

# A New Approach for Vehicle Recognition and Tracking in Multi-camera Traffic System

Wenbin Jiang<sup>(✉)</sup>, Zhiwei Lu, Hai Jin, and Ye Chi

Services Computing Technology and System Lab,  
Cluster and Grid Computing Lab, School of Computer Science and Technology,  
Huazhong University of Science and Technology, Wuhan 430074, China  
wenbinjiang@hust.edu.cn

**Abstract.** In order to ensure recognition accuracy, intelligent traffic video tracking system usually requires various types of information. Therefore, multi-features fusion becomes a good choice. In this paper, a new recognition approach for vehicle types based multi-feature fusion is proposed, which is used for vehicle tracking in a multi-camera traffic system. An improved Canny operator is presented for edge detection. SURF (*Speeded Up Robust Features*) is used for local feature extraction. To improve the performance of distance calculation between features, a refined method based on Hellinger kernel is put forward. A position constraint rule is applied to reduce unnecessary fake matchings. Finally, the information of vehicle types combined with LBP (*Local Binary Pattern*), HOG (*Histogram of Oriented Gradients*) is used for a multi-camera vehicle tracking platform, which adopts Hadoop to realize the parallel computing of the system. Experimental results show that the proposed approach has good performance for the platform.

**Keywords:** Recognition · SURF · Distance calculation · Position constraint · Parallel computing

## 1 Introduction

In the past few years, traffic video surveillance technologies (mainly for vehicles) have gotten more and more attention. Applications based on these technologies have made rapid progress. Simultaneously, a large number of resources have been invested.

Intelligent traffic video surveillance is a new generation of monitoring technology, which uses cutting-edge computer vision technologies to analyze monitoring videos, does some forecasts, and identifies various abnormal conditions.

One of the key technologies for intelligent video surveillance is target recognition. Only after the target being identified from the surveillance video, further analysis about the behaviors and characteristics of the targets can be done.

Recently, multi-camera-based traffic video tracking systems become more and more popular, which track some targets among multiple cameras. The vehicle target recognition is also a very important beginning step for further tracking and other analysis.

The local feature extraction and matching of vehicle targets are very important for vehicle tracking, especially across different cameras. However, vehicle feature extraction and recognition still face many challenges:

- Feature diversity, such as shape, color, size, and fine differences between corresponding key points. Moreover, from different perspectives, the appearances of specific vehicle are usually different;
- Shield and interference, such as shields between vehicles and interference of the complex background. Especially in urban situations, backgrounds are generally complex;
- Computation and accuracy for feature extraction. More robust features and higher accuracy require more computation resources, which is a big bottleneck for real-time vehicle surveillance.

In this paper, we propose a new vehicle recognition approach based on multi-feature fusion. The front faces of vehicles are used for this task (allowing small inclination angles). Like faces of human being, the front faces of vehicles have abundant features to identify them. Considering the symmetry, only the left light of the vehicle is chosen for feature extraction.

In the proposed approach, the feature extraction has two steps. The first is to use an improved Canny [1] algorithm to extract the vehicle light shape from the image of the vehicle front face to get the profile image of the headlight. The second is to extract SURF (*Speeded Up Robust Features*) [2] features from the profile image of the headlight and the image of the vehicle face.

Some standard vehicle face images of different types of vehicles are selected as standard models. Their features are extracted by above approach and stored into a local standard feature database.

To identify a new vehicle, first, extract features from its face image by our proposed approach; then, compare them with the features of various vehicles stored in the standard feature database to find the type of the vehicle.

Finally, the information of vehicle types combined with other features of *Local Binary Pattern* (LBP) and *Histogram of Oriented Gradients* (HOG) is used for a multi-camera vehicle tracking system. Hadoop is used to realize the parallel computing of the system.

The remaining of this paper is organized as follows. Section 2 introduces related work on edge detection, extraction of local features, etc. Section 3 shows the architecture, algorithm and implementation of the proposed recognition system of vehicle, and some improvements in details. Section 4 gives the testing performance analysis. Finally, conclusion is drawn in Sect. 5.

## 2 Related Work

Anagnostopoulos et al. [4] presented a vehicle recognition system by automatic license plate recognition, vehicle manufacturer/model detection and under-vehicle inspection. Sivaraman et al. [5] proposed a novel active-learning approach to build vehicle-recognition and tracking systems. A passively trained recognition system was built

using conventional supervised learning. A second round of learning was then performed to build an active-learning-based vehicle recognizer. Particle filter tracking was integrated to build a complete multiple-vehicle tracking system. Dlagnekov [6] presented a way to use SIFT (*Scale Invariant Feature Transform*) [7] feature of vehicle rear to recognize a specific vehicle. Petrovic et al. [8] presented an application of automatic recognition of vehicle type for secure access and traffic monitoring. It showed that a relatively simple set of features extracted from sections of front face images of cars can be used to obtain high performance verification and recognition of vehicle type (both car model and class). Euclidean distance is used to measure the similarity between the features from different objects.

Multiple features can help improve the effectiveness of the appearance matching and reduce the influence of environment changes, especially in multi-camera system. Tamrakar et al. [9] explored a way combining the features of shape and appearance for object recognition. Some ways were explored in which the geometric aspects of an object can be augmented with its appearance. The main idea is to construct a dense correspondence between the interior regions of two shapes based on a shape-based correspondence so that the intensity and gradient distributions can be compared. Wang et al. [10] employed multiple features such as color histogram, height, moving detection, travel time and speed to match objects across non-overlapping views.

In our proposed approach, SURF feature of vehicle front face image is combined with features of LBP and HOG to make vehicle tracking.

Edge detection is an important step for feature extraction. Existing edge detection algorithms can be roughly divided into five groups: gradient method [3], surface fitting method [11], second derivative method [12], multi-scale method [13] and morphological method [14]. Canny is a good practical edge detection algorithm. Its three performance criteria are: (1) good detection performance; (2) good localization performance; (3) only one response to a single edge. According to the above principles, Canny algorithm has a good intensity estimate of edges, which contains both edge gradient direction and edge intensity information. In our proposed approach, Canny is used for edge detection.

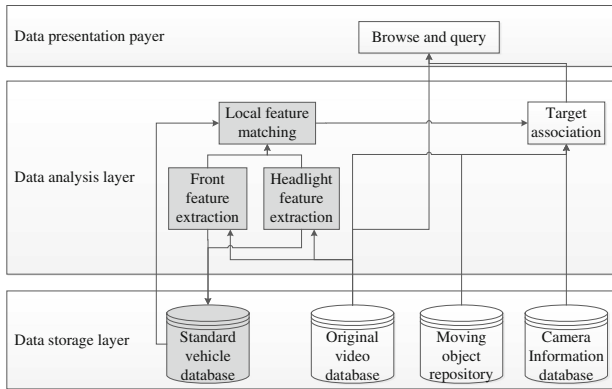
For feature extraction, SIFT [7] is a good image feature proposed by Lowe. It is invariant to rotation and scale, which is often used in image matching and connection. However, the complexity of computation for SIFT is usually high and the amount of data is large. SURF [2, 15] is a speedup and robust version of SIFT, which can offer better performance than SIFT. The calculation of SURF feature consists of keypoints detection, descriptor generation and keypoints matching. [16] gave the comparison of SIFT, *Principal Component Analysis* (PCA)-SIFT [17] and SURF.

For feature matching, there are some methods to compare the feature descriptors, such as L2 metric, Euclidean distance, *Earth Mover's Distance* (EMD) [18], EMD-L1 [19], diffusion distance [20],  $\overline{EMD}$  [21] and Quadratic-Form distances [22]. SIFT and SURF are originally designed to be used with Euclidean distance [2, 7]. To improve the performance of feature matching, a refined distance calculation method based on Hellinger kernel is presented in our proposed approach.

### 3 Design and Implementation

#### 3.1 Framework Architecture

The interactive traffic video tracking platform, which have been presented by our research group [23], consists of the following three parts: data storage layer, data analysis layer, data presentation layer. Figure 1 shows the hierarchical architecture of the interactive traffic video tracking system.



**Fig. 1.** The architecture of the interactive traffic video tracking system

In this architecture, data storage layer stores all data of the system (such as original video, multi-camera information, standard vehicle database). The data analysis layer includes several processes for traffic video, such as local feature extracting, local feature matching, and target association. The data presentation layer is the interface of the system to provide video browsing, object querying and so on.

In this paper, we focus on the recognition system of vehicle based on multi-feature fusion. The system is composed of two key steps: one is local feature extraction (including front feature and headlight feature extraction), and the other is local feature matching. The involved components are shown in Fig. 1 with gray color.

For vehicle type recognition, the input data are images of front faces of vehicles. The output data are types of the vehicles. The main procedure in the recognition system includes the following steps:

- (a) Loading an image;
- (b) Capturing the headlight area, and extracting the edges of the headlight;
- (c) Extracting SURF feature of the front face of the vehicle;
- (d) Calculating the distances between the features of headlight edge image as well as the front face image and ones in the model database respectively;
- (e) Comparing the calculation results. The vehicle type which having the minimum distance is the most likely result.

Here, an improved Canny operator is presented for edge detection. To improve the performance of distance calculation between features, a refined distance calculation method based on Hellinger kernel is put forward. Moreover, a position constraint rule also is applied to reduce unnecessary fake matching. More details about them are shown below.

### 3.2 Edge Extraction

Considering the headlights are main fields that contains abundant special features but with much light noise, we do edge extraction for them first, then make feature extraction. In the process of headlight edge extraction, we adapt an improved Canny algorithm. We can extract the edge feature by following steps: (a) loading the gray image; (b) Gaussian filtering and smoothing; (c) calculating the gradient amplitude and direction; (d) doing non-maximum suppression; (e) double threshold detecting and edge tracing. Then, we get the edge image of the headlight. In the processing, we should choose suitable threshold values in order to get ideal result.

**Gradient Calculation.** Traditional Canny algorithm uses first order partial derivatives of finite difference in  $2 \times 2$  neighborhood to calculate the gradient amplitude and gradient direction of each point in the smoothed image  $I(x, y)$ . Whereby, the partial derivatives  $G_x(i, j)$  and  $G_y(i, j)$  of point  $(i, j)$  along the  $x$  and  $y$  directions respectively are:

$$G_x(i, j) = [I(i, j + 1) - I(i, j) + I(i + 1, j + 1) - I(i + 1, j)]/2 \quad (1)$$

$$G_y(i, j) = [I(i, j) - I(i + 1, j) + I(i, j + 1) - I(i + 1, j + 1)]/2 \quad (2)$$

The gradient amplitude  $M(i, j)$  and the gradient direction  $\theta(i, j)$  of each pixel  $(i, j)$  in the image  $I(x, y)$  are defined as follows respectively:

$$M(i, j) = \sqrt{G_x^2(i, j) + G_y^2(i, j)} \quad (3)$$

$$\theta(i, j) = \tan^{-1}(G_y(i, j)/G_x(i, j)) \quad (4)$$

Here, the gradient amplitude is calculated with the finite difference in  $2 \times 2$  neighborhood, which is more sensitive to noise. Here, we make an improvement that calculates the gradient amplitude with  $3 \times 3$  neighborhood to reduce the influence of noise. To save space, we just show the results of  $x$  direction in details. For  $y$  direction, it has a similar fashion to  $x$  direction.

The gradient amplitude of  $x$  direction is

$$G_x(i, j) = I(i + 1, j) - I(i - 1, j) \quad (5)$$

The gradient amplitudes of  $\pi/4$  and  $3\pi/4$  directions respectively are

$$G_{\pi/4}(i,j) = I(i-1,j+1) - I(i+1,j-1) \tag{6}$$

$$G_{3\pi/4}(i,j) = I(i+1,j+1) - I(i-1,j-1) \tag{7}$$

The difference of horizontal direction is

$$E_x(i,j) = G_x(i,j) + [G_{\pi/4}(i,j) + G_{3\pi/4}(i,j)]/2 \tag{8}$$

The gradient amplitudes and the gradient direction are

$$M(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j) + G_{\pi/4}^2(i,j) + G_{3\pi/4}^2(i,j)} \tag{9}$$

$$\theta(i,j) = \tan^{-1}(E_y(i,j)/E_x(i,j)) \tag{10}$$

where  $G_y$  and  $E_y$  are similar to  $G_x$  and  $E_x$  respectively except different directions.

Finally, formulas above can be turned into the templates as follows:

$$G_x(i,j) = \begin{vmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{vmatrix}, G_y(i,j) = \begin{vmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{vmatrix} \tag{11}$$

The improved method considers the diagonal direction of pixels, and draws it into the differential mean value calculation. The gradient amplitude of a pixel depends on the rest eight pixels in  $3 \times 3$  neighborhood. Compared with the traditional way, the quantity of neighbor pixels of the proposed way is increased largely in our presented way. It enhances the gradient amplitude stability, reduces the impact of the single noise point, improves the edge localization accuracy, and effectively suppresses the noise.

**Self-adaptive Double Threshold Detection.** During the process of edge points detection, self-adaptive double threshold algorithm automatic reduces the influence of the pixels which have smaller gradient amplitude, further eliminates false edges. Higher threshold can eliminate most noise points, but cause the loss of some useful information; while lower threshold can save more image information. Therefore, we present a way for threshold decision based on higher threshold coupled with lower threshold as supplementary decision, which can effectively reduce false edges, and ensure the integrity of image edges greatly. The threshold is set automatically according to the gradient amplitudes of the image pixels, only depending on the initial selection proportion for threshold. No artificial operations are needed.

**Noise Elimination.** From the results of the image processed before, we know that, pixels in the image have not only the gradient amplitude information, but also the gradient direction information. If a point belongs to an edge, its gradient direction generally points to the normal direction of the edge. Isolated noise points often have not obvious gradient direction. Therefore, according to the difference of gradient direction

characteristics between edge points and noise points, we can effectively distinguish edge points.

In the  $3 \times 3$  neighborhood which centers on  $o$ , for a point  $t$  around  $o$ , if  $o$  and  $t$  are in the same edge, the line connecting point  $o$  and  $t$  has similar direction as the edge. Here, define  $Angle_t$  as the absolute value of the angle between the normal direction of the line and the mean gradient direction of the edge. For a point on the edge, its gradient direction should have the similar direction as the normal direction of the line. So  $Angle_t$  should be a very small value.

In the case of a noise point, because its gradient direction is generally different with its adjacent points, the corresponding  $Angle_t$  is large usually. As a consequence, we can detect noise points and edge points in an image by this method effectively, and eliminate the noise points. At the same time, we can realize more precisely edge point detection and subsequent linking.

### 3.3 Local Feature Extraction

In image processing, there are many kinds of features for identifying and recognizing a target. In this paper, SURF is used for local feature extraction, which is scale-invariant and rotation-invariant. The main procedure to extract local features is: (a) loading the gray image and creating integral image; (b) computing interest points with Hessian matrix; (c) locating interest points precisely; (d) assigning dominant orientation; (e) creating interest point descriptors.

After these steps, we get hundreds of 64-length vectors for local features of the image. If the image is the front face image of a standard vehicle, we store the features into standard database.

The process to extract local feature is described briefly in the following sections.

**Keypoints Detection.** In SURF, the scale-space of the image (also called Gaussian pyramid) is mainly used to obtain interest points in different scales. It convolves the image iteratively with Gaussian kernel and sub-samples it repeatedly. This method results in each layer relying on the previous, and thus, high complexity. The sizes of kernels can be changed to create the Gaussian pyramid. Laplacian of Gaussian is approximated to a box filter. This allows multiple layers of the scale-space pyramid to be processed simultaneously without subsampling the image, which leads to better performance.

The matrix of Hessian is used for the interest point localization to determine whether a point is extremum. The integral image is used for the computation of the matrix of Hessian. Each extremum is compared with 26 adjacent points around it after they are picked out. If the value of an extremum point is above or below all of the 26 adjacent points, this extremum point becomes an interest point.

**SURF Interest Point Descriptor.** SURF interest point descriptor relies on the dominant orientations of all the interest points. The descriptor component is built based on the dominant orientations. Haar wavelet response is used for calculating the dominant orientation. The Haar responses are calculated in both  $x$  and  $y$  coordinates in the circle

region centered at interest points with a radius of  $6\sigma$ . The size of Haar wavelet is  $4\sigma$ , and the sum of vectors is calculated in every 60 degrees in the circle. At last, the orientation with the largest sum of vectors is the dominant orientation.

After the dominant orientation is obtained, a square window with a side length of  $20\sigma$  is constructed centered at each interest point. It is divided into  $4 \times 4$  sub-regions. The wavelet response is calculated in both the dominant orientation and the direction vertical to it. The wavelets of  $x$  and  $y$  are defined as  $dx$  and  $dy$ . Four values  $\sum dx$ ,  $\sum dy$ ,  $\sum |dx|$ ,  $\sum |dy|$  are calculated. Then they are combined and normalized to a 64-length vector for each interest point, the SURF descriptor. The SURF descriptor is invariant to scale, rotation and translation of images.

### 3.4 Matching Local Features

**Improvement in the Distance Calculation.** For some tasks such as texture classification and image categorization, it is well known that using Euclidean distance to compare histograms often yields inferior performance compared with other measures [24]. SURF was originally designed to be used with Euclidean distance. However, since it is a histogram question, the question arises as to whether other histogram distance measures can work better with it. Here, we show that using Hellinger kernel does bring a great benefit indeed.

In the following, more details about how to use Hellinger kernel is shown. Suppose  $x$  and  $y$  are  $n$ -vectors with unit Euclidean norm ( $\|x\| = 1$ ), then the Euclidean distance  $d_E(x, y)$  between them is related to their similarity (kernel)  $S_E(x, y)$ :

$$d(x, y)^2 = \|x - y\|^2 = \|x\|^2 + \|y\|^2 - 2x^T y = 2 - 2x^T y$$

where  $S_E(x, y) = x^T y$ . Here, we intend to replace the Euclidean similarity/kernel with Hellinger kernel.

The Hellinger kernel (also known as the coefficient of Bhattacharyya) for two L1 normalized histograms  $x$  and  $y$  (i.e.  $\sum_i^n x_i = 1$  and  $x_i \geq 0$ ) is defined as

$$H(x, y) = \sum_{i=1}^n \sqrt{x_i y_i} \tag{12}$$

SURF vectors can be compared by a Hellinger kernel using a simple algebraic manipulation in two steps: (a) normalizing the SURF vector in L1 norm (originally it has unit L2 norm); (b) calculating the square root of each element

$$S_E(\sqrt{x}, \sqrt{y}) = \sqrt{x}^T \sqrt{y} = H(x, y)$$

and the obtained vectors are L2 normalized since

$$S_E(\sqrt{x}, \sqrt{x}) = \sum_i^n x_i = 1$$

Thus we use Hellinger kernel to compare the original SURF vectors:



$$d(\sqrt{x}, \sqrt{y})^2 = 2 - 2\sqrt{x}^T \sqrt{y} = 2 - H(x, y)$$

**Constraints Between Matching Point Positions.** For the situation in our system, the positions of matching points have some relationship. It means that a point in image A can only match with some points in image B which are close to the corresponding position of the former. In other words, we do not look for matching point from image B for the point in image A out of some areas centered at the point in A. This method can reduce considerable unnecessary fake matchings.

Suppose the threshold for the area is  $T$ . The judgment criteria for position constraints are shown as follow:

For the point A  $(x_1, y_1)$  in image  $I_1$  with size  $(w_1, h_1)$  and the point B  $(x_2, y_2)$  in image  $I_2$  with size  $(w_2, h_2)$ , if the following conditions are met, the two points in image  $I_1$  and  $I_2$  are possible correct matching points.

$$\left| \frac{x_1}{w_1} - \frac{x_2}{w_2} \right| \leq T, \quad \left| \frac{y_1}{h_1} - \frac{y_2}{h_2} \right| \leq T \quad (13)$$

In the formulae above,  $T$  can be fixed according to practical requirements. In our system,  $T$  is usually less than 0.4, according to the experience.

### 3.5 Camera Network and Its Sub-graph Partitioning

Our proposed vehicle type recognition approach has been used in a non-overlapping multi-camera large surveillance platform that we presented in [23].

Generally, it is hard to track targets across multi-camera network. The burden of computation is often heavy. So can we divide the network into several independent sub-network (namely sub-graph) units so that some parallel strategies can be used?

In our multi-camera system, it is divided into some independent sub-networks according to the characteristics of its topology relationship. After this partitioning, target association algorithm can be run independently. The structure of the camera network should not be destroyed by the sub-network partition, which can make sure that the trajectory obtained in each sub-network can be easily merged to make the whole vehicle trajectory.

Two principles should be followed: completeness principle and minimization principle. The completeness principle assures that all nodes involved are within the sub-graph unit when calculating the associate relationship of the targets, and do not need exchange information with other units. The minimization principle makes the structure of each sub-graph unit as small as possible.

Our system uses Hadoop to parallelize the computation of all sub-graphs. First, different tasks from different sub-graphs are mapped to different MapReduce nodes for track analysis independently. Then, all information will be returned to central server to be reduced.

A minimum cost and maximum flow algorithm based on the relationship of the camera network topology is used for target association in our system.

One of the key parts of the target association is how to describe the similarity between objects from different objects. Here, assume target  $a$  and target  $b$  are two vehicles observed in camera  $C_i$  and  $C_j$  respectively, the similarity measure function  $S(O_{i,a}, O_{j,b})$  is defined as:

$$S(O_{i,a}, O_{j,b}) = \omega_t T(t_{i,a}, t_{j,b}) \cdot \omega_l L(l_{i,a}, l_{j,b}) \cdot \omega_a A(a_{i,a}, a_{j,b}) \quad (14)$$

where  $T(t_{i,a}, t_{j,b})$  denotes the time relationship of the two objects;  $L(l_{i,a}, l_{j,b})$  denotes their location relationship; and  $A(a_{i,a}, a_{j,b})$  is for their appearances. Each weight  $\omega$  is for each feature to control the reliability.

It is easy to construct the similarities of the parameters of time and location of the specific objects. They mainly depend on the object appearing time and the topology of the monitoring network. For appearances of the objects, vehicle types obtained by aforementioned proposed approach are main factors for them. Moreover, other features including LBP and HOG are combined together to make their appearance similarities. More details and experimental results about target association and MapReduce-based partitioning can be seen in [23].

## 4 Performance Evaluations

### 4.1 Test Environment

Test experiments are implemented on some Linux servers (Red Hat Enterprise Linux Server release 5.3) with OpenCV 2.4.3 [25]. One server has one eight-core CPU of Intel Xeon E5520 (2.27 GHz) with hyper-thread, and 16 GB memory. GCC 4.1.2 is used for compilation. For multi-camera tracking, a Hadoop cluster consisting of five aforementioned nodes is used.

### 4.2 Standard Model Database

For the standard model database in the system, 15 brands, amounting to 178 kinds of models, are built and recorded in the database. The information of model datasets is shown in Table 1.

**Table 1.** The Model Datasets

Brand	Amount	Brand	Amount	Brand	Amount
Audi	9	BMW	8	Honda	23
BYD	7	Peugeot	12	Buick	11
Volkswagen	37	Toyota	14	Ford	10
KIA	11	Nissan	8	Skoda	4
Hyundai	13	Chevrolet	4	Citroen	7

### 4.3 Evaluation of the Improvement in Distance Calculation

We use a square root (Hellinger) kernel instead of the standard Euclidean distance to measure the similarity between SURF descriptors, which leads to a dramatic performance improvement. This change can be implemented easily. It does not require any additional storage space for the distance calculation.

The time cost of distance calculations is shown in Fig. 2. The dramatic improvements in performance with the improvement method are shown in Fig. 3. These improvements come without additional cost, and additional storage since SURF features can be calculated with Hellinger online with a negligible processing overhead.

A standard image has 440 interest points. The line “Original” stands for the performance of non-modified SURF. The line “RefinedD” stands for the performance of SURF with improvement in the distance calculation. The line “RefineDP” stands for the performance of SURF with improvements in the distance calculation and the position constraint. Figure 2 shows that RefineDP has obvious advantage in computation time cost.

From Fig. 3, we can see that the number of matched points in RefineDP method is improved significantly, since the improvement reduces the dependency between the feature descriptors and the points with extreme values, while improve the sensitivity of the feature descriptor to non-extreme points.

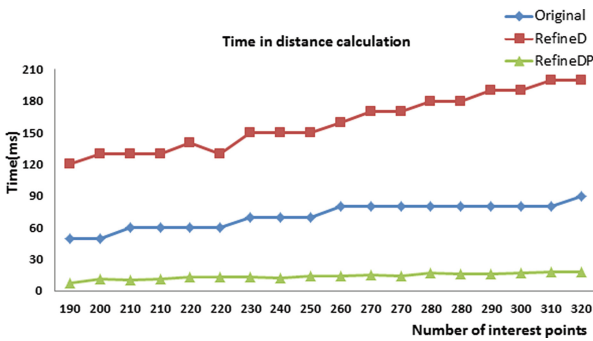


Fig. 2. Time cost of distance calculation

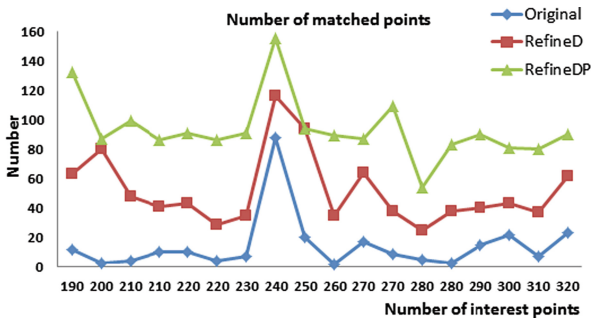


Fig. 3. Numbers of the matched points

### 4.4 Accuracy of Vehicle Recognition

This section shows the experimental results about the accuracy of vehicle recognition in different cases. The tested images not only include vehicle images downloaded from the Internet which are in good resolution, but also ones captured from some traffic videos which are in poor resolutions. In our system, clearer images can be extracted with more local features and get more precise result.

In Fig. 4, the *x*-axis stands for the numbers of feature points in the headlight edge images, and *y*-axis stands for the recognition accuracy. The lines with different legends stand for the numbers of feature points in front faces of vehicles. The numbers of feature points in headlight edge image and local features in front faces are in association with the image clarity. When the number of feature points in the headlight edge images is greater than 20, or the number of feature points in front faces is greater than 200, the accuracy is satisfactory.

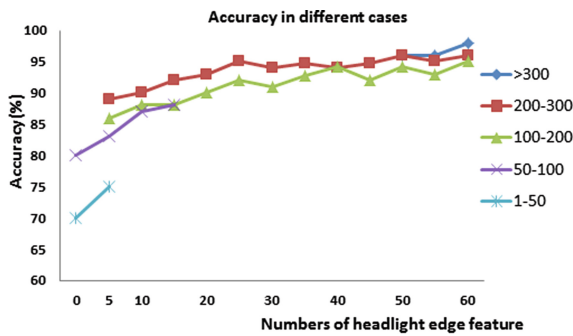


Fig. 4. The accuracy of vehicle recognition in different cases

### 4.5 Tracking Result of the Multi-camera Platform

To verify the effectiveness of the multi-camera platform, a batch of vehicle monitoring videos from 23 cameras in the real campus security monitoring system of Huazhong University of Science and Technology (HUST) are applied and analyzed (see Fig. 5(a)).



(a) The topology of multi-cameras

(b) The trajectory of one tracked vehicle

Fig. 5. Vehicle tracking of the multi-cameras platform

Figure 5(b) gives the trajectory of one tracked vehicle (white Honda SUV). It shows that the multi-camera platform based on Hadoop can track vehicle objects perfectly.

## 5 Conclusion

In this paper, a new recognition approach for vehicle types is designed and implemented. The vehicle type is distinguished by its front face image. The approach has been applied in a multi-camera traffic video monitoring platform, combining with LBP, HOG. The process of the whole camera network is parallelized based on sub-graph partitioning and Hadoop clustering. Experimental results indicate that the proposed modification in SURF can improve algorithm performance. The accuracy of the recognition can meet requirements of practical applications and the tracking result in the multi-camera system is satisfactory.

**Acknowledgment.** This work is supported by National Natural Science Foundation of China under grant No. 61133008, National High-tech Research and Development Program of China (863 Program) under grant No. 2012AA010905, and Scientific Research Foundation of Ministry of Education of China-China Mobile under grant No. MCM20122041.

## References

1. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 679–698 (1986)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006, Part I. LNCS*, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Sun, J., Gu, H.B.: Improved insensitivity to noise image edge detecting method based on the direction information. *J. Chin. Comput. Syst.* **27**(7), 1358–1361 (2006)
4. Anagnostopoulos, C.N., Giannoukos, I., Alexandropoulos, T., Psyllos, A., Loumos, V., Kayafas, E.: Integrated vehicle recognition and inspection system to improve security in restricted access areas. In: *Proceedings of 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*, pp. 1893–1898. IEEE Press, Piscataway (2010)
5. Sivaraman, S., Trivedi, M.M.: A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Trans. Intell. Transp. Syst.* **11**(2), 267–276 (2010)
6. Dlagnekov, L.: *Video-based Car Surveillance: License Plate, Make, and Model Recognition*. Ph.D. dissertation, University of California, San Diego (2005)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
8. Petrovic, V.S., Cootes, T.F.: Analysis of features for rigid structure vehicle type recognition. In: *Proceedings of 2nd International Forum on Mechanical and Material Engineering*, pp. 587–596, Trans Tech Publications Ltd., Zurich-Durnten (2004)
9. Ozcanli, O.C., Tamrakar, A., Kimia, B.B., Mundy, J.L.: Augmenting shape with appearance in vehicle category recognition. In: *Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pp. 935–942. IEEE Press, New York (2006)

10. Wang, Y., He, L., Velipasalar, S.: Real-time distributed tracking with non-overlapping cameras. In: Proceedings of IEEE Conference on Image Processing (ICIP 2010), pp. 697–700. IEEE Press, Piscataway (2010)
11. Archibald, R., Gelb, A., Yoon, J.: Polynomial fitting for edge detection in irregularly sampled signals and images. *SIAM J. Numer. Anal.* **43**(1), 250–279 (2005)
12. Sarkar, S., Boyer, K.L.: On optimal infinite impulse response edge detection filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(11), 1154–1171 (1991)
13. Yi, S., Labate, D., Easley, G., Krim, H.: A shearlet approach to edge analysis and detection. *IEEE Trans. Image Process.* **18**(5), 929–941 (2009)
14. Evans, A., Liu, X.: A morphological gradient approach to color edge detection. *IEEE Trans. Image Process.* **15**(6), 1454–1463 (2006)
15. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)
16. Juan, L., Gwun, O.: A comparison of SIFT, PCA-SIFT and SURF. *Int. J. Image Process.* **3**(4), 143–152 (2009)
17. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), vol. 2, pp. 511–517. IEEE Press, Los Alamitos (2004)
18. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vis.* **40**(2), 99–121 (2000)
19. Ling, H., Okada, K.: An efficient earth mover’s distance algorithm for robust histogram comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(5), 840–853 (2007)
20. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: Proceedings of 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol. 1, pp. 246–253. IEEE Press, New York (2006)
21. Pele, O., Werman, M.: A linear time histogram metric for improved SIFT matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 495–508. Springer, Heidelberg (2008)
22. Pele, O., Werman, M.: The quadratic-chi histogram distance family. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 749–762. Springer, Heidelberg (2010)
23. Jiang, W., Xiao, C., Jin, H., Zhu, S., Lu, Z.: Vehicle tracking with non-overlapping views for multi-camera surveillance system. In: Proceedings of the 15th IEEE International Conference on High Performance Computing and Communications (HPCC 2013), pp. 1213–1220. IEEE Press, Los Alamitos (2013)
24. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), pp. 2911–2918. IEEE Press, Los Alamitos (2012)
25. Pulli, K., Baksheev, A., Korniyakov, K., Eruhimov, V.: Real-time computer vision with OpenCV. *Commun. ACM* **55**(6), 61–69 (2012)