IPSJ Transactions on Computer Vision and Applications Vol. 3 9–20 (Mar. 2011)

Research Paper

Object Detection Using Background Subtraction and Foreground Motion Estimation

Tadaaki Hosaka,[†1] Takumi Kobayashi[†2] and Nobuyuki Otsu[†2]

A method for detecting moving objects using a Markov random field (MRF) model is proposed, based on background subtraction. We aim at overcoming two major drawbacks of existing methods: dynamic background changes such as swinging trees and camera shaking tend to yield false positives, and the existence of similar colors in objects and their backgrounds tends to yield false negatives. One characteristic of our method is the background subtraction using the nearest neighbor method with multiple background images to cope with dynamic backgrounds. Another characteristic is the estimation of object movement, which provides robustness for similar colors in objects and background regions. From the viewpoint of the MRF, we define the energy function by considering these characteristics and optimize the function by graph cut. In most cases of our experiments, the proposed method can be implemented in (nearly) real time, and experimental results show favorable detection performance even in difficult cases in which methods of previous studies have failed.

1. Introduction

Extracting moving objects from image sequences is a fundamental problem in computer vision systems that are used in a variety of applications, such as video surveillance, human tracking, motion analysis, and image synthesis. In most of these cases, a stationary camera is used, which means that background subtraction techniques can be a powerful tool for object detection.

The simplest implementation of background subtraction is to evaluate the difference between a background image (captured previously or estimated) and a current image in a pixel-wise manner, and then thresholding the difference value to determine the pixels that belong to moving objects. This simple background subtraction is used in many applications, because it can be easily implemented with low computational costs and without any *a priori* knowledge of the target objects. However, this method has some major drawbacks. First, it cannot adapt to dynamic background changes, such as fluttering leaves, illumination changes, or camera shaking. Second, the detection performance deteriorates when moving objects and the background contain similar colors. These drawbacks generally produce false positives and false negatives, respectively.

In this paper, we deal with such pixel-level moving objects detection, and present a novel method for overcoming the above-mentioned drawbacks. In our approach, we define an energy function based on a Markov random field (MRF) and minimize the function by graph cut [1),2)]. The proposed method holds multiple background images to cope with dynamic background changes and performs the subtraction process with small computational costs using the nearest neighbor method. Further, object motion is estimated by interframe local patch matching for each pixel, and incorporating this motion information into the energy function leads to favorable performance, even when the colors in the foreground and background are similar. Before describing the details of our method, we briefly review some previous approaches.

1.1 Previous Work

Many algorithms have been proposed to tackle the false detection due to dynamic background changes. Most of these algorithms attempt to model the distribution of the background and often that of target objects as well. Refer to Ref. 3) for an inclusive survey of background modeling techniques. Wren, et al. [4)] used a single Gaussian distribution to model both the background and target objects (usually human), and the distributions were updated every frame. Haritaoglu, et al. [5)] adaptively determined the binarization threshold according to the maximum and minimum of pixel values in multiple background images updated in frame by frame. Further, they achieved human tracking based on the current subtraction result and the location of previously tracked objects. Stauffer and Grimson [6)] utilized the Gaussian mixture model in a pixel-wise manner to model the background and provided an update rule for parameters of the distributions based on the K-means method. They detected moving objects as pixels indicating the low likelihood of background. Elgammal, et al. [7)] stressed

†1 Tokyo University of Science
†2 National Institute of Advanced Industrial Science and Technology

that nonparametric distributions were needed to adapt to more general and complex background variations. From this viewpoint, they constructed a background model by means of the Parzen estimation. Ko, et al. [8] utilized a histogram as a nonparametric model and measured the Bhattacharyya distance to determine the fore/background assignment. Because most of these methods construct pixel-wise distributions, they are generally susceptible to noise. Thus, post processing such as erosion and dilation has often been utilized to obtain less noisy output. As another direction, Dickinson, et al. [9] addressed this problem by modeling homogeneous regions using a mixture of Gaussians in color and spatial coordinates.

Approaches based on MRF have also been proposed, which naturally incorporate prior knowledge of local smoothness, i.e., spatially (and temporally) neighboring pixels tend to have the same affiliation (object/background in this case). This provides better robustness to noise compared with former approaches. The energy minimization on MRF is equivalent to the maximum a *posteriori* (MAP) estimation described in Section 3.1. Kamijo, et al. [10] utilized the spatiotemporal MRF to determine the correct boundary for occluded regions among multiple objects. Because they resorted to the time-consuming Markov chain Monte Carlo method (Gibbs sampler) to minimize their energy function, the application of MRF was limited to local regions. Howe and Deschamps [11] defined a global energy function consisting of two terms, the subtraction value and the local smoothness, and minimized the function by graph cut. This method, however, uses only one background image and cannot adjust to dynamic background changes. Migdal and Grimson [12] modeled the background using the Gaussian mixture model [6], and then defined the energy function incorporating spatial and temporal smoothing effects. Although this approach can adjust to dynamic background changes, the Markov chain Monte Carlo method is required to minimize the energy function, which does not seem practical in terms of computational cost. Sheikh and Shah [13] generated distributions of target objects and background with a single function for each in the entire image and updated them in a frame-by-frame manner. They utilized not only RGB values but spatial locations as feature vectors and constructed these distributions by the Parzen estimation. The energy function consisting of the induced log likelihood ratio and the local smoothness is minimized by graph cut. Although their method is currently thought of as a sophisticated algorithm in terms of quality, the performance tends to deteriorate at the regions where objects and their background have similar colors, because they did not explicitly utilize motion information. Sun, et al. [14] proposed a smoothness term in the energy function to reduce the segmentation errors caused by strong edges in background regions. Mu, et al. [15] reduced computational cost by performing energy minimization only for boundary regions which are separately obtained with simple block-based segmentation using the frame difference and color information. Tang and Miao [16] improved the Gaussian mixture model and the update rule of relevant parameters to accelerate the processing speed of the MAP-MRF estimation. As another application of MRF in regard to background subtraction, McHugh, et al. [17] proposed a simple idea for determining the binarization threshold on a pixel-by-pixel basis using the MRF model.

The above-mentioned methods share the common weakness that the performance of their algorithms often deteriorates when similar colors exist in objects and the background. Zhang and Yang [18] recently found that detection errors (mainly false negatives) due to color similarities occurred around a confusion point, where the posterior probabilities of belonging to the foreground and background were equal. To reduce the false negatives in such regions, they purposely increased the posterior probability for foreground and decreased that for background. However, the performance improvement is still not sufficient, because it does not incorporate object-motion information which can be key to overcoming this difficulty. Yu, et al. [19] utilized the idea of tracking for object detection. In their algorithm, the foreground and background spatial-color (five-dimensional) Gaussian mixture models are combined, and the spatial components are first updated for this joint model using the EM algorithm with color components fixed, which corresponds to the object tracking. Then, graph cut is performed for minimizing the induced energy function, and the color components are finally updated according to the segmentation results. Because the reliability of color components is important for updating spatial components, the specific designation of target objects is required in some way, and the objects coming into frames are difficult to detect. Tang and Gao [20] focused on the contour of segmented objects and incorporated its similarity to the prior shape silhouette in the energy function

of MRF. To consider the time evolution of the silhouette by dynamic modeling, multiple learning samples are usually required, which seems unpractical in real situations. Bugeau and Pérez [21),22)] utilized motion information (optical flow) as well as color information to construct the energy function although their purpose was not necessarily to improve the accuracy when similar colors exist in objects and the background.

The main objective of this study is to resolve this remaining problem caused by color similarities, while maintaining the tolerance for dynamic background changes. Under the framework of MRF, we define an energy function based on background subtraction and motion estimation without any prior knowledge of target objects, and minimize it by graph cut. Our major contribution is in the definition of the energy function, and this will be described in Section 3 after we show the flow of our proposed method.

2. Algorithm Flow

In this section, we provide an overview of the flow of our algorithm. **Figure 1** shows the block diagram of our algorithm, and the process of each block is briefly described below. A detailed explanation is given in the next section from the viewpoint of constructing the energy function.

Generation of initial background samples: We initialize background samples. Even when a background scene without moving objects is not captured, in most cases, the (sub-)median images for a few hundred frames can work well as background samples.

Increment of frame number: The following processes are executed on a frame-by-frame basis.

Background subtraction by nearest neighbor search: For each pixel, we perform background subtraction, in which the difference between the pixel value in the current frame and that of each background sample is calculated. The minimum among calculated subtraction values, which is obtained by the nearest neighbor search, constitutes the data term of our energy function.

Interframe matching for foreground objects: We perform interframe local patch matching only for the object regions of the previous frame. The pixels predicted as the object destination are biased toward the foreground by modifying the data term.

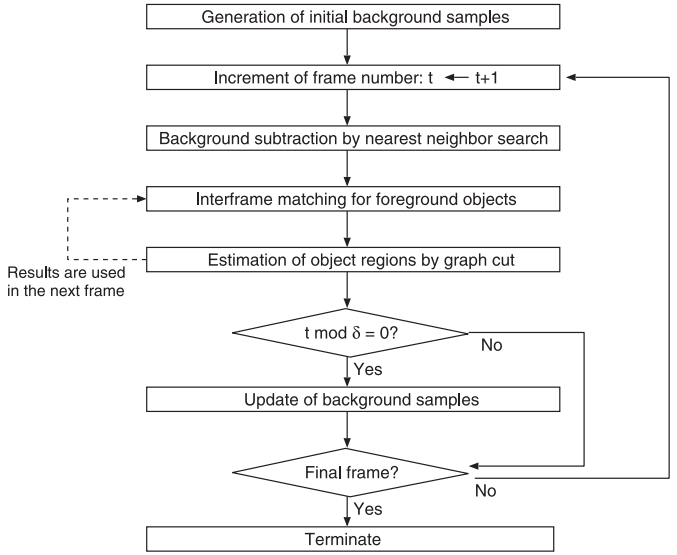


Fig. 1 Block diagram of our algorithm. Note that our algorithm is basically executed on a frame-by-frame basis and that the interframe matching for object regions is performed between the current frame and the adjacent former frame.

Estimation of object regions by graph cut: We estimate the object and background regions by minimizing the energy function consisting of the data term and the spatial smoothing term. This minimization is conducted by graph cut.

Update of background samples: We update the background samples every δ frames by discarding the oldest sample and incorporating the current frame.

3. Energy Function

Let us first define notation used in this paper. The RGB values of pixel i ($i = 1, 2, \ldots, N$; N is the number of pixels) in the t-th frame ($t = 1, 2, \ldots$) are expressed as $\boldsymbol{c}_i^t$ (three-dimensional vector), and the set of them is denoted as $\boldsymbol{C}^t = \{\boldsymbol{c}_1^t, \boldsymbol{c}_2^t, \ldots, \boldsymbol{c}_N^t\}$. In this paper, the detection of moving objects means assigning each pixel a label $l_i^t (\in \{0, 1\})$, which indicates that it belongs to the

moving object by $l_i^t = 1$ and background by $l_i^t = 0$. The array of these labels is expressed as a vector, $\boldsymbol{l}^t = (l_1^t, l_2^t, \ldots, l_N^t)$. The index of a frame number, t, is omitted when there is no possibility of misunderstanding.

3.1 MAP-MRF Estimation and Energy Minimization

In the t-th frame, the posterior probability of labels, $\boldsymbol{l}^t$, for a given image $\boldsymbol{C}^t$ can be defined as the following Gibbs distribution,

$$P(\boldsymbol{l}^t|\boldsymbol{C}^t) = \frac{1}{Z(\boldsymbol{C}^t)} \exp\{-U(\boldsymbol{l}^t; \boldsymbol{C}^t)\}, \tag{1}$$

where $U(\boldsymbol{l}^t; \boldsymbol{C}^t)$ is an energy function that should take smaller values in more favorable configurations of $\boldsymbol{l}^t$, and $Z(\boldsymbol{C}^t) = \sum_{l_1^t} \sum_{l_2^t} \cdots \sum_{l_N^t} \exp\{-U(\boldsymbol{l}^t; \boldsymbol{C}^t)\}$ is a normalization factor. According to the MAP strategy, the detection results, $\boldsymbol{l}^{t*}$, are estimated by maximizing this posterior probability.

Our energy function is defined as

$$U(\boldsymbol{l}^t; \boldsymbol{C}^t) = \lambda \sum_{i \in \mathcal{P}} v_i(l_i^t; \boldsymbol{c}_i^t) + \sum_{(ij) \in \mathcal{N}} s_{ij}(l_i^t, l_j^t; \boldsymbol{c}_i^t, \boldsymbol{c}_j^t), \tag{2}$$

where v_i is generally called the data term representing the unlikelihood of assigning pixel i to an object or background label, and s_{ij} is a smoothing function of neighboring pixels, which reflects the prior knowledge for general images. These two terms are described in more details in the following sections. Parameter $\lambda(>0)$ controls the balance between these two terms. The symbols, $\mathcal{P}$ and $\mathcal{N}$, represent a set of all pixels and that of all nearest neighboring pixel pairs, respectively. This energy function is defined on an MRF where the pixel-wise posterior probability derived from the above joint posterior probability (1) depends only on labels of the nearest neighbors.

This MAP-MRF estimation is equivalent to the minimization of the energy function, i.e.,

$$\boldsymbol{l}^{t*} = \underset{\boldsymbol{l}^t}{\operatorname{argmin}}\ U(\boldsymbol{l}^t; \boldsymbol{C}^t). \tag{3}$$

In this study, this optimization is implemented by graph cut described in Section 3.4 instead of maximizing the posterior probability Eq. (1).

3.2 Data Term for Label Assignment

3.2.1 Nearest Neighbor Search Among Multiple Background Images

We require a background model, which can adapt to dynamic background changes. As described in the previous section, the Gaussian mixture model and the Parzen estimation using multiple background images has been recently used for this purpose. Given a new observation, thresholding the induced background probability or the likelihood ratio between foreground and background probabilities is a representative method for determining the fore/background assignment. However, under this probabilistic strategy, in some cases, the rare patterns with small contributions in the learning samples are erroneously detected as foreground. Even such infrequent background patterns should be correctly recognized as background. We present an efficient nonparametric background modeling that achieves this purpose by means of the nearest neighbor method.

In the t-th frame, L previously observed images $\boldsymbol{c}^{t_\alpha}(t_\alpha < t; \alpha = 1, 2, \ldots, L)$ serve as background samples written as $\boldsymbol{B}^\alpha = \{\boldsymbol{b}_1^\alpha, \boldsymbol{b}_2^\alpha, \ldots, \boldsymbol{b}_N^\alpha\}$. Note that these samples do not necessarily consist of successive L frames. We also retain the estimation results for the corresponding previous frames $\boldsymbol{l}^{t_\alpha}$, which are reexpressed as $\boldsymbol{m}^\alpha = (m_1^\alpha, m_2^\alpha, \ldots, m_N^\alpha)$.

We evaluate the following value to determine whether the pixel i in the t-th frame belongs to objects or the background:

$$d_i^t = \min_{\alpha \text{ s.t. } m_i^\alpha = 0} ||\boldsymbol{c}_i^t - \boldsymbol{b}_i^\alpha||^2. \tag{4}$$

As small values of d_i^t represent affinity to background and vice versa, the data term $v_i(l_i^t; \boldsymbol{c}_i^t)$ is defined by thresholding this value as

$$\begin{aligned} v_i(l_i^t = 0; \boldsymbol{c}_i^t) &= \Theta(d_i^t - T_A), \\ v_i(l_i^t = 1; \boldsymbol{c}_i^t) &= 1 - \Theta(d_i^t - T_A), \end{aligned} \tag{5}$$

where T_A is a threshold, and $\Theta(x)$ is the step function as $\Theta(x) = 1$ $(x \geq 0)$, and 0 otherwise. However, for pixels which have no valid background sample, i.e., $m_i^1 = m_i^2 = \ldots = m_i^L = 1$, we define $v_i(l_i = 0; \boldsymbol{c}_i) = v_i(l_i = 1; \boldsymbol{c}_i) = 0.5$. In the above process Eq. (4), the nearest neighbor of $\boldsymbol{c}_i^t$ is searched for among L background samples in the RGB space. When these L samples include probable background variations by setting L appropriately, this model can adjust to dy-

namic background changes. Furthermore, this method can correctly determine even infrequent patterns as background using the nearest neighbor search.

Considering that the data term v_i takes only the value of 1 or 0, it does not necessarily require L references to implement the process Eq. (4). Once a value below T_A is found, the remaining references are redundant, which significantly contributes to reducing computational costs. The proposed method requires L references only when the data term becomes 1. Therefore, when the region of moving objects is relatively small, which is common in most real situations such as video surveillance, the required calculations can be reduced greatly.

We need to renew the background samples as time advances. Although it is ideal to pick a small number of representative frames that can cover all probable background patterns, the automatic selection of these frames is highly difficult. Thus, in our study, the background images are simply updated every δ frames (usually, a few frames) by discarding the oldest sample and incorporating the current frame. In this process, the objects that are stationary during some frames are eventually incorporated into the background. This seems reasonable because our purpose is to detect moving objects, and when the object moves again, it can soon be correctly detected as the foreground.

In the situation where we can obtain enough background images without moving objects, they are available as L initial background samples. Otherwise, we convert the first $2L$ frames to gray-scale images, then sort the $2L$ intensity values for each pixel, and finally extract L RGB components corresponding to the intensities that lie around the median. In most of the cases that we tried, L images generated from these extracted RGB values serve well as the initial background samples.

3.2.2 Utilization of Motion Information

The performance in many previous studies deteriorated when objects and background contained similar colors. For such difficult regions, the difference value defined by Eq. (4) is small, and this probably leads to false negatives. To solve this problem, Zhang and Yang purposely multiplied the probability of belonging to the foreground [18]. However, they did not explicitly utilize motion information, and as a result, false positives conversely occur around boundaries between the foreground and its background. Each frame generally has significant correlations with the previous frame, and consequently, using the history of detection results is effective for subsequent estimation. In our method, we track the object by a small patch (block) and modify the data term only for relevant pixels. From the viewpoint of optical flow, we implement interframe matching and modify the value of v_i obtained by Eq. (5).

In the t-th frame, the following processes are performed only for pixels where $l_i^{t-1}=1$ (i.e., the object regions estimated in the previous frame).

1) For the $(t-1)$-th frame, we define a 15-dimensional vector consisting of RGB values of pixel i and its four nearest neighboring pixels. This vector is expressed as $\boldsymbol{g}_i^{t-1}$.

2) For the t-th frame, a square search region of w pixels on each side centered at pixel i is considered, and this region is expressed as $\mathcal{W}(i)$. The 15-dimensional vector for pixels in the region are expressed as $\boldsymbol{g}_j^t$ $(j \in \mathcal{W}(i))$.

3) The pixel j^* in region $\mathcal{W}(i)$ having the minimum matching error with $\boldsymbol{g}_i^{t-1}$ is obtained by

$$j^* = \underset{j \in \mathcal{W}(i)}{\operatorname{argmin}} ||\boldsymbol{g}_j^t - \boldsymbol{g}_i^{t-1}||^2. \tag{6}$$

4) In the case of $i \neq j^*$ and $||\boldsymbol{g}_{j^*}^t - \boldsymbol{g}_i^{t-1}||^2 < T_B$, the data term is modified so as to increase the affinity of pixel j^* to objects as

$$v_{j^*}^t(l_{j^*}^t=0; \boldsymbol{c}_{j^*}^t) \leftarrow v_{j^*}^t(l_{j^*}^t=0; \boldsymbol{c}_{j^*}^t) + \beta. \tag{7}$$

In this process, T_B and $\beta\,(>0)$ are predetermined parameters. When the best matching point is the same as the original pixel, i.e., $i=j^*$, we do not modify the data term because a pixel of a false positive in the previous frame often matches itself, and a pixel of a moving object should not match itself.

Parameter β can be regarded as a temporal smoothing term on a three-dimensional MRF, representing that the most similar pixels in terms of the above 15-dimensional local vectors are expected to belong to object regions in the current frame. Further, the spatial smoothing term is also incorporated in Section 3.3.

There are other possibilities for utilization of motion information. For instance, we can employ a probability density estimation based on the proposed 15-dimensional joint features. However, since these inference methods also re-

quire more computational cost, we adopt the above simple motion estimation on the basis of optical flow for real-time implementations. The extension of the current framework is left for future works.

The 15-dimensional vector defined above is expected to incorporate color texture information to some extent and be useful for object detection. For real images, even if some regions are not visually distinguishable, the RGB values of pixels included in these regions usually have at least slight differences. Multiple dimensions in a patch are expected to enhance these (slight) differences due to the increase in possible configurations. Utilizing a patch size larger than the five pixels (reference pixel and its four nearest neighbors) tends to invoke detection errors, particularly in boundaries between the foreground and background. This tendency becomes pronounced when the target objects are fairly small because the background region may occupy most of the patch and the contribution of objects is very small. The image sequences we deal with in Section 4 include relatively small objects; therefore, we limited the patch size to the five pixels. We have also used this 15-dimensional vector for the segmentation of static images [23], and its effectiveness and some comparisons are shown there.

3.3 Spatial Smoothing Term

When neighboring pixels have similar RGB values, labels for these two pixels are expected to be the same. This prior knowledge is incorporated into the energy function as a smoothing term defined by

$$s_{ij}(l_i, l_j; \boldsymbol{c}_i, \boldsymbol{c}_j) = \frac{1}{\ln(||\boldsymbol{c}_i - \boldsymbol{c}_j|| + 1 + \epsilon)} |l_i - l_j|, \tag{8}$$

where $\epsilon \ll 1$. Since a smoothing term utilized in some previous methods such as Refs. 11)–13) does not reflect the influence of the gradient between the pixels, estimation errors tend to arise around foreground and background boundaries. For segmentation of static images, an edge-preserving smoothness term has been known to be highly effective [24),25)].

3.4 Optimization by Graph Cut

Minimization Eq. (3) is a combinatorial optimization problem, and its implementation with respect to whole variables $\boldsymbol{l} = (l_1, l_2, \ldots, l_N)$ seems difficult. Recently, graph cut algorithms have been used to solve this type of MAP-MRF optimization within a practical time scale not only for moving object detection but also for segmentation, stereo matching, and so on [26),27)].

Graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$ consists of a set of vertices $\mathcal{V}$ including two special nodes, called source and sink, and a set of (directed) edges between nodes $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An edge from node i to node j is assigned with the capacity expressed as $q(i, j)$. The minimum cut problem is to divide the set $\mathcal{V}$ into two subgroups, such that each group includes only a source or sink, and the sum of the capacity of excluded edges is minimized. This problem can be solved in the polynomial order with respect to the number of nodes [1),2)].

According to the argument of Boykov and Jolly [26)], in the current problem, the special nodes, source and sink, correspond to a moving object and background, respectively, and the other general nodes correspond to pixels. Defining the capacities as

$$\begin{aligned} q(i,j) &= q(j,i) \\ &= \begin{cases} \frac{1}{\ln(||\boldsymbol{c}_i - \boldsymbol{c}_j||+1+\epsilon)} & ((ij) \in \mathcal{N}) \\ 0 & \text{otherwise,} \end{cases} \\ q(\text{source}, i) &= \lambda \cdot v_i(l_i = 0; \boldsymbol{c}_i), \\ q(i, \text{sink}) &= \lambda \cdot v_i(l_i = 1; \boldsymbol{c}_i), \end{aligned} \tag{9}$$

the solution of this minimum cut problem is equivalent to that of Eq. (3). We use the efficient algorithm proposed by Boykov and Kolmogorov [28)] to solve this minimum cut problem.

4. Experimental Results

We investigated the performance of the proposed method for various sequences. Although the values of the parameters $\lambda, L, \delta, T_A, T_B, \beta$, and w significantly affect the final detection accuracy, it is currently difficult to automatically optimize them depending on sequences, which is the same as in most of the previous studies. Because it seems impractical to use different values for each sequence, we try to empirically (in a brute-force way) find the values of these parameters used in common for every sequence to detect objects as favorably as possible. **Table 1** shows the tuned values of the parameters. Although the optimal value of w actually depends on the velocity of moving objects which can be estimated by the optical flow, $w = 11$ adapted to most of the sequences in our experiments.

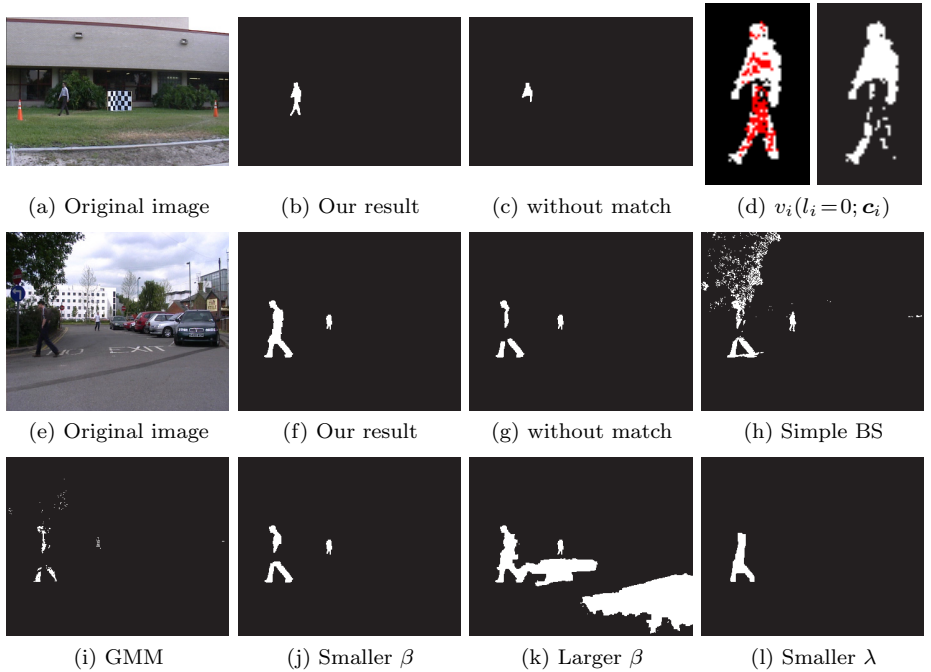
Table 1 The values of parameters.

λ	L	δ	T_A	T_B	β	w
0.22	50	3	250	350	3.6	11/31

Since the frame rate of the sequence for Fig. 5 (a)-(c) is so low that movements of persons appear to be intermittent, we increased the value of w to 31 only for this sequence. The parameters in previous methods used for comparison were also optimized manually.

Figure 2 shows the effectiveness of our method. Both sequences have been utilized in the context of the gait recognition problem [29),30)] and published by the authors. Figure 2 (a) and (e) depict the situation where a man with black pants is entering a region of dark background. Results by the proposed method are shown in Fig. 2 (b) and (f) indicating favorable detection performance. On the other hand, the results without interframe matching (this corresponds to $T_B = 0$) are shown in Fig. 2 (c) and (g), and we can see that motion information is useful in detecting the lower body of the person. Figure 2 (d) shows the value of $v_i(l_i = 0; \boldsymbol{c}_i)$ as binary images for these cases (left: with the interframe matching, right: without the interframe matching). Pixels matching with object regions of the previous frame are painted in red except for self-matched pixels, and it can be seen that parts from the hip to thighs are appropriately tracked by using the interframe matching, while this region shows an affinity to background without the matching. Figure 2 (h) shows the result of simple background subtraction for comparison. Although the detection area becomes large with a smaller threshold, we also have more false positives in the upper-left region of swinging trees. Because the simple background subtraction retains only one background sample, it is difficult to adaptively handle dynamic background changes. Figure 2 (i) shows the result using the method of the Gaussian mixture model [6)], where a mixture Gaussian distribution serves as the background model. This result is obtained by setting the parameters such that false positives in the upper-left region vanish, which conversely leads to false negatives in the object region. The proposed method provides no false positives in the upper-left region because it holds multiple background samples to handle background changes.

Figure 3 shows another example, in which the camera on a tripod is swinging



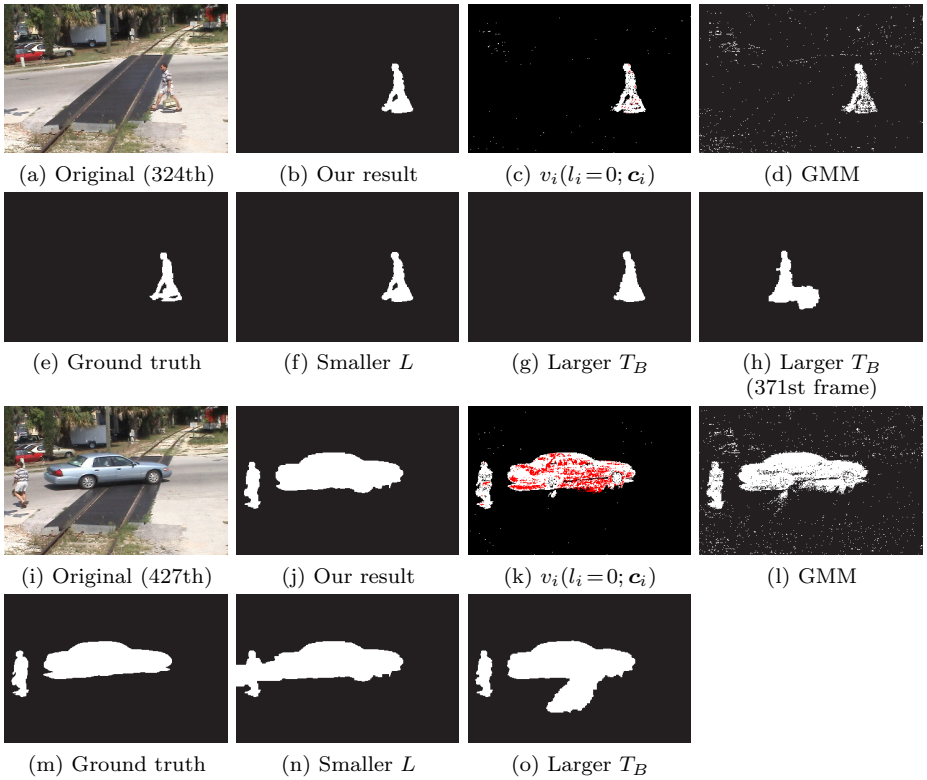
(a) Original image (b) Our result (c) without match (d) $v_i(l_i = 0; \boldsymbol{c}_i)$

(e) Original image (f) Our result (g) without match (h) Simple BS

(i) GMM (j) Smaller β (k) Larger β (l) Smaller λ

Fig. 2 Example of situations where a moving object and its background have similar colors. (a), (e): The black pants of a person is in front of the shade of trees, which blurs the boundary. (b), (f): Detection results of the proposed method. Motion information of the target estimated by interframe matching works effectively for the detection. (c), (g): Detection results without interframe matching. (d): The value of data term $v_i(l_i = 0; \boldsymbol{c}_i)$. Pixels in white and black represent $v_i(l_i = 0; \boldsymbol{c}_i) = 1$, and $v_i(l_i = 0; \boldsymbol{c}_i) = 0$, respectively. Pixels matching with some foreground pixels of the previous frame are red in the left image. (h): Results for simple background subtraction (BS). Setting the threshold reasonably low, false detection in a upper-left region is pronounced because of swinging trees. (i): Results using the background model constructed by the Gaussian mixture model (GMM). Although this method can adjust to the swinging trees by tuning some parameters included in the algorithm, false negatives become pronounced in the object region. (j), (k), (l): Examples of estimation failure due to inappropriate parameter settings.

and captured images are also shaking in every frame. Even in such a difficult case, our results are favorable (Fig. 3 (b) and (j)). Figure 3 (d) and (l) show results by the method of the mixture Gaussian distribution [6)] with the parameters set



(a) Original (324th) (b) Our result (c) $v_i(l_i=0;\boldsymbol{c}_i)$ (d) GMM

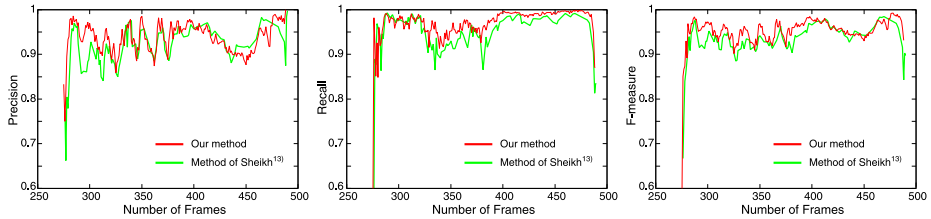
(e) Ground truth (f) Smaller L (g) Larger T_B (h) Larger T_B (371st frame)

(i) Original (427th) (j) Our result (k) $v_i(l_i=0;\boldsymbol{c}_i)$ (l) GMM

(m) Ground truth (n) Smaller L (o) Larger T_B

Fig. 3 Example of a situation where the camera is shaking. These are detection results for the 324th and 427th frames (figures (a) and (i)). Our results show favorable performance (figures (b) and (j)). Figures (d) and (l) show results by the method of the Gaussian mixture model [6] with parameters adjusted such that the object region is detected as much as possible. In this case, false detection is pronounced all over the image compared to our raw subtraction results before performing the graph cut, i.e., the data term $v_i(l_i=0;\boldsymbol{c}_i)$ (figures (c) and (k)). Ground truth data generated by hand [13] is shown in figures (e) and (m). Results by changing the value of one parameter are shown in the figures (f)-(h), (n), (o).

to detect a moving person as correctly as possible, which leads to a lot of false positives in the background region despite the fact that the mixture Gaussian distribution was originally introduced to adjust to dynamic background changes. This is because infrequent background patterns cannot form a component large



(a) Precision (b) Recall (c) F-measure

Fig. 4 Quantitative evaluation. The precision (a), recall (b), and F-measure (c) are calculated for the movies shown in Fig. 3. The superiority of our method compared to the method of Sheikh and Shah [13] is observed in almost all frames. Values before the 270th frame are excluded because there is no moving object in these frames. Around the 270th and 490th frames, the number of true positives is so small that the occurrence of only a few more mistakes may have a large detrimental effect. Because the MAP-MRF estimation tends to yield slightly smoother outlines of objects due to the smoothing term, some of the evaluated values are relatively low for these frames.

enough to be correctly recognized as background. On the other hand, our method on the basis of the nearest neighbor search can recognize such rare patterns as the background correctly, as shown in Fig. 3 (c) and (k), expressing the value of $v_i(l_i=0;\boldsymbol{c}_i)$. Using ground truth data manually produced by Sheikh and Shah [13] (Fig. 3 (e) and (m)), a quantitative evaluation is performed for this movie by calculating the precision, recall, and F-measure, defined as

$$\text{Precision} = \frac{\text{the number of true detected positives}}{\text{the number of whole detected positives}}, \tag{10}$$

$$\text{Recall} = \frac{\text{the number of true detected positives}}{\text{the number of whole true positives}}, \tag{11}$$

$$\text{F-measure} = \frac{2}{(1/\text{Precision}) + (1/\text{Recall})}. \tag{12}$$

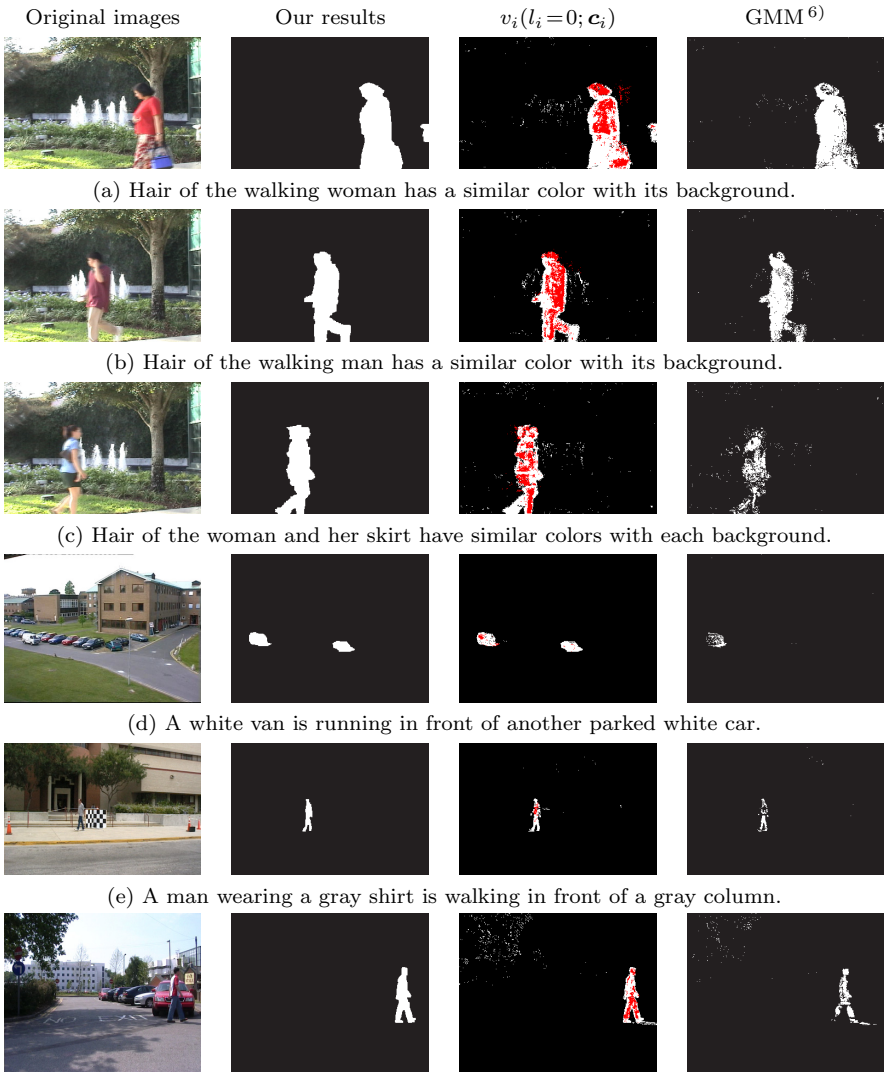
Because the precision and recall are generally in relation of a tradeoff, the F-measure is frequently utilized as an integrated criterion, which becomes 1 for a perfect estimation without excess or deficiency. **Figure 4** shows these three quantities. Since our method incorporates motion information, it is expected that false negatives decrease even when the foreground objects have similar colors to their background (in this sequence, the clothes that a person wears have a similar color to that of the ground). This can be observed from the fact that the recall

of our method is better in almost all frames than that of the method of Sheikh and Shah [13]. On the other hand, the precision provides an indication of false positives. Our precision values are better than or comparable to those of the method of Sheikh and Shah in the former part of the sequence. However, our precision becomes slightly lower in the latter part, because false detection comes into existence at regions beneath the car around the 400th frame and these errors remain (Fig. 3 (j)) until the 475th frame. This may be regarded as a kind of side effect attributed to interframe matching, and in the worst case, this false detection could be propagated with the expansion into the surrounding regions, as shown below. From the viewpoint of F-measure, it can be said that except for a fraction of the latter part, the performance of our method is better than or comparable to that of the sophisticated method of Sheikh and Shah. This indicates the effectiveness of the proposed method.

As described in the first part of this section, parameter tuning is important for obtaining favorable results. Some failure examples due to inappropriate parameter settings are shown in Fig. 2 (j)-(l) and Fig. 3 (f)-(h), (n), (o), where one of the parameters is changed while the others remain unchanged. The smaller value of β (= 1.5, in Fig. 2 (j)) reduces the effect of object tracking, and as a result, false negatives increase as in the case of Fig. 2 (g). On the other hand, a too-large value of β (= 10.0, in Fig. 2 (k)) tends to provide counterresults, because the interframe matching is not necessarily correct and accidentally matched pixels are likely to be determined as the foreground when such a large value of β is used. In the worst scenario, these mismatched pixels are propagated in subsequent frames, as shown in Fig. 2 (k). Also, adjusting the value of λ is generally crucial for energy minimization in MRF. In Fig. 2 (l), the smaller value of λ (= 0.08, in this case) over-smoothes the object outlines, and parts of the head and feet and the smaller person moving around the center of the image are taken into the background. As for the sequence of Fig. 3, it seems that a certain number of background samples are needed to adjust to dynamic background changes due to the shaking of the camera. When setting the smaller value of L (= 10, in Fig. 3 (f), (n)), the number of pixels having $v_i(l_i = 0; \boldsymbol{c}_i) = 1$ increases. As a result, we can observe the expansion of object regions in Fig. 3 (n), in which the smoothing effect works negatively to bridge the two objects, while it works positively to suppress the unfavorable false positives in Fig. 3 (f). In the limit of infinite T_B, the modification of the data term is necessarily performed for the pixel having the minimum matching error, unless it is the self-matched pixel. However, this operation could increase false positives if a pixel in the actual background regions is (weakly) matched, as happens for various reasons. When setting the larger value of T_B (= 300, in Fig. 3 (g), (h), (o)), we run into this undesirable situation in the middle of the sequence (Fig. 3 (h)), and worse, these false positives are propagated around (Fig. 3 (o)).

Further examples in which similar colors are included in both target objects and background regions are shown in **Fig. 5**. These movies except the last two were used in recent works [13),18)], and those previous algorithms could not achieve sufficient detection performance in the extracted frames (for their results, see Refs. 13), 18)). Our results are far more favorable than the previous algorithms, and in most cases, motion information works effectively for detecting objects, even when the foreground and background contain similar colors. In Fig. 5 (a), the detected region on the right-hand edge of the image is another entering object, not a false detection. In Fig. 5 (d), a car around the center of the image is detected because it has just stopped there, and the background samples have not yet been completely updated. We can understand this from the fact that the car has few tracked pixels; note that a pixel matching itself is not painted in red. Finally, we show the results for consecutive frames in **Fig. 6** to enhance the reliability of our proposed method. The base frame in each row corresponds to that of the earlier examples in Figs. 2, 3 and 5. It can be observed that favorable performances are achieved without a drastic deterioration during these frames.

Our method works at a low computational cost, owing to the nearest neighbor search in the subtraction process. An example of computational time is as follows: The speed of our method is 25 fps for the video sequence in Fig. 3 (image size is 360 × 240) on a personal computer with a Core 2 Duo Processor 2.67 GHz and 4 GB RAM without parallel computation. When moving objects are much smaller compared to the image size, as in Fig. 5 (d), the processing speed of our method exceeds 30 fps.



(a) Hair of the walking woman has a similar color with its background.

(b) Hair of the walking man has a similar color with its background.

(c) Hair of the woman and her skirt have similar colors with each background.

(d) A white van is running in front of another parked white car.

(e) A man wearing a gray shirt is walking in front of a gray column.

(f) A man in red shirt is walking in front of a red car.

Fig. 5 Examples of difficult cases, where foreground objects and background regions have similar colors.

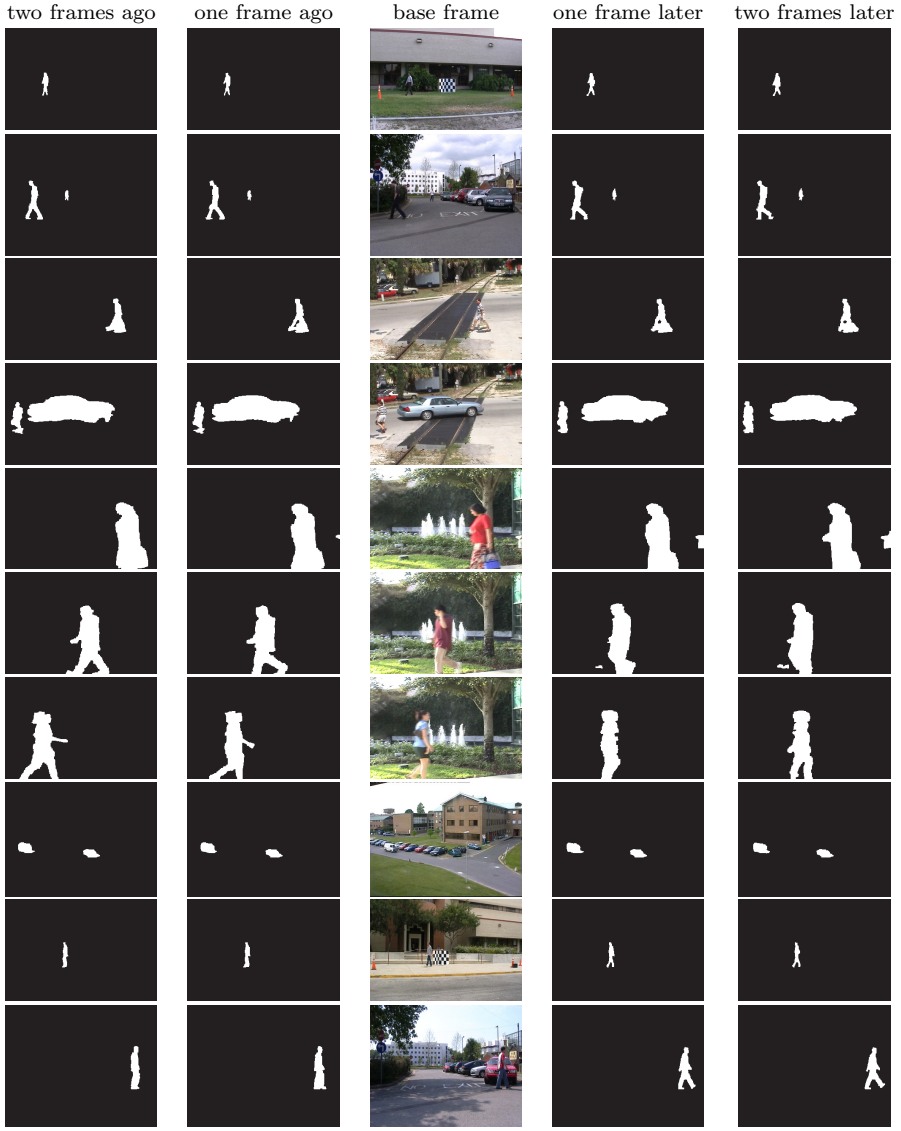


Fig. 6 Results for some consecutive frames. The base frames are those displayed in Figs. 2, 3 and 5.

5. Conclusions

In this paper, we proposed a novel moving object detection method based on a MAP-MRF approach. Our method utilizes the nearest neighbor method with multiple background samples to accommodate dynamic background changes. Besides, object motion estimation is incorporated to improve the detection performance. As a result, the performance of our method is better than that of previous methods, even when similar colors exist in objects and the background.

A remaining issue for our method is how to adaptively determine the set of parameters, $\lambda, L, \delta, T_A, T_B$, and β, the optimal values of which can produce more favorable detection results. Statistical inference methods such as the maximum of the marginal likelihood [31] could be used for this parameter estimation. Another direction is to apply other color spaces such as the normalized RGB space and the HSV color space, which have been used for object detection in the context of adapting to illumination changes [32] or removing shadows [33].

Acknowledgments This study was supported by the advanced surveillance technology project of MEXT in Japan.

References

1) Ford, L.R. and Fulkerson, D.R.: Maximal Flow through a Network, *Canadian Journal of Mathematics*, Vol.8, pp.399–404 (1956).
2) Goldberg, A.V. and Tarjan, R.E.: A New Approach to the Maximum-flow Problem, *J. ACM*, Vol.35, pp.921–940 (1988).
3) Piccardi, M.: Background Subtraction Techniques: A Review, *Proc. IEEE International Conference on Systems, Man and Cybernetics*, pp.3099–3104 (2004).
4) Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P.: Pfinder: Real-Time Tracking of the Human Body, *Proc. IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, pp.780–785 (1997).
5) Haritaoglu, I., Harwood, D. and Davis, L.S.: W^4: Who? When? Where? What? A Real Time System for Detecting and Tracking People, *Proc. 3rd Face and Gesture Recognition Conference*, pp.222–227 (1998).
6) Stauffer, C. and Grimson, W.E.L.: Learning Patterns of Activity Using Real-Time Tracking, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.22, pp.747–757 (2000).
7) Elgammal, A., Duraiswami, R., Harwood, D. and Davis, L.S.: Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, *Proc. IEEE*, pp.1151–1163 (2002).
8) Ko, T., Soatto, S. and Estrin, D.: Background Subtraction on Distributions, *Proc. European Conference on Computer Vision*, pp.276–289 (2008).
9) Dickinson, P., Hunter, A. and Appiah, K.: A Spatially Distributed Model for Foreground Segmentation, *Image and Vision Computing*, Vol.27, pp.1326–1335 (2009).
10) Kamijo, S., Matsushita, Y., Ikeuchi, K. and Sakauchi, M.: Occlusion Robust Tracking Utilizing Spatio-Temporal Markov Random Field Model, *Proc. International Conference on Pattern Recognition*, pp.142–147 (2000).
11) Howe, N.R. and Deschamps, A.: Better Foreground Segmentation through Graph Cuts, *eprint cs/0401017* (2004).
12) Migdal, J. and Grimson, W.E.L.: Background Subtraction Using Markov Thresholds, *Proc. IEEE Workshop on Motion and Video Computing*, pp.58–65 (2005).
13) Sheikh, Y. and Shah, M.: Bayesian Modeling of Dynamic Scenes for Object Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.27, pp.1778–1792 (2005).
14) Sun, J., Zhang, W., Tang, X. and Shum, H.-Y.: Background Cut, *Proc. European Conference on Computer Vision*, pp.628–641 (2006).
15) Mu, Y., Zhang, H., Wang, H. and Zuo, W.: Automatic Video Object Segmentation Using Graph Cut, *Proc. IEEE International Conference on Image Processing*, pp.III 377–III 380 (2007).
16) Tang, Z. and Miao, Z.: Fast Background Subtraction Using Improved GMM and Graph Cut, *Proc. Congress on Image and Signal Processing*, pp.181–185 (2008).
17) McHugh, J.M., Konrad, J., Saligrama, V. and Jodoin, P.M.: Foreground-Adaptive Background Subtraction, *IEEE Signal Processing Letters*, Vol.16, pp.390–393 (2009).
18) Zhang, X. and Yang, J.: The Analysis of the Color Similarity Problem in Moving Object Detection, *Signal Processing*, Vol.89, pp.685–691 (2009).
19) Yu, T., Zhang, C., Cohen, M., Rui, Y. and Wu, Y.: Monocular Video Foreground/Background Segmentation by Tracking Spatial-Color Gaussian Mixture Models, *Proc. IEEE Workshop on Motion and Video Computing*, p.5 (2007).
20) Tang, P. and Gao, L.: Video Object Segmentation Based on Graph Cut with Dynamic Shape Prior Constraint, *Proc. International Conference on Pattern Recognition*, pp.1–4 (2008).
21) Bugeau, A. and Pérez, P.: Track and Cut: Simultaneous Tracking and Segmentation of Multiple Objects with Graph Cuts, *EURASIP Journal on Image and Video Processing*, pp.1–14 (2008).
22) Bugeau, A. and Pérez, P.: Detection and Segmentation of Moving Objects in Complex Scenes, *Computer Vision and Image Understanding*, Vol.113, pp.459–476 (2009).
23) Hosaka, T., Kobayashi, T. and Otsu, N.: Image Segmentation Using MAP-MRF Estimation and Support Vector Machine, *Interdisciplinary Information Sciences*,

Vol.13, pp.33–42 (2007).
24) Li, Y., Sun, J., Tang, C.K. and Shum, H.Y.: Lazy Snapping, *Proc. ACM SIGGRAPH 2004*, pp.303–308 (2004).
25) Rother, C., Kolmogorov, V. and Blake, A.: GrabCut - Interactive Foreground Extraction Using Iterated Graph Cuts, *Proc. ACM SIGGRAPH 2004*, pp.309–314 (2004).
26) Boykov, Y.Y. and Jolly, M.P.: Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Image, *Proc. IEEE International Conference on Computer Vision*, pp.105–112 (2001).
27) Boykov, Y.Y., Veksler, O. and Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.23, pp.1222–1239 (2001).
28) Boykov, Y.Y. and Kolmogorov, V.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.26, pp.1124–1137 (2004).
29) Sarker, S., Phillips, P.J., Liu, Z., Vega, I.R., Grother, P. and Bowyer, K.W.: The HumanID Gait Challenge Problem: Data Sets, Performance, and Analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.27, pp.162–177 (2005).
30) Shutler, J., Grant, M., Nixon, M.S. and Carter, J.N.: On a Large Sequence-Based Human Gait Database, *Proc. Recent Advances in Soft Computing*, pp.66–71 (2002).
31) Inoue, J. and Tanaka, K.: Dynamics of the Maximum Marginal Likelihood Hyperparameter Estimation in Image Restoration: Gradient Descent Versus Expectation and Maximization Algorithm, *Physical Review E*, Vol.65, p.016125 (2002).
32) Xu, M. and Ellis, T.: Colour-Invariant Motion Detection under Fast Illumination Changes, *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, Boston, Mass., USA, Kluwer Academic, pp.101–112 (2002).
33) Cucchiara, R., Grana, C., Piccardi, M. and Prati, A.: Detecting Moving Objects, Ghosts, and Shadows in Video Streams, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.25, pp.1337–1342 (2003).

(Received April 9, 2010)
(Accepted December 7, 2010)
(Released March 3, 2011)

(Communicated by *Kazuhiko Sumi*)

Tadaaki Hosaka received his Doctor of Science degree from Tokyo Institute of Technology, Japan in 2005. After a postdoctoral fellow in National Institute of Advanced Industrial Science and Technology, he currently works for Tokyo University of Science as an assistant professor. His research area includes Bayesian inference, its applications for image recognition, and statistical mechanical informatics.

Takumi Kobayashi received his Master of Engineering degree from the University of Tokyo in 2005, and his Doctor of Engineering degree from University of Tsukuba in 2009. He received the PRMU Award in 2008 and the Best Paper Award in 2010 from the Institute of Electronics, Information and Communication Engineers (IEICE) for excellence in his research. His current research interests are pattern recognition, multivariate analysis, and their applications to image processing.

Nobuyuki Otsu received his Master and Doctor degrees in Mathematical Engineering from the University of Tokyo in 1971 and 1981, respectively. He is currently a fellow of National Institute of Advanced Industrial Science and Technology. His research interests include pattern recognition, image processing, multivariate analysis, and artificial intelligence.