The following manuscript was published in

Chia-Jung Chang, Jun-Wei Hsieh, Yung-Sheng Chen, Wen-Fong Hu, **Tracking multiple moving objects using a level-set method**, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 2, 101-125, 2004.

TRACKING MULTIPLE MOVING OBJECTS USING A LEVEL-SET METHOD

Chia-Jung Chang, Jun-Wei Hsieh*, Yung-Sheng Chen, and Wen-Fong Hu

Department of Electrical Engineering

Yuan-Ze University

135 Yuan-Tung Rd., Nei-Li, Chung-Li

Taoyuan, 32026, Taiwan, R.O.C.

---

* To whom all correspondence should be sent.
JunWei Hsieh
Department of Electrical Engineering, Yuan-Ze University
135 Yuan-Tung Rd., Nei-Li, Chung-Li, Taoyuan, 32026, Taiwan, R.O.C.
Tel: 886-3-463-8800 Ext.430
E-mail: shieh@saturn.yzu.edu.tw

Abstract

This paper presents a novel approach to track multiple moving objects using the level-set method. The proposed method can track different objects no matter how they are rigid, non-rigid, merged, split, with shadows, or without shadows. At the first stage, the system proposes an edge-based camera compensation technique for dealing with the problem of object tracking when the background is not static. Through such camera compensation, different moving pixels can be easily extracted through a subtraction technique. Thus, a speed function with three ingredients, i.e., pixel motions, object variances, and background variances, can be accordingly defined for guiding the process of object boundary detection. According to the defined speed function, different object boundaries can be efficiently detected and tracked by a level-set-based method with a curve evolution technique. Moreover, in order to further understand the video content, this paper creates a relation table to identify and observe different behaviors of tracked objects. However, the above analysis often fails due to the existence of shadows which commonly appear in the analyzed sequence. Therefore, this paper adopts a technique of Gaussian shadow modeling to remove all unwanted shadows. Experimental results show that the proposed method is much more robust and powerful than other traditional methods.

Keywords: Level-set methods, object-relation table, speed function, video surveillance, background compensation, shadow elimination.

1 Introduction

Tracking multiple moving objects is an important problem in many applications like video surveillance [1]-[4], video retrieval [12], teleconference, and so on. In the past, there have been a number of researchers [1]-[11] who have devoted themselves to investigating different methods for solving this problem. These methods can be categorized according to the following two issues: how to characterize the tracked objects and how to track them. For the first issue, we can describe an object with different features like curvatures [2], colors [4], region textures [13], contours [11], or moments. Among them, the most commonly used feature for object tracking is "contour" [11]. However, before tracking, this approach requires the contours being manually assigned in advance, that is, some markers are previously required to be set along the boundaries of the tracked objects. More importantly, the contour-based method cannot handle the problem if the tracked objects have some merging or splitting activities. For the second issue, two common schemes are the Kalman filter [14] and the block matching technique [15]. The Kalman filter is a useful tool for motion prediction but time-consuming in real-time applications. The block matching technique is effective in slow object motion estimation but often fails when images have large light changes. Recently, the level set method has become more popular in object tracking. The technique converts the tracking problem into a higher dimensional space, from which the desired optimal solution can be more accurately derived. Thus, better convergence properties can be found and lead to many successes in various applications [5], [16]. In this approach, a curve is embedded as a zero level set of a higher dimensional surface function. Through defining a proper speed function, the desired curve is gradually evolved and obtained by minimizing a cost function, which is calculated locally based on image gradients curvatures but globally on shapes and poses of the objects. With a simple initialization, the moving objects even split can be well tracked. However, in case that the background is not static, this approach will fail to track objects since motion information cannot be well obtained. On the other hand, if any unexpected shadows exist, the approach also cannot accurately locate desired objects and result in the failure of consequent analysis works, e.g., counting objects, recognizing objects, analyzing the behaviors of these objects, and so on.

In this paper, a novel approach is proposed for solving different problems in tracking multiple objects by taking advantages of a level-set technique. In this scheme, an edge-based camera compensation method is first proposed to estimate desired camera motions from video sequences for making the background be static. Different from other

compensation techniques, this method has better capabilities to stitch images having large displacements and light changes. Then, pixel motions can be easily extracted through a background subtraction. The motion information can drive a speed function to provide different pushing or pulling forces for locating desired objects by using a level-set method. The speed function is parameterized with three ingredients including pixel motions, object variances, and the variance of background. Then, desired moving objects can be successfully detected and tracked through curve evolutions. Furthermore, an object relation table is created to analyze different object behaviors. However, the success of all above video analysis mainly depends on the assumption that analyzed sequences cannot include any shadows. For most tracking approaches [1]-[11], if some unexpected shadows exist, the accuracy of following content analysis will be much disturbed and degraded. Therefore, in this paper, a coarse-to-fine approach is adopted to eliminate unwanted shadows using a shadow modeling technique. Different from other color or intensity-based methods, this method takes advantages of shadow geometries to fully remove unwanted shadows from video sequences. In this paper, a Gaussian function with three features including illumination properties, positions, and orientation of shadows is adopted to model various shadows. The features can precisely reflect different shadow properties at various conditions and thus provide good capabilities for completely eliminating unwanted shadows. Experimental results are provided to prove the superiority of the proposed system.

Section 2 briefly describes the level set method. Details of the proposed method are described in Section 3. Section 4 describes details of the shadow elimination method and Section 5 shows the experimental results. Some concluding remarks are made in Section 6.

2 Level Set Method

The level set technique was proposed by S.J. Osher and J.A. Sethian [5]. The technique has been applied into a wide range of applications, including fluid mechanics [5]-[6], combustion [5], computer animation, image processing [6]-[7],[16]-[19], and so on. In what follows, some brief reviews of the level set method are discussed.

2.1 Zero Level Set

Like Fig.1(a), the black circle represents an object's boundary. The object boundary will change according to different time like Fig. 1(b). The *z* coordinate is the level set function and the set of the curve when $=0$ is named as the zero level set. When the time changes, the level set function and the object boundary will also change. However, the new boundary can be easily obtained from the zero level set $=0$.

2.2 Level Set Equation

In order to track an object boundary, the level set function is set to be zero, i.e.,

$$(\mathrm{X}(t),t)=0\,, \tag{1}$$

where $\mathrm{X}(t)$ is the vector of any dimension coordinates of an object and will change according to t. Then, based on the chain rule, we have

$$_t(\mathrm{X}(t),t)+\nabla\ (\mathrm{X}(t),t)\cdot\mathrm{X}'(t)=0\,, \tag{2}$$

where $_t(\mathrm{X}(t),t)=\dfrac{\partial\ (\mathrm{X}(t),t)}{\partial t}$ and $\nabla\ (\cdot)$ is the gradient of $(\cdot)$. It is assumed that the boundary contour moves toward its normal direction with the speed *F* defined as follows:

$$F=\mathrm{X}'(t)\cdot\vec{n}\,, \tag{3}$$

where $\vec{n}=\dfrac{\nabla}{|\nabla\ |}$. Then, Eq.(2) can be rewritten as

$$_t(\mathrm{X}(t),t)+F|\nabla\ \ (\mathrm{X}(t),t)|=0\,. \tag{4}$$

That is the so-called the level set equation. Since $(X(t),t)$ is a continuous form, we use a uniform mesh of spacing *h*, with grid nodes (*i*, *j*), to obtain a discrete approximation of the solution $(ih,jh,n\Delta t)$ as $_{ij}^{n}$. Then, Osher and Sethian [5] gave its iterative solution as:

$$_{ij}^{n+1}=\ _{ij}^{n}-\Delta t\cdot F_{ij}[max(D^{-x}\ _{ij}^{n},0)^2+min(D^{+x}\ _{ij}^{n},0)^2+max(D^{-y}\ _{ij}^{n},0)^2+min(D^{+y}\ _{ij}^{n},0)^2]^{1/2}, \tag{5}$$

where Δt is the time difference, D^{-x} and D^{-y} the backward differences in the x and y coordinates, D^{+x} and D^{+y} the forward difference in the x and y coordinates, $\max(\bullet)$ a maximum function, and $\min(\bullet)$ a minimum function.

3 Multiple Object Tracking System Using a Level Set Technique

As discussed before, in this paper, a novel approach is proposed for tracking multiple objects based on the level set technique. This approach includes five components, i.e., camera compensation, background subtraction and thresholding, object tracking, shadow elimination, and object status analysis. Different from other level-set based methods, the proposed system can handle the problem when the background is not static and the case when some unwanted shadows appear in the analyzed sequence. The whole flowchart of this system is illustrated in Fig. 2. At the first stage, this approach uses an edge-based algorithm to estimate possible camera motions from video sequences for camera compensation. Then, a background subtraction technique is used to obtain all desired pixel motions. After that, a thresholding technique is applied to roughly estimate different variances of tracked objects and the background. According to the estimated variances, a speed function *F* to drive

curve evolution can be accordingly defined for detecting desired moving objects with the level set technique. Furthermore, for content analysis, an object relation table is then created to analyze different behaviors of tracked objects. In practice, the analyzed video sequence often includes many unwanted shadows, which will significantly degrade the accuracy of video analysis. Therefore, a shadow elimination scheme is then adopted for removing all unwanted shadows. In order to keep the readability of this paper, this section assumes the input sequence has no moving shadow. Then, the method to eliminate unwanted shadows is described in Section 4. In what follows, details of each stage of the system except the shadow elimination are discussed.

3.1 Background Compensation

When analyzing video sequences, if the camera is not static, the background will change at different time and should be first compensated for well tracking objects. In this paper, the camera is assumed to have only translation movements ever larger. Then, an edge-based method is proposed to estimate all possible translations between any two adjacent frames. The proposed method has better capabilities to compensate images if they have large light changes and large displacements.

Assume I_a and I_b are two frames prepared for compensation and shown in Fig. 3 (a) and (b), respectively. Through a vertical edge detector, the positions of vertical edges in I_a and I_b can be obtained and recorded, respectively, as P_a^v = (100, 115, 180, 200, 310, 325, 360, 390, 470) and P_b^v = (20, 35, 100, 120, 230, 245, 280, 310, 390). The vertical edge detector is simple and can be described as follows. Let $g_x(p)$ denote the gradient of a pixel *p* in the *x* direction, e.g.

$$g_x(p(i,j)) = |I(p(i+1,j)) - I(p(i-1,j))|,$$

where $I(p)$ is the intensity of *p*. In addition, let $S_g(i)$ denote the sum of $g_x(p)$ obtained by accumulating $g_x(p)$ along pixels in the *i*th column. If $S_g(i)$ is larger than a threshold, the *i*th column is considered to have a vertical edge. After checking pixels column by column, a set of positions of vertical edges can be found.

Since the images I_a and I_b come from the same scene, there should exist an offset d_x such that $P_a^v(i) = P_b^v(j) + d_x$ and the corresponding relation between *i* and *j* is one-to-one. Then, the offset d_x is the desired translation between I_a and I_b in the *x* direction, i.e., d_x = 80. Based on this idea, we can present a novel method to estimate desired translations

without building any correspondences or involving any optimization processes. In practice, due to noise, some edges will be lost or undetected. The lost or undetected edges will cause that the relations between P_a^v and P_b^v are no longer one-to-one. In order to deal with this problem, this paper defines a function $d_v(i,k)$ to measure the distance of a position $P_a^v(i)$ to the translation solution *k* as

$$d_v(i,k) = \min_{1 \le j \le N_b^v} |P_a^v(i) - k - P_b^v(j)|, \quad (6)$$

where N_b^v is the number of elements in P_b^v. Let T_d be a threshold and set to 4. Given a number *k*, we want to determine the number N_p^v of elements in P_a^v whose $d_v(i,k)$ is less than T_d. In addition, the average value of $d_v(i,k)$ for these N_p^v elements is calculated as E_k^v, which is an index to measure the goodness of *k* to see whether it is a suitable translation. If E_k^v is smaller enough and N_p^v is larger enough, the position *k* can be considered as a good horizontal translation. In more precise words, if $E_k^v \le T_e$ and $N_p^v \ge T_p$, *k* is collected as an element of the set S_x of possible horizontal translations, where T_p and T_e are two thresholds and set to 5 and 2, respectively. Let W denote the width of input images. Through examining different *k* for all $|k| < W$, the set S_x can be obtained.

On the other hand, let P_a^h and P_b^h denote as two sets of horizontal edge positions in I_a and I_b, respectively. With P_a^h and P_b^h, we also can define a distance function d_h as:

$$d_h(i,k) = \min_{1 \le j \le N_b^h} |P_a^h(i) - k - P_b^h(j)|, \quad (7)$$

where N_b^h is the number of elements in P_b^h. Let H denote the height of input images. According to the d_h, with the similar method to obtain S_x, by examining different *k* for all $|k| < H$, the set S_y of possible vertical translations can be obtained. With S_x and S_y, the set S_{xy} of possible solutions can be obtained by: $S_{xy} = \{(x,y) \mid x \in S_x \text{ and } y \in S_y\}$.

Once S_{xy} is obtained, we want to determine the best translation from S_{xy} through a block matching technique. In this technique, the measure of sum of intensity differences is used to denote the difference between two region blocks, i.e.,

$$D(p,q) = \sum_{x,y=-M}^{x,y=M} |I_a(x+p_x, y+p_y) - m_a - I_b(x+q_x, y+q_y) + m_b|, \quad (8)$$

where m_i is the local mean and variance of I_i and $(2M+1)^2$ represents the area of

matching window. Due to the small size of S_{xy}, the best solution of translation can be obtained very quickly. In addition, through edge alignment, the method will restrain many impossible translations from being the best solution. Therefore, even under different light conditions, the proposed method still can perform better than other methods in finding desired translations. The whole algorithm is summarized in details as follows.

Edge-based Translation Estimation Algorithm:

Assume I_a and I_b are two adjacent image frames.

Step 1: Apply a vertical edge detector to find the sets P_a^v and P_b^v of vertical edge positions from I_a and I_b, respectively.

Step 2: Determine the set S_x of possible horizontal translations from P_a^v and P_b^v based on Eq.(6).

Step 3: Apply a horizontal edge detector to find the sets P_a^h and P_b^h of horizontal edges from I_a and I_b, respectively.

Step 4: Determine the set S_y of possible vertical translations from P_a^h and P_b^h based on Eq.(7).

Step 5: Let S_{xy} denote the set of possible translations, i.e.,

$$S_{xy} = \{(x, y) \mid x \in S_x \text{ and } y \in S_y\}.$$

Determine the best solution $V_e = (v_x, v_y)$ from S_{xy} through a matching technique, i.e.,

$$V_e = \arg\min_{r \in S_{xy}} D(P_c, P_c - r), \tag{9}$$

where P_c is the central point of I_a and D is the measure defined in Eq.(8).

Assume I_k and I_{k-1} are two consecutive frames prepared for camera compensation. With the proposed camera compensation algorithm, the camera motion vector $V_{k,e}$ can be estimated from I_k and I_{k-1}. Let W_k and H_k denote the width and height of $I_k(x, y)$, respectively. Based on $V_{k,e}$, the background can be compensated as follows:

$$\bar{I}_k(x, y) = \begin{cases} \bar{I}_{k-1}(x - V_{k,e}^x, y - V_{k,e}^y), \text{ if } V_{k,e}^x \le x \le W_k, V_{k,e}^y \le y \le H_k, \\ I_k(x, y), \text{ else,} \end{cases} \tag{10}$$

where $\bar{I}_k(x, y)$ and $\bar{I}_{k-1}(x, y)$ are the kth and $(k-1)$th frames after compensation and

$(V_{k,e}^x, V_{k,e}^y)$ the coordinates of $V_{k,e}$. Fig. 4 shows the result of camera compensation, where (c) is the result by applying the above camera compensation method to (a) and (b).

3.2 Background Adaptation

After camera compensation, the background of a surveillance scene can be considered still. However, due to different light changes, the background will gradually change at different time. In order to overcome this problem, a background adaptation scheme should be adopted for well tracking objects. Assume $I_k(x, y)$ and $B_k(x, y)$ are the kth frame and the background respectively. Using the previous L images, the background can be adapted by:

$$B_k = \sum_{i=1}^{L} w_i I_{k-i}, \tag{11}$$

where L is set to be 100 in this paper and w_i is the weighting factor. Let m^k be denoted as the mean of I_k. Then, the w_i can be determined according to the distance between m^k and m^{k-i} and the time difference of I_{k-i} to current frame I_k as follows:

$$w_i = (1-\frac{i}{L})(1+|m^i - m^{i-1}|)^{-1}\Omega^{-1},$$

where $\Omega = \sum_{i=1}^{L} \frac{1}{1+|u^i - u^{i-1}|}(1-\frac{i}{L})$.

3.3 Image Difference and Binarization

In this paper, the level set technique is used to detect and track different moving objects. The success of this technique strongly depends on a proper speed function F for guiding the process of curve evolutions. This paper uses three ingredients for defining the F, that is, the pixel motions, object variances, and background variances. In what follows, details of methods to obtain the above three ingredients are described.

Assume $dI_k(x, y)$ is the k th differencing image, i.e.,

$$dI_k(x, y) = |I_k(x, y) - B_k(x, y)|. \tag{12}$$

Then, according to $dI_k(x, y)$, a thresholding technique [21] is applied to obtain possible variances of objects and background, respectively. Let $_g$ denote the global variance of the image like Fig. 5. In addition, $_1$ and $_2$ are the variances of inside and outside parts separated by the dotted line. The optimal threshold T for binarizing an image can be found by minimizing the within-group variance as follows:

$$t = \arg\min_t \ {}^2_w(t) \text{ and } {}^2_w(t) = q_1 \ {}^2_1(t) + q_2 \ {}^2_2(t) \tag{13}$$

where $q_1(t) = \sum_{i=0}^{t} P(i)$, $q_t(t) = \sum_{i=t+1}^{G} P(i) = 1 - q_1(t)$, G the maximum gray value of dI_k, and $P(i)$ the occurrence probability of intensity i in $dI_k(x, y)$. The global variance ^2_g is defined as:

$$^t_g(t) = \sum_{i=0}^{G} | i - \ _g(t) | P(i), \tag{14}$$

where $\ _g(t) = \sum_{i=0}^{G} iP(i)$. From Eq.(14), after some calculations, we have

$$^2_g(t) = \ ^2(t) + q_1(t)q_2(t)[\ _1(t) - \ _2(t)]^2.$$

Then, we have

$$^2_g(t) = \ ^2_w(t) + \ ^2(t), \tag{15}$$

where $^2(t) = q_1(t)q_2(t)[\ _1(t) - \ _2(t)]^2$. Since the ^2_g is a constant, the problem to find a threshold T that minimizes $\ _w$ becomes finding a threshold T which leads to the maximum of ^{2}. Since the number of possible values of T is small, the optimal T can be easily found by trying all the possible values of T which maximizes ^{2}. With this idea, details of finding the optimal threshold T can be illustrated as follows:

Set $Old\_s^2(t) = 0$;

For t = 0 to G

Compute q_1, m_1, q_2, *and* m_2 ;

Compute $new\_s^2(t) = q_1(t)q_2(t)[m_1(t) - m_2(t)]^2$;

If $New\_s^2(t)$ > $Old\_s^2(t)$, *then*

Set $T = t$;

$Old\_s^2(t) = New\_s^2(t)$;

End If

End For

Once T has been obtained, rough boundaries between objects and backgrounds can be estimated and thus their variances $^2_{object}$ and $^2_{background}$ can be, respectively, calculated.

3.4 Object Detection

Once the variances $^2_{object}$ and $^2_{background}$ have been obtained, in what follows, we will describe a level-set method to detect and track different moving objects from video sequences. As defined before, *dI(x,y)* is the different image between two consequent frames. According

to $^{2}_{object}$, and $^{2}_{background}$, we define a speed function F as this equaiton:

$$F(x,y)=1\text{-}\exp(\frac{Max\text{-}|\nabla dI(x,y)|}{2s^{2}}) \quad \text{and} \quad =\begin{cases} _{object}, & if\ dI(x,y)\geq T; \\ _{background}, & if\ dI(x,y)<T, \end{cases} \tag{16}$$

where $|\nabla dI(x,y)|$ is the gradient of $dI(x, y)$, Max the maximum of $|\nabla dI(x,y)|$, and T the threshold obtained from Eq.(13). If $|\nabla dI(x,y)|$ approaches the maximum, then the desired object curve gradually attains a zero speed as it gets closer to object boundaries. The gradient operator is a direction-dependent operation. For detecting any object motions, it is better to use a direction-invariant operator to calculate $|\nabla dI(x,y)|$. Thus, we calculate $|\nabla dI(x,y)|$ with this form:

$$|\nabla dI(x,y)|=\frac{1}{8}\sum_{i=1,2}(|dI(x+i,y+i)-dI(x-i,y-i)|+|dI(x+i,y-i)-dI(x-i,y+i)|) \tag{17}$$

$$+\frac{1}{8}\sum_{(i,j)=(1,2)or(2,1)}(|dI(x+i,y+j)-dI(x-i,y-j)|+|dI(x+i,y-j)-dI(x-i,y+j)|).$$

Assume $X(t)$ and $C(t)$ are an object point and the desired object boundary at time t. If the level set function $(X(t),t)$ is a distance function of $X(t)$ to $C(t)$, $C(t)$ can be detected from $(X(t),t)$ by tracing all points satisfying $(X(t),t)=0$. Let d_{CB} denote the chess board distance as:

$$d_{CB}[(i,j),(h,k)]=\max(|i-h|,|j-k|). \tag{18}$$

Then, the discrete approximation of $(X(t),t)$ is then represented by

$$^{n+1}_{ij}=\min_{(h,k)\in C^{n}} d_{CB}[(i,j),(h,k)], \tag{19}$$

where C^{n} is the approximation of the zero level set $\{ =0\}$ from $^{n}_{ij}$. Initially, when n=0, the value of $^{n}_{ij}$ is set to negative if the point *(i,j)* is inside the initial curve C^{0} and to positive if (i, j) is outside C^{0}. Like Fig. 6(a), the red curve is the initial curve C^{0}. All points on the curve C^{0} are set to zero, negative, and positive if they are on, inside, and outside C^{0}, respectively. When evolutions, the narrow-band updating technique [5] is used here for speeding up the efficiency of object tracking. The whole object detection and tracking algorithm can be summarized in details as follows:

Step 1: Initialize the level set function $^{0}_{ij}$ and set n=0.

Step 2: Calculate $^{n+1}_{ij}$ with the speed function F, C^{n}, and $^{n}_{ij}$ according to Eqs.(5), (16) and (19), respectively.

Step 3: Find the curve C^{n+1} from $\phi_{ij}^{n+1}=0$ with the narrow-band technique [5]. Set $n = n+1$.

Step 4: If C^{n+1} is not changed (the same as C^{n}) or the iteration is enough, then stop; otherwise go to Step 2.

Fig. 6 and Fig. 7 and show the results when different pedestrians appear in the video sequence. Different from other tracking techniques, the proposed level-set method has good capabilities in tracking objects even they have been split or merged. Fig. 16 shows the results when observed objects have some merging and splitting conditions.

3.5 Analyzing Object Statuses

Once different moving objects have been extracted, we should analyze their statuses across different frames based on an object relation table for observing their behaviors. The analysis can provide important information for achieving different security-related applications, like stealer detection, home security maintenance, parking management, and so on. The object relation table is a 2-D matrix $R_{i,j}$, where i denotes the ith object in the current frame and j is the jth object in the previous frame. The value of $R_{i,j}$ is set to one if the objects i and j have some overlapping; otherwise, $R_{i,j}$ is zero. For example, Fig. 8 shows two objects found in two adjacent frames, where objects with gray color denote they appear in the previous frame and the white color denotes they are in the current frame. Then, an object relation table can be created as Table 1. In this table, two measures SOC (Sum of Column) and SOR (Sum of Row) are used to analyze different object behaviors. The SOC and SOR are defined, respectively, as follows:

$$SOC(i)=\sum_{j} R_{i,j} \text{ and } SOR(j)=\sum_{i} R_{i,j}.$$

According to the values of SOC and SOR, we can categorize object behaviors into six different statuses as follows:

Case 1: if $SOC(i)$ is zero, the object i in the current frame is a new object.

Case 2: if $SOC(i)$ is one, the object i appears both in the current and previous frames.

Case 3: if $SOC(i)$ is larger than 1, there are $SOC(i)$ objects appearing in the previous frame and merging together into the object i in the current frame like the case in Fig. 9.

Case 4: if $SOR(j)$ is zero, then the object j in the previous frame disappears in the current frame.

Case 5: if $SOR(j)$ is one, the object j appears both in the current and previous frames.

Case 6: if $SOR(j)$ is larger than one, the object j in the previous frame is split into different objects in the current frame like the case shown in Fig. 10.

According to the above analysis, different strategies can be applied for handling different surveillance situations.

4. Object Shadow Elimination

In the previous section, through the level set method, different objects have been extracted from video sequences. However, due to sunshine or other light conditions, some undesired shadows will appear in these sequences and lead to the failure of consequent analysis works like counting objects, estimating object locations, analyzing object behaviors, and so on. For tackling this problem, in this paper, a coarse-to-fine approach is used to eliminate all unwanted shadows. This technique also has been discussed in our previous work [20]. Firstly, at the coarse stage, a moment-based method is applied to estimating different orientations of each detected object. Then, according to the orientation and silhouette features of these objects, a line-based approach is used to roughly separate moving regions into objects and shadows. At the refined stage, the rough approximation is further refined through a Gaussian shadow modeling technique. The major difficulty in shadow elimination is the choice of a proper model that can reflect various appearances of shadows at different orientations and lighting. According to our analysis in [20], the best model is parameterized by three features including shadow illuminations, positions, and orientation. With these features, no matter how many shadows or what pedestrian-like shadows appear in the analyzed sequence, all the appearing shadows can be completely eliminated. In what follows, details of the proposed method are discussed.

The first thing to detect shadows is to find the orientation of a moving shadow region. Like Fig. 11, with the line determined by the orientation q_R and the gravity center P_G of the shadow object, the unwanted shadow can be roughly detected. The orientation of an object can be estimated from the properties of object moments. Given a two-dimensional function $f(x, y)$, the central moment of *f(x, y)* at order $(p+q)$ can be defined as:

$$(m_{p,q})_R = \sum_{(x,y)\in R} (x-\bar{x})^p (y-\bar{y})^q$$

where $(\bar{x},\bar{y}) = (\frac{1}{|R|}\sum_{(x,y)\in R} x, \frac{1}{|R|}\sum_{(x,y)\in R} y)$ and $|R|$ is the area of R. Then, the orientation q_R of R can be obtained by this equation:

$$q_R = arg \min_q \sum_{(x,y)\in R} \left[(x-\bar{x})\sin q - (y-\bar{y})\cos q \right]^2 . \quad (20)$$

Setting the term $\frac{1}{\partial q} \sum_{(x,y)\in R} \left[(x-\bar{x})\sin q - (y-\bar{y})\cos q \right]^2$ to zero, we obtain

$$q_R = \frac{1}{2} Tan^{-1}\left(\frac{2m_{1,1}}{m_{2,0} - m_{0,2}}\right). \quad (21)$$

Then, according to Eq.(21), the orientation of R can be obtained.

After calculating the orientation of the detected object, in what follows, we want to present a method to detect one boundary line for roughly cutting the shadows from the background. This paper assumes the shadows, wanting to be removed, touch the person at the person's feet. Then, a straight line can be used to roughly separate shadows from the detected object. As shown in Fig. 12(a), the shadow region R_2 expects to be cut from the object region R by the line $\overline{P_R Q_R}$. Since the line $\overline{P_R Q_R}$ is used to roughly cut a shadow into two different parts, the point P_R can be chosen as the maximum vertical difference between two adjacent points along the silhouette of the region R. The silhouette curve $C_R(x)$ of R can be obtained by tracking the vertical position of the first touched pixel of the region R when we track all the pixels along the xth column from top to bottom. Then, the coordinate (x_p, y_p) of the point P_R can be obtained as follows:

$$x_p = \arg\min_x |C_R(x) - C_R(x+1)| \text{ and } y_p = C_R(x_p). \quad (22)$$

Let q_R be denoted as the orientation of the object region R. $\overline{P_R Q_R}$ can be defined as the line that passes through P_R with the orientation q_R. Therefore, without deciding the point Q_R, the equation of $\overline{P_R Q_R}$ can be decided as

$$y = mx + c,$$

where $m = tan q_R$ and $c = y_p - x_p \tan q_R$.

With $\overline{P_R Q_R}$, the region R will be separated into different two parts, i.e., R_1 and R_2. Since the pixel intensities of a shadow region tend to be black, it will look more uniform and have a smaller intensity variance than its corresponding object region. Therefore, according to this observation, the shadow region R_2 can be easily identified by choosing one of the separated regions if it has a smaller variance. However, the line $\overline{P_R Q_R}$ cannot completely detect all shadow pixels from R since the boundary between R and the associated shadow is not a regular curve. The goal at this stage is to find a rough approximation of the shadow

region from R as a base for modeling shadows. Then, with the modeling process, all the unwanted shadows can be removed more accurately at the refined stage.

In order to completely eliminate the unwanted shadow of *R*, we need to build a proper shadow model to model the pixels in R_2. Due to different lighting sources, shadows can have different orientations. Clearly, if the orientation of the shadow is not considered in modeling, small pieces of non-shadow regions near shadow boundaries will be misclassified. In order to model shadows more accurately, it is better to map the original coordinates into elliptic coordinates as follows:

$$\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} \cos\theta_{R_2} & -\sin\theta_{R_2} \\ \sin\theta_{R_2} & \cos\theta_{R_2} \end{pmatrix} \begin{pmatrix} x - m_x \\ y - m_y \end{pmatrix}, \tag{23}$$

where m_x and m_y are the means of the x and y coordinates of pixels in R_2, respectively, and θ_{R_2} is the major orientation of R_2. Through the transformation, the new Gaussian model uses the center of R_2 as the new origin and the major axis of R_2 as the new x axis. Since the Gaussian model has three parameters affecting the results of shadow modeling, different weights should be added to each parameter for improving the model validity and elasticity. Then, the suggested Gaussian object model is defined as follow:

$$G_{suggest}(s,t,I(s,t)) = e^{-\left(\frac{w_s s^2}{\sigma_s^2} + \frac{w_t t^2}{\sigma_t^2} + \frac{w_I (I(s,t)-m_I)^2}{\sigma_I^2}\right)}, \tag{24}$$

where w_s is the weight for the s coordinate, w_t the weight for the t coordinate, w_I the weight for the intensity component, σ_s the variance of s, and σ_t is the variance of t. Here, different weights to each parameter are used for improving the validity and elasticity of this model. In this model, w_s, w_t, and w_I are set to 0.2, 0.3, and 0.5, respectively. In what follows, details of the complete procedure for removing unwanted shadows are described:

Shadow Elimination Algorithm

Step 1: Apply the level set technique to extract a set of moving objects $\{R_k\}_{k=1,\ldots,M}$.

Step 2: For each object R_k, repeat the following steps:

2.1: Use Eq.(21) to calculate the orientation θ_{R_k} of the object R_k.

2.2: Find the separating line $\overline{P_{R_k} Q_{R_k}}$ based on Eq.(22).

2.3: With the help of $\overline{P_{R_k} Q_{R_k}}$, separate the object R_k into two regions $R_{k,1}$

and $R_{k,2}$. Then, the region with a smaller variance is chosen as the shadow region $R_{k,2}$.

2.4: Obtain the Gaussian shadow model G_{R_k} from $R_{k,2}$ based on Eq.(24).

2.5: Eliminate each pixel *(x, y)* in R_k if the pixel satisfies the following rule:

$$G_{R_k}(s,t,I(s,t)) > T_s,$$

where (s,t) is obtained from Eq.(23) and T_s is set to 0.8.

4 Experimental Results

In order to demonstrate the superiority of the proposed scheme, three sets of experiments were used. All the used sequences were obtained by a general video camera and with the same image size 320×240. For the first experiment, a static camera was used to capture a series of image frames for examining the effectiveness of the proposed object detection scheme. As to the second experiment, a moving camera was used for testing the proposed camera compensation method. For the last experiment, a series of images were used for evaluating the performances of our proposed shadow elimination algorithm.

For the first set of experiments, Fig. 13 shows the results when the background is static and only a single object appears in the monitored environment. Fig. 14 shows the case when the analyzed sequence includes multiple pedestrians. Fig. 15 shows another case when multiple pedestrians appear and connect together. Clearly, no matter what kinds of cases are handled, the proposed method works well to track all desired moving pedestrians.

On the other hand, the observed objects often have some interactions like talking or saying hello to each other. In this paper, we use an object relation table to analyze such interactions. Fig. 16 shows the case when two moving pedestrians move closely. It seems they have some conversations. Thus, through building this object relation table, different tracking statuses can be clearly identified and analyzed even though the objects merged and split several times. Fig. 17 shows another case when two groups of pedestrians appear.

The next experiment is to deal with the problem when the background is not static. In Fig. 18, (a) is the original sequence captured by a moving camera. Through the proposed camera compensation, all the images in (a) can be compensated for keeping the background static (as shown in (b)). Then, according to the level set algorithm, the moving pedestrians can be successfully extracted from (a) and shown in (c).

The third set of experiments is used to show the performances of our shadow elimination algorithm. Fig. 19 shows the case when one pedestrian appears in the analyzed sequence.

(b) is the extracted moving object from (a). (c) is the result obtained using the suggested model (Eq.(23)) with weights $w_s = 0.2$, $w_t = 0.3$, and w_I=0.5. On the other hand, the proposed method can also be used to eliminate multiple shadows. Fig. 20 shows two pedestrians walking along a road. (b) is the detected object from (a). (c) is the result of shadow elimination with the suggested model. Fig. 21 shows the whole sequence of pedestrian tracking and shadow elimination. The pink regions mean the detected shadows. Clearly, whatever the surveillance environment is, all unwanted shadows can be completely removed with the proposed method. From these experiments, the superiority of the proposed method can be clearly verified.

5 Conclusions

In this paper, we have presented a novel approach for tracking multiple objects using the level set technique. The framework first uses the camera compensation technique and the background subtraction to obtain desired object motions. Then, through defining a proper speed function, all desired object boundaries can be efficiently and effectively detected by using a level-set method. Furthermore, an object relation table is created to analyze different behaviors of detected objects. In addition to the tracking technique, this paper also proposed a shadow elimination method for eliminating unwanted pedestrian-like shadows from the analyzed sequence. The contributions of this paper can be summarized as follows:

(a) A camera compensation technique was proposed for compensating the background. Then, different object motions can be well extracted through a subtraction technique.

(b) A proper speed function was proposed for tracking desired moving object. Thus, different objects can be quickly detected and tracked through curve evolutions.

(c) When analyzing the moving objects, an object relation table was proposed for efficiently analyzing object's behaviors.

(d) A shadow elimination method was used for removing unwanted shadows using Gaussian shadow modeling. Through properly parameterizing this model, all unexpected shadows can be completely detected and eliminated from the analyzed scenes.

Experimental results have shown the proposed method work very well to track all the pedestrian-like objects no matter how many moving objects and shadows appear in the analyzed sequences.

References

[1] A. Tesei, G.L. Foresti, and C.S. Regazzoni, “Human body modeling for people localization and tracking from real image sequences,” *Proc. Fifth Int’l Conf. Image Processing and its Applications*, pp. 806–809, 1995.

[2] J. Segen, “A camera-based system for tracking people in real time,” *Proc. Int’l Conf. Pattern Recognition,* Vienna, pp. 63–67, 1996.

[3] M. Rossi, and A. Bozzoli, “Tracking and counting moving people,” *Proc. Int’l Conf. Image Processing,* Austin, Texas, pp. 212–216, 1994.

[4] B. Heisele, U. Kressel, and W. Ritter, “Tracking non-rigid, moving objects based on color cluster flow,” *Proc. Conf. Computer Vision and Pattern Recognition,* San Juan, Puerto Rico, pp. 257–260, 1997.

[5] S. Osher and J. A. Sethian, “Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulation,” *Journal of Computer Physical*, vol. 79, pp. 12-49, 1988.

[6] R. Malladi, J. A. Sethian, and B. C. Vemuri, “Shape modeling with front propagation: a level set approach”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158-175, Feb. 1995.

[7] R. Malladi and J. A. Sethian, “Level set and fast marching methods in image processing and computer vision,” *J. Vac. Sci. Tech. B*, vol.13, no: 4, July/Aug. 1995.

[8] R. Malladi and J. A. Sethian, “A real-time algorithm for medical shape recovery,” *International Conference on Computer Vision*, pp.304–310, 1998.

[9] J. A. Sethian, “Curvature and the evolution of fronts,” *Comm. Math. Phys.*, vol. 101, pp.487-499, 1985.

[10] J. A. Sethian, “A review of the theory, algorithms, and applications of level set methods for Propagating Interfaces,” *Acta Numerica*, 1996.

[11] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int’l J. of Computer Vision*, vol. 1, pp. 321-332, 1988.

[12] S. D Dagras *et al.*, “Models for motion-based video indexing and retrieval,” *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 88-101, 2000.

[13] S. Belongie, C. Carson, H. Greenspan, and J. Malick, “Color and texture based image segmentation using EM and its application to content-based image retrieval,” *IEEE International Conf. Computer Vision*, pp.675-682, 1998.

[14] C. M. Kuo, C.H. Hsieh, H.C. Lin, and P.C. Lu, “Motion estimation algorithm with

kalman filter," *IEEE Trans. Electronics Letter*, vol. 30, no. 15, pp. 1204–1206, 1994.

[15] J. Lu and M.L. Liou, "A simple and efficient search algorithm for block-matching motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 429–433, 1997.

[16] T. Oliver and K. Renaud, "Variational principles, surface evolution, PDE's, level set methods, and the stereo problem," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 336–344, 1998.

[17] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of M\moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, no. 3, pp.266–280, March 2000.

[18] B. C. Vemuri, J. Ye, Y. Chen, and C.M. Leonard, "A level-set based approach to image registration," *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 86–93, 2000.

[19] R. T. Whitaker, "A level-set approach to image blending," *IEEE Transactions on Image Processing*, vol. 9, no.11, pp.1849–1861, Nov. 2000.

[20] J. W. Hsieh, W-F. Hu, C-J. Chang, and Yung-Sheng Chen, Shadow Elimination for Effective Moving Object Detection by Gaussian Shadow Modeling, to appear in *Image Vision and Computing Journal*.

[21] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, London, U. K.: Chapman & Hall, 1993.

TABLES

Current / Previous	C	D	SOR
A	0	0	0
B	1	0	1
SOC	1	0	

Table 1 presents the relationships of moving objects in Fig. 8.

Current / Previous	B	SOR
A	1	1
C	1	1
SOC	2	

Table. 2 presents the relationships of objects in Fig. 9.

Current / Previous	A	C	SOR
B	1	1	2
SOC	1	1	

Table 3 presents the relationships of objects in Fig. 10.

FIGURES

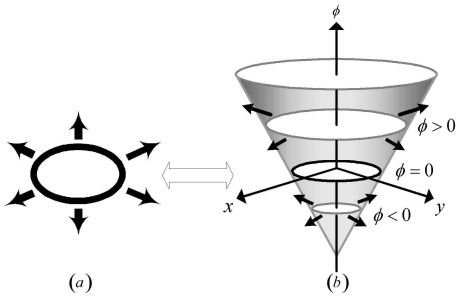


Fig.1 (a) Object boundary (b) Relations between the level set function and the object boundary.

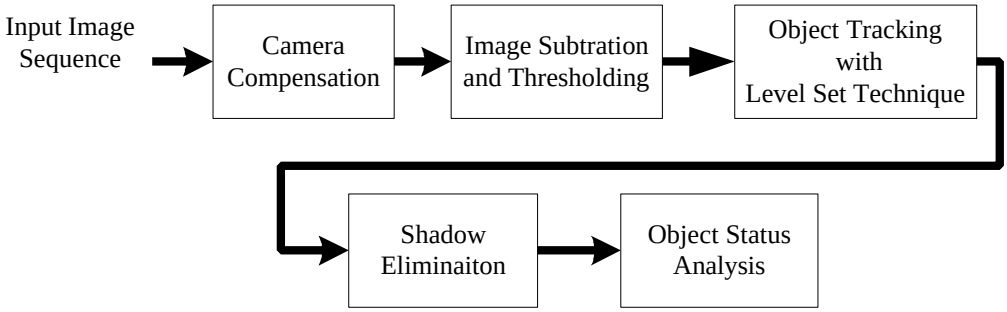
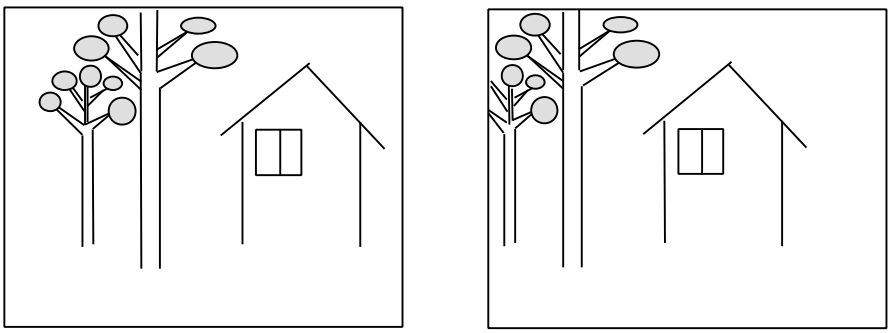


Fig. 2 Flowchart of the proposed system.

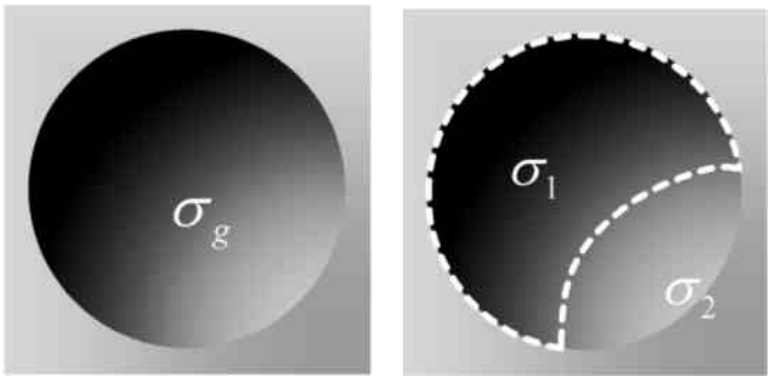


(a) (b)

Fig. 3 Edge results of two images.

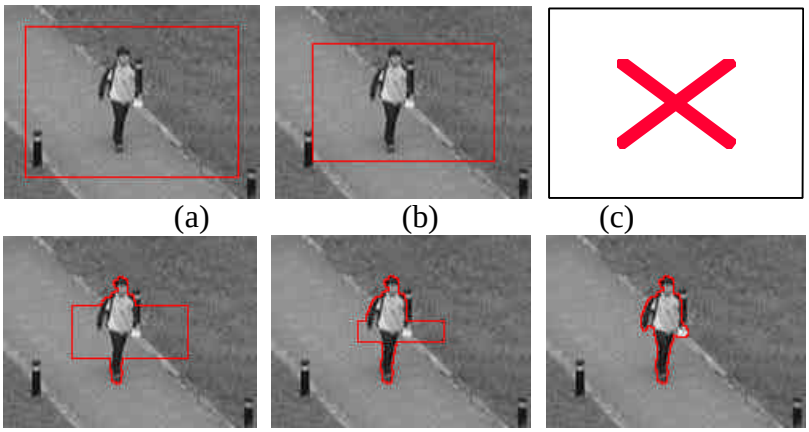
(a) (b) (c)

Fig. 4 Result of camera compensation. (a) Original background. (b) The current video frame. (c) Background after camera compensation.



(a) (b)

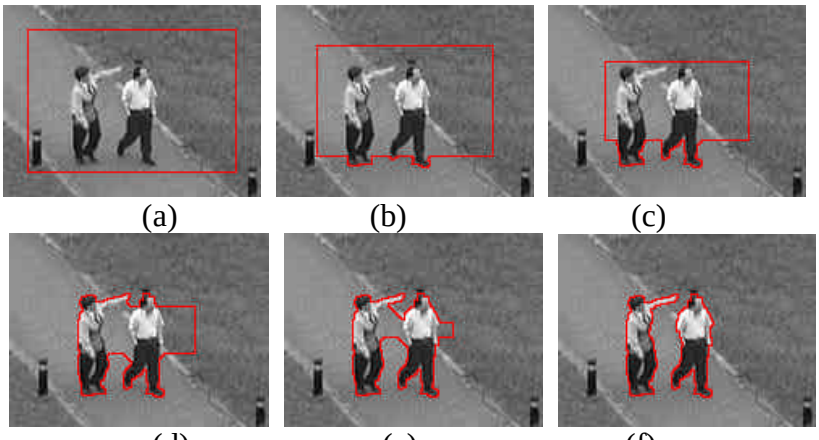
Fig. 5 (a) Source image. (b) Result after thresholding.



(a) (b) (c)

(d) (e) (f)

Fig. 6 Tracking results when the sequence has only one single object. (a) Initial frame. (b)-(d) Evolutions of the tracking curve.



(a) (b) (c)

(d) (e) (f)

Fig. 7 Tracking results when the sequence has two persons. (a) Initial frame. (b)-(d) Evolutions of the tracking curve.

Fig. 8 Statuses of moving objects between two frames

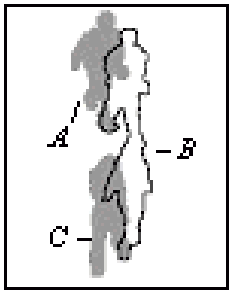


Fig. 9 Moving objects *A* and *C* in the previous frame have occlusion in the current frame.

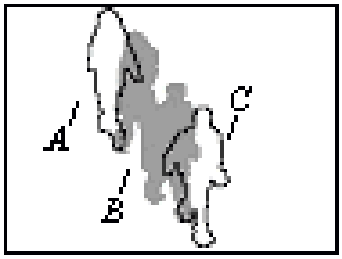


Fig. 10 Moving object *B* in the previous frame is split into two different object *A* and *C*.

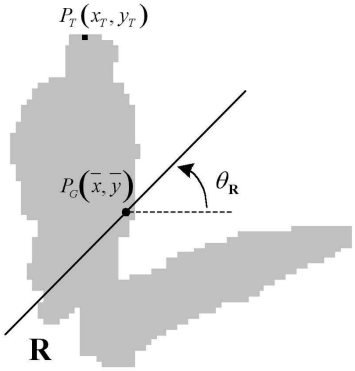
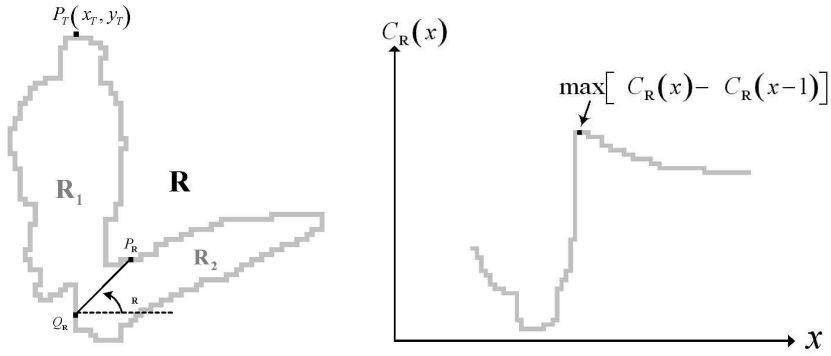
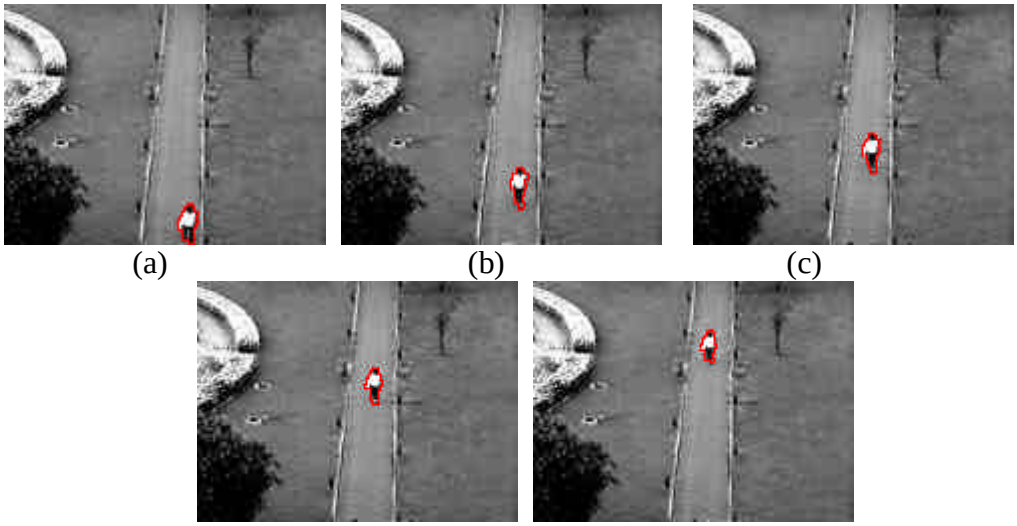


Fig. 11 Object ***R*** and it's gravity center and orientation q_R



(a) (b)

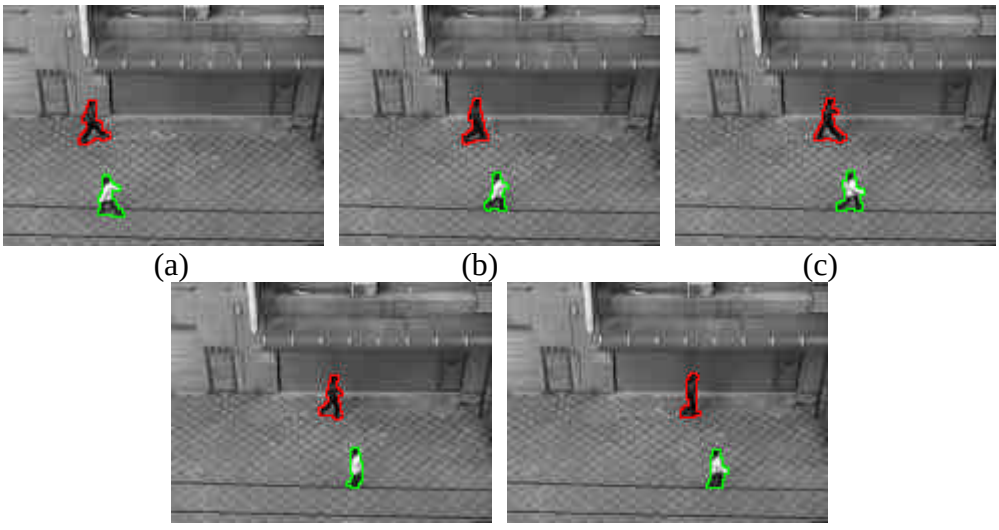
Fig. 12 Object and it's contour information.



(a) (b) (c)

(d) (e)

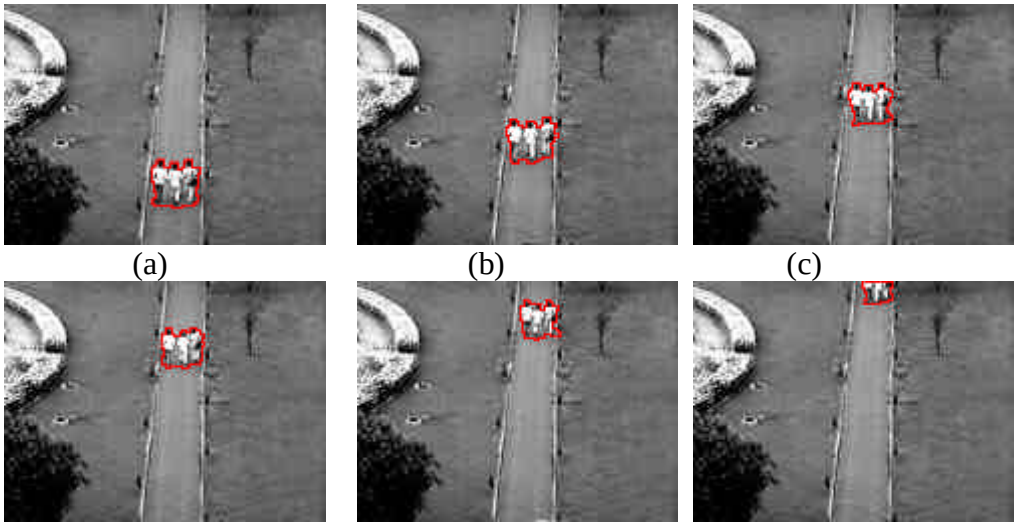
Fig. 13 The tracking results of the sequence with a single moving pedestrian.



(a) (b) (c)

(d) (e)

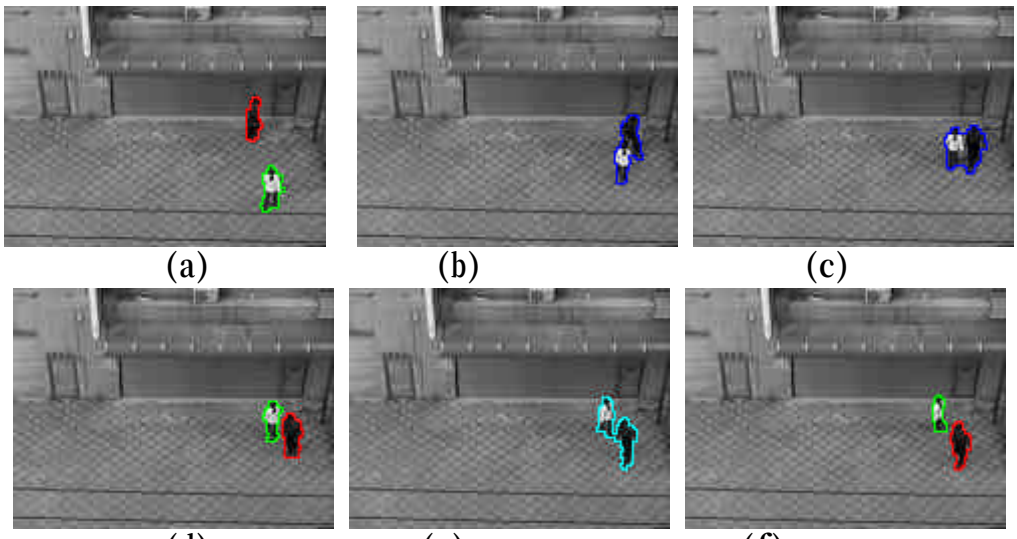
Fig. 14 The tracking results of a sequence including several moving pedestrians.



(a) (b) (c)

(d) (e) (f)

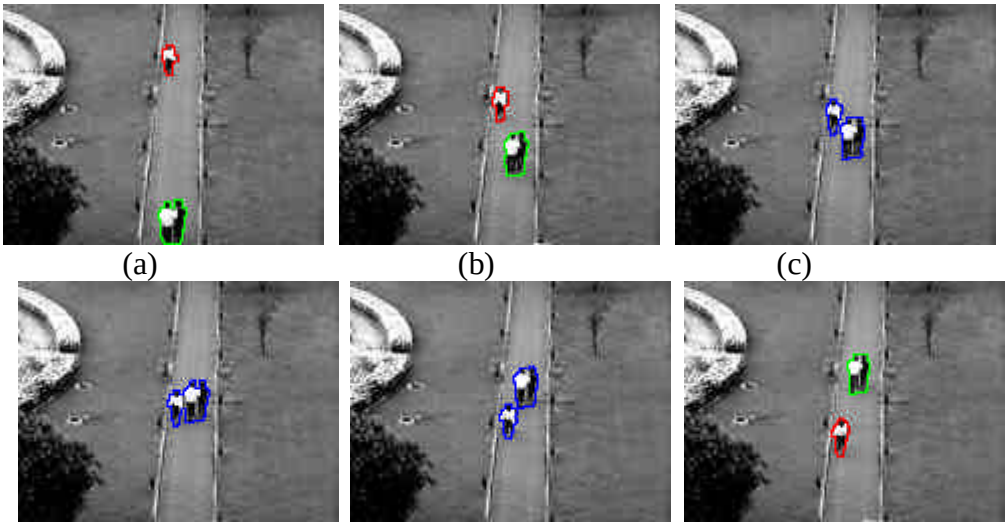
Fig. 15 The tracking results of the sequence when multiple moving pedestrians appear and connect together.



(a) (b) (c)

(d) (e) (f)

Fig. 16 The tracking results of the sequence when the moving pedestrians have interactions. (a), (d), and (f) show the cases when two objects separate. (b), (c), and (e) show the cases when the observed objects merge together. It seems these two objects have conversations.



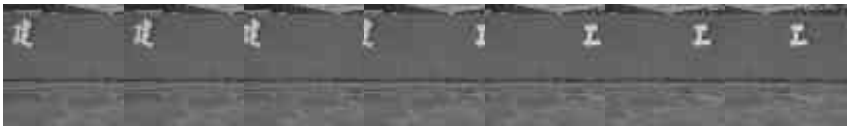
(a) (b) (c)

(d) (e) (f)

Fig. 17 The tracking results of the sequence when groups of pedestrians appear into and have some interactions. (a), (b), and (f) show the cases when two objects separate. (c), (d) and (e) show the cases when two objects merge together.

(i) (ii) (iii) (iv) (v) (vi) (vii)

(a) Original sequence



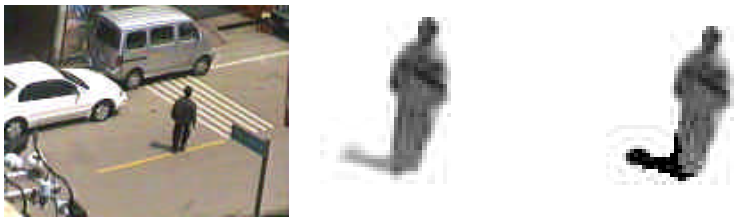
(i) (ii) (iii) (iv) (v) (vi) (vii)

(b) Background after compensation

(i) (ii) (iii) (iv) (v) (vi) (vii)

(c) Results of object tracking

Fig. 18 The tracking results of the sequence when the background is not static. (a) Original sequence. (b) Results of (a) after camera compensation. (d) Final results of tracking by using the level set method.



(a) (b) (c)

Fig. 19 Results of single shadow elimination when the shadow orientation is left. (a) Original image. (b) Extracted moving object from (a). (c) Shadow elimination with the suggested model.

(a) (b) (c)

Fig. 20 Result of shadow elimination when two moving pedestrians appear in the video sequence. (a) Original image. (b) Extracted moving object. (c) Result of shadow elimination with the suggested model.

(a) (b) (c)

(d) (e) (f)

Fig. 21 Results of a video sequence after object tracking and pedestrian shadow elimination. The regions with pink color mean the detected shadows.