IPSJ Transactions on Computer Vision and Applications Vol. 3 148–159 (Dec. 2011)

Research Paper

Maintenance of Blind Background Model for Robust Object Detection

Atsushi Shimada,[†1] Satoshi Yoshinaga[†1] and Rin-ichiro Taniguchi[†1]

An adaptive background model plays an important role for object detection in a scene which includes illumination changes. An updating process of the background model is utilized to improve the robustness against illumination changes. However, the process sometimes causes a false-negative problem when a moving object stops in an observed scene. A paused object will be gradually trained as the background since the observed pixel value is directly used for the model update. In addition, the original background model hidden by the paused object cannot be updated. If the illumination changes behind the paused object, a false-positive problem will be caused when the object restarts to move. In this paper, we propose 1) a method to inhibit background training to avoid the false-negative problem, and 2) a method to update an original background region occluded by a paused object to avoid the false-positive problem. We have used a probabilistic approach and a predictive approach of the background model to solve these problems. The great contribution of this paper is that we can keep paused objects from being trained by modeling the original background hidden by them. And also, our approach has an ability to adapt to various illumination changes. Our experimental results show that the proposed method can detect stopped objects robustly, and in addition, it is also robust for illumination changes and as efficient as the state-of-the-art method.

1. Introduction

A technique of background modeling has been widely applied to foreground object detection from video sequences. It is one of the most important issues to construct a background model which is robust for various illumination changes. Many approaches have been proposed to construct an effective background model; pixel-level approaches [1)–5)], region-level approaches [6),7)], combinational approaches [8),9)] and so on. Almost all of these approaches have a common process of updating the background model. Actually, this process is very beneficial to adapt to various illumination changes. On the other hand, we can say that the traditional background model has an ability to detect "Moving Objects" only. In other words, we cannot apply the traditional background model for scenes such as surveillance of an intersection, a bus stop and so on where some vehicles stops around there. Also, it cannot satisfy the demand of abandoned object detection. The update process of the background model sometimes causes a FN (false-negative) problem when a foreground object stops in the scene. This is because the paused foreground object is gradually learned as the background by the blind updating process. Therefore, we have to handle the following problems (also see **Fig. 1**) in order to keep detecting the paused object.

(1) Over-training of foreground objects
(2) Wrong detection of original background regions

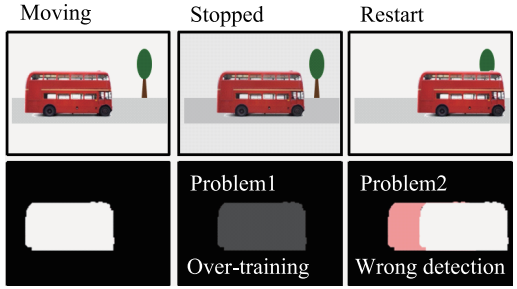

Fig. 1 Problem of blind updating of background model.

The first problem is caused by the blind updating process of background model. Some researches tried to solve this problem by controlling the learning rate of the background model. For example, decreasing the learning rate of some regions in which foreground objects probably stop [10)] or utilizing two background models which have different learning rates [11)] have been proposed. However, these approaches have not resolved the essential problem of over-training since they just extend the time for being learned as background.

The second problem is caused by a paused foreground object when it starts to move again. In such a case, an original background region hidden by the object might be detected wrongly since the paused foreground object has been included

†1 Kyushu University

in the background model. We have to consider another possibility that what will happen if we stop the updating process of foreground regions. In such a case, the FP problem will be caused when an illumination change occurs while the foreground object stops. The hidden region will be detected wrongly since the background model does not know the illumination change occurred in the hidden region. Although the literature [12] proposed an approach which considers an illumination change until a foreground object is regarded as background, it does not handle the illumination change (background change) in the region hidden by the paused foreground object.

In this paper, we propose a novel approach to maintain a background model and tackle the problems mentioned above. Our approach does not stop the update process of the background model even if a pixel is regarded as foreground. Alternatively, a substitute pixel is searched from background pixels and it is used for the model update. This alternative process offers the following advantages.

(1) Our approach can control over-training of paused foreground objects without adjusting the learning rate.
(2) Our approach can update the original background region hidden by paused objects.

There are two main issues to be tackled in our study. One is how to detect object regions precisely under the condition of illumination changes. The other is how to search an appropriate pixel used for the model update. We employ two different kinds of models; one is a probabilistic background model and the other is a predictive model. These models are efficiently combined and used to resolve the two main issues.

2. Framework

The processing flow of our proposed background model is shown in **Fig. 2**. At the first stage (Step 1 in Fig. 2), two kinds of background models; a probabilistic model and a predictive model, are used to calculate the background likelihood. The background likelihoods are integrated in the next step (Step 2) to acquire the object region precisely. Therefore, it is required for each model to output a hypothesis of the background by using its own property for illumination changes. The probabilistic model can generate multi-modal background changes caused by waving trees, fleeting clouds and so on. Therefore, it can reduce the false-positive pixels caused by such changes. The basic idea of the probabilistic model is inspired by the literature [3]. We have modified the original background model to output background likelihood for each pixel value. We will explain about the probabilistic background model in Section 3.1. On the other hand, the predictive model estimates the pixel value which will be observed at the next frame and it has a robust character with respect to illumination changes caused by a weather condition change. According to the literature [13], we can enhance the robustness against illumination changes when the predictive models are used in combination in a small region. In this research, we also introduce the region-level approach with the predictive models. We will explain about the predictive model in Section 3.2.

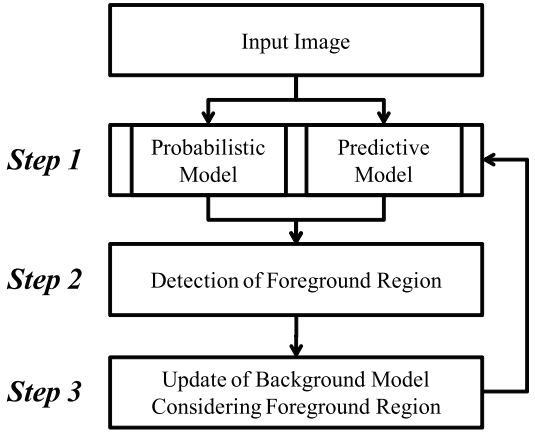

Fig. 2 Processing flow of the proposed method.

At the second stage (Step 2 in Fig. 2), the foreground region is determined by integrating two background likelihoods evaluated by the probabilistic background model and the predictive model. Each background outputs higher likelihoods for a background region including illumination changes, and lower likelihoods for object regions at the previous step. Therefore, we have to combine them appropriately to discriminate between the foreground and the background. We define an energy function based on Markov Random Field (MRF) and give each pixel

a proper label (foreground or background) by minimizing the energy function, this idea is inspired by the literature [14]. The detailed explanation will be given in Section 4.

Finally, at the third stage (Step 3 in Fig. 2), the parameters of both models are updated. Generally, the observed pixel value is directly used for updating the parameters. In our approach, meanwhile, when a pixel is judged as "foreground" at the second stage, we use an alternative background pixel value around the pixel which has similar background model. The idea comes from the assumption that a similar pixel value will be observed in the background regions which have a similar background models. The process avoids the foreground object being trained as "background" since the pixel values on the foreground region are not introduced into the background model. Therefore, a paused object will never be put to the background. We will give a detailed explanation in Section 5.

3. Probabilistic Model and Predictive Model

In this section, we explain about the probabilistic background model and the predictive model.

3.1 Probabilistic Model based on GMM

We have modified the GMM-based background model [3]. The modified background model consists of 2 steps; the evaluation of the background likelihood and the update of model parameters.

3.1.1 Evaluation of Background Likelihood

Let x_i^t be a pixel value on a pixel i at frame t. For simplicity, we omit the notation i when we explain each pixel process. The background likelihood is represented as

$$P(x^t) = \sum_{k=1}^{K} w_k^t \eta(x^t|\mu_k^t, \mathbf{\Sigma}_k^t) \tag{1}$$

where K is the number of distributions. The w_k^t, μ_k^t and $\mathbf{\Sigma}_k^t$ are an estimate of the weight, the mean value and the covariance matrix of the k^{th} Gaussian in the mixture at frame t respectively. The η is a Gaussian probability density function represented as follows.

$$\eta(x^t|\mu^t, \mathbf{\Sigma}^t) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^t-\mu^t)^T\mathbf{\Sigma}^{-1}(x^t-\mu^t)\right) \tag{2}$$

The covariance matrix Σ is actually assumed as follows.

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

This assumes that the red, green, and blue pixel values are independent and have the same variances.

The original approach [3] judges whether or not an observed pixel value belongs to the "background." Our approach does not output such a judgment result explicitly. Instead, we calculate the background likelihood at this processing stage.

3.1.2 Update of Model Parameters

The model parameters are updated in the same way as the original method [3].

The weights of the K distributions at frame t, w_k^t, are adjusted as follows

$$w_k^t = (1-\alpha)w_k^{t-1} + \alpha M_k^t \tag{4}$$

where α is the learning rate and M_k^t is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized.

Every new pixel value x^t is examined against the existing K Gaussian distributions, until a match is found. A match is defined as a pixel value within 2.5 standard deviations of the distribution. The parameters of unmatched distributions remain the same. When a match is found for the new pixel value, the parameters of the distribution are updated as follows.

$$\mu^t = (1-\rho)\mu^{t-1} + \rho y^t \tag{5}$$

$$\sigma^t = (1-\rho)\sigma^{t-1} + \rho(y^t-\mu^t)^T(y^t-\mu^t) \tag{6}$$

The second learning rate ρ is given by

$$\rho = \alpha\eta(y^t|\mu^t, \sigma^t) \tag{7}$$

where y^t is a pixel value which is used for the update of model parameters. We purposely distinguish the notation y^t from x^t since the pixel value y^t depends on the judgment result whether or not the pixel belongs to the background. When the pixel is in the background, the observed pixel value is directly used for the model update. Otherwise, we select a pixel value which is used for the model update from another pixel. The detailed explanation about selecting such an alternative pixel will be given in Section 5.

If none of the K distribution matches the current pixel value, a new Gaussian

distribution is made as follows.

$$w_{k+1}^t = W \tag{8}$$
$$\mu_{k+1}^t = y^t \tag{9}$$
$$\sigma_{k+1}^t = \sigma_k^t \tag{10}$$

where W is the initial weight value for the new Gaussian. If W is higher, the distribution is chosen as the background model for a long time. After this process, the weights are renormalized. Finally, when the weight of the least probable distribution is smaller than a threshold, the distribution is deleted, and the remaining weights are renormalized.

3.2 Predictive Model based on Exponential Smoothing

3.2.1 Exponential Smoothing

We use an exponential smoothing method [15] to acquire a predictive pixel value z^t. Exponential smoothing is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make forecasts. The simplest form of exponential smoothing is given by the following formula.

$$m^t = \beta x^t + (1-\beta) m^{t-1} \tag{11}$$

where m^t is the estimate of the value, x^t is the observed value at frame t. β is the smoothing constant in the range $(0 \leq \beta \leq 1)$.

When there is no trend in time series data, m^t is a good estimate value at frame $t+1$. In single exponential smoothing, the forecast function is simply the latest estimate of the level. If a slope component is now added, whose the estimate value itself is updated by exponential smoothing, the trend can be taken into account. The forecast function, which gives an estimate of the series can be written as follows:

$$z^t = m^t + \frac{1-\beta}{\beta} r^{t-1} \tag{12}$$

where r^t is the current slope and z^t is the estimate of the value with a trend. Since the previous estimate of the value is already known, it is possible to update the estimate of the slope by the following formula.

$$r^t = \beta(z^t - z^{t-1}) + (1-\beta) r^{t-1} \tag{13}$$

3.2.2 Evaluation of Background Likelihood

The predictive model mentioned above is used for two purposes. One is for searching a pixel which has a similar tendency with the pixel hidden by a foreground object, which will be explained in Section 5. The other is for the region-level background model explained in this section. Some papers have reported that the spatial locality information is effective for illumination changes [7),16)]. This idea is derived from a hypothesis that similar changes will be observed around the pixels when an illumination change occurs. In the proposed method, we use not only the predictive value of the target pixel but also the values of four-neighbor pixels simultaneously in order to evaluate the background likelihood.

Let R be a set of four-neighbor pixels around pixel i, the background likelihood $Q(x^t)$ is calculated by the following formula.

$$Q(x_p^t) = \frac{\sum_{i \in R} \phi(x_i^t, z_i^t)}{|R|} \tag{14}$$

The $\phi(x^t, z^t)$ is a range which allows a predictive error, which is defined as follows.

$$\phi(x^t, z^t) = \begin{cases} 1 & \text{if } |x^t - z^t| < th \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

If the difference is smaller than th between the observed value and the predictive value, we regard the prediction as a success. The background likelihood will become higher when a larger number of success predictions is obtained.

3.2.3 Update of Model Parameters

The parameters of the predictive model are updated frame by frame. In the same way with the probabilistic background model, we decide whether or not to use the observed value directly for the model update. The detailed explanation will be given in Section 5.

4. Foreground Detection based on MRF

The probabilistic model and the predictive model output the background likelihood for each pixel. In other words, each pixel has two background likelihoods. Here, they are combined to determine the foreground or the background. We define an energy function based on Markov Random Field (MRF) and give each pixel the proper label (foreground or background) by minimizing the energy function. Our energy function is defined as

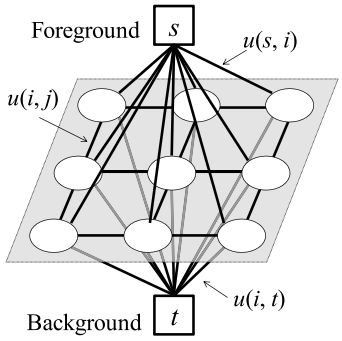

Fig. 3 Graph structure.

$$E(\boldsymbol{L}|\boldsymbol{x}) = \lambda \sum_{i \in \mathcal{V}} G(l_i|x_i) + \sum_{(i,j) \in \mathcal{E}} H(l_i, l_j|x_i, x_j) \tag{16}$$

where $\boldsymbol{L} = (l_1, \ldots, l_N)$ is the array of labels, $\boldsymbol{x}$ is a set of pixel values and N is the number of pixels. $\mathcal{V}$ and $\mathcal{E}$ represent the set of all pixels and the set of all nearest four-neighboring pixel pairs respectively. $G(l_i)$ and $H(l_i, l_j)$ represent the penalty term and the smoothing term respectively and they are calculated as follows.

$$G(l_i|x_i) = \frac{P(x_i) + Q(x_i)}{2} \tag{17}$$

$$H(l_i, l_j|x_i, x_j) = \frac{1}{\ln(|x_i - x_j| + 1 + \epsilon)} \tag{18}$$

$G(l_i)$ is the combined background likelihood. Both the probabilistic model and the predictive model output a higher background likelihood, the term becomes larger. Here, we combined two likelihood with the same weight (i.e., each likelihood is not weighted to be combined). Through some experiments, we found out that such s weighed approach didn't affect the result so much compared with other parameters. Therefore, we avoid the weighted combination to avoid the unnecessary adjustment of magic parameters. The smoothing term $H(l_i, l_j)$ is calculated by examining the similarity between adjacent pixels. Here, we use the color similarity. Therefore, the same label (foreground or background) is likely to be given to pixels which have similar colors.

To minimize the total energy $E(\boldsymbol{L}|\boldsymbol{x})$, we use a graph cut algorithm [17]. We make a graph which has two terminal nodes (Source (s) (Foreground) and Sink (t) (Background)) and some nodes corresponding to pixels (see **Fig. 3**). Edges are made between nodes. We give each edge a cost $u(i, j)$ defined as follows.

$$u(i, j) = H(l_i, l_j|x_i, x_j) \tag{19}$$

$$u(s, i) = \lambda(1 - G(l_i|x_i)) \tag{20}$$

$$u(i, t) = \lambda G(l_i|x_i) \tag{21}$$

5. Update of Model Parameters

If we directly use observed pixel values for the model update process, not only background regions but also foreground regions are gradually trained by the model. It will cause a FN (false negative) problem when a moving object stops in the scene (e.g., a bus stop, an intersection and so on). One of the solutions is to exclude foreground pixels from the update process. However, such an ad-hoc process will generate another problem that the background model on the foreground pixel cannot adapt itself to illumination changes while the foreground object stops. As a result, when the paused object starts to move again, the occluded region will be detected wrongly (FP (false positive) problem). To solve this problem, our approach updates model parameters on the foreground pixels with the help of background pixels.

The specific update process of our proposed approach is as follows. Let F and B be a set of foreground pixels and background pixels judged in Section 4 respectively, the pixel value y_i^t for the model update is calculated as

$$y_i^t = \begin{cases} x_i^t & \text{if } i \in B \\ x_c^t & \text{if } i \in F \end{cases} \tag{22}$$

where $c \in B$ is a pixel which satisfies the following condition.

$$c = \underset{j \in B}{\operatorname{argmin}} f(\Theta_i, \Theta_j) \tag{23}$$

Θ is a set of parameters of the probabilistic model and the predictive model on each pixel. In our experiments, we set Θ to be $\Theta^t = \{\mu_1^t, m^t, r^t\}$, which denotes the average background pixel value of the distribution which has the largest

weight μ_1^t, the exponential smoothing m^t and the slope of the observed value r^t. The most important contribution in this paper is to use x_c^t for the model update. When a pixel is judged as belonging to the foreground, our approach searches the model which has the most similar model parameters with the pixel. The similarity between model parameters is evaluated by the distance function $f(\Theta_i, \Theta_j)$, where we use the L1 norm in our experiments.

In this way, our approach does not use foreground pixel values to update model parameters. Alternatively, we use the pixel value on the background pixel whose model parameters are the most similar with the one on the foreground pixels. This procedure avoids the foreground object from being trained as background. Therefore, even if a foreground object stops in the scene, our approach keeps detecting the foreground object. In addition, the implicit update process of the background models hidden by the foreground object reduces the FP problem when the paused object starts to move again.

6. Experimental Results

6.1 Conditions

We have used several public datasets to investigate the effectiveness of our proposed method. The computational speed of the proposed method was 7 fps for QVGA image size by using a PC with a Core i7 3.07 GHz CPU.

According to our preliminary experimental results, we set some parameters as follows; $\alpha = 0.5$, $\beta = 0.5$, $th = 15$ ⋆1. These parameters were common to following experiments.

6.2 Evaluation of Implicit Model Update

The dataset used in this section is released at PETS2001 ⋆2 including illumination changes in the outdoor scene. We have clipped two subscenes from the original image sequence; one is a scene in which the illumination condition changes from dark to bright, and the other is a scene from bright to dark. Both scenes consist of about 600 frame images. Moreover, we have selected two 10×30 pixel areas; an area with a simple background and an area with a complex background (see **Fig. 4** for details). We have conducted a simulation experiment under the assumption that a quasi-foreground object stopped on the 10×30 pixel region. The background models on the pixels are processed by three competitive methods; the method without update, the traditional background update method and the proposed method. We evaluated how effectively the proposed method updated the pixels hidden by the quasi-foreground object.

Fig. 4 Clipped scenes around illumination changes and examined regions with a simple or a complex background.

Table 1 Comparison of model update methods: "B to D" denotes Bright to Dark, "D to B" denotes Dark to Bright.

		B to D Simple BG	B to D Complex BG	D to B Simple BG	D to B Complex BG
Without Update	Error	102.8	60.9	105.5	63.2
	FP	250	99	250	106
Traditional Update	Error	9.2	8.4	12.0	10.0
	FP	0	0	0	0
Proposed Method	Error	14.0	23.8	14.8	29.6
	FP	7	6	0	13

Table 1 shows the error value and the number of FP pixels around illumination changes. The error value means the difference value between the estimated value of the background model and the observed pixel value. The estimated value was acquired by the mean value of the Gaussian distribution which has the highest

⋆1 The parameter $\alpha = 0.5$ and $\beta = 0.5$ affect the quickness of adaptation to illumination changes. If we set them to larger values, a new observed pixel value will be trained as background soon. On the other hand, large values of these parameters cause the false-positive problem since pixel values observed in the past tend to be forgotten because of the quick adaptation for a new pixel value. Therefore, we have to determine appropriate values through some preliminary experiments.

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

probability in GMM. Meanwhile, we counted up the number of pixels whose error value exceeded a threshold as FP pixels. When we did not update model parameters, the error value was large and a lot of FP pixels were detected wrongly (see "Without Update" row in Table 1). On the other hand, our proposed method could also adapt to the illumination changes even though the investigated area was occluded by the quasi-foreground object. The error value in the complex background became larger than the one in the simple background. However, this did not lead to a sensible increase of the number of FP pixels. These observations applied to both scenes; the scene from dark to bright and the scene from bright to dark. Finally, we show the results when there was no object on the target region. In such a case, the background models on the region should be updated appropriately. We call such a strategy as "Traditional Update" in Table 1. We can see that the error value was almost the same between the "Traditional method" and the "Proposed Method." In other words, the proposed method could update a hidden background model even though it did not use the observed pixel values in the target region. Therefore, we could conclude that the implicit update process of the background model provides us a good solution for updating the region hidden by foreground objects.

6.3 Accuracy of Paused Object Detection

We have used three outdoor scenes ⋆1 to investigate the detection accuracy of paused foreground object regions. The Scene 1, Scene 2 and Scene 3 in **Fig. 5** shows the snapshots of about 100^{th} frame, 60^{th} frame and 150^{th} frame after the moving object stopped. The illumination condition in Scene 1 is relatively stable compared with the other scenes. We prepared six competitive methods to evaluate the effectiveness.

GMM GMM based method [3]

Fusion Model fusion model of spatial-temporal features [8]

Method 1 proposed method without implicit model update and graph cut ⋆2

Method 2 proposed method without implicit model update

Method 3 proposed method without graph cut

Proposed Method background model proposed in this paper

The parameters in these competitive methods were set to be the same as in the original papers or defined in this paper as mentioned above. We have evaluated the accuracy by the precision ratio, the recall ratio and the F-measure given by the following formulas.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{24}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{25}$$

$$\text{F} = 2/\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right) \tag{26}$$

The F-measure indicates the balance between the precision and the recall. A larger value means a better result. The TP, FP and FN denote the number of pixels detected correctly, detected wrongly, undetected wrongly respectively.

Figure 5 shows the evaluated images, and **Table 2** shows the evaluation results. The GMM based method [3] could detect just a few foreground pixels since it had learned the paused foreground object as "background." The fusion model [8], Method 1 and Method 2 also gradually learned the foreground objects as "background." This is why the recall ratios of these methods were very low in all the scenes. On the other hand, Method 3 and our proposed method gave a much better recall ratio than the competitive methods. The F-measures of them were also superior to the others. From these results, we confirmed that our implicit model update process was very effective for detection of paused foreground objects. Compared to Method 3, our proposed method provided a higher accuracy in both the precision and the recall. Method 3 detected not only paused objects but also many salt-and-pepper noises. The Graph cut algorithm provided a smoothing effect which reduced these noises.

Secondly, we have evaluated the precision ratio, the recall ratio and the F-measure with another scene in which the paused object had started to move again. The proposed method gave us a better result than the other methods (see **Table 3**). The GMM based method [3], Method 1 and Method 2 detected many FP pixels (afterimage of the paused foreground object) in the region where the foreground object had been paused (see red circle regions in **Fig. 6**). This

⋆1 We got the ground truth dataset from http://limu.ait.kyushu-u.ac.jp/dataset/

⋆2 The approach is equivalent to the literature [13].

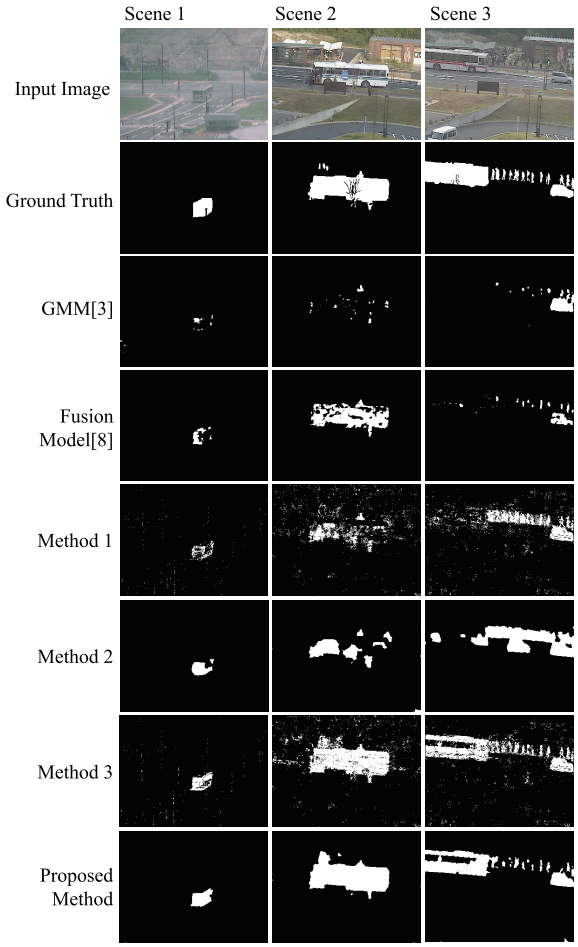

Fig. 5 Result of the object detection after the moving object stopped. Scene 1: 100^{th} frame after stopped, Scene 2: 60^{th} frame after stopped, Scene 3: 150^{th} frame after stopped.

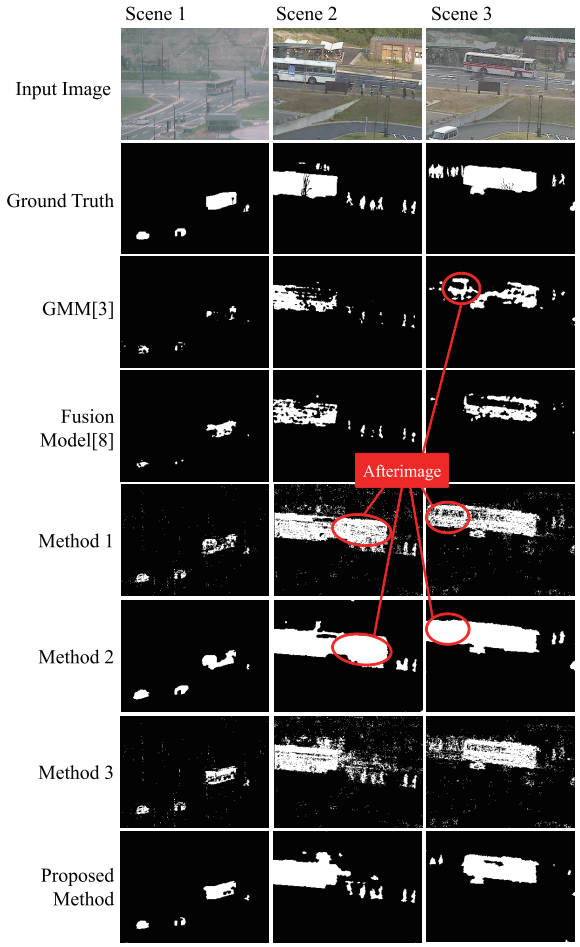

Fig. 6 Result of the object detection after the object restarted to move.

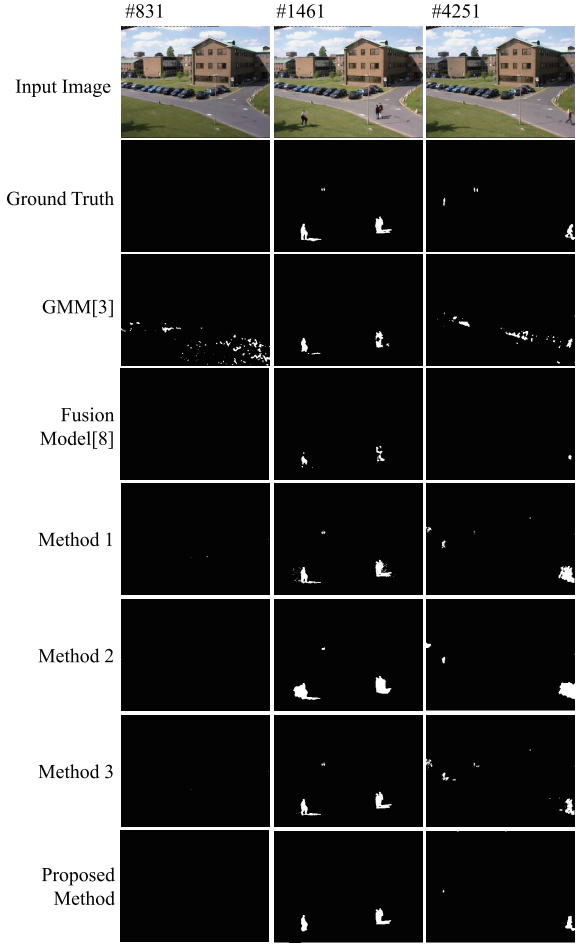

Fig. 7 Result of the object detection with the PETS2001 detaset. # 831: no object with illumination change, # 1461: objects without illumination change, # 4251: objects with illumination change.

is because an illumination change occurred during the period. Meanwhile, the fusion model [8], Method 3 and the proposed method did not detect the occluded region wrongly. However, the fusion model could not detect the inside of the moving object because of over-training of the foreground object. This is why the recall ratio of the fusion model was lower than that of the proposed method. The proposed method output lower precision ratio in the Scene 2. There was a tree

Table 2 Accuracy evaluation of the object detection after the moving object stopped.

		Scene 1	Scene 2	Scene 3
GMM [3]	Precision	0.87	0.95	0.86
	Recall	0.13	0.05	0.16
	F-measure	0.23	0.10	0.27
Fusion Model [8]	Precision	0.98	0.95	0.94
	Recall	0.37	0.69	0.13
	F-measure	0.53	0.80	0.24
Method 1	Precision	0.64	0.63	0.54
	Recall	0.33	0.32	0.34
	F-measure	0.44	0.42	0.42
Method 2	Precision	0.88	0.82	0.46
	Recall	0.59	0.36	0.42
	F-measure	0.70	0.50	0.44
Method 3	Precision	0.77	0.73	0.74
	Recall	0.64	0.92	0.74
	F-measure	0.70	0.82	0.74
Proposed Method	Precision	0.90	0.85	0.87
	Recall	0.76	0.99	0.74
	F-measure	0.82	0.92	0.81

Table 3 Accuracy evaluation of the object detection after the object restarted to move.

		Scene 1	Scene 2	Scene 3
GMM [3]	Precision	0.95	0.93	0.80
	Recall	0.22	0.52	0.46
	F-measure	0.35	0.66	0.58
Fusion Model [8]	Precision	0.98	0.93	0.94
	Recall	0.48	0.61	0.46
	F-measure	0.65	0.73	0.61
Method 1	Precision	0.73	0.49	0.65
	Recall	0.51	0.83	0.78
	F-measure	0.60	0.61	0.71
Method 2	Precision	0.80	0.51	0.67
	Recall	0.73	0.93	0.99
	F-measure	0.76	0.66	0.80
Method 3	Precision	0.83	0.62	0.74
	Recall	0.64	0.88	0.81
	F-measure	0.72	0.72	0.77
Proposed Method	Precision	0.92	0.78	0.91
	Recall	0.78	0.98	0.82
	F-measure	0.85	0.87	0.86

Table 4 Accuracy evaluation of the object detection with the PETS2001 dataset.

		# 831	# 1461	# 4251
GMM [3]	FN	0	392	274
	FP	1,111	29	665
	F-measure	–	0.76	0.21
Fusion Model [8]	FN	0	697	351
	FP	0	3	1
	F-measure	–	0.51	0.22
Method 1	FN	0	94	83
	FP	6	138	587
	F-measure	–	0.89	0.49
Method 2	FN	0	46	58
	FP	1	609	1,108
	F-measure	–	0.76	0.37
Method 3	FN	0	93	66
	FP	2	39	362
	F-measure	–	0.94	0.61
Proposed Method	FN	0	225	121
	FP	0	336	383
	F-measure	–	0.75	0.53

in front of the bus. The tree should be regarded as belonging to the background. However, our proposed method mistakenly detected it since the thin trunk and the branches are filtered out by the smoothing effect of the energy function defined in the Eq. (16). This is the main factor why our proposed method provided a bad precision ratio.

6.4 Evaluation of the Robustness against Illumination Changes

We have used an outdoor image sequence in which the illumination condition had sometimes changed rapidly, which was also used in the Section 6.2. We have selected three images (see **Fig. 7**) from 5,000 frames for evaluation. The parameters of the background models including the competitive methods were set to be the same as in previous experiments.

The recall ratio, the precision ratio and the F-measure are shown in **Table 4**. In the case where FP or FN is set to zero, we showed the F-measure as "–" in Table 4 since it cannot be calculated. The illumination condition of the scene # 831 was changed. The GMM based method [3] detected many FP pixels since it was hard for GMM to adapt to rapid illumination changes.

Meanwhile, our proposed method did not detect any FP pixels as well as the fusion model [8], which was reported to be very robust against various illumination changes. Method 1, Method 2 and Method 3 also adapted to the illumination change. The scene # 1461 included foreground objects under the stable illumination condition. The fusion model [8] detected the foreground object with the smaller size than the ground truth. This is because the fusion process was achieved by calculating a logical AND operation between two kinds of background models. Therefore, the FN became larger and the FP became smaller compared with the GMM based method. On the other hand, the proposed method tended to detect foreground objects with a larger size than the actual size. We guess that a kind of smoothing effect by the energy function would provide such a detection result. We think that such a characteristic is not a critical problem compared with the detection in a smaller size since most of surveillance systems require the position where the object is. If a method tends to detect an object in a smaller size, there is a possibility that the object will not be detected. Finally, the scene # 4251 included foreground objects with illumination changes. This scene is one of the most difficult scenes for object detection. Method 3 and the proposed method gave better results than the other competitive methods (see Fig. 7). The

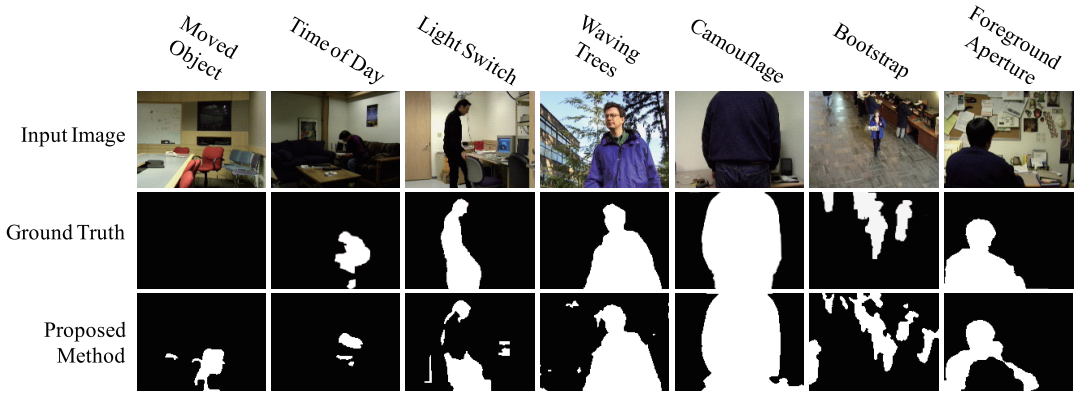

Fig. 8 Result of object detection with the Wallflower dataset.

Table 5 Accuracy evaluation with the Wallflower dataset.

	Error Type	Moved Object	Time of Day	Light Switch	Waving Trees	Camou-flage	Boot-strap	Foreground Aperture	**Total Errors**
Wallflower [9]	FN	0	961	947	877	229	2,025	320	
	FP	0	25	375	1,999	2,706	365	649	11,478
Fusion Model [8]	FN	0	1,349	1,681	198	177	1,235	2,085	
	FP	0	0	1,396	771	342	199	658	10,091
Proposed Method	FN	0	972	1,185	19	65	996	2,268	
	FP	1,130	6	596	441	705	2,117	843	11,337

FP and FN pixels of Method 3 were less than those of the proposed method since small foreground objects were filtered out by a graph cut (smoothing term in the energy function) in the proposed method.

6.5 Comparison with WallFlower Dataset

We have compared our proposed method with the fusion model [8] and WallFlower over the public dataset ⋆1. We counted up the FN and FP pixels in the same way as WallFlower [9]. The comparison results are shown in **Fig. 8** and **Table 5**. We have cited the result of WallFlower from the literature [9] ⋆2.

We got the same level of accuracy in the scene "Time of Day," and a better result in the scene "Waving Trees" and "Camouflage." The scene "Light Switch" included a sudden illumination change around the light switch ON/OFF. We introduced an additional process, in which a sudden illumination change was detected by counting the foreground pixels, and executed the re-initializing process according to it, in the same way as WallFlower. We got the result shown in Table 5 which was not so different from WallFlower's result.

In the scene "Foreground Aperture," there were many FN pixels inside the foreground object. It is very difficult to solve this problem by only using a background subtraction based approach. The approach of WallFlower extended the foreground pixels based on color similarity. Such an approach will be helpful for our proposed method to improve the performance.

Finally, in the scene "Moved Object" and "Bootstrap," a larger number of FP

⋆1 We got the dataset at http://research.microsoft.com/en-us/um/people/jckrumm/WallFlower/TestImages.htm

⋆2 The literature [9] gives several comparison results with other approaches.

pixels were detected wrongly compared with traditional methods. A chair moved and then it stopped in the scene "Moved Object." Our proposed method kept the chair being detected as foreground after it stopped. On the other hand, the chair should not be detected in the ground truth. This is why our proposed method detected FP pixels. The scene "Bootstrap" was not suitable for our proposed method in terms of background initialization since several foreground objects were already included in the images for initialization. If we had intentionally selected some images for initialization, a better result would have been acquired. Alternatively, a robust estimation method is effective to generate some images which are useful for the model initialization.

6.6 Discussion

Through several experiments, we have investigated the effectiveness of the proposed method for object detection. When a moving object stops in the surveillance scene, the proposed method can detect it by updating the background model hidden by the paused object. However, it has a limitation that the initialization of the proposed method has to be achieved by images which do not contain any foreground objects. Otherwise, background regions hidden by such foreground objects will be mistakenly detected if the objects start to move. Also they will not be detected as long as they are stopped.

On the other hand, although the proposed method can detect paused objects, it cannot distinguish overlapping foreground objects, e.g., a moving car in front of a paused bus, and so on. We have to construct hierarchical layers of the background to solve such a problem. This is part of the future work in our research.

7. Conclusion

We have proposed a novel background modeling method. The proposed method could maintain even a background model hidden by paused objects. This process was very effective for not only an implicit background update but also keeping a foreground object to being detected. Our background model can be utilized for scene surveillance including an intersection, a bus stop and so on. Also, we are sure that it contributes to abandoned object detection. Through several experiments, we have confirmed the effectiveness of our approach from the viewpoints of the robustness against illumination changes, the handling of foreground objects and the update of background model parameters. In our future work, we will study about an efficient strategy of initializing the background model, the complement of undetected pixels such as inside the objects. And also, we will apply our approach to an actual visual surveillance system.

References

1) Stauffer, C. and Grimson, W.: Adaptive background mixture models for real-time tracking, *Computer Vision and Pattern Recognition*, Vol.2, pp.246–252 (1999).
2) Cheng, J., Yang, J. and Zhou, Y.: A Novel Adaptive Gaussian Mixture Model for Background Subtraction, *2nd Iberian Conference on Pattern Recognition and Image Analysis*, pp.587–593 (2005).
3) 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).
4) Elgammal, A., Duraiswami, R., Harwood, D. and Davis, L.: Background and Foreground Modeling Using Non-parametric Kernel Density Estimation for Visual Surveillance, *Proc. IEEE*, Vol.90, pp.1151–1163 (2002).
5) 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).
6) Shimada, A. and Taniguchi, R.: Hybrid Background Model using Spatial-Temporal LBP, *IEEE International Conference on Advanced Video and Signal Based Surveillance* (2009).
7) 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, pp.63–73 (2004).
8) Tanaka, T., Shimada, A., Taniguchi, R., Yamashita, T. and Arita, D.: Towards robust object detection: Integrated background modeling based on spatio-temporal features, *Asian Conference on Computer Vision* (2009).
9) 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).
10) Basharat, A., Gritai, A. and Shah, M.: Learning object motion patterns for anomaly detection and improved object detection, *Computer Vision and Pattern Recognition*, pp.1–8 (2008).
11) Porikli, F., Ivanov, Y. and Haga., T.: Robust abandoned object detection using dual foreground, *EURASIP Journal on Advances in Signal Processing* (2008).
12) Li Tian, Y., Feris, R. and Hampapur, A.: Real-time detection of abandoned and removed objects in complex environments, *International Workshop on Visual Surveil-*

lance - VS2008 (2008).
13) Shimada, A. and Taniguchi, R.: Hybrid Background Modeling for Long-term and Short-term Illumination Changes, *IEEJ Trans. EIS*, Vol.130, No.9, pp.1524–1529 (2008).
14) Hosaka, T., Kobayashi, T. and Otsu, N.: Moving Object Detection by Using Markov Random Field Model, *Technical report of IEICE, PRMU*, Vol.107, No.539, pp.437–442 (in Japanese) (2008).
15) Holt, C.C.: Forecasting seasonals and trends by exponentially weighted moving averages, *International Journal of Forecasting*, Vol.20, pp.5–10 (2004).
16) Heikkilä, M., Pietikäinen, M. and Heikkilä, J.: A texture based method for detecting moving objects, *British Machine Vision Conf.*, Vol.1, pp.187–196 (2004).
17) Boykov, Y. and Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.26, pp.1124–1137 (2004).

(Received November 9, 2010)
(Accepted June 13, 2011)
(Released December 28, 2011)

(Communicated by *Tomokazu Takahashi*)

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.

Satoshi Yoshinaga received his B.E. and M.E. degrees from Kyushu University in 2009 and 2011. Since 2007, he has been a Ph.D. student in Graduate School of Information Science and Electrical Engineering at Kyushu University. He is also a Research Fellow of the Japan Society for the Promotion of Science. He has been engaged in visual surveillance.

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.