Extraction of Moving Objects From Their Background Based on Multiple Adaptive Thresholds and Boundary Evaluation

Lu Wang and Nelson H. C. Yung, *Senior Member, IEEE*

***Abstract*—The extraction of moving objects from their background is a challenging task in visual surveillance. As a single threshold often fails to resolve ambiguities and correctly segment the object, in this paper, we propose a new method that uses three thresholds to accurately classify pixels as foreground or background. These thresholds are adaptively determined by considering the distributions of differences between the input and background images and are used to generate three boundary sets. These boundary sets are then merged to produce a final boundary set that represents the boundaries of the moving objects. The merging step proceeds by first identifying boundary segment pairs that are significantly inconsistent. Then, for each inconsistent boundary segment pair, its associated curvature, edge response, and shadow index are used as criteria to evaluate the probable location of the true boundary. The resulting boundary is finally refined by estimating the width of the halo-like boundary and referring to the foreground edge map. Experimental results show that the proposed method consistently performs well under different illumination conditions, including indoor, outdoor, moderate, sunny, rainy, and dim cases. By comparing with a ground truth in each case, both the classification error rate and the displacement error indicate an accurate detection, which show substantial improvement in comparison with other existing methods.**

***Index Terms*—Boundary evaluation, change detection, curvature, edge, foreground extraction, thresholds.**

I. Introduction

NATURAL IMAGE sequences often contain one or more moving objects performing a series of actions in front of a background scene that is almost stationary. It could be a human walking across the camera's field of view, a group of players in a football game, or vehicles traveling along a highway. In terms of computer vision, a moving object is a set of foreground pixels that is not part of the static background pixel set and changes its position between frames. If we can accurately extract these moving objects from their background, the subsequent object recognition or tracking would be enormously simplified. The way in which these moving objects are extracted is commonly known as foreground extraction in applications such as visual surveillance [1] and intelligent user interfaces [2]. Conceptually, assuming that the background is stationary, if we have a copy of the background, then the foreground can be determined by taking the difference between the background image and the input image. Unfortunately, the problem is not so straightforward in practice.

Conventionally, the foreground extraction problem is dealt with by change detection techniques, which can be pixel based or region based. Simple differencing is the most intuitive by arguing that a change at a pixel location occurs when the intensity difference of the corresponding pixels in two images exceeds a certain threshold. However, it is sensitive to pixel variation resulting from noise and illumination changes, which frequently occur in complex natural environments. More robust methods [2]–[4] handle noise and lighting change issues by maintaining an adaptive statistical background model. Recently, Tsai and Lai [5] have proposed using independent component analysis to deal with illumination changes without background model updating.

On the other hand, region-based change detection methods take advantage of interpixel relations, measuring the region characteristics of an image pair at the same pixel location. For instance, the likelihood ratio test [6] uses a hypothesis test to decide whether statistics of two corresponding regions come from the same intensity distribution. Although this method is more immune to noise, it is still fairly sensitive to illumination changes. The shading model (SM) [7] exploits the ratio of intensities in the corresponding regions of two images to cope with illumination changes. Liu *et al.* [8] suggested a change-detection scheme that compares circular shift moments (CSMs), which represent the reflectance component of the image intensity, regardless of illumination. However, both the SM and CSM methods poorly perform over dark regions, as the denominator of the ratio becomes insignificant. Local structural features that are less affected by illumination changes have also been employed to represent the difference of two corresponding regions. Li and Leung [9] proposed an algorithm that combines intensity and texture differences, which is based on the argument that texture is less sensitive to illumination changes, whereas intensity is more representative of homogeneous regions. Unfortunately, an exception occurs under weak illumination when intensity is strongly affected by noise and texture is poorly defined. Instead, Heikkilä and Pietikäinen [10] modeled each pixel as a group of adaptive local binary pattern histograms that are calculated over a circular region around the pixel. It is tolerant against illumination changes and

Manuscript received January 23, 2008; revised November 17, 2008, March 8, 2009, and May 19, 2009. First published July 21, 2009; current version published March 3, 2010. The Associate Editor for this paper was S. Nedevschi.

The authors are with the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong (e-mail: wanglu@eee.hku.hk; nyung@eee.hku.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2009.2026674

simple to compute. The drawback is that it introduces many parameters that need to be empirically set, making it difficult to correctly apply in real scenarios. Although region-based methods are generally more robust than pixel-based methods, one common problem with them is that the object's boundary becomes inaccurate because the pixel in question may not have changed, whereas its neighborhood may have.

Whether the method is pixel based or region based, thresholding of the difference image always presents itself as most challenging. In many cases, the threshold is selected empirically or by trial and error. Obviously, a threshold chosen in this way is ineffective for images with significantly different distributions. As a result, several adaptive threshold selection methods have been proposed. Some of these methods are based on histograms. For example, Otsu's method [11] calculates the best threshold by minimizing the ratio of intraclass and interclass variations, the isodata algorithm [12] searches for the best threshold by an iterative estimation of the mean values of the foreground and background pixels, the triangle algorithm [13] particularly deals with unimodal histograms, Kita [14] analyzes the characteristics of the ridges of clusters on the joint histogram, and Sen and Pal [15] select the threshold by using the fuzzy and rough set theories. Another set of approaches is to assume that the distributions of the changes and the noise of the difference image are Gaussian or Laplacian. For example, Bruzzone and Prieto [16] modeled the difference image as a mixture of two Gaussian distributions, representing changed and unchanged pixels. The means and variances of the class-conditional distributions are then estimated using an expectation–maximization algorithm. Rosin and Ellis [17] exploited the simple statistics of the median and the median absolute deviation by assuming that less than half the image is in motion. Kapur *et al.* [18] selected thresholds by virtue of the entropy of the image. Gray [19] considered the Euler number, and O'Gorman [20] used image connectivity.

The major shortcoming of the single-threshold (ST) approach is that it often ends up with a foreground that is either oversegmented or undersegmented. This comes as no surprise because the foreground and background pixels intertwine in the measurement space, which makes it impossible to have a global threshold that segments well. To alleviate the problem, it appears logical to instead consider multiple thresholds. Rosin and Ellis [17] perform thresholding with hysteresis, where the difference image is first thresholded by two levels, and regions in the intermediate range are not considered to be changed unless they are connected with regions generated by the higher threshold. This method scratches the surface of the problem and stops short of tackling the real issue. Zhou and Hoang [21] proposed a bithreshold method in which pixels in the intermediate range of difference are further evaluated to decide whether they are shadow pixels or not. Reference [22] is similar to [21], except that a silhouette extraction technique is applied on the foreground extraction result to smooth the boundary.

From a classification point of view, applying P thresholds results in $P+1$ classes. For $P=2$, we have background, foreground, and ambiguous pixel classes. These ambiguous pixels belong to either the foreground or the background and need to be further classified. In this paper, we adopt this line of approach. In principle, we first select a low threshold τ_L, a middle threshold τ_M, and a high threshold τ_H by separately applying the triangle algorithm on the (texture, luminance, and chrominance) difference distributions, from which the boundaries of three masks B_L, B_M, and B_H, corresponding to τ_L, τ_M, and τ_H, respectively, are obtained. Then, corresponding points from B_L and B_M are evaluated to see if they are consistent or not. The connected inconsistent points constitute the inconsistent boundary segments (IBSs). For each IBS pair, the segment characterized by being unassociated with a shadow region and having a lower curvature and a larger edge response is chosen as the resulting boundary. The resultant intermediate boundary set B_I (from merging B_L and B_M) and B_H are subjected to the same IBS identification and evaluation as described above to give a representative boundary set B_R. Next, from the fact that real holes always result in similar shapes in the three masks, whereas false holes do not, the IBS identification result is used to verify whether a hole is real or not. Finally, B_R is refined to give a final boundary set that encloses the extracted objects. Experimental results show that the proposed method consistently performs well under different illumination conditions, including indoor, outdoor, normal, sunny, rainy, and dim cases. By referring to a ground truth in each case, the classification error rate, which is 6.8%, indicates an accurate detection, which is a substantial improvement over other existing methods.

II. Proposed Method

A. Overview

The proposed method aims at extracting the moving objects in an input image from their background, where the background image is estimated by median filtering along the temporal direction. As depicted in Fig. 1, the proposed method consists of five steps: 1) difference image calculation; 2) thresholds selection; 3) IBS identification and evaluation; 4) verification; and 5) boundary refinement. Details of these steps are described in the following sections.

B. Difference Image Calculation

In this paper, three differences, namely, texture (ΔT), luminance (ΔY), and chrominance (ΔC), between the input image and the background, as proposed in [23], are calculated. The texture T of an image is measured by the autocorrelation of each square image block of size $(2N+1)\times(2N+1)$ in the intensity space, and ΔT is calculated as the square difference of the texture of the two corresponding blocks. ΔT is insensitive to illumination changes as it measures the local structure difference. ΔY is given by the difference of the Y channel, while ΔC is determined by the sum of the square differences in the Cb and Cr channels of the YCbCr color space. ΔY and ΔC are complements to ΔT, and they are effective in detecting untextured inner regions of the moving objects where color or brightness significantly changes relative to the background. For outdoor scenes, all three differences are considered, whereas for indoor scenes, as the light source(s) may be flickering, only ΔT is considered, as it represents the most stable feature that is insensitive to illumination changes.

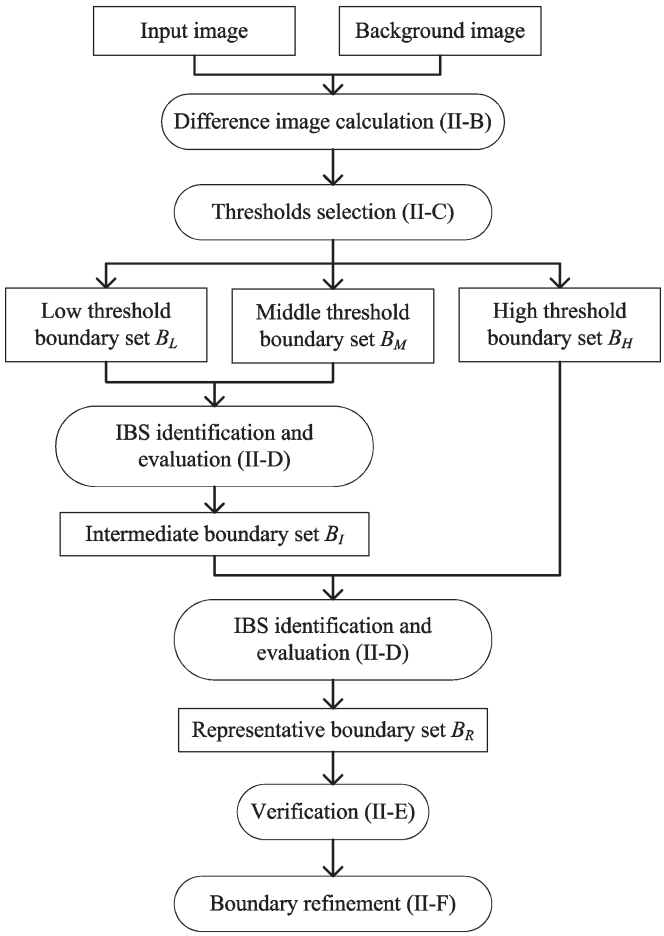


Fig. 1. Overview of the proposed multiple-threshold object extraction.

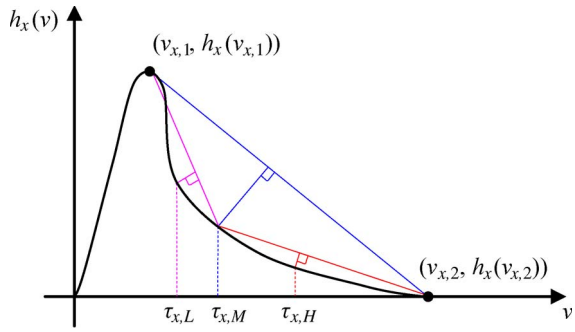


Fig. 2. Thresholds selection using the triangle algorithm.

C. Thresholds Selection

As a general requirement, the selection of thresholds should be systematic and adaptive to a variety of images. In this paper, we propose to use three thresholds for each image difference. Specifically, the low threshold should enable the detection of nearly all the foreground pixels, even if it results in false detection of the background pixels, the high threshold should enable the exclusion of nearly all the background pixels, even if it results in leaving some foreground pixels undetected, and the middle threshold should be approximately optimal in terms of the error rate. The following paragraphs describe how these thresholds are determined.

Since all the distributions of the three differences are unimodal, the triangle algorithm is the most suitable to apply [24]. The original triangle algorithm works as follows. Given a histogram function $h(v)$, a line is constructed between the peak of the histogram $h(v)$ (at $v = v1$) and the largest nonzero value of v (at $v = v2$). Then, the perpendicular distance between this line and the histogram is evaluated. The value of v that corresponds to the maximum distance is taken as the threshold value τ.

To select the needed three thresholds for each difference image, we propose the following method. Given the histogram function $h_x(v)$ of the difference image Δx (for $x = T, Y,$ or C), the triangle algorithm is used to select the middle threshold $\tau_{x,M}$. Then, the low threshold $\tau_{x,L}$ is estimated by applying the triangle algorithm over $[v_{x,1}, \tau_{x,M}]$ of the histogram, whereas the high thresholds $\tau_{x,H}$ for ΔY and ΔC are estimated by applying the triangle algorithm over $[\tau_{x,M}, v_{x,2}]$ of the histogram, as shown in Fig. 2. For $\tau_{T,H}$, we propose a different way to select it. As ΔT is calculated over a $(2N+1) \times (2N+1)$ image block, the thresholded ΔT using $\tau_{T,M}$, i.e., $Mask_{T,M}$, typically includes background pixels outside the boundaries, which, according to our requirement for the high threshold, should be removed as much as possible. To achieve this, $Mask_{T,M}$ is eroded by a disk-shaped structuring element of radius N, from which the pixels that are eroded off form one class (P_1), and the remaining pixels form another class (P_2). Pixels in P_1 are likely to be false positives, whereas pixels in P_2 are likely to be true positives. If we call the modes (i.e., the value that occur the most frequently) of the distributions of P_1 and P_2 as c_1 and c_2, respectively, then $\tau_{T,H}$ would lie between c_1 and c_2. Thus, $\tau_{T,H}$ is calculated by

$$\tau_{T,H} = \frac{c_1 N_2 + c_2 N_1}{N_1 + N_2} \tag{1}$$

where N_1 is the number of pixels in P_1, and N_2 is the number of pixels in P_2. Given the thresholds $(\tau_{x,L}, \tau_{x,M}, \tau_{x,H})$ for $x = T, Y,$ or C, the thresholded masks $Mask_L$, $Mask_M$, and $Mask_H$ are calculated by

$$Mask_j(x,y) = \begin{cases} 1, & \text{if } \Delta T(x,y) > \tau_{T,j} \vee \Delta Y(x,y) > \tau_{Y,j} \\ & \quad \vee \Delta C(x,y) > \tau_{C,j} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $j = L, M, H$, "1" represents the foreground, and "0" represents the background.

It is noted that the area around the boundaries between the shadowed and unshadowed regions still survive in $Mask_H$, because although the texture difference does not respond to the inner area of moving shadows, it has a strong response to shadow boundaries. Therefore, a morphological opening operation of radius N is applied to $Mask_H$ to ensure that shadow boundaries are removed as well. Then, from $Mask_L$, $Mask_M$, and $Mask_H$, respective boundaries B_L, B_M, and B_H are calculated, including both the exterior boundaries of objects and the interior boundaries of holes. Fig. 3 depicts an example of the input image I, the estimated background image B, $Mask_L$, $Mask_M$, $Mask_H$ (after the opening operation),

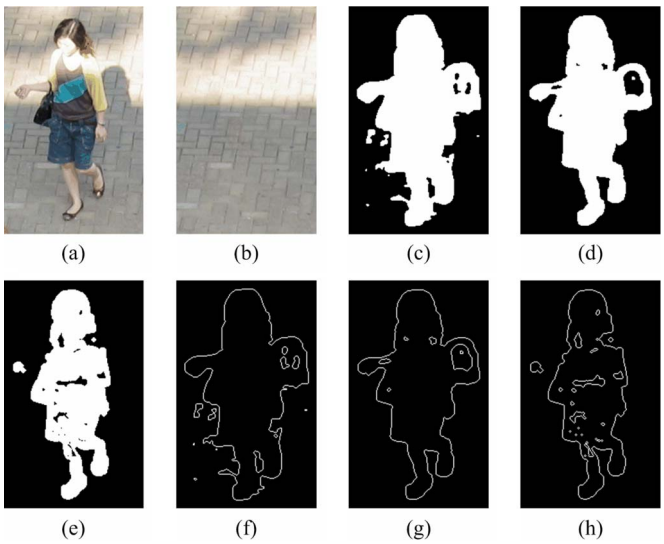


Fig. 3. Example image and the associated masks and boundaries. (a) I. (b) B. (c) $Mask_L$. (d) $Mask_M$. (e) $Mask_H$. (f) B_L. (g) B_M. (h) B_H.

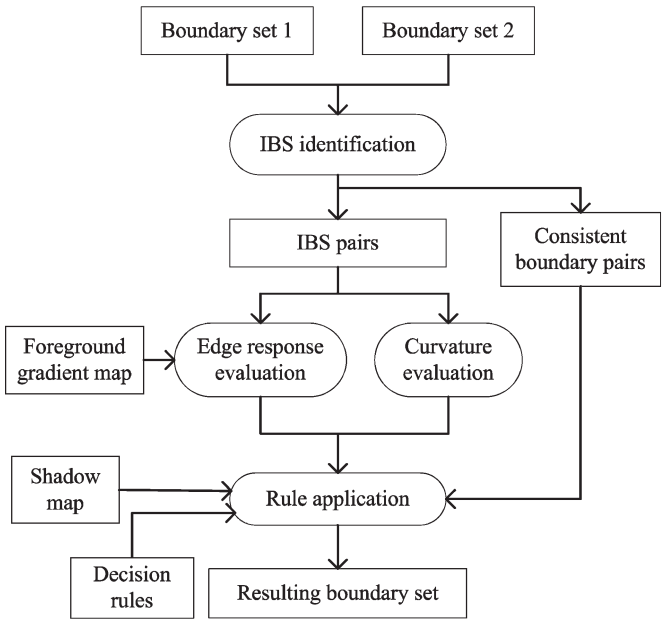


Fig. 4. IBS identification and evaluation.

B_L, B_M, and B_H. Note that B_L, B_M, and B_H may appear to be slightly larger than the original object(s), which is caused by the region-based texture difference. This problem will later be dealt with in the boundary refinement step (see Section II-F).

D. IBS Identification and Evaluation

Fig. 4 depicts the IBS identification and evaluation procedure. Let the two boundary sets involved in the merging process be known as B_{low} (corresponding to the lower threshold) and B_{high} (corresponding to the higher threshold) and their corresponding masks be $Mask_{\text{low}}$ and $Mask_{\text{high}}$. For the first merging step, we have $B_{\text{low}} = B_L$, $B_{\text{high}} = B_M$, $Mask_{\text{low}} = Mask_L$, and $Mask_{\text{high}} = Mask_M$, and for the second merging step, we have $B_{\text{low}} = B_I$ (result of merging B_L and B_M), $B_{\text{high}} = B_H$, $Mask_{\text{low}} = Mask_I$, and $Mask_{\text{high}} = Mask_H$.

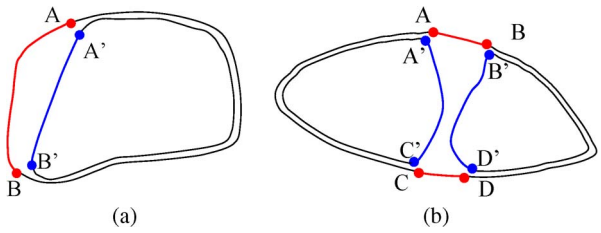


Fig. 5. Two examples of IBS pairs (red and blue lines). (a) B_{high} consists of one boundary. (b) B_{high} consists of two boundaries.

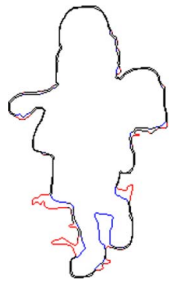


Fig. 6. IBS pairs (red and blue lines) from a boundary pair in Fig. 3(a).

If B_{low} consists of multiple closed boundaries, each boundary is separately evaluated. It is possible that an exterior boundary of B_{low} does not have a correspondent in B_{high} (i.e., all the difference values inside this boundary are lower than the higher threshold) or that an interior boundary of B_{high} does not have a correspondent in B_{low} (i.e., all the difference values inside the boundary are higher than the lower threshold). In these two cases, we do not have the presumed two boundaries to compare; thus, the boundary concerned is simply discarded.

1) IBS Pair Identification: This step identifies boundary points that produce a significant inconsistency between B_{low} and B_{high}. Let the ith boundary in B_{low} be B^i_{low} and its correspondent be B^i_{high}. Two points, i.e., $p^{i,k}_{\text{low}} = (x^{i,k}_{\text{low}}, y^{i,k}_{\text{low}})$ and $p^{i,l}_{\text{high}} = (x^{i,l}_{\text{high}}, y^{i,l}_{\text{high}})$, from B^i_{low} and B^i_{high}, respectively, are consistent (i.e., they come from the same object boundary point) only if 1) they have similar orientations, and the line connecting them has the direction similar to their normal vectors, i.e., $v(p^{i,k}_{\text{low}}) \cdot v(p^{i,l}_{\text{high}}) \approx 1$ and $v(p^{i,k}_{\text{low}}) \cdot (p^{i,k}_{\text{low}} - p^{i,l}_{\text{high}})/\|p^{i,k}_{\text{low}} - p^{i,l}_{\text{high}}\| \approx 1$, where $v(\cdot)$ represents the unit normal vector of a boundary point, and $\|\cdot\|$ denotes the Euclidean norm, and 2) their chessboard distance is smaller than $N+1$, i.e., $\max(|x^{i,k}_{\text{low}} - x^{i,l}_{\text{high}}|, |y^{i,k}_{\text{low}} - y^{i,l}_{\text{high}}|) \leq N$, because the deviation of a boundary point from the original object boundary point (caused by the texture difference) should not be larger than N.

Once the consistent boundary points and their correspondents are found, the remaining boundary points are considered to be inconsistent, as illustrated in red and blue in Figs. 5 and 6. The connected inconsistent points then form the IBS pairs [e.g., AB and A′B′ in Fig. 5(a)].

Note that the inconsistency may involve multiple segments [e.g., AB and CD versus A′C′ and B′D′ in Fig. 5(b)]. In such a case, we take the segments from B^i_{low} (e.g., AB and CD) as one segment and the segments from B^i_{high} (e.g., A′C′ and B′D′) as the correspondent.

2) Boundary Evaluation: Given the IBS pairs, the two segments in each IBS pair are evaluated with respect to the edge response, curvature, and shadow to identify which one of the two is more likely to represent the true boundary. This is based on three assumptions: 1) The true boundary segment is associated with a large edge response; 2) the objects' shapes are usually smooth; and 3) long and convoluted segments are unlikely to be a true boundary. The following sections describe how each IBS pair is evaluated according to these assumptions.

a) Evaluation of IBS edge response: To evaluate the IBS edge response, we propose a method for extracting the foreground edge map by performing edge subtraction between the input and background images. First, the edge map EM_I of the input image I is obtained by the Canny edge detector, and the gradient maps $\nabla I = (\nabla_x I, \nabla_y I)$ and $\nabla B = (\nabla_x B, \nabla_y B)$ of I and B are calculated as well. Then, the normalized gradient maps GM_I and GM_B of I and B are computed as

$$GM_a(x,y) = \begin{cases} \left(\frac{\nabla_x a(x,y)}{\|\nabla a(x,y)\|}, \frac{\nabla_y a(x,y)}{\|\nabla a(x,y)\|}\right) & \text{if } \|\nabla a(x,y)\| \neq 0 \\ (0,0) & \text{otherwise} \end{cases} \tag{3}$$

where $a = I, B$, and

$$\|\nabla a(x,y)\| = \sqrt{(\nabla_x a(x,y))^2 + (\nabla_y a(x,y))^2}.$$

Next, the background edges are subtracted from the input edges. Intuitively, the edge points caused by the moving objects would significantly be different from its corresponding background. To capture this characteristic, we define the following gradient difference:

$$f_{g\_diff}(x,y) = |\|\nabla I(x,y)\| - \|\nabla B(x,y)\|| - GM_I(x,y) \cdot GM_B(x,y) \tag{4}$$

for the nonzero EM_I pixels. In (4), the magnitude and direction differences of gradients are combined to improve the ability of f_{g_diff} in discriminating moving edges from background edges. We then apply the triangle algorithm to the f_{g_diff} histogram to determine a threshold τ_g, which enables the foreground edge map EM_F to be determined as

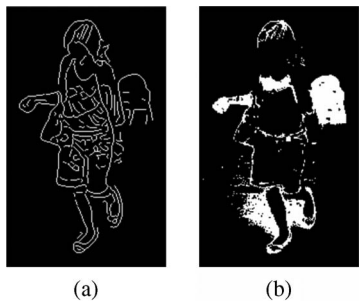
$$EM_F(x,y) = \begin{cases} 1, & \text{if } EM_I(x,y) = 1 \wedge f_{g\_diff}(x,y) > \tau_g \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

and the corresponding foreground gradient map can be obtained as

$$GM_F(x,y) = \begin{cases} GM_I(x,y), & \text{if } EM_F(x,y) = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The EM_F of the input image in Fig. 3(a) is shown in Fig. 7(a).

Given EM_F, we can calculate the edge response of the IBS pairs. As texture difference is calculated using pixels in blocks, the IBS points may not exactly be where their corresponding edge points are. Therefore, to find the edge point that corresponds to an IBS point, a search along the negative normal direction of each IBS point within N pixels is performed. Then, the edge response of a boundary segment C is calculated by



(a) (b)

Fig. 7. (a) EM_F of Fig. 3(a). (b) Moving shadow detection of Fig. 3(a).

$$ER(C) = \frac{1}{N_c} \sum_{n=1}^{N_c} v\left(C(n)\right) \cdot GM_F\left(p(n)\right), \tag{7}$$

where N_c is the number of points on C, $C(n)$ denotes the nth point of C, $v(C(n))$ denotes the unit normal vector of $C(n)$, and $p(n)$ denotes the foreground edge point corresponding to $C(n)$. If there is no foreground edge point that corresponds to $C(n)$, then $p(n)$ is empty, and $GM_F(p(n)) = (0,0)$. In the calculation of $ER(C)$, the dot product operation is applied to consider the edge orientation information because a true boundary point is expected to have a direction similar to that of its corresponding edge point.

b) Curvature evaluation: Curvature is a measure of the smoothness of the curve at a certain point. The curvature of each point $C(n) = (x(n), y(n))$ is given by

$$\kappa(n) = \frac{|\dot{x}(n)\ddot{y}(n) - \dot{y}(n)\ddot{x}(n)|}{(\dot{x}(n)^2 + \dot{y}(n)^2)^{3/2}} \tag{8}$$

where each dot denotes a differentiation with respect to n. The total curvature of C is calculated as the sum of curvature at each point

$$\kappa_c = \sum_{n=1}^{N_c} \kappa(n). \tag{9}$$

We consider the total curvature instead of the mean curvature as it is more representative in that a small κ_c indicates a concise length and a smooth C, which are the characteristics of a true boundary.

c) Shadow detection: Although it is argued that, in general, the segment corresponding to a large ER and a small κ_c is more likely to be the true boundary, there is, however, an exception: Cast shadows caused by strong illumination tend to produce a regular outline (i.e., small κ_c) and a strong edge response (i.e., large ER). In addition, shadows are likely to appear in $Mask_L$ [see Fig. 3(c)] but not in $Mask_H$ [see Fig. 3(e)]. Therefore, shadows need to be explicitly dealt with.

To detect shadows, we adopt the method proposed by Cucchiara *et al.* [25]. It transforms I and B into the hue–saturation–value space and detects a shadow map S as given in (10), shown at the bottom of the next page.

In (10), the first condition refers to the value component. The use of β (less than one) prevents background points that are slightly corrupted by noise to be identified as shadows, whereas α takes shadow intensity into account, i.e., the stronger the light source, the lower the α that is chosen. The second condition assumes that shadows reduce the saturation of points, and the third condition assumes that hue is not affected by shadows too much. Fig. 7(b) depicts the $S(x, y)$ of I given in Fig. 3(a), which appears reasonable overall, except that some true foreground pixels are falsely detected as moving shadows.

d) Decision rules: Given an IBS pair $(C_{\text{low}}, C_{\text{high}})$, with C_{low} coming from B_{low} and C_{high} coming from B_{high}, and the associated ER, κ_c, and S, the decision about whether C_{low} or C_{high} is the true boundary can be made according to the following rule set.

1) For the region enclosed by C_{low} and C_{high}, if it contains a high percentage (we chose 80% in our experiment) of nonzero S pixels, then C_{high} is chosen.
2) If $ER(C_{\text{low}}) = ER(C_{\text{high}}) = 0$, the segment with smaller κ_c is chosen.
3) If $ER(C_{\text{low}}) \geq ER(C_{\text{high}})$ or $ER(C_{\text{low}}) \geq \overline{ER}$, then C_{low} is chosen.
4) If $ER(C_{\text{low}}) = 0$ and $ER(C_{\text{high}}) > 0$, then C_{high} is chosen.
5) If $ER(C_{\text{high}}) - ER(C_{\text{low}}) > \overline{ER}$, then C_{high} is chosen; otherwise, C_{low} is chosen.

$\overline{ER}$ is the estimation of the average edge response of the foreground boundaries and is used to measure whether an edge response is significant or not. It is calculated according to (7) by replacing C with the aggregation of the exterior boundaries of $Mask_M$ (excluding those exterior boundaries of $Mask_M$ that have neglectable edge responses, e.g., smaller than 0.1, as they are quite likely to be the boundaries of false-positive regions and should not be involved in the estimation to avoid estimation bias).

Rule 1 states that if the region enclosed by an IBS pair contains a large percentage of shadow pixels, the region is considered to be a shadow region and discarded. Rule 2 deals with those cases where the foreground edge response is unavailable, in which the segment with a smaller sum of curvature is selected. Rule 3 reflects the preference for C_{low} when it has a stronger or significant edge response. However, if the contrary is true, i.e., $ER(C_{\text{low}}) < ER(C_{\text{high}})$ and $ER(C_{\text{low}}) < \overline{ER}$, we cannot simply choose C_{high} because the difference between $ER(C_{\text{high}})$ and $ER(C_{\text{low}})$ may not be significant. Rule 4 states that if C_{low} has zero edge response and C_{high} does not, then C_{high} is preferred. Rule 5 states that when $ER(C_{\text{high}})$ is significantly larger than $ER(C_{\text{low}})$, we choose C_{high}. If none of the five conditions is met, we choose C_{low}.

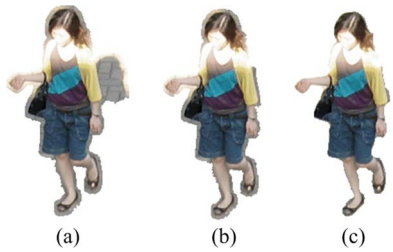


Fig. 8. Foreground extraction results. (a) B_I—After merging B_L and B_M. (b) B_R—After merging B_I and B_H. (c) After boundary refinement.

For those consistent boundary points, the merged boundary point is decided to be the midpoint between the two matched boundary points. Fig. 8(a) depicts B_I after merging B_L and B_M, and Fig. 8(b) depicts B_R after merging B_I and B_H. It can be seen from B_R that the shadow has effectively been suppressed, without sacrificing the accuracy of extracting other foreground pixels. However, there are still halo-like false-positive pixels along B_R, which will be dealt with later.

E. Result Verification

After the boundaries are merged, the resulting boundary is usually a reasonable estimation of the ground truth. However, false-positive regions may still be present in the result. We use the foreground edge map as a measure to remove these regions. For each detected foreground region, its exterior boundary's ER is calculated according to (7). If it is substantially smaller than $\overline{ER}$, this region is considered as a false-positive region. As the ER of a false-positive region is usually significantly smaller than $\overline{ER}$, we select $\overline{ER}/2$ as the threshold. Furthermore, each detected foreground region is also checked to see if it is a shadow region (i.e., it contains a large percentage of shadow pixels), and if it is, it is removed from the detection result.

In addition, to increase the foreground extraction accuracy, we need to differentiate real holes in the foreground region (part of the background) from false holes (part of the foreground). Usually, this problem is solved by optimizing an energy function in which the Markov–Gibbs random field is applied as a knowledge prior [26]. Instead, we solve it within the multithresholding framework. The difference between real holes and false holes is embodied in the following three aspects. First, real holes are always consistently characterized in the three masks, i.e., most of the boundary points are consistent, as illustrated in Fig. 9(b)–(d); on the contrary, false holes tend to produce inconsistent characterizations in the three masks, as depicted in Fig. 9(g)–(i). Second, real holes are normally supported by a large ER, whereas false holes are not, as they might be parts of homogeneous regions. Third, real holes do not contain foreground edge points because they are background regions,

$$S(x,y) = \begin{cases} 1, & \text{if } \alpha \leq \frac{I_V(x,y)}{B_V(x,y)} \leq \beta \wedge |I_S(x,y) - B_S(x,y)| \leq \gamma_S \\ & \quad \wedge \min\left(1 - |I_H(x,y) - B_H(x,y)|, |I_H(x,y) - B_H(x,y)|\right) \leq \gamma_H \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

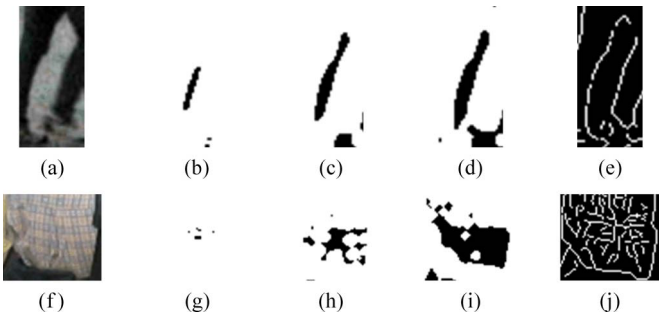


Fig. 9. Example of real holes (first row) and false holes (second row). (a) and (f) I. (b) and (g) $Mask_L$. (c) and (h) $Mask_M$. (d) and (i) $Mask_H$. (e) and (j) EM_F.

but false holes may contain foreground edge points. Therefore, real holes are differentiated from false holes as follows.

For each resulting interior boundary, we have the following conditions.

1) If in any of the two boundary-merging stages more than half of the higher boundary points are determined to be inconsistent, the hole is false.
2) If the ER of the resulting interior boundary is smaller than $\overline{ER}/2$ or if the region inside the resulting interior boundary contains more than ($\overline{ER} \times$ length of the boundary) foreground edge points, the hole is false; otherwise, the hole is true.

F. Boundary Refinement

It is evident that the texture-based method applied here consistently introduces false-positive errors along the object boundary. The thickness introduced will be no more than N pixels if the block size is $(2N+1) \times (2N+1)$. On the other hand, because the foreground edge map has accurate localization of the object boundary, i.e., the deviations of the foreground edge points from their expected true locations are very small, the foreground edge map can be taken as a reference when performing the boundary refinement. Therefore, we erode the detected foreground boundaries by a disk-shaped structuring element with a changeable radius, ranging from 0 to N. For each boundary point, the radius is the smaller one of N, and the maximum radius that ensures that the structuring element would not have overlapping points with the foreground edge map. After performing the erosion, the resultant foreground is likely to be affected by noise introduced in the foreground edge map. Thus, a median filter of size 3×3 is applied to remove the noise. The foreground after boundary refinement of the example image is shown in Fig. 8(c).

G. Computational Complexity

The calculation of the luminance difference and the chrominance difference are trivial as they are pixel-based. However, as a region-based method, direct calculation of the texture difference requires thousands of operations for each pixel, which is unacceptable. To avoid repeated computations, the Integral Image technique [27] is applied to reduce the number of operations needed for calculating the region sums of the texture difference, resulting an improvement of the computational complexity for nearly one order. On the other hand, the cruxes of the proposed method, i.e., multiple-threshold selection, IBS identification, and boundary evaluation, require moderate computational cost. Except for a few operations that are performed on the whole image, such as morphological operations, Canny edge detection, and color space transform, most calculations are in proportion to the length of the objects' boundaries, which are much smaller than the size of the image.

III. Experimental Results

The results presented below focus on human beings, as human shapes are among the most complicated moving objects.

A. Parameter Settings

In our experiment, the size of the block used to compute texture difference is 9×9, i.e., $N = 4$, which is a tradeoff between the stability of the change detection result (stability of change detection result $\propto N$), the computation time (computation time $\propto N^2$), and the halo-like boundary error introduced (displacement boundary error $\propto N$). The parameters for shadow detection are selected as $\beta = 0.9$, $\gamma_s = 0.25$, and $\gamma_H = 0.4$ as they perform well on all the test images. To make α adaptive to different illuminations, we model α as a linear function of illumination

$$\alpha = -0.31 \times lum + 0.63 \tag{11}$$

where lum denotes the illumination and is approximated by the background mean intensity, and the two coefficients 0.31 and 0.63 are learned through experiments.

B. Comparative Analysis

The proposed method has been evaluated on images of various illumination conditions, including indoor, outdoor, moderate, sunny, rainy, and dim cases. Some other change detection methods, including the minimum description length (MDL), the SM [7], the derivative model (DM) [7], and Li's texture-based approach [9], are chosen for comparison. The block size for the first three methods is selected as 9×9, and 5×5 is chosen for Li's method. The Matlab code of the MDL, SM, and DM methods are obtained from the Andra and Al-Kofahi website (http://www.ecse.rpi.edu/censsis/papers/change/). MDL and Li's method adaptively select their thresholds, whereas the thresholds for SM and DM need to be manually set. In our experiment, it is found that the threshold of the DM method can properly be determined by the triangle algorithm, and hence, we use this threshold. For the SM method, we first found the range of thresholds that can produce a complete contour, and then, within this range, we chose the one that produced the lowest error rate when compared with the ground truth in the foreground. The segmentation results are displayed in Figs. 10–16 with the following layout: (a) input image I, (a′) enhanced I, (b) estimated background B, (b′) enhanced B, (c) ground truth, and (d) the result of MDL, (e) the result

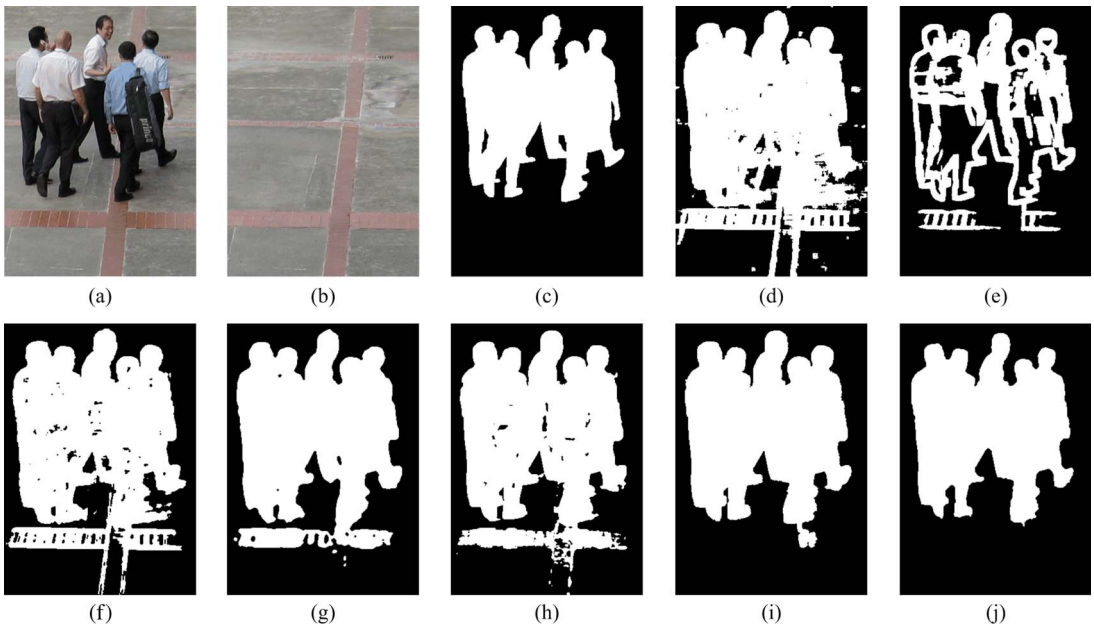


Fig. 10. Case 1—Moderate illumination over a group of people.

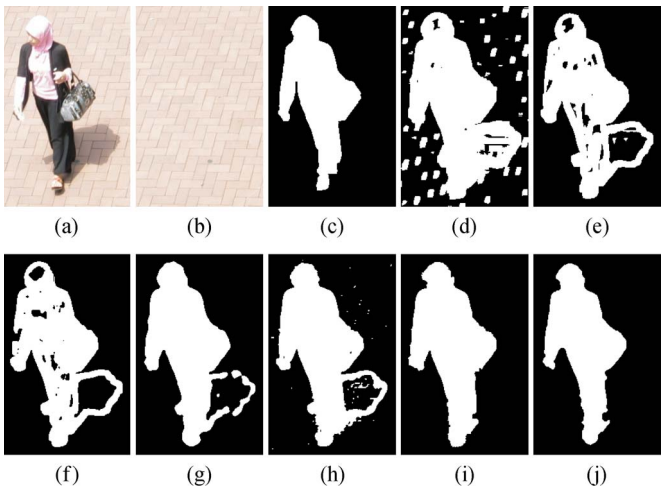


Fig. 11. Case 2—Strong illumination resulting in strong shadow.

of SM, (f) the result of DM, (g) the result of Li's method, (h) the result of ST, (i) the result of the proposed method without refinement, and (j) the result of the proposed method. The ST result is displayed to demonstrate the problems of using just one threshold (τ_M) for the extraction, and the proposed method without refinement is shown to study the impact of not performing refinement at the end. The results are also quantitatively evaluated in terms of the error rate, which is characterized by the Jaccard distance and is defined by

$$\text{Error Rate} = (FP + FN)/(TP + FP + FN) \times 100\% \tag{12}$$

where FP stands for the number of no-change pixels incorrectly detected as change, FN stands for the number of change pixels incorrectly detected as no-change, and TP represents the number of change pixels correctly detected. All the ground-truth foregrounds are accurately determined by hand in advance. The error rates for the proposed method (without or with refinement), ST, and other existing methods are summarized in Table I.

In case 1 (see Fig. 10), the input image contains a group of humans with some insignificant shadows cast on the ground. Note that the ground consists of rectangular patterns constructed by slightly reflective bricks. As can be seen, all four other methods and ST are badly affected by the shadows, whereas the proposed method successfully removes them. Although the MDL method claims to be able to automatically select the threshold, the change detection result is not satisfactory, because the description length is arbitrarily set. The SM method is not able to detect the inner regions of the foreground well enough when both the inner region area and the corresponding background are homogeneous; this is because this method is designed to be insensitive to illumination changes. The DM method is also designed to be illumination invariant; it performs better at the inner flat regions because the intensity values of a block are modeled as a quadratic function, giving it higher discriminability. Li's method can detect the foreground objects reasonably well, which is due to the high discriminability of the texture difference, but the shadows are also taken as change. The ST also responds to the shadows, and without using the proposed multithreshold strategy, the shadows are impossible to delete from the foreground. However, the proposed method fails to detect two holes, i.e., one on the left and one close to the middle, because the holes are not large enough and their boundaries merged in $Mask_L$. On the other hand, Li's method is able to detect one of them, due to the use of a block of smaller size.

In case 2 (see Fig. 11), the shadow is strong. The MDL result is noisy and heavily affected by the shadow. The results of all the methods, except those of the proposed method, are affected by shadow boundaries, and only Li's method and ST can potentially delete the shadow edges by performing a morphological opening operation, whereas SM and DM cannot

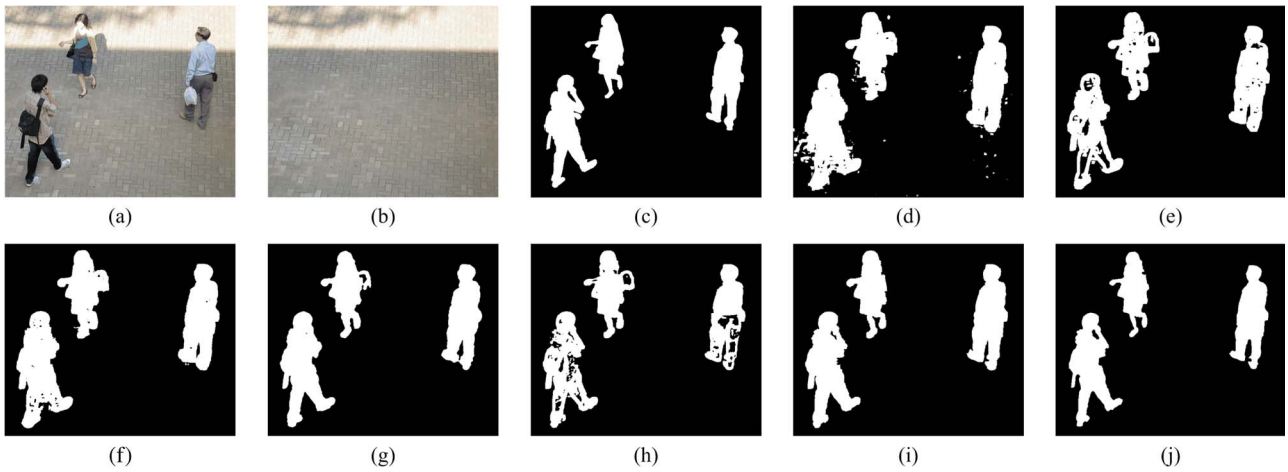


Fig. 12. Case 3—High contrast with strong illumination and cast shadow.

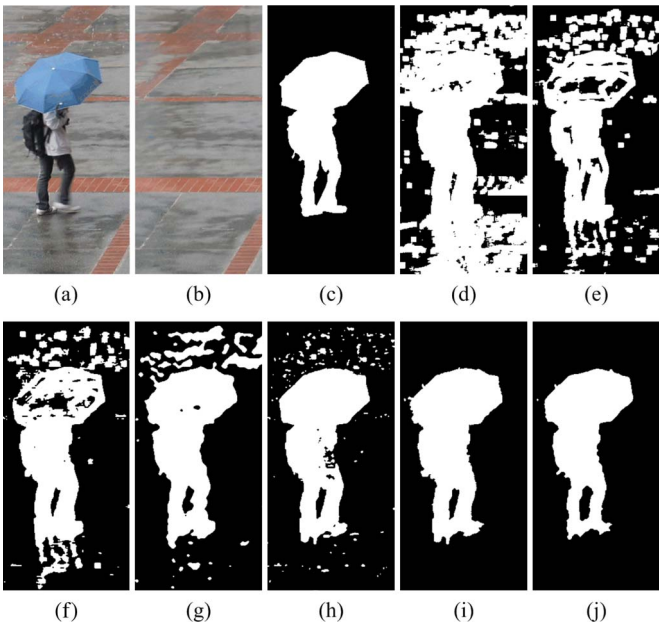


Fig. 13. Case 4—Rain with reflection on the ground.

because the other foreground pixels will be removed at the same time. The superiority of the proposed method over Li's method and ST is that it removes more shadow pixels, e.g., the shadow region at the left side of the woman.

Case 3 (see Fig. 12) contains a scene where the upper part is under the sun and the lower part is in the shadow of a flyover, forming a high-contrast scenario. Li's method performs quite well, except for the shadow and the hole formed by the arm and shoulder of the young man. ST poorly performs with some false holes and parts of the contours missing. The proposed method successfully extracts the contour, removes the shadow, and detects the only hole. The proposed method without refinement has a higher error rate (15.3%) than Li's method (14.7%). This is mainly because the 9×9 block used in the proposed method, compared with the 5×5 block used in Li's method, results in a thicker boundary. After refinement, the error rate of the proposed method is reduced to 5.3%.

Case 4 (see Fig. 13) is taken in the rain. The ST result is degraded by the raindrops, and part of the contour is missing. Due to the use of a high threshold, which is insensitive to the raindrop reflections, the proposed method effectively ignores the raindrops, and owing to the low threshold, the proposed method produces the complete contour. All the other methods are severely affected by the raindrops. MDL, SM, and DM are also affected by the shadow, which is enhanced by the water on the ground.

Case 5 (see Fig. 14) is taken at night. The ST result has some false holes inside the moving objects, and many background pixels are falsely detected as foreground. The proposed method successfully fills in the false holes, detects one real hole, and removes all the background regions. All the other four methods could not detect the inner part of the foreground well, and the SM method also produces a large amount of false positives. Although Li's method performs quite well under normal outdoor conditions, this case illustrates that it poorly performs under weak-illumination situations, where the intensity is strongly affected by noise, and the texture difference becomes less discriminative.

The sixth case (see Fig. 15) is an indoor moderately illuminated image obtained from the PETS database (ftp://ftp.pets.rdg.ac.uk). The difficulty of this case is caused by the flickering illumination, which leads to a large percent of the background pixels being mistakenly detected by MDL, SM, Li's method, and ST. The proposed method gives a much more accurate result (error rate: 6.0%) over the others (error rate: 46.3%–88.8%) in this case.

Case 7 (see Fig. 16) is taken in a hall under weak illumination. From the results, it can be seen that the proposed method again produces the best result. All the other methods leave some interior regions undetected, and MDL, SM, and Li's method also detect the boundaries of the shadow.

We have also carried out similar experiments on 300 images taken under various conditions. However, manually marking all the ground truth is time consuming. Therefore, we only evaluate the results of 60 images, whose distribution over different illumination conditions is similar to that of the 300 images. The average error rate of the proposed method is 6.8%, which

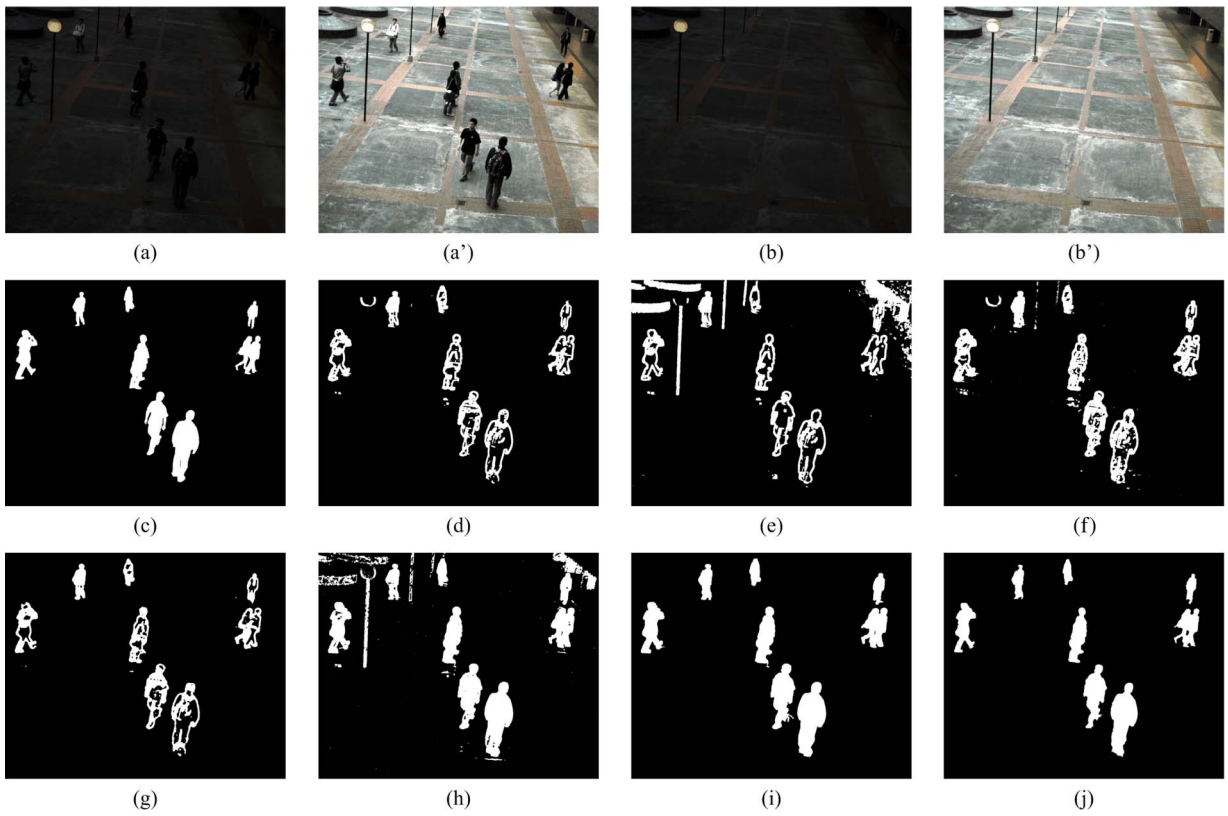


Fig. 14. Case 5—At night with very weak illumination.

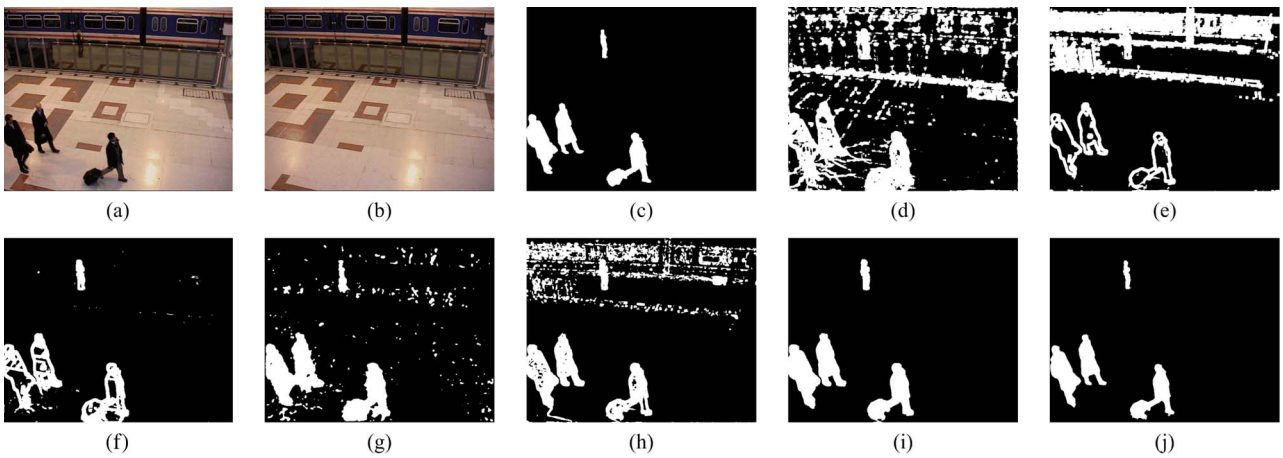


Fig. 15. Case 6—Indoors and moderate illumination.

is consistent with the seven cases depicted above and demonstrates the robustness of the proposed method. The respective number of images, average error rate, and standard deviation of the error rate within each illumination condition are tabulated in Table II.

C. Displacement Error

The proposed boundary refinement step can remove the halo-like boundaries produced by the region-based change detection methods, and the effect can be measured by the displacement error. If a detected boundary point is exactly on the ground-truth boundary, its displacement error is zero; otherwise, if it overlaps with a point on the dilated (or eroded) ground-truth boundary with the dilation (or erosion) radius being r, it has a displacement error of r (or $-r$) pixels. The displacement error distributions before and after boundary refinement of case 2 are shown in Fig. 17, from which we can see that the mode of the distribution has shifted from 4 to 0 pixels. To quantify this improvement, the mean displacement error (MDE) of an extraction result is defined as

$$\text{MDE} = \sum_{r=mnd}^{mpd} |r| p(r) \tag{13}$$

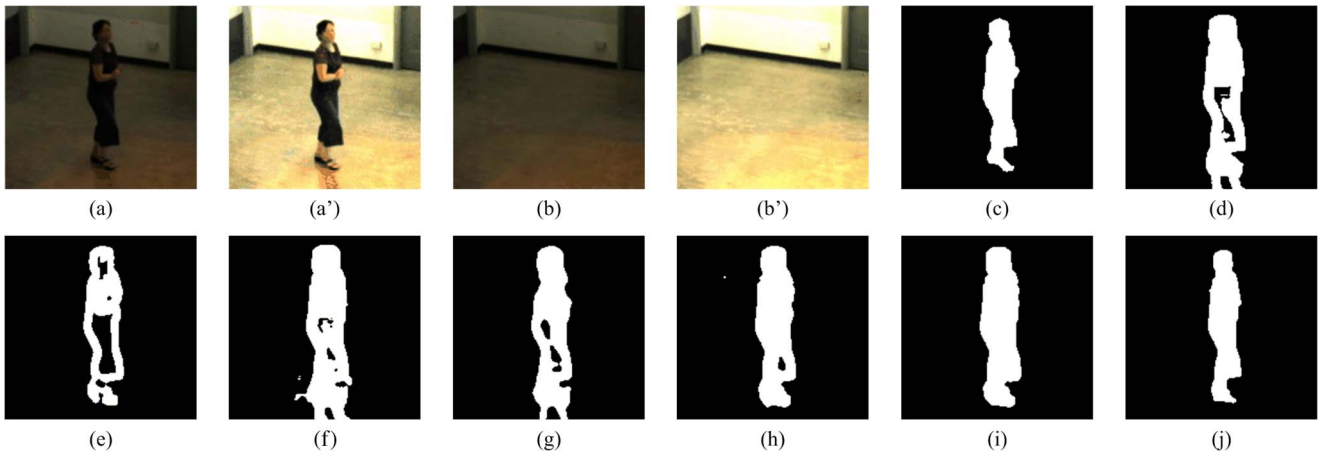


Fig. 16. Case 7—Indoors and weak illumination.

TABLE I
ERROR RATE OF THE PROPOSED METHOD COMPARED WITH SOME EXISTING METHODS

Error Rate (%)	MDL	SM	DM	Li	ST	Proposed – no R	Proposed
Case 1	36.8	47.0	31.0	22.0	25.7	15.7	**5.0**
Case 2	44.4	32.9	32.0	22.0	25.4	17.5	**7.3**
Case 3	28.8	27.2	24.1	14.7	24.9	15.3	**5.2**
Case 4	63.9	49.3	40.9	30.0	20.5	14.0	**6.8**
Case 5	48.8	68.0	40.4	42.6	35.5	19.1	**7.2**
Case 6	79.8	88.8	52.5	46.3	66.8	23.8	**6.0**
Case 7	44.2	47.0	42.6	37.7	30.1	28.0	**7.7**

TABLE II
ERROR RATE OF THE PROPOSED METHOD UNDER DIFFERENT ILLUMINATION CONDITIONS

Illumination condition	Number	Average error rate (%)	Standard deviation of the error rate (%)
Moderate	25	6.1	2.3
Sunny	10	8.1	3.2
Dark	10	6.9	2.5
Rainy	5	8.8	3.6
Indoor	10	6.2	2.2
Total	**60**	**6.8**	----

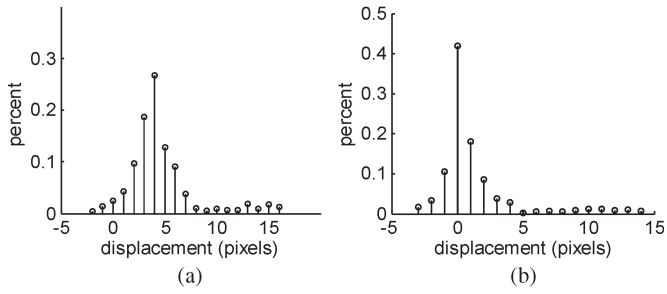


Fig. 17. Displacement error distributions of case 2. (a) Before refinement. (b) After refinement.

where mnd is the maximum negative displacement, mpd is the maximum positive displacement, and $p(r)$ is the percentage of the detected boundary points with displacement r. The MDEs of the above seven cases before and after boundary refinement are summarized in Table III. We can see that the average MDE reduction is 3.48 pixels, which is quite significant, and the actual MDE is smaller than 2 pixels at worst and less than 1 pixel at best.

TABLE III
MEAN DISPLACEMENT ERROR BEFORE AND AFTER BOUNDARY REFINEMENT

MDE (pixels)	Proposed – no R	Proposed	Error reduced
Case 1	6.76	1.17	5.59
Case 2	4.54	1.66	2.88
Case 3	3.60	1.02	2.58
Case 4	4.35	1.79	2.56
Case 5	5.11	1.36	3.75
Case 6	5.19	1.22	3.97
Case 7	3.81	0.77	3.04

D. Computational Cost

Our algorithm is currently implemented in Matlab, and it needs about 30 s on the average to process a 1200×1600 image on a 3.2-GHz personal computer. If the method is implemented in C/C++/C# with code optimization and hardware acceleration, we believe the proposed method can potentially run in real time, particularly if the image size is reduced.

IV. CONCLUSION

A novel foreground-extraction method based on multiple adaptive thresholding and boundary evaluation has been proposed in this paper. By using multiple thresholds, image pixels can be divided into multiple classes, and the problem is reduced into classifying a smaller set of ambiguous pixels into foreground and background pixels. Although thresholding is globally performed, the use of edge response and curvature helps to improve local boundary accuracy during the evaluation stage. Furthermore, true shadows always appear in the regions associated with IBS pairs and can largely be removed from the foreground. This way of shadow removal is statistical, which is better than first extracting changes and then removing shadows from the changes, because shadow detection can hardly be accurate and can potentially produce many faults. By applying a boundary-refinement method, the halo-like regions along the boundary are effectively removed. The classification error rate compares well with the ST approach and other existing change-detection methods.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[2] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[3] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 246–252, Aug. 2000.

[4] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[5] D. M. Tsai and S. C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, Jan. 2009.

[6] Y. Z. Hsu, H. Nagel, and G. Reckers, "New likelihood test methods for change detection in image sequences," *Comput. Vis. Graph. Image Process.*, vol. 26, no. 1, pp. 73–106, Apr. 1984.

[7] K. Skifstad and R. Jain, "Illumination independent change detection for real world image sequences," *Comput. Vis., Graph. Image Process.*, vol. 46, no. 3, pp. 387–399, Jun. 1989.

[8] S. Liu, C. Fu, and S. Chang, "Statistical change detection with moments under time-varying illumination," *IEEE Trans. Image Process.*, vol. 7, no. 9, pp. 1258–1268, Sep. 1998.

[9] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 105–112, Feb. 2002.

[10] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[12] T. W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, no. 8, pp. 630–632, Aug. 1978.

[13] G. W. Zack, "Automatic measurement of sister chromatid exchange frequency," *J. Histochem. Cytochem.*, vol. 25, no. 7, pp. 741–753, 1977.

[14] Y. Kita, "Change detection using joint intensity histogram," in *Proc. Int. Conf. Pattern Recog.*, Hong Kong, 2006, pp. 351–356.

[15] D. Sen and S. K. Pal, "Histogram thresholding using fuzzing and rough measures of associated error," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 879–888, Apr. 2009.

[16] L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[17] P. L. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," in *Proc. Brit. Mach. Vis. Conf.*, Birmingham, U.K., 1995, pp. 347–356.

[18] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vis. Graph. Image Process.*, vol. 29, no. 3, pp. 273–285, 1985.

[19] S. B. Gray, "Local properties of binary images in two dimensions," *IEEE Trans. Comput.*, vol. C-20, no. 5, pp. 551–561, May 1971.

[20] L. O'Gorman, "Binarization and multi-thresholding of document images using connectivity," in *Proc. Symp. Document Anal. Inf. Retrieval*, Las Vegas, NV, 1994, pp. 237–252.

[21] J. Zhou and J. Hoang, "Real time robust human detection and tracking system," in *Proc. IEEE Comput. Vis. Pattern Recog. Workshop*, San Diego, CA, 2005, p. 149.

[22] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure, "Robust foreground extraction technique using background subtraction with multiple thresholds," *Opt. Eng.*, vol. 46, no. 9, pp. 097 004-1–097 004-12, 2007.

[23] W. W. L. Lam, C. C. C. Pang, and N. H. C. Yung, "A highly accurate texture-based vehicle segmentation method," *Opt. Eng.*, vol. 43, no. 3, pp. 591–603, 2003.

[24] P. L. Rosin, "Unimodal thresholding," *Pattern Recognit.*, vol. 34, no. 11, pp. 2083–2096, Nov. 2001.

[25] R. Cucchiara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[26] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Kauai Marriott, HI, 2001, pp. 511–518.

Lu Wang received the B.Eng. and M.Eng. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 2003 and 2005, respectively. She is currently working toward the Ph.D. degree with the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, University of Hong Kong.

Her research interests include image processing, computer vision, and pattern recognition.

Nelson H. C. Yung (S'82–M'85–SM'96) received the B.Sc. and Ph.D. degrees from the University of Newcastle Upon Tyne, Newcastle upon Tyne, U.K.

From 1985 to 1990, he was a Lecturer with the University of Newcastle Upon Tyne. From 1990 to 1993, he was a Senior Research Scientist with the Department of Defence, Australia. In late 1993, he joined the University of Hong Kong (HKU) as an Associate Professor. He is the founding Director of the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, HKU. He acts as consultant to government units and a number of local and international companies. He is the author or a coauthor of five books and book chapters and more than 150 journal and conference papers in the areas of digital image processing, parallel algorithms, visual traffic surveillance, autonomous vehicle navigation, and learning algorithms. He was a Guest Editor of the *SPIE Journal of Electronic Imaging*. He also serves as a Reviewer for a number of IEEE, IET, and SPIE journals.

He is a Chartered Electrical Engineer. He is a Member of the Hong Kong Institution of Engineers and the Institution of Electrical Engineers. He was the Regional Secretary of IEEE Asia-Pacific Region, a Council Member and the Chairman of Standards Committee of Intelligent Transportation Systems–Hong Kong (ITS-HK), and the Chair of Computer Division, International Institute for Critical Infrastructures. He was a member of the Advisory Panel of the ITS Strategy Review, Transport Department, Government of the Hong Kong Special Administrative Region. His biography has been published in *Who's Who in the World* since 1998.