A Novel Real-Time Method for Moving Vehicle Detection

Haihui Wang[1,2], Zhihong Sun[1], Shuangyu Chen[1]
[1]School of Computer Science and Technology, Wuhan Institute of Technology, China
[2]Hubei Provincial Key Laboratory of Intelligent Robot, China
wanghaihui69@sina.com, 1843317439@qq.com, chenshuangyu2013@163.com

Abstract

As one major part of the Internet of Things, the perception layer plays an important role in cognizing, sensing and controlling the physical environment. It is a fundamental task to design a compatible and extensible IoT perception layer network framework, which could ensure reliability of IoT perception layer. In this paper, based on thorough investigation on perception layer network topology, a new framework of perception layer based on convergence and cooperation is presented. The heterogeneity of network topology in perception layer based on scenarios and applications is analyzed thoroughly. Then a cooperative and converged framework and one information security framework as example are carefully described, which shows that the proposed framework could simplify the perception layer design in heterogeneous networks and facilitate application functionality extension. Finally, some conclusion and future work is presented.

Keywords: Internet of Things (IoT), IOT perception layer, Cooperative and converged framework.

1 Introduction

Transportation management departments in various countries have introduced video monitoring systems for a safe and orderly transportation system. Moving vehicles detection is one of most fundamental challenges in Intelligent Transportation System (ITS) [1]. However, the existing digital video surveillance systems provide the infrastructure only to capture, store, and distribute videos, leaving the task of threat detection exclusively to human operators. Vehicle detection is regarded as a fundamental task in ITS and allows for some more applications such as vehicle tracking, license plate detection and segmentation, and recognition of car type and color. Vision based vehicle detection can assist in supervising the traffic by using only a few cameras and without relying on sophisticated equipment such as radars and lidars.

This paper presents a vehicle detection algorithm based on improved Adaboost algorithm and the inter-frame difference method. The main contributions of this work are:

- **Novel dataset:** Some benchmark vehicle data sets are acquired by collecting only the rear views of the vehicles ahead, using cameras mounted on the driving vehicle [2-3]. Other data sets are formed by collecting images obtained from cameras located very high above the ground [4-5]. We form our own data set of monocular images, which consists of images recorded from cameras fixed on the traffic light posts. Our data set consists of images showing the rear as well as the frontal view of the vehicles. These images are collected over a period of 24 hours. So they encompass various illumination variations.
- **Reduction in classification overhead:** Our algorithm works by combining the extraction of useful spatial-temporal information from the images with the classification scheme. The Region of Interest (ROI) for vehicle detection is significantly reduced by applying the trained classifier only on the foreground contours generated via difference of framing technique. This approach speeds up the detection while maintaining the accuracy, so the detection rate is highly improved.
- **Robustness to poorly lit traffic scenes:** We equip our detector with the capability to detect vehicles even in low-illuminated environments.
- **Speed:** Our detection method can detect vehicles in frames extracted after every 30 ms, which renders our method as being real-time.
- **Extended Set of Haar-like Features:** Our last contribution in this paper is to model vehicle shapes in terms of a set of traditional and extended rectangles that are geared towards different vehicle parts. Based on this shape model, a series of Haar-like features are designed to describe local differences. Accordingly, we employ a compact feature pool that is well tailored to vehicle shapes. By employing an extended set of Haar-like Features, the detection rate is highly improved with an Additional time cost.

The rest of this paper is organized as follows: Section 2 presents the related work, draw a series of vehicle detection methods and our method. In Section 3, we proposed the system model of our method and draw Adaboost algorithm in detail. In Section 4, we present Moving vehicles detection, including Inter-frame difference rule, Multiscale detection and Gamma correction. In Section 5, we describe Informed Haar-like features. In Section 6, presents and analyzes our experimental results. Finally, in Section 7, our algorithm is summarized.

*Corresponding author: Haihui Wang; E-mail: wanghaihui69@sina.com
DOI: 10.6138/JIT.2016.17.7.20151027g

2 Related Work

Moving vehicles detection has been widely researched and applied. The methods of moving object detection can be divided into four major types: background subtraction, frame difference method, optical flow and feature matching method [6]. Methods relying on background subtraction [7-8] and frame difference [9-10] encapsulate the moving target object by extracting foreground region. The output of these methods is vulnerable to changes in weather, illumination, shadow interference, and other moving objects. Optical flow based methods [11-12] and feature matching methods [13] search and match the particular features throughout the whole frame. However, they usually cannot meet the real-time requirements for practical applications The HOG features have been used for object detection. For instance, HOG + SVM rigid templates [14] and Deformable Part Model (DPM) detectors [15-16] use the HOG for modeling the target.

Most of the vehicle detection methods used in video monitoring systems either has low detection rate and detection speed. Figure 1 shows the results on some traditional methods. Figure 1(a) shows the input frame. Figure 1 shows the results on some traditional methods.

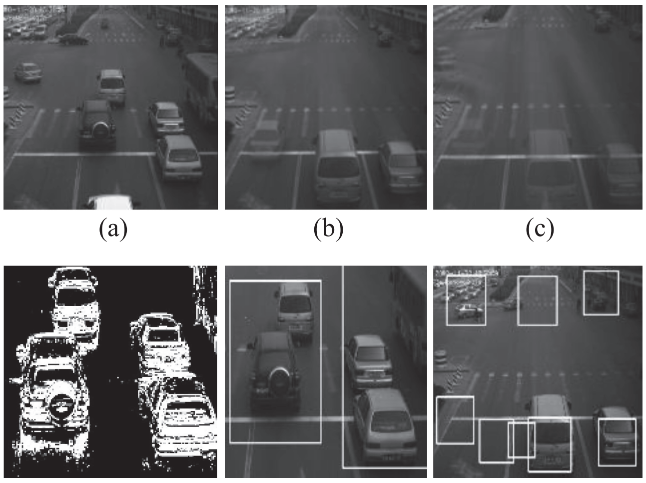


Figure 1 Output of Some Traditional Vehicle Detection Methods. (a) Shows the Input Frame, (b) and (c) Portray the Process of Background Subtraction, as a Result of which Motionless Vehicles Are Erased Gradually. (d) Shows a Binary Image Obtained from Background Subtraction. (e) Is the Detection Result of (d). (f) Shows Some False Positives Obtained by Using a Template Matching Method, which Results in Wastage of Processing Time

The core problem of object detection focuses on individual monocular image frames. Some methods explore leveraging additional information during training and test stages to improve detection output. Such as stereo images [17], tracking [18], or data from other sensors (such as lidar [19] or radar) are used for a fast or accurate detection. In our method, we simply combine spatial-temporal information from previous frames and a fast classifier to organize our detection system, which achieves high accuracy at a high frame-rate.

3 System Model

Adaboost [20] is a kind of machine learning algorithm which is widely used for objection classification. The training phase needs data set for positive samples (images containing the target object) and negative samples (images devoid of the target object). Features are extracted from these samples which are then used to train the classifier. We apply the method proposed into the domain of vehicle detection, and we train the classifier using Haar-like features [21]. Automobile tails and heads contain a large amount of important information we need to acquire, such as license plate, vehicle-logo, vehicle color and other vehicles' characteristic information. The symmetries of automobile tails and heads are helpful in extracting and matching Haar-like features. So we use Haar-like features of automobile tails and heads to train a cascaded classifier.

(1) Haar-Like Features

In order to train a classifier, we use the value of a set of simple features rather than the pixels. The commonly used Haar-like features are classified into three categories: edge features, linear features and center features. Figure 2 shows three kinds of Haar-like features: (a) edge features, (b) center features, and (c), (d) linear features.

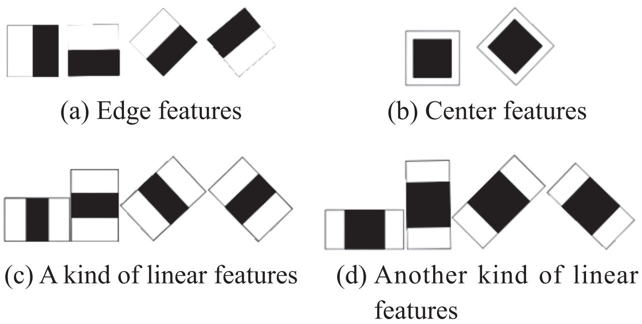


Figure 2 The Commonly Used Haar-Like Features

The template windows are made up of rectangles. We get a feature by subtracting the sum of the pixels in white rectangles from the sum of pixels in the black rectangles [22]. Haar-like features reflect the change of gray value in images, and they are robust to noise, illumination etc.

For features are shown in Figures 2(a), 2(b) and 2(c), the feature is found as:

$$H = SUM_{white} - SUM_{black} \tag{1}$$

For features shown in Figure 2(d), the feature is found as:

$$H = SUM_{white} - 2 * SUM_{black} \tag{2}$$

We select 2 rectangle features to evaluate their performances through our data sets by a statistical experiment. Table 1 shows feature A and feature B's performances through our data sets. The two features have a size of 24 by 24 pixels. Detection rate in last row is generated by the mean value.

Table 1 Feature A and feature B's Performances

Feature	A 		B 	
Sample	Positive	Negative	Positive	Negative
Number	3,000	7,000	3,000	7,000
Summation value of feature	66,941	254,923	763,645,556	879,726
Mean value of feature	22.3	36.4	-374.9	125.7
Detection rate	93.3%	57.1%	67.4%	54.7%

From Table 1, we can conclude that feature A has better performance than feature B as a descriptor of vehicle model.

(2) Classifier Training

For 24 by 24 images, there are over 10 thousand of rectangles and the number is far more than the pixels, which makes the features over-complete. Even though each feature can be acquired easily, computing and matching all the rectangles need too much time. Adaboost method provides a way to find a few most representative and useful features to finish this task.

We select a feature by considering its detection rate on a given data set, The desired feature has the ability to effectively separate vehicles and non-vehicles. Then the selected features are combined as weak classifiers to build a much more robust and efficient classifier.

The feature vectors extracted for the target object in different frames are not exactly the same; rather they lie within a range of values. Hence, it is reasonable to determine this range by describing a threshold. It can be assumed that the rectangular features whose values fall within the prescribed range represent the target. The weak learning algorithm trains a weak classifier for each such rectangular feature, a weak classifier consists of a threshold, an offset value and the value of the feature vector. The key part of training a weak classifier is achieving appropriate thresholds.

A training set $\{(x1, y1), (x2, y2), (x3, y3), ..., (xn, yn)\}$, where n is the number of samples, xi is a sample and $yi \in \{0, 1\}$ is the label of samples: 0 for negative sample which is an image without an automobile tail or head and 1 for positive sample which is an image having an automobile tail or head. We give our algorithm for training a weak classifier in Algorithm 1.

Algorithm 1: Training a week classifier
Input: Image characteristic value $V_{i1}, V_{i2}, ..., V_{in}$
Output: the weak classifier for feature h(x)
Begin
1: compute all the feature $V_{i1}, V_{i2}, ..., V_{in}$,
sort in ascending order, $V_{i1} \le V_{i2} ... \le V_{in}$,
2: assign a weight ωj $(1 \le j \le n)$ to each sample,
3: initialize the value of feature i ξ_i, $\xi_i = \sum_{y_i=-1} \omega_i$
4: **if** $(q_i \ \ y_{ik} = 0, (1 \le k \le n))$ **then**
5: $\xi_i = \xi_i - \omega_{ik}$, $\xi_k = \xi_i$,
6: **else**
7: $\xi_i = \xi_i + \omega_{ik}$, $\xi_k = \xi_i$,
8: select $\xi_{\min}$ (while tag = s)
9: assign the threshold of feature i, θi = V_{is}
10: **if** $(Avg \sum_{y_i=0} V \ge Avg \sum_{y_i=1} v)$ **then**
11: $p_i = 1$,
12: **else**
13: $p_i = -1$;
14: $h_i(x) = \begin{cases} 1, & p_i f_i(x) < p_i\theta_i \\ 0, & otherwise \end{cases}$, where $h(x)$ is the weak classifier for feature i.
End

The procedure for acquiring a strong classifier is given in Algorithm 2. In Adaboost Algorithm, we input n samples, initialize the weight of the sample $q_{1,i}$, and we suppose the number of weak classifiers is T, on the entire sample set, for t = 1, 2, 3, ... First, our sample weights renormalization. Secondly, for each feature f, we train a weak classifier h_f and calculate the characteristics of weak classifiers weighted error rate e_f. Then, selecting the weak classifiers h_t, which has the smallest error rate e_t. Thirdly, we will readjust sample weights $q_{t+1,i}$. Finally, we get a strong classifier.

Algorithm 2: Adaboost algorithm
Input: sequence of n samples $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$
Output: the final strong classifier:
Begin

1: Initialize the weight of the sample $q_{1,i}$,

2: $q_{1,t} = \begin{cases} 1/2h, y_i = 1 \\ 1/2k, y_i = 0 \end{cases}$

3: T is the number of weak classifiers,

4: for ($t = 1, t \leq T, t$++)

5: $q_{t,i} = \dfrac{q_{t,i}}{\sum_j^n q_{t,j}}$

6: For each feature f, train a weak classifier h_f. The error of h_f is evaluated with respect to weight q_i:

7: $e_f = \sum_i q_i \,|h_f - y_i|$

8: circulation end

9: Select the weak classifier h_t which has the minimum error rate e_t:

10: $e_t = \min \sum_i q_i\, |h_f - y_i| = \sum_i q_i\, |h_t - y_i|$

11: Update the samples' weights by adjusting weights:

12: $q_{t+1,i} = q_{t,i}\beta_t^{1-\theta_i}$, ($\beta_t = e_t / 1 - e_t$)

13: **if** ($h_t(x) = 1$), the samples are correctly classified **then**:

14: $\theta i = 0$

15: **else**

16: $\theta i = 1$

17: $H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \sum_{t=1}^{T} \alpha_t h_t, \; where \; \alpha_t = \log \beta_t^{-1} \\ 0 & others \end{cases}$

End

4 Moving Vehicles Detection

4.1 Inter-Frame Difference Rule

In this section, we show how to use a three inter-frame difference method to obtain spatial and temporal information to improve detection. The basic idea is to generate a foreground contour and then recognize cars by a detector in the contour zones.

$$\begin{cases} D_{t,t-1}(x, y) = |I(x, y, t) - I(x, y, t-1)| \\ D_{t+1,t}(x, y) = |I(x, y, t+1) - I(x, y, t)| \end{cases} \tag{3}$$

$I(x, y, t)$ corresponds to the frame at time t and $I(x, y)$ is the gray value of the pixel at the position (x, y). $D(x, y)$ is a difference frame. A self-tuning threshold T is used to do binarization operation on $D(x, y)$. T is obtained by computing the luminance histogram of the image.

$$\begin{cases} b_{t,t-1}(x, y) = \begin{cases} 255 \;;\; D_{t,t-1}(x, y) \geq T \\ 0 \quad\;;\; D_{t,t-1}(x, y) < T \end{cases} \\ b_{t+1,t}(x, y) = \begin{cases} 255 \;;\; D_{t+1,t}(x, y) \geq T \\ 0 \quad\;;\; D_{t+1,t}(x, y) < T \end{cases} \end{cases} \tag{4}$$

We obtain a binary mask image $B(x, y)$ by performing "or" operation on certain binary images.

$$B(x, y) = \begin{cases} 255; & b_{t,t-1}(x, y) \cup b_{t+1,t}(x, y) = 255 \\ 0; & b_{t,t-1}(x, y) \cup b_{t+1,t}(x, y) = 0 \end{cases} \tag{5}$$

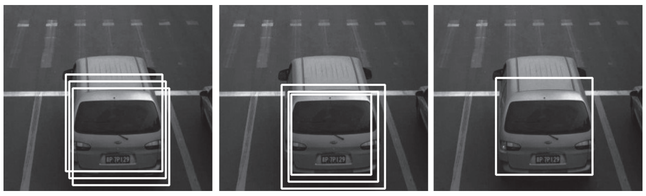
Then we perform corrosion, expansion operation on $B(x, y)$, and remove isolated noise points and some empty points of $B(x, y)$. Finally, the moving region is obtained by finding the connected domain motion regions and marking the regions with extended rectangles, as a result of which a foreground mask image is rebuilt.

4.2 Multiscale Detection

The size of positive samples we use is 24 × 24 pixels. In theory, the classifier can be only used to detect vehicles with a size of 24 × 24 pixels in an image. To detect vehicles of various sizes, we use a multiple scale detection mechanism.

There are two methods of multiple scale detection for Adaboost: the first one is detection using a sliding window whose scale is varied for each iteration. The second method shrinks the frame by a certain scale, but since this method needs to compute Haar features every time, it is not suited to real time detection. We use the first method because the vehicle features can be computed under any scale with the same time cost. The configuration parameters for the sliding window include the scale change and the shifting step. If the values of these parameters are too large, some instances of the target object will suffer from missed detection. On the other hand, if the values are too small, this will result in wastage of processing time. Therefore, a balance has to be found.

The sliding window works like a view finder. The contents of the sliding window are fed to the classifier, which then determines whether the area enclosed inside the window belongs to the target. Usually a vehicle will be detected and marked many times under a scale or different scales. We use the non-maximum suppression method to ensure that a vehicle has only one marked rectangle (the bounding box). Figure 3 shows the results of non-maximum suppression.



(a) (b) (c)

Figure 3 Result of Multiple Scale Detection. (a) Errors within a Scale, (b) Errors in Different Scales, (c) the Final Bounding Box Obtained after Non-Maximum Suppression

4.3 Gamma Correction

Low-light condition problem has received increasing attention in road safety. In the European Union, almost one third of road accidents occur during nighttime [23]. Therefore, the provision of accurate detection results under such conditions is very critical. Considering that the detection ability of our system declines sharply in night time and low illumination environments, we use gamma correction to reduce the effect of low light intensity on feature extraction. Gamma correction is a kind of nonlinear operation used to deal with luminance or tristimulus values in image systems. The image type is judged by analyzing the cumulative histogram distribution of the image pixels. In case of dimly lit scenes, we use Equation (6) to apply the gamma correction:

$$It(x, y) = 255 - (I(x, y) - 255)^2 / 255 \qquad (6)$$

$I(x, y)$ is the intensity of the original pixel in the dark image, and $It(x, y)$ is the intensity of the same pixel after gamma correction. Figure 4 shows a comparison of the original image and the processed image.

(a) (b) (c)

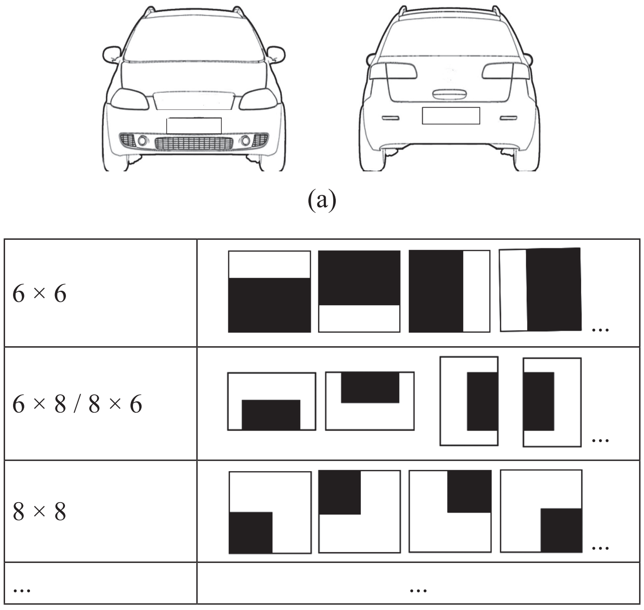
Figure 4 The Result of Gamma Correction. (a) Shows the Nonlinear Mapping between the Original Image and the Processed Image. (b) Shows the Source Image. (c) Gives the Destination Image

5 Informed Haar-Like Features

Before presenting the final test set results of our "core" method, we also consider some possible "addons" features based on the specific common visual appearance. We find that the heads and the tails of most vehicles share a common visual appearance especially the geometry of the window, plate, and car lights region of the vehicle body. Based on this simple vehicle shape model, we design a pool of rectangle features (rectangular templates) that is adapted to these local structures. Our templates are specific for vehicles and therefore lead to a very good performance.

To avoid huge feature pools and high dimensional feature vectors, the informed features have given proper sizes. We have designed 20 kinds about 36,000 informed features. Some features are show in Figure 5.

To improve the performance of our detector, we employ some informed Haar-like Features which drive



(b)

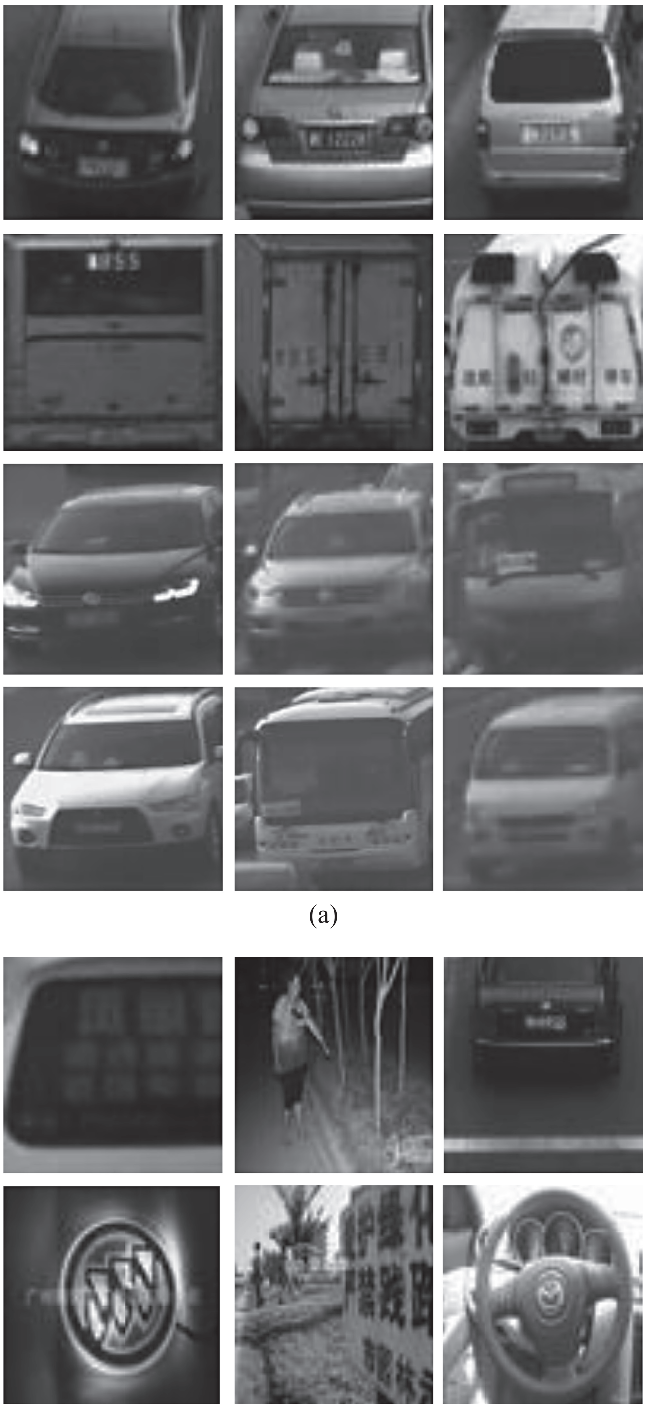
Figure 5 Simple Visual Appearance of Vehicles and Some Informed Rectangle Features: (a) Shows Simple Visual Appearance of Vehicles. (b) Shows Some Informed Rectangle Features

more computation. To deal with the additional memory and computational resources produced by the informed features, some features with less contribution in feature pools can be ignored. As We know: features consist of only 2 pixels contribute little to model vehicle shapes, and small features (less than 4 pixels) at the edge of sample image may contain information of background, so we ignore those thousands of useless features to improve the training and detection speed.

6 Experiments

We now present our experiments, which are carried out on an Intel dual core 2.93 GHz processor with 2G memory. The system mainly includes a training module and a detection module. We compile our own data set consisting of vehicle images extracted from videos captured in different environments (Figure 6). Our data set has 3000 positive samples of size 352 × 288 pixels, sampled from the corresponding videos after intervals of 30 ms. The positive examples consist of 1,200 cars, 1,200 buses and vans, and 600 special vehicles such as street sprinklers and trucks. Moreover, we use 7,500 negative samples. We normalize the positive samples to 24 × 24 pixels to compute features easily.

The training module trains an Adaboost classifier which can be used to detect vehicle tails. This classifier is then used by the detection module, which uses the sliding

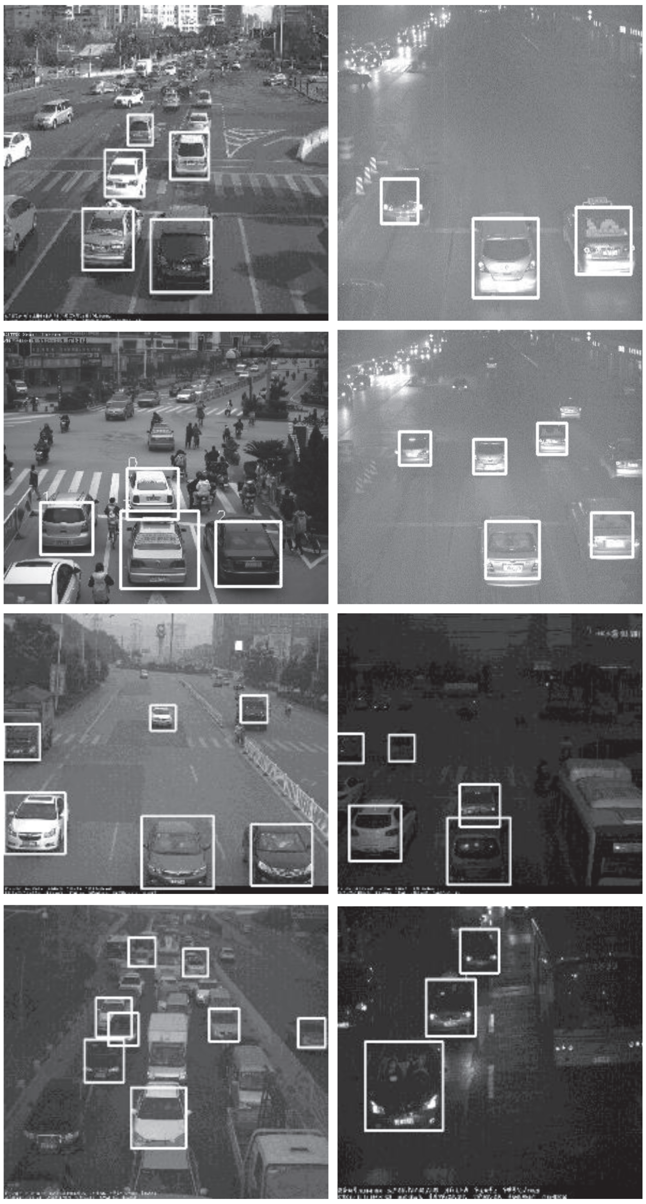


(a)

(b)

Figure 6 Some Samples in Our Dataset. (a) Shows Some Positive Samples which Contain Vehicle Tail and Head. (b) Shows Some Negative Samples

window approach for detecting vehicles in different scales. The detection results are marked and counted using the set of bounding boxes in each frame. Figure 7 shows some detection results obtained via our experiments on the test data sets. It can be seen that for all these scenarios, our algorithm gives excellent results. The last row of columns (a) and (b) shows some missed detection in crowded traffic scenes.



(a) (b)

Figure 7Output of Vehicle Detection System. (a) To Portray the Detection Results on a Set of Real Traffic Scenes Having Complex Backgrounds and Containing Multiple Targets and Special Vehicles. (b) To Show the Detection Output on Poorly Illuminated Images Contained in Our Data Set

In case of cars, an average detection rate of 98% is achieved by the detector, while for vans and buses this rate is 96%. For other special vehicles (street sprinklers and trucks) which are harder to detect due to geometric anomalies, the average detection rate is 88%. The traditional Adaboost detection methods give higher error rates rate for special vehicles. However, our proposed method maintains the average detection rate of 94% on our data set, which contains a high proportion of images having special vehicles. The root cause behind this robustness is the removal of useless regions of the search space by using frame-difference rule.

Next, we evaluate the performance of our algorithm against two other methods, namely the inter-frame difference method and the optical flow method. We compile a test data set of 468 positive samples for this purpose. Table 2 shows the detection results for daytime scenes (only vehicles that are big enough are counted).

Table 2 Detection Results of the Three Methods in Day Time

Method used	Actual number	Counted number	Detection rate (%)	Total processing time (ms)
Inter-frame difference	1,381	1,127	81.61	12,435
Dense optical flow	1,381	1,196	86.80	21,947
Our method	1,381	1,304	94.42	14,656
Our method + informed features	1,381	1,341	97.10	16,739

It can be observed that our method improves the detection rate by 12.81% and 7.62% as compared to the inter-frame difference method and the dense optical flow method, and informed features help to improve the detection rate by 2.68 on the base of our method with an additional time cost, respectively. The results obtained for the four detection methods on nighttime scenes are enlisted in Table 3. The number of frames is 987.

Table 3 Detection Results of the Three Methods in Dark Condition

Method used	Actual number	Counted number	Detection rate (%)	Total processing time (ms)
Inter-frame difference	2,753	1,437	52.20	30,476
Dense optical flow	2,753	1,563	56.77	44,518
Our method	2,753	2,390	86.80	35,370
Our method + informed features	2,753	2,456	89.21	36,739

It can be seen that our method offers drastic improvements in detection rates over the other two methods, increasing the detection rate by 34.6% and 29.83%, and informed features help to improve the detection rate by 2.41, respectively. Nextly, we determine the quantitative results in terms of false positives generated by these three methods, as well as the traditional Adaboost method. We define the impact of false positive in terms of the error rate R_{error}, which is computed as follows:

$$R_{error} = (E_d + E_n) / (N_d + N_n) \tag{7}$$

Where E_d is the number of false positives in all the day time frames and E_n is the number of false positives in all the night time frames. N_d is the number of vehicles we detected in all the daytime frames and N_n is the number of vehicles we detected in all nighttime frames. Figure 8 shows the error rates obtained for all the methods.

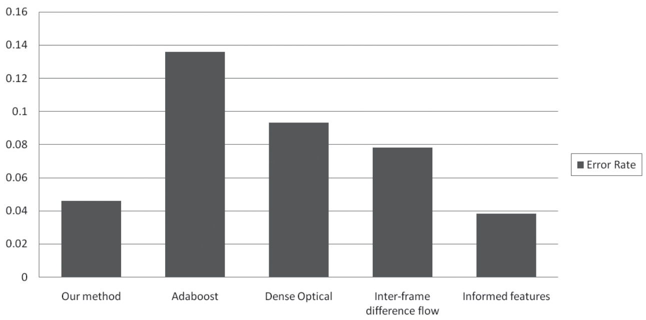


Figure 8 Error Rates

We have proved that our vehicle detection algorithm is robust to weather conditions, light conditions, shadows of other vehicles, etc.

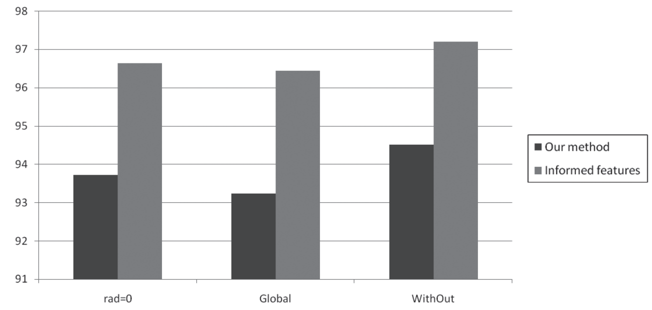
To optimize our detector, we analyze the influences of two methods of image processing.

Image normalization: we analyze the influence of intensity normalization on our features as previous works on rectangular features typically employ various ways of normalization: using local normalization inside each detection window or applying global normalization on the input images. However, according to the results in Figure 9(a), our features obtain best detection rate without normalization.

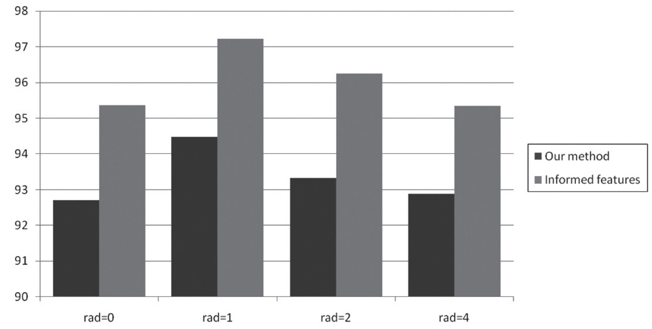
Image smoothing: we analyze the influence of image smoothing on our features as previous works on rectangular features typically employ filters of different radius, while pre-smoothing input images with binomial filters of radius 1 improves the performance by more than 3%, larger radii produce worse results; Figure 9(b) shows the influence of different filters on detection rate.

7 Conclusions

Experimental results show that the proposed vehicle detection algorithm based on improved Adaboost algorithm and inter-frame difference rule is more practical than some traditional methods. Our algorithm not only can accurately detect the position of the vehicle, but also can be more accurate statistics perspective of the vehicle. The



(a) The result of image normalization



(b) The result of image smoothing

Figure 9 Influences of Two Methods of Image Processing, All Results Are Base on Daytime Experiments

application of traditional vehicle detection by Adaboost algorithm shows the multi-scale detection of slow, error detection more difficult to meet real-time and accurate practical application requirements. The proposed method can reduce the detection time for each frame, and to a large extent reduce the false detection rate, providing more time and more accurate data for subsequent license plate detection, vehicle tracking, it can be applied to traffic junctions HD mount monitor, and provide a more intelligent, more secure transportation services. The present method under low illumination, the detection rate has decreased, which is later required further in depth study of the place.

Acknowledgements

This work was supported by the Project on Science Research Program of Hubei Provincial Department of Education in China (T201206), and the National Natural Science Fund Project in China (61175013). We are grateful to those who helped us in compiling our dataset, and those who made valuable suggestions to improve the quality of this paper.

References

[1] B. Salehi, Y. Zhang and M. Zhong, Automatic Moving Vehicles in Formation Extraction from Single-Pass WorldView-2 Imagery, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 5, No. 1, pp. 135-145, February, 2012.

[2] C. Caraffi, T. Vojíř, J. Trefný, J. Šochman and J. Matas, A System for Real-Time Detection and Tracking of Vehicles from a Single Car-Mounted Camera, *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, 2012, pp. 975-982.

[3] Z. Sun, G. Bebis and R. Miller, Improving the Performance of On-Road Vehicle Detection by Combining Gabor and Wavelet Features, *IEEE 5th International Conference on Intelligent Transportation Systems*, Singapore, 2002, pp. 130-135.

[4] Grupo de Tratamiento de Imágenes, *Vehicle Image Database*, 2012, http://www.gti.ssr.upm.es/data/Vehicle_database.html

[5] H.-H. Nagel, *Image Sequence Server*, 1997, http://i21www.ira.uka.de/image_sequences/

[6] Y. Wang, *Research on Multiple Moving Objects Real-Time Detecting and Tracking in Complex Scene*, Master's Thesis, Wuhan Institute of Technology, Wuhan, China, 2011.

[7] Y.-H. Zhang, X.-X. Xu and Q.-L. Chen, A Vehicle Detection System with Adaptive Background Update and Shadow Suppression, *Journal of Shanghai University (Natural Science)*, Vol. 2005, No. 5, pp. 465-471, May, 2005.

[8] C. Wang, Z.-H. Xi and C.-L. Xiao, Moving Objects Detection Based on Background Substraction Method, *Applied Science and Technology*, Vol. 2009, No. 10, pp. 16-18, October, 2009.

[9] L. Zhang, L.-M. Chen, W. He and L.-M. Guo, Application of an Improved Frame-Difference Method Based on Video in Traffic Flow Measurement, *Journal of Chongqing University (Natural Science Edition)*, Vol. 2004, No. 5, pp. 31-33, May, 2004.

[10] H. Wang, J. Bai, C. Li and Q. Wu, A System for Moving Object Detection and Shadow Extermination, *2005 IEEE International Workshop on VLSI Design and Video Technology*, Suzhou, China, 2005, pp. 117-120.

[10] Z. Chen, G. Wang and C. Liu, Statistics of Vehicle Flows Based on Computer Vision, *Journal of Huazhong University of Science and Technology*, Vol. 34, No. 5, pp. 46-49, May, 2006.

[12] S. Huang, L. Tao and T. Zhang, An Improved Algorithm of Moving Object Detection Based on Optical Flow. *Journal of Huazhong University of Science and Technology*, Vol. 33, No. 5, pp. 39-41, May, 2005.

[13] Y. Sheng, H. Xie, B. Li and Q. Huang, Designs for Virtual Loops Based on Fast Matching Algorithm, *Journal of Huazhong University of Science and Technology*, Vol. 32, No. 3, pp. 106-108, March, 2004.
[14] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, *IEEE 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 886-893.
[15] P. Felzenszwalb, D. McAllester and D. Ramanan, A Discriminatively Trained, Multiscale, Deformable Part Model, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1-8.
[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp. 1627-1645, September, 2010.
[17] C. G. Keller, M. Enzweiler, M. Rohrbac, D. F. Llorca, C. Schnorr and D. M. Gavrila, The Benefits of Dense Stereo for Pedestrian Detection, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, pp. 1096-1106, December, 2011.
[18] A. Ess, B. Leibe, K. Schindler and L. van Gool, Robust Multiperson Tracking from a Mobile Platform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 10, pp. 1831-1846, October, 2009.
[19] C. Premebida, J. Carreira, J. Batista and U. Nunes, Pedestrian Detection Combining RGB and Dense LIDAR Data, *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, 2014, pp. 4112-4117.
[20] Y. Freund and R. E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119-139, August, 1997.
[20] P. Viola and M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, 2001, pp. 511-518.
[22] R. Lienhart and J. Maydt, An Extended Set of Haar-Like Features for Rapid Object Detection, *2002 International Conference on Image Processing*, New York, 2002, pp. 900-903.
[23] R. O'Malley, E. Jones and M. Glavin, Rear-Lamp Vehicle Detection and Tracking in Low-Exposure Color Video for Night Conditions, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 2, pp. 453-462, June, 2010.

Biographies

Haihui Wang, born in 1969. He received the PhD degree in Pattern Recognition and Artificial Intelligence from Huazhong University of Science and Technology, China, in 2003. He is currently a professor in Wuhan Institute of Technology, China. He's main research field includes digital image processing, machine vision, computer control technology.

Zhihong Sun, born in 1990, a postgraduate at School of Computer Science & Engineering, Wuhan Institute of Technology, China. His specialty is computer software and theory, and his research interests includes digital image processing, pattern recognition.

Shuangyu Chen, born in 1990, a postgraduate at School of Computer Science & Engineering, Wuhan Institute of Technology, China. Her specialty is computer software and theory, and his research interests includes digital image processing, machine vision.