**The 6th International Symposium on Visual Computing (ISVC 2009), in Las Vegas, USA, Vol.5875, pp.500-510 (2009)**

# Codebook-based Background Subtraction to Generate Photorealistic Avatars in a Walkthrough Simulator

Anjin Park[1], Keechul Jung[2] and Takeshi Kurata[1]

[1]Center for Service Research, National Institute of Advanced Industrial Science and Technology, Japan
[2]Department of Digital Media, Soongsil University, Korea
[1]{anjin.park,t.kurata}@aist.go.jp, [2]kcjung@ssu.ac.kr

**Abstract.** Foregrounds extracted from the background, which are intended to be used as photorealistic avatars for simulators in a variety of virtual worlds, should satisfy the following four requirements: 1) real-time implementation, 2) memory minimization, 3) reduced noise, and 4) clean boundaries. Accordingly, the present paper proposes a codebook-based Markov Random Field (MRF) model for background subtraction that satisfies these requirements. In the proposed method, a codebook-based approach is used for real-time implementation and memory minimization, and an edge-preserving MRF model is used to eliminate noise and clarify boundaries. The MRF model requires probabilistic measurements to estimate the likelihood term, but the codebook-based approach does not use any probabilities to subtract the backgrounds. Therefore, the proposed method estimates the probabilities of each codeword in the codebook using an online mixture of Gaussians (MoG), and then MAP-MRF (MAP: Maximum A-Posteriori) approaches using a graph-cuts method are used to subtract the background. In experiments, the proposed method showed better performance than MoG-based and codebook-based methods on the Microsoft DataSet and was found to be suitable for generating photorealistic avatars.

## 1 Introduction

Before constructing buildings, details such as the suitability of the floor layout navigation signs and whether users will feel comfortable in the buildings should be considered. Virtual reality techniques are used to investigate virtual structures in detail. However, since information is generally displayed through monitors and a keyboard or mouse is used to navigate the structure, it is difficult to evaluate the relationship between the details of the structure and the sense of absolute direction of the user. Therefore, we are developing simulation environment, referred as to *Walkthrough Simulator* (WTS), in order to enable subjects to navigate virtual constructions from the perspective of the customer. The hexahedral-shaped device shown in the center of Fig. 1 is the WTS. The virtual building is displayed inside the device using multi-projectors, as shown on the right-hand side of Fig. 1.

In some buildings, such as public institutions, guides provide instructions to customers or visitors to help them reach their destination. In the virtual building, guides

are displayed as avatars. In the present study, we use a *Photorealistic Avatar*, in which the appearance of an actual person is used as CG texture, as a guide. The present paper focuses on displaying the photorealistic avatar in virtual buildings. The image of the person that is used to create the photorealistic avatar is extracted by the camera in front of the simulator, as shown in Fig. 1(A), and the photorealistic avatar is displayed in a fixed location in the virtual world inside the simulator, as shown in Fig. 1(B).

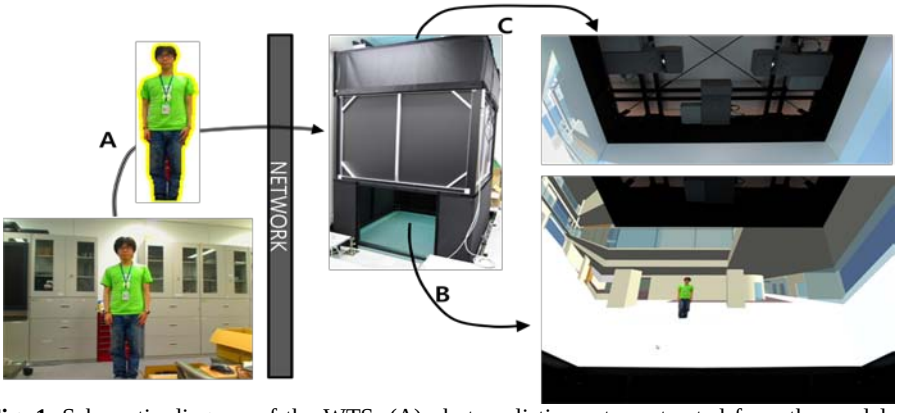



**Fig. 1.** Schematic diagram of the WTS: (A) photorealistic avatar extracted from the modeled background, (B) photorealistic avatar integrated into the virtual building, and (C) multi-projectors.

In the present paper, it is assumed that the moving foreground in front of the fixed camera outside the simulator is an individual whose image will be used to generate the photorealistic avatar, and the actual person who will take a role of the guide can stand any places, for example, a room with complex backgrounds. The present paper uses background subtraction to extract the appearance of the guide from images captured by a camera. There are four requirements for the background subtraction: 1) extraction must be performed in real time, 2) memory consumption must be limited, 3) the image must be extracted with little noise, and 4) the boundaries of the avatar must be clear.

The proposed method integrates a codebook-based approach, which helps to perform extraction in real time and reduces the required memory, and an edge-preserving MRF model, which can eliminate noise and generate clear boundaries. Although the codebook-based algorithm [10] can model an adaptive and compact background over a long period of time with limited memory, it cannot be used as the likelihood term in the edge-preserving MRF, because the similarity (rather than the probability) is used to compare input pixels with the modeled background. Therefore, online mixture of Gaussians (MoG) is used to estimate the probabilities for all codewords in the codebook. In addition, the proposed method models the prior term using the codebook-based method in order to substantially reduce extraction errors caused by high-contrast edges in cluttered backgrounds, thereby reducing errors on the boundaries of extracted foregrounds.

## 2 MRF Modeling for Background Subtraction

### 2.1 Related Research

The simplest background model assumes that pixel values can be modeled by a single Gaussian distribution [1]. However, this basic model cannot handle multiple backgrounds, such as trees moving in the wind. The MoG has been used to model non-static backgrounds [2]. However, it is difficult to detect sudden changes in the background when the learning rate is low, and slowly moving foreground pixels will be absorbed into the background model when the learning rate is high [7]. Sheikh and Shah [4] proposed a MAP-MRF framework, which results in clear boundaries without noise by enforcing spatial context in the process, but this technique [4] cannot be used when long periods of time are needed to sufficiently model the background, primarily due to memory constraints, because they used a kernel density estimation technique [7]. In order to address the memory constraint problem, Kim et al. [7] proposed a codebook background subtraction algorithm intended to model pixel values over long periods of time, without making parametric assumptions. However, since this algorithm did not evaluate probabilities, but only calculated the distance from the cluster means, it is hard to extend this algorithm to the MAP-MRF framework.

### 2.2 Energy Function

In the present paper, background subtraction is considered as an MRF framework. The MRF is specified in terms of a set of sites $\mathcal{S}$ and a set of labels $\mathcal{L}$. Consider a random field consisting of a set of discrete random variables $\mathbf{F} = \{F_1, \ldots, F_n\}$ defined on the set $\mathcal{S}$, such that each variable $F_s$ takes a value $f_s$ in $\mathcal{L}$, where $s$ is index of the set of sites. For a discrete label set $\mathcal{L}$, the probability that random variable $F_s$ takes the value $f_s$ is denoted as $P(F_s = f_s)$, and the joint probability is denoted as $P(\mathbf{F} = \mathbf{f}) = (F_1 = f_1, \ldots, F_n = f_n)$, abbreviated as $P(\mathbf{f})$, where $\mathbf{f} = \{f_1, \ldots, f_n\}$. Here, $\mathbf{f}$ is a configuration of $\mathbf{F}$.

If each configuration, $\mathbf{f}$ is assigned a probability $P(\mathbf{f})$, then the random field is said to be an MRF [11] with respect to a neighborhood $N = \{N_s | s \in \mathcal{S}\}$, where $N_s$ is the set of sites neighboring $s$, if and only if it satisfies the following two conditions: the positivity property $P(\mathbf{f}) > 0, \forall \mathbf{f} \in \mathbf{F}$ and the Markovian property $P(f_s | f_{S-\{s\}}) = P(f_s | f_{N_s})$, where $f_{N_s} = \{f_{s'} | s' \in N_s\}$ denotes the set of labels at the sites neighboring $s$.

Since $\mathbf{F}$ is generally not accessible, its configuration $\mathbf{f}$ can only be estimated through an observation *obs*. The conditional probability $P(\mathbf{f}|obs)$ is the link between the configuration and the observation. A classical method of estimating the configuration $\mathbf{f}$ is to use MAP estimation. This method aims at maximizing the posterior probability $P(\mathbf{f}|obs)$, which is related to the Bayes rule as follows: $P(\mathbf{f}|obs) = \frac{P(obs|\mathbf{f})P(\mathbf{f})}{P(obs)}$..

Since the problem lies in maximizing the previous equation with respect to $\mathbf{f}$, which$P(obs)$ does not act on, the MAP problem is equivalent to

$$P(\mathbf{f}|obs) = \text{argmax}_{\mathbf{f}\in\mathbf{F}}\left(\sum_{s\in\mathcal{S}} D_s(f_s) + \sum_{\{s,s'\}\in N} V_{s,s'}(f_s, f_{s'})\right), \quad (1)$$

in an energy function, where $P(\mathbf{f})$ is the Gibbs distribution, and pairwise cliques are considered. For more information on the MAP-MRF, please refer to the paper by Geman and Geman [5].

In the present paper, $D_s(f_s)$ in Eq. 1 is referred to as the likelihood term derived from the modeled background, which reflects how each pixel fits into the modeled data given for each label, and $V_{s,s'}(f_s, f_{s'})$ is referred to as the a prior term that encourages spatial coherence by penalizing discontinuities between neighboring pixels $s$ and $s'$. In addition, $V_{s,s'}(f_s, f_{s'})$ is replaced by $V_{s,s'} \cdot \delta(f_s, f_{s'})$, where $\delta(f_s, f_{s'})$ denotes the delta function defined by 1 if $f_s \neq f_{s'}$, and otherwise denotes the delta function defined by 0. Thus, this is a penalty term when two pixels are assigned different labels.

### 2.3 Graph Cuts

To minimize the energy function (Eq. 1), we use a graph-cuts method [8], because this method showed the best performance among the conventional energy minimization algorithms [9]. The procedure for energy minimization using the graph-cuts method includes building a graph, wherein each cut defines a single configuration, and the cost of a cut is equal to the energy of its corresponding configuration [9].

For the graph-cuts method, a graph $G = \langle v, \varepsilon \rangle$ is first constructed with vertices corresponding to the sites. Two vertices, namely, *source* (*Src*) and *sink* (*Sin*), also referred to as terminals, are needed in order to represent two labels, and each vertex has two additional edges, $\{s, Src\}$ and $\{s, Sin\}$. Therefore, the sets of vertices $v$ and edges $\varepsilon$ are $v = \mathcal{S} \cup \{Src, Sin\}$ and $\varepsilon = N \cup_{s \in \mathcal{S}} \{\{s, Src\}, \{s, Sin\}\}$,, where $N$ are referred to as *n-links* (neighboring links) and $\{s, Src\}$ and $\{s, Sin\}$ are referred to as *t-links* (terminal links). The weights of the graph are set for both *n-links* and *t-links*, where the *t-links* connecting each terminal and each vertex correspond to the likelihood term and the *n-links* connecting neighboring vertices correspond to the prior term.

Note that the background subtraction problem can be solved by finding the least energy consuming configuration of the MRF among the possible assignments of the random variables **F**. Minimizing the energy function defined in Eq. 1 is equivalent to finding the cut with the lowest cost, because the costs of two terms are assigned to the weights of the graph. Specific labels are then assigned to two disjointed sets connected by *Src* and *Sin* by finding the cut with the lowest cost in the graph. The minimum-cost cut of the graph can be computed through a faster version of max-flow algorithm, proposed in [9]. The obtained configuration corresponds to the optimal estimate of $P(\mathbf{f}|obs)$.

## 3 Proposed Energy Function

### 3.1 Likelihood Term

The likelihood term is derived from the modeled background data to measure the cost of assigning the label $f_p$ to the pixel $p$, and $D_p(f_p)$ is defined as follows:

$$\begin{cases} D_p(f_p = \text{foreground}) = 1, & D_p(f_p = \text{background}) = 0, & \text{if } P(p) < T_f, \\ D_p(f_p = \text{foreground}) = 0, & D_p(f_p = \text{background}) = 1, & \text{if } P(p) > T_b, \\ D_p(f_p = \text{foreground}) = T_b^p, & D_p(f_p = \text{background}) = P(p), & \text{otherwise,} \end{cases}$$

where $T_f$ and $T_b$ are thresholds for hard constraints [10] in constructing graphs, $T_b^p$ is a threshold to extract moving objects from the background, and $P(p)$ is the probability that a pixel $p$ is included in the background. In the present paper, the codebook-based algorithm and MoGs are used to estimate the probabilities for the background.

The codebook algorithm is used to construct a background model from long input sequences and adopts a quantization technique to minimize the required memory. For each pixel, the codebook algorithm builds a codebook consisting of one or more codewords. Samples at each pixel are quantized into a set of codewords based on color and brightness information. The background is then encoded on a pixel-by-pixel basis.

Let $\mathbf{X}$ be a training sequence for a single pixel consisting of $n_x$ RGB-vectors: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}\}$, and let $\mathbf{C}$ be the codebook for a pixel consisting of $n_c$ codewords. Each pixel has a different codebook size based on its sample variation. Each codebook $\mathrm{c}_i, \mathrm{i} = 1, \ldots, n_c$ consists of an RGB vector $\mathbf{v}_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ and a 7-tuple $\mathbf{aux}_i = \langle \check{I}_i, \hat{I}_i, \check{\mathbf{T}}_i, \hat{\mathbf{T}}_i, \tau_i, q_i, f_i \rangle$, where $\check{I}_i$ and $\hat{I}_i$ denote the minimum brightness and maximum brightness, respectively, of the $i$th codeword, $\check{T}_i$ and $\hat{T}_i$ denote the thresholds for the RGB vector $\mathbf{v}_i$, $\tau_i$ denotes the maximum negative run-length (MNRL), which is defined as the longest interval during the training period in which the codeword did not recur, $q_i$ denotes the last access time at which the codeword occurred, and $f_i$ is the frequency with which the codeword occurs.

1 $n_c \leftarrow 0, \mathbf{C} \leftarrow \emptyset$ (empty set)
2 **For** $t$ = 1 to $n_x$ do
2.1 $\mathbf{x}_t = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$
2.2 Find the codeword $\mathbf{c}_m$ in $\mathbf{C} = \{\mathbf{c}_i | 1 \le i \le n_c\}$ that matches $\mathbf{x}_t$ based on two conditions $\check{T}_i \le \mathbf{x}_t \le \hat{T}_i$ and $\check{I}_i \le I \le \hat{I}_i$
2.3 If $\mathbf{C} = \emptyset$, or, if there is no match, then $n_c = n_c + 1$. Create a new codeword $\mathbf{c}_{n_c}$ by setting $\mathbf{v}_{n_c} \leftarrow (R, G, B)$ and $\mathbf{aux}_{n_c} = \langle I - t_l, I + t_l, \mathbf{v}_{n_c} - \mathbf{t_v}, \mathbf{v}_{n_c} + \mathbf{t_v}, n_c - 1, n_c, 1 \rangle$
2.4 Otherwise, update the matched codeword $\mathbf{c}_m$, consisting of $\mathbf{v}_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $\mathbf{aux}_m = \langle \check{I}_m, \hat{I}_m, \check{\mathbf{T}}_m, \hat{\mathbf{T}}_m, \tau_m, q_m \rangle$, by setting
$\mathbf{v}_m \leftarrow \left( \frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1} \right)$ and
$\mathbf{aux}_m = \langle \min\{I, \check{I}_m\}, \max\{I, \hat{I}_m\}, \min\{\mathbf{v}_m, \check{\mathbf{T}}_m\}, \max\{\mathbf{v}_m, \hat{\mathbf{T}}_m\}, \max\{\tau_m, t - q_m\}, t, f_m + 1 \rangle$
**end for**

**Fig. 2.** Algorithm for codebook construction

After construction, the codebook may be sizeable because it contains all of the codewords that may include moving foreground objects and noise. Therefore, the codebook is refined by eliminating the codewords that contain moving foreground objects. The MNRL in the codebook is used to eliminate the codewords that include moving objects, based on the assumption that pixels of moving foreground objects appear less frequently than moving backgrounds. Thus, codewords having a large $\tau$ are eliminated by the following equation: $\mathbb{C} = \{\mathbf{c}_m | \mathbf{c}_m \in \mathbf{C} \wedge \tau_m \le T_{\mathbb{C}}\}$, , where $\mathbb{C}$ denotes the background model, which is a refined codebook, and $T_{\mathbb{C}}$ denotes the threshold value. In the experiments, $T_{\mathbb{C}}$ was set to be equal to half the number of training frames.

In the case of using codebook-based algorithms, it is difficult to use an MRF because the MRF does not evaluate probabilities, but rather calculates the distance from the RGB vectors and the brightness of the codewords.

To evaluate the probabilities from the codebooks, a mixture of *K* Gaussian distributions proposed by Stauffer and Grimson [2] is chosen to model the recent history of each pixel, which is included in the same codewords. The probability of observing the current pixel value $\mathbf{x}_t$ is $P(\mathbf{x}_t) = \sum_{i=1}^{K} w_{i,t} * \eta(\mathbf{x}_t, \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t})$,, where *K* is the number of distributions, $\boldsymbol{\mu}_{i,t}$ is an estimate of the weight of the *i*th Gaussian in the mixture at time *t*, $\boldsymbol{\mu}_{i,t}$ and $\boldsymbol{\Sigma}_{i,t}$ are the mean value and covariance matrix, respectively, of the *i*th Gaussian in the mixture at time *t*, and $\eta$ is a Gaussian probability density function. In the experiments, *K* is determined by the number of frames used for background modeling, and the covariance matrix is assumed to be of the following form: $\boldsymbol{\Sigma}_{k,t} = \sigma_k^2 \mathbf{I}$

### 3.2 Prior Term

Since a common constraint is that the labels should vary smoothly almost everywhere while preserving sharp discontinuities that may exist, e.g., at boundaries [8], the costs of the smoothness are assigned for discontinuity-preservation between two neighboring pixels, and we use a generalized Potts model [8]. As such, $V_{p,p'}$ is defined as follows:

$$V_{p,p'} = dis(p,p')^{-1} e^{(-\beta \cdot \|p-p'\|^2)}, \quad (2)$$

where the contrast term $\|p - p'\|^2$ denotes the dissimilarity between two pixels $p$ and $p'$, and $dis(\cdot)$ is the Euclidean distance between neighboring pixels in the image domain. When $\beta = 0$, the smoothness term is simply the Ising model, which promotes smoothness everywhere. However, it has been shown that it is more effective to set $\beta > 0$, because this relaxes the tendency to smooth regions of high contrast. The constant $\beta$ is chosen to be $\beta = (\langle \|p - p'\|^2 \rangle)^{-1}$, where $\langle \cdot \rangle$ denotes the expectation over an image. This choice of $\beta$ ensures that the exponential term in Eq. 2 switches appropriately between high and low constants.

However, when the scene contains a cluttered background, notable segmentation errors often occur around the boundary, which generates flickering artifacts in the final results displayed in the virtual world [6]. These errors occur because the MRF model contains two terms for color and two terms for contrast. A straightforward idea is to subtract the contrast of the background image from the current image [6]. However, since only one background image is used for this approach, the nonstationary background motion that is ubiquitous in the real world cannot be modeled.

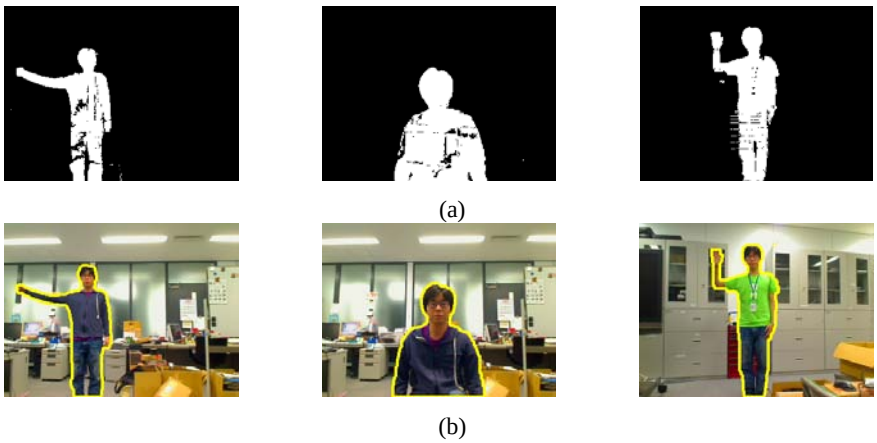
To overcome this problem, the contrast of the background is modeled using the codebook-based algorithm described in Section 3.1. The difference is that the codebook for the smoothness terms uses $V_{p,p'}$ instead of $I$ as input and does not use $\mathbf{x}_t$. This means modeling contrasts between adjacent pixels. After modeling the contrasts, if the contrasts of the input frame are within the ranges $\check{V}_m$ and $\hat{V}_m$ of any codeword *m*, then the contrast is considered to be background contrast, and $V_{s,s'}$ is set 0. Otherwise, $V_{s,s'}$ is set as the value of an input frame. This approach helps not only to eliminate the flickering artifacts but also facilitates the use of the generalized Potts model.

## 4 Experimental Results

Background subtraction was used to generate a photorealistic avatar in the virtual world for the WTS. Section 4.1 presents the resultant images displayed in the WTS, and Section 4.2 presents a quantitative evaluation to verify the effectiveness of the proposed method. All of the experiments were carried out on a 2.40-GHz Pentium 4 CPU.

### 4.1 Simulated Environment

The proposed method was based on the codebook-based method [7]. Images resulting from use of the codebook-based method (Fig. 3(a)) and the proposed method are shown in Fig. 3(b) and (b), respectively. As shown in Fig. 3, the results of the codebook-based method include some noise and holes in extracted regions. However, by applying an edge-preserving MRF framework, the proposed method includes no noise or holes and has clean boundaries. The photorealistic avatar, based on the resultant images presented in Fig. 3(b), was integrated into the WTS as shown in Fig. 4. In addition, MoG-based [2] and codebook-based [7] methods were compared to the proposed method, as shown in Fig. 5.

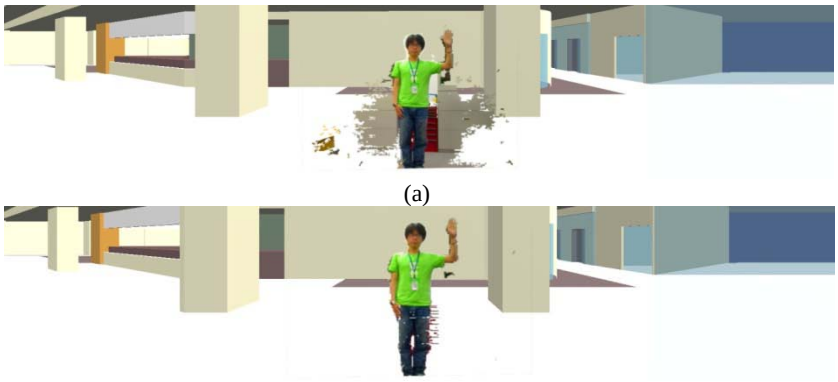


(a)

(b)

**Fig. 3** Resultant images of (a) the codebook-based method [7] and (b) proposed method.

**Fig. 4.** Photorealistic avatar integrated into a virtual building.



(a) (b)

**Fig. 5.** Guide representation in a virtual building using (a) MoG- [2] and (c) codebook-based [7] methods.

### 4.2 Qualitative Analysis

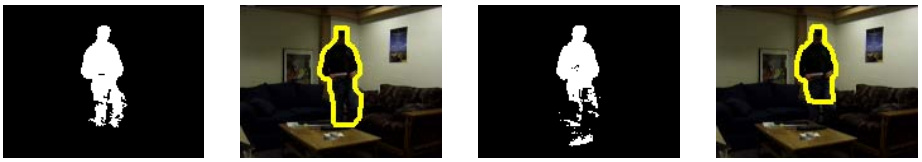
We tested four data sets described in [3]: Waving Trees, Camouflage, Time of Day, and Moved Object. We chose these four sets because the background images to be modeled might include nonstationary background motion, as in the Waving Trees and Camouflage sets, and because the sequential background images might change gradually as a result of changing light conditions throughout the day, as in the case of the Time of Day set. In the present study, in contrast to [3], moving objects are considered to be in the foreground, because the photorealistic avatar can use objects to express information to the subject. In the experiments, codebook-based [7] and MoG-based [2] results were compared with the results of the proposed method, and the results of these tests are shown in Figs. 6–9.

(a) (b) (c) (d)

**Fig. 6.** Sample results for Waving Trees obtained using (a) the MoG-based method, (b) the graph-cuts method using MoG, (c) the codebook-based method, and (d) the proposed method.

(a) (b) (c) (d)

**Fig. 7.** Sample results for Camouflage obtained using (a) the MoG-based method, (b) the graph-cuts method using MoG, (c) the codebook-based method, and (d) the proposed method.



(a) (b) (c) (d)

**Fig. 8.** Sample results for Time of Day obtained using (a) the MoG-based method, (b) the graph-cuts method using MoG, (c) the codebook-based method, and (d) the proposed method.

(a) (b) (c) (d)

**Fig. 9.** Sample results for Moved Object obtained using (a) the MoG-based method, (b) the graph-cuts method using MoG, (c) the codebook-based method, and (d) the proposed method.

The accuracy rates were evaluated by two criteria: the number of false positives and the number of false negatives. The number of false positives is the number of foreground pixels that were misidentified, and the number of false negatives is the number of background pixels that were identified as foreground pixels. As shown in Table 1, the proposed method had the best performance, except in the case of the Time of Day data set, as shown in Fig. 8. Since brightness values were used to deal with shadows, the proposed approach worked poorly in dark areas of images. Therefore, the leg regions of the human were not extracted by the proposed method. On the other hand, since shadow regions are included in the results shown in Fig. 7, the proposed method had better performance than the MoG-based methods. The processing times for each step of the proposed method are presented in Table 2. Approximately nine frames per second could be extracted using the proposed method.

**Table 1.** False positives and false negatives (%)

| | | MoG [3] | Codebook [7] | MoG + graph cuts | Proposed method |
|---|---|---|---|---|---|
| Fig. 9 | F. Positive | 6.89 | 3.17 | 3.08 | 0.02 |
| | F. Negative | 2.43 | 2.55 | 0.18 | 0.04 |
| Fig. 10 | F. Positive | 16.13 | 9.11 | 5.88 | 0.0 |
| | F. Negative | 38.09 | 1.15 | 19.27 | 0.93 |
| Fig. 11 | F. Positive | 5.21 | 9.80 | 2.18 | 11.2 |
| | F. Negative | 1.38 | 0.54 | 0.91 | 0.09 |
| Fig. 12 | F. Positive | 11.10 | 12.34 | 5.11 | 1.57 |
| | F. Negative | 9.75 | 8.27 | 9.33 | 8.12 |

**Table 2.** Processing times for each step of the proposed method (msec)

| Resolution | Codebook construction | MoG | Graph construction | Graph cuts |
|---|---|---|---|---|
| 160×120 | 8 | 5 | 7 | 25 |
| 320×240 | 20 | 10 | 20 | 60 |

## 5 Conclusions

In the present paper, we proposed a codebook-based MRF model for background subtraction to generate a photorealistic avatar displayed in the virtual world. Although an edge-preserving MRF can eliminate the noise and generate suitable object boundaries, the MRF depends on how the likelihood terms in the energy function are estimated. The proposed method uses a codebook-based method to estimate the likelihood term, which not only reduces the required memory and enables real-time implementation. Moreover, the proposed method used online MoG to estimate the probability for each codeword, which resulted in minimization of the required memory, reduced noise, and clean boundaries. In addition, the proposed method enabled the photorealistic avatar to be displayed clearly in the virtual world, as compared with previously proposed methods, such as codebook and MoG.

However, the proposed method was not able to extract the foreground in dark regions because brightness values were used to handle shadows. Therefore, in future studies, we intend to investigate how to extract the foreground in the dark regions more effectively. Moreover, we intend to extend the proposed method to extract the foreground inside a WTS that contains non-static backgrounds due to virtual world displayed inside the WTS.

**Acknowledgement**: This research was partially supported by the Ministry of Economy, Trade and Industry (METI), and partially supported by the Japan Society for the Promotion of Science (JSPS).

## References

1. Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A.: Pfinder: Real-Time Tracking of the Human Body, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, (1997) 780-785.
2. Stauffer, C. and Grimson, W.: Learning Patterns of Activity using Real-Time Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, (2000) 747-757
3. Toyama, K., Krumm, J., Brumitt, B., and Meyers, B.: Wallflower: Principles and Practice of Background Maintenance, *Proceedings of International Conference on Computer Vision*, (1999) 255-261.
4. Sheikh, Y. and Shah, M.: Bayesian Modeling of Dynamic Scenes for Object Detection, IEEE Transactions on Object Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, (2005) 1778-1792.
5. Geman, S. and Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Image, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, (1984) 721-741.
6. Sun, J., Zhang, W., Tang, X., and Shum, H-Y.: Background Cut, *Proceedings of European Conference on Computer Vision*, Part II, (2006) 628-641.
7. Kim, K., Chalidabhongse, T.H., Harwood, D., and Davis, L.: Real-Time Foreground-Background Segmentation using Codebook Model, *Real-Time Imaging*, vol. 11, no. 3, (2005) 172-185.
8. Boykov, Y., Veksler, O., and Zabih, R.: Fast Approximation Energy Minimization via Graph Cuts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, (2001) 1222-1239.
9. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C.: A Comparative Study of Energy Minimization Methods for Markov Random Fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, (2008) 1068-1080.
10. Boykov, Y. and Kolmogorov, V.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9. (2004) 1124-1137.
11. Li, S.Z.: *Markov Random Field Modeling in Computer Vision*, Springer, (2001).