Acta Technica 62 No. 2B/2017, 117–128

Vehicle continuity tracking in traffic monitoring video based on mean hash[1]

Honghui Fan[2], Hongjin Zhu[2,4], Jiawei Li[2], Dongming Yuan[3]

Abstract. To further improve the vehicle detection and tracking accuracy, the detection and tracking of target vehicles in traffic surveillance video were discussed. Depending on the characteristics of the traffic video monitoring, Gaussian mixture model algorithm and the improved background subtraction algorithm were discussed and optimized to detect vehicle. And mean hash algorithm was used on the target vehicle continuously. Cloudy and foggy surveillance monitoring videos were utilized to verify the proposed proposal. At the same time, vehicle detection and tracking at different speeds were also verified. Through these experiments, it was verified that the target vehicle could adapt to the background model and update when it was traveling at low speed, and it was possible to maintain a relatively stable detection and tracking of the target vehicle. Verification experiments meant it had a beneficial effect on the influence of the weather environment.

Key words. Gaussian mixture model, background difference, vehicle detection, mean hash, vehicle continuity tracking.

1. Introduction

In recent decades, with the rapid development of information technology and greatly improve the performance of hardware, video monitoring system has been widespread deployment, around the city roads everywhere [1]. In the field of computer vision detection and tracking of moving targets has been a research hot spot, in robot navigation, intelligent transportation systems, and other fields have a wide range of applications. Tens of thousands of cameras can produce huge amounts of traffic monitoring data every day, through the traffic surveillance video data real-time

[1]The authors acknowledge the Changzhou Science and Technology Support Projects (CE20175026, CE20165028), IUR Prospective Joint Project of Jiangsu Province (BY2016030-05) and Liyang science and technology planning project(LA2016007).

[2]School of Computer Engineering, Jiangsu University of Technology, Changzhou, 213001, China

[3]Shenzhi Elevator Co., Ltd, Jilin, Liyang, 213300, China

[4]Corresponding author

analysis and processing, can build a lot of smart services: such as vehicle tracking, traffic control and traffic violation detection and so on [2, 3]. Therefore, the intelligent transportation system (ITS) becomes research and development hot spots in the world transportation vying [4, 5]. Although the intelligent transportation system has been got rapid development and a series of significant achievements have been made, but still have a certain gap with people's expectations. Collins and others using multiple cameras to realize intelligent traffic detection system, can identify the target vehicle state, and can be updated road information in real time [6, 7]. These research works have improved the intellectuality of the intelligent transportation system, but continuities of tracking of the same vehicle needs to be improved.

Continuous tracking of moving targets, the first thing to the moving object of video sequence detected in the frame, to provide initial targets information for follow-up. Moving target detection is mainly divided into two parts, background model and foreground detection. General methods of background model include mean filter method, median filter method and Gaussian mixture model (GMM) method [8–10]. Which of Gaussian mixture model method can detect the moving objects more completely, and adaptable to changes in the scene [11]. In the background after the success of the modelling, the use of background difference algorithm, can get the foreground image-moving targets. Using background subtraction algorithm would be influenced by external factors, such as light, weather and other factors, and if using the improved background subtraction algorithm, can eliminate these problems [12–14].

In this paper, Gaussian mixture model was used to the background model and update. Then improved background subtraction algorithm foreground detection to detect the moving vehicles. Finally, the hash algorithm was adopted to the same continuous tracking of the vehicle for monitoring video.

2. Background update and foreground detection principle

2.1. Background updating based on Gaussian mixture model

Gaussian model could represent the characteristics of each pixel in image [15]. For any pixel in the tth moments the observed values of $X_t = [r_t, g_t, b_t]^{\mathrm{T}}$. So the probability representation of the background is given by the equation

$$P(X_t) = \sum_{i=1}^{k} \omega_{(i,t)} \eta(X_t, u_{(i,t)}, \sum_{(i,t)}) . \tag{1}$$

In (1), $\omega_{(i,t)}$ is the weight of the ith Gauss components at the T moment. Quantity $\eta(X_t, u_{(i,t)}, \sum_{(i,t)})$ is the probability density function of the ith Gaussian component at time T, and the mean value is $u_{(i,t)}$. The covariance matrix is $\sum_{(i,t)}$. The current value of pixels is matched with the$K$-th Gaussian distribution of the Gaussian mixed model. If new pixel values and Kth Gaussian distribution match successfully, determine the pixels as the background, otherwise the prospects for points.

If a new pixel value with one of the Gaussian distribution satisfies $\left|X_{(i,t)} - u_{(i,t)}\right| \leq D\sigma_i$ (D representing custom parameters for the user, typically 2.5, and σ_i is variance), the weight w of the kth Gaussian distributions in the model is adjusted according to the equation

$$\omega_{(k,t)} = (1-\alpha)\omega_{(k,t-1)} + \alpha(M_{(k,t)})\,, \tag{2}$$

where, α is the learning rate, and its size determines the speed of the background update. If the detected pixel value $X_{(i,t)}$ matches the kth Gaussian distribution, then $M_{(k,t)} = 1$, the other values of $M_{(k,t)} = 0$. To match the success of the Gaussian distribution, the mean $u_{(i,t)}$ and variance $\sigma^2_{(i,t)}$ also need to be updated according to the formulae

$$\rho = \alpha\eta(X_t\,|u_k, \sigma_k\,)\,, \tag{3}$$

$$u_{(i,t)} = (1-\rho)u_{(i,t-1)} + \rho X_{(i,t-1)}\,, \tag{4}$$

$$\sigma^2_{(i,t)} = (1-\rho)\sigma^2_{(i,t-1)} + \rho \times (X_{(i,t-1)} - u_{(i,t-1)})^{\mathrm{T}} \times (X_{(i,t-1)} - u_{(i,t-1)})\,. \tag{5}$$

For matching of unsuccessful Gaussian distribution, the mean and variance remain unchanged when the background is updated. Therefore, the algorithm of mixed Gaussian model can update background accurately. The experimental results are shown in Figure 1.

Figure 1 is the result of a background update using a mixed Gaussian model algorithm in a complex environment. From Figure 1(A) to 1(F), there are shown respectively the numbers of frame 1, 50, 100, 200, 100, and 350. In the experimental results, we can see that the 350th frame is basically recovered from the original background. This result verifies the validity of the background update of the mixed Gaussian model algorithm

2.2. Foreground detection based on improved background difference algorithm

The common background difference algorithm is similar to the image subtraction operation, that is, the two images are point-to-point subtracted. Background different results are finalization processing. Mathematical expression is shown in (6). Symbol $R(i,j)$ is the difference image, $F(i,j)$ is the current frame, and $G(i,j)$ is the background image. Where the value of $R(i,j)$ is 1, the region is the target area to be detected. Symbol T is a gray scale threshold whose size is related to the sensitivity of the recognition target.

$$R(i,j) = \begin{cases} 1, & |F(i,j) - G(i,j)| > T, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The conventional background difference algorithm is simple to implement, but

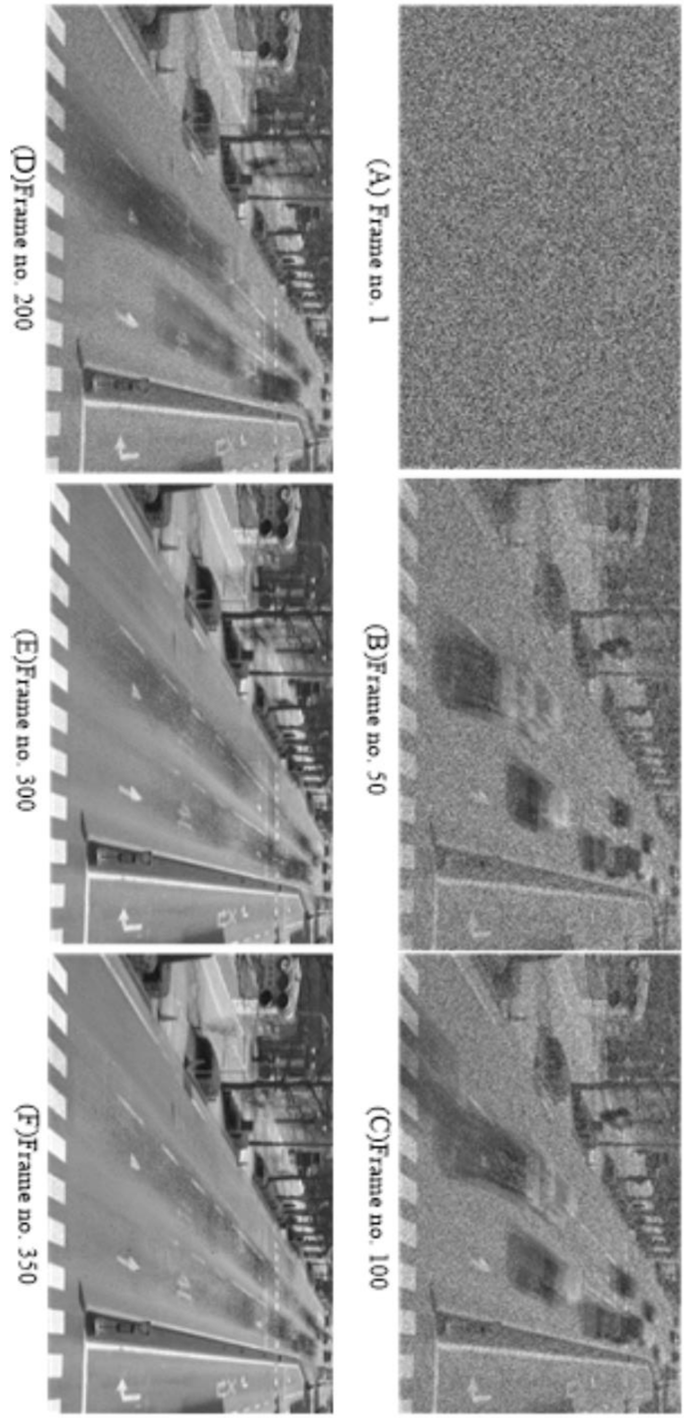

Fig. 1. The algorithm of the Gaussian mixture model realizes background update

because the gray scale threshold T is a fixed value, the segmentation effect is greatly affected in a complicated environment. Dynamic thresholds can solve this problem. It can automatically find the appropriate threshold for segmentation. Therefore, a background difference algorithm based on dynamic threshold is proposed, the basic principle is built on the two images have been obtained when the light changes, which can dynamically adjust the threshold. Therefore, the background difference algorithm could be improved based on the original algorithm to increase the dynamic threshold increment T. The corresponding mathematical expression is

$$\Delta T = \lambda \times \frac{1}{M \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |F(i,j) - G(i,j)| \tag{7}$$

$$R(i,j) = \begin{cases} 1, & |F(i,j) - G(i,j)| > T + \Delta T, \\ 0, & \text{otherwise}. \end{cases} \tag{8}$$

Symbol λ in (7) is the suppression factor. It could be accorded with the actual situation value. Product $M \times N$ represents the size of each processed image, the numerical result indicates the number of pixels in the detection area. Symbol ΔT reflects the overall environment changes. It changes with the change of the light. As a result, the background subtraction algorithm based on dynamic threshold could effectively inhibit the effects of the light changes. The experimental results as shown the Fig. 2.

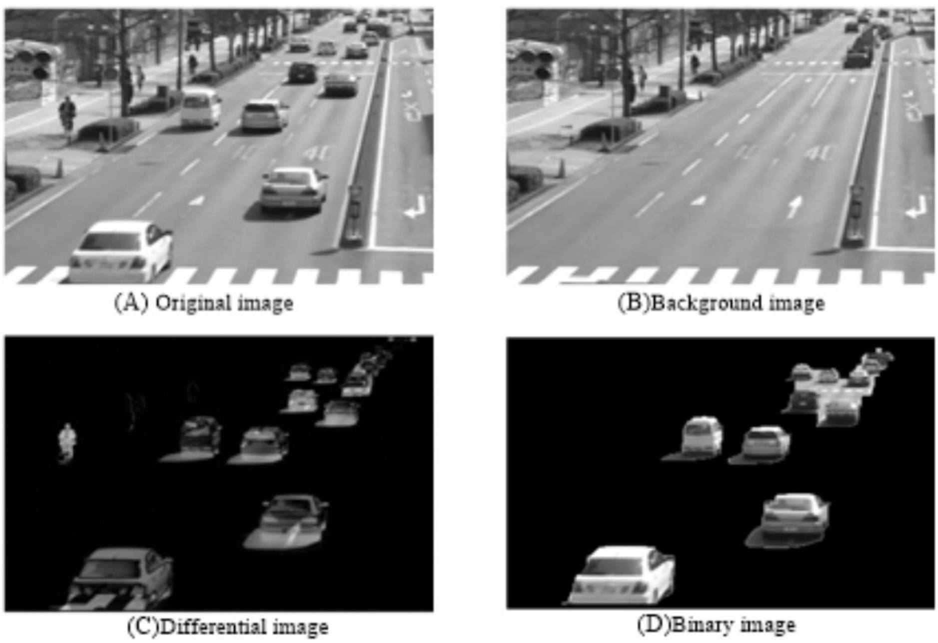

Fig. 2. Application of improved difference method in complex background

3. Persistence tracking based on mean hash principle

Hash algorithm maps a binary value of any length to a fixed-length binary value, which is called a hash value [16]. So hash value is a unique and extremely compact numerical data representation, so as long as the hash of a letter is changed, then the subsequent hash value will change. For two different inputs it is not possible to have the same hash value, so the hash value of the data can verify the integrity of the data.

Average hash algorithm is used primarily for a similar image search. Average hash of their main role is to each image to create a "fingerprint" string, then compare different fingerprint images. The result of the hash is very close, and it means that the image is very similar. Average hash algorithm is mainly using images of low frequency information, and the working process was illustrated in Fig. 3.

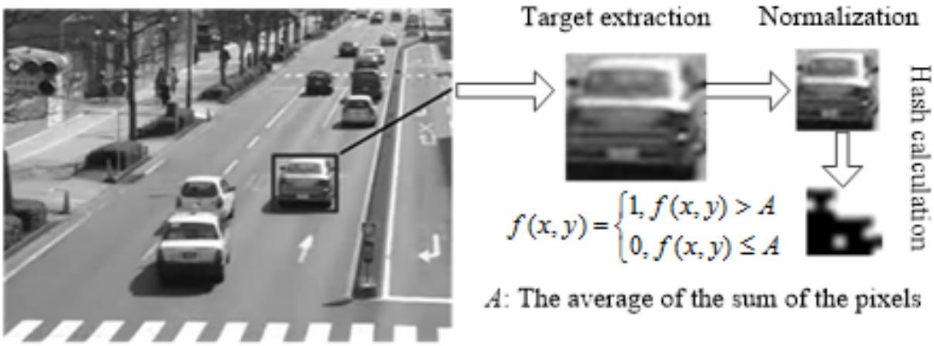

Fig. 3. Hash calculation process diagram

A picture is a two-dimensional signal. It contains the different frequency components, and the low frequency component is the region where the brightness changes are low and describe a wide range of information. The place of intense brightness change is the high-frequency component (such as the edge of the object), which describes specific details. This means that the high frequency can provide the details of images, and low frequency can provide a framework of images. In general, the frequency of large and detailed images is high, and small images lack the details of the image, so the miniature image seems to have a lot of low frequency components. The down sampling in image process, that has also the form of narrowing the pictures, is actually a process of loss of high-frequency information.

Hamming distance could compare the similarity of two images. The first is to calculate the hash of two images, that is, 64-bit 0 or 1, and then calculate the separate digits. If the Hamming distance value is 0, it means that the two images are very similar, if the Hamming distance is less than 5, it means there are some differences, if the Hamming distance is greater than 10, it shows the two images are different. As it can measure the similarity of two images, it should be used for target tracking. In the process of processing, as long as each frame to find the closest target location, according to the threshold of the same target can be matched. The principle of matching the template is similar, but the similarity measure used by the template match is the correlation of the two images.

Assume the original image size is $m \times n$, distribution range of pixel value is from 0 to 255, reduce image size is 8×8, so the pixel value distribution range is from 0 to 63. Assume that the image is $f(x, y)$. Then the specific steps are as follows:

Step 1: The quickest way to eliminate the high frequency and detail is to reduce size of the image. The image is reduced to 8×8 sizes, and it is comprised of 64 pixels.

Step 2: Convert 8×8 small images into gray-scale images.

Step 3: Calculate the average gray scale level in the all 64 pixels.

Step 4: The gray scale of each pixel is compared with the average. A mark greater than or equal to the average is 1 and the rest is 0.

Step 5: Combine the previous comparison results into a 64 bit integers, which is the hash key for this picture. The order of the combinations is a need, just make sure all the pictures are in the same order.

Thus, continuous tracking of the same vehicle based on the mean hash algorithm can be achieved. The target vehicle is to be tracked to find and then save, calculate its hash code, scan the next frame image and calculate the hash code of each scan window, and compare the target vehicle hash code. The distance from the smallest scanning window is the most similar to the target vehicle, which is the image of the target vehicle location. In order to expedite, only the image area around the previous frame is scanned. In order to accommodate changes in the target, it is also necessary to update the target vehicle to be tracked for each frame after successful tracking.

4. Experimental results

In order to verify the validity of the proposed method two groups of test experiment were used. Group 1 was under the condition of the same vehicle speed, and the vehicle in different weather conditions under the condition of continuous tracking. Group 2 was under the condition of the same weather environment, and the vehicle at unusual speed under the condition of continuous tracking. The experimental environment was based on Windows 7, visual studio 2010 and opencv2.4.9. We used foggy and low speed vehicle, cloudy and low speed vehicle and high speed vehicle to verify the vehicle tracking result. The three videos with a size of 240×360 pixels and the frame rate was 30. In order to quantitatively compare the performance of the proposed method, three metrics were used to evaluate the tracking performance.

The first set of video images (cloudy and low speed vehicle) had the total of 311 frames. The tracking results were shown in Fig. 4. A second set of video images (foggy and low speed) had the total of 313 frames, and tracking results were shown in Figure 5.

The first and second set videos were used for a group of comparative experiments. In the experiments, under the condition of vehicles running at low speed (less than 60 km/hour), the weather environment was different. In the cloudy and foggy weather for the same vehicle continuous tracking test, number of successful tracking was recorded. For this group of experiments, in the case of cloudy weather, the target vehicle in the first group of video appeared in a total of 188 frames, using

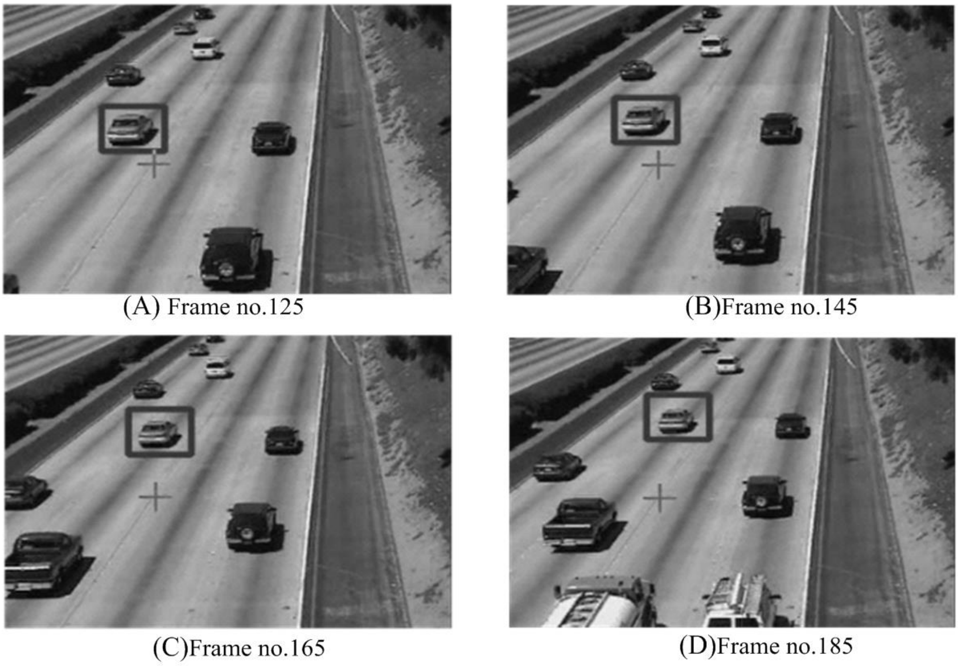

Fig. 4. Cloudy and low speed vehicle tracking experimental result

the proposed method in this paper could successfully track 181 frames. In the foggy weather, the target vehicle in the second group of video appeared in a total of 118 frames, target vehicle could be successfully tracked 101 frames.

Group 3 video images (cloudy and high speed vehicle) had a total of 353 frames, tracking results were shown in Fig. 6.

The first and second set video was used for a group of comparative experiments. In the experiments, under the condition of vehicles running at low speed (less than 60 km/hour), the weather environment was different. For the cloudy and foggy weather for the same vehicle and continuous tracking test, a number of successful tracings were recorded. For this group of experiments, in the case of cloudy weather, the target vehicle in the main group of video appeared in a total of 188 frames, using the proposed method in this paper could successfully track 181 frames. In the foggy weather, the target vehicle in the second group of video appeared in a total of 118 frames, target vehicle could be successfully tracked 101 frames.

In order to quantitatively compare the performance of the proposed method, using an accuracy rate, omission rate and error rate three metrics to evaluate the tracking performance. Accuracy rate relates to the target vehicle has success tracking number of frames and the target vehicle in the ratio of total number of frames. Omission rate refers to the target vehicle has no success tracking the number of frames and the target vehicle in the ratio of total number of frames. Error rate relates to the target vehicle has error tracking number of frames and the target

Fig. 5. Foggy and low speed vehicle tracking experiment result

vehicle in the ratio of total number of frames. The first group, the second and third group detailed performance parameters of the video image tracking results were shown in Table 1.

Table 1. Vehicle continuous tracking performance evaluation

Video sequence number	Evaluation factors	Accuracy rate (%)	Omission rate (%)	Error rate (%)
1	Cloudy Low speed	96.28	3.19	0.53
2	Fog Low speed	85.59	5.08	9.33
3	Cloudy High speed	55	36.5	8.5

As shown in Table 1, the first set of video tracking accuracy was highest, the third set of video omission rate was highest, the second set of video error rate was highest. The first and second groups of video tracking results could be seen that the weather environment affect the target vehicle tracking accuracy. Accuracy rate had fallen by about 10 %. In the first group and the third group of video tracking results it could be seen that vehicle speed affects the target vehicle tracking accuracy. Accuracy rate had fallen by about 40 %. The weather environment changes and vehicles run

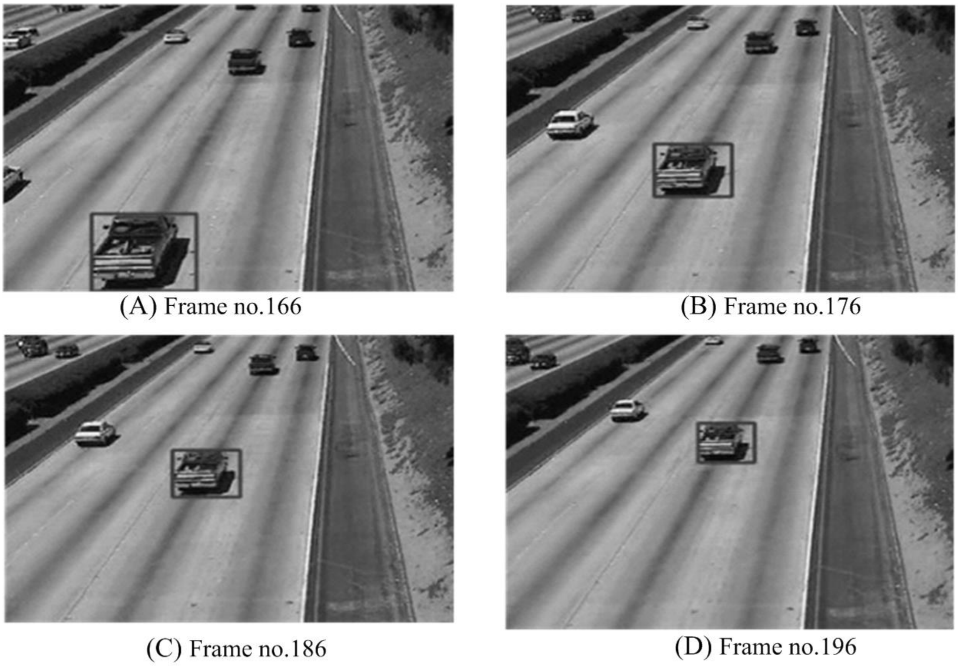

Fig. 6. Cloudy and high speed vehicle tracking experiment result

at low speed. Using the proposed method to keep track of the same vehicles, good results can be achieved. But there was no confirmation that the weather changes with vehicle speed would affect the vehicle tracking accuracy. By comparison, vehicle speed had a greater impact on the vehicle continuous tracking in proposed method.

5. Conclusion

In this paper, moving target vehicle detection and tracking were discussed. In the intelligent transportation system we used a Gaussian mixture model to update road background, and the improved background different algorithm was used to detect foreground. Simulation experiments were conducted in different weather conditions and vehicle speeds. Experiments showed that the proposed method could adapt to the background model and update when the target vehicle was running at low speed, and had a beneficial effect on the influence of the change of the weather environment. And the proposed method could remain relatively stable detection of the target vehicle and tracking.

However, when the vehicle was traveling at high speed, the accuracy of the target vehicle's continuous tracking using the proposed method was not satisfactory. At the same time, this study had not been compared with other algorithms, which were a future research direction.

References

[1] H. J. Zhu, H. H.,Fan, F. Y. Ye, S. S. Zhu, P. Z. Gan: *A novel method for moving vehicle tracking based on horizontal edge identification and local autocorrelation images.* DYNA - Ingeniería e Industria *91* (2016), No. 1, 61–68.

[2] X. J. Wang, F. Pan, W. H. Wang: *Tracking of moving target based on video motion nuclear algorithm.* Journal on Smart Sensing and Intelligent Systems *8* (2015), No. 1, 181–198.

[3] E. Taniguchi, H. Shimamoto: *Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times.* Transportation Research Part C: Emerging Technologies *12* (2004), Nos. 3–4, 235–250.

[4] H. H. Fan, H. J. Zhu: *Separation and extraction of moving vehicles in traffic images based on horizontal roof edges.* International Review on Computers and Software *7* (2012), No. 3, 1347–1351.

[5] rederick J. Engelbrecht, M. J. Booysen, G. J. van Rooyen, F. J. Bruwer: *Survey of smartphone-based sensing in vehicles for intelligent transportation system applications.* IET Intelligent Transport Systems *9* (2015), No. 10, 924–935.

[6] H. J. Zhu, H. H. Fan, F. Y. Ye, X. R. Zhao: *Improving vehicle detection accuracy based on vehicle shadow and superposition elimination.* Open Mechanical Engineering Journal *9* (2015), No. 1, 1039–1044.

[7] H. J. Zhu, H. H. Fan, S. Q. Guo: *Moving vehicle detection and tracking in traffic images based on horizontal edges.* TELKOMNIKA Indonesian Journal of Electrical Engineering *11* (2013), No. 11, 6477–6483.

[8] X. M. Zhang, Y. L. Xiong: *Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter.* IEEE Signal Processing Letters *16* (2009), No. 4, 295–298.

[9] S. Shrestha: *Image denoising using new adaptive based median filters.* Signal & Image Processing: International Journal (SIPIJ) *5* (2014), No. 4, 1–13.

[10] W. Li, S. Prasad, J. E. Fowler: *Hyperspectral image classification using gaussian mixture models and Markov random fields.* IEEE Geoscience and Remote Sensing Letters *11* (2014), No. 1, 153–157.

[11] T. Bouwmans, J. Gonzàlez, C. F. Shan: *Special issue on background modeling for foreground detection in real-world dynamic scenes.* Machine Vision and Applications *25* (2014), No. 5, 1101–1103.

[12] H. J. Zhu, T. Koga: *Face detection based on adaboost algorithm with local autocorrelation image.* IEICE Electronics Express *7* (2010), No. 15, 1125–1131.

[13] J. Zhang, J. L. Tang, H. H. Fan: *Variable length niche algorithm and application in fuzzy image edge detection.* Metallurgical & Mining Industry *7* (2015), No. 6, 368–372.

[14] H. J. Zhu, H. H. Fan: *Face detection based on adaboost algorithm with normalized differential images.* International Review on Computers and Software *7* (2012), No. 2, 845–849.

[15] H. H. Santosh Dadi, P. Venkatesh, P. Poornesh, L. N. Rao, A. Kumar: *Tracking multiple moving objects using gaussian mixture model.* International Journal of Soft Computing and Engineering *3* (2013), No. 2, 114–119.

[16] G. Wu, W. X. Kang: *Vision-based fingertip tracking utilizing curvature points clustering and hash model representation.* IEEE Transactions on Multimedia *19* (2017), No. 8, 1730–1741.

Received May 7, 2017