Object Detection under Varying Illumination Based on Adaptive Background Modeling Considering Spatial Locality

Tatsuya Tanaka[1], Atsushi Shimada[1], Daisaku Arita[1,2],
and Rin-ichiro Taniguchi[1]

[1] Department of Intelligent Systems Kyushu University, Japan
{tatsuya,atsushi,rin}@limu.is.kyushu-u.ac.jp
[2] Institute of Systems, Information Technologies and Nanotechnologies, Japan
arita@isit.or.jp

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. And foreground object is detected based on the estimated PDF. The other method is based on the evaluation of the local texture at pixel-level resolution while reducing the effects of variations in lighting. Fusing their approach realize 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 detection of objects in image. Without prior information about the objects, we can get object regions by subtracting a background image from an observed image. However, when simple background subtraction technique is applied to video-based surveillance which usually captures outdoor scenes, it often detects not only objects but also a lot of noise regions. This is because it is quite sensitive to small illumination changes caused by moving clouds, swaying tree leaves, etc.

There are many approaches to handle these background changes [1, 2, 3, 4, 5, 6, 7, 8, 9]. Shimada et al. proposed a background estimation method, in which mixture-of-Gaussians is used to approximate background model, and the number of Gaussians is changed dynamically to adapt to the change of the lighting condition. However, in principle, Gaussian Mixture Model (GMM) can not make a well-suited background model and can not detect foreground objects accurately when the intensity of the background changes frequently. Especially when the intensity distribution of the background is very wide, it is not easy to represent the distribution with a set of Gaussians. In addition, if the number of Gaussians

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 645–656, 2009.

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 problem, Elgammal et al employed non-parametric representation of the background intensity distribution, and estimated the distribution by Parzen density estimation [1]. 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 approach, the computational cost is greatly reduced by efficient updating algorithm of probability distribution function.

Though these methods previously described are effective against gradual or periodical change of background, they can not handle sudden illumination changes because the background model is established based on statistical characteristics of observed pixel values in a certain duration. To solve such a problem, it is effective to fuse a background model which can adapt to sudden illumination changes with the background model established according to the observation in the past. Then, in this paper, we propose an enhanced background modeling method under varying illumination with a "long-term model" and a "short-term model". The long-term model approximates the change of the pixel value such as gradual or periodical change of background, which is acquired by a long-term observation, and it is represented in a probability density function. The short-term model, on the other hand, approximates the sudden background change such as a illumination change based on Radial Reach Filter which is known as a robust background model against varying illumination [4].

2 Long-Term Model

In this section, we describe about the long-term model. The LTM represents the background in a certain duration. We use the 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 $N(\mathbf{0}, \Sigma)$ 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.

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

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

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

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:

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).
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.
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,

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

or hypercube, is used instead of Gaussian distribution function. For example, in 1-dimensional case, the kernel is represented as follows.

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

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

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

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

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

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

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, Tanaka et al have proposed a fast algorithm to compute the PDF, whose computation cost does not depend on N.

In background modeling we estimate $\mathrm{P}(\boldsymbol{X})$ 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:

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

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 the latest N pixel values are stored in the memory, which achieves quite fast computation of PDF estimation.

3 Short Term Model

In this section, we describe the short-term model (STM). STM handles short-term changes of pixel values and detects foreground objects using Radial Reach Filter, which is known as robust background subtraction method under varying illumination [4].

3.1 Radial Reach Filter (RRF)

RRF judges each pixel as either the foreground or the background based on Radial Reach Correlation (RRC), which is defined to evaluate local texture similarity at pixel-level resolution without suffering from the effects of variation in brightness.

RRC is calculated for each pixel (x, y). At first, pixels whose brightness differences to $f(x, y)$, the brightness of the pixel (x, y), exceed a threshold are searched for in every radial extension reach in 8 directions around the pixel (x, y). Then, the signs of brightness differences (positive difference or negagive difference) of the 8 pairs, each of which is a pair of one of eight found pixels and the center pixel (x, y), are represented in a binary code. The correlation value of the codes between the input image pixel and the background image pixel is regarded as a representation of their similarity.

The position of pixel (x, y) in the image is represented as the vector $\boldsymbol{p} = (x, y)$, and the directional vector $\boldsymbol{b}_k(k = 0, 1, \ldots, 7)$ is defined as follows. $\boldsymbol{d}_0 = (1, 0)^T\mathrm{C}\boldsymbol{d}_1 = (1, 1)^T$ $\boldsymbol{d}_2 = (0, 1)^T\mathrm{C}\boldsymbol{d}_3 = (-1, 1)^T\mathrm{C}$ $\boldsymbol{d}_4 = (-1, 0)^T\mathrm{C}\boldsymbol{d}_5 = (-1, -1)^T$ $\boldsymbol{d}_6 = (0, -1)^T\mathrm{C}$ and $\boldsymbol{d}_7 = (1, -1)^T$. Then the reach $\{r_k\}_{k=0}^{7}$ for these directions are defined as follows:

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

where $f(\boldsymbol{p})$ represents the pixel value of the position of $\boldsymbol{p}$ in the image, and T_P represents the threshold value of brightness difference.

Based on the brightness difference between the center pixel and the pixels selected by the reach group (defined by equation (9)), the coefficients of incremental encoding, or polarity encoding, of the brightness distribution around a pixel in the reference 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 encoding string is calculated for the input image g. Here, please note that the reach group $\{r_k\}_{k=0}^{7}$ is defined based on the reference image f.

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

Based on the obtained $b_k(\boldsymbol{p})$, ${b_k}'(\boldsymbol{p})$, the number of matches (correlation), $B(\boldsymbol{p})$ between the two incremental encodings 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. $B(\boldsymbol{p})$ represents the similarity, or correlation value, of the brightness distribution around the pixel $\boldsymbol{p}$ in the two images, and it is called Radial Reach Correlation (RRC).

Since RRC of the input image pixels and the background image pixels represents their similarities, it can be used as a measure to detect foreground pixels. In other words, pixels whose RRC is smaller than a certain threshold T_B can be judged as foreground pixels. In the following formula, the foreground detection result is represented in $C(\boldsymbol{p})$, and it is called RRC image.

$$C(\boldsymbol{p}) = \begin{cases} 1 & \text{if } B(\boldsymbol{p}) < T_B \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

3.2 Construction of Background Model and Foreground Detection

In RRC, the similarity between incremental encoding of the background image and the input image is calculated referring to the reach group defined in the background image f, and foreground is detected based on the similarity. In principle, it is possible that a fixed background image can be prepared in advance, if the background does not change. However, if the background often changes, using such fixed background image does not produce an accurate result. Rather, we should update the background image properly. In STM, sudden changes of background should be reflected and the background model is constructed based on the observation of pixel values in very recent frames. In our approach, the change of the background is represented in a single Gaussian distribution at every pixel. Then, the average and the variance of the distribution are used to represent the background image f, and RRC is calculated referring to f.

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 $\boldsymbol{\sigma}_t^2$ of Gaussian distribution are updated as follows:

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

$$\boldsymbol{\sigma}_t^2 = (1-\rho)\boldsymbol{\sigma}_{t-1}^2 + \rho(\boldsymbol{X}_t - \boldsymbol{\mu}_{t-1})^T(\boldsymbol{X}_t - \boldsymbol{\mu}_{t-1}) \tag{15}$$

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

$$\rho = \frac{\alpha}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{X}_t - \boldsymbol{\mu}_t)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}_t - \boldsymbol{\mu}_t)\right) \tag{16}$$

α is a constant parameter, or an internal learning rate, and it is possible to adapt to a sudden background change by enlarging α. Applying the above calculation to every pixel, the parameters of Gaussian distribution are updated.

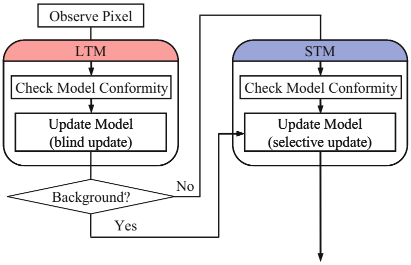


Fig. 1. Flowchart

Table 1. Fusion rule and selective update of STM

LTM	STM	Fused Result	Update(STM)
BG	—	BG	○
BG	—	BG	○
FG	BG	BG	○
FG	FG	FG	×

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

Step 1: The background image f is created from the mean value of Gaussian distribution in each pixel.
Step 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. Here, we set the threshold T_P is 2.5σ.
Step 3: The parameters of Gaussian distribution are updated by equation (14)∼(16), if the conditions for model update in STM is satisfied. In the other cases, the parameters are not updated. (The condition for model update in STM will be described in section4.)

4 Fusion of LTM and STM

In this section, we describe the fusion rule of LTM and STM. The processing flow is shown in figure 1. First of all, background subtraction is done according to the long-term model. Then if a pixel is labeled as foreground, the pixel is examined whether it is the foreground or the background referring to the short-term model. Finally, the pixel is regarded as foreground by the rule described in table 1.

Next, we describe how to update the background model. In general, 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 adds every new sample to the model. The selective update enhances detection accuracy of the foreground, because foreground pixels are not added to the model. However, for instance, if the background changes while the object has been detected, regions which should be regarded as the background keep being detected as the foreground. The blind update, on the other hand, allows foreground objects to be added to the background model. In case of the long-term model, this drawback is not significant since LTM is created by observing the pixel value for a long time. Therefore, we have decided to use the blind update for the long-term model. On the other hand, the blind update is not suitable for the short-term model since it is very sensitive for the changes of the pixel values. Considering these effects, we use the selective update for the short-term model. In particular, only when the observed pixel value is finally judged as the background, the short-term model is updated.

5 Experimental Results

5.1 Computational Cost

To evaluate the computational time to process one image frame, we have used data set of PETS(PETS2001)[2] after the image resolution was reduced into 320×240 pixels. The data set includes images in which people are passing through streets, tree leaves are flickering, and the illumination condition are varying rapidly. For the evaluation of computation speed, we have used a PC with a Pentium IV 3.2GHz and 2.0GB memory.

Figure 2 shows the processing speed of the proposed method. Where, for the parameters of LTM, we have used $N = 500$, $h = 9$. For the parameters of STM, we have used $T_B = 6$ and $T_P = 2.5\sigma$. The horizontal axis shows the frame number. The left vertical axis shows the computational time and the right one shows the number of pixels labeled as foreground by LTM.

The computation cost to maintain LTM is 23msec/frame in average, and it does not change largely. On the other hand, the computation cost in STM varies according to the number of pixels labeled as foreground by LTM. This is because STM was only applied to pixels judged as foreground by the LTM. The total computational time was about 60msec, and this is enough to achieve object detection in real-time.

5.2 Comparison of Characteristics between LTM and STM

We have verified the characteristics of LTM and STM. To compare the characteristics of each model, we have conducted experiments of object detection by LTM, STM and their fused model. The data set includes images in which tree leaves are flickering, and the illumination condition are varying rapidly.

Figure 3 shows results of the experiment. Figure 3(a), 3(b), 3(c) and 3(d) show the input image sequence, the object areas detected by LTM, ones by STM and ones by the fused model, respectively.

First, figure 3(b) shows LTM could adapt the background changes such as flickering tree leaves. However, ground and the roof were mis-detected, because it could not adapt sudden illumination changes. Thus though stochastic adaptive background model is effective against gradual or periodical change of background, it can not handle sudden illumination changes because the background model is established based on statistical characteristics of observed pixel values in a certain duration. On the other hand, STM could adapt the sudden illumination changes. However, it could not handle a background changes such as ones caused by flickering of tree leaves. Because STM is based on the evaluation of

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

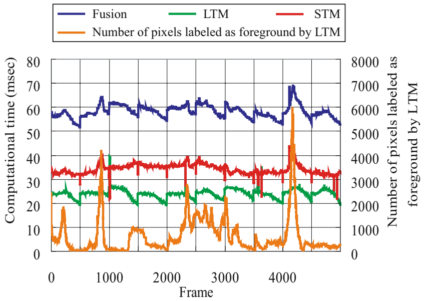
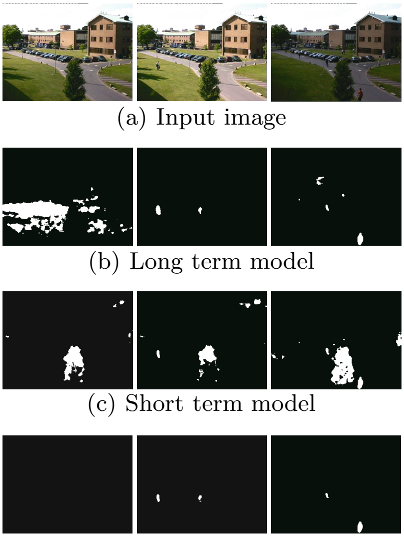


Fig. 2. Computational time of proposed method



(a) Input image

(b) Long term model

(c) Short term model

(d) Fused result

Fig. 3. Performance comparison between LTM and STM

the local texture, though it is effective against sudden illumination changes, it can not handle such the background changes. As shown Figure 3(d), fusing these approaches realizes robust object detection under varying illumination condition.

5.3 Object Detection Accuracy

To evaluate the object detection accuracy, we have used two scenes shown in Figure 4. One of them is an outdoor scene(PETS2001) which was used in Section 5.1. And the other is an indoor scene, which we took a rate of 15fps. The image resolution is 320 × 240 pixes. The indoor scene includes sudden illumination changes caused by turning off and on the light. Using these data sets, we have examined precision and recall of object detection on the basis of ground truth[3].

Precision and recall are respectively defined as follows:

$$\text{precision} = \frac{\#\ \text{correctly detected pixels}}{\#\ \text{of detected pixels}} \tag{17}$$

$$\text{recall} = \frac{\#\ \text{of correctly detected pixels}}{\#\ \text{of pixels which should be detected}} \tag{18}$$

Outdoor scene and indoor scene were composed of about 5000 frames and about 3000 frames respectively, and their first 500 frames are used for initialization.

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

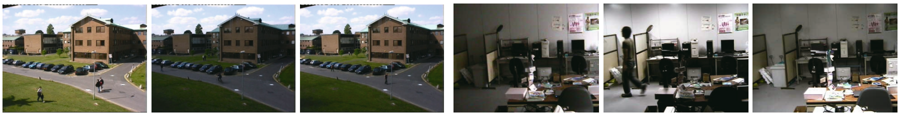

(a) Outdoor scene (PETS2001)
(b) Indoor scene

Fig. 4. Experimental data

Table 2. Object detection accuracy

	Outdoor scene		Indoor scene	
	Recall	Precision	Recall	Precision
Proposed method	**71.6%**	**72.6%**	**52.1%**	**60.0%**
Radial Reach Filter	37.5%	22.4%	26.9%	24.9%
Gaussian Mixture Model	61.3%	58.2%	35.6%	46.1%
Parzen density estimation	56.3%	51.6%	37.8%	58.5%

The recall and the precision were evaluated in the rest of the data. Ground truth represents regions that should be detected as object regions. Test data are extracted in every 15 frames and their ground truth is added manually.

Table 2 shows the average accuracy of the proposed method, RRF [4], Gaussian Mixture Model [8] and adaptive background model based on Parzen density estimation [9]. This table also shows the proposed method outperformed the other methods.

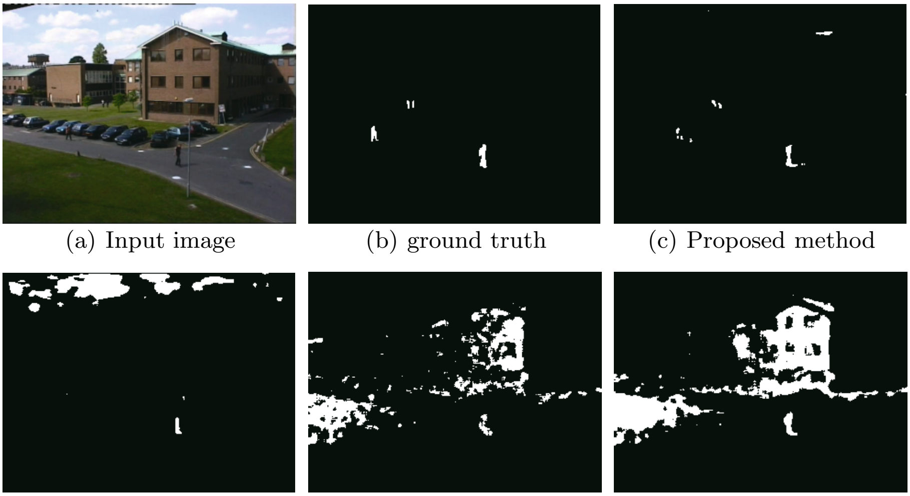

(a) Input image
(b) ground truth
(c) Proposed method
(d) RRF
(e) GMM
(f) Parzen density estimation

Fig. 5. Object detection by the proposed method, Radial Reach Filter, Gaussian Mixture Model and Adaptive background model based on Parzen density estimation

Figure 5 shows results of object detection by the proposed method. Figure 5(a) $\sim$ 5(f) show the input image , ground truth, the object areas detected by our approach, the result in case of RRF, the one of Gaussian Mixture Model and the one of adaptive background model based on Parzen density estimation, respectively. For the proposed method, we have used $N = 500$, $h = 9$, $T_B = 6$, $T_P = 2.5\sigma$. Where, an initial frame was used as a background image f in RRF. This result shows that, RRF can not handle the background change such as movements of cloud since the background model was created at the first frame and was not updated in the later process. As a result, some non-object region had been mis-detected. On the other hand, though Gaussian Mixture Model and the adaptive background model based on Parzen density estimation could handle those background changes, the ground and the wall of building had been mis-detected because they were not possible to adapt sudden illumination changes. Our method, which combines two complementary approaches properly, could detect object regions robustly compared with the other methods.

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 Filter, which is known as a robust background subtraction method under varying illumination. In our experiment, we have got a good result that the computational time was 60msec (about 15fps) and the precision ratio and recall ratio were superior to the traditional approaches under varying illumination.

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 LTM, they should be further examined, by STM, whether it is weather 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 Long-term model and Short-term model**
  Our combination rule of LTM and STM is rather simple and straightforward, i.e., logical AND of the results acquired by LTM and STM. Therefore, it is necessary to establish more sophisticated combination mechanism to make better use of the characteristics of the both models.

References

1. Elgammal, A., Harwood, D., Davis, L.: Non-parametric Model for Background Subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)

2. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principle and Practice of Background Maintenance. In: International Conference on Computer Vision, pp. 255–261 (1999)
3. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical Modeling of Complex Background for Foreground Object Detection. IEEE Transactions on Image Processing 13(11), 1459–1472 (2004)
4. Satoh, Y., Kaneko, S., Niwa, Y., Yamamoto, K.: Robust object detection using a Radial Reach Filter (RRF). Systems and Computers in Japan 35(10), 63–73 (2004)
5. Monari, E., Pasqual, C.: Fusion of Background Estimation Approaches for Motion Detection in Non-static Backgrounds. In: CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (2007)
6. Ukita, N.: Target-color learning and its detection for non-stationary scenes by nearest neighbor classification in the spatio-color space. In: Proceedings of IEEE International Conference on Advanced Video and Signal based Surveillance, pp. 394–399 (2005)
7. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 246–252 (1999)
8. Shimada, A., Arita, D., Ichiro Taniguchi, R.: Dynamic Control of Adaptive Mixture-of-Gaussians Background Model. In: CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (2006)
9. Tanaka, T., Shimada, A., Arita, D., Ichiro Taniguchi, R.: A Fast Algorithm for Adaptive Background Model Construction Using Parzen Density Estimation. In: CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (2007)