KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 5, NO. 4, April 2011 782

Probabilistic Background Subtraction in a Video-based Recognition System

Heesung Lee[1], Sungjun Hong[2] and Euntai Kim[2]
[1]Samsung S1 Co., Ltd.
Samsung S1 building, Soonhwa-dong, Joong-gu, Seoul, Korea
[e-mail: 4u2u@yonsei.ac.kr]
[2]School of Electrical and Electronic Engineering, Yonsei University
Sinchon-dong, Seodaemun-gu, Seoul, Korea
[e-mail: {imjune, etkim}@yonsei.ac.kr]
*Corresponding author: Euntai Kim

Received January 24, 2011; revised March 20, 2011; accepted April 14, 2011; published April 29, 2011

Abstract

In video-based recognition systems, stationary cameras are used to monitor an area of interest. These systems focus on a segmentation of the foreground in the video stream and the recognition of the events occurring in that area. The usual approach to discriminating the foreground from the video sequence is background subtraction. This paper presents a novel background subtraction method based on a probabilistic approach. We represent the posterior probability of the foreground based on the current image and all past images and derive an updated method. Furthermore, we present an efficient fusion method for the color and edge information in order to overcome the difficulties of existing background subtraction methods that use only color information. The suggested method is applied to synthetic data and real video streams, and its robust performance is demonstrated through experimentation.

Keywords: Background subtraction, probabilistic approach, belief, video-based recognition system, foreground detection.

This work was supported by “Cognitive model-based global localization for indoor robots” (Project number:10031687) of the Ministry of Knowledge Economy, Republic of Korea.

DOI: 10.3837/tiis.2011.04.009

1. Introduction

In video-based recognition systems, stationary cameras are used to monitor an area of interest. These systems focus on a section of the foreground in the video stream and the recognition of the events occurring in this area. The foreground includes moving objects that have entered the scene. With increasing processor power, more attention has been given to the visual monitoring system because the extraction of the foreground is crucial in many vision applications such as video surveillance [1][2], traffic monitoring [3], gesture recognition [4], gait recognition [5][6], human detection and behavior recognition [7], and human-computer interaction [8]. The detection of moving objects can be achieved by comparing each new frame with a background model and by distinguishing the moving objects from the background [9][10] through a process known as background subtraction. Roughly speaking, background subtraction consists of two phases: background modeling and foreground extraction.

Many studies of background subtraction have been reported [11][12], and most of them have focused on background modeling. For example, a statistical background subtraction method based on Gaussian distribution was proposed in [9]. In that study, the intensity value of a pixel was modeled using a single unimodal Gaussian distribution. However, that model does not work well in the case of dynamic natural environments since they include repetitive motions. To overcome this problem, the Gaussian mixture model (GMM) and its variations were employed. In [13], the plain GMM was used to model complex backgrounds. In [14], the GMM was combined with color and gradient information. The GMM was combined with Bayesian frameworks and mean-shift analysis in [15] and [16], respectively, and variations of GMM were reported in [17] and [18].

In addition to the GMM, non-parametric techniques have been proposed for background modeling using the general kernel density estimation method to construct a statistical representation of the background model [1]. In [19], an advanced version using an adaptive non-parametric technique was proposed. In [20], the background model was constructed pixel by pixel using a non-parametric quantization and clustering technique in which each pixel was clustered into a set of codebooks. In [21], the color similarity problem in background subtraction was solved by shifting the confusion point and improving the foreground model accuracy. Additional edge information was used to design a background model in [22], and the fusion of edge and intensity information was used in [14] and [23]. In [24], a motion-based approach was proposed for background subtraction. The algorithm utilizes salient motion information by integrating frame to frame optical flow over time. A texture-based approach was also proposed [25], in which the algorithm uses textural information to model the background, and the local binary pattern is selected as the measure of texture. Each pixel is modeled as a group of histograms that are calculated over a circular region around the pixel.

Some research has also focused on foreground extraction. Unlike background modeling, in which all of the images are used to update the model, most of the works on foreground extraction used only the current frame to extract the foreground from the image [10][20][25]. Only a few studies utilized earlier frames [26][27]. Even those that utilized the earlier frames used only some of the images, employing those within the finite window size using an autoregressive process.

Our method employs a new probabilistic approach to effectively utilize all earlier images to improve the quality and reliability of background subtraction. The probabilistic framework is one of the most promising tools for handling the uncertainty associated with the fuzzy theory [28]. General background subtraction methods ignore information from the past frames in the decision process (detection phase). The information of the past frames is only used to update the background model. To address this limitation, we represent the posterior probability of the foreground based on the current image as well as all earlier images and derive an updated method. Furthermore, we present an efficient fusion method for the color and edge information in order to overcome the difficulties of existing background subtraction methods that use only the color information.

The remainder of this paper is organized as follows. In Section 2, we describe the structure of the background model and its implementation. In Section 3, we present a strategy for combining the color and edge information. Furthermore, we propose a new background subtraction method based on the belief and derive updated equations for both the belief and the background model. In Section 4, we demonstrate the performance of the proposed method by applying it to synthetic data and real video streams. Finally, in Section 5, we offer our research conclusions.

2. Background Modeling

The proposed background subtraction method consists of two phases: modeling and detection. In the modeling phase, background images are collected to construct a statistical model. In the detection phase, each input image is compared with the background model, and the foreground is segmented. Also in the detection phase, a new measure is defined and used instead of the simple probability. The architecture of the proposed method is shown in **Fig. 1**.

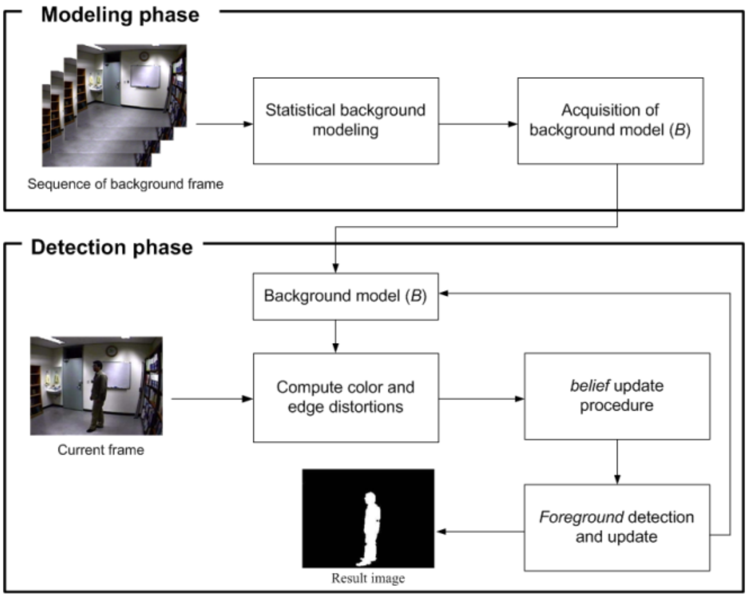


Fig. 1. The architecture of the proposed algorithm

The background model used in this paper combines a color model and an edge model. In this section, we define the distortion values used to represent the difference between the current image and the background model. Later, we design a background model based on the above distortions.

2.1 Color Model and Color Distortion

Let X_i be a sequence of N background color images defined as

$$X_i = \{\mathbf{X}_i(1), \mathbf{X}_i(2), ..., \mathbf{X}_i(N)\}$$
$$= \{[X_i^R(1), X_i^G(1), X_i^B(1)], [X_i^R(2), X_i^G(2), X_i^B(2)], ..., [X_i^R(N), X_i^G(N), X_i^B(N)]\},$$

where i is the pixel location, and R, G, and B denote red, green, and blue colors, respectively. The number in the parentheses denotes the index of the image in the sequence. **Fig. 2** shows the three-dimensional RGB color model used in [9].

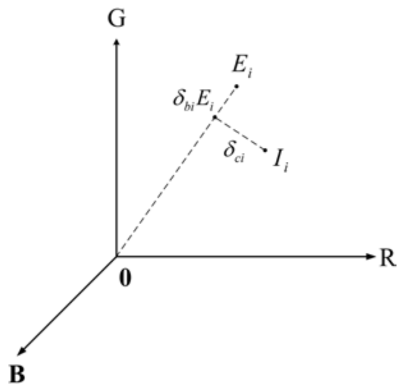


Fig. 2. The color model in three-dimensional RGB color space

In **Fig. 2**, $\mathbf{I}_i = [I_i^R, I_i^G, I_i^B]$ is the ith pixel in the current image, and $\mathbf{E}_i = [E_i^R, E_i^G, E_i^B]$ is the expected RGB color values computed according to

$$\mathbf{E}_i = \frac{1}{N}\left[\sum_{n=1}^{N} X_i^R(n), \sum_{n=1}^{N} X_i^G(n), \sum_{n=1}^{N} X_i^B(n)\right]. \quad (1)$$

In order to represent the discrepancy between $\mathbf{I}_i$ and $\mathbf{E}_i$, we define the *brightness distortion* and *chromaticity distortion*. These two distortions are employed to discriminate the foreground from the background. The brightness distortion of the ith pixel δ_{bi} is defined as the value that brings the current image color $\mathbf{I}_i$ closest to the expected chromaticity line OE_i that passes through the origin and the point $\mathbf{E}_i$. δ_{bi} is obtained using

$$\delta_{bi} = \arg\min_{\delta_{bi}}\left[(\mathbf{I}_i - \delta_{bi}\mathbf{E}_i)^2\right]. \quad (2)$$

The variable represents the strength of the brightness with respect to the expected value. The chromaticity distortion δ_{ci} is defined as the orthogonal distance between the current image color and the expected chromaticity line, and it is computed as

$$\delta_{ci} = \|\mathbf{I}_i - \delta_{bi}\mathbf{E}_i\|. \quad (3)$$

Commercial cameras usually have different sensitivities to different colors. In order to balance the three color channels, we rescale the pixel color values according to their standard deviation

$$\mathbf{S}_i = \left[\sigma_i^R, \sigma_i^G, \sigma_i^B\right], \tag{4}$$

where σ_i^R, σ_i^G, and σ_i^B are the standard deviations of the red, green, and blue values of the ith pixel, respectively, computed over N background frames. Therefore, using the color standard deviation $\mathbf{S}_i$, we can rewrite (2) and (3) as

$$\begin{aligned}\delta_{bi} &= \arg\min_{\alpha_i}\left[\left(\frac{I_i^R - \delta_{bi}E_i^R}{\sigma_i^R}\right)^2 + \left(\frac{I_i^G - \delta_{bi}E_i^G}{\sigma_i^G}\right)^2 + \left(\frac{I_i^B - \delta_{bi}E_i^B}{\sigma_i^B}\right)^2\right] \\ &= \left(\frac{I_i^R E_i^R}{\sigma_i^{R2}} + \frac{I_i^G E_i^G}{\sigma_i^{G2}} + \frac{I_i^B E_i^B}{\sigma_i^{B2}}\right) \Big/ \left(\left(\frac{E_i^R}{\sigma_i^R}\right)^2 + \left(\frac{E_i^G}{\sigma_i^G}\right)^2 + \left(\frac{E_i^B}{\sigma_i^B}\right)^2\right)\end{aligned} \tag{5}$$

and

$$\delta_{ci} = \sqrt{\left(\frac{I_i^R - \delta_{bi}E_i^R}{\sigma_i^R}\right)^2 + \left(\frac{I_i^G - \delta_{bi}E_i^G}{\sigma_i^G}\right)^2 + \left(\frac{I_i^B - \delta_{bi}E_i^B}{\sigma_i^B}\right)^2} \tag{6}$$

respectively, as in [9].

2.2 Edge Distortion

When an object enters the field of view, it occludes some part of the background, and we can detect it by comparing the colors of the current image with those of the background model. However, when a foreground object and background model are similar, the subtraction results in a false negative (a missing foreground pixel) [14][23]. In this paper, we combine color and edge information to improve the performance of a background subtraction method. A sobel mask is employed to detect the edges in each color channel [23]. The *edge distortion* δ_{ei} is defined as the signed city block distance between the edge of the current image and that of the expected background image, and it is computed according to

$$\delta_{ei} = \left\|\mathbf{EI}_i - \mathbf{G}_i\right\|_c = \left(EI_i^R - EG_i^R\right) + \left(EI_i^G - EG_i^G\right) + \left(EI_i^B - EG_i^B\right), \tag{7}$$

where $\mathbf{EI}_i = \left[EI_i^R, EI_i^G, EI_i^B\right]$ is a vector of the color edges of the current image, and $\mathbf{G}_i = \left[EG_i^R, EG_i^G, EG_i^B\right]$ is a vector of the color edges of the background model.

2.3 Parameters of the Background Model

Assume that we are given N background images $\mathrm{X} = \{\mathbf{X}(1), \mathbf{X}(2), ..., \mathbf{X}(N)\}$ and we train a background model B based on these images. The background model B is statistically modeled on a pixel by pixel basis using a collection of local models. The goal of the background model B is to construct and maintain a statistical representation of the scene captured by the camera [25]. As in subsection 2.1, we denote the background pixels for the ith pixel as

$$X_i = \{\mathbf{X}_i(1), \mathbf{X}_i(2), ..., \mathbf{X}_i(N)\} = \left\{\left[X_i^R(n), X_i^G(n), X_i^B(n)\right], n = 1, ..., N\right\} \tag{8}$$

and let B_i be the background model for the ith pixel. B_i consists of a collection of local models and is represented as

$$B_i = \{\mathbf{B}_i^1, \mathbf{B}_i^2, ..., \mathbf{B}_i^{J_i}\}, \tag{9}$$

where $\mathbf{B}_i^j$ is the j th background local model, and J_i is the number of local models of B_i. For the i th pixel, we classify the background pixels X_i into J_i disjoint classes and compute the local models $\mathbf{B}_i^j$ $(\mathrm{j}=1,...,J_i)$. For the ease of explanation, X_i^j is the set of background pixels that belong to $\mathbf{B}_i^j$ among X_i, and $\left|X_i^j\right|$ is the size of set X_i^j. Thus,

$$\bigcup_{j=1}^{J_i} X_i^j = X_i \tag{10}$$

and

$$\sum_{j=1}^{J_i}\left|X_i^j\right| = N. \tag{11}$$

Each local model $\mathbf{B}_i^j$ of B_i is defined as

$$\mathbf{B}_i^j = \left[\mathbf{E}_i^j, \mathbf{S}_i^j, \mathbf{G}_i^j, \nu_{bi}^j, \nu_{ci}^j, \nu_{ei}^j, w_i^j\right], \tag{12}$$

where $\mathbf{E}_i^j$ is the expected color value of X_i^j, $\mathbf{S}_i^j$ is the standard deviation of X_i^j, and $\mathbf{G}_i^j$ is the background edge obtained by applying the edge operation to $\mathbf{E}_i^j$. These variables can be computed as

$$\mathbf{E}_i^j = \frac{1}{\left|X_i^j\right|}\left[\sum_{n\in X_i^j} X_i^R(n), \sum_{n\in X_i^j} X_i^G(n), \sum_{n\in X_i^j} X_i^B(n)\right] \tag{13}$$

$$\mathbf{S}_i^j = \sqrt{\frac{1}{\left|X_i^j\right|}\sum_{n\in X_i^j}\left(\mathbf{X}_i(n) - \mathbf{E}_i^j\right)^2}. \tag{14}$$

In these equations, ν_{bi}^j, ν_{ci}^j, and ν_{ei}^j are the variations in the brightness, chromaticity, and edge distortions of X_i^j, respectively, and are computed using

$$\nu_{bi}^j = RMS\left(\delta_{bi}^j\right) = \sqrt{\frac{1}{\left|X_i^j\right|}\sum_{n\in X_i^j}\left(\delta_{bi}^j(n) - 1\right)^2} \tag{15}$$

$$\nu_{ci}^j = RMS\left(\delta_{ci}^j\right) = \sqrt{\frac{1}{\left|X_i^j\right|}\sum_{n\in X_i^j}\delta_{ci}^j(n)^2} \tag{16}$$

$$\nu_{ei}^j = RMS\left(\delta_{ei}^j\right) = \sqrt{\frac{1}{\left|X_i^j\right|}\sum_{n\in X_i^j}\delta_{ei}^j(n)^2}, \tag{17}$$

where $\delta_{bi}^j(n)$, $\delta_{ci}^j(n)$, and $\delta_{ei}^j(n)$ are the brightness, chromaticity, and edge distortions between the j th local model and the n th frame background pixel, respectively. Since different pixels have different distributions of δ_{bi}^j, δ_{ci}^j, and δ_{ei}^j, we normalize them according to ν_{bi}^j, ν_{ci}^j, and ν_{ei}^j, respectively, as follows:

$$\overline{\delta_{bi}^j} = \frac{\delta_{bi}^j - 1}{\nu_{bi}^j} \tag{18}$$

$$\overline{\delta_{ci}^j} = \frac{\delta_{ci}^j}{\nu_{ci}^j} \tag{19}$$

$$\overline{\delta_{ei}^{j}} = \frac{\delta_{ei}^{j}}{v_{ei}^{j}}. \tag{20}$$

We then apply the same thresholds to all pixels to determine the foreground. The last parameter w_i^j is defined as a weight assigned to the local model $\mathbf{B}_i^j$. Since the X_i^j $(j = 1,..., J_i)$ sets have different numbers of background pixels, each local model $\mathbf{B}_i^j$ associated with X_i^j has its own weight defined as

$$w_i^j = \frac{\left|X_i^j\right|}{\sum_{j=1}^{J_i}\left|X_i^j\right|}, \tag{21}$$

and

$$\sum_{j=1}^{J_i} w_i^j = 1. \tag{22}$$

Table 1 shows the algorithm for the construction of B_i from X_i.

Table 1. An implementation of the background model

procedure background modeling
initialize $J_i := 1$ and $B_i := \{\}$
for $n = 1$ to N (the number of training background images)
begin
$j_n :=$ arg $\min_{j=1}^{J_i}\left(\left\lvert\delta_{bi}^{j}(n)\right\rvert + \left\lvert\delta_{ci}^{j}(n)\right\rvert\right)$ (23)
if $\left\lvert\delta_{bi}^{j_n}(n)\right\rvert + \left\lvert\delta_{ci}^{j_n}(n)\right\rvert < \varepsilon$ then update $\mathbf{B}_i^{j_n}$.
- $\mathbf{E}_i^{j_n} = \left(\frac{\left\lvert X_i^{j_n}\right\rvert}{\left\lvert X_i^{j_n}\right\rvert + 1}\right)\mathbf{E}_i^{j_n} + \left(\frac{1}{\left\lvert X_i^{j_n}\right\rvert + 1}\right)\mathbf{X}_i(n)$ (24)
- $\mathbf{S}_i^{j_n} = \sqrt{\left(\frac{\left\lvert X_i^{j_n}\right\rvert}{\left\lvert X_i^{j_n}\right\rvert + 1}\right)\mathbf{S}_i^{j_n\,2} + \left(\frac{1}{\left\lvert X_i^{j_n}\right\rvert + 1}\right)\left(\mathbf{X}_i(n) - \mathbf{E}_i^{j_n}\right)^2}$ (25)
- $X_i^{j_n} := X_i^{j_n} \cup \{\mathbf{X}_i(n)\}$ (26)
else $J_i = J_i + 1$ and create a new $\mathbf{B}_i^{J_i}$.
- $\mathbf{E}_i^{J_i} = \left[X_i^R(n), X_i^G(n), X_i^B(n)\right]$ (27)
- $\mathbf{S}_i^{J_i} = [0, 0, 0]$ (28)
- $X_i^{J_i} := \{\mathbf{X}_i(n)\}$ (29)
- $B_i := B_i \cup \{\mathbf{B}_i^{J_i}\}$ (30)
end
for $j = 1$ to J_i
begin
compute the $\mathbf{G}_i^j$, v_{bi}^j, v_{ci}^j, v_{ei}^j, and w_i^j using $\mathbf{E}_i^j$, $\mathbf{S}_i^j$, and $\left\lvert X_i^j\right\rvert$.
End

In **Table 1**, the nth frame background pixel $\mathbf{X}_i(n)$ is compared with J_i local models $\mathbf{B}_i^j$ $(j=1,\ldots,J_i)$ in terms of brightness and chromaticity distortions, and the fittest local model $\mathbf{B}_i^{j_n}$ and the associated set $X_i^{j_n}$ are selected. If the nth frame background pixel $\mathbf{X}_i(n)$ is sufficiently close to the local model $\mathbf{B}_i^{j_n}$, we update $\mathbf{B}_i^{j_n}$ and $X_i^{j_n}$ using (24), (25), and (26). After the training phase, we delete some local model $\mathbf{B}_i^j$ whose weights are less than epsilon since local models with low weights are likely to model camera jitters or other outliers instead of the background pixels.

3. Background Subtraction

3.1 Simple Background Subtraction

As mentioned above, using only the color information can yield a false negative result in background subtraction. In this subsection, we combine the color and edge information to detect the foreground. First, we represent the difference between the background model and the current image in terms of color and edge distortions. We normalize δ_{bi}^j, δ_{ci}^j, and δ_{ei}^j according to v_{bi}^j, v_{ci}^j, and v_{ei}^j, respectively, and determine the threshold for each distortion. Using the normalized distortions, we assign a binary label from the set {foreground (***F***), background (***B***)} to each pixel as follows:

$$x_i = \begin{cases} \boldsymbol{F} & if\ \overline{\delta_{ci}^j} > \tau_{CD}\ or\ \overline{\delta_{bi}^j} < \tau_{BD}\ or\ \overline{\delta_{ei}^j} > \tau_{ED},\ else \\ \boldsymbol{B} & \text{otherwise} \end{cases}, \tag{31}$$

where τ_{CD}, τ_{ED}, and τ_{BD} are threshold values.

In this paper, we employ the probabilistic approach to deal with the uncertainties of the color and edge information and to improve the robustness of the background subtraction. In this section, we compute the posterior probability $p\left(x_i = \boldsymbol{F} \mid \text{brightness, chromaticity, and edge of the curret frame}\right)$. For the normalized brightness distortion, if $\overline{\delta_{bi}^j}$ is higher than the upper threshold τ_{BD}^{high}, then the probability that the ith pixel belongs to the foreground is 0. If $\overline{\delta_{bi}^j}$ is lower than the lower threshold τ_{BD}^{low}, then the probability is 1. For values between τ_{BD}^{high} and τ_{BD}^{low}, the probability is scaled linearly between 0 and 1. In summary,

$$p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{bi}^j}\right) = \begin{cases} 0 & if\ \overline{\delta_{bi}^j} > \tau_{BD}^{high} \\ \dfrac{\overline{\delta_{bi}^j} - \tau_{BD}^{high}}{\tau_{BD}^{low} - \tau_{BD}^{high}} & if\ \tau_{BD}^{low} < \overline{\delta_{bi}^j} < \tau_{BD}^{high} \\ 1 & if\ \overline{\delta_{bi}^j} < \tau_{BD}^{low} \end{cases}. \tag{32}$$

Similarly, for the normalized chromaticity distortion $\overline{\delta_{ci}^j}$ and the normalized edge distortion $\overline{\delta_{ei}^j}$, the probability that the ith pixel belongs to the foreground is computed as

$$p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{ci}^{j}}\right) = \begin{cases} 0 & if \ \ \overline{\delta_{ci}^{j}} < \tau_{CD}^{low} \\ \dfrac{\overline{\delta_{ci}^{j}} - \tau_{CD}^{low}}{\tau_{CD}^{high} - \tau_{CD}^{low}} & if \ \ \tau_{CD}^{low} < \overline{\delta_{ci}^{j}} < \tau_{CD}^{high} \\ 1 & if \ \ \overline{\delta_{ci}^{j}} > \tau_{CD}^{high} \end{cases} \tag{33}$$

and

$$p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{ei}^{j}}\right) = \begin{cases} 0 & if \ \ \overline{\delta_{ei}^{j}} < \tau_{ED}^{low} \\ \dfrac{\overline{\delta_{ei}^{j}} - \tau_{ED}^{low}}{\tau_{ED}^{high} - \tau_{ED}^{low}} & if \ \ \tau_{ED}^{low} < \overline{\delta_{ei}^{j}} < \tau_{ED}^{high} \\ 1 & if \ \ \overline{\delta_{ei}^{j}} > \tau_{ED}^{high} \end{cases}, \tag{34}$$

respectively, where τ_{CD}^{high} and τ_{CD}^{low} are upper and lower thresholds of chromaticity distortion, and τ_{ED}^{high} and τ_{ED}^{low} are upper and lower thresholds of edge distortion, respectively. $p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right)$, $p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right)$, and $p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right)$ are computed according to the weighted sum of $p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{bi}^{j}}\right)$, $p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{ci}^{j}}\right)$, and $p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{ei}^{j}}\right)$, respectively, as

$$\begin{aligned} p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) &= \sum_{j=1}^{J_i} w_j \, p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{bi}^{j}}\right) \\ p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right) &= \sum_{j=1}^{J_i} w_j \, p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{ci}^{j}}\right) \\ p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right) &= \sum_{j=1}^{J_i} w_j \, p\left(x_i = \boldsymbol{F} \mid \overline{\delta_{ei}^{j}}\right), \end{aligned} \tag{35}$$

where $\overline{\boldsymbol{\delta}_{\mathbf{b}i}} = \left[\overline{\delta_{bi}^{1}}, \overline{\delta_{bi}^{2}}, \ldots, \overline{\delta_{bi}^{J_i}}\right]$, $\overline{\boldsymbol{\delta}_{\mathbf{c}i}} = \left[\overline{\delta_{ci}^{1}}, \overline{\delta_{ci}^{2}}, \ldots, \overline{\delta_{ci}^{J_i}}\right]$, and $\overline{\boldsymbol{\delta}_{\mathbf{e}i}} = \left[\overline{\delta_{ei}^{1}}, \overline{\delta_{ei}^{2}}, \ldots, \overline{\delta_{ei}^{J_i}}\right]$. We can then obtain the probability $p\left(x_i = \boldsymbol{F} \mid \left[\overline{\boldsymbol{\delta}_{\mathbf{b}i}}, \overline{\boldsymbol{\delta}_{\mathbf{c}i}}, \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right]\right)$ that the i th pixel belongs to the foreground by combining $p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right)$, $p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right)$, and $p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right)$ through the algebraic s-norm [29]

$$\begin{aligned} & p\left(x_i = \boldsymbol{F} \mid \left[\overline{\boldsymbol{\delta}_{\mathbf{b}i}}, \overline{\boldsymbol{\delta}_{\mathbf{c}i}}, \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right]\right) \\ &= s\left[p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right), p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right), p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right)\right] \\ &= s\left[s\left[p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right), p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right)\right], p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right)\right] \\ &= s\left[p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) + p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right) - p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right), p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right)\right] \\ &= p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) + p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right) + p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right) \\ & - p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right) - p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right) p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right) \\ & - p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right) p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) + p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{b}i}}\right) p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{c}i}}\right) p\left(x_i = \boldsymbol{F} \mid \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right). \end{aligned} \tag{36}$$

Based on the derived probability $p\left(x_i = \boldsymbol{F} \mid \mathbf{z}_i = \left[\overline{\boldsymbol{\delta}_{\mathbf{b}i}}, \overline{\boldsymbol{\delta}_{\mathbf{c}i}}, \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right]\right)$, we design a simple binary classifier

$$\delta_{sim}\left(\mathbf{z}_i\right) = x_i = \begin{cases} \boldsymbol{F} & \text{if} \quad p\left(x_i = \boldsymbol{F} \mid \mathbf{z}_i = \left[\overline{\boldsymbol{\delta}_{\mathbf{b}i}}, \overline{\boldsymbol{\delta}_{\mathbf{c}i}}, \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right]\right) > \tau_{\mathrm{th}} \\ \boldsymbol{B} & \text{if} \quad p\left(x_i = \boldsymbol{F} \mid \mathbf{z}_i = \left[\overline{\boldsymbol{\delta}_{\mathbf{b}i}}, \overline{\boldsymbol{\delta}_{\mathbf{c}i}}, \overline{\boldsymbol{\delta}_{\mathbf{e}i}}\right]\right) < \tau_{\mathrm{th}} \end{cases}, \tag{37}$$

where τ_{th} is a threshold value. This equation (37) combines the color and edge information to detect the foreground. In (37), the decision is based on the distortions in the current image frame. However, the distortions of the past images are as important as those of the current image because

1) abrupt changes of state (***F*** or ***B***) between consecutive frames are likely to be noise and
2) past image frames provide information about the probabilistic nature of each pixel.

In the next subsection, we propose a new probabilistic approach that utilizes not only the current image, but also all past images in the decision procedure.

3.2 Probabilistic Background Subtraction

Let $\mathbf{z}_i^t$ denote the normalized distortion vector of the ith pixel in the tth frame, which is defined as

$$\mathbf{z}_i^t = \left[\overline{\boldsymbol{\delta}_{\mathbf{b}i}^t}, \overline{\boldsymbol{\delta}_{\mathbf{c}i}^t}, \overline{\boldsymbol{\delta}_{\mathbf{e}i}^t}\right], \tag{38}$$

where $\overline{\boldsymbol{\delta}_{\mathbf{b}i}^t}$, $\overline{\boldsymbol{\delta}_{\mathbf{c}i}^t}$, and $\overline{\boldsymbol{\delta}_{\mathbf{e}i}^t}$ are the normalized brightness, chromaticity, and edge distortion vectors of the ith pixel in the tth frame, respectively. Furthermore, let $\mathbf{z}_i^{1:t}$ represent the cumulative sequence of the measurements from 1 to t

$$\mathbf{z}_i^{1:t} = \left[\mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^t\right]. \tag{39}$$

We define *belief* as the posterior probability of x_i conditioned on current and past distortions. That is, the belief is defined as

$$bel\left(x_i^t\right) = p\left(x_i^t \mid \mathbf{z}_i^{1:t}\right). \tag{40}$$

Here, x_i^t is the state (class) variable of the ith pixel in the tth frame, and it can take on one of two values: foreground (***F***) or background (***B***). $bel\left(x_i^t = \boldsymbol{F}\right)$ is computed as

$$bel\left(x_i^t = \boldsymbol{F}\right) = \frac{\rho_i^t}{1+\rho_i^t}. \tag{41}$$

The method used to compute ρ_i^t is given in the Appendix. Therefore, considering all past and current distortions, the class of the ith pixel is determined according to

$$\delta_{prob}\left(\mathbf{z}_i^{1:t}\right) = x_i^t = \begin{cases} \boldsymbol{F} & \text{if} \quad bel\left(x_i^t = \boldsymbol{F}\right) > \zeta_{\mathrm{th}} \\ \boldsymbol{B} & \text{if} \quad bel\left(x_i^t = \boldsymbol{F}\right) < \zeta_{\mathrm{th}} \end{cases}, \tag{42}$$

where δ_{prob} is a binary classifier and ζ_{th} is a threshold that balances the trade-off between sensitivity to foreground motion and robustness against outliers.

3.3 Update the Background Model using the Current Image

A background scene can be changed in the detection phase. For example, the illumination can change over time, new objects can enter the background, or one of the background objects can exit the scene. All of these changes can alter the background appearance. For these reasons, background subtraction is known as a powerful preprocessing step only in a controlled indoor environment [30]. To overcome these changes, the background model can be updated during run-time. We also update the background model in our proposed algorithm. If the i th pixel is classified as background, we update the local model using color value distortions between the current image and background model. First, using the brightness and chromatocity distortions, we select the best matched model $\mathbf{B}_i^{j_i} = \left[\mathbf{E}_i^{j_i}, \mathbf{S}_i^{j_i}, \mathbf{G}_i^{j_i}, \nu_{bi}^{j_i}, \nu_{ci}^{j_i}, \nu_{ei}^{j_i}, w_i^{j_i}\right]$ among the J_i local models using $j_i := \arg\min_{j=1}^{J_i}\left(\left|\overline{\delta_{bi}^{j}}\right| + \left|\overline{\delta_{ci}^{j}}\right|\right)$. The new match model $\tilde{\mathbf{B}}_i^{j_i} = \left[\tilde{\mathbf{E}}_i^{j_i}, \tilde{\mathbf{S}}_i^{j_i}, \tilde{\mathbf{G}}_i^{j_i}, \tilde{\nu}_{bi}^{j_i}, \tilde{\nu}_{ci}^{j_i}, \tilde{\nu}_{ei}^{j_i}, \tilde{w}_i^{j_i}\right]$ is then computed as follows:

$$\tilde{\mathbf{E}}_i^{j_i} = \left(\frac{\left|X_i^{j_i}\right|}{\left|X_i^{j_i}\right|+1}\right)\mathbf{E}_i^{j_i} + \left(\frac{1}{\left|X_i^{j_i}\right|+1}\right)\mathbf{I}_i \tag{43}$$

$$\tilde{\mathbf{S}}_i^{j_i} = \sqrt{\left(\frac{\left|X_i^{j_i}\right|}{\left|X_i^{j_i}\right|+1}\right)\mathbf{S}_i^{j_i\,2} + \left(\frac{1}{\left|X_i^{j_i}\right|+1}\right)\left(\mathbf{I}_i - \tilde{\mathbf{E}}_i^{j_i}\right)^2}. \tag{44}$$

$\tilde{\mathbf{G}}_i^{j_i}$ is obtained by applying the edge operation to $\tilde{\mathbf{E}}_i^{j_i}$. According to (15), (16), (17), and (21), $\tilde{\nu}_{bi}^{j_i}$, $\tilde{\nu}_{ci}^{j_i}$, $\tilde{\nu}_{ei}^{j_i}$, and $\tilde{w}_i^{j_i}$ are updated to

$$\tilde{\nu}_{bi}^{j_i} = \sqrt{\frac{1}{\left|X_i^{j}\right|+1}\left(\sum_{n\in X_i^{j_n}}\left(\delta_{bi}^{j}(n)-1\right)^2 + \left(\delta_{bi}^{j_n}-1\right)^2\right)} = \sqrt{\left(\frac{\left|X_i^{j_i}\right|}{\left|X_i^{j_i}\right|+1}\right)\nu_{bi}^{j_i\,2} + \left(\frac{1}{\left|X_i^{j_i}\right|+1}\right)\left(\delta_{bi}^{j_i}-1\right)^2} \tag{45}$$

$$\tilde{\nu}_{ci}^{j_i} = \sqrt{\frac{1}{\left|X_i^{j}\right|+1}\left(\sum_{n\in X_i^{j_n}}\delta_{ci}^{j}(n)^2 + \delta_{ci}^{j_n\,2}\right)} = \sqrt{\left(\frac{\left|X_i^{j_i}\right|}{\left|X_i^{j_i}\right|+1}\right)\nu_{ci}^{j_i\,2} + \left(\frac{1}{\left|X_i^{j_i}\right|+1}\right)\delta_{ci}^{j_i\,2}} \tag{46}$$

$$\tilde{\nu}_{ei}^{j_i} = \sqrt{\frac{1}{\left|X_i^{j}\right|+1}\left(\sum_{n\in X_i^{j_n}}\delta_{ei}^{j}(n)^2 + \delta_{ei}^{j_n\,2}\right)} = \sqrt{\left(\frac{\left|X_i^{j_i}\right|}{\left|X_i^{j_i}\right|+1}\right)\nu_{ei}^{j_i\,2} + \left(\frac{1}{\left|X_i^{j_i}\right|+1}\right)\delta_{ei}^{j_i\,2}}, \tag{47}$$

$$\tilde{w}_i^{j_i} = \frac{\left|X_i^{j_i}\right|+1}{\sum_{j=1}^{J_i}\left|X_i^{j}\right|+1} \text{ and } w_i^{\forall j\neq j_i} = \frac{\left|X_i^{j}\right|}{\sum_{j=1}^{J_i}\left|X_i^{j}\right|+1} \tag{48}$$

$$X_i^{j_i} := X_i^{j_i} \cup \left\{\mathbf{I}_i\right\}. \tag{49}$$

In these equations, $\delta_{ei}^{j_i}$, $\delta_{ei}^{j_i}$, and $\delta_{ei}^{j_i}$ are the distortions between the current image and $\mathbf{B}_i^{j_i}$. Local models with a small number of pixels are assigned small weights. We delete the local model, the weights of which fall below a certain threshold, for computational efficiency. Furthermore, if a new object enters the environment and remains stationary so that a pixel continues to be classified as foreground for a long time, then we reestablish a new background local model for the pixels. More specifically, for the i th pixel, we store the sequence of classes (***F*** or ***B***) as

$$\beta_i = \left\{\beta_i^t, \beta_i^{t-1}, \beta_i^{t-2}, \ldots, \beta_i^{t-p+1}\right\} \tag{50}$$

for the previous p time frames. If

$$n(\beta_i) > (1-\varepsilon)p, \tag{51}$$

then we reconstruct a new local background model in the same way as in the modeling phase, where β_i^{t-j} is the class of the i th pixel in j previous frames, and $n(\cdot)$ is the number of pixels classified as foreground.

4. Experiments

4.1 Application to a Simple Synthetic Pixel

To show the effectiveness of the proposed algorithm, it is applied to the identification problem in a single-pixel. Assume that the state of the pixel changes as shown in **Fig. 3**. In the figure, the state 1 indicates that the pixel belongs to the foreground, and the state 0 indicates that the pixel belongs to the background. The pixel is assumed to be corrupted by noise such as illumination change, and the posterior $p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$ is not the same as in **Fig. 3**. The posterior deviates from the true state and is assumed to be contaminated as shown in **Fig. 4**.

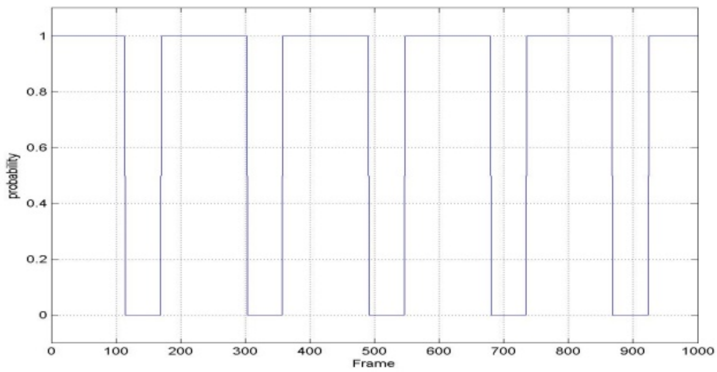


Fig. 3. The true state of the *i*th pixel

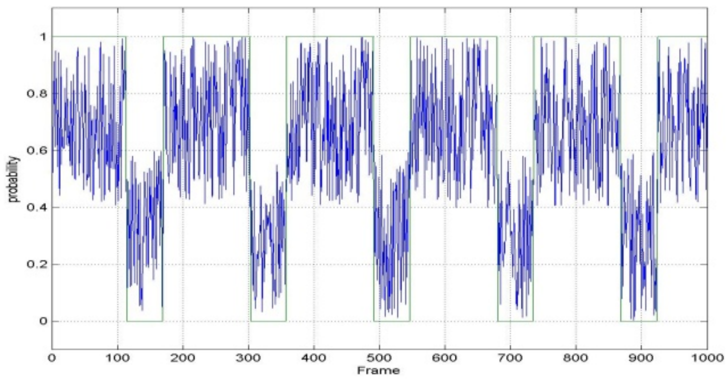


Fig. 4. True state and $p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$

The posterior $p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$ is assumed to be

$$p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right) = \begin{cases} 1 - U\left(0, 0.6\right) & if \;\; x_i^t \;\; is\; a \;\; foreground \\ 0 + U\left(0, 0.6\right) & if \;\; x_i^t \;\; is\; a\; background \end{cases}, \tag{52}$$

where $U\left(a, b\right)$ is the disturbance from the uniform distribution over the interval $\left[a, b\right]$. If we use the simple binary classifier δ_{sim} in (37) with $\tau_{th} = 0.5$ and the current measurement $\mathbf{z}_i^t$, we obtain the decision as shown in **Fig. 5**.

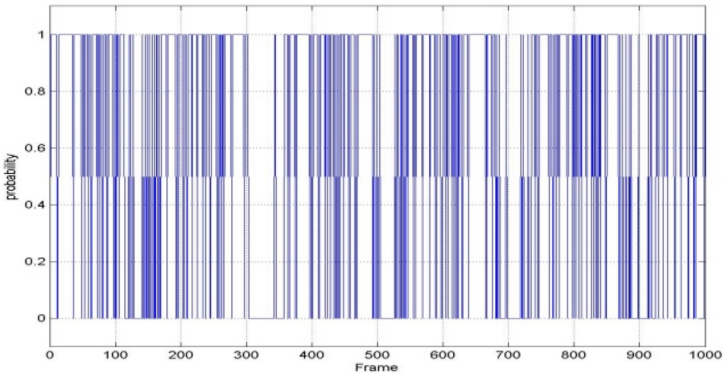


Fig. 5. The estimation of the simple binary classifier δ_{sim}

That is, if $p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$ is greater than $\tau_{th} = 0.5$, we conclude that the pixel belongs to the foreground, and if $p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$ is less than $\tau_{th} = 0.5$, we conclude that the pixel belongs to the background. If we compare **Figs. 3** and **5**, we can see that the decision made by the simple binary classifier δ_{sim} based on $p(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t)$ is highly sensitive to measurement noise. Now, we apply the probabilistic binary classifier δ_{prob} from (42). The parameters used in this simulation are given in **Table 2**.

Table 2. The system parameters and thresholds

System Parameters	π_{11}	0.8	π_{12}	0.2
	π_{21}	0.2	π_{22}	0.8
	$P\left(x_i^t = \boldsymbol{F}\right)$	0.5		
Thresholds	τ_{th}	0.5	ζ_{th}	0.5

Fig. 6 illustrates the superimposition of $bel\left(x_i^t = \boldsymbol{F}\right)$ onto the actual state.

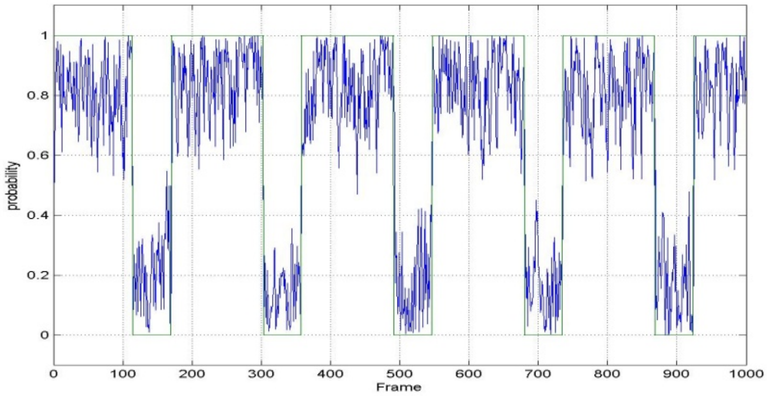


Fig. 6. True state and $bel\left(x_i^t = \boldsymbol{F}\right)$

If we use the probabilistic binary classifier δ_{prob} with $\zeta_{th} = 0.5$ based on $bel\left(x_i^t = \boldsymbol{F}\right)$, we obtain the decision shown in **Fig. 7**.

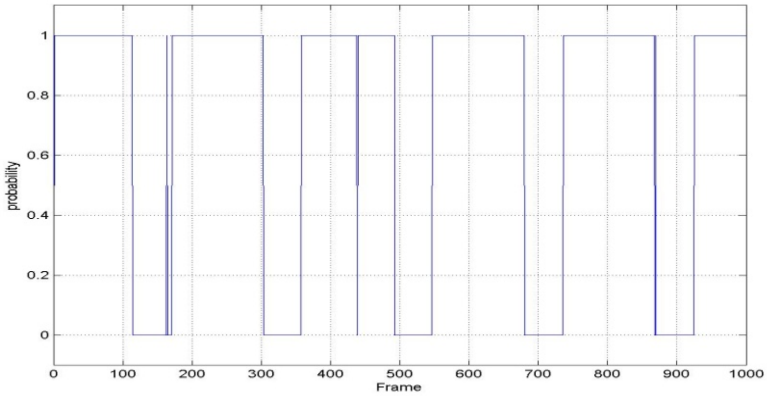


Fig. 7. The estimation of the probabilistic binary classifier δ_{prob}

Based on **Figs. 5** and **7**, the probabilistic classifier δ_{prob} demonstrates more robust background subtraction than does the simple method. The estimation performances of the two methods are summarized in **Table 3**, which illustrates that the probabilistic method outperforms the simple method. In the experiment, we use the general posterior $p\left(x_i^t = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$. Therefore, we can also expect better results due to the combination of our probabilistic framework not only with our approach, but also with other background subtraction methods.

Table 3. The estimation rates in computer simulation (%)

	Simple binary classifier	Probabilistic binary classifier
Accuracy	83.2	99.1
Error rate	16.8	0.9

4.2 Real Video Sequence

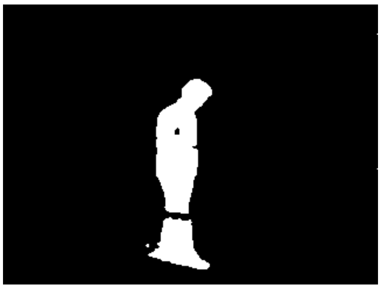
In this subsection, the proposed methods are applied to a real video stream. The video sequences are recorded by a stationary camera in image sizes of 320x240. For thresholds in (32), (33), and (34), the distortion histograms are constructed in the background modeling phase [9]. The thresholds are distortion values at the δ_p and $1-\delta_p$ detection rates in the histogram, respectively, where δ_p is the desired detection rate. We use the same system parameters and decision thresholds as in the previous simulation, which are given in **Table 2**. We set up the experimentation room through which a subject moves. In the recorded video sequence, the subject both obscures the background and casts shadows on the floor and wall during movement. Examples of the background scene and the current frame are shown in **Fig. 8-(a)** and **(b)**, respectively.

(a) background scene

(b) current image including the foreground

Fig. 8. The background scene and the current frame

We apply the simple fusion classifier δ_{sim} and the probabilistic fusion classifier δ_{prob} to these images. **Fig. 9-(a)** and **(b)** show the background subtraction results of the two methods.



(a) Result of δ_{sim}

(b) Result of δ_{prob}

Fig. 9. The results of the proposed background subtraction method

Since the performance of the video-based recogntion system relies on the quality of the background subtraction, it requires high quality foreground images. Therefore, δ_{prob} demonstrates a more efficient subtraction result than does δ_{sim}.

4.3 Benchmark problems

In this subsection, the proposed algorithm is applied to benchmark background subtraction problems. Seven benchmark problems with different characteristics were introduced in [26], and several background subtraction methods were applied to the problems for comparison.

The test sets of the benchmark problems have one of the following:

- ***Moved objects***: Background objects can be moved. These objects should be considered as background, not foreground.
- ***Time of day***: Gradual illumination variation changes the background appearance.
- ***Light changes***: Sudden changes in illumination and other scene parameters change the background appearance.
- ***Waving trees***: Background objects can be vacillated by the wind.
- ***Camouflage***: Characteristics of foreground objects may be subsumed by the background model.
- ***Bootstrapping***: In some environments, a clean background scene that does not include the foreground cannot be obtained.
- ***Foreground aperture***: When a homogeneously colored object moves, a change in the interior pixels cannot be detected.

We applied the proposed method to the test sets of the benchmark problems given in [26]. **Fig. 10** shows the background subtraction results of the proposed method. The third column is the ideal result included in the benchmark test set, and the fourth column is the result of the proposed method. While the proposed method performs well for moved objects, time of day, waving trees, and foreground aperture, it missed a relatively high number of foreground and background pixels suffering from light switch, bootstrapping, and camouflage.

Table 4. Performance comparison of the proposed method and previous methods

	Proposed method		Kim *et al.* [32]		Toyama *et al.* [26]	
	FP	FN	FP	FN	FP	FN
Moved object	0	0	1423	0	0	0
Time of day	484	178	51	255	25	961
Light switch	1227	395	201	2059	375	947
Waving trees	1139	47	238	11	1999	877
Camouflage	1042	1133	17	1475	2706	229
Bootstrapping	994	678	178	1467	365	2025
Foreground aperture	894	423	520	2223	649	320
Total errors	8634		10116		11478	

We evaluate each result in terms of the numbers of false positives (FP) and false negatives (FN), where FP denotes the number of background pixels incorrectly classified as foreground, and FN is the number of foreground pixels incorrectly classified as background. **Table 4** shows that the proposed method outperforms the competing methods. Furthermore, the precision and recall [33] of the proposed method are reported in **Table 5**.

Table 5. Precision and recall of the proposed method

	Precision	Recall
Moved object	Not a Number (NaN)	0
Time of day	0.7389	0.0739
Light switch	0.6997	0.1626
Waving trees	0.8367	0.3239
Camouflage	0.9005	0.5540
Bootstrapping	0.6932	0.1281
Foreground aperture	0.8371	0.2568

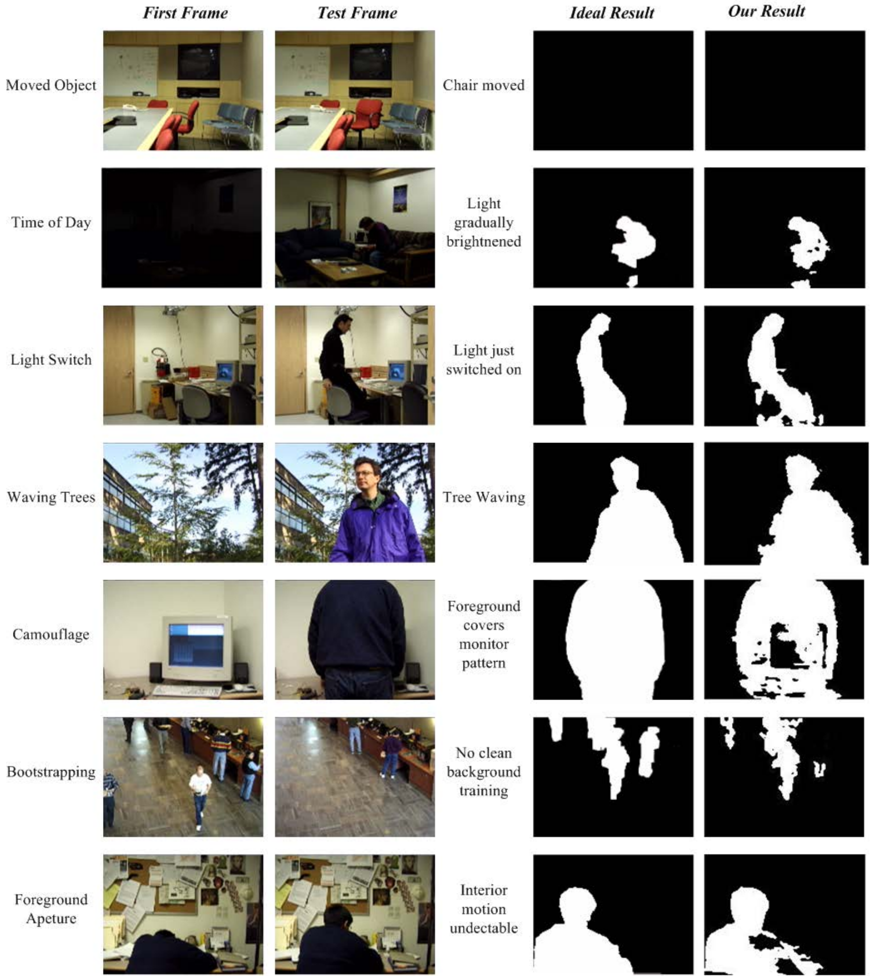


Fig. 10. Tests of the proposed algorithm

Table 4 compares the results of the proposed method with those of previous methods [26][32], where we directly report their performance without reimplementation.

For the real-time system, we employ a speed-up technique as in [9]. Since our background is modeled per pixel, the proposed method can be parallelized by dividing the images into segments and performing the operations independently on each segment. **Table 6** shows the processing speed of the suggested method.

Table 6. The processing speed (frames/sec)

Image size	Modeling phase	Detecting Phase
160x120	40.77	28.94
240x180	18.08	12.14
320x240	10.01	6.66

We used the OpenCV library [31] on a standard PC with a Pentium 2.4GHz Quad core processor. The result shows that the proposed algorithm is sufficiently efficient for real-time processing.

5. Conclusions

In this paper, we presented a new probabilistic background subtraction algorithm that automatically extracts the foreground from a video stream. This method can be used as an essential step toward applications such as video surveillance, traffic monitoring, gesture recognition, human detection and tracking for video teleconferencing, and human-computer interaction. To effectively utilize the information from past image sequences and to improve the quality and reliability of the background subtraction, the proposed method employs the belief that reflects not only the current image but also all of the earlier images. Furthermore, we present an efficient fusion method for the color and edge information in order to overcome some of the drawbacks of the existing methods. The validity of the suggested method was demonstrated via experimentation. The suggested method showed more robust background subtraction than did the previous method, even in the presence of a relatively high degree of noise. Although we employ some speed-up techniques for the real-time system, the processing speed of the proposed method is relatively slow compared with that of the current background subtraction method. In future work, the speed of the system should be increased. Furthermore, we should provide the ROC curve to analyze the changes in performance according to the thresholds.

Appendix

We can rewrite the belief $bel\left(x_i^t\right)$ as

$$p\left(x_i^t \mid \mathbf{z}_i^{1:t}\right)=\frac{p\left(\mathbf{z}_i^t \mid x_i^t,\mathbf{z}_i^{1:t-1}\right)p\left(x_i^t \mid \mathbf{z}_i^{1:t-1}\right)}{p\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{1:t-1}\right)}=\frac{p\left(\mathbf{z}_i^t \mid x_i^t\right)p\left(x_i^t \mid \mathbf{z}_i^{1:t-1}\right)}{P\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{1:t-1}\right)}. \tag{53}$$

Applying the Bayes rule

$$p\left(\mathbf{z}_i^t \mid x_i^t\right)=\frac{p\left(x_i^t \mid \mathbf{z}_i^t\right)p\left(\mathbf{z}_i^t\right)}{p\left(x_i^t\right)} \tag{54}$$

to (53) yields

$$p\left(x_i^t \mid \mathbf{z}_i^{1:t}\right)=\frac{p\left(x_i^t \mid \mathbf{z}_i^t\right)p\left(\mathbf{z}_i^t\right)}{p\left(x_i^t\right)}\frac{p\left(x_i^t \mid \mathbf{z}_i^{1:t-1}\right)}{P\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{1:t-1}\right)}. \tag{55}$$

Therefore, the belief over the state x_i^t in the tth frame is represented as

$$p\left(x_i^t=\boldsymbol{F} \mid \mathbf{z}_i^{1:t}\right)=\frac{p\left(x_i^t=\boldsymbol{F} \mid \mathbf{z}_i^t\right)p\left(\mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{p\left(x_i^t=\boldsymbol{F} \mid \mathbf{z}_i^{1:t-1}\right)}{P\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{1:t-1}\right)} \tag{56a}$$

and

$$p\left(x_i^t=\boldsymbol{B} \mid \mathbf{z}_i^{1:t}\right)=\frac{p\left(x_i^t=\boldsymbol{B} \mid \mathbf{z}_i^t\right)p\left(\mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\right)}\frac{p(\left(x_i^t=\boldsymbol{B} \mid \mathbf{z}_i^{1:t-1}\right)}{P\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{1:t-1}\right)}. \tag{56b}$$

To remove the terms that are difficult to compute, we divide the two equations and obtain

$$\frac{bel\left(x_i^t=\boldsymbol{F}\right)}{bel\left(x_i^t=\boldsymbol{B}\right)}=\frac{\dfrac{p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^t\right)p\left(\mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\dfrac{p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)}{P\left(\mathbf{z}_i^t\mid \mathbf{z}_i^{1:t-1}\right)}}{\dfrac{p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^t\right)p\left(\mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\right)}\dfrac{p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)}{P\left(\mathbf{z}_i^t\mid \mathbf{z}_i^{1:t-1}\right)}}$$
$$=\frac{p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^t\right)}\frac{p\left(x_i^t=\boldsymbol{B}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)}{p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)}. \tag{57}$$

Applying the total probability theorem to (57)

$$\begin{aligned}&p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)\\&=p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{F},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)\\&+p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{B},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)\end{aligned} \tag{58a}$$

and

$$\begin{aligned}&p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)\\&=p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{F},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)\\&+p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{B},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)\end{aligned} \tag{58b}$$

yields

$$\begin{aligned}&\frac{bel\left(x_i^t=\boldsymbol{F}\right)}{bel\left(x_i^t=\boldsymbol{B}\right)}=\frac{p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^t\right)}\frac{p\left(x_i^t=\boldsymbol{B}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\\&\times\frac{p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{F},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)+p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{B},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)}{p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{F},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)+p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{B},\mathbf{z}_i^{1:t-1}\right)p\left(x_i^{t-1}=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)}.\end{aligned} \tag{59}$$

Under the Markov assumption, the current state x_i^t is influenced only by one time earlier state x_i^{t-1}, and (59) is simplified as

$$\begin{aligned}&\frac{bel\left(x_i^t=\boldsymbol{F}\right)}{bel\left(x_i^t=\boldsymbol{B}\right)}=\frac{p\left(x_i^t=\boldsymbol{F}\mid \mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\mid \mathbf{z}_i^t\right)}\frac{p\left(x_i^t=\boldsymbol{B}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\\&\times\frac{p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{F}\right)p\left(x_i^{t-1}=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)+p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{B}\right)p\left(x_i^{t-1}=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)}{p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{F}\right)p\left(x_i^{t-1}=\boldsymbol{F}\mid \mathbf{z}_i^{1:t-1}\right)+p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{B}\right)p\left(x_i^{t-1}=\boldsymbol{B}\mid \mathbf{z}_i^{1:t-1}\right)}.\end{aligned} \tag{60}$$

For simplicity, we assume that the state transition is static and the transition probability is constant. That is, the state transition is represented by four constants:

$$\begin{aligned}\pi_{11}&=p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{F}\right)\\\pi_{12}&=p\left(x_i^t=\boldsymbol{F}\mid x_i^{t-1}=\boldsymbol{B}\right)\\\pi_{21}&=p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{F}\right)\\\pi_{22}&=p\left(x_i^t=\boldsymbol{B}\mid x_i^{t-1}=\boldsymbol{B}\right),\end{aligned} \tag{61}$$

where $\pi_{11}+\pi_{21}=1$ and $\pi_{12}+\pi_{22}=1$. Substituting (61) into (60) yields

$$\begin{aligned}\frac{bel\left(x_i^t=\boldsymbol{F}\right)}{bel\left(x_i^t=\boldsymbol{B}\right)}&=\frac{p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\mid\mathbf{z}_i^t\right)}\frac{p\left(x_i^t=\boldsymbol{B}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{\pi_{11}p\left(x_i^{t-1}=\boldsymbol{F}\mid\mathbf{z}_i^{1:t-1}\right)+\pi_{12}p\left(x_i^{t-1}=\boldsymbol{B}\mid\mathbf{z}_i^{1:t-1}\right)}{\pi_{21}p\left(x_i^{t-1}=\boldsymbol{F}\mid\mathbf{z}_i^{1:t-1}\right)+\pi_{22}\left(x_i^{t-1}=\boldsymbol{B}\mid\mathbf{z}_i^{1:t-1}\right)}\\&=\frac{p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}{p\left(x_i^t=\boldsymbol{B}\mid\mathbf{z}_i^t\right)}\frac{p\left(x_i^t=\boldsymbol{B}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{12}bel\left(x_i^{t-1}=\boldsymbol{B}\right)}{\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}bel\left(x_i^{t-1}=\boldsymbol{B}\right)}.\end{aligned}\tag{62}$$

Next, consider two equations

$$\begin{aligned}&\left(\pi_{11}+\pi_{21}\right)\left\{bel\left(x_i^{t-1}=\boldsymbol{F}\right)+bel\left(x_i^{t-1}=\boldsymbol{B}\right)\right\}=1\\&\left(\pi_{12}+\pi_{22}\right)\left\{bel\left(x_i^{t-1}=\boldsymbol{F}\right)+bel\left(x_i^{t-1}=\boldsymbol{B}\right)\right\}=1.\end{aligned}\tag{63}$$

These two equations are clear from the definitions of π_{ij} and $bel\left(x_i^t\right)$. Adding the equations yields

$$\begin{aligned}&\left(\pi_{11}+\pi_{21}\right)\left\{bel\left(x_i^{t-1}=\boldsymbol{F}\right)+bel\left(x_i^{t-1}=\boldsymbol{B}\right)\right\}\\&+\left(\pi_{12}+\pi_{22}\right)\left\{bel\left(x_i^{t-1}=\boldsymbol{F}\right)+bel\left(x_i^{t-1}=\boldsymbol{B}\right)\right\}=2.\end{aligned}\tag{64}$$

By rearranging (64), we obtain

$$\begin{aligned}&\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}bel\left(x_i^{t-1}=\boldsymbol{B}\right)=\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)\\&=2-\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)-\left(\pi_{11}+\pi_{21}\right)bel\left(x_i^{t-1}=\boldsymbol{B}\right)-\pi_{12}bel\left(x_i^{t-1}=\boldsymbol{B}\right)-\left(\pi_{12}+\pi_{22}\right)bel\left(x_i^{t-1}=\boldsymbol{F}\right)\\&=2-\pi_{11}bel(x_i^{t-1}=\boldsymbol{F})-bel(x_i^{t-1}=\boldsymbol{B})-\pi_{12}bel(x_i^{t-1}=\boldsymbol{B})-bel(x_i^{t-1}=\boldsymbol{F})\\&=1-\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\left(1-bel\left(x_i^{t-1}=\boldsymbol{B}\right)\right)-\pi_{12}bel\left(x_i^{t-1}=\boldsymbol{B}\right)-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\\&=1-\left\{\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{12}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)\right\}.\end{aligned}\tag{65}$$

Similarly, we obtain

$$\begin{aligned}&\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{12}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)\\&=1-\left\{\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)\right\}.\end{aligned}\tag{66}$$

Applying (65) and (66) to (62) yields

$$\begin{aligned}\frac{bel\left(x_i^t=\boldsymbol{F}\right)}{1-bel\left(x_i^t=\boldsymbol{F}\right)}&=\frac{p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}{1-p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}\frac{1-p\left(x_i^t=\boldsymbol{F}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{12}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)}{\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)}\\&=\frac{p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}{1-p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}\frac{1-p\left(x_i^t=\boldsymbol{F}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{12}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)}{1-\left\{\pi_{11}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{12}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)\right\}}\\&=\frac{p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}{1-p\left(x_i^t=\boldsymbol{F}\mid\mathbf{z}_i^t\right)}\frac{1-p\left(x_i^t=\boldsymbol{F}\right)}{p\left(x_i^t=\boldsymbol{F}\right)}\frac{1-\left\{\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)\right\}}{\pi_{21}bel\left(x_i^{t-1}=\boldsymbol{F}\right)+\pi_{22}\left(1-bel\left(x_i^{t-1}=\boldsymbol{F}\right)\right)}\\&=\rho_i^t\end{aligned}\tag{67}$$

Here, $p\left(x_i^t=\boldsymbol{F}\right)$ and $p\left(x_i^t=\boldsymbol{B}\right)$ are prior probabilities of the foreground and the background, respectively. If we know the size ratios of the foreground and the background to the whole image, we can use them as $p\left(x_i^t = \boldsymbol{F}\right)$ and $p\left(x_i^t = \boldsymbol{B}\right)$. If we do not know these values in advance, we simply set $p\left(x_i^t = \boldsymbol{F}\right)$ and $p\left(x_i^t = \boldsymbol{B}\right)$ to 0.5. However, if we know the characteristics of the monitored image and more accurate probabilities, we can improve the performance. In (67), the belief $bel\left(x_i^t = \boldsymbol{F}\right)$ in the t th frame is computed from the one time earlier belief $bel\left(x_i^{t-1} = \boldsymbol{F}\right)$ and the current probability $p\left(x_i = \boldsymbol{F} \mid \mathbf{z}_i^t\right)$ derived in Section 2; the equation can then be implemented recursively. Furthermore, the first term in (67) refers to a short-term background model, and the third term in (67) refers to a long-term background model; these two models are combined in the belief form of the proposed method.

References

[1] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," in *Proc. IEEE*, vol. 90, pp. 1151-1163, Jul. 2002. Article (CrossRef Link)

[2] I. Hariaoglu, D. Harwood and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, Aug. 2000. Article (CrossRef Link)

[3] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proc. the 13th Conf. Uncertainty in Artificial intelligence*, 1997.

[4] S. Arseneau and J. R. Cooperstock, "Real-time image segmentation for action recognition," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1999. Article (CrossRef Link)

[5] H. Lee, S. Hong and E. Kim, "An efficient gait recognition based on a selective neural network ensemble," *International Journal of Imaging Systems and Technology*, vol. 18, no. 4, pp. 237-241, Oct. 2008. Article (CrossRef Link)

[6] H. Lee, S. Hong and E. Kim, "Neural network ensemble with probabilistic fusion and its application to gait recognition," *Neurocomputing*, vol. 72, pp. 1557-1564, Mar. 2009. Article (CrossRef Link)

[7] N. M. Oliver, B. Rosario and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831-843, Aug. 2000. Article (CrossRef Link)

[8] V. I. Pavlovic, R. Sharma and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677-695, Jul. 1997. Article (CrossRef Link)

[9] T. Horpraser, D. Harwood and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. International Conference on Computer Vision*, 1999.

[10] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-6, 2008. Article (CrossRef Link)

[11] A. Neri, S. Colonnese, G. Russo and P. Talone, "Automatic moving object and background separation," *Signal Processing*, vol. 66, no. 2, pp. 219-232, Apr. 1998. Article (CrossRef Link)

[12] J. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 761-774, Aug. 2006. Article (CrossRef Link)

[13] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp.246-252, 1999. Article (CrossRef Link)

[14] O. Javed, K. Shafique and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proc. the Workshop on Motion and Video Computing*, pp. 22-27, Dec. 2002. Article (CrossRef Link)
[15] D. Lee, J. Hull and B. Erol, "A Bayesian framework for Gaussian mixture background modeling," in *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 973-976, Sept. 2003. Article (CrossRef Link)
[16] F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *Proc. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 37-45, Mar. 2003.
[17] G. Dally, J. Migdal and W. Grimson, "Background subtraction for temporally irregular dynamic textures," in *Proc. IEEE Workshop on Applications of Computer Vision*, pp. 1-7, 2008. Article (CrossRef Link)
[18] Z. Zivkovic "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. IEEE International Conference on Pattern Recognition*, vol. 2, pp. 28-31, Aug. 2004. Article (CrossRef Link)
[19] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 302-309, 2004. Article (CrossRef Link)
[20] K. Kim, T. Chalidabhongse, D. Harwood and L.S. Davis, "Real-time foreground-background segmentation using codebook model," *Real-time imaging*, vol. 11, pp. 167-256, Jun. 2005. Article (CrossRef Link)
[21] X. Zhang and J. Yang, "The analysis of the color similarity problem in moving object detection," *Signal Processing*, vol. 89, no. 4, pp. 685-691, 2009. Article (CrossRef Link)
[22] M. Mason and Z. Duric, "Using histograms to detect and track objects in color video," in *Proc. Applied Imagery Pattern Recognition Workshop*, pp. 154-159, 2001. Article (CrossRef Link)
[23] S. Jabri, Z. Duric, H. Wechsler and A. Rosenfeld, "Detection and location of people in video images using adaptive fusion of color and edge information," in *Proc. International Conference on Pattern Recognition*, vol. 4, pp. 627-630, 2000. Article (CrossRef Link)
[24] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 774-780, Aug. 2000. Article (CrossRef Link)
[25] M. Heikkila and M. Pietikainen "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 657-662, Apr. 2006. Article (CrossRef Link)
[26] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. IEEE International Conference on Computer Vision*, vol.1, pp. 255-261, 1999. Article (CrossRef Link)
[27] A. Monnet, A. Mittal, N. Paragios and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proc. IEEE International Conference on Computer Vision*, vol. 2, pp.1305-1312, 2003. Article (CrossRef Link)
[28] E. Kim, M. Park, S. Ji and M. Park, "A new approach to fuzzy modeling," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 3, pp. 328-337, Aug. 1997. Article (CrossRef Link)
[29] L.-X.Wang, *A Course in Fuzzy Systems and Control*, Prentice-Hall International Inc., 1997.
[30] T. B. Moeslund, A. Hilton and V. Kruger, "A survey of advances in video-based human motion capture analysis," *Computer Vision and Image Understanding*, vol. 104, pp. 90-126, 2006. Article (CrossRef Link)
[31] G. Bradski, A. Kaehler and V. Pisarevsky, "Learning-based computer vision with Intel's open source computer vision library" *Intel Technology Journal*, vol. 9, 2005.
[32] J. Kim and C. Lee, "Bayesian rule-based complex background modeling and foreground detection," *Optical Engineering*, vol. 49, no. 2, 027006, 2010. Article (CrossRef Link)
[33] D. Olson and D. Delen, "Advanced data mining techniques," *Springer*, 2008.

Heesung Lee received the BS, MS, and PhD degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2003, 2005, 2010, respectively. He is currently a senior engineer of Image Computing Research Team (ICRT) in Samsung S1 Co.,Ltd. His current research interests include computational intelligence, pattern recognition, biometrics, and neural networks.

Sungjun Hong received the BS degree in electrical and electronic engineering and computer science, from Yonsei University, Seoul, Korea, in 2005. He is a graduate student of the combined master's and doctoral degree programs at Yonsei University. He has studied machine learning, biometrics and optimization.

Euntai Kim received the BS (with top honors), MS, and PhD degrees in electronic engineering from Yonsei University, Seoul, Korea, in 1992, 1994, and 1999, respectively. From 1999 to 2002, he was a full-time lecturer with the Department of Control and Instrumentation Engineering at Hankyong National University, Gyeonggi-do, Korea. Since 2002, he has been with the School of Electrical and Electronic Engineering at Yonsei University, where he is currently a professor. He was a visiting scholar with the University of Alberta, Edmonton, Canada, in 2003, and a visiting researcher with the Berkeley Initiative in Soft Computing (BISC), UC Berkeley, USA, in 2008. His current research interests include computational intelligence and machine learning and their application to intelligent service robots, unmanned vehicles, home networks, biometrics, and evolvable hardware.