J Real-Time Image Proc (2009) 4:317–328
DOI 10.1007/s11554-009-0122-4

ORIGINAL RESEARCH PAPER

Fast and accurate extraction of moving object silhouette for personalized Virtual Reality Studio @ Home

Chinta Rambabu · Kiyoung Kim · Woontack Woo

Received: 26 May 2008 / Accepted: 6 May 2009 / Published online: 3 June 2009

Abstract Accurate segmentation of moving object silhouette in a real-time video is very important for object silhouette extraction in the vision-based interactive systems. However, the inherent problem of moving object segmentation based on the background subtraction criteria is to distinguish the changes occurring from background disturbing effects such as noise, shadows and illumination changes. The present paper proposes a hybrid method based on the background subtraction criteria that preserves the boundary of moving object and also robust against the noise and illumination changes. In the proposed method, the object regions are well identified by fusing the results from the background difference and motion-based change detection criterion. The shadows and highlights are well detected by utilizing the normalized luminance and background difference in Hue and Saturation component. The paper also introduces a novel connected component analysis procedure for detecting the object blob from the noise blobs, and a robust pixel-based background update scheme for updating the dynamic changes in the background. Moreover, the computational complexity of the proposed algorithm is analyzed. The proposed method has been implemented and evaluated regarding the segmentation quality and the frame rate. Further, the method has been shown to successfully extract the moving object silhouette and robust against the disturbing effects. Moreover, the proposed method has been tested in the VR@Home platform.

Keywords Object silhouette · Background subtraction · Interactive systems · Connected component analysis · Change detection · Shadow elimination

C. Rambabu (✉)
Imaging Informatics Group, Bioinformatics Institute,
30 Biopolis Street, Matrix, Singapore 138671, Singapore
e-mail: chinta@bii.a-star.edu.sg

K. Kim · W. Woo
U-VR Lab, GIST, Kwangju 500 712, South Korea

K. Kim
e-mail: kkim@gist.ac.kr

W. Woo
e-mail: wwoo@gist.ac.kr

1 Introduction

Fast and accurate segmentation of moving object silhouette from a real-time video is an important preliminary step in most computer vision and video analysis applications like perceptual human computer interfaces [1, 2], traffic monitoring, video surveillance systems [3], human gait recognition [4], and 3D object modeling [5]. The aim of moving object segmentation is to extract the shape information of moving objects from the video sequence. Accurate moving object extraction from real-time video will greatly improve the performance of object tracking, recognition, classification and activity analysis, but it becomes a challenging research problem due to Camera noise and disturbing effects like illumination changes, small motion in the background regions, shadows and highlights, and occlusions. Several algorithms have been proposed for its solution. The conventional moving object segmentation algorithms can be roughly classified into two categories based on their primary segmentation criterion such as spatial homogeneity (region-based criterion) and change detection. The region-based approaches [6] tend to track the object boundary more precisely but the computational complexity is very high due to the fact that both the spatial segmentation and motion estimation steps are computationally intensive operations. On the other hand, the change detection approaches [1, 3, 7–10]

usually do not use the spatial features but rely only on the frame difference. The change detection techniques can be further classified as motion-based change detection and background subtraction algorithms. In motion-based change detection [7], the consecutive frame difference is utilized to distinguish the foreground from the background. However, the value of frame difference depends on the speed of object, so that the quality of segmentation cannot be maintained if the speed of object changes significantly in the sequence. On the contrary, the background subtraction methods utilize the background model as a reference, in order to distinguish the moving objects from the background scene. The basic idea of background subtraction is to subtract the current frame from the predetermined background model, acquired before the objects come in. The regions where there is a significant difference between the current frame and the estimated background indicate the position of the objects of interest. However, background subtraction which is based on a static background scene and hypothesis analysis, is often not suitable for real time environment. Various change detection techniques have been developed based on the background subtraction [1, 3, 8, 9, 11]. Most of the techniques focus on background initialization, including the background update and hypothesis test for detecting the object of interest. Several techniques exist in the literature; they have been proposed under the restricted environment conditions and as motivated by specific application. However, they lack robustness in the presence of disturbing effects like shadows, ghosts, highlights, reflections, illumination changes, dark environment and in situations where the foreground objects are very similar to the background. In order to get the accurate moving object silhouette, a hybrid technique can be developed by combining the multiple complementary features. In this paper, we aim at formulating a hybrid method based on background subtraction and motion information, that preserves the boundary of moving object and is robust against the disturbing effects. In the proposed method, a linear recursive method is introduced that models the background from a certain number of static background frames, where the background pixels are modeled as Gaussian distribution, characterized by their mean and variance but it utilizes less number of static background frames to get the noise-less background model. In order to obtain the accurate object boundary and track small motion in the background regions, we combine the results from motion-based change detection and background subtraction. The foreground pixels are clearly distinguished from the background by introducing confident level thresholding, obtained from the background noise. Moreover, the background regions with small motion are well identified by comparing the frame difference with the mean of the difference image by using the statistical parameters of the difference image. In general, change detection cannot detect the inherent unwanted shadows and highlights in the foreground regions. The proposed method introduced a detection method that uses the normalized luminance (V), and background difference in Hue and Saturation components, to distinguish the moving shadows and highlights very effectively. The proposed method also derives a novel order-queue based method to find the object from the background noise. This procedure uses only three raster scans for detecting connected regions in a binary image. Nevertheless, the proposed method introduces a robust update method that updates the dynamic background changes in the incoming images. Finally, we evaluate the results with the conventional methods regarding the segmentation quality and the frame rate. The present paper also discusses about the VR@Home platform.

The remaining sections of the paper are organized as follows. Section 2 reviews some of the background subtraction techniques. The next section presents a detailed discussion of the proposed hybrid background method. Section 4 gives some of the results, including the evaluation and comparison analysis, of the proposed method. Finally, the paper is concluded in Sect. 5.

2 Background subtraction techniques

C. Wren et al. [1] have developed a system so called *Pfinder* that segments the human body from the static background and then tracks it in real-time. The *Pfinder* uses a simple method where the background pixel is modeled as a unimodal Gaussian distribution and is updated by linear update, and the foreground pixels are modeled with mean and covariance, are updated recursively. However, it requires an empty scene during background training. In the W^4 system [3], the background is modeled by maximum and minimum intensity values, and the largest inter-frame absolute difference is observed in the background scene. These parameters are estimated from the first few frames of video without any foreground objects and are updated by those parts of the scene not containing foreground objects. First, the difference images are calculated with the current frame and the both maximum and minimum frames. The difference images are used to classify the foreground based on the inter-frame absolute difference image. Then, a simple sequential morphological filter has been used to eliminate the noise regions. Fancois [12] has proposed a system that uses HSV color space instead of RGB color space. In this system, the background is modeled as Gaussian distribution which is generated by considering the mean and standard deviation of each pixel. In foreground detection, the current frame is subtracted from the mean model and then the resultant difference pixels are compared with the standard deviation model. Finally, the background model is updated recursively. Horprasert [13] has introduced a new color model that separates

the luminance component from the chrominance. The expected chromaticity is obtained by the arithmetic mean of each pixel's RGB values, calculated over a number of static background frames. Based on the luminance and chromaticity distortions, several thresholds are determined to classify the pixel to be foreground, background, shadow or highlighted background. In [4], the background is estimated by using the temporal median of N static background frames, the typical values of N ranging from 50 to 200. The difference image is determined with current frame and background estimate. A pixel is marked as foreground if the resultant difference is greater than the pre-determined threshold which is an estimate of noise standard deviation generated off-line. It also uses template matching to select the candidate matches. Moreover, D. Hong [8] has developed a vision-based interface system based on background subtraction that models the background with well-known *RGB* and normalized *rgb* color models. The mean and variance of N static background frames are calculated over each color component. Each color space has its own classification part in which the current frame is converted in each color space. Within the each color space, the pixels are classified into four different categories, namely background, foreground with shadows, background with shadow and foreground. These methods work well for the static background scene, but they may fail if the background pixels are multi-modally distributed. Several techniques have been developed to deal with multi-model background distribution. Toyamma et al. [14] has introduced an auto regressive background modeling method based on linear Wiener filtering to learn and predict background changes. The filter co-officiates are updated for each frame time from the sample co-variances of the observed background values. However, the pixel-level mixture of Gaussian (MOG) [15, 16] background model is used to model the multiple-modal distributed backgrounds. Moreover, Elgammal et al. [17] has developed a non-parametric background model for dynamic background modeling, where each background pixel is compared with a recent sample of intensity values for this pixel by using Gaussian kernel function. Background subtraction is computed from the intersection of a short and long term model results. This method does not cope with the false detections produced by camera noise. Several methods have been proposed based on the background subtraction criteria. The next sections give a detailed discussion about the proposed background subtraction method and its simulation results.

3 Hybrid background subtraction method

In this section, we present a hybrid background subtraction method that preserves the accurate boundary of moving object (MO) in the consideration of HSV color model. In general, the RGB color space is not well behaved with respect to color perception, as a distance computed between two colors in RGB space does not reflect their perceptional similarity. The HSV color space corresponds closely to the human perception of color and can deal better with noise in the scene region. The geometry of HSV color space is more suitable for developing the algorithms that rely on intensity measurements and color information. Hence, the proposed method exploits the well-known HSV color space. The HSV model, however, separates the intensity (V) from the chromatic components (H, S). In this work, each color channel is processed independently in order to improve the accuracy of segmentation results. When an object enters the scene, we assume that the color (R, G, B) for given a pixel is changed to $(R + \delta, G + \delta, B + \delta)$ due to change in illumination, where δ is the change in illumination. In the RGB to HSV transform, the luminance component V is proportional to the maximum of R, G and B, i.e. $V = \max(R,G,B)$, V changes with the degree of δ, and the H component is proportional to $\frac{X-Y}{\max-\min}$, where X and Y are two of R, G, and B. Therefore, H component does not change with the change in illumination. Whereas S component equal to $\frac{\max-\min}{\max}$, changes with the degree of $\frac{\max-\min}{\max(\max+\delta)} \cdot \delta$ which is far smaller than δ. Hence, the HSV components are processed separately so that the proposed method can adapt to different environmental illumination condition.

Figure 1 shows the overall structure of the proposed algorithm. The proposed method consists of five sequential stages. First, we determine the background model from a certain number of static background frames, before the objects move in. In the modified background modeling step, the background pixels are modeled as uni-model Gaussian $N(\mu,\sigma^2)$, characterized by their mean and variance, are updated recursively during the background training. Then, we perform the core object detection step, where the moving objects are distinguished from the static background scene. In order to get the accurate boundary and track small motion in the background regions, we fuse the results from motion-based change detection and background subtraction stages.

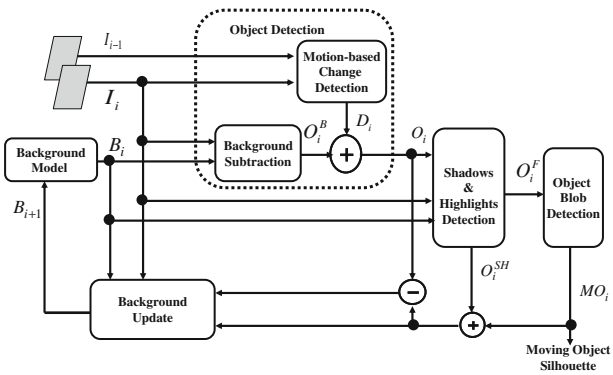

Fig. 1 Proposed hybrid background subtraction scheme

In the background subtraction step, first, the incoming frame is compared with the background model by using the predetermined threshold image which is obtained from the background noise (standard deviation of the background pixels). Then, the binary confidence map is generated for each channel. Higher the magnitude at a pixel, the more one is confident that the pixel belongs to the foreground. Similarly, we perform motion-based change detection where the difference between the current and previous frame, is compared with the mean of the difference image by using the normalized standard deviation of the frame deference. This step preserves well the boundary of the moving object and also detects the ghosts where the regions having small motion. Finally, we perform the logical OR between the results from both the stages. Next we perform the step of detecting the shadows and highlights where the shadow and highlight pixels are detected from the foreground mask by using the normalized luminance (V), and background subtraction of Hue and Saturation component. The proposed method utilizes the background parameters to distinguish the shadows and highlights very efficiently. In the subsequent step, we introduced a novel object blob detection procedure for detecting the object blob from noise blobs in the binary mask which is acquired from the previous stage. The proposed object detection procedure uses a 3×3 median filter and ordered queue-based connected component analysis for detecting the blobs but it utilizes only three raster scans for detecting the blobs in the binary image. Finally, we update the background model with the incoming images. The modified update procedure is different for the pixel positions which are detected as belonging to the foreground, as ghosts and part of the background. The entire procedure will be repeated for every incoming frame.

3.1 Background model

The goal of the background modeling is to train the background from a certain number of static background frames (with no video objects), which has less noise and random illumination changes. The conventional background modeling approaches can be broadly grouped as Uni-modeling, MOG modeling and Non-parametric modeling. In the Uni-modeling approach [1, 13], the intensity values of a background pixel can be modeled by a Gaussian modal $N(\mu,\sigma^2)$. The MOG [15] has been proposed to model the complex and non-static multiple backgrounds. A few Gaussian cannot accurately model the background having fast illumination changes. In Non-parametric modeling, the probability density at each pixel is estimated from many samples by using kernel density estimation technique. It is able to adapt very quickly to changes in the background and detect the targets with high sensitivity. However, it utilizes large memory for long time period modeling.

In the proposed approach, each background pixel is modeled as Gaussian model, characterized by its mean μ and its standard deviation σ. During the background training, the statistical parameters are updated by using recursive linear update technique. Let I_i denote the input background frame at time 'i'. The pixel distributions in the first frame are unknown, so we initialize the mean μ_0 with the values of first frame and set the variances to zero, then the distributions are updated from the next consecutive background frames. The mean μ_i for background frame at time 'i' can be defined as

$$\begin{aligned} &\mu_0 = I_0; \quad i = 0 \\ &\mu_i = (1-\alpha)\mu_{i-1} + \alpha I_i \quad i \geq 1 \end{aligned} \tag{1}$$

Similarly, the variance $\sigma^2{}_i$ for background frame at time 'i' can be defined as

$$\begin{aligned} &\sigma_0^2 = (I_1 - \mu_0)^2; \quad i = 1 \\ &\sigma_i^2 = (1-\alpha)\sigma_{i-1}^2 + \alpha(I_i - \mu_i)^2 \quad i \geq 2 \end{aligned} \tag{2}$$

where α is the learning rate of linear model. The mean of the pixels at (x, y) in the N continuous frames can be formulated as

$$B(x,y) = [\mu_i | i = N] \tag{3}$$

Similarly, the N frame standard deviation at (x, y) can be obtained by

$$\sigma_N(x,y) = \left[\sqrt{\sigma_i^2(x,y)} | i = N\right] \tag{4}$$

where the background model $B = \langle B^H, B^S, B^V \rangle$ and $\sigma_N = \langle \sigma_N^H, \sigma_N^S, \sigma_N^V \rangle$. This method utilizes less number of static background frames and less memory space to model the background while compared to the conventional N frame averaging method.

3.2 Moving object detection

In this section, we present an object detection procedure where the moving objects are distinguished from the static background scene. In order to get the accurate object boundary and track small motion in the background regions, we fuse the results from motion-based change detection and background subtraction stages. In the background subtraction step, first, current video frame is compared with the reference background model. Background subtraction is an efficient way to discriminate the moving objects from the still background. Then, we generate a normalized confidence map for each channel by using the pre-determined thresholds and decide the pixel characteristic based on the confidence map.

The more is the magnitude of confidence at a pixel, the more one is confident of the fact that the pixel belongs to

the foreground. It is intuitive that the change caused by a moving object can be large while the change caused by noise varies only around the mean value of the corresponding pixel in the background frames. However, a generic background point will have a small variance, while a point in moving object will have a higher variance value. Hence, the background variance can be used as a threshold to decide whether the pixel belongs to the background or occluding region. The background subtraction is performed for every pixel $I_i(x, y)$, as follows: at each pixel of a given current frame, the pixel level change detection is performed by computing the *Mahalanobis Distance* $\delta(x, y)$ from the pre-determined background model.

$$\delta(x,y) = |I_i(x,y) - B(x,y)| \tag{5}$$

where the *Mahalanobis Distance* image $\delta = \langle \delta^H, \delta^S, \delta^V \rangle$ and the current frame $I = \langle I^H, I^S, I^V \rangle$. This operation is applied for each HSV color channel, resulting in three difference images. Next, we perform a confident thresholding step for every channel by using predetermined threshold $\beta\sigma_N$, derived from the background model. A pixel can be considered as a foreground if the following condition is met:

$$O_i^B(x,y) = \begin{cases} 1 & \text{If} \begin{pmatrix} \left(\delta^H(x,y) > \beta\sigma_N^H(x,y)\right) & \vee \\ \left(\delta^S(x,y) > \beta\sigma_N^S(x,y)\right) & \vee \\ \left(\delta^V(x,y) > \beta\sigma_N^V(x,y)\right) & \end{pmatrix} \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

Here β is the confidence level, which controls the number of segmented regions. The β selection is purely based on the environmental condition, like outdoor or indoor. If the β value is less than one, then more are the false positives, on the contrary, higher the β value leads to under segmentation. In the present work, the choice of confidence level is done based on the receiver operating curve (ROC).

3.2.1 Motion-based change detection

We compute change detection between two consecutive frames to track motion in the background objects for updating the background model. In general a simple frame difference can produce very high noise. In order to reduce the noise level, we normalize standard deviation of frame difference at each pixel and then classified them as low or high motion regions based on the empirical threshold. The motion-based change detection mask can be obtained by thresholding on the normalized statistics of the frame difference between the current frame I_i and previous frame I_{i-1} that is, $D_i(x,y) = 1$ if,

$$|(I_i(x,y) - I_{i-1}(x,y)) - \mu_d| > \sigma_d T_d$$

and zero otherwise. Note that μ_d and σ_d are mean and standard deviation of the frame difference $I_i - I_{i-1}$, and T_d is empirical threshold that controls low motion regions in the background region. If the threshold value T_d less than one, then higher the low motion regions are included, so we choose the value of T_d empirically. Finally we perform the logical *OR* between the results from both the stages, in order to get the accurate boundary and track small motion in the background regions. The binary mask that represents the moving object regions can be generated as $O_i = D_i \vee O^B{}_i$. This step preserves well the boundary of the moving object and also detects the ghosts, small motion in the background regions.

3.3 Shadows and highlight detection

This section describes an improved shadow elimination method based on the conventional HSV-based shadow detection algorithm [18]. We present a shadow and highlight detection criteria that identify the moving shadows and highlights in the scene by utilizing the normalized luminance value between the current and background frame, and the background difference in Hue and Saturation components. By analyzing the color properties of the shadows and highlights, they have similar chromaticity with the background but lower brightness for shadows and higher brightness for highlights, than those of the pixels in the background model. A foreground pixel can be considered as shadowed or highlighted, if the following conditions hold:

$$\begin{cases} O^S(x,y) = 1 & \text{If} \begin{pmatrix} \left(\gamma \le \frac{I_i^V(x,y)}{B^V(x,y)} \le \phi\right) & \wedge \\ \left(\left(I_i^S - B^S(x,y)\right) < \beta\sigma_N^S(x,y)\right) & \wedge \\ \left(\left|I_i^H - B^H(x,y)\right| < \beta\sigma_N^H(x,y)\right) & \end{pmatrix} \text{(Shadow)} \\ O^H(x,y) = 1 & \text{Else if} \begin{pmatrix} \left(\frac{1}{\phi} \le \frac{I_i^V(x,y)}{B^V(x,y)} \le \frac{1}{\gamma}\right) & \wedge \\ \left(\left(I_i^S - B^S(x,y)\right) < \beta\sigma_N^S(x,y)\right) & \wedge \\ \left(\left|I_i^H - B^H(x,y)\right| < \beta\sigma_N^H(x,y)\right) & \end{pmatrix} \text{(Highlight)} \\ O(x,y) & \text{Otherwise(Object)} \end{cases} \tag{7}$$

where $I_i(x, y)$ and $B(x, y)$ are the pixel values at co-ordinate (x, y) in the current frame and the background, respectively. The parameter ϕ allows one to avoid identification as shadow of the points where the background was slightly changed by noise. Whereas γ takes into account how strong the light source is with respect to the reflectance and irradiance of the objects, the confidence level allows one to avoid classification of foreground points as shadows or highlight. Thus stronger and higher the sun (in the outdoor scene), the lower α should be chosen. The aforementioned parameters are set based on the environment condition. The proposed method utilizes the background statistical parameters to distinguish the shadows and highlights very efficiently.

3.4 Post-processing

In this section, we introduce an order-queue based connected component analysis procedure for detecting the moving object blob in a given binary image. The confidence thresholding and consecutive shadow elimination steps usually produce noise blobs on the object as well as over-segmentation in the background blob. A series of simple median filters with variable window size can be used to elevate the salt and pepper noise, and fills the holes in the high-confidence regions (desired object) and removes the low-confidence isolated regions. However, the size of the filter's window depends on the size of the noise regions in an image. The large sized noise regions can be eliminated by sequential median filtering with variable window size. However, the sequential median filtering procedure leads to alter the shape of object boundary and also increases the false detection rate. On the contrary, we introduce a novel noise blob procedure for detecting the object of interests from the background blob. This procedure requires only three raster scans for entire computation. Figure 2 illustrates the object blob detection procedure.

The proposed post-processing step consists of three sequential stages. First we detect the regions by performing the blob detection process, and then classify the background and object blob from noise blobs based on maximum area criteria. Finally, we extract the moving object blob by noise blob filtering.

3.4.1 *Queue-based blob detection*

In the Blob detection procedure, a single scan has been employed, and a FIFO queue Q has been utilized. At this stage, the plateau of pixels with the gray value 0 or 255 are detected and labeled by a raster scan of the input binary image. For each pixel p which is not yet visited, its 4 or 8-connected neighborhood $q \in N_G(p)$, where $N_G(p)$ is 8 or 4-connected neighbors of p, is inspected. Initialize the queue Q with p which has the gray value 0 or 255 and not yet visited. Pixel p is De-queued from the queue Q one at a time and its label is assigned to the non-labeled neighboring pixel q $(q \in N_G(p))$ with same gray value. Then, labeled pixel q becomes a candidate, and it is inserted in the queue. Each pixel in the same plateau gets current label, when the queue of candidates is emptied. Thus, each plateau is scanned in a breadth-first order, and the visited pixels are labeled with the current label. This procedure requires only two raster scans for detection and labeling of the blobs. The detailed algorithmic description of the blobs detection step is given in Algorithm 1.

Algorithm 1 Blob Detection(O, B_L, O_L)

Input: Binary image O $\{O : D_O \rightarrow \mathbb{N}\}$
Input: Binary gray level [0, 255]
Output: label image O_L $\{O_L : D_{O_L} \rightarrow \mathbb{N}\}$
Initialization: $INIT \leftarrow$ **-1**; *Current Label* $\leftarrow$ **0**;
for all $p \in D_L$ **do**
  $O_L(p) \leftarrow INIT$;
end for
Let G be a subset of $\mathbb{Z}^2 \times \mathbb{Z}^2$ and $N_G(p)$ stands for the neighbors of a pixel p on the grid G
for all $p \in D_O$ with $(O_L(p) == INIT)$ and $(O(p) == B_L)$ **do**
  $O_L(p) \leftarrow$ *Current Label*
  FIFO INIT(Q);{Initialize the queue Q}
  FIFO ADD(p, Q);{Enqueue the pixel p}
  while Queue Q not empty **do**
    Dequeue the pixel p from queue Q;
    $p \leftarrow$ FIFO REMOVE(Q);{Dequeue the pixel p}
    for all $q \in N_G(p)$ with $O_L(q) = INIT$ **do**
      if $O(p) == O(q)$ **then**
        $O_L(q) \leftarrow$ *Current Label*
        FIFO ADD(q, Q);{Enqueue the neighboring pixel q}
      end if
    end for
  end while
  Current Label := *Current Label* + 1;
end for

Complexity of blob detection algorithm: Let us denote by n the number of image pixels in the image O whose domain is denoted as $\mathbf{D}_O \subset \mathbb{Z}^2$, and $\mathbf{G}$ is a subset of $\mathbb{Z}^2 \times \mathbb{Z}^2$. $N_G(p)$ stands for the neighbors of a pixel p on the grid $\mathbf{G}$. Let $B_1, B_2, \ldots, B_P$ be P blobs, each B_i having n_i pixels. Total number of comparisons for blobs detection and

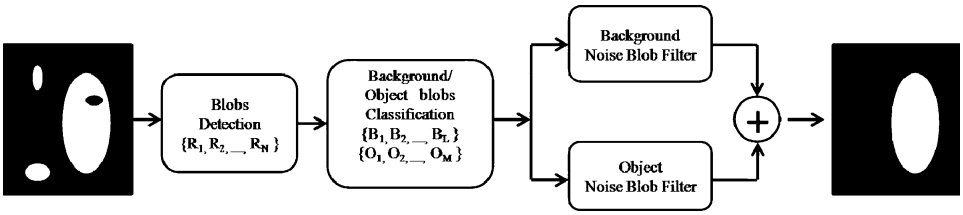

Fig. 2 Procedure for object blob detection

labeling (on an average) is $n + N_G * \left(\sum_{i=1}^{P} n_i\right)$, where N_G stands for neighborhood pixels on the grid G. Thus the computational complexity of the proposed blobs detection and labeling (approximately) is $O(n)$.

3.4.2 Blob filtering

In this procedure, a two-level classification criterion has been utilized to classify the moving object and background from the noise blobs. First, we separate the connected component regions which are labeled in the blob detection procedure, into background and object blobs based on their intensity value, zero for background and one for object regions. Then, the background blobs are further classified as a background blob and the object noise blobs, similarly, the object blobs are separated as an object blob and the background noise blobs based on maximum area criteria. Let $B_{\text{blobs}} = \{B_1, B_2, \ldots, B_L\}$ and $O_{\text{blobs}} = \{O_1, O_2, \ldots, O_M\}$ be set of background and object blobs, respectively, for a given set of connected component regions, $R = \{R_1, R_2, \ldots, R_N\}$ in an binary image. The background blobs B_{blobs} can be classified as a background blob, $B = \max_{\text{area}} \{B_1, B_2, \ldots, B_L\}$ and object noise blobs, $O_{\text{nblobs}} = \{B_1, B_2, \ldots, B_L\} - \{B\}$. Similarly, we classify the object blobs O_{blobs} as an object blob, MO = $\max_{\text{area}} \{O_1, O_2, \ldots, O_M\}$ and background noise blobs, $B_{\text{nblobs}} = \{O_1, O_2, \ldots, O_M\} - \{\text{MO}\}$.

The Background noise blob filter, first detects the noise blobs O_{nblobs} in the background B and then, merges the detected noise blobs with the background B. Similarly, the object noise blob filter first detects the noise blobs B_{nblobs} in the object MO and then re-labels the detected noise blobs as moving object MO. The final object MO_i can be obtained by performing the logical OR operation between the filtered outputs.

3.5 Background update

This section describes the modified background update method that update the background model with the incoming images. An adaptive scheme makes possible a constant updating of previous background and the present frame. The modified update scheme is different for pixel positions which are detected as belongs to foreground, as ghosts and part of the background:

$$B_{i+1}(x,y) = \begin{cases} B_i(x,y) & \text{If}(x,y) \in \left(MO_i \vee O_i^{SH}\right) \\ & (\text{Object} \vee \text{shadow} \vee \text{highlight}) \\ I_i(x,y) & \text{Else if}(x,y) \in \left(O_i - \left(MO_i \vee O_i^{SH}\right)\right) \\ & (\text{Small motion in the background}) \\ \alpha B_i(x,y) + (1-\alpha) I_i(x,y) & \text{Otherwise} \end{cases}$$

where B_{i+1} is the updated background and $\alpha \in (0,1)$ is a time constant that defines background adaptation speed and determines the sensitivity of the background update to the variations. The co-efficient α serves as parameter that controls the rate of the adaptation of the background to the current frame. For each incoming frame, we update the background model $B = \langle B^H, B^S, B^V \rangle$ dynamically based on the pixel position (x, y) in the classified regions, namely moving object MO_i, shadows and highlight (O^{SH}), ghost regions $\left(O_i - \left(\text{MO}_i \vee O_i^{\text{SH}}\right)\right.$ where the background regions have small motion and background noise due to change in illumination. The ghost regions are replaced in the background model with their corresponding intensity values of incoming frame, and the noise regions due to change illumination are updated linearly. However, the background model remains the same for pixels belonging to moving object and shadow regions. Hence, the background update method tends to track the dynamic changes in the background and also robust against the noise and illumination changes.

3.6 Complexity analysis

Let us define n to be the number of pixels in the gray scale image F whose domain is denoted as $D_F \subset \mathbb{Z}^2$. The number of operations to perform the core object detection is $17*n$, which involves $6*n$ differences and $11*n$ logical operations. The present shadows and highlight detection step requires only $2n$ operations, and post-processing requires only three raster scans for entire computation. Moreover, the background update requires 'n', the number of operations. Thus the total number of operations for the entire computation per frame (in the worst case) is $17*n + 2*n + 3*n + n$. The computational complexity(per frame) of the proposed method is then $O(n)$.

4 Experimental results

The proposed hybrid background technique has been coded with *OpenCV* Library and implemented in Pentium 1.7 GHz, and then tested in real-time with the Sony CCD camera, 320 × 240, 30 fps. Figures 3 and 4 illustrate the simulation results from the proposed method for own captured video sequences "ROOM" and "WALL" respectively. The "ROOM" sequence contains shadow effects of moving object can be seen on the floor. The goal here to detect the moving object without detecting the various illumination influences such as shadows, reflections, etc. Moreover, the "WALL" sequence contains wall as background in which a moving object gestures in the scene that changes illumination in the background. The segmented moving object is evaluated with the ground truth based on the pixel based measuring scheme [19].

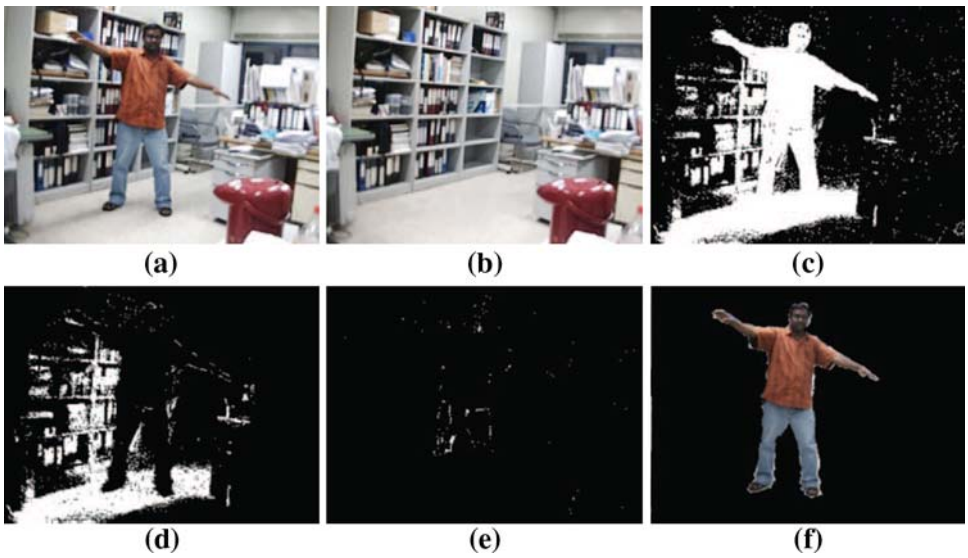

Fig. 3 **a** The current frame from "ROOM" video sequence, **b** mean of the background learnt from 300 frames, **c** segmented object, **d** shadows and illumination changes, **e** highlight detection, and **f** final object silhouette

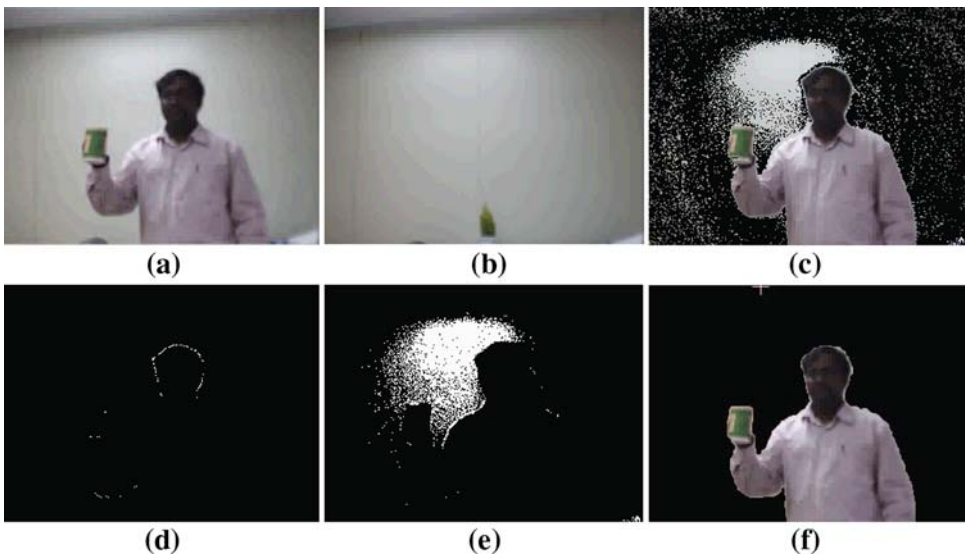

Fig. 4 **a** The current frame from "WALL" video sequence, **b** mean of the background learnt from 300 frames, **c** segmented object, **d** shadows and illumination changes, **e** highlight detection, and **f** final object silhouette

4.1 Performance evaluation

The proposed and conventional background subtraction methods were evaluated by manually creating the ground-truth foreground image. The ground-truth was developed by an automatic segmentation of the video sequence followed by a manually correction using a graphical interface. The methods were then evaluated based on the foreground detection accuracy (segmentation quality) and the detection error rates, namely miss detection rate and false alarm rate. Given three images, including the input image, the background image and the ground-truth foreground image, the foreground detection accuracy is equal to the ratio of the number of the ground-truth pixels detected as foreground to the total number of ground-truth pixels. Moreover, the foreground detection error rate, which is the ratio of the number of false foreground pixels to the total number of detected foreground pixels to the total number of detected foreground pixels. The performance of proposed scheme depends on a set of parameters, namely, the confident level β and the background learning rate α. The ground-truth and the foreground images from running the proposed method for different confident levels, are depicted in the Fig. 5. From Fig. 5, it is observed that all of the lost foreground pixels and the error pixels are near the foreground silhouettes.

Figure 6 shows the evaluation of the proposed method based on the segmentation quality and error rates for different confident levels. In this experiment, we kept the background learning rate α as constant and then varied the confident level in the attempt to obtaining the best segmentation quality.

As sown in Fig. 6, a larger confident level lead to a larger threshold ($\beta\sigma$), which intern gives less miss

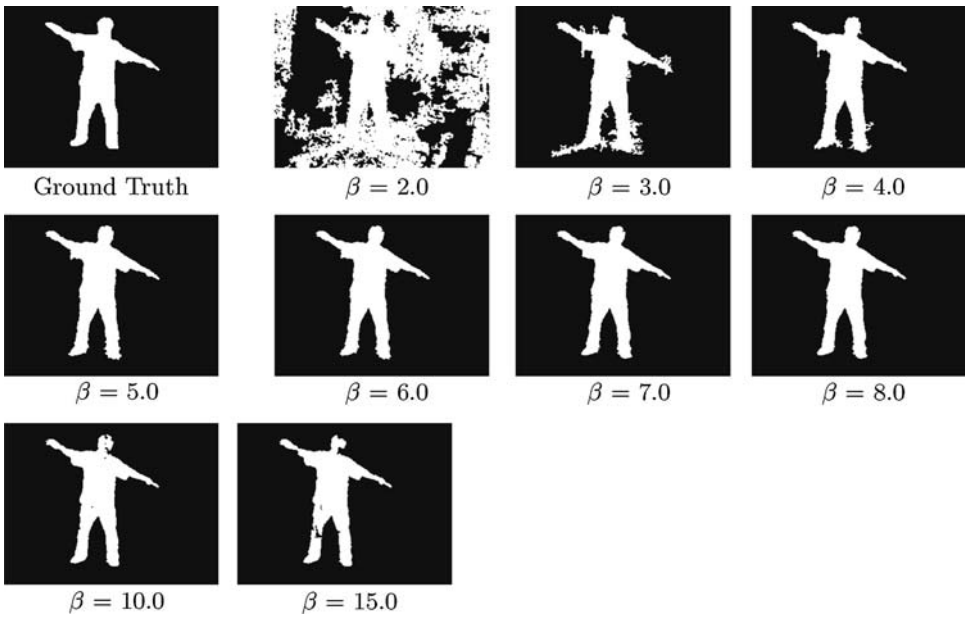

Fig. 5 Shows the ground-truth image and the foreground images for different confident levels

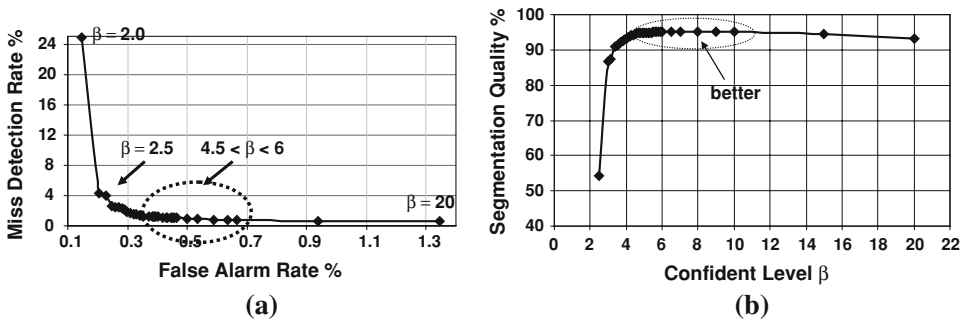

Fig. 6 **a** Shows ROC for different values of β and **b** shows the performance of the proposed scheme.

detection rate but more false alarm rate. Hence, the area under the ROC from the proposed scheme is very less and close to origin. For values of confident level (β) between 4.5 and 6, the proposed algorithm has high segmentation quality and low error rates in the different environments. As a trade-off between the segmentation quality and the error rates, the confident level can be set to 5. In this experiment, we set the confident level based on the ROC and the environment conditions.

4.2 Comparison and analysis

The moving object detection results and efficiencies of the proposed method were compared with the conventional Normalized RGB-based background subtraction method [8] as shown in Fig. 7. Table 1 lists the overall segmentation quality and error rates, and an average execution time of each frame with the two algorithms.

As shown in Table 1, the proposed method, by comparison, achieved good segmentation quality (is around 92%) while compared to the conventional method. Moreover, the proposed approach has been executed in real-time. The average execution time was less than 50 ms, much less than the execution time of the conventional Normalized *RGB* method. Moreover, we perform simulations with different conventional background techniques [8, 12, 13]. Table 2 summarizes the experiments by showing the precision, recall and segmentation quality calculated for each method. The best performance(segmentation quality) shown by proposed and Horprasert's method [13]. The precision can be used to analyze the tendency of an algorithm to oversegment. The higher the precision, the less likely is over-segmentation. The algorithms with lowest over-segmenation are proposed, Horprasert's [13] and Hong's [8] method with 95–80%. On the other hand, the recall can be used to estimate the tendency of under-segmentation. The higher the recall the less likely is under-segmentation. The proposed, Horprasert's [13] and Francois's [12] method shows lowest under-segmentation with 90–85% recall. By comparing the objective quality of methods it was noticed that the proposed method outperformed with the best precision and recall. Apart from the segmentation quality, the computational complexity of an algorithm is another important criteria for real-time

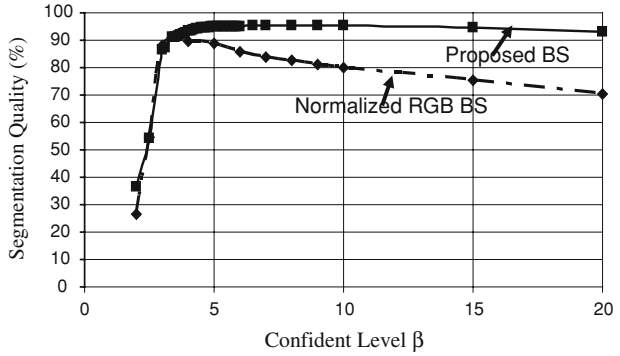

Fig. 7 Performance of the both background subtraction methods

Table 1 Overall segmentation quality and error rates of the proposed and conventional method [8]

Performance			
Algorithm	False positive (%)	False negative (%)	Segmentation quality (%)
NRGB [8]	2–3.5	1.5–5	75–80
Proposed	0.5–1.5	0.3–1.0	90–94
Time per frame			
NRGB [8]	0.3 s		
Proposed	0.05 s		

Table 2 Objective measures and mean time consumption for different background subtraction methods

Algorithm	Objective measures			Mean time per frame (s)
	Precision (%)	Recall (%)	Segmentation quality (%)	
Francois [12]	70	90	78	0.045
Horprasert [13]	83	85	84	0.25
NRGB [8]	83	77	79	0.3
Proposed	95	90	93	0.05

applications. Computational complexity can be split into two parts, namely time and memory consumption. It is true for real-time applications where a given frame rate must be achieved. Table 3 lists mean time per frame of proposed and conventional algorithms for video sequences "ROOM" and "WALL", and also summarizes the memory consumption of background models. From the simulation results, the fastest methods are proposed and Francois's method [12] with average time per frame is less than 50 ms which is much less than the other conventional methods [8, 13]. We considered memory consumption in terms of the amount of memory used for one background model. As for

Table 3 Mean time per frame (s) of proposed and conventional algorithms, and memory consumption (MB) of background models

Algorithm	Mean time per frame (s)		Memory consumption of background model
	"ROOM"	"WALL"	
Francois [12]	0.045	0.042	2.4
Horprasert [13]	0.25	0.22	3.2
NRGB [8]	0.3	0.28	4.8
Proposed	0.05	0.43	1.2

implementation, we find that the proposed method and Horprasert's method [13] uses less memory in the range between 1.2–2.5 MB which is less than the amount of memory used by Francois's [12] method with 3.2MB and Hong's [8] method with 4.8 MB. Hence, the proposed method is accurate and fast enough to be implemented in the Interactive Interface Systems.

4.3 VR@Home [20, 21]

The proposed scheme has been tested with the VR@Home platform [20, 21]. VR@Home is a personal virtual reality (VR) studio for personal broadcasting at home. Recently, many people have tried to create their own contents and to share them with others through internet. The broadcasting has been being generalized. However, there have been not many easy tools which can help users to produce creative programs. The proposed VR@Home is a new software platform which enables end-users at home to install a personal virtual studio capable of creating realistic 3D contents. This platform helps to create the realistic 3D scene by merging the real world and the virtual contents, and also enables the users to interact with virtual contents through the intuitive user interface. The latest version described in [20, 21] roughly consists of six components as shown in the Fig. 8.

The platform supports two processes, the off-line and the online process. In the off-line process, users are able to generate realistic 3D models or 3D backgrounds with off-the-shelf cameras. The flexible camera calibration and voxel-based modeling techniques are included in the platform. In the online process (during broadcasting), users are able to interact with virtual objects obtained in the off-line process by adopting recent augmented reality (AR) technologies. Especially, tangible AR interfaces are included in the platform. In addition, it provides the shadow generation module based on the light source estimation method. Shadows of the augmented virtual objects improve the immersiveness visually. With the help of the background subtraction method proposed in this paper, users can also substitute their background in real time.

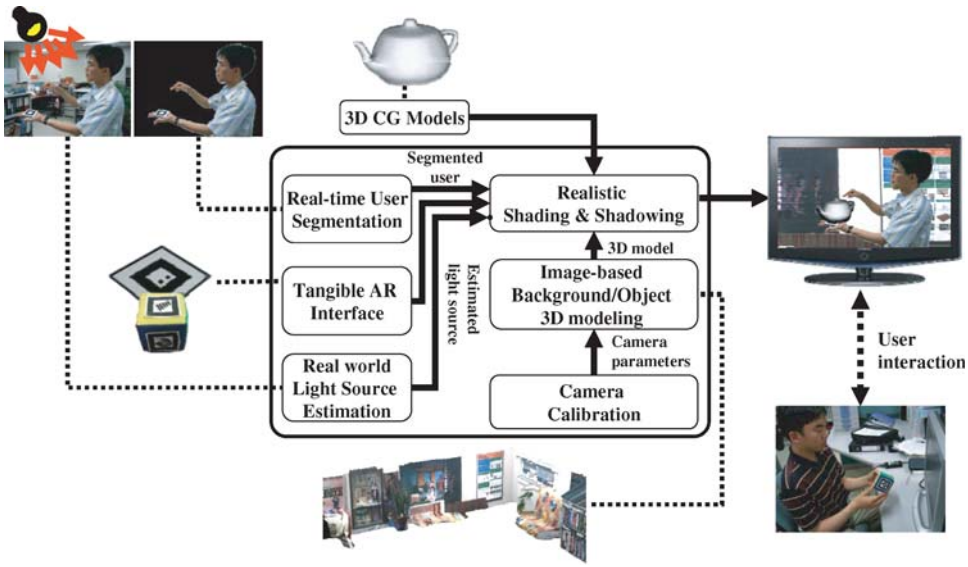

Fig. 8 Flow diagram of VR@Home: a personal virtual reality(*VR*) studio for personal broadcasting at home

5 Conclusions

A hybrid background subtraction method using HSV color model has been presented to adapt background noisy and various illumination conditions. The proposed method try to free of several disadvantages encountered by conventional techniques based on the background subtraction criterion. A statistical background model was firstly set up by obtaining the mean and variance of each pixel's color components from the first N static background frames without any moving objects. The moving object regions are well identified by fusing the results from background difference and motion-based change detection criterion. The foreground pixels are well distinguished from the background by introducing the confident level thresholding and the background regions with small motion are well identified by comparing the consecutive frame difference with the mean of the difference image. The paper introduced a methodology for detecting the shadows and highlights, and also proposed a novel connected component analysis method for isolating the moving object blob from the background noise. Nevertheless, the proposed method also introduced a robust background update procedure for tracking the dynamic changes in the incoming images. The computational complexity of the proposed method has been analyzed.

The proposed scheme has been implemented and executed in real-time, and evaluated regarding the segmentation quality and frame rate. The results of running the proposed and the conventional methods on the test sequence clearly establish the superiority of the proposed method. From the experimental results, we conclude that the proposed method successfully extract object silhouette more accurately against the noise and illumination changes, while compared to the conventional methods. The proposed scheme has been tested in the VR@Home platform.

References

1. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 780–785 (1997)
2. Hu, S., Mortensen, J., Buxton, B.F.: A real-time tracking system developed for an interactive stage performance. Trans. Eng. Comput. Technol. **5**, 102–105 (2005)
3. Haritaoglu, I., Harwood, D., Davis, L.S.: W^4: real-time surveillance of people and their activities. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 809–830 (2000)
4. Zhang, R., C., V., Metaxas, D.: Human gait recognition. In: IEEE Workshop on Articulated and Nonrigid Motion (in conjunction with CVPR). Rutgers University, Piscataway, pp. 18–18 (2004)
5. Mlayim, Y., U.Y., Atalay, V.: Silhouette-based 3D model reconstruction from multiple images. IEEE Trans. Syst. Man Cybern. **B33**(4), 582–591 (2003)
6. Wang, D.: Unsupervised video segmentation based on watersheds and temporal tracking. IEEE Trans. Circuits Syst. Video Technol. **8**(5), 539–546 (1998)
7. Emrullah, D., Touradj, E.: Change detection and background extraction by linear algebra. Proc. IEEE. **89**(10), 1368–1381 (2001)
8. Hong, D., Woo, W.: A background subtraction for a vision-based user interface. In: Proceedings of ICICS-PCM. Singapore 1B3.3.1–5 (2003)
9. Spagnolo, P., Leo, M., Attolico, G., Distante, A.: A supervised approach in background modelling for visual surveillance. In: Audio- and Video-Based Biometric Person Authentication. LNCS, vol. 2688. Springer, Berlin, pp. 592–599 (2004)
10. Chien, S.Y., Ma, S.Y., Chen, L.G.: Efficient moving object segmentation algorithm using background registration technique. IEEE Trans. Circuits Syst. Video Technol **12**(7), 577–586 (2002)
11. Zhao, J.M., Chen, C.: Robust background subtraction in HSV color space. In: Proceedings of SPIE: Multimedia Systems and Applications, vol. 4861, pp. 325–332 (2002)

12. Francois, A., Medioni, G.G.: Adaptive color background modeling for real time segmentation of video streams. In: Proceedings of International Conference on Imaging Science, Systems, and Technology. Vegas, NA (1999)
13. Horprasert, T., D.H., Davis, L.: A statistical approach for real time robust background subtraction and shadow detection. In: IEEE Frame Rate Workshop (1999)
14. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: International Conference on Computer Vision, pp. 780–785 (1999)
15. Harville, M.: A framework for high-level feedback to adaptive per-pixel mixture of gaussian models. In: Proceedings of European Conference on Computer Vision, vol. III. Springer, London, pp. 543–560 (2002)
16. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, Fort Collins, CO, USA, pp. 248–252 (1999)
17. Elgammal, A., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Proceedings of the 6th European Conference on Computer Vision, vol. III. Springer, London, pp. 751–767 (2000)
18. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting moving shadows: algorithms and evaluation. IEEE Trans. Pattern Anal. Mach. Intell. **25**(7), 918–923 (2003)
19. Chalidabhongse, T.H., Kim, K., Harwood, D., Davis, L.: A perturbation method for evaluating background subtraction algorithms. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 2003) (2003)
20. Lee, W., Kim, K., Rambabu, C., Yu, J., Lee, J., Lee, K., Woo, W.: VR@Home: A personal VR studio platform. In: Proceeding of Fourth International Symposium on Ubiquitous VR, vol. 191, GIST, U-VR Lab, S. Korea, pp. 53–56 (2006)
21. Wonwoo, Lee, Rambabu, C., Woontack Woo, J.L.: VR@Home: an immersive contents creation system for 3D user-generated contents. In: Technologies for E-Learning and Digital Entertainment. LNCS, vol. 4469. Springer, Berlin, pp. 81–91 (2007)

Author Biographies

Chinta Rambabu received his B.E degree in Electronics and Communication Engineering from AU, India in 1995, M. Tech in Automation and Computer Vision from E&ECE, IIT Kharagpur, India in 1998 and Ph.D. from IIT Guwahati, India in 2005. From 2005 to 2006, he worked as a Research Fellow in UVR Lab, GIST, South Korea. He is currently employed as a Research Fellow at Bioinformatics Institute (BII), Imaging Group, Singapore. His areas of interest are computer vision, image/video processing, Multi-dimensional Microscopic image analysis and VLSI Signal processing. He has published several papers in international journals and conferences in these areas.

Kiyoung Kim received his B.S. in Department of Computer Science from Chung-ang University, Seoul, Korea in 2003 and M.S. in the Department of Information and Communication from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2004. He is a Ph.D. student in DIC, GIST since 2004. His research interests include 3D computer vision, computer graphics, and marker-less tracking for augmented reality, etc.

Woontack Woo received his B.S. in Electronics Engineering from Kyungpook National University in 1989 and his M.S. in Electronics and Electrical Engineering from POSTECH in 1991. In 1998, he received his Ph.D. in Electrical Engineering Systems from University of Southern California (USC). In 1999, as an invited researcher, he joined Advanced Telecommunications Research (ATR), Kyoto, Japan. Since February 2001, he has been with the Gwangju Institute of Science and Technology (GIST), where he is an Associate Professor in the Department of Information and Communications (DIC) and Director of Culture Technology Institute (CTI). His research interests include 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for ubiquitous computing, etc.