, 2004: 1–10.

[14] MATTHEWS N D, AN P E, CHARNLEY D, et al. Vehicle detection and recognition in greyscale imagery[J]. *Control Engineering Practice*, 1996, 4(4): 473–479.

[15] LIAO Shengcai, ZHU Xiangxin, LEI Zhen, et al. Learning multi-scale block local binary patterns for face recognition[M]//*Advances in Biometrics*. Springer Berlin Heidelberg, Berlin, Germany, 2007: 828–837.

[16] OJALA T, PIETIKAINEN M, MAENPAA T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(7): 971–987.

[17] PEREZ P, VERMAAK J, Blake A. Data fusion for visual tracking with particles[J]. *Proceedings of the IEEE*, 2004, 92(3): 495–513.

[18] NEGRI P, CLADY X, HANIF S M, et al. A cascade of boosted generative and discriminative classifiers for vehicle detection[J]. *EURASIP Journal on Advances in Signal Processing*, 2008(2): 1–12.

[19] PONSA D, SERRAT J, LóPEZ A M. On-board image-based vehicle detection and tracking[J]. *Transactions of the Institute of Measurement and Control*, 2011, 33(7): 783–805.

[20] WANG Shenzheng, LEE H J. Detection and recognition of license plate characters with different appearances[C]//*Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Shanghai, China, 2003: 979–984.

[21] ALONSO J D, ROS VIDAL E, ROTTER A, et al. Lane-change decision aid system based on motion-driven vehicle tracking[J]. *IEEE Transactions on Vehicular Technology*, 2008, 57(5): 2736–2746.

[22] ZHOU Yong. *Several key problem research of the intelligent vehicle*[D]. Shanghai: Shanghai Jiao Tong University, 2007: 91–92. (in Chinese)

Biographical notes

WANG Ke, born in 1984, is currently a lecturer at *College of Automobile Engineering, Chongqing University, China*. He received his PhD degree in mechanical engineering from *Hunan University, China*, in 2013. His research interests include active safety, intelligent vehicle and computer version.
E-mail: yeswangke@163.com

HUANG Zhi, born in 1977, is currently an associate professor at *State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, China*. He received his PhD degree from *Hunan University, China*, in 2006. His research interests include automobile active safety technology, intelligent vehicle and. anti-lock braking system.
E-mail: huangzhi@hnu.edu.cn

ZHONG Zhihua, born in 1962, an expert in automotive engineering, was elected as a member of the *Chinese Academy of Engineering* in 2005. He is currently a professor and a PhD candidate supervisor at *State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, China*. His research focuses on the vehicle safety technology, shaping technology of the auto body.