Special issue: 3D image and video technology

Research Paper

Object segmentation under varying illumination: stochastic background model considering spatial locality

Tatsuya TANAKA[1], Atsushi SHIMADA[2], Daisaku ARITA[3], and Rin-ichiro TANIGUCHI[4]

[1,2,4]*Kyushu University*

[3]*Institute of Systems, Information Technologies and Nanotechnologies*

ABSTRACT

We propose a new method for background modeling. Our method is based on the two complementary approaches. One uses the probability density function (PDF) to approximate background model. The PDF is estimated non-parametrically by using Parzen density estimation. Then, foreground object is detected based on the estimated PDF. The method is based on the evaluation of the local texture at pixel-level resolution which reduces the effects of variations in lighting. Fusing those approachs realizes robust object detection under varying illumination. Several experiments show the effectiveness of our approach.

KEYWORDS

Object detection, adaptive background model, illumination change, parzen density estimation, radial reach filter

1 Introduction

Background subtraction technique has been traditionally applied to object detection, which is one of the most important modules of many vision systems. It is quite useful because without prior information about the target objects, we can get object regions by subtracting a background image from an observed image. However, when a simple background subtraction method is applied to images captured under varying illumination condition, such as video-based surveillance in outdoor scenes, it often detects not only objects but also a lot of noise regions. This is because it is quite sensitive even to small illumination changes caused by moving clouds, swaying tree leaves, etc.

There have been many approaches to handle these illumination changes [1]–[10]. In principle, they are categorized into two approaches. One is off-line learning approach, where the sophisticated background model is constructed in advance by observing the target area for a certain period [2], [5], [6], [10]. The other is on-line approach, where the background model is constructed and modified as the observation proceeds [1], [3], [7], [8]. The former may robustly extract foreground objects when possible changes of illumination are included in the training data, but, of course, when there occur the illumination changes which are not observed in the learning phase, objects are not correctly detected. In principle, it is quite difficult to learn all the possible illumination changes in advance. Particularly, outdoor scenes contains many kinds of illumination changes and background changes, which makes the off-line learning very difficult.

On the contrary, the on-line approach can adapt the illumination changes by modifying the background model according to the on-going observation. Therefore, potentially, it is superior to the off-line learning approach, and, in this line, the following methods have been developed so far. One of the most popular methods is background modeling based on Gaussian Mixture Model (GMM) [1]. Shimada et al. improved this method so as that the number of Gaussians can be changed dynamically to adapt to the change of the lighting condition [7]. However, in principle, GMM cannot make a well-suited background model and cannot de-

Received October 4, 2009; Revised December 16, 2009; Accepted December 24, 2009.

[1]*tatsuya@limu.is.kyushu-u.ac.jp*, [2]*atsushi@limu.ait.kyushu-u.ac.jp*, [3]*arita@isit.or.jp*, [4]*rin@computer.org*

DOI: 10.2201/NiiPi.2010.7.4

tect foreground objects accurately when the intensity of the background changes frequently. Especially when the intensity distribution of the background has large variance, it is not easy to precisely represent the distribution with a set of Gaussians. In addition, if the number of Gaussians is increased, the computation time to estimate the background model is also increased. Thus, GMM is not powerful enough to represent the various changes of the lighting condition.

To solve the above problem, Elgammal et al. employed non-parametric representation of the background intensity distribution, and estimated the distribution by Parzen density estimation [3]. However, in their approach, the computation cost of the estimation is quite high, and it is not easy to apply it to real-time processing. Tanaka et al. proposed its fast algorithm to estimate the background intensity distribution [9]. In this method, the computational cost is greatly reduced by efficient updating algorithm of probability distribution function.

Though these methods are effective against the changes of illumination or background which are observed previously, they cannot handle sudden illumination changes because the background model is established based on statistical characteristics of observed pixel values in a certain duration.

To handle sudden changes of illumination, it is rather effective to consider invariance of features in a local region, not a single pixel. Sato et al. proposed Radial Reach Correlation (RRC for short) to evaluate foregroundness based on local texture described in the magnitude relation between the center pixel and its neighbor pixel [5]. In principle, this magnitude relation does not change under the changes of illumination and, thus, their method seems more robust than the pixel-based methods, which only use distribution information of the center pixel values. However, it cannot handle the changes of the textural information caused by the small background fluctuation such as swaying tree leaves.

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. Therefore, we propose integrated background modeling combining the pixel-level and the region-level background modelings. A method of combinational use of pixel-level and the region-level background model has been proposed by Toyama et al. [2]. However, their method uses region-level background model to complement foreground aperture. On the other hand, our method uses region-level background model to reduce noise regions which pixel-level background model detected by mistake.

2 Pixel-level background modeling

In this section, we describe the pixel-level background modeling, which represents the recent distribution of each pixel value in a certain duration. We distinguish between foreground and background referring to the observed distribution. The key issue is to estimate the distribution precisely and fast. Here, we have adopted a fast algorithm to estimate the background intensity distribution [9].

2.1 Basic algorithm

At first, we describe basic background model estimation and object detection process. The background model is established to represent recent pixel information of an input image sequence, reflecting the change of intensity, or pixel-value, distribution as quickly as possible.

We consider values of a particular pixel (x, y) over time as a “pixel process”, which is a time series of pixel values, e.g., scalars for gray values and vectors for color images. Each pixel is judged to be either a foreground pixel or a background pixel by observing the pixel process. In Parzen density estimation, or the kernel density estimation, the probability density function (PDF) of a pixel value is estimated referring to the latest pixel process, and, here, we assume that a pixel process consists of the latest N pixel values. Let $\boldsymbol{X}$ be a pixel value observed at pixel (x, y), and $\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_N\}$ be the latest pixel process. The PDF of the pixel value is estimated with the kernel estimator K as follows

$$P(\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N} K(\boldsymbol{X} - \boldsymbol{X}_i) \tag{1}$$

Usually a Gaussian distribution function $N(\boldsymbol{0}, \Sigma)$ is adopted for the estimator K[1]. In this case the equation (1) is reduced into the following formula:

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

where d is the dimension of the distribution (for example, $d = 3$ in color image pixels).

To reduce the computation cost, the covariance matrix in equation (2) is often approximated as follows:

$$\boldsymbol{\Sigma} = \boldsymbol{\sigma I} \tag{3}$$

where $\boldsymbol{\sigma}$ is a diagonal matrix whose elements represent the variance in each dimension. This means that each

[1] Here, Σ, the covariance matrix, works as the smoothing parameter.

dimension of the distribution is independent from one another. By this approximation, equation (2) is reduced into the following:

$$P(\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d}\frac{1}{(2\pi[\boldsymbol{\sigma}]_j^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\frac{([\boldsymbol{X}]_j-[\boldsymbol{X}_i]_j)^2}{[\boldsymbol{\sigma}]_j^2}\right) \quad (4)$$

where $[\boldsymbol{X}]_j$ means the j-th component of the vector and $[\boldsymbol{\sigma}]_j$ means the (j, j)-th component of the diagonal matrix. This approximation might make the density estimation error a little bigger, but the computation is considerably reduced.

The detailed algorithm of background model construction and foreground object detection is summarized as follows:

StepP-1 When a new pixel value $\boldsymbol{X}_{N+1}$ is observed, $P(\boldsymbol{X}_{N+1})$, the probability that $\boldsymbol{X}_{N+1}$ occurs is estimated by equation (4).

StepP-2 If $P(\boldsymbol{X}_{N+1})$ is greater than a given threshold, the pixel is judged to be a background pixel. Otherwise, it is judged to be a foreground pixel.

StepP-3 The newly observed pixel value $\boldsymbol{X}_{N+1}$ is kept in the "pixel process," while the oldest pixel value $\boldsymbol{X}_1$ is removed from the pixel process.

Applying the above calculation to every pixel, the background model is generated and distinction between a background pixel and a foreground pixel is accomplished.

2.2 Fast algorithm

When we estimate the generation probability of pixel value $\boldsymbol{X}$ in every frame using equation (4) and estimate the background model, its computation cost becomes quite large. To solve this problem, at first, a kernel with rectangular shape, or hypercube, is used instead of Gaussian distribution function. For example, in 1-dimensional case, the kernel is represented as follows.

$$K(u) = \begin{cases} \frac{1}{h} & \text{if } -\frac{h}{2} \le u \le \frac{h}{2} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where h is a parameter representing the width of the kernel.

Using this kernel, equation (1) is represented as follows:

$$P(\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h^d}\psi\left(\frac{||\boldsymbol{X}-\boldsymbol{X}_i||}{h}\right) \quad (6)$$

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

$$\psi(u) = \begin{cases} 1 & \text{if } u \le ||\frac{1}{2}|| \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

When an observed pixel value 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 equation (6), 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 have developed a fast algorithm to compute the PDF, whose computation cost does not depend on N [9].

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:

$$P_t(\boldsymbol{X}) = P_{t-1}(\boldsymbol{X}) + \frac{1}{Nh^d}\psi\left(\frac{||\boldsymbol{X}-\boldsymbol{X}_t||}{h}\right) - \frac{1}{Nh^d}\psi\left(\frac{||\boldsymbol{X}-\boldsymbol{X}_{t-N}||}{h}\right) \quad (8)$$

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

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

- increasing the probabilities of pixel values which are inside of the kernel located at the new pixel value $\boldsymbol{X}_t$ 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, $\boldsymbol{X}_{t-N}$ by $\frac{1}{Nh^d}$.

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

2.3 Preliminary experiment

We have evaluated computation time to process one image frame. For the proposed algorithm, we have used $h = 5$ and changed the number of N.

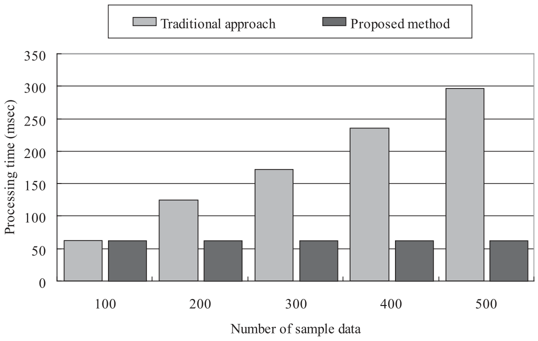


Fig. 1 The number of samples, or N, and required processing time.

Fig. 1 shows comparison between the proposed method and Elgammal's method [3]. In the Elgammal's method, the computation time is almost proportional to the length of the pixel process in which the PDF is estimated, and, from the viewpoint of real-time processing, we cannot use long image sequence to estimate the PDF. For example, when we use a standard PC environment, like our experiment, only up to 200 frames can be used for the PDF estimation in the Elgammal's method. On the other hand, in our method, when we estimate the PDF, we just update it in the local region, i.e., in the kernel located at the oldest pixel value and in the kernel located at the newly observed pixel value, and the computation cost does not depend on the length of the pixel process at all. For more detail of comparison results, refer to our paper [9].

3 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 updated according to the background changes of the input image frames.

3.1 Radial Reach Correlation (RRC)

Each pixel is judged as either the foreground or the background based on Radial Reach Correlation (RRC), which is defined to evaluate local texture similarity without suffering from illumination changes. RRC is calculated at each pixel (x, y). At first, pixels whose intensity differences to $f(x, y)$, the intensity of the pixel (x, y), exceed a threshold are searched for in every radial extension reach of 8 directions around the pixel (x, y). The searched 8 pixels are called as *peripheral pixels* hereafter. Then, the signs of intensity 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, incremental code hereafter, 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 incremental 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 $\boldsymbol{p} = (x, y)$, 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:

$$r_k = \min\{r|\ |f(\boldsymbol{p} + r\boldsymbol{d}_k) - f(\boldsymbol{p})| \geq T_P\} \tag{9}$$

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 intensity.

Based on the intensity difference between the center pixel and the peripheral pixels (defined by equation (9)), the coefficients of the incremental code of the intensity distribution around the pixel in the background image f is given by the following formula:

$$b_k(\boldsymbol{p}) = \begin{cases} 1 & \text{if } f(\boldsymbol{p} + r_k\boldsymbol{d}_k) \geq f(\boldsymbol{p}) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

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.

$$b_k{}'(\boldsymbol{p}; g) = \begin{cases} 1 & \text{if } g(\boldsymbol{p} + r_k\boldsymbol{d}_k) \geq g(\boldsymbol{p}) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

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 is calculated as follows.

$$B(\boldsymbol{p}) = \sum_{k=0}^{7}\{b_k(\boldsymbol{p}) \cdot b_k{}'(\boldsymbol{p}) + \overline{b_k(\boldsymbol{p})} \cdot \overline{b_k{}'(\boldsymbol{p})}\} \tag{12}$$

where $\overline{x} = 1 - x$ represents the inversion of a bit x. $B(\boldsymbol{p})$ represents the similarity, or correlation value, of the intensity distribution around the pixel $\boldsymbol{p}$ in the two images, and it 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 T_B can be judged as foreground pixels.

3.2 Construction of background model and foreground detection

Using RRC, the similarity between incremental encodings of a background image pixel and its corresponding pixel in the observed image is calculated, and 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 encodings of the background image in advance. However, usually, due to the illumination changes and various noises, it is almost impossible to prepare them in advance. Even if we manage to prepare such fixed background encodings, accurate results cannot be acquire, and, therefore, we should rather update the background encodings properly.

The original method mentioned in Subsection 3.1 requires a static background image to calculate RRC. In contrast, our method used an adaptive background image which is acquired by updating process of background model. One of the solutions is to use pixel-level background model described in Subsection 2.2; for example, collecting mode value on each pixel to reconstruct background image. In the region-level background modeling here, however, sudden changes of background should be reflected and the background model is constructed based on the observation of pixel values in very recent frames. From this viewpoint, it is not appropriate to use the background model which is constructed through a long-term observation. In our approach, therefore, another background model is constructed based on a single Gaussian distribution on each pixel. And the parameters of the Gaussian is updated to reflect the recent changes of observed pixel values. When RRC has to be calculated, the latest background image f is reconstructed from the mean value of Gaussian.

The update process of the Gaussian parameters is summarized as follows. Again, we represent the pixel value of pixel (x, y) at time t as d dimensional vector $\boldsymbol{X_t}$. Then, the average $\boldsymbol{\mu}_t$ and the variance $\boldsymbol{s}_t^2$ of Gaussian distribution are updated as follows:

$$\boldsymbol{\mu}_t = (1-\rho)\boldsymbol{\mu}_{t-1} + \rho \boldsymbol{X}_t \tag{13}$$

$$\boldsymbol{s}_t^2 = (1-\rho)\boldsymbol{s}_{t-1}^2 + \rho(\boldsymbol{X}_t - \boldsymbol{\mu}_{t-1})^T(\boldsymbol{X}_t - \boldsymbol{\mu}_{t-1}) \tag{14}$$

where ρ is the learning rate, which is represented in the following formula:

$$\rho = \frac{\alpha}{(2\pi)^{\frac{n}{2}}|\boldsymbol{S}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{X}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{X}_t - \boldsymbol{\mu}_t)\right) \tag{15}$$

where α is a constant parameter which does not affect the computational time, but control adjustability for illumination changes. It is possible to adapt to a sudden background change by enlarging α. Therefore, the α should be larger when illumination change frequently occurs. And S is a diagonal matrix whose elements consist of the element of s^2. Applying the above calculation to every pixel, the parameters of Gaussian distribution are updated.

The detailed algorithm of background model construction and foreground detection in the region-level modeling is summarized as follows:

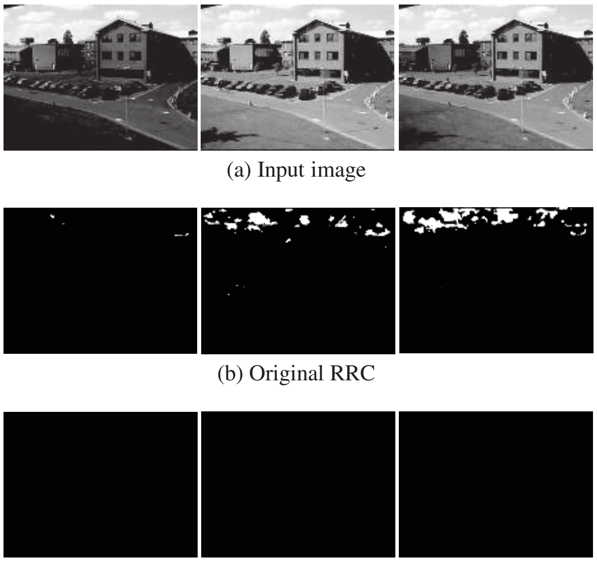
StepR-1 The background image f is created from the mean value of Gaussian distribution at each pixel.

StepR-2 RRC is constructed based on the background image f in Step1, and each pixel of the input image is judged as either the foreground or the background, referring to the threshold T_P.

StepR-3 The parameters of Gaussian distribution are updated by equation (13)~(15), if the conditions for model update is satisfied. In the other cases, the parameters are not updated. (The condition for model update will be described in Section 4.)

3.3 Preliminary experiment

We have verified the effectiveness of our region-level background model. Fig. 2 shows the result of RRCs. The illumination condition was rapidly changed. The original RRC detected cloud which had moved gradually. On the other hand, our RRC did not detect it



(a) Input image

(b) Original RRC

(c) Our RRC

Fig. 2 Comparison between Original RRC and our RRC. Left: 500th image, Center: 2000th image, Right: 4500th image.

because of updating procedure of background image which is used for reconstruction of RRC.

4 Combination of pixel-level and region-level background modelings

In this section, we describe how to combine the previous two background modelings, i.e., the pixel-level and the region-level ones. Outline of the processing flow is shown in Fig. 3.

Step-1 Using StepP-1 and StepP-2, foreground candidate pixels are detected by the pixel-level background modeling. Also, the parameters in the pixel-level modeling is updated by StepP-3.

Step-2 StepR-1, StepR-2 of the region-level background modeling is applied to the foreground candidate pixels detected in Step-1 and the final foreground pixels are detected (see Table 1).

Step-3 Based on the final result, parameters of Gaussian distributions in the region-level modeling are updated.

The above procedure is applied to every pixel in every frame, and the foreground object detection and background model construction is accomplished.

Here, combining the two modelings is simple, and pixels which are judged as foregrounds in the both modelings are finally decided as foreground pixels. As mentioned in Section 1, the pixel-level and the region-level background modelings represent different type of background pixels, and, therefore, pixels which are judged as background in either of the both modelings can be compensatively detected as background pixels. In other words, pixels judged as foreground in the both modelings should be the final result of foreground detection.

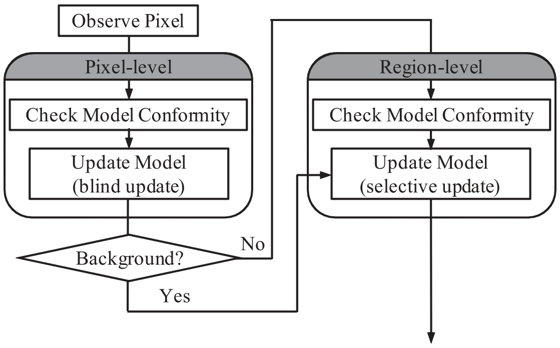


Fig. 3 Flowchart.

Next, we consider how to update the background model. In principle, there are two methods to update background models. The one is selective update, which updates the model only when the pixel is labeled as background. The other is blind update, which blindly adds every new sample to the model. The selective update basically enhances detection accuracy of the foreground, because foreground pixels are not added to the model. However, if the intensity of the occluded background of the object is changed, the occluded background can be incorrectly detected as foreground when it re-appears. This is because its current intensity is different from that of the previous one which is represented in the background model. On the contrary, although the background model can be slightly degraded, pixels judged as foreground, in the blind update, are also included in the background model and such change of the pixel value can be learned shortly.

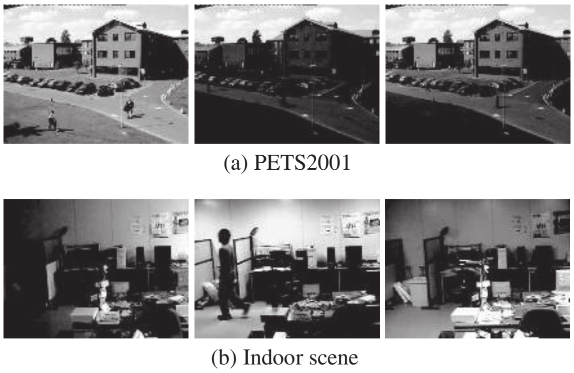
In case of the pixel-level modeling, the degradation problem is not significant since the pixel-level modeling is created by observing the pixel value for a certain duration where foreground objects do not appear very often. Therefore, we have decided to employ the blind update for the pixel-level modeling. However, the blind update is not suitable for the region-level modeling. In order to quickly adjust the pixel value changes, the region-level background modeling is designed to be sensitive for the pixel value changes, and, as a result, pixels having values which were represented shortly before in the background model tend to be incorrectly detected as foreground. Considering these effects, we use the selective update for the region-level background modeling. The parameters of the region-level modeling is updated only when the observed pixel value is finally judged as the background.

5 Experimental results

We evaluated our proposed method using PETS

Table 1 Fusion rule and selective update in the region-level background modeling.

Pixel-level	Region-level	Fused Result	Update(Region-level)
BG	—	BG	○
BG	—	BG	○
FG	BG	BG	○
FG	FG	FG	×


(a) PETS2001

(b) Indoor scene

Fig. 4 Experimental data.

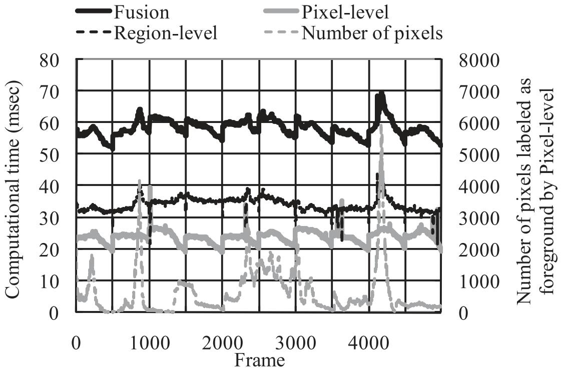


Fig. 5 Computational time of proposed method.

dataset (PETS2001)[2], which are resized into 320 × 240 pixel size, and indoor scenes of our laboratory room, which we capture as images with 320×240 pixel, 15 fps (see Fig. 4). PETS2001 dataset includes images where people are passing through streets, tree leaves are swaying, and the illumination condition is varying rapidly due to the weather condition changes. Indoor scenes include sudden and large change of illumination caused by ON and OFF of lighting. We used a PC with Intel Core2 2.66 GHz and 2 GB memory.

5.1 Computational cost

We have evaluated the processing speed of the proposed method. For the parameters of the pixel-level background modeling, we have used $N = 500$, $h = 9$. For the parameters of the region-level modeling, we have used $T_B = 6$, $T_P = 2.5\sigma$ and $\alpha = 0.05$. These parameters were decided through preliminary experiments. When we had changed the parameter *alpha*, we got almost the same result. The horizontal axis shows the frame number. Fig. 5 shows the processing speed of the proposed method. The left vertical axis shows the computational time and the right one shows the number of pixels labeled as foreground by the pixel-level modeling.

The computation time required in the pixel-level modeling is around 20 to 25 msec at every frame, and it does not change largely. This is because, in the pixel-level modeling, the probability distribution function of the pixel value is calculated by partially updating the PDF estimated in the previous frame, i.e., the probabilities which are inside of the kernel located at the oldest pixel value and at the newly observed pixel value. This computation cost is independent of the image data.

On the other hand, the computation cost required by the region-level background modeling varies according to the number of pixels labeled as foreground by the pixel-level modeling. This is because the region-level modeling is only applied to pixels judged as foreground by the pixel-level modeling. The total computational time was about 60 msec, and this is fast enough to achieve object detection in real-time.

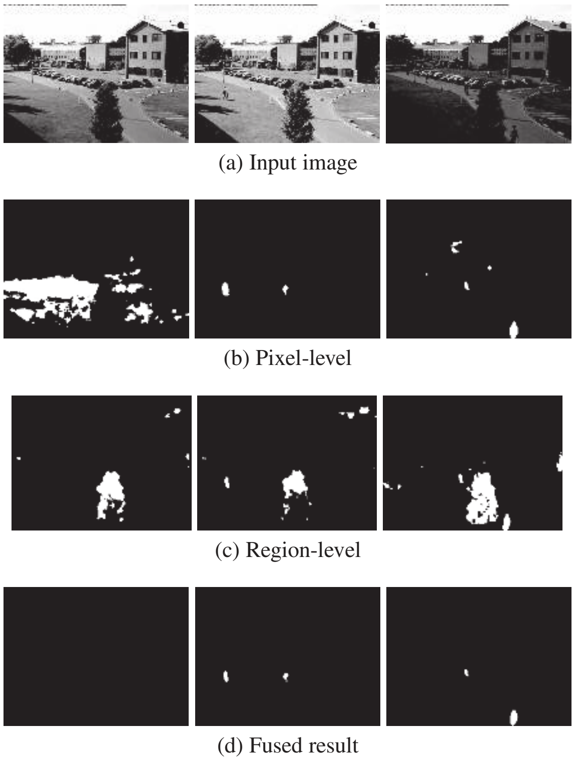
5.2 Comparison of the pixel-level and the region-level background modeling

To clarify the characteristics of the background modelings, we have compared the performance of the pixel-level and the region-level background modelings using PETS2001 dataset. It includes rapid illumination change caused by the weather condition change, swaying tree leaved, etc, and it is quite difficult to detect objects in the image sequence using simple background modeling.

Fig. 6 shows results of the experiment. Fig.6 (a), 6 (b), 6 (c), 6 (d) the input image sequence, the object regions detected by the pixel-level background modeling, ones by the region-level modeling and ones by the integrated model, respectively.

First, Fig. 6 (b) shows the pixel-level modeling could adapt the illumination changes by swaying tree leaves. However, the ground and the roof were partly misdetected, because it could not adapt sudden illumination changes. By referring to observation of pixel values in a certain period, stochastic background model is effective against periodical change of background, such as swaying tree leaves. However, it cannot handle sudden illumination changes, which is not represented in the intensity distribution observed in the previous frames. On the other hand, the region-level modeling can adapt the sudden illumination changes because it effectively exploits illumination independent local textural information (see Fig. 6 (c)). However, fluctuation

[2] Benchmark data of International Workshop on Performance Evaluation of Tracking and Surveillance, which is available from ftp://pets.rdg.ac.uk/PETS2001/.


(a) Input image

(b) Pixel-level

(c) Region-level

(d) Fused result

Fig. 6 Performance comparison between Pixel-level and Region-level.

of background caused by the swaying tree leaves destroys the invariability of local texture information, and, thus, the region-level modeling cannot handle such situations, i.e., detects such pixels as foregrounds. As shown Fig. 6 (d), integration of these approaches can handle both types of the illumination changes and realizes robust object detection under varying illumination condition.

5.3 Object detection accuracy

To evaluate the object detection accuracy, we have compared our proposed method with RRC [5], SR-feature [10], Fast Parzen density estimation [9], Adaptive Gaussian Mixture model [7]. We have examined precision and recall of foreground pixel detection on the basis of manually acquired ground truth [3].

Precision and recall are respectively defined as follows:

$$\text{precision} = \frac{\text{\# correctly detected pixels}}{\text{\# of detected pixels}} \tag{16}$$

$$\text{recall} = \frac{\text{\# of correctly detected pixels}}{\text{\# of pixels which should be detected}} \tag{17}$$

As shown in Table 2 and Table 3, the proposed method outperforms RRC, background modeling based on the fast Parzen density estimation, one on the adaptive Gaussian mixture. Compared with SR-feature method, which is robust against illumination changes and other noises, our method exhibits similar accuracy. It is important to note that SR-feature method requires explicit off-line training using many samples to acquire background model, while our method does not require the off-line training. Additionally, we evaluated the accuracy when our region-level background model was replaced by the original RRC (see the bottom row in Table 2 and Table 3). The accuracy of this method is lower than our proposed method. It became clear that our region-lebel background model performed well according to this experimental result.

Finally, Fig. 7 shows the object detection result acquired by applying our method to PETS2001 dataset. The same parameter values are used as ones in the previous experiment. For comparison, Fig. 7 (d)~ 7 (h) show results acquired by RRC, SR-feature, fast Parzen density estimation, adaptive Gaussian mixture model and combinational use of Parzen density estimation and original RRC. In RRC, the initial frame was used the background image f. In SR-feature method, a set of background images including all the possible variations of illumination changes should be trained, and, therefore, every odd frame, 2668 frames, out of all the frames, 5336 frames are used as training samples.

At first, Fig. 7 (d) indicates that RRC can adapt global illumination change. However, it cannot adapt local illumination changes caused by moving clouds, and, therefore, pixels in such condition are mis-detected as foreground.

Background modelings based on fast Parzen density estimation and on adaptive Gaussian mixture can adapt local and fluctuating illumination changes. However, they cannot adopt rapid illumination changes, and misdetect part of ground regions and building walls.

On the contrary, our proposed method and SR-feature method correctly detect object regions in both situations. Again, since SR-feature method requires explicit off-line training using many samples to acquire background model, it is clear that our method, which does not require the off-line training is much easier to use than the SR-feature method.

On the other hand, the combinational use of Parzen density estimation and the original RRC could detect object region (see Fig. 7 (d)). However, it also detects some noise regions compared with our proposed

[3] Several kinds of ground truth have been opened to the public through the web, http://limu.ait.kyushu-u.ac.jp/dataset.

Table 2 Object detection accuracy (PETS2001).

	recall	precision
Proposed method	**71.6%**	**72.6%**
Radial Reach Correlation	37.5%	22.4%
SR-Feature	64.9%	69.4%
Parzen density estimation	56.3%	51.6%
Gaussian Mixture Model	61.3%	58.2%
Parzen + RRC(original)	55.7%	63.9%

Table 3 Object detection accuracy (Indoor scene).

	recall	precision
Proposed method	**52.1%**	**60.0%**
Radial Reach Correlation	26.9%	24.9%
SR-Feature	47.8%	56.7%
Parzen density estimation	37.8%	58.5%
Gaussian Mixture Model	35.6%	46.1%
Parzen + RRC(original)	49.5%	46.4%

method.

5.4 Discussion

Through our experiment described in Subsection 5.1, 5.2 and 5.3, our proposed method performed better than traditional approaches. Considering an application of video surveillance, a system will work at 15 fps. This is fast enough to analyze observing scenes. On the other hand, the accuracy of our method was higher than other methods. Since, we have evaluated the accuracy pixel by pixel, the values of the accuracy are not high in Table 2 and Table 3. Through qualitative evaluation, we have found out that our proposed method could detect object regions, but the regions are smaller than those of ground truth. From the view point of video surveillance, however, the object size detected by our proposed method was large enough to know where each object was in the image. Whatever the reason, we will introduce such a new criterion in our future works. For example, if we define a new criterion; for example, we regard the result as successful when more than 80% of object region is detected, the result of accuracy will become larger.

6 Conclusion

In this paper, we have proposed a new method for background modeling based on the combination of non-parametric background model using Parzen density estimation and Radial Reach Correlation, which are known as a robust background subtraction method under varying illumination. In our experiment, we have got a good result that the computational time was 60 msec (about 15 fps) and the precision ratio and recall ratio were superior to the popular approaches under varying illumination.

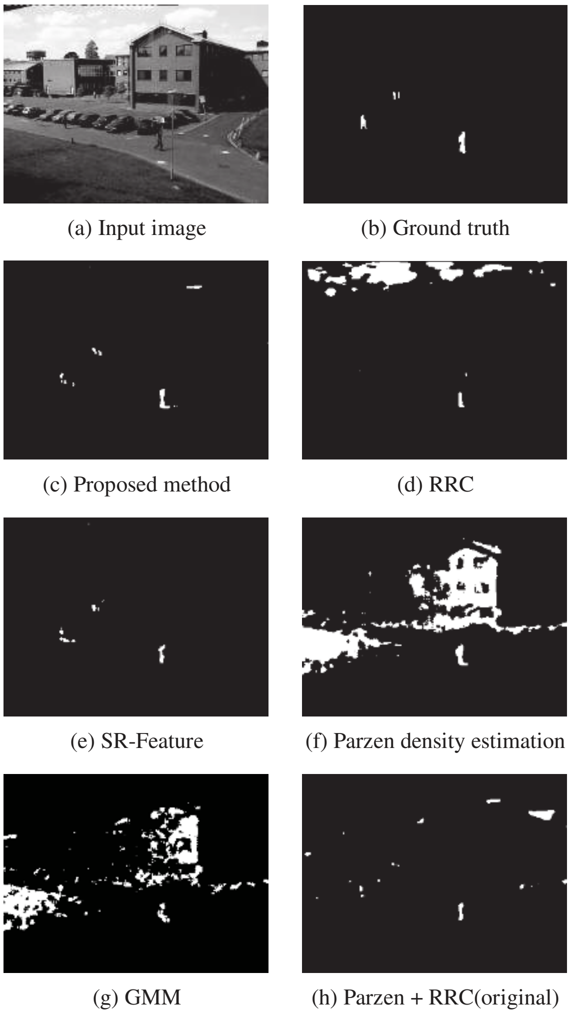

(a) Input image
(b) Ground truth
(c) Proposed method
(d) RRC
(e) SR-Feature
(f) Parzen density estimation
(g) GMM
(h) Parzen + RRC(original)

Fig. 7 Result of object detection.

Future works are summarized as follows:

- **Stabilization of computational time**
  When a sudden background change takes place or when the proportion of the area to be detected on the image becomes large, the computation cost becomes large. In other words, the computational time varies largely. This is because if the pixels are labeled as foreground by the pixel-level background modeling, they should be further examined, by the region-level modeling, whether it is foreground or background. It is not a good characteristic for real-time processing and, therefore,

we should develop a mechanism to stabilize the computation cost.

- **Cooperation between the pixel-level and the region-level modelings**
  Our combination rule of the pixel-level and the region-level modelings is rather simple and straightforward, i.e., logical AND of the results acquired by the modelings. Therefore, it is necessary to establish more sophisticated combination mechanism to make better use of the characteristics of the both models.
- **Selective update of background models**
  Our proposed method cannot detect objects which stop in the observing area. In other words, stopped objects should be regarded as foreground, but actually they gradually become background. This problem is caused by blind updating of background model. Therefore, we have to select background models which are updated or not.
- **Handling of rapid illumination changes**
  The RRC in our proposed method is robust for a certain level of illumination changes. However, if the illumination condition changes rapidly; turning light switch on/off, not only object regions but also a lot of noise regions will be detected. Therefore, it is necessary to introduce a new method which detects the rapid illumination changes between interframes.

References

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR1999)*, vol.2, pp.246–252, 1999.

[2] K. Toyama, J. Krumm, B. Brumitt and B. Meyers, "Wallflower: Principle and Practice of Background Maintenance," *Proc. of 7th Int. Conf. on Computer Vision (ICCV1999)*, pp.255–261, 1999.

[3] A. Elgammal, D. Harwood and L. Davis, "Non-parametric Model for Background Subtraction," *Proc. of 6th European Conf. on Computer Vision (ECCV2000)*, vol.2, pp.751–767, 2000.

[4] L. Li, W. Huang, I. Y. H. Gu and Q. Tian, "Statistical Modeling of Complex Background for Foreground Object Detection," *IEEE Trans, on Image Processing*, vol.13, no.11, pp.1459–1472, 2004.

[5] Y. Satoh, S. Kaneko, Y. Niwa and K. Yamamoto, "Robust object detection using a Radial Reach Filter (RRF)," *Systems and Computers in Japan*, vol.35, no.10, pp.63–73, 2004.

[6] N. Ukita, "Target-color Learning and Its Detection for Non-stationary Scenes by Nearest Neighbor Classification in the Spatio-Color Space," *Proc. of IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS2005)*, pp.394–399, 2005.

[7] A. Shimada, D. Arita and R. Taniguchi, "Dynamic Control of Adaptive Mixture-of-Gaussians Background Model," *CD-ROM Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS2006)*, 2006.

[8] E. Monari and C. Pasqual, "Fusion of Background Estimation Approaches for Motion Detection in Non-static Backgrounds," *CD-ROM Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS2007)*, 2007.

[9] T. Tanaka, A. Shimada, D. Arita and R. Taniguchi, "A Fast Algorithm for Adaptive Background Model Construction Using Parzen Density Estimation," *CD-ROM Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS2007)*, 2007.

[10] K. Iwata, Y. Sato, R. Ozaki and K. Sakaue, "Robust Background Subtraction Based on Statistical Reach Feature Method," *IEICE Trans. on Information and Systems*, vol.J92-D, no.8, pp.1251–1259, 2009.

Tatsuya TANAKA

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.

Atsushi SHIMADA

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.

Daisaku ARITA

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.

Rin-ichiro TANIGUCHI

Rin-ichiro TANIGUCHI received his B.E., M.E., and D.E. 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.