Lim and Kang *EURASIP Journal on Advances in Signal Processing* 2011, **2011**:60
http://asp.eurasipjournals.com/content/2011/1/60

EURASIP Journal on
Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

# Object tracking system using a VSW algorithm based on color and point features

Hye-Youn Lim and Dae-Seong Kang*

**Abstract**

An object tracking system using a variable search window (VSW) algorithm based on color and feature points is proposed. A meanshift algorithm is an object tracking technique that works according to color probability distributions. An advantage of this algorithm based on color is that it is robust to specific color objects; however, a disadvantage is that it is sensitive to non-specific color objects due to illumination and noise. Therefore, to offset this weakness, it presents the VSW algorithm based on robust feature points for the accurate tracking of moving objects. The proposed method extracts the feature points of a detected object which is the region of interest (ROI), and generates a VSW using the given information which is the positions of extracted feature points. The goal of this paper is to achieve an efficient and effective object tracking system that meets the accurate tracking of moving objects. Through experiments, the object tracking system is implemented that it performs more precisely than existing techniques.

**Keywords:** background modeling, meanshift, object tracking, search window, SIFT

## 1. Introduction

The object tracking means tracing the progress of objects as they move over a sequence of images. Visual object tracking in complex environments is an important topic in the intelligent surveillance field. A good tracking algorithm should be able to work well in many real circumstances, such as background clutters, occlusions, and different illuminations [1,2].

The object tracking methods can be divided into three groups according to feature values of the object which seems to be foreground: color-based method, boundary-based method, and model-based method. The color-based method is used for the color probability distribution of object tracking. Typical color-based methods are meanshift and continuously adaptive meanshift (Camshift) algorithms [3,4]. It should facilitate fast calculation because of the simple implementation. Therefore, the color-based method is widely used in object tracking because it is easy to extract and robust against partial occlusion. However, they are vulnerable to sudden illumination changes and backgrounds with similar colors. The boundary-based method is used for contour information of object such as condensation algorithm [5]. It is suitable for tracking rigid object which seldom changes their boundaries such as the heads of people. However, it is difficult for real-time processing because of complicated calculations. The model-based method, in other words motion templates, tracks the object after learning the templates in advance [6]. As a result, these methods are combined to achieve more robust tracking results such as tracking algorithm based on combining the color and boundary-based methods [7,8].

The tracking algorithm proposed in this article employs point-based and color-based multiple features, i.e., it is an effective improvement of meanshift with scale invariant feature transform (SIFT) algorithm. It is a method that extracts the feature points of a detected object, and generates a variable search window (VSW) using the given information. This information is the positions of extracted feature points. This approach can solve the problem of a similar color distribution and improve the performance of the object tracking. The main contributions of this article are as follows: (1) the improvement of meanshift with SIFT algorithm is proposed for the object tracking, and (2) the performance of the proposed tracking algorithm can experimentally be proved against the existing algorithms.

---

* Correspondence: dskang@dau.ac.kr
Department of Electronics Engineering, Dong-A University, 840 Hadan 2-Dong, Saha-Gu, Busan, Korea

The rest of this article is organized as follows. In Section 2, Gaussian mixture modeling (GMM) and post-processing of detected objects are introduced. Typical tracking methods and the proposed tracking algorithm are described in Sections 3. The experimental results and the tracking performances are in Section 4. Finally, conclusions are given in Section 5.

## 2. Object detection

Most methods for object detection are based on per-pixel background models [9-12]. A pixel-based method does not consider the general things in the frame and therefore shadows and noise must be handled afterwards. A flowchart for the object detection method is shown in Figure 1.

### 2.1. Gaussian mixture model

A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities [13,14]. This method is suggested by Stauffer et al., which models each pixel as a mixture of Gaussian distributions and uses an online approximation to update the model. The model assumes that each pixel in the frame is modeled by a mixture of $K$ Gaussian distributions where different Gaussian distributions represent different colors. The probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta(X_t, \mu_{i,t}, \sum_{i,t}), \quad (1)$$

where $K$ is the number of the distributions, $\omega_{i,\ t}$ is an estimate of the weight (what portion of the data is accounted for by this Gaussian) of the $i$th Gaussian in the mixture at time $t$, $\mu_{i,\ t}$ is the mean value of the $i$th Gaussian in the mixture at time $t$, $\Sigma_{i,\ t}$ is the covariance matrix of the $i$th Gaussian in the mixture at time $t$, $X_t$ is a random variable vector and $\eta(X_t, \mu_{i,\ t}, \Sigma_{i,\ t})$ is a Gaussian probability density function.

A pixel value $X_t$ that matches the Gaussian distribution can be defined as

$$\begin{cases} \text{matching}: & |X_t - \mu_{i,t}| < \lambda\sigma_{i,t} \\ \text{ummatching}: & |X_t - \mu_{i,t}| \geq \lambda\sigma_{i,t} \end{cases} \quad i = 1, 2, \cdots, K, \quad (2)$$

where $\lambda$ is 2.5 and $\sigma$ is standard deviation. So, a match is defined as a pixel value within 2.5 standard deviations of a distribution.

The prior weights of the $K$ distributions at time $t$ are adjusted as follows:

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha(M_{i,t}), \quad (3)$$

where $M_{i,\ t}$ is 1 for the model matched and 0 for the remaining models, and $\alpha$ is a learning rate.

The mean and the variance parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows:

$$\mu_{i,t} = (1-\rho)\mu_{i,t-1} + \rho X_t, \quad (4)$$

$$\sigma_{i,t}^2 = (1-\rho)\sigma_{i,t-1}^2 + \rho(X_t - \mu_{i,t})^T(X_t - \mu_{i,t-1}), \quad (5)$$

where $\rho = \alpha\eta(X_t|\mu_i,\sigma_i)$.

If $X_t$ does not match any Gaussian distributions, the least probable distribution is replaced with a new distribution which has its mean value, an initially high variance and low prior weight.

After the updates, all the components in the mixture are ordered by the value of $\omega/\sigma$. Then, the first $B$ distributions which exceed certain threshold $T_{\text{bg}}$ are retained for a background distribution and $B$ can be defined as

$$B = \arg\min_b \left( \sum_{k=1}^{b} \omega_k > T_{\text{bg}} \right), \quad (6)$$

where $T_{\text{bg}}$ is a measure of the minimum portion of the data that should be accounted for by the background.

### 2.2. Post-processing

The GMM is used for the segmentation, extraction of objects, and background area. However, detected objects can contain noise types such as shadows and illuminations. Therefore, it needs to remove the shadows using morphological filters. A deterministic non-model-based approach among shadow removal techniques is used for general surroundings. This approach is based on the fact that it can consider a pixel as a shadow if it has similar

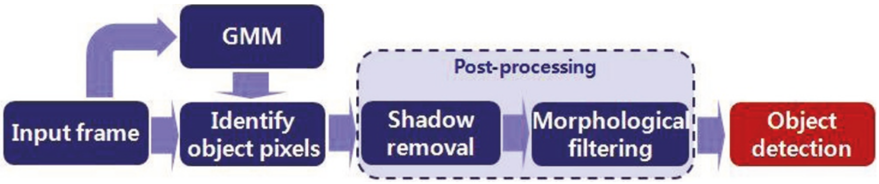


**Figure 1 Flowchart for the moving object detection based on background modeling**.

chromaticity but lower brightness than identical pixels in the background image. Equation 7 shows the decision as to whether or not a certain pixel is part of the shadow [15,16] as follows:

$$\text{Shadow} = \begin{cases} 1, \text{ if } & \text{BR}_{\text{img}} < \text{BR}_{\text{bg}}, \quad \& \quad \text{CH}_{\text{img}} = \text{CH}_{\text{bg}} \pm T \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

where $\text{BR}_{\text{img}}$ is the brightness of an input image, $\text{BR}_{\text{bg}}$ is the brightness of a background image, $\text{CH}_{\text{img}}$ is the chromaticity of an input image, $\text{CH}_{\text{bg}}$ is the chromaticity of a background image and $T$ is threshold value (= 0.5).

Figure 2 shows a comparison with the resulting images of the shadow removal and non-removal. The process of using the shadow removal considerably reduces a lot of the noises. So, the detected object becomes clearer.

## 3. Object tracking

### 3.1. Meanshift algorithm

The meanshift algorithm, which iteratively shifts a datum point to the average of data points in its neighborhood, is a robust statistical method. This algorithm finds local maxima in any probability distribution. It is used for tasks such as clustering, mode-seeking, probability density estimations, and tracking [17,18]. Table 1 shows the finding of the maximization of probability distribution using the meanshift algorithm [19].

### 3.2. SIFT algorithm

The SIFT algorithm was introduced by David G. Lowe, a professor at the University of British Columbia (UBC). This algorithm is used in various applications, such as feature extraction and matching. Figure 3 shows the steps of the SIFT algorithm. This is divided into the detector and descriptor categories largely. It generally has four steps [20,21]. In this article, we use detected feature points (= keypoints) using the SIFT algorithm, i.e., the proposed method is implemented until the extraction step of keypoints.

The first stage of computation searches over all scales and image locations. It is implemented efficiently with a difference-of-Gaussian (DOG) image to identify potential interest points that are invariant to scale and orientation. The scale space of an image is defined as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{8}$$

where $I(x,y)$ is an input image, $G(x,y,\sigma)$ is a variable-scale Gaussian, and * is the convolution operation.

Stable keypoint locations in scale space can be computed from the DOG separated by a constant multiplicative factor $k$:

$$\begin{aligned} D(x, y, \sigma) &= \left(G(x, y, k\sigma) - G(x, y, \sigma)\right) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \tag{9}$$

Figure 4 shows Gaussian and DOG pyramids of region of interest (ROI) in Video 1. For each octave of scale space, the initial image is repeatedly convolved with Gaussian to produce the set of scale space images shown in Figure 4a. Adjacent Gaussian images are subtracted to produce the DOG images in Figure 4b. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated. In Figure 4b, the DOG images show that values higher than 0 represent 255 because of seeing difference well, but real values of the DOG images are 0 or low values.

To detect the local maxima and minima of $D(x,y,\sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below. At each candidate location, a detailed model

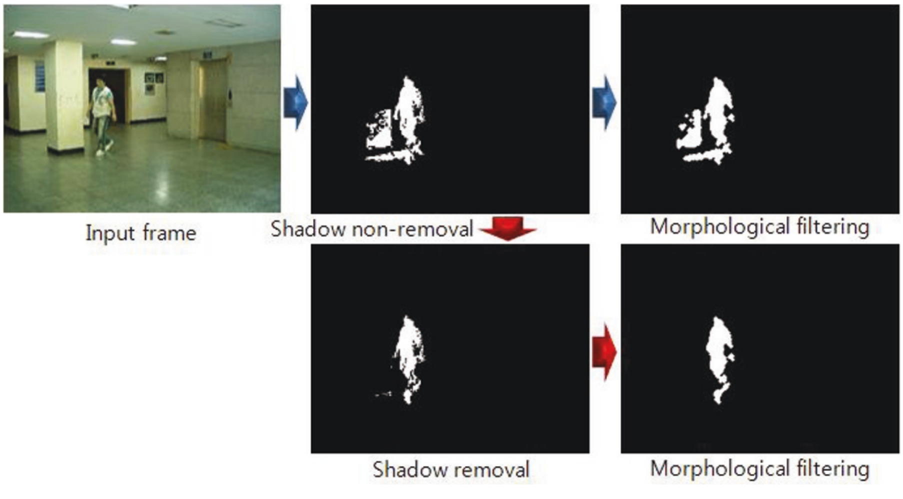


**Figure 2 A comparison of the resulting images of shadow removal and non-removal.**

**Table 1 Meanshift algorithm for finding the maximization of probability distribution**

Given the distribution $\{q_u\}_{u=1,\ldots,m}$ of a detected object and the location $y_0$ of the detected object in the previous frame.

Step 1. Compute the distribution $p_u$ in the current frame with $y_0$ and compute the Bhattacharyya distance (= $\rho$) between $p_u$ and $q_u$ as follows:

$$\rho\left(p(y_0), q\right) = \sum_{u=1}^{n} \sqrt{p_u(y_0) q_u}.$$

Step 2. Compute the weight ($w_i$) is defined as follow:

$$w_i = \sum_{u=1}^{m} \delta\left[b(x_i) - u\right] \sqrt{\frac{q_u}{p_u(y_0)}},$$

where $b(x_i)$ is the color bin of the color at $x_i$.

Step 3. Derive the new location $y_1$ can be defined as follow:

$$y_1 = \frac{\sum_{i=1}^{n} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)},$$

where $h$ is a window radius and $g()$ is a kernel $G$.

Step 4. Compute the Bhattacharyya distance (= $\rho$) between $p_u$ and $q_u$ in the new location $y_1$.

Step 5. While $\rho|p(y_0), q| < \rho|p(y_1), q|$
Do $y_1 = 1/2(y_0+y_1)$.

Step 6. If $\|y_1\text{-}y_0\| < \varepsilon$ Stop.
Otherwise, $y_0 = y_1$ and go to Step 1.

is fit to determine the location and scale. Keypoints are selected based on measures of their stability. If this value is below a threshold, signifying that the structure has low contrast (sensitive to noise), the keypoint will be removed. For poorly defined peaks in scale-normalized Laplacian of Gaussian operators, the ratio of the principal curvatures of each candidate keypoint is evaluated. If the ratio is below a threshold, the keypoint is retained. Figure 5 shows the extraction of feature using the SIFT algorithm in Video 1. Figure 5a has 184 features in the whole region of a frame, and Figure 5b has 32 features in the ROI. Therefore, we reduce processing time due to extract feature points in the ROI only.

### 3.3 VSW algorithm

The proposed tracking system has three steps. The first step involves background modeling construction using the GMM. The second step is an execution of the post-processing of the detected objects for noise removal. The last step is the tracking of the moving object using the VSW algorithm. Finally, the proposed tracking system finds the most accurate object through a new search window. During the meanshift tracking, a color histogram can easily be computed. However, this process does not update the size of a search window and convergence into a local maxima point is easily done. Camshift tracking can update the search window. However, it is sensitive to non-specific color objects due to illumination and noise. It may include some similarly colored background areas that distract the tracking process. Therefore, to overcome this weakness, in this article, we present the VSW algorithm which generates a VSW with robust feature points for an accurate tracking of moving objects.

A flowchart for the whole system is shown in Figure 6. The detected object by means of background modeling

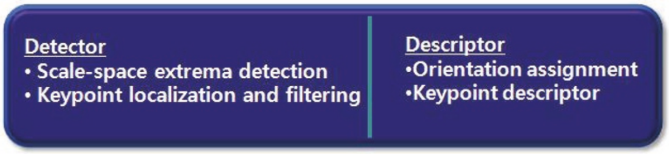
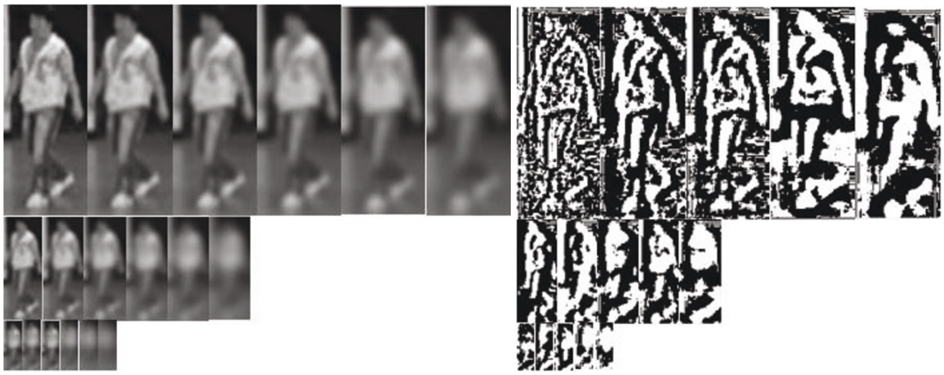
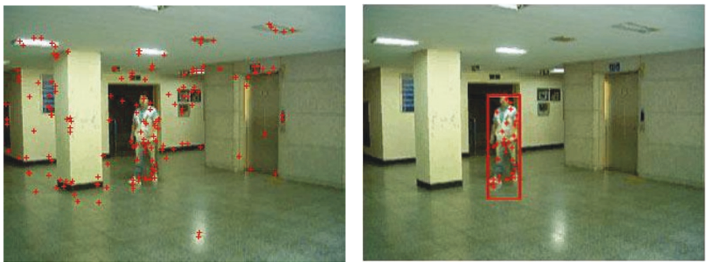

Detector
• Scale-space extrema detection
• Keypoint localization and filtering
Descriptor
•Orientation assignment
•Keypoint descriptor

**Figure 3 Steps of the SIFT algorithm**.



(a) (b)

**Figure 4 Gaussian and DOG pyramids of ROI in Video 1 (160 frame)**. **(a)** Gaussian pyramid. **(b)** DOG pyramid.



(a) (b)

**Figure 5 Extraction of feature points using the SIFT algorithm in Video 1 (140 frame)**. **(a)** 184 features in the whole region. **(b)** 32 features in the ROI.

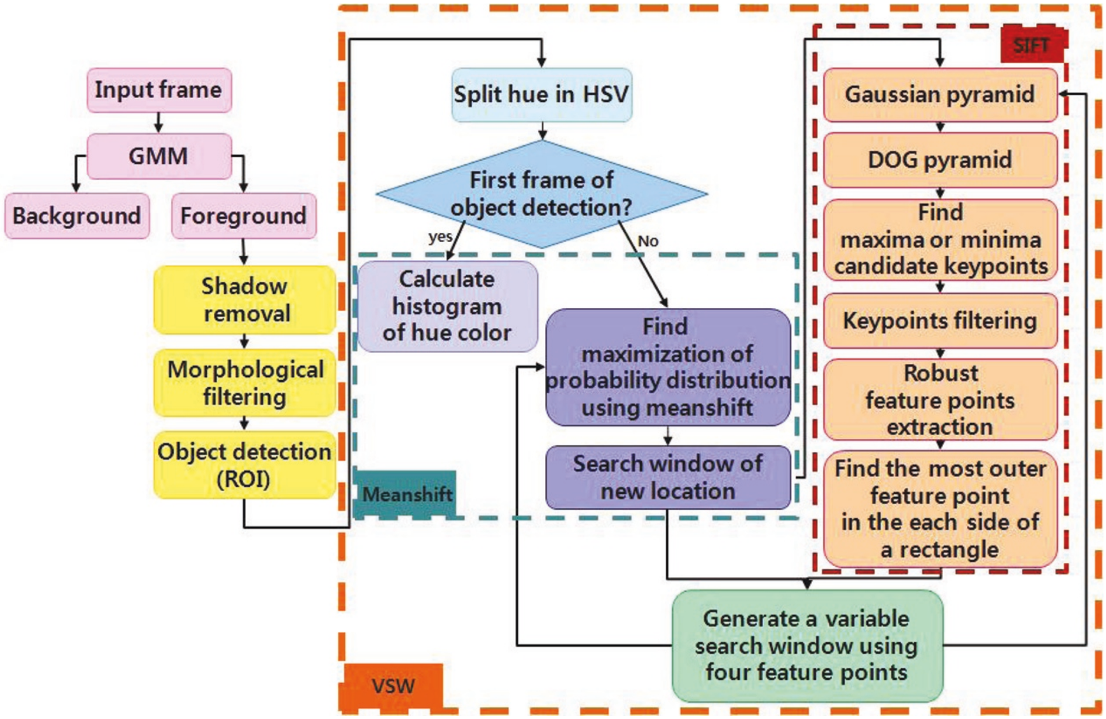

Input frame
GMM
Background
Foreground
Shadow removal
Morphological filtering
Object detection (ROI)
Split hue in HSV
First frame of object detection?
yes
No
Calculate histogram of hue color
Find maximization of probability distribution using meanshift
Search window of new location
Meanshift
SIFT
Gaussian pyramid
DOG pyramid
Find maxima or minima candidate keypoints
Keypoints filtering
Robust feature points extraction
Find the most outer feature point in the each side of a rectangle
Generate a variable search window using four feature points
VSW

**Figure 6** Flowchart for the whole system.

is set as the ROI. The next step involves the splitting off of the hue color in this region and calculation of a histogram of the hue color during the first frame of the object detection. The frames are found to maximize probability distribution using both the meanshift and a search window of new location. The search window is enlarged to the region at one pixel per each side of rectangle because enlarged region can prohibit extraction of non-feature point. The Gaussian pyramid images are created to this region, and the DOG pyramid images are created to the Gaussian pyramid images. Maxima or minima candidate keypoints are found in the DOG pyramid images. After filtering keypoints, robust feature points are extracted in only the ROI by the SIFT. The outermost feature point on the each side of a rectangle is found. We then take four feature points, and generate a VSW with them. Finally, we can track moving objects with the calculated histogram in advance, and create a new variable window. A blue dotted rectangle represents the meanshift. A red dotted rectangle indicates the SIFT. An orange dotted rectangle denotes the VSW algorithm.

Table 2 shows the steps of the proposed method. According to the condition of step 5, the object tracking is stopped or not. If the condition is true, then the proposed algorithm stops the object tracking, and if the condition is false, then the proposed algorithm continues the object tracking.

```
VSW algorithm: The proposed VSW algorithm
Track_window = track_object_rect + 1 (per the each side of rectangle)
Img = Track_window(ROI)
1. Feature extraction : SIFT
2. Search window change with feature points
For all $i$ such that $0 \leq i \leq n$ do ($n$ is the number of feature points)

  If $i == 0$ then (set min($x$, $y$) with first feature point)
    Min_x = Feature_$x_i$, Min_y = Feature_$y_i$
  Else if $i == 1$ then (set max($x$, $y$) with second feature point)
    Max_x = Feature_$x_i$, Max_y = Feature_$y_i$
      If Min_x > Max_x then Swap Min_x with Max_x
      End if
      If Min_y > Max_y then Swap Min_y with Max_y
      End if
  Else (set min($x$, $y$) and max($x$, $y$) with over third feature point)
      If Min_x > Feature_$x_i$ then Min_x = Feature_$x_i$
      End if
      If Min_y > Feature_$y_i$ then Min_y = Feature_$y_i$
      End if
      If Max_x > Feature_$x_i$ then Max_x = Feature_$x_i$
      End if
      If Max_y > Feature_$y_i$ then Max_y = Feature_$y_i$
      End if
  End if

End for

Track_object_rect = {(Min_x, Min_y), (Max_x, Max_y)}
```

**Table 2 The steps of the proposed algorithm**

| |
|---|
| Step 1. Assign an initial search window to ROI. |
| Step 2. Compute the color probability distribution of the search window and generate a new search window by the meanshift algorithm. (The new search window has the same size but the different location as the initial search window.) |
| Step 3. Extract the feature points for the SIFT algorithm around the new search window. |
| Step 4. Make a rectangle with four points of the edge among detected feature points and assign the new search window to the rectangle. |
| Step 5. Stop object tracking when the number of the feature points is under 4 or the location of the search window is not moved. Otherwise, repeat from Steps 2 to 4. |

The proposed algorithm shows the proposed VSW algorithm. Track_object_rect indicates a detected object place and track_window indicates a search window. To prohibit extraction non-feature point, the search window is enlarged to the region at one pixel per the each side of rectangle. The SIFT algorithm is used to feature points, and $n$ is the number of feature points. The most outer feature point in the each side of a rectangle is found for changing the search window. Therefore, we set four features into Min_$x$, Min_$y$, Max_$x$ and Max_$y$.

Figure 7 shows an example of the generation of a VSW. The left image in Figure 7 is the region which is enlarged one pixel per the each side of a rectangle of the detected object. A blue rectangle which represents the edge of an image belongs to the search window. The existing meanshift algorithm is used with this fixed search window. However, in this article, to compensate for the weakness this algorithm, we extract feature points with the SIFT algorithm in only the ROI. In Figure 7, the right image shows the extracted feature point as expressed the '+' sign. The outermost feature points among all feature points generate the red rectangle shown in the new search window. Thus, the search window then changes from the area in the blue to the area in the red rectangle. Generating a VSW in each frame can increase the accuracy of the object tracking performance.

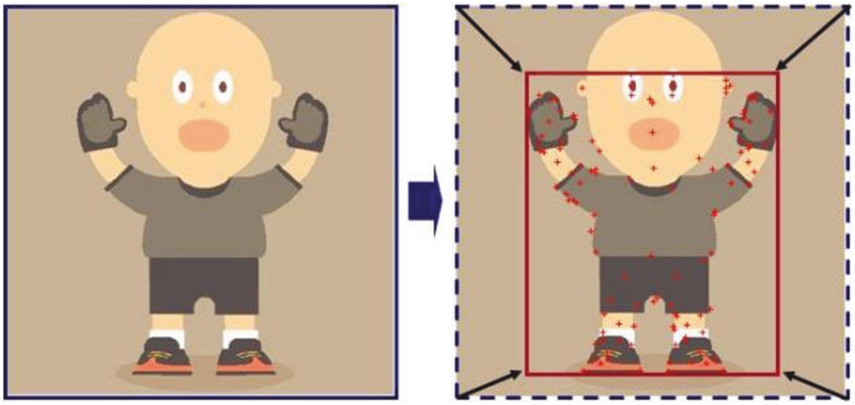


**Figure 7** An example of a generation of a VSW.

Figure 8 shows the resulting image of the generation of the VSW during the tracking of a moving object in an experiment. The left image is the resulting object tracking frame using the proposed method. The right image is the ROI of an enlarged section of the left image. Feature points are extracted by the SIFT algorithm in the region of object detection which is the region within the dark blue dotted line. The red '+' signs denote the feature points. The outermost feature points among the feature points generate a new search window, as denoted here by the red solid line forming the rectangle, i.e., we can generate a VSW in each frame for more accurate object tracking.

Figure 9 shows the processing of the generation of the VSW from input frame. Through the proposed method, the region of detected object is more detailed because the background region is deleted.

## 4. Experimental results

The proposed algorithm is implemented in Microsoft Visual C++ and carried out on a PC with a 2.0 GHz Intel Core 2 processor with 2 GB of memory. Table 3 shows the detailed information of each video sequence. In the experiments, four video sequences are used. Especially, Intelligent room [22] and Pets 2006 [23] videos are mainly used to evaluate performance of the object tracking system. The others are personally captured by the videos for the experiment.

To compute the tracking error, we create the ground-truth images, which are the images of the actual object region using Photoshop CS4. Figure 10 shows an example of the processing of the ground-truth images. We set a standard search window in the creation of the ground-truth image. In Figure 10, the center-point is

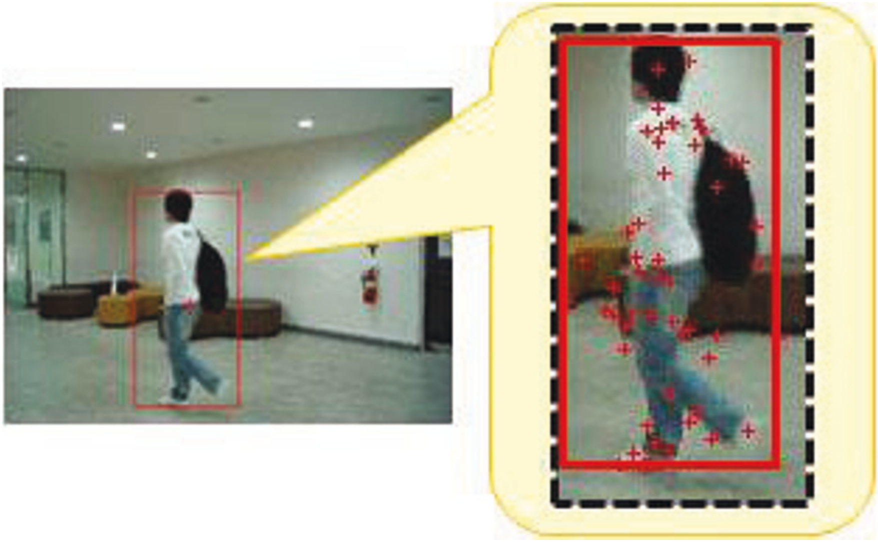


**Figure 8** The resulting image of the generation of the VSW while tracking a moving object.

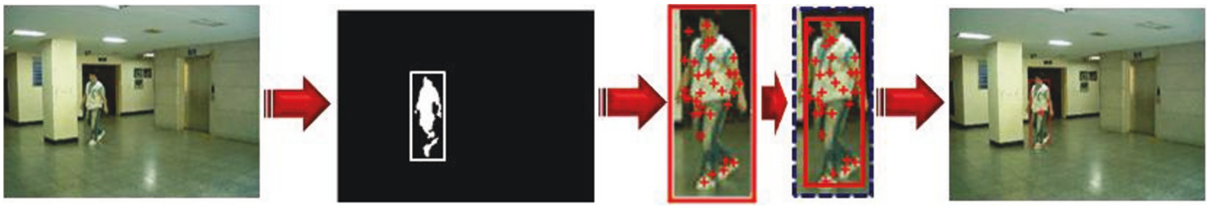


**Figure 9** **The processing of the generation of the VSW from input frame**.

denoted by a black '+' sign and the search window is denoted by the white rectangle. We set a new standard of distance error with the ground-truth images. It is marked visually every five frames after the initial detection of the object. Therefore, we can compare it with the detected object using different algorithms.

Figure 11 shows the comparison with resulting images of single-object tracking using each algorithm in Video 1 and Intelligent room. In Figure 11, a red rectangle denotes the proposed method, a green rectangle indicates the meanshift algorithm, a blue rectangle represents the Camshift algorithm, and a yellow rectangle denotes a meanshift + optical flow algorithm [24]. The proposed method tracked the object region more accurately than the other algorithms. Moreover, for the tracking of the object with the proposed method, it is clear that the search window is perfectly adapted to the size of the detected object. Through experimental results, color-based meanshift and Camshift algorithms missed the object because of illumination noise. The red rectangle as regards the size of the object can well change itself according to the variable size. When the size of object is bigger, the small red rectangle changes the big one.

Figure 12 shows the error comparison of the search window region using each algorithm in Video 1 and Intelligent room. To estimate accuracy of detected object in each algorithm, we measure the error of the region as follows:

$$\mathrm{ER} = \mathrm{MGR} + \mathrm{FSR}, \tag{10}$$

where ER is the error of the region, MGR is the miss region in ground-truth region, and FSR is the false region in search window region.

The criterion of ER is based on the region of detected object (= true region) in ground-truth images such as Figure 10. High value of ER means that the probability of false object tracking is high. In Figure 12, a blue dotted line has the most errors among the lines and a red dotted line has the least errors among the lines. Video 1 and Intelligent room got more error of the region as they went on.

Figures 13 and 14 show the comparison of the resulting images of multi-object tracking in Video 2 and Pets 2006 using each algorithm. Tracking multi-object using the meanshift algorithm based on color shows that several objects are missed in 661-frame of Video 2 and in 1002, 1040, 1136, 1175-frame of Pets 2006. Tracking multi-object using the Camshift algorithm based on color shows that nearby objects are not recognized multi-object but single-object in 619-frame of Video 2 and in 1040-frame of Pets 2006. Also, this tracking indicates that some objects among them are missed in 1002, 1175-frame of Pets 2006. Tracking based on color and feature points such as the meanshift + optical and the proposed method well tracks object. However, a tracking accuracy in the proposed method is higher than it in the meanshift + optical method, i.e., the proposed method is most similar with the object region of the ground-truth image.

Figure 15 shows the tracking result images of one-object among multi-object using the proposed method in Video 2 and Pets 2006. A red line represents the center of the search window, i.e., it is an object's route. Through experimental results, the proposed method can well track an object, cannot miss it among other objects.

Table 4 shows the comparison of the accuracy comparison for each algorithm. For estimating tracking accuracy, the accuracy is defined as follows:

$$\mathrm{accuracy}(\%) = \frac{\text{the number of total frames} - \text{the number of false tracking frames}}{\text{the number of total frames}} \times 100. \tag{11}$$

**Table 3 Detailed information of each video sequence (M, multiple objects; S, single object)**

| | No. of total frames | Frame rate (frame/s) | Frame size | No. of objects |
|---|---|---|---|---|
| Intelligent room | 300 | 25 | 320 × 240 | S |
| Pets 2006 | 3020 | 25 | 300 × 240 | M |
| Video 1 | 235 | 25 | 320 × 240 | S |
| Video 2 | 800 | 25 | 320 × 240 | M |

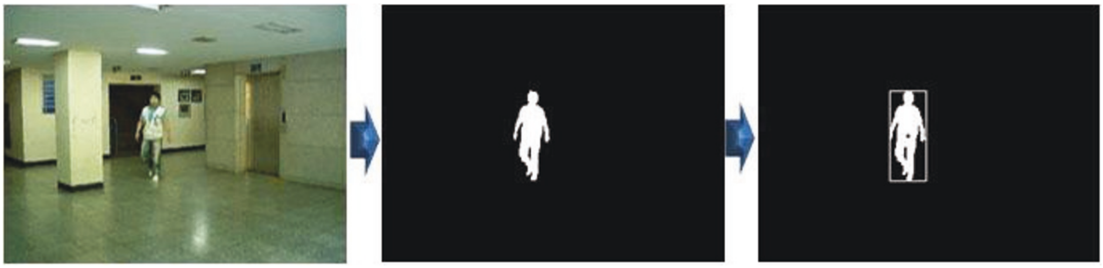
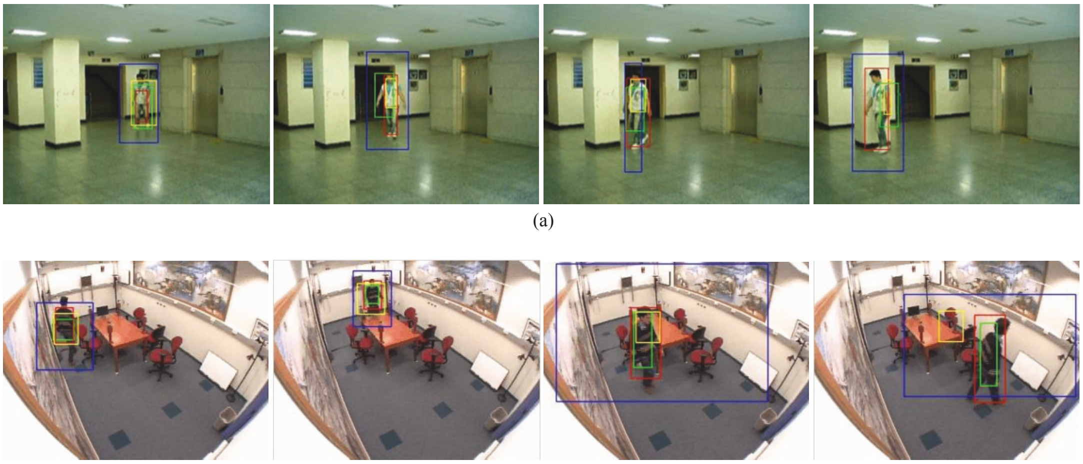


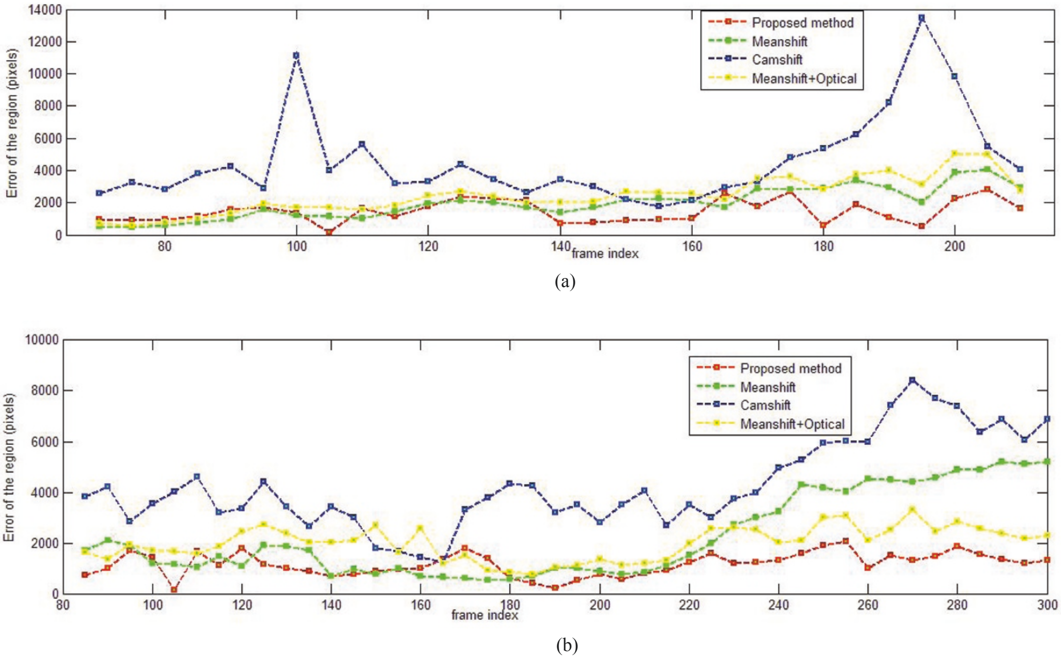
**Figure 10 An example of the processing of the ground-truth image**.



(a)

(b)

**Figure 11 A comparison of the resulting images of single-object tracking using each algorithm**. **(a)** Video 1 (frames 80, 120, 160 and 180). **(b)** Intelligent room (frames 97, 156, 267, and 293).



(a)

(b)

**Figure 12 Error comparison of the search window's region using each algorithm**. (a) Video 1. (b) Intelligent room.

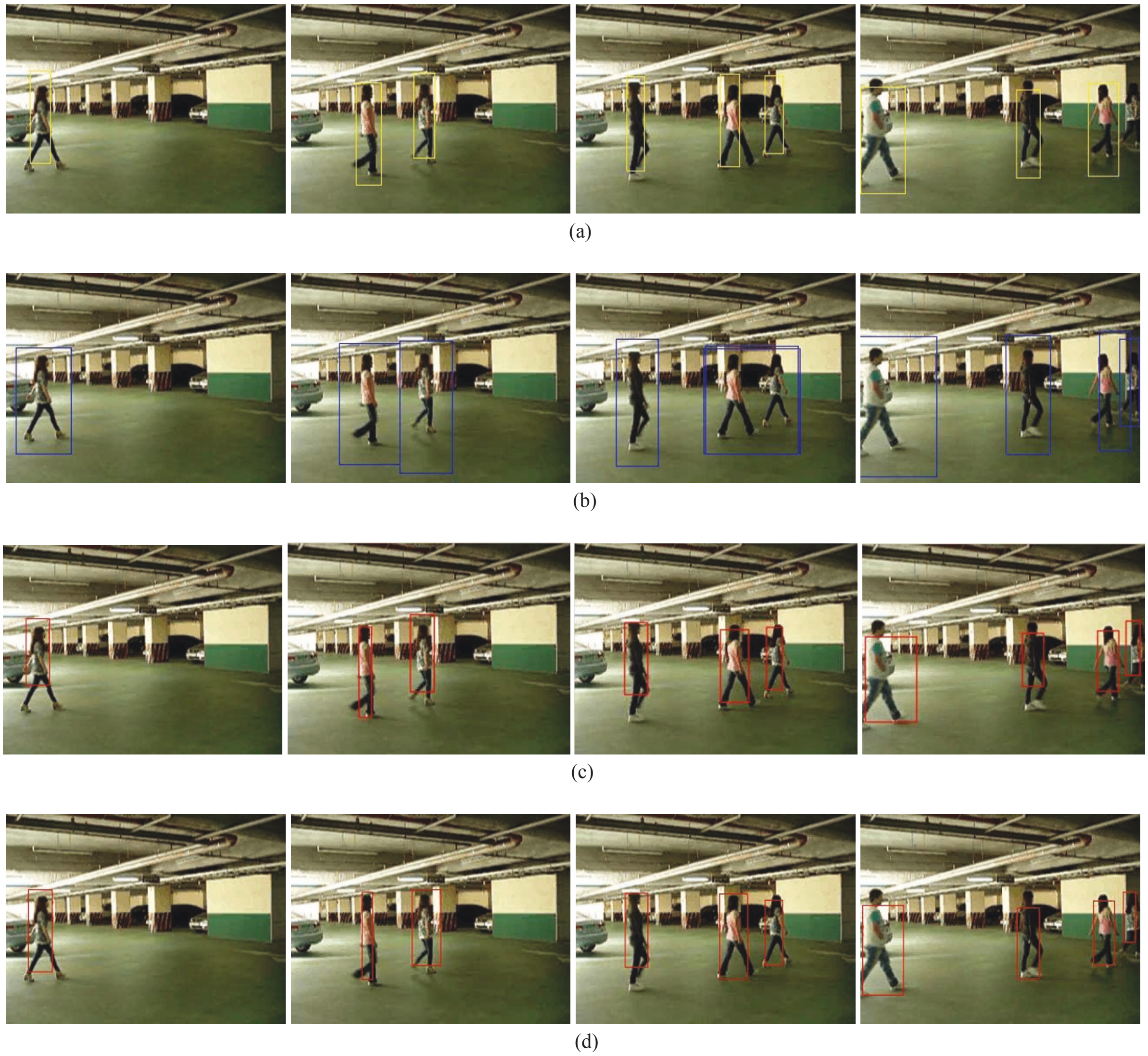


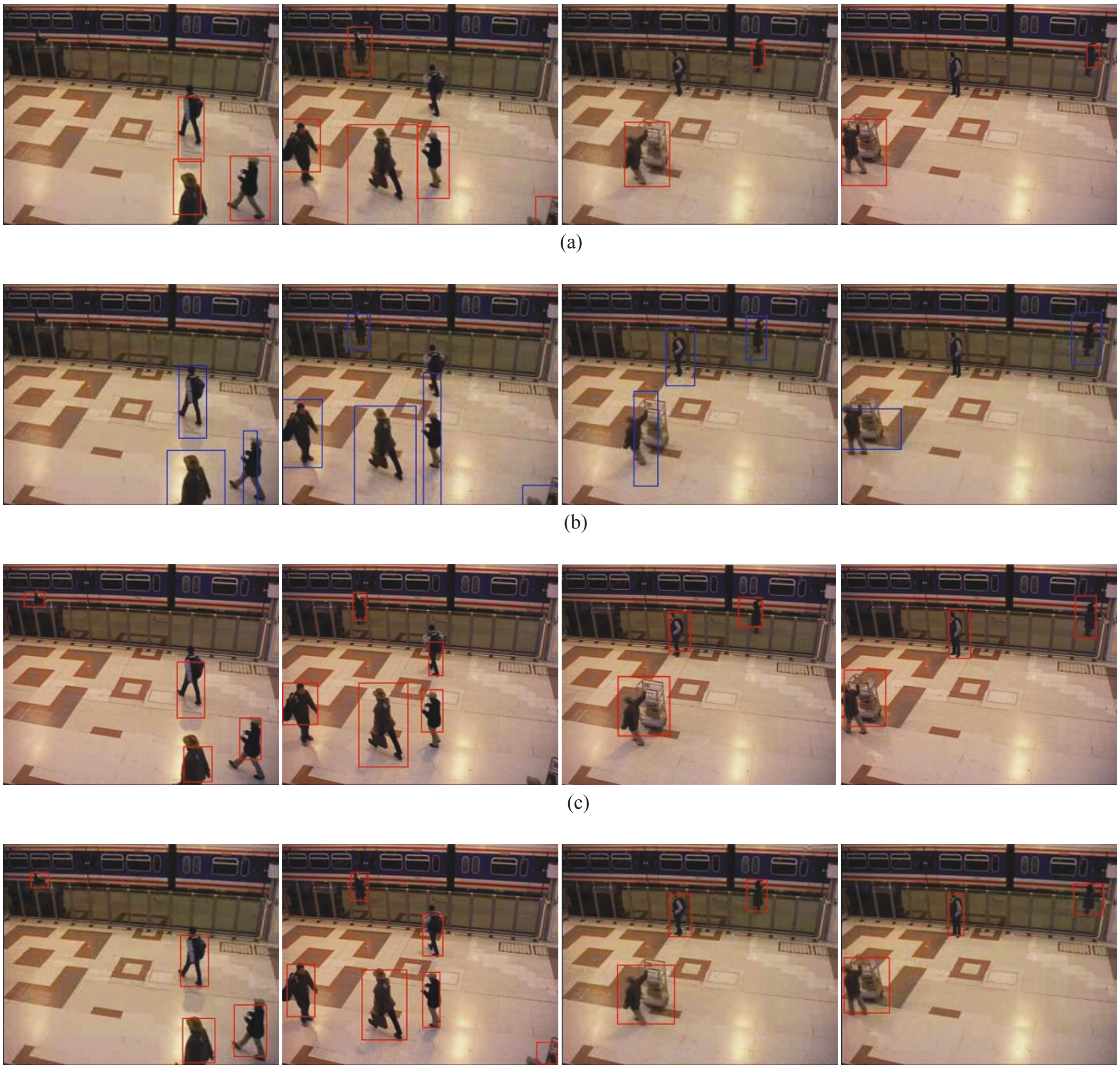
**Figure 13 A comparison of the resulting images of multi-object tracking in Video 2 using each algorithm (frames 540, 586, 619, and 661)**. (a) Meanshift. (b) Camshift. (c) Meanshift + optical. (d) Proposed method.

In general, the accuracy of the proposed method is higher than other algorithms. The average accuracy for the proposed method is 97.17%. Through experiments, the proposed method can increase the tracking accuracy at about 3.99%.

Table 5 indicates the comparison of the average processing time for each algorithm. In case of Intelligent room, it takes 0.03058 s to process one frame. Owing to extracting feature points of the SIFT algorithm, it is slower than only color-based algorithms. However, the proposed method is faster than the meanshift + optical, and it is sufficient time to track the object in real time. The average processing time for the proposed method is 0.03642s, and the average processing time for meanshift + optical is 0.03709s. Through experiments, the proposed method can reduce the processing time at about 0.00067 s per a frame.

## 5. Conclusions

A VSW algorithm based on color and feature points is proposed for accurate tracking of moving objects. When the size of object changes, and the tracked object has a similar color to the background color in an image, the color-based meanshift and Camshift algorithms easily miss the object. This article has demonstrated that the search window’s size in the meanshift algorithm can be changed using robust feature points to solve the problems encountered when tracking an object with a fixed search window size and a color similar to the background. In general, the accuracy of the proposed



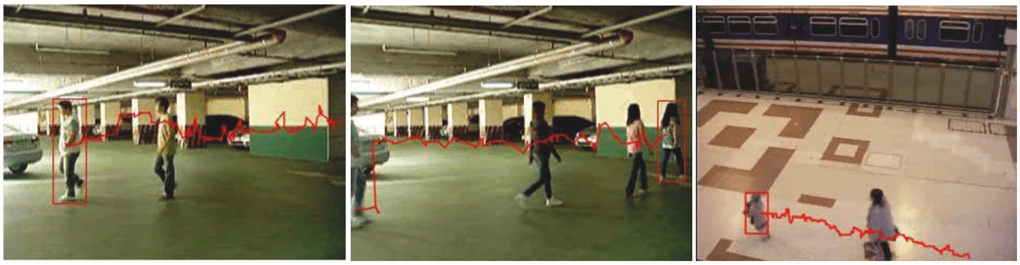
(a)

(b)

(c)

(d)

**Figure 14 A comparison of the resulting images of multi-object tracking in Pets 2006 using each algorithm (frames 1002, 1040, 1136, and 1175)**. (a) Meanshift. (b) Camshift. (c) Meanshift + optical. (d) Proposed method.



(a)

(b)

(c)

**Figure 15 The tracking result images of one-object among multi-object using the proposed method in Video 2 and Pets 2006**. (a) Video 2 (frame 199). (b) Video 2 (frame 655). (c) Pets 2006 (frame 460).

**Table 4 The comparison of the accuracy for each algorithm (%)**

| | The proposed method | Meanshift | Camshift | Meanshift + optical |
|---|---|---|---|---|
| Video 1 | 98.15 | 95.74 | 97.02 | 97.45 |
| Intelligent room | 98.67 | 96.33 | 97 | 98.33 |
| Video 2 | 97.50 | 92.50 | 90.38 | 96.38 |
| Pets 2006 | 94.37 | 87.75 | 76.92 | 92.35 |
| The average accuracy | 97.17 | 93.08 | 90.33 | 96.13 |

**Table 5 The comparison of the average processing time for each algorithm (s/frame)**

| | The proposed method | Meanshift | Camshift | Meanshift + optical |
|---|---|---|---|---|
| Video 1 | 0.03062 | 0.02937 | 0.02906 | 0.03199 |
| Intelligent room | 0.03058 | 0.03025 | 0.03347 | 0.03154 |
| Video 2 | 0.04062 | 0.03784 | 0.03886 | 0.04083 |
| Pets 2006 | 0.04387 | 0.04079 | 0.04102 | 0.04399 |

method is higher than other algorithms. The average accuracy for the proposed method is 97.17%. Through experiments, the proposed method can increase the tracking accuracy at about 3.99%. In this article, we improve the object tracking accuracy through the experiment of various videos. Therefore, combining multiple-features makes the object tracking more robust in tracking applications. According to the experimental results, the proposed method shows more precise performance than other algorithms.

### Abbreviations

Camshift: continuously adaptive meanshift; DOG: difference of Gaussian; GMM: Gaussian mixture model; ROI: region of interest; SIFT: scale invariant feature transform; VSW: variable search window.

### Acknowledgements

This study was supported by the Dong-A University research fund.

### Competing interests

The authors declare that they have no competing interests.

Received: 24 December 2010 Accepted: 17 September 2011
Published: 17 September 2011

### References

1. A Yilmaz, O Javed, M Shah, Object tracking: a survey. ACM J Comput Surv. **38**(4), 45 (2006)
2. J-S Hu, C-W Juan, J-J Wang, A spatial-color mean-shift object tracking algorithm with scale and orientation estimation. Pattern Recogn Lett. **29**(16), 2165–2173 (2008). doi:10.1016/j.patrec.2008.08.007
3. Y Cheng, Mean shift, mode seeking, and clustering. IEEE Trans Pattern Anal Mach Intell. **17**(8), 790–799 (1995). doi:10.1109/34.400568
4. D Comaniciu, V Ramesh, P Meer, Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell. **25**(5), 564–577 (2003). doi:10.1109/TPAMI.2003.1195991
5. M Isard, A Blake, Condensation-conditional density propagation for visual tracking. Int J Comput Vis. **29**(1), 5–28 (1998). doi:10.1023/A:1008078328650
6. C Papageorgiou, M Oren, T Poggio, A general framework for object detection, in *International Conference on Computer Vision*, pp. 555–562 (1998)
7. M Dixit, KS Venkatesh, Combining edge and color features for tracking partially occluded humans, in *ACCV 2009* 140–149 (2010). Part II, LNCS 5995
8. Y Akazawa, Y Okada, K Niijima, Robust tracking algorithm based on color and edge distribution for real-time video based motion capture systems, in *IAPR workshop on Machine Vision Applications*, pp. 60–63 (2002)
9. N Friedman, S Russell, Image segmentation in video sequences: a probabilistic approach, in *Proc 13th Conf Uncertainty in Artificial Intelligence (UAI)*, pp. 175–181 (August 1997)
10. P KaewTrakulPong, R Bowden, An improved adaptive background mixture model for realtime tracking with shadow detection, in *Proc 2nd European Workshop on Advanced Video Based Surveillance Systems (AVBS '01)*, pp. 1–5 (September 2001)
11. DS Lee, Effective Gaussian mixture learning for video background subtraction. IEEE Trans Pattern Anal Mach Intell. **27**(5), 827–832 (2005)
12. A Pnevmatikakis, L Polymenakos, Kalman tracking with target feedback on adaptive background learning, in *Machine Learning for Multimodal Interaction (MLMI)* 114–122 (2006). LNCS 4299
13. C Stauffer, W Eric, L Grimson, Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal Mach Intell. **22**(8), 747–757 (2000). doi:10.1109/34.868677
14. T Bouwmans, FE Baf, B Vachon, Background modeling using mixture of gaussians for foreground detection–a survey. Recent Patents Comput Sci. **1**(3), 219–237 (2008)
15. A Prati, I Mikic, MM Trivedi, R Cucchiara, Detecting moving shadows: algorithms and evaluation. IEEE Trans Pattern Anal Mach Intell. **25**(7), 918–923 (2003). doi:10.1109/TPAMI.2003.1206520
16. R Cucchiara, C Grana, M Piccardi, A Prati, S Sirotti, Improving shadow suppression in moving object detection with HSV color information, in *Proc IEEE Intelligent Transportation Systems Conf*, pp. 334–339 (August 2001)
17. GR Bradski, Computer vision face tracking for use in a perceptual user interface. Intel Technol J. **2**(2), 12–21 (1998)
18. T Kailath, The divergence and Bhattacharyya distance measures in signal selection. IEEE Trans Commun Technol. **15**(1), 52–60 (1967). doi:10.1109/TCOM.1967.1089532
19. D Comaniciu, V Ramesh, P Meer, Real-time tracking of non-rigid object using mean shift, in *Proc Conf Computer Vision and Pattern Recognition*. **2**, 142–149 (2000)
20. DG Lowe, Distinctive image features from scale-invariant keypoints. IJCV. **60**(2), 91–110 (2004)
21. H Zhou, Y Yuan, C Shi, Object tracking using SIFT features and mean shift. Comput Vis Image Understand. **113**(3), 345–352 (2009). doi:10.1016/j.cviu.2008.08.006
22. ATON Dataset, http://cvrr.ucsd.edu/aton/shadow/
23. Pets 2006, http://www.cvg.rdg.ac.uk/PETS2006/data.html
24. HY Lim, DS Kang, Object tracking based on MCS tracker with motion information, in *the 2nd International Conf Information Technology Convergence and Services (ITCS)*. **2**, 12 (2010)

doi:10.1186/1687-6180-2011-60
**Cite this article as:** Lim and Kang: **Object tracking system using a VSW algorithm based on color and point features.** *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:60.