Video-based multiclass vehicle detection and tracking

Zhiming Qian, Hongxing Shi , Jiakuan Yang and Lianxin Duan

Chuxiong Normal University
Chuxiong, 675000, China

Abstract
This paper presents a real time multiclass vehicle detection and tracking system. The system uses a combination of machine learning and feature analysis to detect and track the vehicles on the road. Multiclass SVM and PCA methods are utilized to create multiclass training samples. The online classifiers are trained using these samples to achieve detection and classification of vehicles in video sequences of traffic scenes. The detection results provide the system used for tracking. Each class vehicle is tracked by SIFT method. The system combines the advantages of both multiclass detection and tracking in a single framework. Experimental results from highway scenes are provided which demonstrate the effectiveness of the method.
Keywords: *Vehicle detection, Vehicle tracking, Online learning, Feature analysis.*

1. Introduction

Video based intelligent transportation systems (ITS) are getting large attention as an attractive field, not only because they are easy to install and operate, but also because they have the potential to provide a much richer description about vehicle. As the basic parts, detection and tracking of vehicle is a fundamental problem in ITS. For this task, we need to first detect the vehicle and segment them from the video images, and then track them across different frames while maintaining the correct identities.

Robust detection and tracking of vehicles on the road based on video is a challenging problem. Roads are dynamic environments, with the illumination and background changes. The sizes and the locations of vehicles on the road are diverse. There is high variability in the appearance of vehicles with viewpoint, illumination, and possible articulation. Moreover, partial occlusion of vehicles of interest by other vehicles or objects on the road is also an important factor influencing detection and tracking.

For the last two decades researchers have spend quality time to develop different methods that can be applied in the field of video based vehicle detection and tracking [1-3].

In the following section, we will present a brief overview of recent related works in video vehicle detection and tracking.

Video vehicle detection is a process of detection the presence or absence of a vehicle in the sequences. The result of detection is used as initialization process for tracking. There are four main approaches to detect vehicle regions, they are：

1. Frame differencing method [4,5]: this method detects moving vehicle regions by subtracting two consecutive image frames in the image sequence. It works well in case of uniform illumination conditions, otherwise it creates non-vehicular region and also frame differencing method does not work well if the time interval between the frames being subtracted is too large.

2. Background subtraction method [6,7]: this method is one of the widely used methods to detect moving vehicle regions. It subtracts the generated background image from the input image frame to detect the moving vehicle regions. This difference image is then thresholded to extract the vehicle regions. The problem with the stored background frame is that they are not adaptive to the environment changes which may create non-existent vehicle regions and also works for stationary background.

3. Feature based method [8,9]: this method made use of sub-features to detect moving vehicle regions. These features are grouped by analyzing their motion between consecutive frames. Thus a group of features segments a moving vehicle from the background. The advantages of this method is that the problem of occlusion between the vehicle regions can be handled well, the feature based methods have less computational complexity compared to background subtraction method. But the disadvantage is that if the features are not grouped accurately, then there may be failure in detecting vehicles correctly.

4. Motion based method [10,11]: this method assumes that vehicles tends to move in a consistent direction over time and that foreground motion has different saliency. It is less sensitive to noise and very effecting on small moving objects, but the disadvantage is that calculation of motion information consumes time, and it can not be used to

detect static obstacles which can represent a big threat to detection task.

After vehicle detection, ITS will carry out the task of vehicle tracking. Vehicle tracking is a process that generates the trajectory of the vehicle over time by locating its position in every frame of the video sequences. The existing tracking approaches may be classified into four major categories:

1. Region based method [12,13]: this method subtracts image frame containing vehicles from the background frame which is then further processed to obtain vehicle regions (blobs). Then these vehicle regions are tracked. It can work well in free flowing traffic conditions, but the disadvantage is that it has difficulty in handling shadows and occlusion.

2. Active contour based method [14,15]: this method represents vehicle by bounding contour of the object and dynamically update it during the tracking. The advantage of active contour tracking over region-based tracking is the reduced computational complexity. But the disadvantage of the method is their inability to accurately track the occluded vehicles and tracking need to be initialized on each vehicle separately to handle occlusion better.

3. Feature based method [16,17]: this method extracts suitable features from the vehicle regions and these features are processed to track the vehicles correctly. The method has low complexity and also can handle occlusions well. The disadvantage is the recognition rate of vehicles using tow-dimensional image features is low, and the problem that which set of sub features belong to one object is complex.

4. Model based method [18,19]: this method tracks vehicle by matching a projected model to the image data. The advantages of model based vehicle tracking is it is robust to interference between nearby images and also be applied to vehicle classification. But the method has high computational cost and they need detailed geometric object model to achieve high tracking accuracy.

Above approaches can effectively accomplish detection and tracking tasks. However, these approaches need more system computation and have certain application conditions. In order to reduce calculation time and to improve system efficiency, learning-based approaches have been adopted by many researchers to detect and track video vehicles efficiently.

Based on how the learning takes place over time, the learning-based approaches can be categorized as offline learning and online learning. Offline learning requires all the training data to be available from the beginning of learning process. These kind of approaches try to produce results which are consistent with all the collected data samples. On the other hand, online learning requires the training data to arrive sequentially over time and additionally. These kind of approaches provide the machine with the ability to learn continuously and adapt all the time to its inputs.

There are some offline learning methods to detect and track video vehicle which have obtained good results. Sun et al [20] employed support vector machines to learn Haar wavelet features for vehicle detection. Junior and Nunes [21] used multilayer feed forward neural network based approach to detect vehicles. Khammari et al [22] utilized a gradient analysis and Adaboost classification to accomplish rear vehicle detection. Negri et al [23] presented an algorithm for the on-board vision vehicle detection problem using a cascade of boosted classifiers. Withopf and Jahne [24] presented a learning algorithm for real-time vehicle tracking in video sequences which uses an improvement of a feature selection method. Chen et al [25] proposed a framework for spatiotemporal vehicle tracking using unsupervised learning-based segmentation and object tracking.

In offline learning methods, large amount of training samples could be required for obtaining a generic detector. The quality and quantity of the training samples directly determine the detection and tracking performance of the system. In order to resolve the problem, online learning methods have been an area of great recent interest in the vehicle detection and tracking. Nguyen et al [26] employed online boosting algorithm for car detection from high resolution aerial images. Chang and Cho [27] presented a real time vision based vehicle detection system using an online Adaboost algorithm. Sivaraman and Trivedi [28] proposed a general active-learning framework for on-road vehicle detection and tracking.

In real world, the vehicle type is various. Comparing the strategy that all vehicles are categorized as single class, multiclass vehicle detection and tracking have great practical significance and applicable value great practical importance. In this paper, a framework for video multiclass vehicles detection and tracking is introduced. The proposed framework has the following characteristics: (1)It has multiclass vehicles detection ability; (2)It can be update based on new training samples which come from video images to adapt new environment; (3)It can track vehicles accurately in real-time environment. The proposed framework in this paper has been validated with video vehicle sequences from real-world traffic scenes.

2. The proposed framework

2.1 Overall structure

Given an input of a video sequence taken from roadway vehicles, system first outputs the types and locations of the vehicles in the images, then a feature information description of the detected vehicles is obtained, and finally this description is used to match the detected vehicles in the next frame. The framework contains three main processes: vehicle classification, vehicle detection, and vehicle tracking. In the vehicle classification process, using offline learning to create multiclass classifier, once the created multiclass classifier recognizes a potential vehicle in an image, the system generates a train sample for a corresponding vehicle detector. The vehicle detectors were then trained by online learning based on these generated train samples. In the vehicle detection process, using the trained vehicle detectors to classify and locate vehicles from video sequence, while at the same time the vehicle detectors will continue to be trained to improve detection ability. In the vehicle tracking process, the tracker analyzes the feature information of the detected vehicles in the previous image frames and matches the feature information of the detected vehicles in the current image. If the matching result is accurate, the tracker outputs the label information for the detected vehicle. A general overview of the system framework can be seen in Fig. 1.

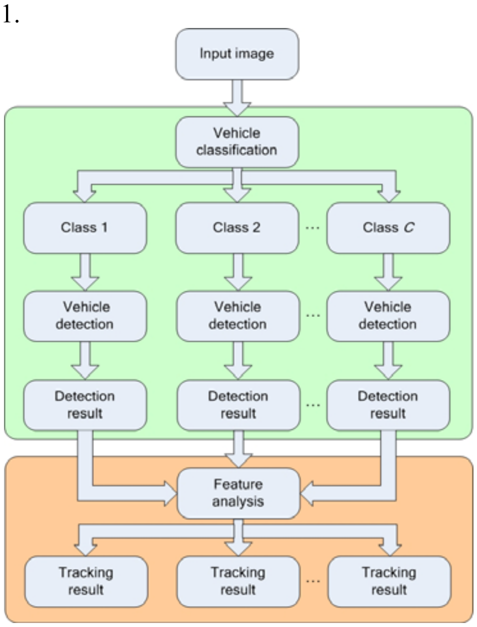


Fig. 1 Overview of our framework.

2.2 Vehicle classification

In order to achieve vehicle classification task, multiclass SVM is employed to our framework. The SVM has been introduced as one of the most efficient learning algorithms in computer vision. While many challenging classification problems are inherently multiclass, the original SVM is only able to solve binary classification problems. Due to significant appearance variation across different vehicles, a direct solution of vehicle classification using single SVM module should be avoided. The better method is to use a combination of several binary SVM classifiers to classify vehicles. The "one against one" and the "one against all" are the two most popular methods for multiclass SVM. Hsu and Lin [29] had compared the performance of the two methods with a large set of different problems. Experiments show that the "one against one" method may be more suitable for practical use.

To be useful, the task of vehicle classification should categorize vehicles into a sufficiently large number of classes. However as the number of class increases, the processing time required also increases. Therefore, a simple classification method is needed which can quickly categorize vehicles at a coarse level. Based on the application, further classification can be done. In the paper, we use the "one against one" method in the LibSVM [30] to learn Haar wavelet features for vehicle classification.

The one-against-one method constructs an SVM for every pair of classes by training it to discriminate the two classes. If k is the number of classes, then $k(k\text{-}1)/2$ classifiers are constructed and each one trains data from two classes. The decision function for class pair ij is defined by

$$f_{ij}(x) = (\phi(x) \cdot w^{ij}) + b^{ij} \tag{1}$$

It is found by solving the following optimization problem:

$$\min \frac{1}{2}\left\|w^{ij}\right\|^2 + C\sum_n \xi_n^{ij} \tag{2}$$

$$\begin{cases} \phi(x_n)\cdot w^{ij} + b^{ij} \geq 1 - \xi_t^{ij}; & \xi_t^{ij} \geq 0, \quad x_t \text{ in the ith class} \\ \phi(x_n)\cdot w^{ij} + b^{ij} \leq \xi_t^{ij} - 1; & \xi_t^{ij} \geq 0, \quad x_t \text{ in the jth class} \end{cases} \tag{3}$$

Finally, the "max wins" voting strategy is used to determine the class of a test pattern in this approach. Fig. 2 shows the flowchart of vehicle classification.

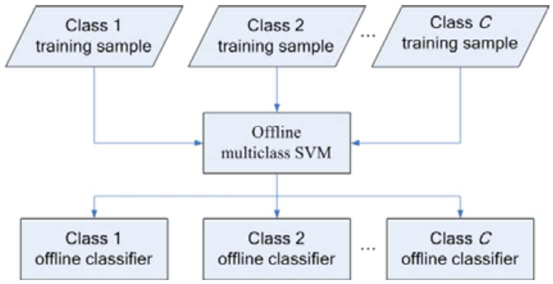


Fig. 2 Flowchart of offline classification.

2.3 Sample creation

Due to various complexities, the classification results using multiclass SVM may be inconsistent with the expected results. However, the classification results are very important for online learning, and its accuracy can directly affects the performance of the detection system. In order to eliminate these false results, we consider using eigenvehicle method to filter the classification results as post processing. Eigenvehicle method is based on the well-known method eigenface [31]. However as the method is used for vehicle detection we named it as eigenvehicle method. The main idea is to decompose vehicle images into a small set of characteristics feature images called eigenvehicle, which may be thought of as the principal components of the original images. The eigenvehicle function as the orthogonal basis vectors of a subspace called vehiclespace. For each class of vehicle, we prepare M=50 vehicle images as the train set. Each image in the train set is transformed into a vector of size N and placed into the set:

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \ldots\ldots, \Gamma_M\} \tag{4}$$

The average matrix is calculated, then subtracted from the original samples and the result stored in the variable Φ_i:

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \tag{5}$$

$$\Phi_i = \Gamma_i - \Psi \tag{6}$$

In the next step the covariance matrix C is calculated according to

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \tag{7}$$

Calculate the eigenvectors and eigenvalues of the covariance matrix:

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T\Phi_n)^2 \tag{8}$$

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & l = k \\ 0 & otherwise \end{cases} \tag{9}$$

Finally, the eigenvehicle will be obtained

$$L = A^T AL_{nm} = \Phi_m^T \Phi_n \tag{10}$$

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad l = 1, \ldots\ldots, M \tag{11}$$

where L is a $M \times M$ matrix, v are M eigenvectors of L.

While a new sample detected by multiclass SVM coming into, it is transformed into its eigenvehicle components. First we compare sample image with mean image of the same class and multiply their difference with each eigenvector of the L matrix. Each value would represent a weight ω and would be saved on a vector.

$$\omega_k = u_k^T(\Gamma - \Psi) \qquad l = 1, \ldots\ldots, M \tag{12}$$

$$\Omega_{new}^T = [\omega_1, \omega_2, \ldots\ldots, \omega_M] \tag{13}$$

Calculate the average Euclidean distance of between the new sample and all the eigenvehicle of the same class. If the value D is bellow an established threshold θ, the input sample is consider to belong to a vehicle image of the corresponding class.

$$D = \frac{1}{M}\sum_{k=1}^{M}\left\|\Omega_{new}^T - \Omega_k\right\| \tag{14}$$

2.4 Online learning

With video sequences as input, a series of training samples are collected by the system and then fed into the boosting learning algorithm. Boosting is one of the mostly applied methods in vehicle detection. Boosting for vehicle detection as described in the previous section most works offline. Hence, all training samples must be given in advance, which is not the case for vehicle detection in video environment.

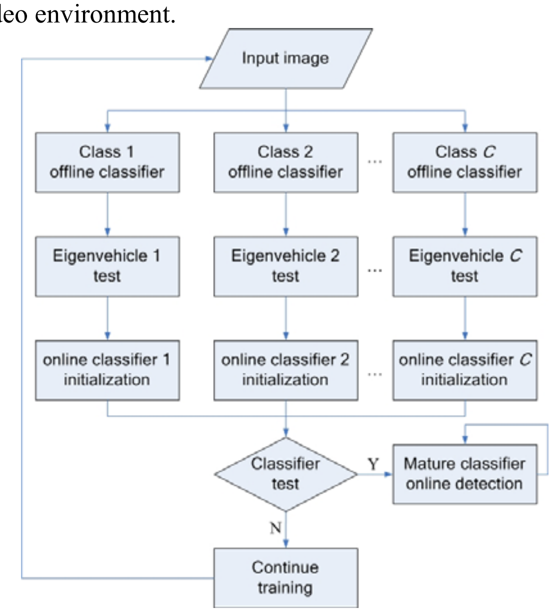


Fig. 3 Flowchart of online detection.

Since for online learning each training sample is discarded directly after an update all steps have to be online. In this paper, we select Haar-like features as the weak classifier, and use Grabner et al's [32] online boosting method creating vehicle detector. The main steps of online learning are briefly described below:

A selector $s_n(x)$ can be considered a set of w weak classifiers $\{h_1(x),\ldots,h_w(x)\}$ that are related to a subset of features $F_n=\{f_1,\ldots,f_w(x)\}$, where F is the full feature pool. At each time the selector $s_n(x)$ selects the best weak hypothesis according to the estimated training error.

To start the learning process a fixed set of n selectors $s_1,\ldots,s_n$ is initialized randomly. Whenever a new training sample (x, y) arrives the selectors are updated. These updates are performed with respect to the importance weight λ of the current sample, which is initialized with λ= 1.

To update the selector s_n first all weak classifiers $h_{n,m}(x)$ are estimated by evaluating the feature $f_{n,m}$ on the sample image x and the corresponding errors:

$$\varepsilon_{n,m}=\frac{\lambda_{n,m}^{wrong}}{\lambda_{n,m}^{corr}+\lambda_{n,m}^{wrong}} \quad (15)$$

are computed. The weights, $\lambda_{n,m}^{corr}$ and $\lambda_{n,m}^{wrong}$ are estimated from the correctly and wrongly classified examples seen so far. Then, the selector s_n selects the weak classifier h_{n,m^+} with the smallest error $\varepsilon_n=\varepsilon_{n,m^+}$, where $m^+=\mathrm{argmin}_m(\varepsilon_{n,m})$:

$$s_n(x)=h_{n,m^+}(x) \quad (16)$$

According to the error ε_n the voting weight α_n and the importance weight λ are updated:

$$\alpha_n=\frac{1}{2}\ln\left(\frac{1-\varepsilon_n}{\varepsilon_n}\right) \quad (17)$$

$$\lambda=\begin{cases}\lambda\dfrac{1}{2(1-\varepsilon_n)} & s_n(x)=y\\ \lambda\dfrac{1}{2\varepsilon_n} & otherwise\end{cases} \quad (18)$$

The importance weight λ is passed to the next selector s_{n+1}. In order to increase the diversity of the classifier pool F_n and to adapt to changes in the environment the worst weak classifier h_{n,m^-}, where $m^-=\mathrm{argmax}_m(\varepsilon_{n,m})$, is replaced by a classifier randomly chosen from the feature pool F. Finally, a strong classifier is computed by a linear combination of N selectors:

$$H(x)=sign\left(\sum_{n=1}^{N}\alpha_n.s_n(x)\right) \quad (19)$$

After all online classifiers are constructed, we will obtain C different vehicle classifiers. When a new image entering, it will be analyzed use these classifiers based on the "max wins" voting strategy, so that achieve the task of vehicle detection and classification.

$$R=\max_i sign(H_i(x)) \qquad i=1,\ldots\ldots,C \quad (20)$$

Fig. 3 shows the flowchart of online detection.

2.5 SIFT feature analysis

SIFT(Scale Invariant Feature Transform)is a well-established local feature descriptors, which was proposed in 1999 by Lowe [33]. Duo to SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes, it has been widely applied to object tracking and image matching. For multiclass vehicle tracking, we need a kind of feature which can describe different vehicles accurately, the SIFT feature is very suitable in the circumstance. The SIFT algorithm includes four steps: scale-space extrema detection, feature point localization, orientation assignment and generation of feature point descriptors. Main process is as follows:

Interest points for SIFT features correspond to local extrema of difference-of-Gaussian filters at different scales. Given a Gaussian-blurred image described as the formula

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \quad (21)$$

where L is the scale space of an 2D image, $I(x,y)$ is the gray value of input image in the coordinates (x,y), $G(x,y,\sigma)$is a variable scale Gaussian, whose result of convolving an image with a difference-of-Gaussian filter is given by

$$D(x,y,\sigma)=L(x,y,k\sigma)-L(x,y,\sigma) \quad (22)$$

which is just be different from the Gaussian-blurred images at scales σ and $k\sigma$. Interest points are identified as local maxima or minima of the DoG images across scales. Each pixel in the DoG images is compared to its 8 neighbors at the same scale, plus the 9 corresponding neighbors at neighboring scales. If the pixel is a local maximum or minimum, it is selected as a candidate feature point. Remove the low contrast candidate points and eliminated the edge response, then use Hessian matrix to compute the principal curvatures and eliminate these feature points that have a ratio between the principal curvatures greater than the ratio.

Finally, an orientation histogram was formed from the gradient orientations of sample points within a 4×4 region with 8 orientations around the feature point in order to get an orientation assignment. So the descriptor of SIFT that was used is 4× 4 × 8 = 128 dimensions.

IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013
ISSN (Print): 1694-0784 | ISSN (Online): 1694-0814
www.IJCSI.org

2.6 Feature matching and updating

For each vehicle detected from multiclass detection framework, extract SIFT feature and establish vehicle information database (VID). The VID consists of four parts: vehicle class, vehicle number, vehicle location (rectangle coordinates) and SIFT feature point descriptor (feature priority, feature point coordinate, orientation and scale), each vehicle detected from multiclass detection framework is tracked in a new video frame sequences by separately comparing its feature point with the same class vehicle from the VID. The Euclidean distance is introduced as a similarity measurement of feature characters.

Suppose N_i as the feature number of the current vehicle matching the ith vehicle of the VID, N as the total feature number of the current vehicle, the matching rate between the current vehicle and the ith vehicle of the VID can be defined as $P_i=N_i/N$. Set the threshold T for the matching parameters. When P_i is greater than T, the current vehicle is considered equivalent to matching the ith vehicle. Supposing that M_j is the number of the jth class vehicle of the VID, $\{P_{ij}(j=1,\ldots,M_j)\}$ is the matching results of the current vehicle and all vehicles of the VID with the same class, and n is the number of elements in the set $\{P_{ij}\,|P_{ij}>T, j=1, \ldots,M_j\}$. When n=1, the ith vehicle is matching with the jth vehicle of the VID with the same class; when n>1, we select max(p_i) as the matching result.

The VID stores the data of vehicle which appears in the recent video sequences. It needs to be updated after one frame, input the current vehicle data and delete the data of the long term unmatched vehicle. We set a feature priority for each feature point of the VID in the vehicle information update process.

Suppose R_{ij} as the feature priority of the jth feature point of the ith vehicle, the specific update process is as following:

(1) Add new vehicle: if the current vehicle is not matching all the vehicle of VID with the same class, this vehicle will be considered as a new vehicle, add its information into the VID, and set its feature priority of all feature points $R=R_{\max}$.

(2) Update feature priority: if the current vehicle matches the ith vehicle of the VID, the information of the ith vehicle will be update, set the feature priority of these matching feature points $R_{ij}= R_{\max}$, and use new coordinate of these matching feature points to replace original coordinate. In addition, the feature priority of unmatched feature points between the current vehicle and the ith vehicle is replaced with $R_{ij} = R_{ij}$ -1, the new feature points of unmatched feature is added into the VID. After all matches of the current frame are finished, if there are no matching vehicles to be found from the VID, all the feature priority of these vehicles will be replace with $R_{ij} = R_{ij}$ -1. When a frame image is completely processed, the feature point whose feature priority is equal to zero will be removed from the VID.

(3) Delete vehicle: When a frame image is completely processed, the vehicle whose feature priority of feature point meets the following condition will be deleted from the VID.

$$R_1 + R_2 + \ldots + R_{All} < \theta \tag{23}$$

Fig. 4 shows the flowchart of vehicle tracking.

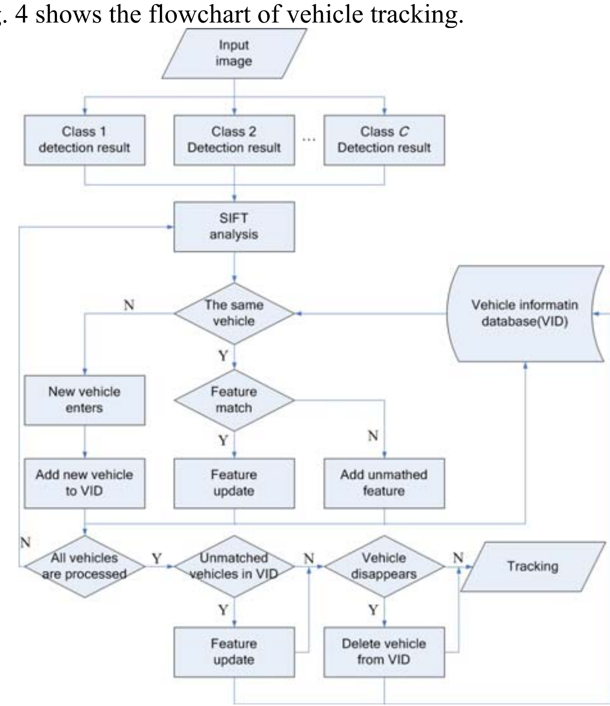


Fig. 4 Flowchart of vehicle tracking.

3. Experiment

We consider the samples from a profile viewpoint for vehicles, and all video sequences which are achieved a frame rate of about 20 fps were generated by shooting around Chuxiong city under highways conditions. All our experiments shown below on a standard PC (Intel Core2 Duo E7500 2.93GHz with 2 GB RAM). The strong classifier consists of 50 selectors and the shared feature pool provides 250 weak classifiers. Set the threshold θ=0.2, the number of class C=4 (motorcycle, bus, truck, and car).

In the training phrase, the data set is the image segmentation data, where each class is a vehicle type collected from a 32×16 region of a vehicle image. The training set consists of 500 samples per class. Some training images are shown in Fig. 5. In the test phrase, the data set is the video sequences, which consists of more than 1 hour of RGB video taken on city highways during the day. The test is divided into two parts, namely the detection test and the tracking test.

Fig. 5 A subset of the training samples for the four classes.

In the detection test, if the classifiers obtain detection result which gives the desired location and classification, the result will be considered to include in the detection rate; if all the classifiers do not obtain detection results or the detection results give the incorrect classification, the detection result will be considered to include in the error rate. The online training result as shown in Fig. 6, the result indicates that, with the increasing of sample size, detection rate increases continuously and finally, it fluctuates smoothly in some ranges. We use the classifier with half hour of training as the final vehicle classifier. In order to evaluate performance of the proposed method, we make a comparison of detection rate and error rate with and offline boosting classifiers. Establish a classifier for each vehicle class using 1200 positive samples and 1500 negative samples, and use the same dataset to test two methods. The experimental results are shown in Table 1. It clearly shows that our method performs better than the other method. More significantly, we create online multiclass classifiers which are suitable for video sequence with small training samples. Some detection results in the video sequences are shown in Fig. 7.

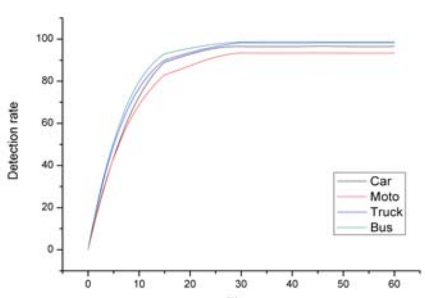
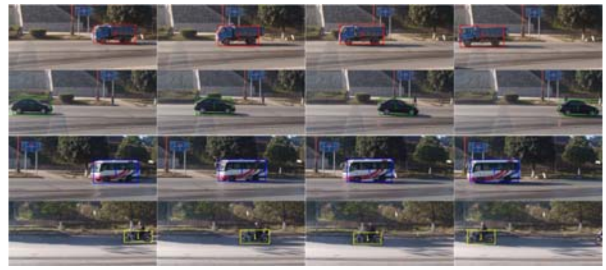
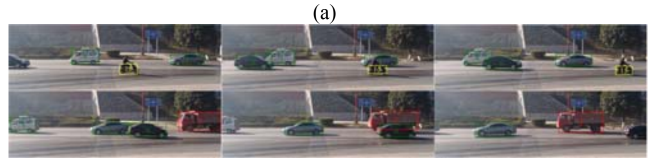


Fig. 6 Detection rate versus the time.



(a)



(b)

Fig. 7 (a)Single class detection results in the experimental sequences. (b) Multiclass detection results in the experimental sequences.

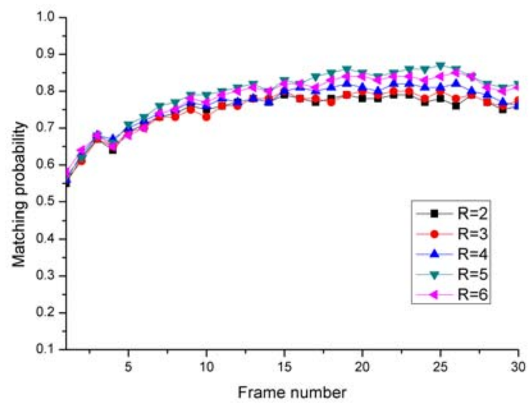


Fig. 8 Influence of R_{max} on matching probability.

In the tracking test, if the classifiers obtain detection result which gives the desired location and identifier, the result will be considered as the correct tracking in current frame, otherwise the result will be considered as the incorrect tracking in current frame. Since there are no suitble methods to compare the multiclass tracking effect, we just test our method on test data. Table 2 shows the tracking results for our method.

Our research also shows the performance of matching algorithm when the parameters R_{max} takes different values. The experimental results as shown in Fig. 8. It shows that while the SIFT features of vehicle were progressively increased with the vehicle packs with the target area between the first and ten frames. Between the ten and twenty frames, the matching algorithm achieves stability while the vehicle appears utterly. A proper value for the parameter R_{max} depends on the scene being modeled. In case of a simple scene, a small value for R_{max} is sufficient.

For complex scenes, more feature information is needed to match the vehicles. The proximity value $R_{\max}$ for feature matching is easy to find by experimenting with different values. Values between 4 and 5 gave good results for all of our test sequences. It should be noticed that the bigger the value of $R_{\max}$, the slower the processing, and the greater the memory requirements.

Table 1: Comparison of detection results of two methods.

Class	Offline boosting detection rate	Offline boosting error rate	Proposed method detection rate	Proposed method error rate
Motocycle	71%	36%	92%	13%
Bus	85%	21%	98%	8%
Truck	78%	31%	96%	10%
Car	82%	27%	95%	12%

Table 2: Tracking results on video sequences.

Class	Tracked number	Vehicles not tracked	Average number of frames during tracking
Motocycle	71	6	35
Bus	85	3	46
Truck	78	2	42
Car	82	4	31

4. Conclusion

We have proposed a real-time vision framework that detects and tracks multiclass vehicles in video sequences. The method by learning a small number of labeled offline samples and a large number of unlabeled online samples to establish the vehicle classifier, and by analyzing the SIFT feature of detected vehicles to achieve vehicle tracking. The framework is able to run in real time with simple, low-cost hardware. Our experimental results demonstrate effective, multiclass vehicle detection and tracking in real traffic environments by applying the proposed framework. If new classes of vehicles or unfamiliar environments are encountered, the proposed framework can adapt itself to the changes and detect vehicles successfully.

Acknowledgements

This work was supported by the Natural Science Foundation of Yunnan Province, China (No. 2011FZ187).

Reference

[1] Z. H. Sun, G. Bebi, R. Miller, On-road vehicle detection: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (5) (2006) 694–711.

[2] M. M. Trivedi, T. Gandhi, J. McCall, Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety, IEEE Transactions on Intelligent Transportation Systems 8 (1) (2007) 108–120.

[3] B. T. Morris, M. M. Trivedi, Learning, modeling, and classification of vehicle track patterns from live video, IEEE Transactions on Intelligent Transportation Systems 9 (3) (2008) 425–437.

[4] D. Dailey, F.W. Cathy, S. Pumrin, An algorithm to estimate mean traffic speed using uncalibrated cameras, IEEE Transactions on Intelligent Transportation Systems 1 (2) (2000) 98–107.

[5] T. N. Schoepflin, D. J. Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation, IEEE Transactions on Intelligent Transportation Systems 4 (2) (2003) 90–98.

[6] C.Wern, A. Azarbayejani, T. Darrel, A. Petland, Pfinder, real-time tracking of human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.

[7] X. Gao, T. Boult, F. Coetzee, V. Ramesh, Error analysis of background adaption, IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 503–510.

[8] D. Koller, K. Danilidis, H. H. Nagel, Model-based object tracking in monocular image sequences of road traffic scenes, International Journal of Computer Vision 10 (3) (1993) 257-281.

[9] S. M. Smith, J. M. Brady, ASSET-2: real-time motion segmentation and shape tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) 814-820.

[10] A. Bainhridge-Smith, R. G. Lane, Deremining optical flow using a differential method, Image and Vision Computing 15 (1) (1997) 11-22.

[11] L. Wilson, Detecting salient motion by accumulating directionally-consistent flow, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 774–780.

[12] D. J. Dailey, F. W. Cathy, S. Pumrin, An algorithm to estimate mean traffic speed using uncalibrated cameras, IEEE Transactions on Intelligent Transportation Systems 1 (2) (2000) 98– 107.

[13] S. Gupte, O. Masoud, R. F. K. Martin, N. P. Papanikolopoulos, Detection and classification of vehicles, IEEE Transactions on Intelligent Transportation Systems 3 (1) (2002) 37-47.

[14] 14. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russell, Toward robust automatic traffic scene analysis in real-time, IEEE Conference on Pattern Recognition, 1994, pp. 126–131.

[15] N. Paragios, R. Deriche, Geodesic active contours and level sets for the detection and tracking of moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (3) (2000) 266–280.

[16] L. Huang, M. Barth, Real-time multi-vehicle tracking based on feature detection and color probability model, IEEE Conference on Intelligent Vehicles Symposium, 2010, pp. 981–986.

[17] B. Coifman, D. Beymerb, P. McLauchlan, J. Malik, A real-time computer vision system for vehicle tracking and traffic surveillance, Transportation Research Part C: Emerging Technologies 6 (4) (1998) 271–288.

[18] M. Haag and H. Nagel, Combination of edge element and optical flow estimate for 3D model-based vehicle tracking

in traffic image sequences, International Journal of Computer Vision 35(3) (1999) 295–319.
[19] T. N. Tan, K. D. Baker, Efficient image gradient based vehicle localization, IEEE Transactions on Image Processing 9(8) (2000) 1343–1356.
[20] Z. Sun, G. B. D. Dimeo, A real-time precrash vehicle detection system, IEEE Conference on Application of Computer Vision, 2002, pp. 171–176.
[21] O. Ludwig, U. Nunes, Improving the generalization properties of neural networks: an application to vehicle detection, IEEE Conference on Intelligent Transportation Systems, 2008, pp. 310–315.
[22] A. Khammari, Vehicle detection combining gradient analysis and adaboost classification, IEEE Conference on Intelligent Transportation Systems, 2005, pp. 66–71.
[23] P. Negri, X. Clady, S. Hanif, L. Prevost, A cascade of boosted generative and discriminative classifiers for vehicle detection, EURASIP Journal on Advances in Signal Processing 2008 (2002) 1-12.
[24] D. Withopf, B. Jahne, Learning algorithm for real-time vehicle tracking, IEEE Conference on Intelligent Transportation Systems, 2006, pp. 516–521.
[25] S. C. Chen, Spatiotemporal vehicle tracking: the use of unsupervised learning-based segmentation and object tracking, IEEE Robotics and Automation Magazine 12(1) (2005) 50–58.
[26] T. Nguyen, H. Grabner, B. Gruber, On-line boosting for car detection from aerial, IEEE Conference on Research, Innovation and Vision for the Future, 2007, pp. 87–95.
[27] W. C. Chang, C. W. Cho, Online boosting for vehicle detection, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 40(3) (2010) 892-902.
[28] S. Sivaraman, M. M. Trivedi, A general active-learning framework for on-road vehicle recognition and tracking, IEEE Transactions on Intelligent Transportation Systems 11 (2) (2010) 267–276.
[29] C. W. Hsu, C. J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415-425.
[30] C. C. Chang, C. J. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 1–39.
[31] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.
[32] H. Grabner, H. Bischof, On-line boosting and vision, IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 260–267.
[33] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91-110.

Zhiming Qian received a Master Degree in Engineering from Yunnan University, Kunming, China, in 2009. He is now an associate professor at Chuxiong Normal University, Chuxiong, China. His current areas of interest include image processing and pattern recognition.

Hongxing Shi received a Master Degree in Science from Yunnan University, Kunming, China, in 2007. He is now an associate professor at Chuxiong Normal University.

Jiakuan Yang received the B.E. degree from Yunnan Normal University, Kunming, China, in 2003. Now he is a lecturer at Chuxiong Normal University.

Lianxin Duan received the B.E. degree from Yunnan Normal University, Kunming, China, in 1996. Now he is an experimentalist at Chuxiong Normal University.