美和技術學院學報　第二十八卷第一期　民國九十八年
Journal of Meiho Institute of Technology, Vol. 28 No. 1 pp. 55~76

Robust Background Subtraction Algorithm in Intelligence Traffic System

Chung-Cheng Chiu*, Min-Yu Ku**

Abstract

A video-based Intelligence Traffic System (ITS) must be capable of continuous operation under various road conditions. Moreover, background subtraction is a very important part of ITS applications for successful segmentation of objects from video sequences. Accuracy and computational time of the initial background extraction are crucial in any background subtraction method. This paper proposes the probability-based background extraction algorithm to segment objects from surveillance videos. With the proposed algorithm, the initial background can be extracted accurately and quickly by calculating the color probabilities of each pixel to decide the background pixel color. After the initial background extraction, the intrusive objects can be segmented correctly and immediately. Meanwhile, the color background images can be updated in real time to overcome any variation in illumination conditions. Experimental results for various environmental sequences and a quantitative evaluation are provided to demonstrate the robustness, accuracy, effectiveness, and economy of computation time of the proposed algorithm.

Key words: background extraction, object segmentation, image processing.

* Associate Professor, Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University
** Graduate Student, Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University

Ⅰ.Introduction

One of important research efforts in Intelligent Traffic Systems (ITS) is the development of algorithm that automatically extracts and updates the background image to subtract the moving objects from various road backgrounds. Currently, vision-based Intelligent Traffic Systems offer a number of advantages such as vehicle counts, vehicle classifications, traffic parameters, etc., can be measured. Besides, cameras are much less disruptive to install than magnetic loop detectors. Successful subtraction of foreground objects from a complex background scene is an important initial step in intelligent traffic systems applications. However, in real-world situations, there exist several kinds of environment variations that will make the background extraction and foreground segmentation more difficult. In order to cope with that, the approach here should be able to immune to these variations. Real-world environmental variations, including variations in the actual background objects and changes in background illumination, can be classified as follows.

1. Variation of background objects:
   - A dynamic background contains dynamic objects, such as waving curtains or moving tree branches.
   - A static background is an entirely stationary background.
2. Variation of background illumination:
   - A gradual variation of illumination occurs when light sources such as daylight change gradually.
   - A sudden variation of illumination occurs when light sources change suddenly or shine intermittently, as in the case of animated billboards or lights switching on and off.

Overcoming these background variations is an important factor in the accurate and swift extraction of the initial background. Although background extraction has been studied for several decades and many methods have been proposed to deal with these variations, they all require a large amount of memory and processing time. In addition to coping with the variations described, the main goal of the algorithm proposed in this paper is to reduce the memory consumption and to shorten the processing time substantially for extracting the background from color image sequences.

This algorithm can improve the acquisition efficiency of the background image and the adaptive threshold of moving object segmentation methods described in the following articles. In the proposed algorithm, color image sequences are used instead of monochromatic image sequences to obtain better performance. Although color images contain three times the number of pixels compared to monochromatic images of the same size, the computational complexity is not substantially greater. This paper proposes an accurate and fast background extraction algorithm for color image sequences while considering the variations in different illumination and traffic conditions.

To verify its feasibility, the proposed algorithm was installed at four locations in Hsinchu and Taipei, Taiwan. The vehicle recognition of the proposed algorithm uses visual length and visual width, by Chiu *et al.* [1][2], to detect and recognize different vehicle types. Under varying conditions of weather and illumination, the average processing time of the algorithm was less than 32 msec for each color frame. Thus far, the algorithm has been operating for over 10 months to verify the stability of the algorithms.

The remainder of this article is organized as follows. Section 2 contains an overview of the related works. Section 3 gives details of the probability-based background subtraction algorithm. Experimental results and conclusions are given in Sections 4 and 5, respectively.

Ⅱ.Related Works

Many studies have presented algorithms for vision-based detection and classification of vehicles and human activities in consecutive image frames of the intelligent traffic systems [3][4]. Others have developed the scoreboard algorithm and cubical model for estimating stationary backgrounds and segmenting vehicle occlusion in monocular image sequences for automated visual traffic surveillance [5–8]. The running model and running average algorithms are at the heart of the scoreboard algorithm, and temporal difference methods overcome inter-frame and reference differencing, which are problems for traffic detection [9–12]. Other algorithms have been proposed to display traffic congestion and detect accidents at intersections using background extraction [13][14], while a probabilistic line feature grouping algorithm has also been proposed to analyze traffic flow and accumulate traffic lane lines from traffic surveillance videos for intelligent traffic systems [15][16].

Generally speaking, temporal differencing and background subtraction are the two main approaches to segmenting moving foreground objects in the intelligent traffic applications. The principle of temporal differencing is that moving foreground objects are detected using the pixel-wise difference of consecutive image frames. If the absolute luminance value of a pixel differs significantly from the prior frame, the pixel is marked as a moving foreground object; otherwise, the pixel is considered to be part of the background. Although temporal differencing is very adaptable to suddenly changing environments, the segmented foreground objects are always fragmented. This method cannot provide complete information for the foreground classification and tracking required for subsequent procedures.

The principle of background subtraction involves comparing an observed image with an estimated image that contains no objects of interest from the video sequence. The estimated image is a reliable background image constructed from the background model in advance. A significant pixel-wise difference between the observed and estimated images indicates the location of the foreground objects. Therefore, the segmented foreground objects will not suffer from the fragmentation problem. Because the background subtraction method is efficient for noise control and exact subtraction, background subtraction is more popular than temporal differencing.

Therefore, background extraction is a fundamental task for foreground segmentation algorithms in the intelligent traffic applications. However, many real-world sources of environmental noise affect the accuracy of the background extraction, including variation in the background objects themselves and changes in background illumination. For flexible and reliable performances, the background extraction approach should be able to cope with aforementioned variations.

Background extraction can be divided into non-recursive and recursive methods. In non-recursive background extraction, the background image is extracted by observing the difference of each frame in a buffer memory that stores the video sequence. The large amount of memory and long processing time required are the two major drawbacks to non-recursive background extraction. Recursive background extraction uses weighting parameters to approximate the background image and does not require storage of the video sequence in the buffer memory beforehand. Moreover, these weighting parameters are recursively updated by a statistical formula. Although recursive background extraction does not require a large amount of memory and processing time, the pixels of an improperly extracted background cannot be corrected immediately. Ren et al. [17] proposed a background extraction method that involved calculating the mean of the background Gaussian distribution in the background map. Thongkamwitoon et al. [18] proposed statistical background subtraction methods that made the background extraction more robust to non-stationary backgrounds, illumination changes, and other artifacts, while Li et al. [19] proposed a Bayesian framework that incorporated spectral, spatial, and temporal features to characterize the background appearance. These methods adapt to both gradual and sudden background changes, but the long computation time, the sensitivity to the environment, and inefficient background updating are still issues that must be resolved. Jodoin et al. [20] proposed a statistical background subtraction method based on the assumptions that the background distribution is temporally stationary and the background is piecewise ergodic in time. The method can reduce the memory requirements and speed up the processing time although it requires a background frame to model each pixel with the unimodal probability density function and the multimodal probability density function beforehand. Further, most existing methods perform the background subtraction with one or more heuristic thresholds. For backgrounds with different complexities, the thresholds should be adjusted empirically. In addition, these methods are often tested only on a few background environments (e.g., laboratories, campuses, etc.).

Ⅲ.Probability-Based Background Subtraction Algorithm

In this paper, the proposed probability-based background extraction algorithm used the extracted static background image in place of the previous frame. The proposed method can overcome fragmental problems, which always occur inside the detected objects, of the temporal difference. The analog output image from the stationary video camera consists of the background image and moving object image. Because the background image is motionless, the color information, red (R), green (G), and blue (B), of the background pixels should be the same in an image sequence during the background extraction. However, some background pixels may have different pixel colors in an image sequence, because of the moving objects passing through these background pixels and illuminative variation. Therefore, each pixel in an image sequence will have as many different colors as the candidates for a background pixel. For any given pixel of an image sequence, the probability of the background color will be higher than the probabilities of the colors counted when the moving objects pass through in the image sequence.

In order to extract these precise background pixels in an image sequence, the proposed algorithm calculates the probabilities of the colors of each pixel and uses a convergent value to decide the background color, whose probability is the maximum one and greater than the convergent value, of the pixel in an image sequence. By the proposed convergent equation, the convergent value and the background pixel colors of the frame can be determined and extracted very quickly and accurately. The extracted background can be updated in real-time to overcome the variation of the illuminative condition, too.

Assume that $f_i(x,y)$ is the ith frame in an image sequence, where x and y represent the pixel coordinates and the image size is $M\times N$. Each pixel in the f_i is defined as $Q_i(x,y)$, and the $Q_i(x,y)$ is composed of red ($R_i(x,y)$), green ($G_i(x,y)$), and blue ($B_i(x,y)$) components. The parameter $C(x,y,n(x,y))$ is the pixel number of the n-th classification located on (x,y), and the parameters $R(x,y,n(x,y))$, $G(x,y,n(x,y))$, and $B(x,y,n(x,y))$ are defined respectively as the red, green, and blue colors of the n-th classification located on (x,y), where the parameter, $n(x,y)$, recording the cluster number located on (x,y). The initial background image extraction of the proposed algorithm is shown below.

Step 1. Capture the first $M \times N$ color frame f_i, where i=1, and set

$$\begin{cases} R(x,y,n(x,y)) = R_1(x,y) \\ G(x,y,n(x,y)) = G_1(x,y) \\ B(x,y,n(x,y)) = B_1(x,y) \\ C(x,y,n(x,y)) = 1 \end{cases} \tag{1}$$

Where $n(x,y)=1$, $x = 0 \sim M-1$, and $y = 0 \sim N-1$.

Step 2. Capture the next frame f_i, where $i=i+1$.

Step 3. Sequentially calculate the $Q_i(x, y)$ pixel from the frame f_i, where $Q_i(x, y)=\{R_i(x, y), G_i(x, y), B_i(x, y)\}$, $x=0\sim M-1$ and $y=0\sim N-1$. If all the pixels of the frame f_i have been calculated, go to the step 2.

Step 4. Calculate the color difference, $D_i\ (x,y,n(x,y))$, between the fetched pixel and each classified cluster of the pixel located on (x, y).

$$\begin{aligned} D_i\ (x,y,n(x,y)) = &\left|R_i(x,y) - R(x,y,n(x,y))\right| + \\ &\left|G_i(x,y) - G(x,y,n(x,y))\right| + \\ &\left|B_i(x,y) - B(x,y,n(x,y))\right| \ , \ \forall n(x,y). \end{aligned} \tag{2}$$

Step 5. Find the minimum value, $D_i\ (x,y,m)$, of the $D_i\ (x,y,n(x,y))$.

$$D_i\ (x,y,m) = \min_{\forall n(x,y)} D_i(x,y,n(x,y))\ , \quad m \subset n(x,y)\ . \tag{3}$$

Step 6. If ($D_i\ (x,y,m) <$ TH_D), the best classification is found. The input pixel is classified to the mth classification, $\{R_i\ (x,y,m), G_i\ (x,y,m), B_i\ (x,y,m)\}$. Therefore, update the number of the mth classification, $C(x,y,m)=C(x,y,m)+1$, and the R, G, and B values of the pixels on (x,y). To reduce computing time, we use the update method of Eq. (4) to replace the weighting average method.

$$\begin{cases} \text{If } (R(x,y,m) < R_i(x,y)), \text{ then } & R(x,y,m) = R(x,y,m)+1, \\ & \text{Else } R(x,y,m) = R(x,y,m)-1. \\ \text{If } (G(x,y,m) < G_i(x,y)), \text{ then } & G(x,y,m) = G(x,y,m)+1, \\ & \text{Else } G(x,y,m) = G(x,y,m)-1. \\ \text{If } (B(x,y,m) < B_i(x,y)), \text{ then } & B(x,y,m) = B(x,y,m)+1, \\ & \text{Else } B(x,y,m) = B(x,y,m)-1. \end{cases} \tag{4}$$

Then, go to Step 8.

Step 7. If ($D_i\ (x,y,m)$ >*TH_D*), set the input pixel as a new classification of the pixel located on (*x*,*y*). The color information of the new cluster is defined by Eq. (5).

$$n(x,y) = n(x,y)+1, \text{ and } \begin{cases} R(x,y,n(x,y)) = R_i(x,y) \\ G(x,y,n(x,y)) = G_i(x,y) \\ B(x,y,n(x,y)) = B_i(x,y) \\ C(x,y,n(x,y)) = 1 \end{cases} \tag{5}$$

Then, go to Step 3.

Step 8. If (*i* > *Fn*), calculate the maximum classification probability, $P_i^{\max}(x,y)$, of the pixel.

$$P_i\ (x,y,n(x,y)) = \frac{C(x,y,n(x,y))}{i} \tag{6}$$

$$P_i^{\max}(x,y) = \max_{\forall n(x,y)} P_i\ (x,y,n(x,y)) \tag{7}$$

Else if (*i* ≤ *Fn*), go to Step 3.

Step 9. If ($P_i^{\max}(x,y)$ > *TH_E*), extract the pixel as a background pixel and the pixel in location (*x*,*y*) will not be processed in the follow-up steps before the initial background image is extracted. To speed up the extraction of the background pixels, the convergent value, *TH_E,* will be adjusted as defined by Eq. (8).

$$TH\_E = 0.5 \times 0.5^{\left\lceil \frac{i-Fn}{10} \right\rceil} \tag{8}$$

Step 10. If all the background pixels have been extracted, stop the calculation. Otherwise, go to Step 3.

Repeat Steps 2~10 until all the background pixels of the frame have been extracted. In the initial background image extraction of the proposed algorithm, the parameter, *TH_D*, will affect the number of the clusters during the initial background extraction. The number of the clusters will decrease with increasing values of *TH_D*. Nevertheless, it will cause a great deal distortion to the colors of the background pixels. The parameter, *Fn*, is the threshold value of the frame number to begin extracting the background, which affects the computational time and memory requirements. In order to adapt easily to the background variations, the automatic convergent value, *TH_E*, will decrease a half per 10 frames to speed up the extraction of background pixels.

After obtaining the initial background image, the intrusive objects of input image can be segmented by calculating the *RGB* difference between the background image and the input image for each pixel. In order to overcome the influence of various environmental effects and illumination changes, the threshold value of the object segmentation must be changed to adapt to different illumination conditions in an image sequence. Hence, the proposed object segmentation uses the variation of the color difference histograms between the input frame and the background frame as the threshold values. At the same time, the segmentation results are fed back to the background update to update the background image.

The parameter, f_B, is defined as the extracted background image. The parameters $R_B(x, y)$, $G_B(x, y)$, and $B_B(x, y)$ are defined respectively as the red, green, and blue colors of the f_B. After the initial background extraction, the intrusive object segmentation processes and the background updating are shown below.

Step 1. Capture the input frame f_i, where the parameter i is the serial number of the input frame.

Step 2. Count the color difference histograms, $H_{i,R}$, $H_{i,G}$, and $H_{i,B}$, of the input frame f_i. The $H_{i,R}$, $H_{i,G}$, and $H_{i,B}$ are defined as the histograms of $|R_i(x,y)-R_B(x,y)|$, $|G_i(x,y)-G_B(x,y)|$, and $|B_i(x,y)-B_B(x,y)|$, where x=0~M-1 and y=0~N-1, respectively. In the meantime, segment the intrusive objects with the parameters $V_{i-1,R}$, $V_{i-1,G}$, and $V_{i-1,B}$. The three parameters, $V_{i-1,R}$, $V_{i-1,G}$, and $V_{i-1,B}$, are the valleys of the color difference histograms, $H_{i-1,R}$, $H_{i-1,G}$, and $H_{i-1,B}$, which are computed from the (i-1)-th frame.

$$\begin{cases} \text{If} \begin{pmatrix} \left(\left|R_i(x,y)-R_B(x,y)\right|>V_{i-1,R}\right) \\ \vee\ \left(\left|G_i(x,y)-G_B(x,y)\right|>V_{i-1,G}\right) \\ \vee\ \left(\left|B_i(x,y)-B_B(x,y)\right|>V_{i-1,B}\right) \end{pmatrix}; \\ \qquad \text{the pixel belongs to a intrusive object image,} \\ \qquad Q_O(x,y)=\{R_i(x,y),G_i(x,y),B_i(x,y)\}. \\ \text{Else} \\ \qquad \text{the pixel belongs to the static image,} \\ \qquad Q_S(x,y)\ =\{R_i(x,y),G_i(x,y),B_i(x,y)\}. \end{cases} \tag{9}$$

Step 3. Compute the valleys, $V_{i,R}$, $V_{i,G}$, and $V_{i,B}$, of the color difference histograms, $H_{i,R}$, $H_{i,G}$, and $H_{i,B}$, after a smooth filter, respectively.

Step 4. Label each object of the intrusive object image, $Q_O(x,y)$, by the connect-component labeling method.

Step 5. Delete the isolated pixels and small objects and fill the hollow inside the objects by the constrained run-length. Then, the static image, $Q_S(x,y)$, will delete the pixels which are the hollow inside the objects.

Step 6. Update the background image with the static image, $Q_S(x,y)$.

$$\begin{cases} R_B(x,y)=\dfrac{R_B(x,y)\times\left(2^n-1\right)}{2^n}+\dfrac{R_S(x,y)}{2^n} \\ G_B(x,y)=\dfrac{G_B(x,y)\times\left(2^n-1\right)}{2^n}+\dfrac{G_S(x,y)}{2^n} \\ B_B(x,y)=\dfrac{B_B(x,y)\times\left(2^n-1\right)}{2^n}+\dfrac{B_S(x,y)}{2^n} \end{cases}, \quad \text{where } x=0\sim M-1, y=0\sim N-1. \tag{10}$$

In the foreground segmentation of the proposed algorithm, the valleys as the threshold values are calculated by the previous frame, f_{i-1}, because the valleys of the previous frame are similar to the valleys of the current frame. The foreground segmentation calculates the valleys from the color difference histograms of the current frame, and segments the intrusive objects by the valleys of the previous frame simultaneously. Therefore, the robust object segmentation can reduce the computational complexity substantially.

I. Experimental Results

In the following experiments, the frame rate of the images captured by a stationary CCD color camera is 30 frames per second (fps) with an image size of 320×240 pixels. The proposal algorithm was tested on the platforms with Intel Pentium CPU 1.8 and 3.0 GHz, respectively, with the 1 GB of RAM, and software developed using Visual C++ 2005. The proposed algorithm was tested on numerous image sequences captured under different illumination and road conditions. The test environments are including JianGuo North Road, TiDing Blvd in Taipei, and No. 68 Expressway, Jhong-Hua Road in Hsinchu. In all of the tests, the backgrounds were automatically and quickly extracted from the image sequences for various background objects. Once the initial background image was complete, the system exactly separated the foreground image from the input frames, and constantly updated the background image. Visual examples, objective evaluation, and computational time of the experiments are described in the following three subsections.

A. Experimental results under various environments

The proposed algorithm has been tested on numerous color image sequences captured in different places including the No. 68 Expressway and Jhong-Hua Road in Hsinchu. Some results of the initial background extraction are shown in the Figs. 1 and the Figs. 2. In the Figs. 1 and the Figs. 2, the figures (a)–(e) are the original images, the figures (f)–(n) are the constructed background, and the figure (o) is the extracted background. These image sequences contain various moving speeds, object sizes, illuminations, and background object variations. The probability-based background extraction algorithm computes the classification probabilities of each pixel before the 30-th frame, and extracts the background pixels after the 31-th frame. The frame number to extract all the background pixels depends on the moving speed of the intrusive objects. Once the initial background image was complete, the system exactly separated the foreground image from the input frames to recognize the vehicle, and constantly updated the background image. The experimental results indicate that the proposed method is adaptable to various illumination conditions and performs quite well for extracting and updating backgrounds on urban roads.

(a) (b) (c) (d) (e)

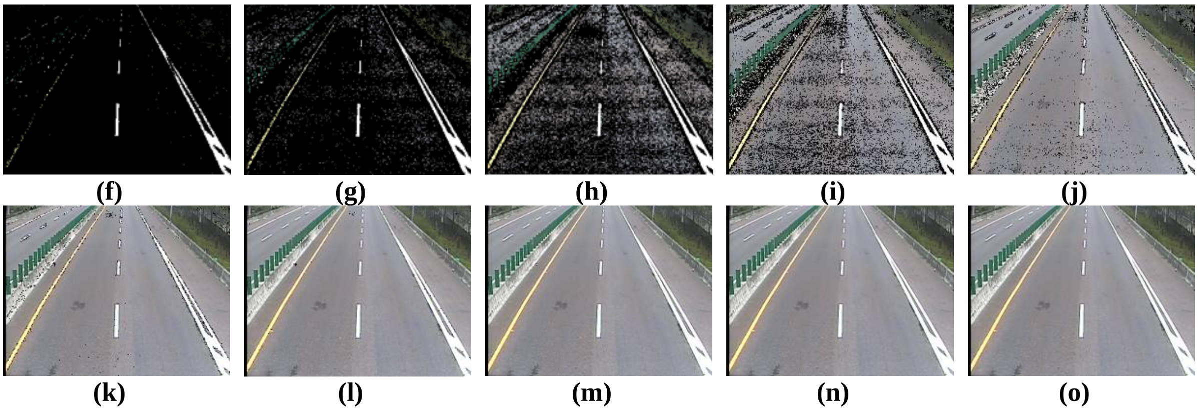


Fig.1 Experimental results of the initial background extraction algorithm: (a)–(e) the 31st, 40th, 48th, 56th, and 64th frames of the original image sequence of the No. 68 Expressway in Hsinchu; (f)–(n) constructed background at the 31st, 40th, 48th, 56th, 64th, 72nd, 80th, 88th, and 96th frames; (o) the extracted background.

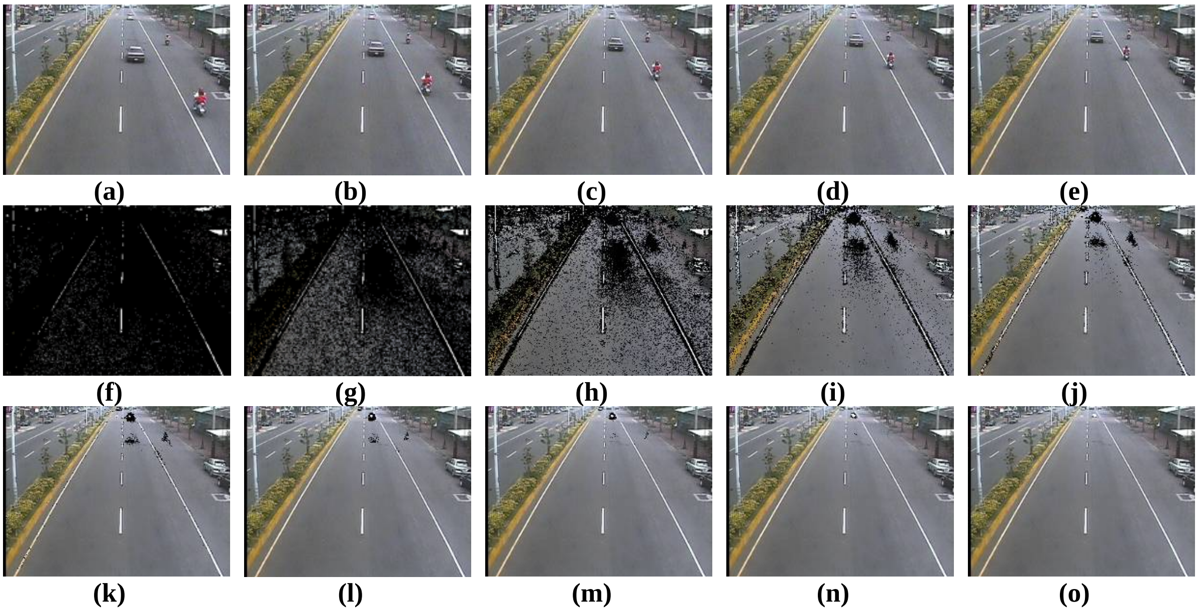


Fig.2 Experimental results of the initial background extraction algorithm: (a)–(e) the 31st, 40th, 48th, 56th, and 64th frames of the original image sequence of the Jhong-Hua Road in Hsinchu; (f)–(n) constructed background at the 31st, 40th, 48th, 56th, 64th, 72nd, 80th, 88th, and 96th frames; (o) the extracted background.

B. Quantitative evaluations

In order to get a numerical evaluation of the proposed algorithm, the performance of the proposed algorithm must be evaluated quantitatively on the benchmark sequence. The “ground truths”, f_i^{gro}, of the benchmark sequence were manually segmented. From the comparison between the “ground truths” and the foregrounds, the sum of the different pixels is an index to the accuracy of the foreground which was segmented by the proposed method. The benchmark image sequence, Hall Monitor (HM), is a MPEG-4 test sequence, commonly for evaluating the effectiveness of background extraction and object segmentation techniques. In this paper, a measurement is introduced to evaluate the error rate of the foreground segmentation and the background extraction of the proposed algorithm at the same time. The error rate is defined by the following:

$$\begin{aligned} Error\_Rate &= \frac{Error\_Rate\_Count}{Frame\_Size} \\ &= \frac{Foreground\_Error\_Count + Background\_Error\_Count}{Frame\_Size} \\ &= \frac{Foreground\_Error\_Count}{Frame\_Size} + \frac{Background\_Error\_Count}{Frame\_Size} \\ &= ER_{Fore} + ER_{Bac} \\ &= \frac{\left(\sum_{(x,y)} \left(Q_{F\_i}(x,y) \oplus f_i^{gro}(x,y)\right)\right)}{Frame\_Size} \quad , \end{aligned} \tag{11}$$

where the $Q_{F\_i}(x,y)$ is the foreground pixels segmented by the proposed algorithm, the $f_i^{gro}(x,y)$ is the ground truth pixels of the foreground, and the operator “$\oplus$” is the logical operation exclusive or, symbolized XOR. There are 300 frames in the benchmark image sequence (HM). Before the 101st frame, the background image is initially extracted. The error rates of the benchmark image sequence, between 102nd frame and 300th frame, are shown as Fig.3. The average error rate is 0.88%. These numerical results indicate that the proposed algorithm can stably and continuously obtain the better quality for the background extraction and the object segmentation.

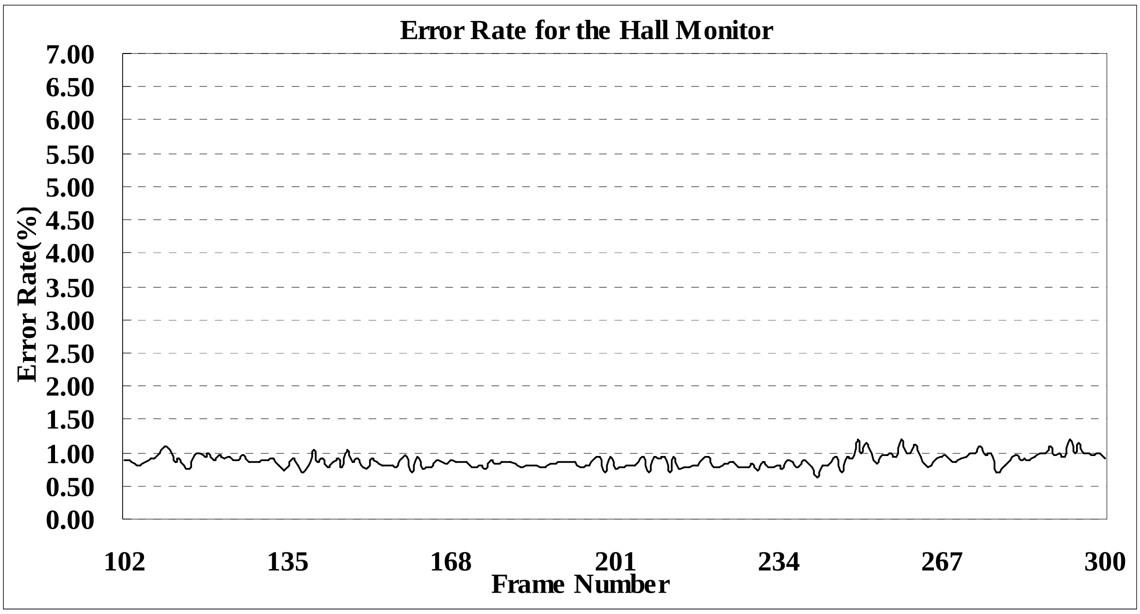


Fig.3 Error rate in each frame of the Hall Monitor.

Figure 4 shows the experimental results for a benchmark sequence, Hall Monitor (HM), in CIF format. The clothing of the two people is similar to the background, and shadows caused by hallway lighting appear in the background. Figure 4(a) and 4(b) are the input and background images of the proposed approach, respectively, and Fig. 4(c) is the foreground image extracted by the proposed approach. The experimental results indicate that the proposed method performed quite satisfactorily for complicated background changes.

(a) The 258-th frame **(b) The background image of the 258-th frame** **(c) The object image of the 258-th frame**

Fig. 4. The experimental results of object segmentation in Hall Monitor sequence (HM) with two people.

Two experimental results for road traffic are shown in Fig. 5 and Fig. 6. The first sequence in Fig. 5 is from the No. 68 Expressway in Hsinchu. Most changes in the foreground segmentation are due to a large number of the high speed passing vehicles on the ground. The high speed passing vehicles also cause noise that decreases the efficiency of the foreground segmentation. Figures 5(a), 5(d), and 5(g), captured by the 2926-th, 4371-th, and 27643-th frames, are the original input image and Fig.5(b), Fig.5(e), and Fig.5(h) are the background image extracted by the proposed algorithm, respectively. The Fig.5(c), Fig.5(f), and Fig.5(i) are the foreground images extracted by the proposed approach and show that the proposed approach is not affected by the high speed passing vehicles during the foreground segmentation. The average processing time of the robust object segmentation for each frame is under 11.24~23.25 milliseconds on the platforms with Intel Pentium CPU 1.8 and 3.0 GHz, respectively.

Figure 6 shows some experimental results of the robust object segmentation on the Jhong-Hua Road in Hsinchu. Most changes in the background are due to a large number of the passing vehicles on the ground. The passing vehicles also cause noise that decreases the efficiency of the background extraction. The Fig.6(a), Fig.6(d), and Fig.6(g) are the images captured by the 12383-th, 21251-th, and 21285-th frames, respectively. The Fig.6(b), Fig.6(e), and Fig.6(h) are the background images of the Fig.6(a), Fig.6(d), and Fig.6(g), respectively. The Fig.6(c), Fig.6(f), and Fig.6(i) are the intrusive objects of the Fig.6(a), Fig.6(d), and Fig.6(g), respectively. The average processing time of the robust object segmentation for each frame is under 15.38~27.78 milliseconds on the platforms with Intel Pentium CPU 1.8 and 3.0 GHz, respectively. In the original Fig.6(c) in the paper, the separated area of the red car can be connected by the morphology and it didn’t affect the vehicle counting results. According to the characteristic of the traffic image, vehicles in the end of the road can be considered as the stopping objects such as the upper part of the Figs.6(c), 6(f), 6(i). Though these noises will confuse the background extraction occasionally, sizes are usually small and fragment. According to the after-process of the recognition of Chiu *et al.*[1][2], the moving objects between the 3/4 (vertical axis) part from the bottom of image have counted and classified. Therefore, they didn’t affect our recognition in the traffic image sequences.

(a) The 2926-th frame

(b) The background image of the 2926-th frame

(c) The object image of the 2926-th frame



(d) The 4371-th frame **(e) The background image of the 4371-th frame** **(f) The object image of the 4371-th frame**



(g) The 27643-th frame **(h) The background image of the 27643-th frame** **(i) The object image of the 27643-th frame**

Fig.5 The experimental results of object segmentation on the No. 68 Expressway in Hsinchu.



(a) The 12383-th frame **(b) The background image of the 12383-th frame** **(c) The object image of the 12383-th frame**



(d) The 21251-th frame **(e) The background image of the 21251-th frame** **(f) The object image of the 21251-th frame**

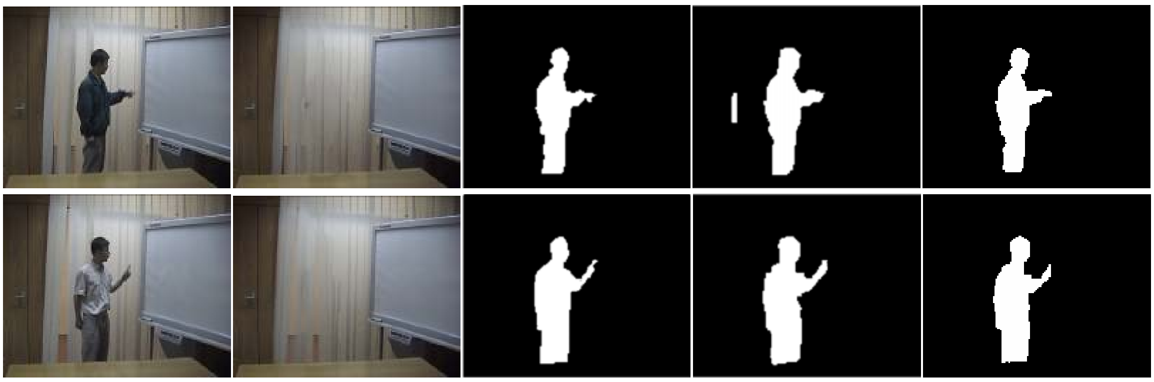


(g) The 21285-th frame **(h) The background image of the 21285-th frame** **(i) The object image of the 21285-th frame**

Fig.6 The experimental results of object segmentation on the Jhong-Hua Road in

Hsinchu.

We compared our algorithm to Li et al.'s [19] approach in the office environment experiments. The scenario backgrounds included a waving curtain; these were suited to evaluating the effectiveness of background extraction and foreground subtraction techniques. To subtract foreground objects from the test image sequences effectively, the problems of noise caused by the irregular waving curtain and vague foreground objects where the color of the foreground object is similar to the background must be handled well. From left to right in each row of Fig. 7, the images show the input frames, the background images extracted and maintained by the proposed algorithm, the handmade "ground truth" [19], the results of Li et al.'s [19] method, and the results of the proposed algorithm. Figures 7(d) indicates that the noise due to the waving curtain (1816 frame) is still considered as foreground even when Li et al.'s [19] approach is adapted to background updating and subtraction. The proposed algorithm can eliminate the noise effect as shown in Figs. 7(e).



(a) (b) (c) (d) (e)

Fig. 7. Li et al.'s [19] sequence. (a) The 1816th and 2268th frames of the sequence. (b) Background images of the proposed approach. (c) Ground truths [19]. (d) Detection results of Li's approach [19]. (e) Detection results of the proposed approach.

C. Computational time

To analyze the computation time, the proposed algorithm was tested on various CPUs, including Pentium CPUs with clock speeds of 1.8 GHz and 3.0 GHz. Table 1 shows the computation time on the various CPUs for the JianGuo N. Road, TiDing Blvd, No. 68 Expressway, and Jhong-Hua Road image sequences with an image size of 320×240 pixels during the initial background extraction for various environments. On the 3.0 GHz Pentium CPU, the average time required to finish the initial background extraction for the various environments was approximately 2.51 s, which corresponds to a frame rate greater than 45 fps. On the 1.8-GHz CPU, the average time required to finish the initial background extraction was approximately 5.12 s, and the frame rate was around 23 fps.

Table 1. Processing time of the initial background extraction for various environments.

Environment / CPU	JianGuo N. Rd.			TiDing Blvd			No. 68 Expressway			Jhong-Hua Rd.		
	Total time (s)	**Total frames**	**Avg. fps***	**Total time (s)**	**Total frames**	**Avg. fps***	**Total time (s)**	**Total frames**	**Avg. fps***	**Total time (s)**	**Total frames**	**Avg. fps***
1.8 GHz	5.75	106	19	4.14	102	25	7.52	102	13	3.06	109	35
3.0 GHz	2.45		44	2.55		40	3.49		29	1.55		70

***Avg. fps, average frames per second**

Table 2 shows the average processing time for the object segmentation and background update algorithm for 2000 successive frames following the initial background extraction. After the algorithm had extracted the initial background, the object segmentation and background update algorithm proceeded at a frame rate of around 39 fps on the 1.8-GHz Pentium CPU. On the 3.0-GHz Pentium CPUs, the average frame rates of the object segmentation and background update algorithm were around 75 fps. The processing time was much better than that of Li *et al*. [19], who obtained a frame rate of only 3 fps for an image size of 320 × 240 pixels, on a 1.7-GHz Pentium CPU. The experimental results indicate that the proposed method performed quite satisfactorily for complicated background changes.

Table 2. Processing time of the object segmentation and background update algorithm for 2,000 successive frames.

Environment / CPU	JianGuo N Rd.		TiDing Blvd		No. 68 Expressway		Jhong-Hua Rd.	
	Total time (s)	**Avg. fps***	**Total time (s)**	**Avg. fps***	**Total time (s)**	**Avg. fps***	**Total time (s)**	**Avg. fps***
1.8 GHz	54.68	37	50.86	39	46.68	43	55.02	36
3.0 GHz	30.82	65	24.22	83	22.50	89	30.86	65

***Avg. fps, average frames per second**

II. Conclusions

An efficient and robust object segmentation algorithm was proposed in this paper. The background subtraction algorithm has a probability-based background extraction algorithm to construct an initial background from different variations and roads image sequences. Intrusive objects can be segmented by the robust object segmentation algorithm, and the background image can be sequentially updated by the background update algorithm. The proposed algorithm can extract and update the initial background from various environments, and efficiently reduce the processing time. The background and intrusive object information provide a good basis for follow-up studies including object detection, counting, tracking, and recognition. Although background extraction has been studied for several decades and many methods have been proposed to deal with different background and variations, they all require a long processing times, and can not automatically adjust the frame number that needs to extract the initial background. In addition to coping with the variations described, the main advantage of the proposed algorithm is that it can substantially reduce processing time of the background extraction.

There are other contributions of the proposed algorithm. First, the frame number that needs to extract the initial background can automatically adjust according to the speed and size of the moving objects. Secondly, the threshold value to detect the moving objects between the input image and the background image can adapt to the features of input frames. Because the proposed algorithm can segment the intrusive objects in 320×240 pixels around 39 fps on a 1.8-GHz CPU, the algorithm is quite suitable for use in a real-time system. Experimental results obtained with various image sequences reveal that the proposed algorithm can successfully extract the background image and segment the intrusive objects.

Nevertheless, our proposed method is heavily dependent on the camera setting and characteristic. The segmentation may become unsatisfactory when the auto white balance function of camera was started automatically. Therefore, our on-going research is to develop an anti-AWB algorithm which can automatically adjust the color to adapt the background images of the cameras. Last but not least, the result after high-level recognition will be fed back to update the background to handle the situations with structure variation and severe shadow effect.

References

[1] Chiu, C. C., Wang, C. Y., Ku, M.Y., & Lu, Y. B. (2006). Real-time Recognition and Tracking System of Multiple Vehicles. *IEEE International Conference on Intelligent Vehicles Symposium*, 478-483.

[2] Ku, M. Y., Chiu, C. C., Chen, H. T., & Hong, S. H. (2008). Visual Motorcycle Detection and Tracking Algorithm. *WSEAS Trans. on Electronics*, *5*(4), pp.121-131.

[3] Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W4: real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *22*(8), 809-830.

[4] Gupte, S., Masoud, O., Martin, R. F. K., & Papanikolopoulos N. P. (2002). Detection and classification of vehicles. *IEEE Trans. Intelligent Transportation Systems*, *3*(1), 37-47.

[5] Hi, A. H. S. & Yung, N. H. C. (1998). A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence. *In Proc. IEEE Symp. Circuits and Systems*, *4*(31), 241–244.

[6] Pang, C. C. C., Lam, W. W. L., & Yung, N. H. C. (2003). A novel method for handling vehicle occlusion in visual traffic surveillance. *in Proc. SPIE Conf. SPIE-IS&T Electronic Imaging*, *5014*, 437-447.

[7] Pang, C. C. C., Lam, W. W. L., & Yung, N. H. C. (2004). A novel method for resolving vehicle occlusion in a monocular traffic-image sequence. *IEEE Trans. Intelligent Transportation Systems*, *5*(3), 129–141.

[8] Pang, C. C. C., Lam, W. W. L., & Yung, N. H. C. (2007). A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images. *IEEE Trans. on Intelligent Transportation Systems*, *8*(3), 441-459.

[9] Anderson, C., Burt, P., & Wal, G. V. D. (1985). Change detection and tracking using pyramid transformation techniques. *in Proc. SPIE Conf. Intelligent Robots and Computer Vision*, *579*, 72-78.

[10] Ali, A. T., & Dagless, E. L. (1992). Alternative practical methods for moving object detection. *in Proc. IEEE Conf. Image Processing and its Applications, April*, 77-80.

[11] Rosin, P. L., & Ellis, T. (1995). Image difference threshold strategies and shadow detection. *in Proc. British Machine Vision Conference*, 347-356.

[12] Li, G., Zeng, R., & Lin, L. (2006). Moving target detection in video monitoring system. *in Proc. WCICA Conf. Intelligent Control and Automation*, *2*, 9778-9781.

[13] Arth, C., Bischof, H., & Leistner, C. (2006). TRICam—an embedded platform for remote traffic surveillance. *in Proc. Computer Vision and Pattern Recognition Workshop Conf. Digital Object Identifier, June*, 125.

[14] Wu, B.-F., Juang, J.-H., Tsai, P.-T., Chang, M.-W., Fan, C.-J., Lin, S.-P., Wu, J. Y.-J., & Lee, H. (2007). A new vehicle detection approach in traffic jam conditions. *in Proc. IEEE Symp. Computational Intelligence in Image and Signal Processing*, *April*, 1-6.

[15] Kim, Z., & Malik, J. (2003). Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. *in Proc. IEEE Conf. Computer Vision*, *1*, 524-531.

[16] Hsieh, J.-W., Yu, S.-H., Chen, Y.-S., & Hu, W.-F. (2006). Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Trans. Intelligent Transportation Systems*, *7*(2), 175-187.

[17] Ren, Y., Chua, C.-S., & Ho, Y.-K. (2001). Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, *24*(1-3), 183-196.

[18] Thongkamwitoon, T., Aramvith, S., & Chalidabhongse, T. H. (2004). An adaptive real-time background subtraction and moving shadows detection. *in Proc. IEEE Conf. Multimedia and Expo*, *2*, 1459-1462.

[19] Li, L., Huang, W., Gu, I. Y.-H., & Tian, Q. (2004). Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Processing*, *13*(11), 1459-1472.

[20] Jodoin, P.-M., Mignotte, M., & Konrad, J. (2007). Statistical background subtraction using spatial cues. *IEEE Trans. Circuits and Systems for Video Technology*, *17*(12), 1758–1763, Dec.

運用於智慧型交通系統中之強健型背景抽取法

瞿忠正*、古閔宇**

摘要

影像式智慧型交通系統必需具備在各式道路及天候下均能運作之能力，因此，快速的背景抽取與精確成功的前景切割，將是影響該系統是否順利運行的關鍵因素。本篇提出一個運用機率模式之強健型背景相減法，本方法可以迅速且有效率的自連續彩色影像中抽取背景，而其中背景抽取的概念是藉由統計影像中每一個像素色彩變化及出現的機率，設計一個門檻值來做爲背抽取的依據，在初始背景完成抽取後，除籍由影像相減對移動物體正確且即時切割外，並同時更新初始背景影像，而所提之方法可以達到即時且高效率之背景抽取及移動物件切割，目前藉由各種道路環境的實測結果，驗證了所提之方法的強健性、準確性與即時性。

關鍵字：背景抽取、物件切割、影像處理。

*國防大學理工學院電機電子工程學系副教授
**國防大學理工學院電機電子工程學系博士生