Chapter 2
Background/Foreground Detection[1]

2.1 Introduction

With the acquisition of an image, the first step is to distinguish objects of interest from the background. In surveillance applications, those objects of interest are usually humans. Their various shapes and different motions, including walking, jumping, bending down, and so forth, represent significant challenges in the extraction of foreground pixels from the image.

To tackle that problem, we consider two approaches: one is based on the fact that the background is stationary in the image captured by a fixed monocular camera, while the other is based on the assumption that the foreground contains objects of interest that are moving. The next sections elaborate on the two approaches in detail.

2.2 Pattern Classification Method

A popular detection methodology is based on the concept that once the background is extracted, the foreground can conveniently be obtained by subtracting the viewed image from the background model [162, 164].

2.2.1 Overview of Background Update Methods

Current background extraction methods include the multi-frame average, selection, selection-average, and random update methods [1–7] and Kalman filter-based method [8, 9]. These are briefly elaborated and compared below.

[1] Portions reprinted, with permission from Xinyu Wu, Yongsheng Ou, Huihuan Qian, and Yangsheng Xu, A Detection System for Human Abnormal Behavior, *IEEE International Conference on Intelligent Robot Systems*. ©[2005] IEEE.

H. Qian et al., *Intelligent Surveillance Systems*, Intelligent Systems,
Control and Automation: Science and Engineering 51, DOI 10.1007/978-94-007-1137-2_2,

2.2.1.1 Multi-Frame Average Method

The multi-frame average method can be described as follows. For an image sequence $B_i \;\; i = 1,..,n$,

$$B_n = \frac{\sum_1^n B_i}{n} \tag{2.1}$$

Here, if the number of frames, n, is as large as needed, then B_n should be the background. However, in this method, too much memory is required to store the image sequence B_i. Hence, an approximate method is developed, which can be presented as

$$B_{pt} = kB_{pt-1} + (1-k)C_{pt-1} \qquad 0 < k < 1 \tag{2.2}$$

where

- B_{pt} —- a pixel in the current background;
- B_{pt-1} —- a pixel in the last background;
- C_{pt-1} —- a pixel in the current image; and
- k —- a threshold value.

The disadvantages of this method are: 1) the threshold value k is difficult to determine, and 2) k needs to be adjusted, depending on the degree of environmental change. When there are many objects in an active area, the noise in that area can be great, because the objects can also be deemed as background.

2.2.1.2 Selection Method

In this method, a sudden change in pixels is deemed to be the foreground, which will not be selected as the background.

$$\begin{aligned} &If \;\; |C_{p1} - C_{pt-1}| > T \\ &Then \;\; B_{pt} = B_{pt-1} \qquad\qquad (don't\ update) \\ &Else \;\; B_{pt} = C_{pt-1} \qquad\qquad (update), \end{aligned} \tag{2.3}$$

where

- B_{pt} —- a pixel in the current background;
- B_{pt-1} —- a pixel in the last background;
- C_{pt} —- a pixel in the current image of the image sequence;
- C_{pt-1} —- a pixel in the last image of the image sequence; and
- T —- a threshold value.

The disadvantage of this method is that determining the threshold value of T is difficult. If T is too small, then the background will be insensitive to change. If T is too large, then the background will be too sensitive to change.

2.2.1.3 Selection-Average Method

To address the disadvantages of the multi-frame average and selection methods, Fathy [19] proposed an improved method - the selection-average method - which is expressed as follows.

$$\begin{aligned}
&If\ |C_{pt}-B_{pt}|<T_1\\
&Then\, If\ |C_{pt}-C_{pt+1}|<T_2\\
&Then\ \ B_{pt+1}=(B_{pt}+C_{pt+1})/2 \qquad\qquad (update)\\
&Else\ \ B_{pt+1}=B_{pt} \qquad\qquad\qquad\quad (don't\ update),
\end{aligned} \tag{2.4}$$

where

- B_{pt+1} —- a pixel in the current background;
- B_{pt} —- a pixel in the last background;
- C_{pt+1} —- a pixel in the current image of the image sequence;
- C_{pt} —- a pixel in the last image of the image sequence; and
- T_1,T_2 —- threshold values.

2.2.1.4 Kalman Filter-based Adaptive Background Update Method

The Kalman filter-based model [8, 9] allows the background estimate to evolve as lighting conditions change with changes in the weather and time of day. In this method, the moving objects in the frame stream are treated as noise, and a Kalman filter is used to estimate the current background. However, it is obvious that in many cases, such as during rush hour or in a traffic jam, the moving object cannot simply be deemed noise. This method also requires much calculation time and is not discussed in detail here.

2.2.1.5 Another Adaptive Background Update Method

Another use of the adaptive background model can be found in [11]. The background equations are as follows:

$$\begin{aligned}
A &= AND(M_{bkgnd}, I_{obj.mask})\\
B &= AND(\frac{I_{current\ frame}+15\cdot M_{bkgnd}}{16}, NOT(I_{obj.mask}))\\
M_{bkgnd} &= OR(A,B),
\end{aligned} \tag{2.5}$$

where M_{bkgnd} is the estimated background, $I_{current\ frame}$ is the current image, and $I_{obj.mask}$ is a binary mask containing the foreground detected so far. $I_{obj.mask}$ is, in turn, computed as

$$I_{ojb.mask} = OR(I_{substraction_{m}ask,I_{optical_{m}ask}})$$

$$I_{substraction_{m}ask} = \begin{cases} 0 & if\ \ |I_{current\ frame} - M_{bkgnd}| < K\sigma \\ 1 & otherwise \end{cases} \tag{2.6}$$

where $I_{substraction\_mask}$ is the binary mask of the foreground computed by background subtraction, and I_{optica_mask} is the foreground computed using optical flow estimation as discussed in the following paragraphs. The threshold used is a multiple (K) of the standard deviation σ of camera noise (modeled as white noise). K is determined empirically to be six.

2.2.1.6 Current Applications of Background Update Methods

In some complex cases, parts of the background are almost hidden behind an object; however, in the video stream, we can glimpse them. Figure 2.1 shows such an example. In the middle of this street is a place where people always stand, waiting for the signal. At that place, we can glimpse the background in only a few fragments of the video stream.

We capture an image sequence in the scene presented in the figure. To observe the result of the application of recent methods to this image sequence, we choose the multi-frame average and the selection-average method because the former is basic and the latter is popular. The results are shown in Figures 2.2-2.6. The selection-average method quickly eliminates moving objects. However, objects that move slowly cannot be removed completely. Another disadvantage of the selection-average method is that it needs an initial background. In this case, the initial background is difficult to capture. Here, the initial background in Figures 2.3-2.5 is set as Figure 2.2.

Figure 2.3 is almost the same as Figure 2.2. However, because $T1$ is small, the background cannot be updated correctly.

Figure 2.5 seems to strike a balance between updating and filtering with $T1 = 50$ and $T2 = 3$. Figure 2.6 uses a better initial background captured by the method proposed in the next section.

It can be seen that neither the average- nor the noise filter-based method works in complex scenarios, because the noise is too large to handle. Of course, any method that needs an initial background will not work, either.

In the next section, we propose a new method that can retrieve the background from a noisy video stream.

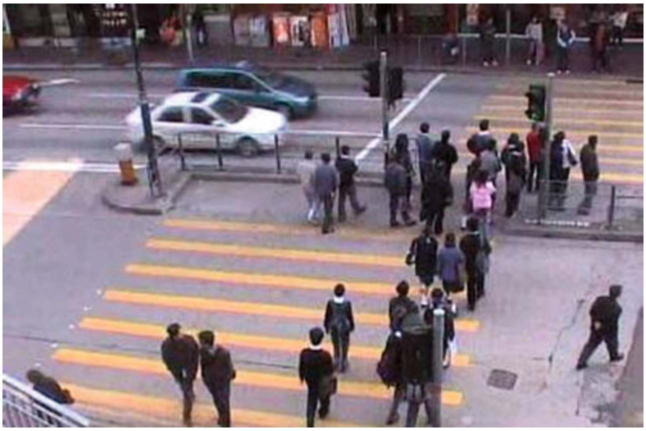

Fig. 2.1 People stay on street

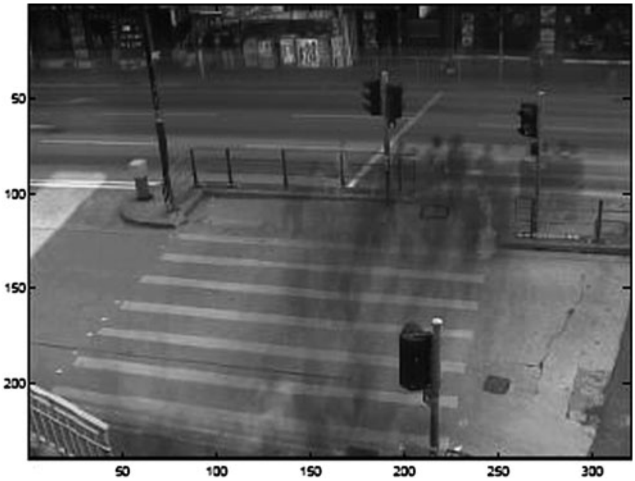

Fig. 2.2 The Result Background of Multi-frame Average Method

2.2.2 Pattern Classification-based Adaptive Background Update Method

Definition: $\sim$

$$\begin{aligned} &For\ two\ n \times m\ images\ I_1,\ I_2. \\ &If\ \ |I_1(i,j) - I_2(i,j)| \leq T_1 \\ &Then\ \ I_1 \sim I_2, \end{aligned} \tag{2.7}$$

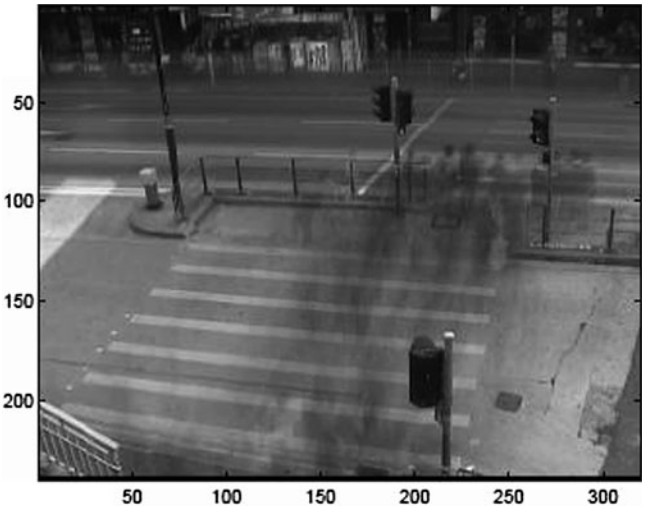

Fig. 2.3 The Result Background of Selection Average Method (T1 = 3 and T2 = 3)

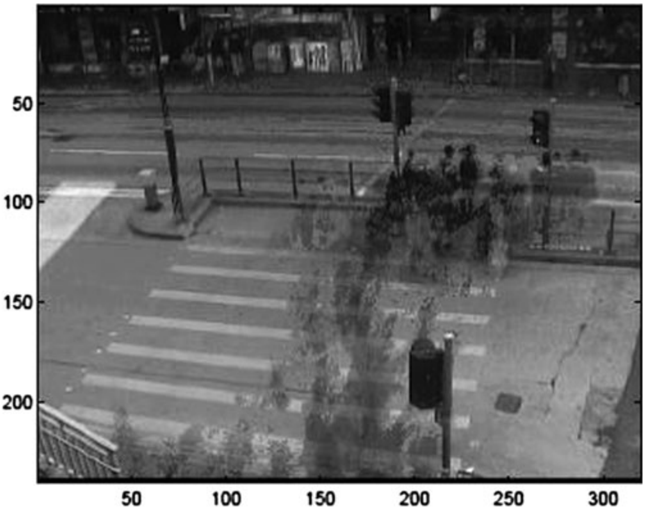

Fig. 2.4 The Result Background of Selection Average Method (T1 = 15 and T2 = 9)

where $1 \leq i \leq n$, $1 \leq j \leq m$, and $I_n(i, j)$ denote the gray value of the pixels located at i columns and j rows in the image I_n, and T_1 is the threshold.

A novel pattern match-based adaptive background update method is expressed as follows. For an image sequence I, there is a classification result B; i.e., for any element $b_i \in B$, if $I_1, I_2 \in b_i$, then we have $I_1, I_2 \in I$ and $I_1 \sim I_2$. b_i has such properties as

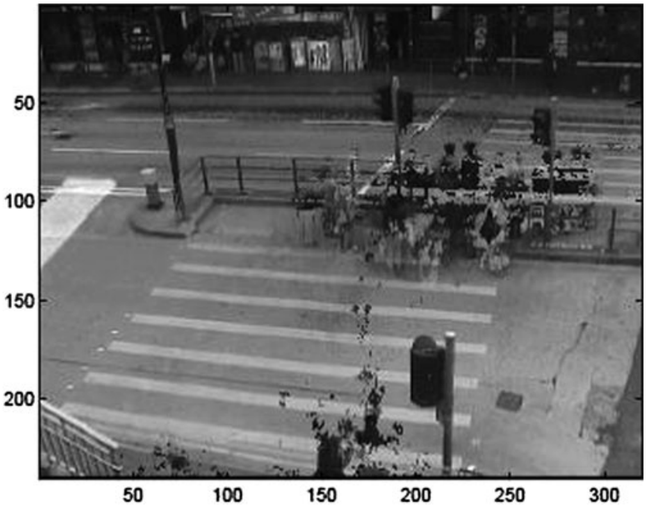

Fig. 2.5 The Result Background of Selection Average Method (T1 = 50 and T2 = 3)$^{(a)}$ (a) The initial background is captured by multi-frame average method proposed in next section

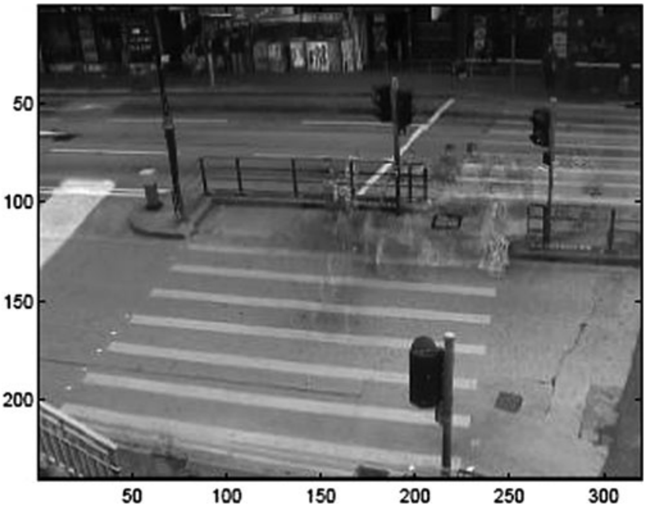

Fig. 2.6 The Result Background of Selection Average Method (T1 = 50 and T2 = 3)$^{(a)}$ (a) The initial background is captured by pattern classification-based method proposed in next section

- $b_i.img = \frac{\Sigma_{I \in b_i} I}{count(B)}$;
- $b_i.lasttime = X$, where X is the maximum suffix of the element in b_i; and
- $b_i.fitness = X - Y$, where X is the maximum suffix of the element in b_i and Y is the minimal one.

Now, the background is defined as $b_i.img$, where $b_i.fitness > Th_f$ *and* $b_i.lasttime > Time_{current} - Th_t$. Here, Th_f and Th_t are two thresholds. There may be many b_i that satisfy this restriction. We can simply choose the one that fits best, or we can deem all of them to be possible background. This method is rewritten into the algorithm below.

```
Step 1:  Set $\mathbb{B} = \emptyset$ as a set of candidate background. Here, $b \in \mathbb{B}$ has three fields:
    $b.img$, $b.lasttime$ and $b.fitness$, which are the background image, the time when
    $b.img$ was captured, and the fitness value of this background candidate.
Step 2:  Capture a image $I_{current}$ from the camera, record the time $t_{current}$ from the
    real-time clock.
Step 3:
    $if(I_{current} \sim b \text{ and } b \in \mathbb{B})\{$
        $b.img = k * I_{current} + (1-k) * b.img;$
        $b.fitness = b.fitness + (t_{current} - b.lasttime);$
        $b.lasttime = t_{current};$                                   (2.8)
        $goto$ **Step 2**;
    $\}$
Step 4:
    $if(I_{current} \sim I_{current-1})\{$
        $allocate\ a\ new\ candidate\ b_{current}.$
        $b_{current}.img = k * I_{current} + (1-k) * I_{current-1};$
        $b_{current}.fitness = 0;$
        $b_{current}.lasttime = t_{current};$
        $add\ b_{current}\ into\ \mathbb{B};$
    $\}$
Step 5:
    $if(element\ number\ in\ \mathbb{B} > POLLMAX)$
      $delete\ a\ candidate\ b_0\ in\ \mathbb{B},\ which\ has\ the\ lowest\ fitness,\ and\ is'nt\ b_{current};$
Step 6:
    $for\ any\ b \in\ \mathbb{B}$
    $if(b.fitness > Th_f\ and\ b == b_{current});$
      $b_{active} = b;$
    $else$
      $set\ b_{active}\ as\ the\ one\ has\ the\ largest\ fitness.;$
```

Step 7: Repeat **Step 6** again to get b_{active_2}.
Step 8: Output b_{active} and b_{active_2};
Step 9: Goto **Step 2**;

Th_f is a threshold of time, which indicates when the background should be updated after it changes. *POLLMAX* is set to 32. In the results, b_{active} is the updated background, and b_{active_2} is a candidate background, which should be the background in shadow.

Figure 2.7 shows selected frame segments in the image sequence, which are 80 × 80 pixels in size. In the figure, the number below the frames is the frame ID, and can is the variable $lasttime$ in Equation 2.8. Figure 2.8 shows the classes taken from the frame segments. In this case, the background should be the class $\{301\}$, whose $fitness$ is 297. The results for this method run with the same video stream are illustrated in Figure 2.9, which shows that all moving objects are eliminated. Figure 2.10 compares the subtraction results using the pattern classification- and average-based methods, which reveal that the former has the advantage over the latter.

The pattern classification method can easily be rewritten for color images. Figure 2.11 shows the color image after background subtraction.

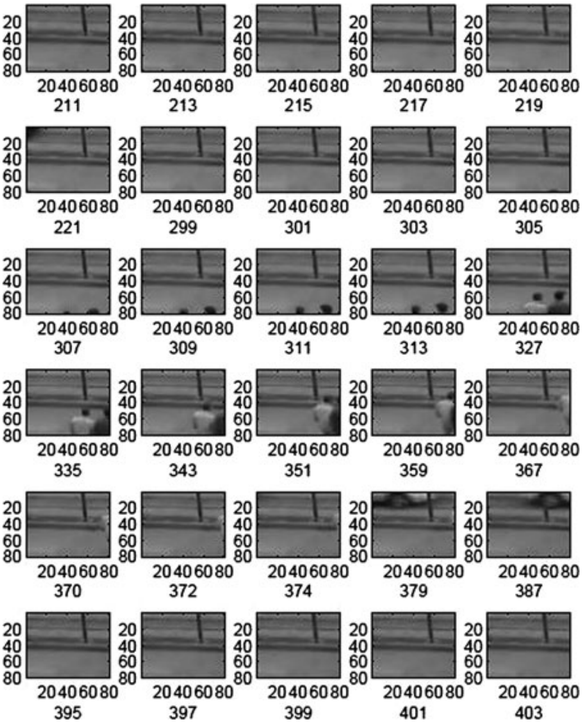

Fig. 2.7 A Selected Frame Segments

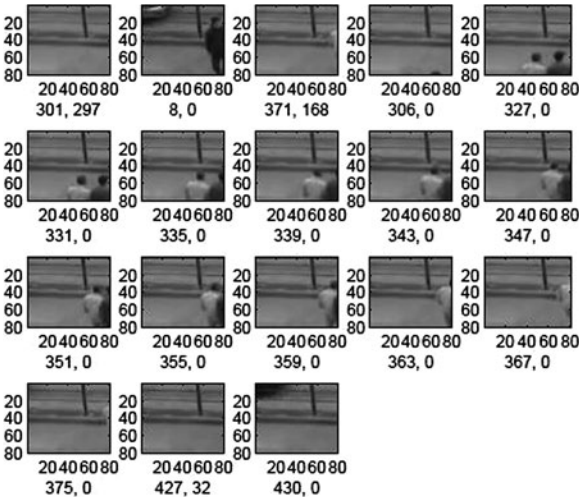

Fig. 2.8 The Classes in The Segments

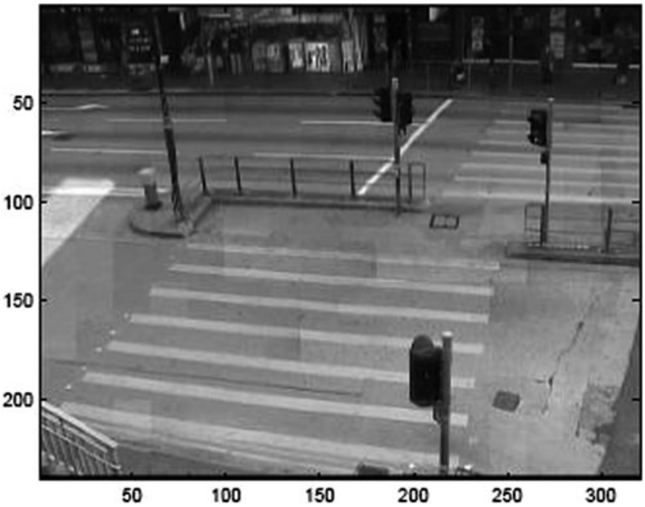

Fig. 2.9 The Background Captured by Pattern Classification Based Method

2.3 Frame Differencing Method

Based on the assumption that human objects are normally in motion, we adopt the frame differencing method to detect human targets. This method requires only that the camera be kept static for a certain period, so it is applicable to moving cameras.

Frame differencing is the simplest method for the detection of moving objects, because the background model is equal to the previous frame. After performing a binarization process with a predefined threshold using the differencing method, we can find the target contour, and the target blob is obtained through the contour-filling process. However, if the target is moving very quickly, then the blob may contain too many background pixels, whereas if the target is moving very slowly, then target information in the blob may be lost. It is impossible to obtain solely foreground pixels when using the frame difference as the background model, but by using the following method, we can remove the background pixels and retrieve more foreground pixels, on the condition that the color of the foreground is not similar to that of the pixels of the nearby background. By separately segmenting the foreground and background in a rectangular area, we can label and cluster the image in the rectangular area again to obtain a more accurate foreground blob.

Figure 2.12 shows the foreground detection process in the frame differencing method, where (a) and (b) show the target detection results using the frame differencing method and blob filling results, respectively. In Figure 2.12 (c), it can be seen that the left leg of the target is lost. After the labeling and clustering process, we can retrieve the left leg (see Figure 2.12 (d)).

Feature selection is very important in tracking applications. Good features result in excellent performance, whereas poor ones restrict the ability of a system to distinguish the target from the background in the feature space. In general, the most desirable property of a visual feature is its uniqueness in the environment. Feature selection is closely related to object representation, in which the object edge or shape feature is used as the feature for contour-based representation, and color is used as a feature for histogram-based appearance representation. Some tracking algorithms use a combination of these features. In this chapter, we use color and spatial information for feature selection. The apparent color of an object is influenced primarily by the spectral power distribution of the illumination and the surface reflectance properties of the object. The choice of color and space also influences the tracking process. Digital images are usually represented in the red, green, blue (RGB) color space. However, the RGB color space is not a perceptually uniform color space because the differences among the colors do not correspond to the color differences perceived by humans. Additionally, RGB dimensions are highly correlated. Hue, saturation, and value (HSV) give an approximately uniform color space, which is similar to that perceived by humans; hence, we select this color space for this research. The color information alone is not sufficient. If we combine it with the spatial distribution information, then the selected features become more discriminative.

The main task is to segment the image using this feature. We choose the spatial-color mixture of Gaussians (SMOG) method to model the appearance of an object and define the Mahalanobis distance and similarity measure [168]. We then employ the k-means algorithm followed by a standard expectation maximization (EM) algorithm to cluster the pixels. Our approach is different in that we do not cluster and track the whole region but only the moving target in a rectangle, as described in the

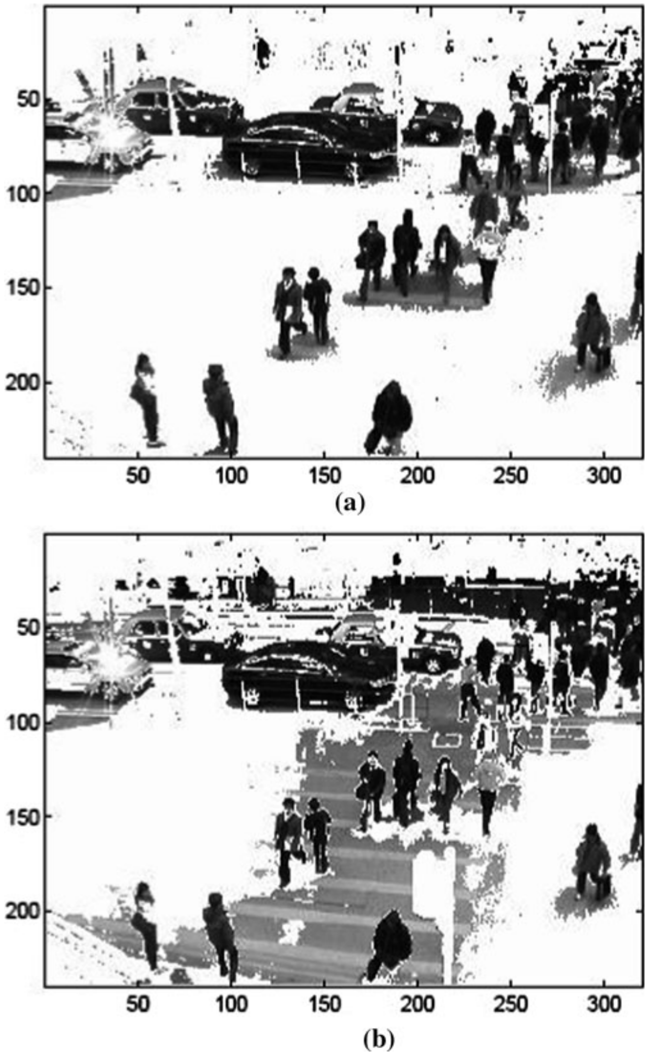

Fig. 2.10 Example of Background Subtraction Result Operated By: (a) Pattern Classification Method, (b) Average Method

previous section. Figure 2.13 shows the clustering and tracking results for the whole region of the rectangle, and Figure 2.14 shows them for the moving target.

A standard method that clusters and tracks objects in the whole region of a rectangle can track targets properly but requires more particles and computation time, because the whole region contains many background pixels. When we choose a new particle at any place in the image, the similarity coefficient is likely to be high. Thus, more particles are required to find good candidate particles from the complex

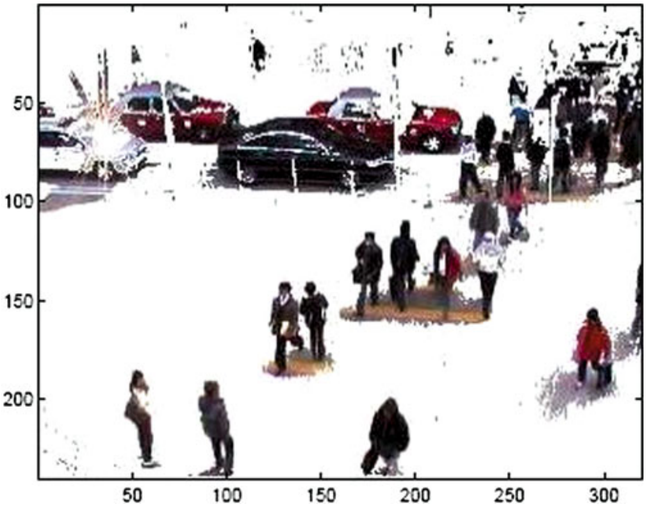

Fig. 2.11 The Color Version of The Background Subtraction Result

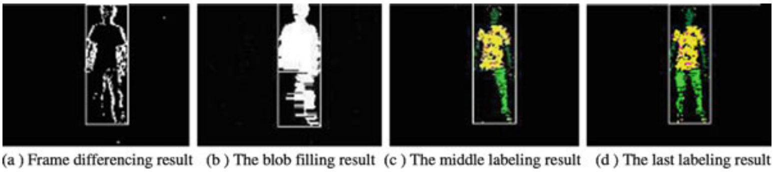
(a) Frame differencing result (b) The blob filling result (c) The middle labeling result (d) The last labeling result

Fig. 2.12 Foreground detection process by frame differencing

background. We use the standard method to track a target in a real case, and find that the frame rate can reach 10 frames per second (fps) at a resolution of 160*120 pixels.

In contrast, if we cluster and track only the moving target, then fewer particles and less time are needed and the frame rate can reach 15 fps. To save computation time, therefore, the system clusters and tracks only moving targets.

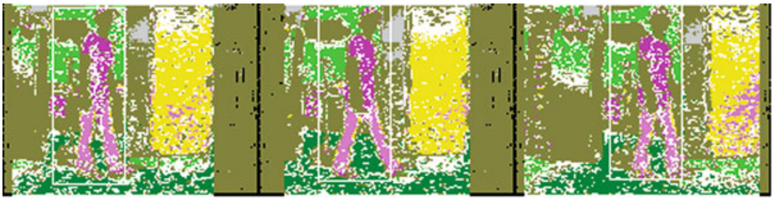

Fig. 2.13 Clustering and tracking results on the whole region in rectangles

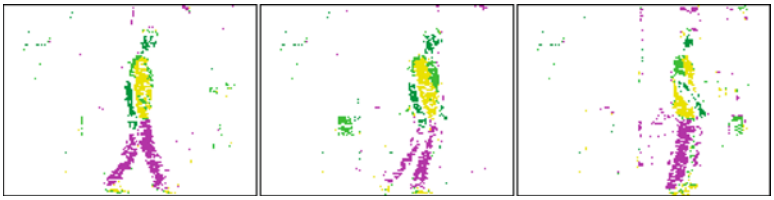

Fig. 2.14 Clustering and tracking results on moving targets

2.4 Optical Flow Method

Blob detection is the first task in video surveillance systems, because the precise process of segmentation and tracking depends heavily on the ability of a system to distinguish the foreground from the background. The usual approach is to subtract the background image, but such an approach fails when the environment becomes crowded as the background image becomes unavailable. Optical flow is one way to solve this problem [51] [57]. In our research, we use an optical flow algorithm that is independent of the background frame. Figures 2.15 and 2.16 demonstrate the experimental results for blob detection using optical flow. In a crowded environment in which a background frame is not available, the traditional subtraction algorithm fails to detect blobs, whereas the optical flow one performs well.

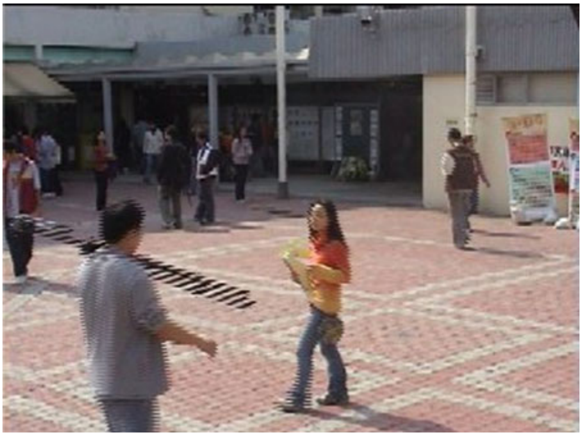

Fig. 2.15 Current frame

The traditional subtraction method can process information faster than the optical flow-based approach because of its simplicity, but is dependent on the availability of a background image. This means that in crowded environments, including campuses, shopping malls, subways, and train stations, the subtraction approach will fail, whereas the optical flow-based approach will work well. Although the latter approach is expensive in terms of computation, the use of super workstations makes its application feasible.

2.5 Conclusion

In this chapter, we introduce the first processing step, which is based on the acquired image, i.e., extracting the foreground, which the latter steps concern, from the background. Background and foreground detection are two sides of the same problem, which means that the solution of one will result in the solution of the other.

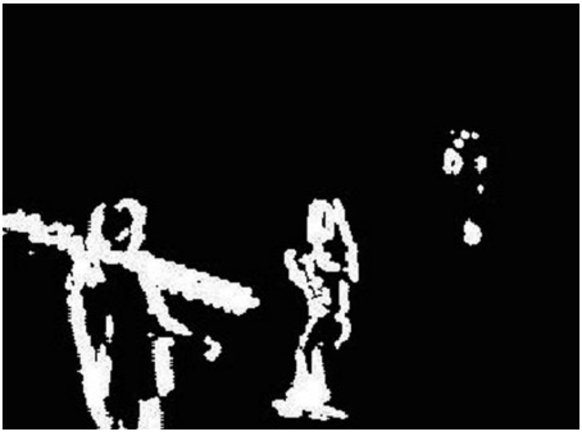

Fig. 2.16 Blob detection by optical flow

One approach is to develop a background model based on the pattern classification method, which adaptively updates the background, even at traffic crossings with heavy vehicle and pedestrian streams. Another is to identify moving targets in the foreground based on dynamic image information. The frame differencing method utilizes two consecutive frames to detect moving contours followed by a filling process to obtain the foreground. The optical flow method to detect moving objects is another foreground detection approach.

After the detection of the background and foreground, we can analyze the foreground blobs and connect different blobs in the spatial-temporal relationship for a given time domain.

Springer

http://www.springer.com/978-94-007-1136-5

Intelligent Surveillance Systems
Qian, H.; Wu, X.; Xu, Y.
2011, XX, 168 p., Hardcover
ISBN: 978-94-007-1136-5