International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.2 (2014), pp.223-236
http://dx.doi.org/10.14257/ijsip.2014.7.2.21

A New Method for Trip-line Detection in Surveillance Video

ZuKuan WEI[1, 2], NaChen[2], YoungHwan OH [3],
HongYeon KIM [1] and JaeHong KIM[4]

[1]Cloud Computing Research Department, ETRI, Daejeon 305-700, Korea
[2]College of Computer Science & Engineering,UESTC,ChengDu 611-731, P.R. China
[3]Dept. of Info & Comm, Korea Nazarene Univ, Cheonan, 331-718, Korea
[4]Dept. of Computer & Info. Engi, YoungDong Univ, Chungbuk, 370-701, Korea
anlexwee@uestc.edu.cn,lanna20071055@163.com, kimhy@etri.re.kr,
yhoh@kornu.ac.kr, jhong@yd.ac.kr

Abstract

Trip-line detection is one of the very important research topics of intelligent monitoring category. This paper will present a trip-line detection system. In our trip line detection algorithm, Gauss model is used to model background, background differencing is used to detect object. Scanning the set of points on the line, if object is cross the line, for double-direction trip-line detection system directly alerts, but for single-direction trip-line detection, the moving direction of the object must be analyzed first. If the moving direction of the object violate passing rule, system directly alerts.

Keywords: *Trip-line detection; Gaussian Mixture Models; Background differencing; Target matching*

1. Introduction

The problem of trip-line detection is increasing the worldwide attention, particularly for those crowded environments which need strict security measures. Such as subways, airports, generator rooms, and other need strict regional control public venues.

We define trip-line to be a virtual line that users according to their requirements added on the video image. Therefore, trip-lines are virtual lines, is actually not exist. Crossing the trip-line means that moving object and the trip-line has intersection. Happening trip-line event means that exist moving object cross the double-direction trip-line or violate single-direction trip-line passing rules.

According to the actual application scenarios, trip-line detection is divided into two kinds. The first one is double-direction trip-line detection. In this case, ignore the direction of the moving target, as long as the target through the trip-line, the system will alarm immediately. A typical application is the modern intelligence community fence intrusion detection; Another one is a single-direction trip-line detection, when moving targets cross the trip-lines in k-th frame, the system should first find out the moving targets from the k-1 frames, then analyze its direction, and finally determine whether violate the passing rules. If targets violate the passing rules, trigger the alarm. In other words, double-direction trip-line is no need to consider the direction of moving targets, while single-direction trip-line needs to be considered. Thus, double-direction trip-line become undirected trip-line, single-direction trip-line also become a one-way trip-line.

Existing trip-line detection methods can be grouped into two categories. The first category is based on non-video images trip-line detection. These methods are mainly used as a signal

ISSN: 2005-4254 IJSIP

detector of some electronic devices, such as infrared generators, vibration cables, etc.. When a target enters the monitored area, the signal detector receives abnormal signals, and signals are sent to the remote host. These signals may be trigger the alarm. The advantage of this kind of method is good anti-interference, disadvantage is high cost. The second category is based on the video image trip-line detection. These methods are mainly using the camera for image acquisition, the use of intelligent video analysis technology to analyze video images, in order to detect trip-lines goals. Advantage of these methods is intelligent and humane, and the detection accuracy is high. Existing trip-line detection based on video image mainly adopts the method of target detection and target tracking. This method on the basis of the moving target is detected, track moving objects in the scene, in order to determine whether the target cross the trip-line.

This paper is to study trip-line detection based on the video image. Study of trip-line detection focuses on two issues. The first is target detection. The real environment is always very complex, such as light and the change of the weather will affect the target detection. In addition, because the hardware itself, image noise is inevitable. Therefore, how to accurately detect the moving targets in various scenarios still the main problems of current research. Second, cross trip-line judgment. On the basis of target detection, how to quickly determine whether trip-line and target have overlap pixel, also is a key problem to be solved. For single-direction trip-line detection, how to find out the matching targets in the previous frame also is a problem to be solved. For trip-line detection, multi-Gauss Model is used to model back ground. Moving objects are detected by background differencing. When foreground object is detected on trip-lines, for double-direction trip-line, system directly alerts; while for single-direction trip-line, the moving direction of the object must be analyzed at first. If the direction of the moving object forbidden passing rule, system alerts.

The rest of the paper is arranged as follows. Section 2 outlines the architecture of the proposed system. Section 3, target detection. Section 4, remove the shadow. Section 5, cross the trip-line detection. Section 6, detect trip-line event. Section 7 experimental results. Section 8 conclusions.

2. The Architecture of the Proposed System

Figure 1 shows the system diagram. The system has four main modules: (1) Target Detection; (2) Remove the shadow; (3) Cross the trip-line detection; (4) Trip-line event detection. Our paper mainly concentrated in the four parts. The system global structure introduced in this paper offers the following main contributions to detect single-direction trip-line and double-direction trip-line in surveillance videos. It is based on a Mixture of Gaussians background subtraction scheme. The main idea of this method is using mixture gauss models for modeling for any of the pixels in the image. After that background differencing is used to detect object. Scanning the set of points on the line, if object is cross the line, for double-direction trip-line detection system directly alerts, but for single-direction trip-line detection, the moving direction of the object must be analyzed first. If the moving direction of the object violate passing rule, system directly alerts.

3. Target Detection

The flow chart of target detection is shown in Figure 2. In this section, we first describe Gaussian Mixture Models, and then background subtraction. Remove the shadow module described in Section 4.

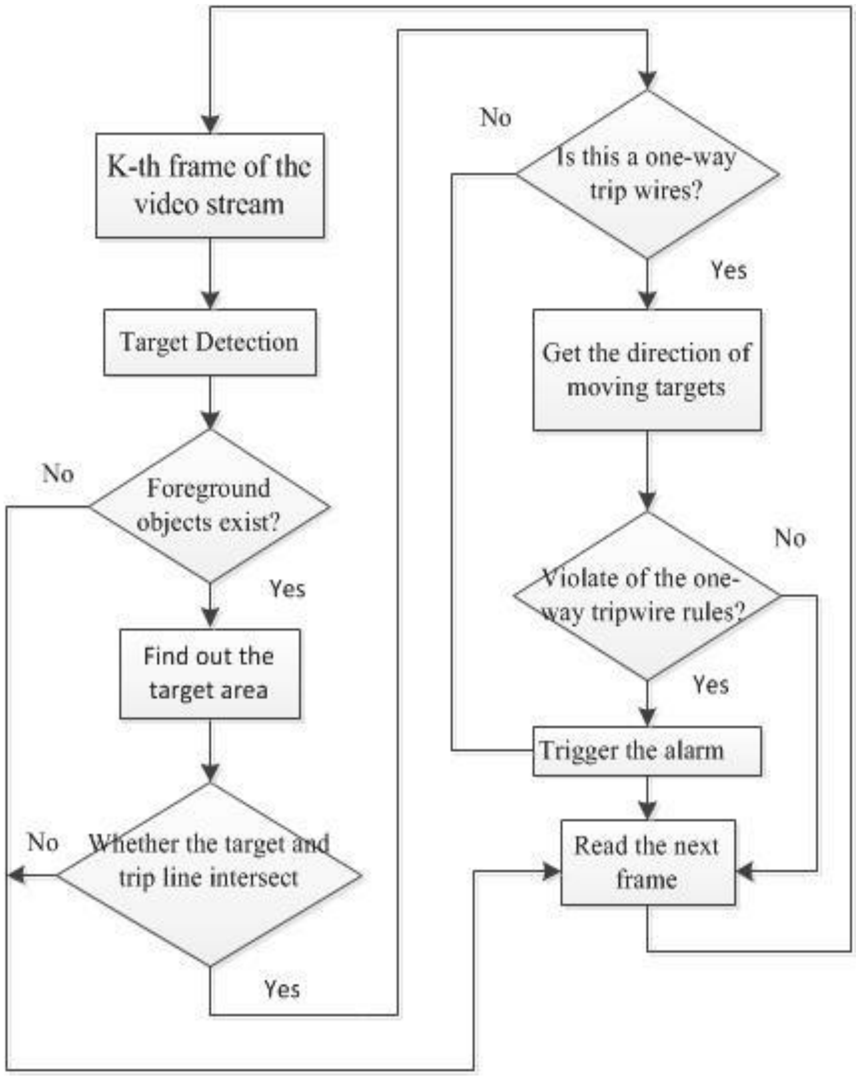


Figure 1. Trip-line Determination Diagram

3.1. Gaussian Mixture Models (GMM)

We employ the mixture of Gaussian method to build background model due to its robustness and efficiency. Stauffer and Grimson [1] introduced a mixture of K Gaussians (K ranges from three to five) to build the background model. For a pixel X at time t, the probability of the pixel can be written as follows:

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

Where K is the number of distributions, $\omega_{i,t}$ is an estimation of the weight of the i-th Gaussian in the mixture at timet, $\mu_{i,t}$ is the mean of the i-th Gaussian distribution, $\sum_{i,t}$ is the covariance matrix of the i-th Gaussian in the mixture at time t, and where η is a Gaussian probability density function.

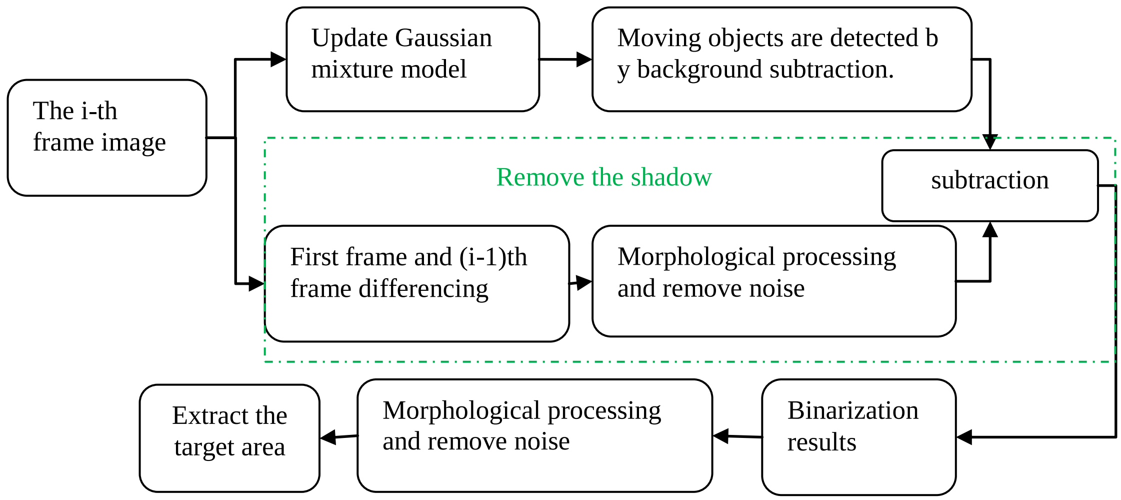


Figure 2. Target Detection Flow Chart

$$(X_{t,}\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T\Sigma^{-1}(X_t-\mu_t)} \tag{2}$$

where n is the dimension of the intensity at the pixel X. K is determined by the available memory and computational power. Currently, from 3 to 5 are used. For computational reasons, the covariance matrix is assumed to be of the form:

$$\sum_{k,t} = \sigma_k^2 I \tag{3}$$

The prior weights of the K distributions at time t,$\omega_{i,t}$are as follows

$$\omega_{k,t} = (1-\alpha)_{\omega_{k,t-1}} + \alpha\left(M_{k,t}\right) \tag{4}$$

Where α is the learning rate and $M_{k,t}$ is 1 for the model which matched and 0 forthe remaining models. We assume that the red, green and blue pixel values are independent and have the same variance. After Gaussians are ordered by the value ω/σ, the first B distributions are chosen as the background model, where

$$B = \arg\min_b \left(\sum_{k=1}^{b}\omega_k > T\right) \tag{5}$$

where T is a minimum separation of foreground and background. Theμandσ parameters for unmatched distributions remain the same. The parameters of the distribution, which matches the new observation, are updated as follows:

$$\mu_t = (1-\rho)_{\mu_{t-1}} + \rho X_t \tag{6}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t-\mu_t)^T(X_t-\mu_t) \tag{7}$$

Where

$$\rho = \alpha \eta (X_t \mid \mu_k, \sigma_k) \tag{8}$$

is the learning factor for adapting current distributions.

3.2. Background Differencing

Background differencing algorithm basic idea is that first extract accurate background model from the initial video sequence, and then the current video frame image with the background image do subtraction, in order to detect moving targets. After get the background image $B_k(x,y)$, with the current frame $I_k(x,y)$ and background $B_k(x,y)$ differencing, when the pixel value of the difference image $D_k(x,y)$ is greater than the thread TH, the detection is judged as the moving targets, like the expression given by Eq. (9) and Eq. (10)[2, 3].

$$D_k(x,y) = | I_k(x,y) - B_k(x,y) | \tag{9}$$

Where $I_k(x,y)$ represents current frame pixel grayscale value, $B_k(x,y)$ represents background frame pixel grayscale value, $D_k(x,y)$ represents the difference image. $D_k(x,y)$still needs to be binarization processing. The process is as follows:

$$R_k(x,y) = \begin{cases} 1, \text{if} D_k(x,y) \ge T \\ 0, \quad \text{else} \end{cases} \tag{10}$$

where T set for the threshold and $R_k(x,y)$ represents the binary image pixel grayscale value.

4. Remove the Sadow

Remove the shadow in trip-line detection is very important. For the trip wires detection shadows might lead to two serious problems. The first problem is that the shadow will merge two or more targets into a single target, as shown in Figure. 3(a). The second problem is the shadow will be treated as part of the goal. When the shadows and trip-line intersect, it will affect the detection results, as shown in Figure 3(b). We can see that from Figure 3(b), the girl did not crossing the trip-line, but her shadow area across the trip-line triggered the alarm.

Figure 3. Problems caused by Shadow. (a)Shadow Merge. (b) Shadow Cross the Trip-line, Trigger False Alarms

We use a background subtraction to remove the shadow. After we get moving objects through GMM and background differencing. We use S represents the foreground image of the i-th frame, $S_i(x,y)$ represents foreground image of the pixel(x, y) of the i-th frame, R represents the result of morphological processing and remove noise in Figure 4, $R_i(x,y)$ represents the binarization image of the pixel(x, y). In order to improve operational efficiency, we do not update the background imagehere, but put the first frame as a background image.

Mentioned in this article morphological processing and remove noise modules, we mainly use Gaussian Filter [4], Binarization, erosion[5], dilation[5] and smooth process[6]. Due to the space, here no longer describe these algorithms in detail.

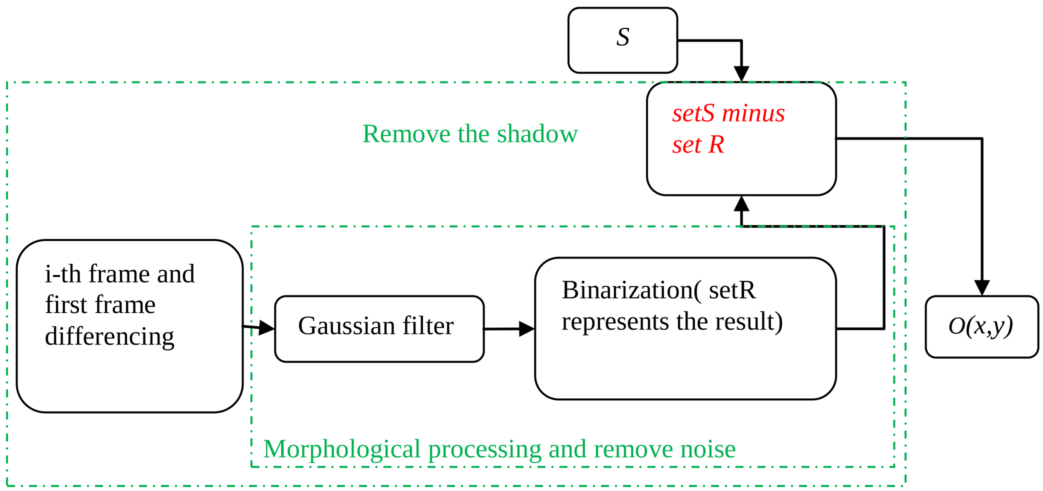


Figure 4. Remove the Shadow Flowchart

We use the first frame represents background image $B(x,y)$. After we get the background image $B(x,y)$, with the current frame $I_i(x,y)$ and background $B(x,y)$ differencing, then the image morphological processing and remove noise. Finally set R and S do set differencing, like he expression given by Eq. (11) and Eq. (12).

$$O_i(x,y)=\begin{cases}0, & \text{if } S_i(x,y)==0\\ 0, & \text{else if } S_i(x,y)!=0 \text{ and } R_i(x,y)==0\\ S_i(x,y), & \text{else}\end{cases} \tag{11}$$

$$S_i(x,y)\in S, R_i(x,y)\in R \tag{12}$$

Where$O_i(x,y)$represents the image after remove of the shadow. Figure 5 is remove the shadow comparison chart. Figure 5(a) shows the foreground contours without remove the shadow. Figure 5 (b) shows the foreground contours which use the above algorithm to remove the shadow.

(a) (b)

Figure 5. Remove the Shadow Comparison Chart: (a) Foreground Contours without Remove the Shadow. (b) Foreground Contours after Remove the Shadow

5. Cross the Trip-Line Detection

After remove the shadow, we get the image $o_i(x, y)$, then in order to reduce noise effect on detection results, do morphology processing and remove noise. Finally, extracting the foreground contours[7].

Then at the bottom of the foreground contours area designated height of 15 pixels of rectangular box, and set it to ROI. As shown in Figure. 6, the red and green box is the target area, the black box is the need to focus on ROI. At this point, the only need to pay attention to ROI regions and trip wires have overlapping part, without the need to focus on moving the entire region. Only when the ROI and trip-line intersect, we believe that happened ”trip-line ”event. The purpose of setting ROI can not only improve the detection accuracy but also can reduce the operation time and improve the efficiency of algorithm.

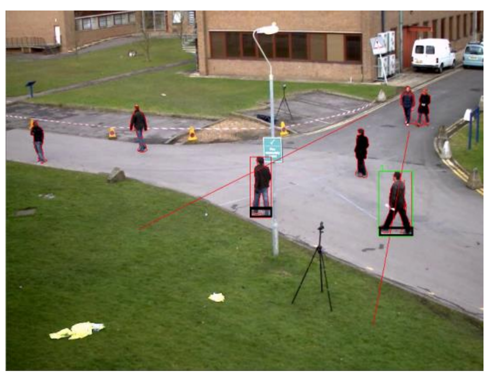


Figure 6. ROI and Target Area

Suppose L is the set of points on the trip-line, $R_{roi}$is the set of points on the ROI region. Considering the image noise, trip wires with target ROI area parts overlap condition is defined as : If the point (x, y)not only on L but also on R_{roi} , and overlapping points number is much bigger than the threshold TH, specific calculation is as follows:

$$\exists(x, y), s.t.(x, y) \in L \ \&\& (x, y) \in R \tag{13}$$

$$sum = \begin{cases} sum + 1, if \ (x, y) \in L and (x, y) \in R_{roi} \\ sum, \qquad else \end{cases} \tag{14}$$

$$\text{sum}\begin{cases} \geqq TH \Rightarrow trip\_lineevent \\ otherwise \Rightarrow ambigious \end{cases} \tag{15}$$

6. Trip-line Event Detection

System directly alert, if one object across the line in k-th frame, for double-direction trip-line detection, but for single-direction trip-line detection, the moving direction of the object must be analyzed first. If the moving direction of the object violate passing rule, system directly alerts.

In order to get the direction of moving targets, we should first find out the trip-line target in the (*k-1*)-th frames, then calculate the center of mass position of the trip-line target in *k*-th frames and the (*k-1*)-th, and then analyze the direction of target.

6.1. Target Matching

This paper uses the Euler distance matching method based on color histogram [8]. Specific process is as follows: $I_k(x, y)$ is the image where target cross trip-line,, $I_{k-1}(x, y)$ is the previous frame image before $I_k(x, y)$. $H_{i,k-1}(R,G,B)$ and $H_{i,k}(R,G,B)$ are the *i*-th color component histogram of this two frames. Euclidean distance is defined as follows:

$$d^2 = \sum_{i=0}^{255} \left(\frac{H_{i,k}(R,G,B)}{N_k} - \frac{H_{i,k-2}(R,G,B)}{N_{k-2}}\right)^2 \tag{16}$$

where N_{k-2} and N_k are the total number of pixels of $H_{i,k-2}(R,G,B)$ and $H_{i,k}(R,G,B)$. When $d^2 \leqq TH$, considered two goal is the same goal, the smaller thed^2 represents the target matching degree is higher.

6.2. The Direction of Target

R_k is a rectangular target area in $I_k(x, y)$, R_{k-1} is for R_k Euler distance matching results. $Pmass_i(x, y)$ is the barycentre of R_i .

$$\text{Pmass}_i(x, y) = \left(\frac{X_i + Wid_i}{2}, \frac{Y_i + Hei_i}{2}\right) \tag{17}$$

(X_i, Y_i) is the R_i coordinates of the upper left corner, Wid_i and Hei_i is is the width and height of R_i. Get the direction of the moving target by the following formula:

$$Ang_i = \text{arctan}\ gent((Y_k - Y_{k-1}), (X_k - X_{k-1})) \tag{18}$$

Ang_irepresents the direction of the moving target.

7. Experimental results

In this section, experimental results of the algorithm are carried out on several sequences from AV video library [9]. Experiment was divided into two parts: One part is single-direction trip-line detection and the other part is double-direction trip-line detection. Experiments were carried out in a public outdoor environment, the following one degrees of complexity is considered:

(1) Low complexity: Up to four moving objects need to detect.

(2) Medium complexity: Up to eight moving objects need to detect.

Assess the performance of the algorithm we use PED, PAT and Score [10]. The PED score represents the ratio of real alarms in the ground truth that were successfully detected by the module to the total number of alarms in the round truth. See Eq. (19). The PAT score represents the ratio of alarms that correspond to real alarms in the ground truth to the total number of alarms detected by the module Eq. (20). The Score provide a very good metric to describe how well the algorithm does overall. Thus, we define an overall score as Eq. (21).

$$PED = \frac{Number\ of\ \mathrm{Real}\ Alarms\ Detected}{Number\ of\ \mathrm{Real}\ Alarms} \tag{19}$$

$$PAT = \frac{Number\ of\ \mathrm{Real}\ Alarms\ Detected}{Total\ Number\ of\ Alarms\ Detected} \tag{20}$$

$$Score = \frac{(1-\log 2(x))TrueDetected}{TotalDetected - TotalEvents(long\ 2(x))} \tag{21}$$

In Eq. (21), TotalEvents is the number of alarms in the ground truth data, Totaldetected is the total number of alarms issued, TrueDetected is the number of true positives, and x is the relative importance we wish to give to the PED score over the PAT score. For the experiment reported here, we use x = 0.8 because we believe finding true alarms more important than some false positives.

7.1. Single-direction Trip-Line Detection

We use red line represents trip-line, with green and blue lines represent the two directions perpendicular trip-line. As shown in Figure. 7, passing rules of single-direction trip-line detection areas follows: If $|\mathrm{Ang_i} - \mathrm{Ang_{green}} \leqq 90|$, with the green line drawing $\mathrm{R_i}$; If $|\mathrm{Ang_i} - \mathrm{Ang_{blue}} \leqq 90|$,with the blue line drawing $\mathrm{R_i}$. $\mathrm{Ang_{green}}$ represents the direction of green line,$\mathrm{Ang_{blue}}$represents the direction of blue line. As shown in Figure.7 (b) and Figure.7 (c).

(a) (b) (c)

Figure 7. (a)Single-direction Passing Rule. (b)Moving Targets Direction Consistent with the Green Line, Moving Targets Marked with Green Lines. (C)Moving Targets Direction Consistent with the Blue Line, Moving Targets Marked With Blue Lines

Here we only show experimental results of the 61-th frame. As shown in Figure 8, every frame in our algorithm should be remove the shadow as shown in Figure 8 (a) and Figure 8 (b) (Figure 8 (a) is 61-th frame without remove the shadow, Figure 8 (b) is 61-th frame after remove the shadow) $\mathrm{T_i}$ cross the trip-line in Figure 8 (c), We need to determine whether trip-line event occurs. Find the matching result in Figure 8 (j), the result is shown in Figure 8 (l), then via the matching result calculate the direction of moving target$\mathrm{T_i}$. If$|\mathrm{Ang_i} - \mathrm{Ang_{green}} \leqq 90|$, with the green line drawing$\mathrm{T_i}$; if $|\mathrm{Ang_i} - \mathrm{Ang_{blue}} \leqq 90|$, with the blue line drawing $\mathrm{T_i}$.

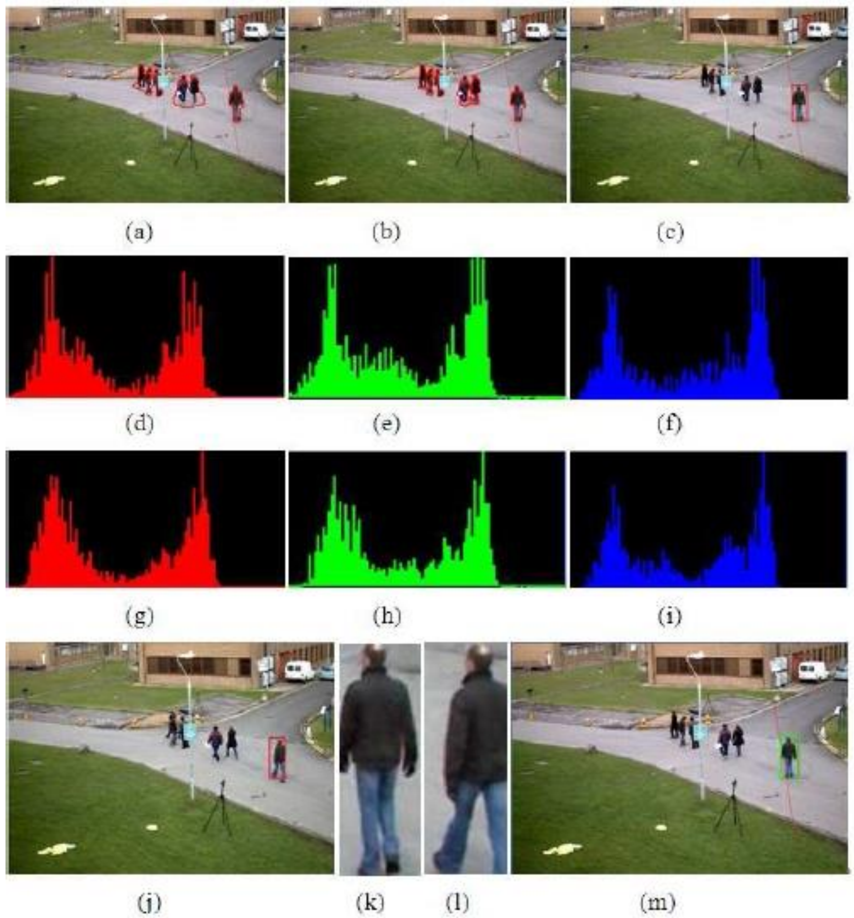


Figure. 8. (a) 61-th Fame without Remove the Shadow. (b) 61-th Frame after remove the Shadow. (c)$\mathrm{T_i}$ Cross the Trip-line in 61-th frame. (d)R Component of Ti in 61-th Frame. (e) G Component of $\mathrm{T_i}$in 6-thframe.(f)B Component of $\mathrm{T_i}$ in 61-th Frame. (g)R Component of $\mathrm{T_i}$ in 60-th frame. (h) G Component of Ti in 60-th Frame. (i) B Component of $\mathrm{T_i}$ in 60-th frame. (j)Matching Result in 60- th Frame. (k) The screenshot of $\mathrm{T_i}$ in 61-th Frame. (l) The Screenshot of Ti in 60-th Frame. (m) Trip-line Event occurs in 61-th Frame

From Figure 9 (a) we can see that, in low complexity situation, PED and PAT are almost always on the rise until then reaches the highest point they will remain unchanged. PED steady state value is 98.3%, PAT steady state value is 94.9%. Score is overall growth, but the local has a small decline. Score steady state value is 89.5%. As shown in Figure 9 (b), in medium complexity situation, they all appear the volatility, but can grow to a relatively stable state. Compared with the data in Figure 9 (a), these three values in the steady state value have decreased to some extent. PED steady state value is 92.1%, PAT steady state value is 89.5% and Score steady state value is 85.5%.

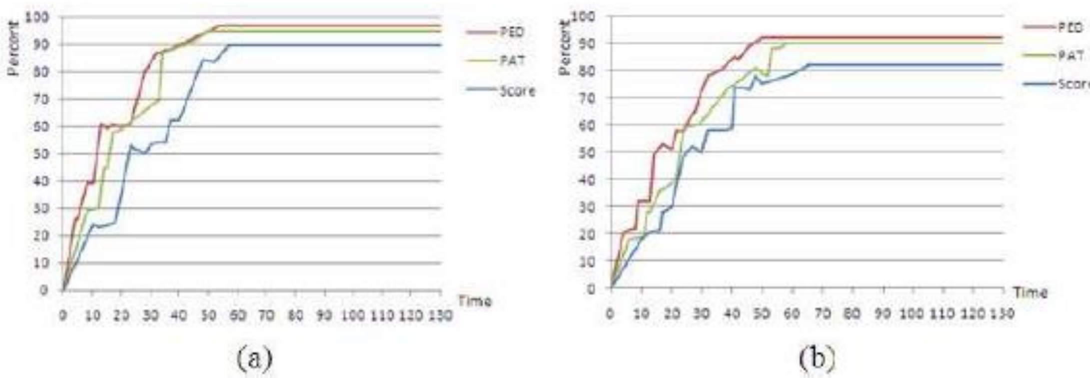


Figure 9. Result Plots for the Test Sequences

7.2. Double-Direction Trip-Line Detection

We use red color draws trip-line. If a target T_k crosses the trip-line, with the yellow line drawing T_k, system triggered alarm. As shown in Figure 10 (a).

From Figure 10 (b) and Figure 10 (c) we can see that, the accuracy of double-direction detection is higher than single-direction detection. Because double-direction detection needn't to judge the direction of the moving target. In low complexity situation, PED and PAT are almost always on the rise until then reaches the highest point they will remain unchanged. PED steady state value is 99.2%, PAT steady state value is 98.1% and Score steady state value is 95.2%. In medium complexity situation, PED steady state value is 98.4%, PAT steady stat value is 97.1%, Score steady state value is 93.9%.

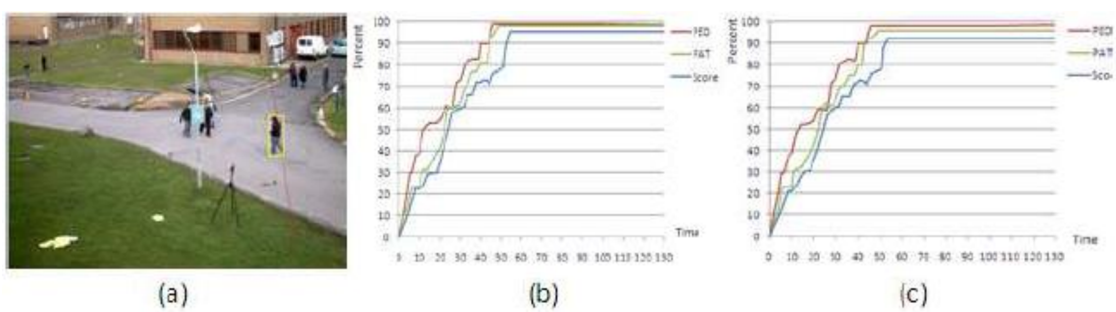


Figure 10. Cross the trip-line

8. Conclusions

In this paper, we have presented a new method to robustly and efficiently detect trip-line event. The mixture of Gaussians BGS method is employed to detect foregrounds by using the Gaussian mixture model. If foregrounds cross the trip-line, for double-direction trip-line detection, system directly alerts, but for single-direction trip-line detection, the moving direction of the object must be analyzed first. If the moving direction of the object violate passing rule, system directly alerts. From part 7 we can see that, our algorithm detection accuracy is high, and the performance is relatively stable. In the future work, we will focus on solving the problem of the detection efficiency of the fast moving target.

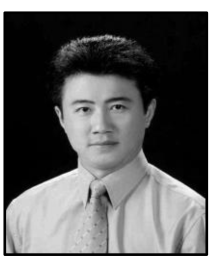
Acknowledgements

This paper can be successfully completed with the help of many people. No matter what the process of accessing to information or exploring knowledge, they have given me a lot of help. Without their help, this study would not have been completed so successfully.

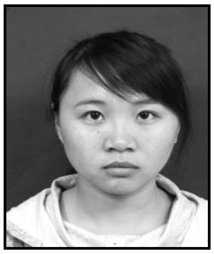
References

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition V2, **(1999)**, pp. 2246-2252.
[2] X. H. Zhou, B. Liu and H. Q. Zhou, "A motion detection method based on symmetrical differencing and background subtraction", The computer simulation, vol. 4, **(2005)**, pp. 117-120.
[3] H. Wen and T. Hong, "Based on background subtraction and symmetrical difference target detection", Micro computer information, vol. 9, **(2007)**, pp. 99-101.
[4] H. Yang and X. Y. Wang, "Image denoising using bilateral filter and Gaussian scale mixtures in shiftable complex directional pyramid domain", Computers and Electrical Engineering, **(2011)**, pp. 656-668.
[5] X. Cheng, H. Zhao and B. Liu, "Fast image dilation/erosion with cooperation of multiple neighbors", Pattern Recognition (CCPR), **(2010)**, pp. 1-4.
[6] S. Sari, H. Roslan and T. Shimamura, "Noise estimation by utilizing mean deviation of smooth region in noisy image, Computational Intelligence", Modelling and Simulation (CIMSiM):2012 Fourth International Conference, **(2012)**, pp. 232-236.
[7] L. D. Cohen, "Multiple Contour Finding and Perceptual Grouping using Minimal Paths", Journal of Mathematical Imaging and Vision, vol. 14, **(2001)**, pp. 225-236.
[8] D. Yue and K. L. S. Li, "Color Histogram Matching Algorithm in Video Detection", Computer and Information Technology, vol. 5, **(2008)**, 004.
[9] Video Data, For further detail, please visit our website, http://www.cvg.cs.rdg.ac.uk/PETS2009/data.html.
[10] N. Bird, S. Atev and N. Caramelli. "Real time, online detection of abandoned objects in public areas", Robotics and Automation, Proceedings 2006 IEEE International Conference on, **(2006)**, pp. 3775-3780.

Authors



Zukuan Wei, received a B.S. degree in Computer Science and Engineering from Zhejiang University, P.R.China in 1990. He then received M.E.degree in Computer Science and Engineering from Chongqing University of Posts and Telecommunication, P.R.China in 1997, and Ph.D. degree in Computer Science and Engineering from INHA University, Korea in 2001 respectively. He was a visiting scholar of Computer Science Department at DUKE University, U.S.A in 2006 and researcher in Dept. of Cloud Computing, Electronics and Telecommunications Research Institute, Korea, 2013. He is now an associate professor in the Department of Computer Engineering at University of Electronic Science and Technology of China (UESTC). His research interests include spatial database system, large storage system, cloud computing and mobile computing etc.



Na Chen, was born in 1988 and is now studying in the Department of Computer Engineering at University of Electronic Science and Technology of China (UESTC). Her research interests include pattern recognition, image and information processing.

Younghwan Oh, received the M.S. degrees in computer science from Inha University in 1997 and the Ph.D. in computer science from Inha University in 2001. His research interests included in stream data processing and data fusion in wireless sensor network. He is a professor in Korea Nazarene University from 2002.

HoneYeon KIM, e-mail:kimhy@etri.re.kr; 1992: Inha University, B.S. in Dept. of statistics; 1994 : Inha University, M.S. in Dept. of computer engineering; 1999:Inha University, ph.D. in Dept. of computer engineering; 1999~Present senior researcher in Dept. of Cloud Computing Research, Electronics and Telecommunications Research Institute; Study Interests : Storage system, file system, Database system.

Jae-Hong Kim, received a B.S. degree in Computer Science from Inha University, Korea in 1988. He then received both M.E. and Ph.D. degrees in Computer Science and Engineering from Inha University, Korea in 1990 and 1994, respectively. He was a visiting researcher of Information Processing Center at Hiroshima University in 1991. He is now an associate professor with the Department of Computer Engineering, and the dean of the School of Science and Engineering at Youngdong University. His research interests include query optimization, geographic information system, and multimedia database system.