Proceedings of the 7th Asian Control Conference,
Hong Kong, China, August 27-29, 2009

A real-time motion detection algorithm for traffic monitoring systems based on consecutive temporal difference

Zhen Yu, Yanping Chen
Dept. of Automation
Xiamen University
Xiamen, China
Yuzhen20@xmu.edu.cn, chen_yanping1@163.com

***Abstract*—Moving objects detection is a basic task in video analysis and applications. Many algorithms have been proposed to detect moving objects under different situations in the past decades. In this paper, we propose a new method to detect moving objects in a non-stationary, complex background for an automatic traffic monitoring system. As a pre-treatment, the square neighborhood algorithm is adopted to compensate the disturbance caused by shaking of camera. Then, an improved temporal difference method is applied to obtain the moving areas. Some post-treatments are used to optimize the detection by eliminating noise from the moving areas. The proposed method is excellent in real-time performance because it requires only a little memory and computation. Experiment results show that this method can detect the moving objects efficiently and accurately form the video recorded by a shaking camera with changing background and noises.**

***Keywords*—Motion detection, traffic monitoring, mixture of Gaussians, square neighborhood matching.**

I. INTRODUCTION

Intelligent transportation system provides an attractive alternative to the traditional traffic systems, which depend almost on the facilities of the system for traffic circulation and safety. A video camera, coupled with computer vision techniques, makes up of a video-based intelligent transportation system [2]. Detection of moving objects is the first relevant step in this system. To meet the special requirements of efficiency and accuracy of a successful video-based system, moving objects detection algorithm should be characterized by some important features, such as accuracy, real-timeness, etc. The accuracy of detection is a basic requirement of the system. In general, the accurate detection is time-consuming. Moreover, a real time system ensures that the detection information is provided in time, and the management commands from the control center are responded timely. In fact, a precise moving object detection method makes tracking more reliable and faster, and supports correct classification, which is quite important for a system to be successful [3].

During the past decades, researchers in vision technique have already proposed various algorithms for detecting moving objects, such as, consecutive temporal difference (consecutive frames subtraction) [9,10,11,12], optical flow approach [1,4, 5,13,14], and background subtraction [6, 7, 8,15,18,19], etc. Among these methods, background subtraction algorithms are most popular, because they are relatively simple in computing in a static scene. However, the background is assumed to be static in this method. Thus, shaking cameras, waving trees, lighting changes are quite probable to cause serious problems to a background subtraction model [1]. In addition, a successful background subtraction method needs to model the background as accurate as possible, and to adapt quickly to the changes in the background. These requirements add extra complexity to the computation of the model and make a real-time detection difficult to achieve. Optical flow approach is quite excellent because it can detect the moving objects independently and it works very well in changing environments, even in the absence of any previous information of the background. However, the computational cost of the approach is very expensive, which makes it very difficult to be applied in a real-time system. And, this approach is quite vulnerable to disturbs, such as the headlights of the vehicles. Thus, it is not fit for the traffic control system.

Temporal difference is the simplest method to extract moving objects and robust to dynamic environments. However, it easy to cause small holes and cannot detect the entire shape of a moving object with uniform intensity. Also, any changing elements in the background can be easily classified as the foreground by temporal difference. In this paper, we propose an improved temporal difference approach. Three consecutive frames are used for computing. The pre-treatments are applied to compensate the camera motion in the input frame. Moreover, the post-treatments are employed to optimize the differential results by filling the small holes and removing uninteresting moving objects.

The rest of the paper is organized as follows. Section II presents the main results of the paper. First, an overview of the framework of the proposed method is described. Then, the details of the implementation are presented, including camera motion compensation, improved consecutive temporal difference, small holes fulfillment and uninteresting motions removal. The efficiency and accuracy of the proposed method are illustrated by some experiment tests in Section III. Finally, Section IV concludes the paper with some remarks.

II. THE FRAMEWORK OF THE PROPOSED METHOD

Figure 1 illustrates the block diagram of our algorithm. The camera motion of the input frame is compensated and consecutive temporal difference is performed to extract moving areas from the image. The moving areas include the target areas (moving objects we are interested in) and some uninteresting motion areas (the moving background). Post treatments are used to optimize the detection by filling the small holes in the detected objects and removing uninteresting motions. Since shadows won't have large change between two consecutive frames and little change of shadows to be detected can be removed by the post treatments, shadows have little effect on the accuracy of detection. Thus, they are not handled here to improve the efficiency of this method. To perform this method in real-time and with high accuracy, we design every block carefully. Since the consecutive temporal difference approach requires no background model and little memory, this approach is quite efficient and accurate in that it has a low computational cost and it adapts quickly to the changes of the background.

First, the input frame is compensated by using a camera motion compensation algorithm, which takes some spots in the frame as basic pixels and estimates the motion of the camera with a square neighborhood matching method. The whole input frame is then adjusted to make up the motion of the camera. The square neighborhood matching method will be fully described in the following subsection. It shows how we record the deviation of the image.

Second, an improved consecutive temporal difference approach is used to quickly obtain moving areas of the input frame. This approach makes use of three consecutive frames. The three frames are divided into two groups. The first group includes the two previous frames (the two consecutive frames that go before the input frame), while the second group includes the input frame and the frame before it. By subtracting the two groups separately, we get two results of different areas from the two subtractions. Since the intersection of the two results is just the very part of the moving area in the frame previous to the input frame, we can obtain the moving areas by subtracting the second results by the intersection of the two difference frames.

Third, post treatments are used in this phase to fill the small holes in the detected objects and to remove uninteresting motions from the moving areas obtained in the second phase. Since shadows are not handled in this method, post treatments deal mostly with the two problems mentioned above. The math morphology techniques are used. The moving areas are expanded and corroded first to fill the small holes and to get connecting fields, shaping entire objects. Then the area of the connecting fields are computed and segmented to decide whether the field belongs to the target objects or uninteresting motions. In this way, we shape the entire object by removing the small holes and eliminate uninteresting motions from the detection, leaving only the objects that we are interested in.

A. Camera motion compensation

Since temporal difference approach is quite sensitive to any changes between two consecutive frames, a shaking camera may cause a breakdown to the whole approach, for any pixel in the new frame may be different from the one in its original position in the last frame, due to the motion of the whole image. To avoid errors like that, the camera motion of the input frame must be compensated. Camera motion compensation uses a new method, which takes some symbolic spots as the basic pixels for matching, coupled with a square neighborhood matching algorithm to decide the motion of the image.

For a given environment, first we choose some special spots to act as basic pixels. The spots chosen should scatter in every corner of the frame to ensure an accurate compensation of the whole frame. For every basic pixel in the chosen spots, first we define a neighbor area for it, normally a square of N×N pixels for the pixel in the center of the square (N is 10 in our paper), with the basic pixel in the center of the square. For each square, we establish a coordinate with the center pixel in the square as the origin, and define X-component, Y-component, and XYL-component, XYR-component for the line going through the origin and 45 degrees away from the Y-component on the left and on the right separately (XYL-component for the left and XYR-component for the right), as shown in figure 2. For the basic pixel in the center of the square, all the pixels that are on the lines of the four components in the frame are matched, by computing the pixel intensity. The locations of the pixels that match the

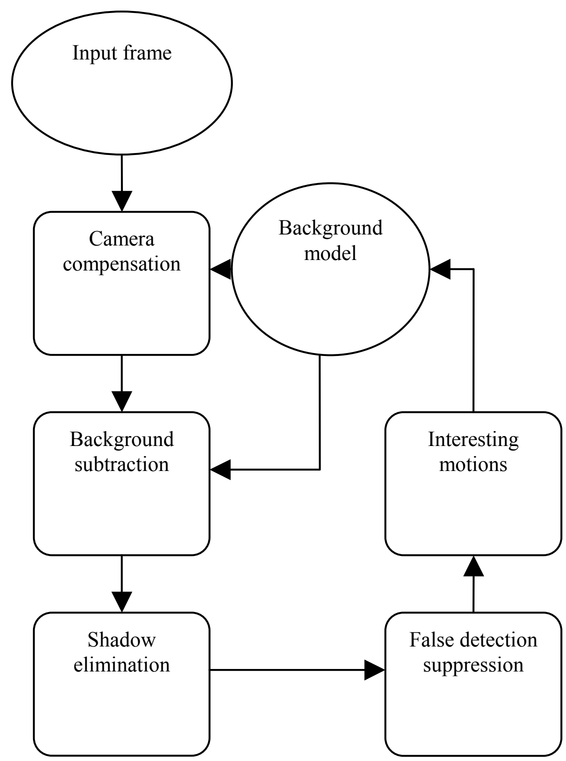


Figure 1. Block diagram of the proposed algorithm

basic pixel are recorded for further use. Theoretically, estimating the motion of a pixel, we can obtain the motion displacement of the image. But in practice, we often choose more pixels to calculate to reduce errors. How many spots should be chosen to calculate depends on the sensitivity requirement of the detection (we take about 100 pixels here).

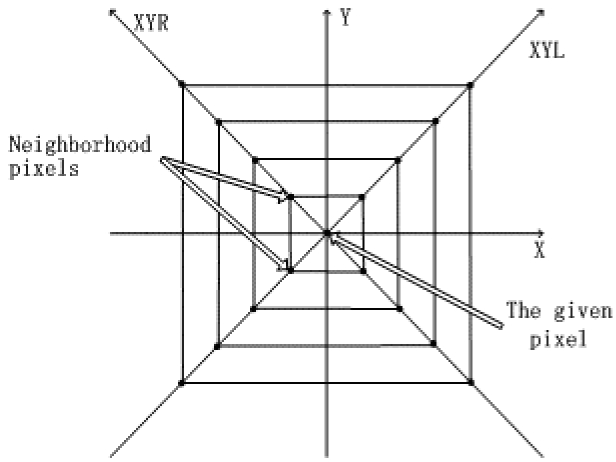


Figure 2. Square neighborhood matching model

When more than one basic pixel are used to match to their neighbor points, and each pixel has more than one matching pixels, the displacement of the motion can be calculated by computing the number of matching pixels for each location in the corresponding squares. In other words, for each location in the square (e.g. the positive point 2 in the Y-component of the square), we calculate the times when the given pixel matches to the pixel at this location, then we choose the one with the most matching times as the displacement of the motion, and adjust the image wholly to compensate the displacement at last.

B. Consecutive temporal difference

After the pretreatment of the input frame, we begin to detect the moving areas. In this paper, we obtain the moving area of the input frame by performing three subtractions of the consecutive frames to remove the background in the input frame. The implementation of the method is described in the following.

Let $F_I(k-2)$, $F_I(k-1)$ and $F_I(k)$ represent three consecutive frames, with $F_I(k-2)$ comes earliest and $F_I(k)$ (representing the k^{th} input frame) comes latest. First, we divide the three frames into two groups, with $F_I(k-2)$ and $F_I(k-1)$ a group, $F_I(k-1)$ and $F_I(k)$ another group.

First, we perform subtractions of the two groups separately to get two difference frames. Suppose a frame has m rows and n columns, and the value of the pixel at position (i,j) in the k^{th} input frame is $f(k,i,j)$, $F(k)$ can be represented as:

$$F(k)=\begin{pmatrix} f(k,1,1) & \dots & f(k,1,n) \\ \vdots & & \vdots \\ f(k,m,1) & \cdots & f(k,m,n) \end{pmatrix}$$

Then we obtain two difference frames by subtracting the two groups separately:

$$\begin{cases} D(k)=F(k)-F(k-1) \\ D(k-1)=F(k-1)-F(k-2) \end{cases}$$

The subtraction here is defined as:

$$\begin{cases} D(k)=\begin{pmatrix} f(k,1,1)-f(k-1,1,1) & \dots & f(k,1,n)-f(k-1,1,n) \\ \vdots & & \vdots \\ f(k,m,1)-f(k-1,m,1) & \cdots & f(k,m,n)-f(k-1,m,n) \end{pmatrix} \\ D(k-1)=\begin{pmatrix} f(k-1,1,1)-f(k-2,1,1) & \dots & f(k-1,1,n)-f(k-2,1,n) \\ \vdots & & \vdots \\ f(k-1,m,1)-f(k-2,m,1) & \cdots & f(k-1,m,n)-f(k-2,m,n) \end{pmatrix} \end{cases}$$

Then we define the value of every pixel in the difference frames as:

$$\begin{cases} d(k,i,j)=\begin{cases} 0 & f(k,i,j)-f(k-1,i,j)\le\delta \\ 1 & f(k,i,j)-f(k-1,i,j)>\delta \end{cases} \\ d(k-1,i,j)=\begin{cases} 0 & f(k-1,i,j)-f(k-2,i,j)\le\delta \\ 1 & f(k-1,i,j)-f(k-2,i,j)>\delta \end{cases} \end{cases}$$

By performing the subtraction above, we get two difference frames, with the pixels with value 0 the background pixels and the pixels with value 1 belonging to moving areas. Here, $D(k)$ record the moving areas in both $F_I(k)$ and $F_I(k-1)$, while $D(k-1)$ record the moving areas in both $F_I(k-1)$ and $F_I(k-2)$.

Second, we intersect the two difference frames obtained in the first step to get the intersection of the pixels with value 1. Since the moving areas in $F_I(k-1)$ are recorded in both of the two difference frames, by intersecting $D(k)$ and $D(k-1)$, we can easily get the moving areas in $F_I(k-1)$. The moving area in $F_I(k-1)$, that is $M(k-1)$, is calculated like this:

$$M(k-1)=D(k)\cap D(k-1)$$

Third, we obtain the moving areas in $F_I(k)$ by subtracting $D(k)$ from $M(k-1)$. As we have said before, $D(k)$ contains both the moving areas in

$F_I(k)$ and $F_I(k-1)$, and $M(k-1)$ is the moving areas in $F_I(k-1)$, by subtracting $D(k)$ from $M(k-1)$, we get the very moving areas in $F_I(k)$ (the input frame), $M(k)$.

$$M(k) = D(k) - M(k-1)$$

C. Post treatment

There are two problems of consecutive temporal difference approach, its inability to detect the entire moving objects (always causing small holes in the objects) and its sensitivity to any changes between two consecutive frames (being inclined to detect uninteresting motions in the detection). Post treatment must be taken to handle these problems.

Since the small holes in the objects sometimes cause trouble in objects segmentation, especially when the holes are connected inside the object, leading an object to be separated into more one by mistake, measurements must be taken to avoid situations like these. In this paper, we remove the small holes in the objects with the math morphology techniques [16]. The most basic operations of math morphology are expansion and corrosion. To remove the small holes in the objects, first we expand the moving areas, to have the small holes filled up and form the entire shape of the detected object. But the shape of the object is somewhat distorted after expansion, so corrosion is used to shape the object back. Corrosion is a process used to eliminate the points on the border, to make the border contract into the center and to remove some small and meaningless points, thus recover the shape of the object.

Another problem is that the moving areas obtained with the temporal difference approach contain some uninteresting motions. A stationary background is unnatural in outdoor environments, swaying leaves and shaking trees are probable to be detected as moving objects by temporal difference approach. Here, the math morphology is again used to eliminate these uninteresting motions [17]. After the treatment of expansion and corrosion of last step, we get the connecting fields of moving objects. Since vehicles (or interesting objects) are supposed to have a relatively large size compared to the uninteresting motions, we compute the area of every connecting field. If the area the connecting field is less than the threshold value, the field is considered as the uninteresting motion and removed from the moving areas. In this way, we shape the entire object by removing the small holes and eliminate uninteresting motions from the detection, leaving the objects that we are interested in.

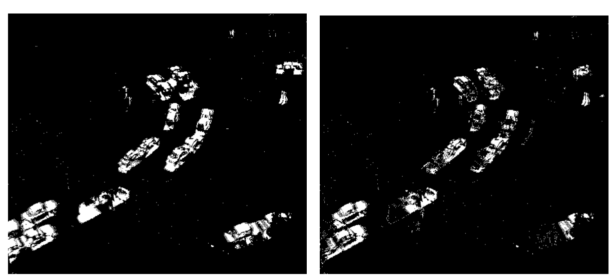
III. EXPERIMENTAL RESULTS

In this section, the effectiveness and accuracy of the proposed method is demonstrated with two cases in traffic monitoring systems with complex backgrounds where shaking cameras, changing elements are presented. The first video sequence used in this section is downloaded from the Internet and the second one is got with a live camera. The average speed for these sequences is about 70 fps and 54fps respectively in 1GB Pentium III machines.

Figure 3 demonstrates the ability of the proposed method to detect moving vehicles on a road. Figure 3(a), 3(b) and 3(c) are three consecutive frames in a video sequence in which cars and walking persons are presented. The frame 3(c) is taken as the input frame. Figure 3(d) is the difference frame of the original frames 3(a) and 3(b) obtained with temporal difference, and 3(e) is that of frames 3(b) and 3(c). The detected moving areas of the input frame, which is derived by subtracting 3(e) by the intersection of 3(d) and 3(e), is shown in Figure 3(f). From Figure 3(f), we can see that there are some small holes and uninteresting motions in the moving areas. Math morphology is applied to shape the entire objects and remove uninteresting motions, and the final detected region is shown in Figure 3(g). The average speed for this sequence runs 70 fps because the camera is relatively stable and the background is much more stationary in this sequence.

(a) the original frame (b) the original frame

(c) the original input frame (d) the difference frame of frames (a) and (b)



(e) temporal difference of frames (b) and (c) (f) detected moving region of the input frame

(g) the final detected region

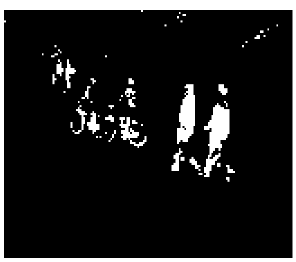
Figure 3. Experimental results of moving objects detection on a road

Figure 4 demonstrates the capability of our method to handle changing background. The three original images on top of the figure are taken from a video sequence got with a live video in a windy day. In the frames, the trees are shaking and some sundries are blown to fly over the floor. The final result is shown in Figure 4(g). From the result, we can see that our algorithm works well in disturbing background. It removes the uninteresting motions from the detected region with satisfactory performance. The speed for this sequence runs about 54 fps.

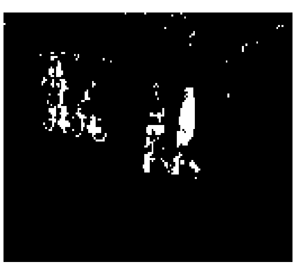
(a) the original frame

(b) the original frame

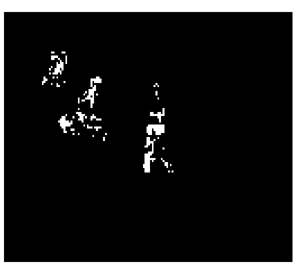
(c) the original input frame



(d) temporal difference of frames (a) and (b)



(e) temporal difference of frames (b) and (c)



(f) detected moving region of the input frame

(g) the final detected region

Figure 4. Experimental results of motion detection from a video sequence taken with a live camera in a windy day

IV. Conclusions

Based on consecutive temporal difference and the square neighborhood matching algorithm, we propose a method for motion detection in complex background in traffic monitoring systems, which is proved to be efficient, accurate and robust. Compared with other similar motion detection algorithms, the main improvement of the proposed method is that it requires only a little time and memory, thus suitable for use in real-time applications. In addition, no prior knowledge of the background is needed for the implementation of this method, and it is quite robust to background changes, not accumulating previous mistakes. Tests on the standard data sets also demonstrate that it has an outstanding performance.

References

[1] Roth S.,Black M.J., “On the spatial statistics of optical flow,” Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on Volume 1, 17-21 Oct. 2005 Page(s):42 - 49 Vol. 1

[2] Zhou J., Gao D., Zhang D., “Moving Vehicle Detection for Automatic Traffic Monitoring,” Vehicular Technology, IEEE Transactions on Volume 56, Issue 1, Jan. 2007 Page(s):51 - 59

[3] Cucchiara R., Grana C., Piccardi M., Prati A., “Detecting Moving Objects, Ghosts, and Shadows in Video Streams,” Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 25, Issue 10, Oct. 2003 Page(s):1337 - 1342

[4] Stocker A.A., “An improved 2D optical flow sensor for motion segmentation,” Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on Volume 2, 26-29 May 2002 Page(s):II-332 - II-335 vol.2

[5] Pathirana P.N., Lim A.E.K., Carminati J., Premaratne M., “Simultaneous estimation of optical flow and object state, A modified approach to optical flow calculation,” Networking, Sensing and Control, 2007 IEEE International Conference on 15-17 April 2007 Page(s):634 - 638

[6] Piccardi M.,“Background subtraction techniques a review,” Systems, Man and Cybernetics, 2004 IEEE International Conference onVolume 4, 10-13 Oct. 2004 Page(s):3099 - 3104 vol.4

[7] Thongkamwitoon T., Aramvith S., Chalidabhongse T.H., “An adaptive real-time background subtraction and moving shadows detection,” Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 2, 30-30 June 2004 Page(s):1459 - 1462 Vol.2

[8] Monnet A., Mittal A., Paragios N., Visvanathan Ramesh, “Background modeling and subtraction of dynamic scenes,” Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on 13-16 Oct. 2003 Page(s):1305 - 1312 vol.2

[9] Fu-Yuan Hu, Yan-Ning Zhang, Lan Yao, “An effective detection algorithm for moving object with complex background,” Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on Volume 8, 18-21 Aug. 2005 Page(s):5011 - 5015 Vol. 8

[10] Durus M., Ercil A., "Robust Vehicle Detection Algorithm," Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th 11-13 June 2007 Page(s):1 – 4

[11] Yeon-sung Choi, Piao Zaijun, Sun-woo Kim, Tae-hun Kim, Chun-bae Park, "Motion Information Detection Technique Using Weighted Subtraction Image and Motion Vector," Hybrid Information Technology, 2006. ICHIT '06. Vol1. International Conference on Volume 1, Nov. 2006 Page(s):263 – 269

[12] Ho M.A.T., Yamada Y., Umetani Y., "An HMM-based temporal difference learning with model-updating capability for visual tracking of human communicational behaviors," Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on 20-21 May 2002 Page(s):163 – 168

[13] Lim S., Apostolopoulos J.G., Gamal A.E., "Optical flow estimation using temporally oversampled video," Image Processing, IEEE Transactions on Volume 14, Issue 8, Aug. 2005 Page(s):1074 – 1087

[14] Kinoshita K., Enokidani M., Izumida M., Murakami K., "Tracking of a Moving Object Using One-Dimensional Optical Flow with a Rotating Observer," Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on 5-8 Dec. 2006 Page(s):1 – 6

[15] Paolo Spagnolo, Tiziana D'Orazio, Marco Leo, and Arcangelo Distante, "Advances in Background Updating and Shadow Removing for Motion Detection Algorithms," Computer Analysis of Images and Patterns, Volume 3691/2005 Page(s): 398-406

[16] Cui Xing, Yan Qingdong, "A highway vehicles detection system based on frames difference," Microcomputer Information, 2007-10 (chinese)

[17] Zhu Minghan, Luo Dayong, "Moving objects detection and tracking based on two consecutive frames subtraction background model," Computer Measurement & Control 2006.14(8) (chinese)

[18] Zhen Tang, Zhenjiang Miao, "Fast Background Subtraction and Shadow Elimination Using Improved Gaussian Mixture Model," Haptic, Audio and Visual Environments and Games, 2007. HAVE 2007. IEEE International Workshop on 12-14 Oct. 2007 Page(s):38 – 41

[19] Qi Zang, Klette R., "Robust background subtraction and maintenance," Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on Volume 2, 23-26 Aug. 2004 Page(s):90 - 93 Vol.2