IPSJ Transactions on Computer Vision and Applications Vol. 2 156–168 (Nov. 2010)

Research Paper

Object Detection Based on Combining Multiple Background Modelings

Tatsuya Tanaka,[†1] Satoshi Yoshinaga,[†1]
Atsushi Shimada,[†1] Rin-ichiro Taniguchi,[†1]
Takayoshi Yamashita[†2] and Daisaku Arita[†3]

We propose a new method for background modeling based on combination of multiple models. Our method consists of three complementary approaches. The first one, or the pixel-level background modeling, uses the probability density function to approximate background model, where the PDF is estimated non-parametrically by using Parzen density estimation. Then the pixel-level background modeling can adapt periodical changes of pixel values. The region-level background modeling is based on the evaluation of local texture around each pixel, which can reduce the effects of variations in lighting. It can adapt gradual change of pixel value. The frame-level background modeling detects sudden and global changes of the image brightness and estimates a present background image from input image referring to a model background image, and foreground objects can be extracted by background subtraction. In our proposed method, integrating these approaches realizes robust object detection under varying illumination, whose effectiveness is shown in several experiments.

1. Introduction

Background subtraction technique has been often applied to detection of objects in images. It is quite useful because we can get object regions without prior information about the objects by subtracting a background image from an observed image. However, simple background subtraction often detects not only objects but also a lot of noise regions, because it is quite sensitive to small illumination changes caused by moving clouds, swaying tree leaves, water ripples glittering in the light, etc.

To handle these background changes, many techniques have been proposed [1)–11)]. In general, they are classified into three categories: pixel-level, region-level and frame-level background modelings.

pixel-level modeling In most of the pixel-level background modelings, the distribution of each pixel value is described in a probabilistic framework [1),9)]. A typical method is Elgammal's method where the probability density function is estimated by Parzen density estimation in a non-parametric form. Probabilistic methods construct the background model referring to observed images in the past, and, therefore, they are effective for repetitive brightness changes, which are caused by fluctuation of illumination, swaying tree leaves etc. However, these approaches including other pixel-level background models [8),10),11)] can not handle sudden illumination changes correctly, which are not observed in the previous frames.

region-level modeling Several approaches which consider spatial information such as texture have been proposed [5),12)–14)]. Methods based on Radial Reach Correlation (RRC) and Local Binary Pattern (LBP) are typical region-level, or texture-based, background modelings [5),12)]. In those methods, local texture information around a pixel, which is represented in terms of magnitude relation of the pixel and its peripheral pixels, is described. Usually, this magnitude relation does not change even if the illumination condition changes, and, thus, region-level background modeling often gives us more robust results than the pixel-level modeling. However, local changes of brightness, such as the change due to tree leave swaying, can not be handled correctly. BPRRC [13)] is also region-level background model which is robust against illumination changes. However, it requires some background changes in advance to train the changes.

frame-level modeling Fukui, et al. have proposed a method to estimate the current background image from a model background image [15)]. In their method, background candidate regions are extracted from the current image and the background generation function is estimated referring to the brightness of the candidate regions. The generation function is estimated under the assumption that the illumination changes uniformly in the image, and, therefore, nonuniform illumination changes can not be handled. In addition, the model background image is acquired in advance, and, unpredicted

†1 Kyushu University
†2 OMRON Corporation
†3 Institute of Systems, Information Technologies and Nanotechnologies

changes of the background can not be handled as well.

As mentioned above, each approach has merits and demerits depending on the assumptions of characteristics of the background and the illumination. Therefore, to achieve more robust object detection, or to acquire more effective background model, we should combine adaptively background modelings having different characteristics. In this paper, we propose integrated background modeling combining the pixel-level, the region-level and the frame-level background modelings.

2. Basic Background Modeling

Before we describe our integrated modeling approach, we propose improvements to each of the three basic background models. These improvements provide better performance of the integrated modeling.

2.1 Pixel-level Background Modeling

For pixel-level background modeling, probabilistic modeling of pixel value is the most popular method, in which fast and accurate estimation of the probability density function (PDF) of each pixel value is quite important. To estimate the PDF, Parzen density estimation is quite effective. However, to acquire an accurate estimation, sufficient of samples are required, and a typical method [1)] requires a lot of computation, which is proportional to the number of samples. Therefore, it cannot be applied in real-time processing. To solve this problem, we have designed a fast algorithm of PDF estimation. In our algorithm, we have used a rectangular function (See **Fig. 1**) as the kernel function K, instead of Gaussian function, which is often used in Parzen density estimation.

$$K(u) = \begin{cases} \frac{1}{h} & \text{if } -\frac{h}{2} \le u \le \frac{h}{2} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where h is a parameter representing the width of the kernel, i.e., some smoothing parameter. Using this kernel, the PDF is represented as follows:

$$P(\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{h^d}\psi\left(\frac{|\boldsymbol{X}-\boldsymbol{X}_i|}{h}\right) \tag{2}$$

where, $|\boldsymbol{X}-\boldsymbol{X}_i|$ means the chess-board distance in d-dimensional space, and $\psi(u)$ is calculated by the following formula.

$$\psi(u) = \begin{cases} 1 & \text{if } u \le \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

When an observed pixel value X is inside of the kernel located at $\boldsymbol{X}$, $\psi(u)$ is 1; otherwise $\psi(u)$ is 0.

Thus, we estimate the PDF based on Eq. (2), and $P(\boldsymbol{X})$ is calculated by enumerating pixels in the latest pixel process whose values are inside of the kernel located at $\boldsymbol{X}$. However, if we calculate the PDF, in a naive way, by enumerating pixels in the latest pixel process whose values are inside of the kernel located at $\boldsymbol{X}$, the computational time is proportional to N. Instead, we propose a fast algorithm to compute the PDF, whose computation cost does not depend on N.

In background modeling we estimate $P(\boldsymbol{X})$ referring to the latest pixel process consisting of pixel values of the latest N frames. Let us suppose that at time t we have a new pixel value $\boldsymbol{X}_{N+1}$, and that we estimate an updated PDF $\boldsymbol{P}_t(\boldsymbol{X})$ referring to the new $\boldsymbol{X}_{N+1}$. Basically, the essence of PDF estimation is accumulation of the kernel estimator, and, when a new value, $\boldsymbol{X}_{N+1}$, is acquired the kernel estimator corresponding to $\boldsymbol{X}_{N+1}$ should be accumulated. At the same time, the oldest one, i.e., the kernel estimator at N frames earlier, should be discarded, since the length of the pixel process is constant, N. This idea leads to reduction of the PDF computation into the following incremental computation:

$$P_t(\boldsymbol{X}) = P_{t-1}(\boldsymbol{X}) + \frac{1}{Nh^d}\psi\left(\frac{|\boldsymbol{X}-\boldsymbol{X}_{N+1}|}{h}\right) - \frac{1}{Nh^d}\psi\left(\frac{|\boldsymbol{X}-\boldsymbol{X}_1|}{h}\right) \tag{4}$$

where P_{t-1} is the PDF estimated at the previous frame.

The above equation means that when a new pixel value is observed, the PDF can be acquired by:

- increasing the probabilities of pixel values which are inside of the kernel located at the new pixel value $\boldsymbol{X}_{N+1}$ by $\frac{1}{Nh^d}$
- decreasing those which are inside of the kernel located at the oldest pixel value, a pixel value at N frames earlier, $\boldsymbol{X}_1$ by $\frac{1}{Nh^d}$.

In other words, the new PDF is acquired by local operation of the previous PDF, assuming the latest N pixel values are stored in the memory, which achieves quite fast computation of PDF estimation. **Figure 2** illustrates how the PDF, or

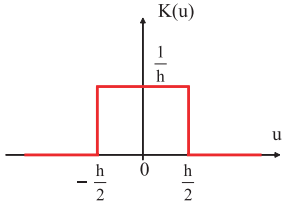


Fig. 1 Kernel function of our algorithm.

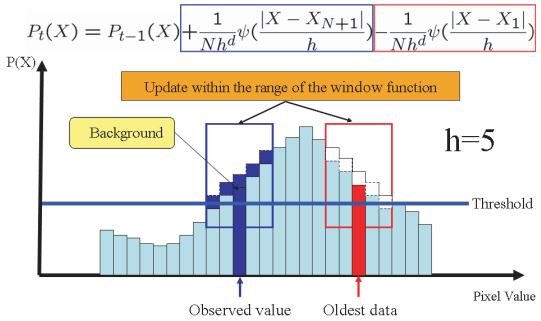


Fig. 2 Update of background model.

the background model, is modified. Please refer to the paper [16] for experimental results of the fast algorithm.

2.2 Region-level Background Modeling

To realize robust region-level background modeling, we have improved Radial Reach Correlation (RRC) [5] so that the background model is properly updated according to background changes of input image frames.

2.2.1 Radial Reach Correlation (RRC) [5]

In order that each pixel is robustly judged as either the foreground or the background without the influence of illumination changes, Radial Reach Correlation (RRC) has been introduced to evaluate local texture similarity. RRC is calculated at each pixel (x, y). At first, pixels whose brightness differences to the brightness of the center pixel (x, y) exceed a threshold are searched for in every radial reach extension in 8 directions around the pixel (x, y). We refer to the searched 8 pixels as *peripheral pixels* hereafter. Then, the signs of brightness differences (positive difference or negative difference) of the 8 pairs, each of which is a pair of one of eight peripheral pixels and the center pixel (x, y), are represented in a binary code. The basic idea is that the binary code represents intrinsic information about local texture around the pixel, and that it does not change under illumination changes. To make this idea concrete, the correlation value of the binary codes extracted from the observed image and the reference background image is calculated to evaluate their similarity.

Suppose that the position of a pixel is represented as a vector $\boldsymbol{p} = (x, y)^T$, and that the directional vectors of radial reach extensions are defined as $\boldsymbol{d}_0 = (1, 0)^T$, $\boldsymbol{d}_1 = (1, 1)^T$ $\boldsymbol{d}_2 = (0, 1)^T$, $\boldsymbol{d}_3 = (-1, 1)^T$, $\boldsymbol{d}_4 = (-1, 0)^T$, $\boldsymbol{d}_5 = (-1, -1)^T$ $\boldsymbol{d}_6 = (0, -1)^T$, and $\boldsymbol{d}_7 = (1, -1)^T$. Then the reaches $\{r_k\}_{k=0}^7$ for these directions are defined as follows referring to the reference image f, or the background image here:

$$r_k = \min\{r| \ |f(\boldsymbol{p} + r\boldsymbol{d}_k) - f(\boldsymbol{p})| \geq T_P\} \tag{5}$$

where $f(\boldsymbol{p})$ represents the pixel value of the position of $\boldsymbol{p}$ in the image f, and T_P represents the threshold value to detect a pixel with different brightness. When r_k cannot be detected, we regard the pixel on the bound of image as a pixel with different brightness. If such case often occurs, region-revel background model cannot represent spatial features. However, the problem rarely rises in the scenes used in our experiments.

Based on the brightness difference between the center pixel and the peripheral pixels (defined by Eq. (5)), the coefficients of incremental code, or polarity code, of the brightness distribution around the pixel in the background image f are given by the following formula:

$$b_k(\boldsymbol{p}) = \begin{cases} 1 & \text{if } f(\boldsymbol{p} + r_k\boldsymbol{d}_k) \geq f(\boldsymbol{p}) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $k = 0, 1, \ldots, 7$. In the same manner, the incremental codes are calculated for the input image g, except that the reach group $\{r_k\}_{k=0}^7$ is established in the background image f, not in the input image g.

$$b_k{}'(\boldsymbol{p}) = \begin{cases} 1 & \text{if } g(\boldsymbol{p} + r_k\boldsymbol{d}_k) \geq g(\boldsymbol{p}) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Based on the obtained $b_k(\boldsymbol{p})$, $b_k{}'(\boldsymbol{p})$, the number of matches (correlation), $B(\boldsymbol{p})$, between the two incremental codes are calculated as follows.

$$B(\boldsymbol{p}) = \sum_{k=0}^{7}\{b_k(\boldsymbol{p}) \cdot b_k{}'(\boldsymbol{p}) + \overline{b_k(\boldsymbol{p})} \cdot \overline{b_k{}'(\boldsymbol{p})}\} \tag{8}$$

where $\overline{x} = 1 - x$ represents the inversion of a bit x. $B(\boldsymbol{p})$ represents the similarity, or correlation value, of the brightness distribution around the pixel $\boldsymbol{p}$ in the two images. This is called Radial Reach Correlation (RRC).

Since RRC between an input image pixel and its corresponding background image pixel represents their similarity, it can be used as a measure to detect foreground pixels. That is, pixels whose RRC are smaller than a certain threshold can be judged as foreground pixels.

2.2.2 Background Modeling Based on Adaptive RRC

Using RRC, the similarity between incremental codes of a background image pixel and its corresponding pixel in the observed image can be calculated. Pixels which are not "similar" to their corresponding pixels in the background image are detected as foreground pixels. In principle, if the background does not change, we can prepare adequate codes of the background image in advance. However, usually, due to the illumination changes and various noises, it is almost impossible to prepare them. Even if we manage to prepare such fixed background codes, accurate results can not be acquired, and, therefore, we should update the background codes properly. Here, we have developed a mechanism to update the background codes according to the following formula:

$$\hat{b}_k^{t+1}(\boldsymbol{p}) = (1-\alpha)\cdot \hat{b}_k^t(\boldsymbol{p}) + \alpha \cdot \hat{b}_k'^t(\boldsymbol{p}) \tag{9}$$

where $\hat{b}_k^t(\boldsymbol{p})(k=0,1,\ldots,7)$ represents the incremental code of a pixel $\boldsymbol{p}$ at time t. α is a learning rate, and when it is large enough the above code can be quickly adapted to the current input image, i.e., adapted to sudden background changes. The range of $\hat{b}_k^t(\boldsymbol{p})$ is $[0,1]$, and when $\hat{b}_k^t(\boldsymbol{p})$ is close to either 0 or 1, it means that the magnitude relation of brightness between the center pixel and its peripheral pixel does not change. Otherwise, i.e., if $\hat{b}_k^t(\boldsymbol{p})$ is close to 0.5, the magnitude relation is not stable. According to this consideration, a peripheral pixel is sought again when $T_r \le \hat{b}_k^t(\boldsymbol{p}) \le 1 - Tr$ holds. T_r is a threshold value to invoke re-searching of peripheral pixels.

Then, we re-define the similarity of the incremental codes as follows:

$$\hat{B}^t(\boldsymbol{p}) = \sum_{k=0}^{7} |\hat{b}_k^t(\boldsymbol{p}) - \hat{b}_k'^t(\boldsymbol{p})| \tag{10}$$

Similarity or dissimilarity is judged by comparing the similarity value with a threshold T_B. Therefore, when $\hat{B}^t(\boldsymbol{p})$ is smaller than T_B, we regard the pixel as background.

The detailed procedure of background modeling is summarized as follows:

Step1 The incremental codes of the current frame g are calculated, and foreground pixels are discriminated from background pixels according to the similarity (defined in Eq. (10)) of the incremental codes of the input image pixels and those of the background pixels.

Step2 The incremental codes of the background pixels judged in the integration process (see Section 3) are updated according to Eq. (9).

Step3 When $T_r \le \hat{b}_k^t(\boldsymbol{p}) \le 1-Tr$ becomes to hold, its peripheral pixel is sought again in the current frame, and $\hat{b}_k^t(\boldsymbol{p})$ is re-initialized using the newly found peripheral pixel.

As mentioned above, we selectively update the region-level background model. Though selective update process sometimes propagates misclassification to successive frames, we ease this problem by integrating the detection results from multiple background models. For details, see Section 3 and experimental results.

2.3 Frame-level Background Modeling

Frame-level background modeling proposed by Fukui, et al. [15], which is based on brightness normalization of a model background image, is designed to be robust against sudden illumination changes. They assume that pixels having the same brightness value change their brightness in the same way when the illumination condition changes, i.e., they assume that the illumination changes occur uniformly in the entire image. Therefore, in principle, their method can not detect objects robustly under non-uniform illumination changes. In addition, since the model background image is prepared in advance, it can not handle unexpected background changes as well.

In our method, on the contrary, the model background image is adaptively generated referring to recent pixel values, and unexpected background changes can be dealt with. Then, the brightness of the model background image is further normalized based on on-line training, and the influence of non-uniform illumination changes is reduced. Finally, objects can be simply detected by subtracting the normalized background image from an observed image.

To realize robust brightness normalization, we have designed a multi-layered perceptron, by which the mapping between pixel brightness of the model background image and that of an observed image is established. To reflect the locality

of the brightness distribution, the input vector of the perceptron consists of the image coordinates (x, y) and the brightness value (R, G, B) of a pixel, and this combination can handle the non-uniform illumination changes. The detailed procedure of the frame-level background modeling is as follows

Step1 Learning: the mapping between an input vector, (x, y, R, G, B), of a pixel in a model background image and an output vector, (R', G', B') of the pixel at the same position in the observed image is learned. (x, y) is the coordinates of the pixel and (R, G, B), (R', G', B') are the color brightness values of the pixels. To achieve on-line training, we have to also acquire training data on-line, which is achieved in the integration process of the background modelings. Details will be presented in the next section.

Step2 Normalization: the brightness of the model background image is normalized using the perceptron learned in Step1, which means that the background image corresponding to the observed image is estimated.

Step3 Object detection: subtraction of the normalized background image, which is estimated in Step2, from the observed image gives us the object detection result. That is, pixels which have large brightness difference (larger than a given threshold, T_{det}) are detected as foreground pixels, or object pixels.

3. Combination of Multiple Background Modelings

Finally, we present our major contribution, i.e., integrated background modeling, which is realized by integrating pixel-level, region-level and frame-level background modelings. One of the important issues here is how to select training samples for the frame-level background modeling, which should exclude wrong samples as much as possible. In the preliminary experiments, we have found that background pixels judged in the integration process (see Step3) contain little false negatives (i.e., pixels which are, in reality, foregrounds), and the training samples are selected from those pixels. The flowchart of the integration process is shown in **Fig. 3** and the detailed processing flow is as follows. Note that the initialization process is achieved by using the first frame image.

Step1 Objects are detected based on the **pixel-level** background modeling.

Step2 Objects are detected based on the **region-level** background modeling.

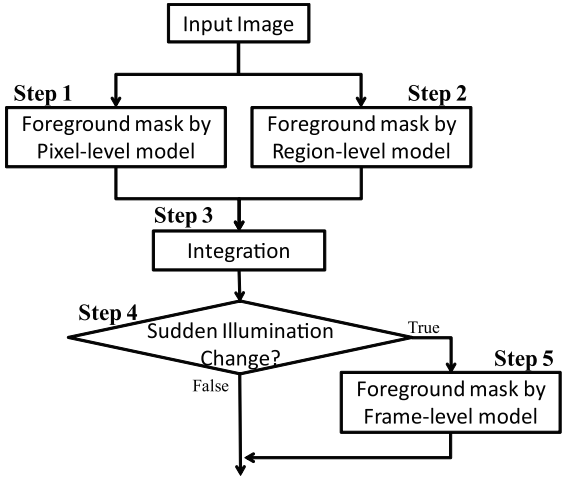


Fig. 3 Flowchart of integration process.

Step3 Object detection results of Step1 and Step2 are combined. That is, pixels which are judged as foregrounds by both of the above modelings are judged as foregrounds and other pixels are judged as backgrounds. Then, the parameters of the background models are modified. First, the PDF of the pixel value of the input images, which is maintained in the pixel-level background model, is updated. In addition, when a pixel is judged as a background pixel here, the parameters of region-level background model are modified.

Step4 When the brightness difference between the current frame and the previous frame is large at a certain number of pixels, we establish TTL (Time To Live) to the frame-level background model, and TTL represents the duration where the frame-level model is activated. By using TTL, we activate the frame-level model only when the illumination condition suddenly changes.

Step5 If TTL > 0, objects are finally detected based on the frame-level background model and TTL is decreased. Otherwise, object detection result acquired in Step3 is adopted. The frame-level background modeling is achieved as follows:

(5-1) A model background image is generated so that each pixel has the

most frequent pixel value in its PDF. The PDF is maintained in the pixel-level background model.

(5-2) Training samples to adjust the model background image generated in (5-1) are selected from pixels judged as background in Step3 (called as background candidate pixels (BCPs)). In practice, at each frame, 100∼200 pixels out of BCPs are randomly sampled and used as training samples.

(5-3) After the multi-layered perceptron is trained at each frame, referring to the training samples acquired in (5-2), the model background image is adjusted using the perceptron, and, the subtraction of the adjusted model background image from the observed image becomes the final object detection result.

As explained above, the pixel-level, region-level and frame-level background model are integrated. Note that the frame-level background model is used only when the illumination condition suddenly changes. This is because pixel-level and region-level are adaptable to other illumination changes such as periodic changes or gradual changes. Therefore, we selectively use the frame-level background model from the viewpoint of computational cost. On the other hand, we detect the sudden illumination changes by independent process from pixel-level and region-level background models. This is because whether or not sudden illumination changes should be investigated by successive two frame images. Actually, the judgment of sudden illumination change is investigated by calculating the brightness difference between the current frame and the previous frame. Finally, when sudden illumination change occurs, we acquire another foreground mask, which means the final detection result just detected by frame-level model.

4. Experiment

To evaluate the performance of our method, we have used two outdoor scene data sets of PETS (PETS2001) [⋆1], which are often used in video-based surveillance, and two indoor scene data sets publicly available for evaluation of object detection [⋆2]. The outdoor scenes include people passing through streets, tree leaves are flickering, and the weather conditions change rapidly. One of the indoor scenes includes sharp pixel value changes caused by camera aperture changes. Another includes sudden illumination changes caused by turning on/off of the light. We have also used Wallflower data set [3] to compare our proposed method with the Wallflower approach.

4.1 Preliminary Experiment

4.1.1 Region-level Background Modeling

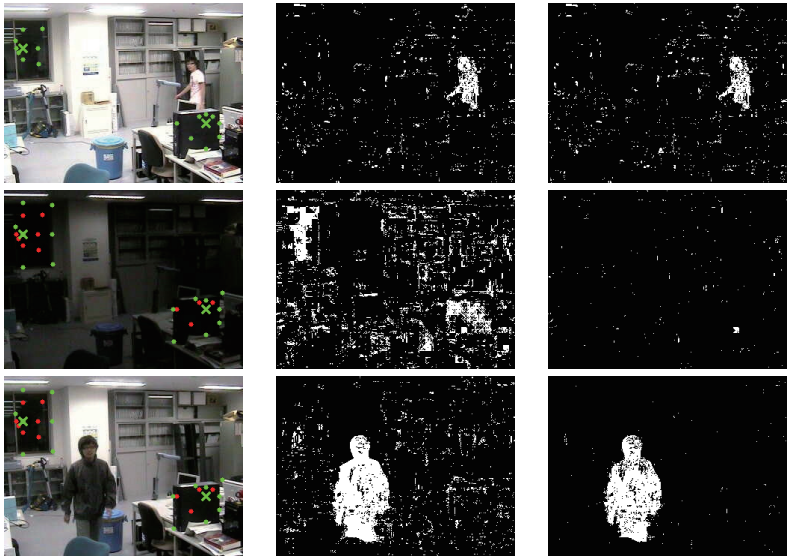
First, we investigated the effectiveness of our Adaptive RRC (ARRC). **Figure 4** comparatively shows the performance of ARRC where (a), (b) and (c) show input images, the results without re-searching of peripheral pixels, and the ones with re-searching, respectively. T_r, a threshold value to invoke the re-searching process, is set to be 0.3. Another parameter α in Eq. (9) was set to be 0.05, which is commonly used in the following experiments. The figure shows that re-searching of peripheral pixels gives us better results. Green and red marks in the input images show typical examples of peripheral pixels which are sought again: green crosses are center pixels, red dots are their initial peripheral pixels, and green dots are peripheral pixels which are found in the re-searching process. When there is no red dot, peripheral pixels which are the same as the initial ones are found again in the re-searching process. When there is no green dot, no peripheral pixel is found in that direction.

In case that the re-searching process is not invoked, we find incorrect detections in the window area of the upper left part of the images and around the PC in the lower right part. In these cases, brightness difference between the center pixel and its peripheral pixels becomes small when the illumination changes suddenly, and it means that peripheral pixels whose magnitude relation to the center pixel are not stable are selected. As a result, when the incremental codes of the input image pixels are affected by noise, such background pixels can be incorrectly detected as foregrounds.

On the other hand, when the re-searching is invoked, peripheral pixels whose

⋆1 Benchmark data of International Workshop on Performance Evaluation of Tracking and Surveillance. Available from http://ftp.pets.rdg.ac.uk/PETS2001/

⋆2 Several kinds of test images and their ground truth is available from http://limu.ait.kyushu-u.ac.jp/dataset/



(a) Input image (b) RRC without re-searching (c) Improved RRC with re-searching

Fig. 4 Effect of our improved RRC. Green and red marks in the input images show typical examples of peripheral pixels which are sought again: green crosses are center pixels, red dots are their initial peripheral pixels, and green dots are peripheral pixels which are found in the re-searching process.

magnitude relation to the center pixel are not stable are abandoned, and new peripheral pixels which correctly represent the local texture information are searched for. Thus, ARRC with the re-searching process outperforms the one without the re-searching process.

4.1.2 Frame-level Background Modeling

Secondly, we have evaluated the performance of background image estimation in the frame-level modeling using test data, an example of which is shown in **Fig. 5**: the left indicates an observed image, the center is the model background image, and the right is hypothetic foreground-candidate regions which are manually established for this experiment. In the observed images, there are non-uniform illumination changes due to turning on/off of the light, and, referring to pixels randomly sampled from background-candidate regions in the observed image and their corresponding pixels in the model background image, an adjusted background image is generated.

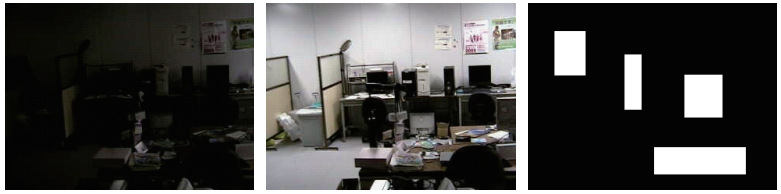


Fig. 5 Experimental data for background estimation. The left indicates an observed image, the center is the model background image, and the right is hypothetic foreground-candidate regions which are manually established for this experiment.

Table 1 Result of background image estimation.

	average error	std. deviation
Fukui [15]	9.5	13.9
NN (R, G, B)	8.5	10.4
NN (X, Y, R, G, B)	5.4	9.3

Here, we have examined three algorithms: an estimation method by Fukui, et al. [15], a three layered perceptron which accepts pixel position as its input, and one without the pixel position. Training data given to the algorithms consists of input vectors (x, y, R, G, B) randomly sampled from the background-candidate regions in the observed image and their corresponding output vectors (R', G', B') at the same pixel positions in the model background image. The number of the samples here is 100. The perceptron consists of 5 nodes in the input layer (3 nodes when the pixel position is not referred to), 3 nodes in the output layer and 3 nodes in the middle layer.

Table 1 shows the accuracy of estimation, i.e., the error between the observed image and the image estimated from the model background image. It indicates that our method, the three layered perceptron accepting pixel position information, is better than other methods. This is because, referring to the pixel position information, our method can robustly estimate the background image under non-uniform illumination changes. In other words, since the other methods do not take into account the pixel position information, they can not deal with non-uniform illumination changes. If the dynamic range of the model background

image is rather small (i.e., the model background image is acquired in a dark situation), the estimated one is not very accurate. In such cases, the input image, which is brighter, should be compensated instead of the model background image.

4.2 Experiment on the Integrated Background Modeling

In this experiment, we have used the following parameters:

Pixel-level The width of the rectangular kernel is 9 and the number of samples is 500. See the paper [10] for the details.

Region-level $T_p = 10$, $T_B = 2.5$, $T_r = 0.3$ and $\alpha = 0.05$ in Section 2.2.

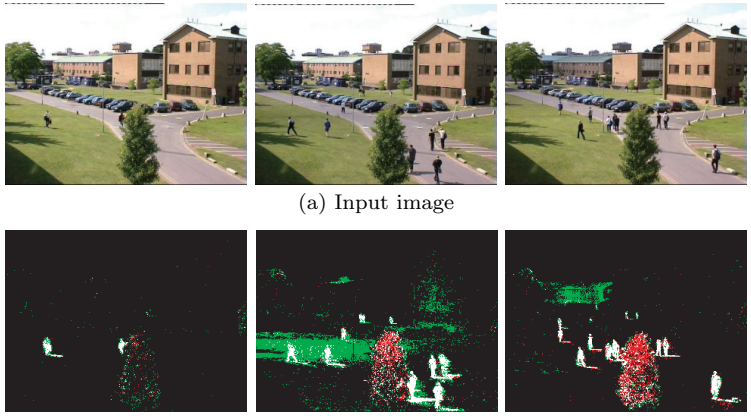
Frame-level When the number of pixels which have large difference value between the current frame and previous frame (we set the difference threshold to be 10) exceeded the half number of total pixel, the frame-level background model is used. $T_{det} = 30$. The initial value of TTL is 60, which has been decided according to the time-lag necessary for the pixel-level background model to adjust for illumination changes.

4.2.1 Effect of Integration

At first, the characteristics of the pixel-level and the region-level background modelings are compared with each other using an outdoor image sequence. The image sequence contains rapid illumination changes due to weather condition changes and swaying tree leaves, which can not be dealt with by simple background subtraction techniques. However, it does not include sudden illumination changes such as the change of camera parameters, light switches. Hence, the frame-level background model was not activated in the outdoor scenes. **Figure 6** shows a typical experimental result, where the top row indicates some of the input frames and where the bottom row indicates their processed results. In this figure, each pixel is represented as follows:

- black: judged as "background" by the both of the pixel-level and the region-level background modelings
- red: "background" by the pixel-level and "foreground" by the region-level
- green: "foreground" by the pixel-level and "background" by the region-level
- white: "foreground" by the both modelings

From these results, we can observe two important characteristics. First, red pixels indicate that small fluctuation of the background due to swaying tree leaves



(a) Input image

(b) Detection result

Fig. 6 Detection results by the pixel-level and the region-level background modelings. In the detection result, black pixels are judged as "background" by the both of the pixel-level and the region-level background modelings. Red pixels are judged as "background" by the pixel-level and "foreground" by the region-level. Green pixels are judged as "foreground" by the pixel-level and "background" by the region-level. White pixels are judged as "foreground" by the both modelings.

causes incorrect detection in the region-level modeling. This is because the magnitude relation between the center pixel and its peripheral pixels has changed due to the fluctuation of the background. Secondly, green pixels indicate that rapid illumination changes cause incorrect detection in the pixel-level modeling. This is because the pixel-level modeling statistically represents the pixel value distribution based on the past pixel values, and because rapid illumination changes which have not been observed previously can not represented in the current background model. Considering the above observations, we can summarize that the pixel-level and the region-level modelings well complement each other, and that, as in presented in Section 3, integrating the results of the both modelings by intersection operation (i.e., white pixels in Fig. 6) gives us fairly accurate result.

The effect of the frame-level background modeling is shown in **Fig. 7**, where illumination condition is suddenly changed due to the extinction of a light. It is shown that the pixel-level background modeling causes a lot of incorrect detections everywhere in the image. Combining the pixel-level and the region-level

(a) Input image (b) Most probable pixel values (c) Estimated background model

(d) Pixel-level background modeling (e) Pixel and region-level modelings (f) Integration of the three modelings

Fig. 7 Effects of integration of different background modelings.

modelings provides a better result but there still remain a lot of incorrect detections because of the non-uniform illumination change. Finally, integrating the three modelings, i.e., the pixel-level, the region-level and the frame-level modelings, provides the best result. It proves the effectiveness of the frame-level modeling, which can compensate position-dependent sudden illumination changes.

4.2.2 Accuracy Evaluation

We have evaluated the accuracy of object detection in terms of precision and recall, comparing the proposed method with adaptive Gaussian mixture model [9], fast Parzen estimation [10], RRC [5] and spatial locality model [17]. **Table 2** shows the result, and it is clearly shown that our proposed method outperforms the other competitive methods based on pixel-level, region-level and their combination modeling. **Fig. 8** shows their object detection example.

Considering the experimental results, we can see the following characteristics:

- The pixel-level background modeling (based on Parzen density estimation and GMM) can adapt background changes in the outdoor scenes, but can not handle the sudden illumination changes correctly in the indoor scenes.
- RRC can handle illumination changes, but it can not handle background changes such as changes due to moving clouds. This is because RRC employs a fixed reference image [⋆1], and texture information which does not appear in the initial frame can not be handled correctly. Therefore, background pixels

Table 2 Comparative results of object detection accuracy.

		Outdoor 1	Outdoor 2	Indoor 1	Indoor 2
GMM [9]	Recall	61.3 %	34.9 %	38.1 %	35.6 %
	Precision	58.2 %	55.0 %	59.7 %	46.1 %
Parzen [10]	Recall	56.3 %	46.8 %	43.0 %	37.8 %
	Precision	51.6 %	72.8 %	42.0 %	58.5 %
RRC [5]	Recall	37.5 %	24.8 %	35.3 %	26.9 %
	Precision	22.4 %	20.7 %	51.2 %	24.9 %
Adaptive RRC (Section 2.2.2)	Recall	50.0 %	46.6 %	47.1 %	69.7 %
	Precision	65.0 %	46.1 %	83.1 %	75.8 %
Spatial Locality [17]	Recall	71.6 %	60.2 %	62.6 %	52.1 %
	Precision	72.6 %	55.6 %	74.4 %	60.0 %
Proposed method without frame-level model	Recall	77.9 %	65.5 %	75.4 %	68.3 %
	Precision	69.1 %	69.7 %	90.8 %	70.4 %
Proposed method	Recall	77.9 %	65.5 %	73.9 %	76.1 %
	Precision	69.1 %	69.7 %	92.1 %	77.4 %

(a) Input images (b) Ground truth (c) Proposed

(d) GMM (e) RRC (f) Parzen

Fig. 8 Examples of object detection result.

⋆1 In this experiment, incremental codes for reference are generated from the initial frame.

are incorrectly detected as foreground pixels. The adaptive RRC proposed in this paper (Section 2.2.2) brings better results than the original RRC since it can update the background model in response to background changes.

- Two combinational models bring better results than methods which use pixel-level or region-level model only; one is the spatial locality model [17] and the other is our proposed method without frame-level background model.
- Our proposed method can detect objects robustly against both of background changes and illumination changes. Meanwhile, objects detected by our proposed method tended to be shrunk by the integration process of pixel-level and region-level background model. This is why some false negative pixels still exist. In this study, we integrated two background models by calculating the logical product, which is an immediate cause of false negatives. In addition, our proposed method detected not only object regions but also their shadow regions. The shadow regions were responsible for increase in false positive pixels. From the viewpoint of object detection, we are sure that the object regions detected by our proposed method gives us enough information for post-processing such as event detection, crowd analysis and so on.
- The integrated model brings better results than the single use of each model. For example, the accuracy of our proposed method without frame-level background model is superior to a single model (i.e., Parzen or Adaptive RRC

Table 3 Parameter ranges which affect the variation of precision and recall within 10%.

	Outdoor 1	Indoor 2
T_p	[5,15]	[5,15]
T_B	[1.5,2.5]	[2.5,3.5]
T_r	[0.1,0.4]	[0.1,0.4]
T_{det}	[20,40]	[20,40]

Table 4 Performance evaluation using Wallflower dataset.

		Moved Object	Time of Day	Light Switch	Waving Trees	Camou-flage	Boot-strap	Foreground Aperture	**Total Errors**
Frame Difference	FN	0	1,165	2,479	3,509	9,900	1,881	3,884	
	FP	0	193	86	3,280	170	294	470	27,311
Mean + Threshold	FN	0	873	1,116	17	194	415	2,210	
	FP	0	1,720	15,116	3,268	1,638	2,821	608	29,996
Mean + Covariance	FN	0	949	1,857	3,110	4,101	2,215	3,464	
	FP	0	535	15,123	357	2,040	92	1,290	35,133
GMM	FN	0	1,008	1,633	1,323	398	1,874	2,442	
	FP	0	20	14,169	341	3,098	217	530	27,053
Block Correlation	FN	0	1,030	883	3,323	6,103	2,638	1,172	
	FP	1,200	135	2,919	448	567	35	1,230	21,683
Temporal Derivative	FN	0	1,151	752	2,483	1,965	2,428	2,049	
	FP	1,563	11,842	15,331	259	3,266	217	2,861	46,167
Bayesian Decision	FN	0	1,018	2,380	629	1,538	2,143	2,511	
	FP	0	562	13,439	334	2,130	2,764	1,974	31,422
Eigen-background	FN	0	879	962	1,027	350	304	2,441	
	FP	1,065	16	362	2,057	1,548	6,129	537	17,677
Linear Prediction	FN	0	961	1,585	931	1,119	2,025	2,419	
	FP	0	25	1,3576	933	2,439	365	649	27,027
Wallflower	FN	0	961	947	877	229	2,025	320	
	FP	0	25	375	1,999	2,706	365	649	11,478
Proposed Method	FN	0	1,349	1,681	198	177	1,235	2,085	
	FP	0	0	1,396	771	342	199	658	10,091

FN: False Negative, FP: False Positive

in Table 2). In addition, we can see that frame-level background model enhances the robustness against illumination changes in the indoor scenes (see the results of "our proposed method without frame-level background model" and "our proposed method" in Table 2).

We have conducted some additional experiments using Outdoor scene 1 and Indoor scene 2. We changed some parameters in each scene and investigated how the parameters affected the recall and precision ratio. **Table 3** shows the parameters which affected the variation of precision and recall within 10%. We can see that the parameter setting is not so severe and some ranges of each parameter are allowed.

We have also evaluated our method using Wallflower dataset [3], in which images and their ground truth data for various background subtraction issues are included. **Table 4** shows the results, in which accuracy of the methods other than ours is cited from the Wallflower paper. The column of total errors indicates the summation of false positive and false negative pixels in each scene. The total errors of our proposed method is fewer than the one of Wallflower. **Figure 9** illustratively shows some of the comparative results. Although these results indicate that the performance of our method is almost the same as that of Wallflower, our method requires no off-line training and it is much more useful than Wallflower, which requires advance learning of background images. With regard to the "Light Switch" scene, our method detected more false positives than Wallflower. To reduce the number of false positive pixels, we will introduce illumination likelihood model [18] in the future works.

4.2.3 Computational Cost Evaluation

Computational cost of the proposed method is evaluated using an image sequence shown in Fig. 7. For the evaluation of computational cost, we have used a PC with an Intel Core2 3.16 GHz and 4.0 GB memory. Our source code was implemented in C++. **Figure 10** indicates its required computation time. Required computation times of the pixel-level and the region-level background modelings are about 40 msec and 15 msec, respectively, and they are relatively stable. On the contrary, the frame-level background modeling is used only when the brightness of the image suddenly changes, and, as a result, its computational cost is nearly zero in most of the frames. However, when the frame-level background

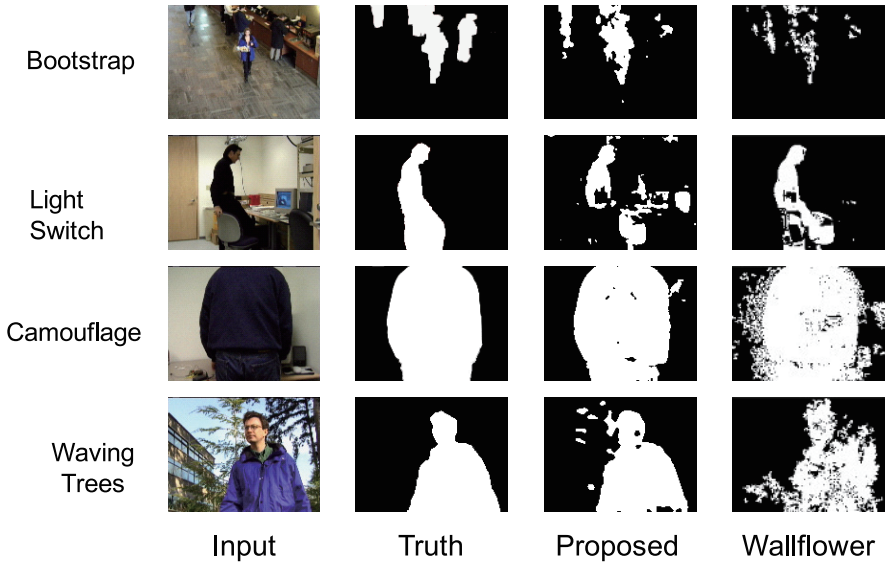


Fig. 9 Illustrative comparison between the proposed method and Wallflower.

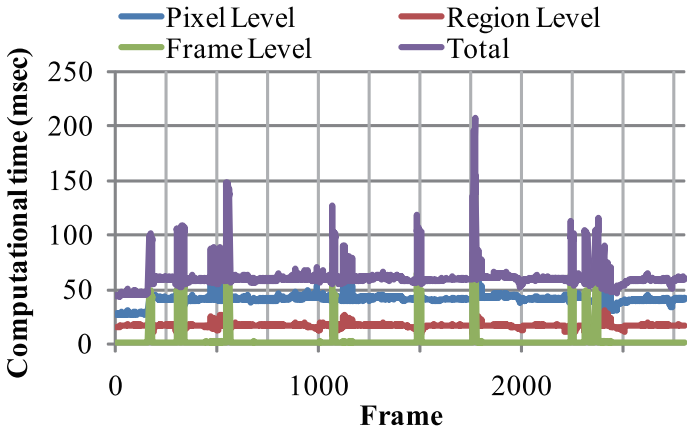


Fig. 10 Computation time of the proposed method.

modeling is activated, its computation time becomes 50∼80 msec. It amounts to almost the half of the total computation time, and varies depending on the image frame due to the neural network learning. Thus, the computation time of each frame largely varies depending on its data. From the viewpoint of real-time processing, computation reduction of the frame-level background modeling and stabilization of the computational cost are important future works.

5. Conclusion

In this paper, we have presented combinational background modeling and robust object detection based on this modeling. By integrating several background modelings having different characteristics, we can establish more robust background model against variety of background and illumination changes.

Our future works are summarized as follows:

- **Improvement of the estimation accuracy of background image**
  Background image estimation in Step5 of Section 3 should be more accurate. Since this accuracy directly affects object detection result, the accurate estimation is inevitable.
- **Stabilization of computation time**
  Computation time required in the frame-level background modeling becomes larger compared with other background modelings. This is partly because the training of the perceptron is executed. In addition, the time varies depending on input images. Therefore, to realize a practical online system, reduction and stabilization of its computation time is quite important.

References

1) Elgammal, A., Harwood, D. and Davis, L.: Non-parametric Model for Background Subtraction, *6th European Conference on Computer Vision* (*ECCV*), Vol.2, pp.751–767 (2000).
2) Elgammal, A., Duraiswami, R. and Davis, L.: Efficient kernel density estimation suing the Fast Gauss Transform with applications to color modeling and tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.25, No.11, pp.1499–1504 (2003).
3) Toyama, K., Krumm, J., Brumitt, B. and Meyers, B.: Wallflower: Principle and Practice of Background Maintenance, *International Conference on Computer Vision*, pp.255–261 (1999).
4) Li, L., Huang, W., Gu, I.Y.-H. and Tian, Q.: Statistical Modeling of Complex Background for Foreground Object Detection, *IEEE Transactions on Image Processing*, Vol.13, No.11, pp.1459–1472 (2004).
5) Satoh, Y., Kaneko, S., Niwa, Y. and Yamamoto, K.: Robust object detection using a Radial Reach Filter (RRF), *Systems and Computers in Japan*, Vol.35, No.10, pp.63–73 (2004).
6) Monari, E. and Pasqual, C.: Fusion of Background Estimation Approaches for Motion Detection in Non-static Backgrounds, *CD-ROM Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance* (2007).
7) Ukita, N.: Target-color Learning and Its Detection for Non-stationary Scenes by Nearest Neighbor Classification in the Spatio-Color Space, *Proc. IEEE International Conference on Advanced Video and Signal based Surveillance*, pp.394–399 (2005).
8) Stauffer, C. and Grimson, W.: Adaptive background mixture models for real-time tracking, *IEEE International Conference on Computer Vision and Pattern Recognition* (*CVPR*), Vol.2, pp.246–252 (1999).
9) Shimada, A., Arita, D. and Taniguchi, R.: Dynamic Control of Adaptive Mixture-of-Gaussians Background Model, *CD-ROM Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance* (2006).
10) Tanaka, T., Shimada, A., Arita, D. and Taniguchi, R.: A Fast Algorithm for Adaptive Background Model Construction Using Parzen Density Estimation, *CD-ROM Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance* (2007).
11) Luo, R., Li, L. and Gu, I.Y.: Efficient Adaptive Background Subtraction Based on Multi-resolution Background Modeling and Updating, *Advances in Multimedia Information Processing — PCM 2007*, pp.118–127 (2007).
12) Zhang, S., Yao, H. and Liu, S.: Dynamic background modeling and subtraction using spatio-temporal local binary patterns, *IEEE International conference on Image Processing*, pp.1556–1559 (2008).
13) Yokoi, K.: Probabilistic BPRRC: Robust Change Detection against Illumination Changes and Background Movements, *IAPR Conference on Machine Vision Applications*, pp.148–151 (2009).
14) Zhao, G. and Pietikainen, M.: Dynamic Texture Recognition Using Volume Local Binary Patterns, *Workshop Dynamical Vision*, pp.165–177 (2007).
15) Fukui, S., Ishikawa, T., Iwahori, Y. and Itoh, H.: Extraction of Moving Objects by Estimating Background Brightness, *Journal of the Institue of Image Electronics Engineers of Japan*, Vol.33, No.3, pp.350–357 (2004).
16) Tanaka, T., Shimada, A., Arita, D. and Taniguchi, R.: Non-parametric Background and Shadow Modeling for Object Detection, *Proceedings of the 8th Asian Conference on Computer Vision*, pp.159–168 (2007).
17) Tanaka, T., Shimada, A., Arita, D. and Taniguchi, R.: Object Detection under

Varying Illumination based on Adaptive Background Modeling Considering Spatial Locality, *Proc. 3rd Pacific-Rim Symposium on Image and Video Technology*, pp.645–656 (2009).
18) Pilet, J., Strecha, C. and Fua, P.: Making Background Subtraction Robust to Sudden Illumination Changes, *10th European Conference on Computer Vision*, pp.567–580 (2008).

(Received November 10, 2009)
(Accepted July 29, 2010)
(Released November 10, 2010)

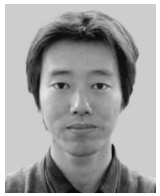
(Communicated by *Yoshito Mekada*)

Tatsuya Tanaka received his B.E. and M.E. degrees from Kyushu University in 2007 and 2009. He is currently working for Toshiba Corporation. In his master course he was engaged in computer vision and image processing.

Satoshi Yoshinaga received his B.E. degree from Kyushu University in 2009. He is a graduate student at Kyushu University. He has been engaged in image processing.

Atsushi Shimada received his M.E. and D.E. degrees from Kyushu University in 2004 and 2007. Since 2007, he has been an assistant professor in Graduate School of Information Science and Electrical Engineering at Kyushu University. He has been engaged in image processing, pattern recognition and neural networks.

Rin-ichiro Taniguchi received his B.E., M.E., and D. degrees from Kyushu University in 1978, 1980, and 1986. Since 1996, he has been a professor in Graduate School of Information Science and Electrical Engineering at Kyushu University, where he directs several projects including multiview image analysis and software architecture for cooperative distributed vision systems. His current research interests include computer vision, image processing, and parallel and distributed computation of vision-related applications.



Takayoshi Yamashita received his M.E. degree from Nara Institute of Science and Technology in 2000. Since then he has been working for OMRON Corporation. He has been engaged in face recognition and human activity sensing.

Daisaku Arita received his B.E. degree from Kyoto University in 1992 and received his M.E. and D.E. degrees from Kyushu University in 1994 and 2000. Since 2006, he has been a researcher at Institute of Systems, Information Technologies and Nanotechnologies, Fukuoka. His research interests include real-time vision system, conversational informatics, and free viewpoint video.