Object Modeling with Color Arrangement for Region-Based Tracking

Dae-Hwan Kim, Seung-Won Jung, Suryanto, Seung-Jun Lee, Hyo-Kak Kim, and Sung-Jea Ko

In this paper, we propose a new color histogram model for object tracking. The proposed model incorporates the color arrangement of the target that encodes the relative spatial distribution of the colors inside the object. Using the color arrangement, we can determine which color bin is more reliable for tracking. Based on the proposed color histogram model, we derive a mean shift framework using a modified Bhattacharyya distance. In addition, we present a method of updating an object scale and a target model to cope with changes in the target appearance. Unlike conventional mean shift based methods, our algorithm produces satisfactory results even when the object being tracked shares similar colors with the background.

Keywords: Bhattacharyya coefficient, kernel-based tracking, mean shift, object tracking, spatiograms.

I. Introduction

Object tracking is a fundamental task in the field of computer vision, and its objective is to identify the location of the object of interest from frame to frame. Selecting the features suitable to represent the object is crucial to the performance of object tracking. The commonly used features are color, edge, optical flow, and texture [1]. Among several tracking schemes using various features, the mean shift algorithm using color histograms is widely used [2]-[4]. Comaniciu and others [2] presented the kernel-based tracking (KBT) algorithm that uses radially symmetric spatial kernels to represent objects. While KBT is computationally inexpensive and tracks the object successfully even when the target object is partially occluded, it tends to fail when the background bears a similar color to the target object.

To improve the KBT [2], Birchfield and Rangarajan [5], [6] proposed spatiograms that include spatial means and covariances for each histogram bin to incorporate the spatial information. In addition, Collins and others [7] treated tracking as a binary classification problem and used online feature selection to switch to the most discriminative color space from a set of different color spaces according to localization based on mean shift. This work was extended by Wang and Yagi [8] who represented the target using reliable features selected from color and shape-texture cues. In recent research, Ning and others [9] used the joint color-local binary pattern (LBP) histogram to represent a target object and then applied it to the mean shift framework. Wang and others [10] presented a fusion scheme to combine multiple spatially distributed fragments effectively.

In this paper, we propose an improved mean shift algorithm capable of tracking the object successfully even when its color

Manuscript received June 17, 2011; revised Sept. 21, 2011; accepted Oct. 12, 2011.

Dae-Hwan Kim (phone: +82 3290 3682, dhkim@dali.korea.ac.kr), Seung-Won Jung (jungsw@dali.korea.ac.kr), Suryanto (suryanto@dali.korea.ac.kr), Seung-Jun Lee (junlee@dali.korea.ac.kr), Hyo-Kak Kim (hkkim@dali.korea.ac.kr), and Sung-Jea Ko (sjko@korea.ac.kr) are with the Department of Electronics Engineering, Korea University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0111.0383

is shared by its surroundings. The proposed method utilizes the color arrangement of the target object for object modeling. The color arrangement encodes the information about the spatial relationship among the colors that compose the object. During tracking, the reliability of each color feature is calculated adaptively based on the consistency of the color arrangement information. Moreover, we present an object scale prediction scheme and a target model update scheme to handle changes in the target appearance.

This paper is organized as follows. In section II, we first present a brief overview of the spatiogram and its limitation. In section III, we introduce our new histogram model and derive the mean shift procedure using the modified similarity measure, followed by the scale prediction and target model update schemes. In section IV, we demonstrate through experiment the advantages of our method over existing algorithms, and we conclude our work in section V.

II. Spatiogram-Based Tracking

Before introducing the proposed method, it is worthwhile to review the spatiograms [5] and discuss some drawbacks of the tracking method based on spatiograms.

The spatiogram is an extended histogram that contains the mean and covariance of pixel locations in addition to the pixel count of a histogram bin. Let $\{\mathbf{x}_i\}_{i=1,\ldots,N_h}$ be a set of pixel coordinates inside a region R centered at location $\mathbf{y}$, and N_h is the number of pixels in R. The spatiogram of R is represented as

$$\mathbf{H}_R(\mathbf{y}) = < n_b(\mathbf{y}), \mu_b(\mathbf{y}), \Sigma_b(\mathbf{y}) >, \quad b = 1,\ldots,B, \tag{1}$$

where n_b is the kernel-weighted number of pixels whose quantized values fall into the b-th bin, μ_b and Σ_b are the mean vector and covariance matrices of the pixel coordinates corresponding to the b-th bin, respectively, and B is the total number of bins in the spatiogram. Mathematically,

$$n_b(\mathbf{y}) = C_h \sum_{i=1}^{N_h} k(\| (\mathbf{x}_i - \mathbf{y}) / h \|^2)\delta_{ib}, \tag{2}$$

$$\mu_b(\mathbf{y}) = \frac{1}{\sum_{i=1}^{N_h} \delta_{ib}} \sum_{i=1}^{N_h} (\mathbf{x}_i - \mathbf{y})\delta_{ib}, \tag{3}$$

$$\Sigma_b(\mathbf{y}) = \frac{1}{\sum_{i=1}^{N_h} \delta_{ib}} \sum_{i=1}^{N_h} (\mathbf{x}_i - \mu_b(\mathbf{y}))^T (\mathbf{x}_i - \mu_b(\mathbf{y}))\delta_{ib}, \tag{4}$$

where C_h is a normalization constant such that $\sum_{b=1}^{B} n_b(\mathbf{y}) = 1$ and a bandwidth h defines the scale of R. $k(\cdot)$ is a kernel function that weights the spatiogram entry based on the pixel distance from the object center, and δ_{ib} is 1 if the value of $\mathbf{x}_i$ falls into the b-th bin and 0 otherwise. Given the target model spatiogram, the tracking objective is to find the location at which the candidate model spatiogram is most similar to the target model.

The tracking method based on spatiograms has two limitations. First, it is sensitive to slight spatial changes of the features as analyzed in [11]. Second, the spatiogram-based mean shift algorithm tends to converge to the background region when the object being tracked bears a similar color to that of the background. The existence of background pixels having a similar color to that of the object can decrease the spatial mean difference between the target and candidate models since the spatial mean difference is calculated based on the center obtained in the previous frame.

III. Proposed Method

The proposed method utilizes the color arrangement of the target object to judge whether a similar color exists in the background region or not. In other words, the candidate region is regarded to be contaminated by the background when the spatial distribution of the colors inside the region is different from that of the target object. For instance, as shown in Fig. 1(a), the proposed method models the information that the upper body with a green color has a yellow head at the top of the body and red pants at the bottom of the body. During a tracking process, the proposed method infers that the yellow color region is spatially changed by the similar colored pixels in the background region when the color distribution near the yellow region differs from the target model as shown in Fig. 1(d). In other words, the proposed method computes the reliability that represents how the background region influences the target model by referring to the color arrangement and then utilizes it for tracking.

First, we introduce our new color histogram model for object tracking. Based on the proposed color histogram model, we describe a mean shift procedure using the modified Bhattacharyya distance. We assume that the color arrangement of the target is not severely changed between consecutive frames.

The proposed tracking method consists of the following steps. The target object obtained from the detection process is represented by the proposed histogram model in the initial frame. Then, in the subsequent frame, a candidate region whose location comes from the previous frame is modeled in the same way as target modeling. After constructing two models for the target and candidate regions, we localize the target object in the subsequent frame by finding the position that maximizes the similarity between the two models. Finally,

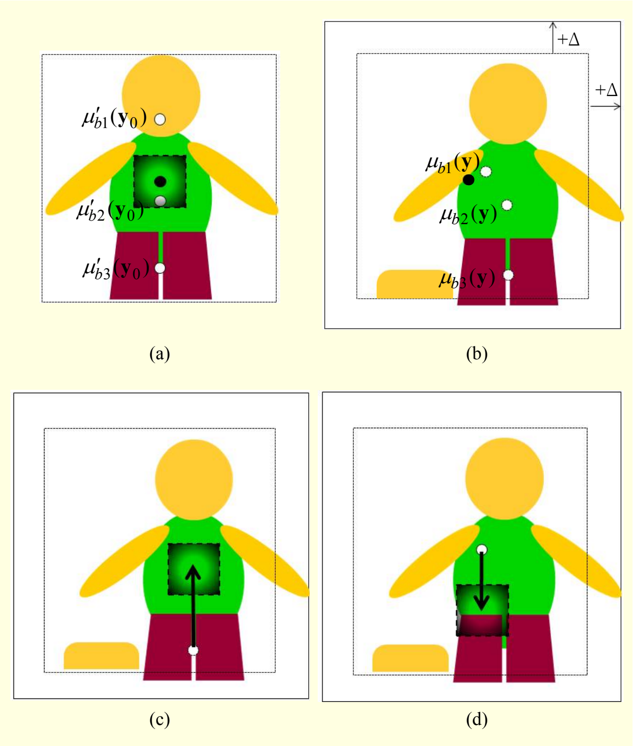


Fig. 1. (a) Target object and Gaussian mask in initial frame, (b) candidate region with expansion in subsequent frame, (c) location of Gaussian mask indexed by reference vector for $b3$, and (d) location of Gaussian mask indexed by reference vector for $b1$.

we update both the object scale and the target model to cope with the changes in the target appearance. These steps are explained in the following subsections.

1. Target Representation

Let T be the region of the target object to be tracked, centered at $\mathbf{y}_0$ in the initial frame. The proposed method models T as

$$\mathbf{P}'_T(\mathbf{y}_0) = < n'_b(\mathbf{y}_0), \mu'_b(\mathbf{y}_0), \sigma'_b(\mathbf{y}_0), \Phi'_b(\mathbf{y}_0) >, \qquad (5)$$

where $n'_b(\mathbf{y}_0)$ and $\mu'_b(\mathbf{y}_0)$ are as in (2) and (3), respectively, $\sigma'_b(\mathbf{y}_0)$ is the standard deviation vector of the pixels in the b-th bin, and $\Phi'_b(\mathbf{y}_0)$ is the Gaussian-weighted average of the pixels around $\mathbf{y}_0$. Mathematically,

$$\Phi_b(\mathbf{y}) = w(\mathbf{y}) * rgb(\mathbf{y}), \qquad (6)$$

where the notation $*$ denotes the convolution sum, w is the Gaussian filter, and $rgb(\mathbf{y})$ is the RGB vector at location $\mathbf{y}$. We calculate the convolution sum for each RGB channel independently. The Gaussian filter size K proportional to the object size is given by

$$K = \max\{\tau \cdot \min(s_x, s_y), 1\}, \qquad (7)$$

where s_x and s_y are the width and height of the target object, respectively, and τ=0.3 is a predefined ratio.

Similarly, the candidate region R, centered at location $\mathbf{y}$ in the subsequent frame, can be modeled as

$$\mathbf{P}_R(\mathbf{y}) = < n_b(\mathbf{y}), \mu_b(\mathbf{y}), \sigma_b(\mathbf{y}), \Phi_b(\mathbf{y} + \mu_b(\mathbf{y}) - \mu'_b(\mathbf{y}_0)) > . \qquad (8)$$

In our modeling, R is expanded by Δ pixels in the x and y directions from T, to include pixels that belong to the target object. Δ is a parameter depending on the speed of the moving object, and Δ=5 is sufficient for all the test videos in our experiments. In the candidate model, the position of the Gaussian mask used to calculate the weighted average $\Phi_b(\cdot)$ at each color bin is indexed by $\mu'_b(\mathbf{y}_0)$ from $\mu_b(\mathbf{y})$. Note that $\mu'_b(\mathbf{y}_0)$ comes from the target model, and hereafter we call the vector the *reference vector*. If the change of the spatial mean in the b-th bin is negligible, the weighted average $\Phi_b(\cdot)$ of the candidate model is similar to $\Phi'_b(\mathbf{y}_0)$ of the target model. In the presence of occlusion or in the case that the background bears a color similar to that of the object, the mean location of a color bin is shifted and the corresponding $\Phi_b(\cdot)$ will be different from $\Phi'_b(\mathbf{y}_0)$. Thus, by comparing $\Phi_b(\cdot)$ with $\Phi'_b(\mathbf{y}_0)$ at each color bin, we can determine the reliability of each bin during tracking.

As an illustration, consider an object to be tracked in the initial frame as shown in Fig. 1(a). Assume that the target object consists of three colors corresponding to bins $b1$ (yellow), $b2$ (green), and $b3$ (red). The mean locations of each color are represented by a white dot and labeled $\mu'_{b1}(\mathbf{y}_0)$, $\mu'_{b2}(\mathbf{y}_0)$, and $\mu'_{b3}(\mathbf{y}_0)$. Figure 1(a) also shows the Gaussian mask placed on the object's center, marked by a black dot. In Fig. 1(b), the border of the candidate region is delineated by a black outline. Note that due to the existence of the background whose color also falls into the bin $b1$, the mean location of the bin $b1$ is shifted. In Figs. 1(c) and (d), we show the Gaussian masks used for $\Phi_{b3}(\cdot)$ and $\Phi_{b1}(\cdot)$, respectively. Unlike $\Phi_{b3}(\cdot)$, $\Phi_{b1}(\cdot)$ is different from $\Phi'_b(\mathbf{y}_0)$, indicating that the color bin $b1$ is not reliable for tracking.

For simplicity, we hereafter use the simplified notation for the target model as $\mathbf{P}' = < n'_b, \mu'_b, \sigma'_b, \Phi'_b >$.

2. Target Localization

In this subsection, we derive the mean shift procedure to localize the target object in the subsequent frame. The basic mean shift method [2] is based on maximizing the Bhattacharyya coefficient given by

$$\vartheta(h(\mathbf{y}), h') = \sum_{b=1}^{B} \sqrt{n_b(\mathbf{y}) n'_b}\ , \qquad (9)$$

which measures the similarity between two histograms.

Our proposed target localization method is based on maximizing the following modified Bhattacharyya coefficient

$$\rho(\mathbf{y}) = \rho(\mathbf{P}(\mathbf{y}), \mathbf{P}') = \sum_{b=1}^{B} \Psi_b(\mathbf{y})\sqrt{n_b(\mathbf{y})n'_b}\ , \tag{10}$$

with

$$\Psi_b(\mathbf{y}) = \exp\left\{-\frac{|\Phi_b(\mathbf{y}) - \Phi'_b|}{\gamma_c}\right\}, \tag{11}$$

where γ_c is a constant value and $|\Phi_b(\mathbf{y}) - \Phi'_b|$ measures the distance between two color vectors [12].

A Taylor series expansion around $n(\mathbf{y}_0)$ yields a linear approximation to the coefficient in (10):

$$\rho(\mathbf{y}) \approx \rho(\mathbf{y}_0) + \Lambda_n(\mathbf{y};\mathbf{y}_0)\ , \tag{12}$$

where

$$\begin{aligned}\Lambda_n(\mathbf{y};\mathbf{y}_0) &= [n(\mathbf{y}) - n(\mathbf{y}_0)]^T \frac{\partial \rho}{\partial n}(\mathbf{y}_0) \\ &= \frac{1}{2}\sum_{b=1}^{B} \Psi_b(\mathbf{y}_0)\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}n_b(\mathbf{y}) - \frac{1}{2}\rho(\mathbf{y}_0).\end{aligned} \tag{13}$$

Substituting (2) to (12) results in

$$\rho(\mathbf{y}) \approx \frac{1}{2}\rho(\mathbf{y}_0) + \frac{C_h}{2}\sum_{i=1}^{N_h} v_i k\left(\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\|^2\right), \tag{14}$$

where

$$v_i = \sum_{b=1}^{B} \Psi_b(\mathbf{y}_0)\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}\delta_{ib}\ . \tag{15}$$

To maximize the modified Bhattacharyya coefficient (10), the sum in the right-hand side of (14) has to be maximized since the left-hand side is independent of **y**. It represents the kernel density estimation computed with the kernel profile $k(\cdot)$ at **y** with the data $\mathbf{x}_i$ weighted by v_i. The target center, which is equal to the mode of the density, can be found iteratively using the mean shift procedure [2].

A new location $\mathbf{y}_1$ is thus given by

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h} v_i g(\|\frac{\mathbf{y}_0 - \mathbf{x}_i}{h}\|^2)\mathbf{x}_i}{\sum_{i=1}^{N_h} v_i g(\|\frac{\mathbf{y}_0 - \mathbf{x}_i}{h}\|^2)}, \tag{16}$$

where $g(\mathbf{x}) = -\frac{dk(\mathbf{x})}{d\mathbf{x}}$ is the negative derivative of the kernel profile. As in [2], if the Epanechnikov kernel profile is used, the derivative of the kernel is constant and disappears:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h} v_i \mathbf{x}_i}{\sum_{i=1}^{N_h} v_i}\ . \tag{17}$$

The weight of each pixel in the candidate region, v_i in (15), consists of the conventional KBT weights and our proposed reliability factor in (11).

3. Scale Prediction

The scale of the target often changes in time and thus the bandwidth h in (2) has to be predicted accordingly. This is possible due to the scale invariance property of the Bhattacharyya distance. To adapt the bandwidth, we utilize the standard deviation vector of the pixel coordinates in reliable bins. To determine whether a bin is reliable or not, we check two conditions based on the information from the localization step.

The first is a mean shift weight obtained by (15), which indicates the probability of the pixel belonging to the target object. The i-th pixel quantized to the b-th bin is considered as a reliable candidate if

$$v_i \geq \varphi \max(v_j),\ \text{for}\ j = 1, \cdots, N_h\ , \tag{18}$$

where $\varphi = 0.4$ in our experiment. Therefore, we can pick out the pixels that approximately distinguish the target shape from the background.

We then apply the second condition to the bins consisting of the reliable candidate pixels. The color similarity $\Psi_b(\mathbf{y})$ in

```
Algorithm 1. (Scale prediction)
•Initialization:
    After target localization, create a current candidate model $\mathbf{P}_{\text{curr}}(\mathbf{y})$ on the region expanded by $\Delta$ pixels in the $x$ and $y$ directions from the target rectangular box,
        $\mathbf{P}_{\text{curr}}(\mathbf{y}) = < n_{b\_\text{curr}}(\mathbf{y}), \mu_{b\_\text{curr}}(\mathbf{y}), \sigma_{b\_\text{curr}}(\mathbf{y}), \Phi_{b\_\text{curr}}(\mathbf{y} + \mu_{b\_\text{curr}}(\mathbf{y}) - \mu'_b) >.$
•Reliability check:
    Step1: Check the reliability by (18) using $\mathbf{P}'$ and $\mathbf{P}_{\text{curr}}(\mathbf{y})$.
    Step2:
        -Calculate a resized reference vector,
            $\mu'_{b\_\text{resized}} = (\mu'_{b\_\text{resized}\_x}, \mu'_{b\_\text{resized}\_y}) = \left(\mu'_{b\_x} \times \frac{\sigma_{b\_\text{curr}\_x}(\mathbf{y})}{\sigma'_{b\_x}}, \mu'_{b\_y} \times \frac{\sigma_{b\_\text{curr}\_y}(\mathbf{y})}{\sigma'_{b\_y}}\right).$
        -Check the reliability by (19) using $\Phi'_b$ and $\Phi_{b\_\text{curr}}(\cdot)$ that is recalculated by
            $\Phi_{b\_\text{curr}}(\cdot) = \Phi_{b\_\text{curr}}(\mathbf{y} + \mu_{b\_\text{curr}}(\mathbf{y}) - \mu'_{b\_\text{resized}}).$
•Calculation of scale update ratio:
    For the reliable bins, accumulate the standard deviation ratio to update the object rectangle size,
        $\boldsymbol{\pi} = (\pi_x, \pi_y) = \frac{1}{|\text{Z}|}\left(\sum_{b \in \text{Z}} \frac{\sigma_{b\_\text{curr}\_x}(\mathbf{y})}{\sigma'_{b\_x}}, \sum_{b \in Z} \frac{\sigma_{b\_\text{curr}\_y}(\mathbf{y})}{\sigma'_{b\_y}}\right),$
    where $\boldsymbol{\pi}$ is a scale update ratio vector, Z is a set of reliable bins, and |Z| is the number of reliable bins.
```

```
Algorithm 2. (Target model update)
•Initialization:
    Create a new target model $\mathbf{P}'_{\text{new}}$ on the resized target region after
    scale prediction,
        $\mathbf{P}'_{\text{new}} = < n'_{b\_\text{new}}, \mu'_{b\_\text{new}}, \sigma'_{b\_\text{new}}, \Phi'_{b\_\text{new}} >$.
•Update of $n'_b$ :
        $n'_b = s n'_{b\_new} + (1-s) n'_b$ ,
    where $s$ is the Bhattacharyya coefficient between $n'_b$ and
    $n'_{b\_\text{new}}$ in (9).
•Update of $\mu'_b$ :
        $\mu'_b = (\pi_x \mu'_{b\_x}, \pi_y \mu'_{b\_y})$ .
•Update of $\Phi'_b$ :
    Recalculate $\Phi'_{b\_\text{new}}$ at the position of $\mathbf{y} + \mu'_{b\_\text{new}} - \mu'_b$ . If the
    color similarity between $\Phi'_b$ and $\Phi'_{b\_\text{new}}$ satisfies (19), $\Phi'_b$
    is updated as
        $\Phi'_b = \kappa \Phi'_{b\_\text{new}} + (1-\kappa)\Phi'_b$ ,
    where $\kappa = 0.7$ in (19).
```

(11) is used as the second criterion. The *b*-th bin is determined as reliable if

$$\Psi_b(\mathbf{y}) \ge \kappa,\ b \in \Upsilon, \tag{19}$$

where $\kappa = 0.7$ in our experiment and Υ is a set of bins that passes the previous step. We eliminate the bins affected by a background region even though they have high mean shift weights. If the number of reliable bins remaining after the two reliability checks is equal to zero, the size update is skipped. We found that a strict reliability check is required for the stable update of the object size and confirmed that our parameter setting of $\varphi = 0.4$ and $\kappa = 0.7$ produces the accurate scale prediction results. The proposed scale prediction scheme is described in Algorithm 1.

For applying a statistical approach to scale prediction, we estimate the object size using not only our prediction scheme but also a Kalman filtering method [13], [14]. We obtain smooth scale adaptation results by combining the Kalman filter approach. The object size updated by $\boldsymbol{\pi}$ is utilized as a 2×1 measurement vector in the Kalman filtering method, and a Kalman state model in our system is given by

$$\mathbf{s}_k = A\mathbf{s}_{k-1} + \boldsymbol{w}, \tag{20}$$

where the 2×1 state vector $\mathbf{s} \in \Re^2$ consists of two elements, object width and height. The 2×2 identity matrix *A* is a process model and the random variable $\boldsymbol{w}$ represents the process noise with normal distribution $p(\boldsymbol{w})$~$N(0, Q)$. In our experiments, we set

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}, P_0 = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}, \tag{21}$$

where *R* is the covariance matrix for the measurement noise and P_0 is the initial error covariance.

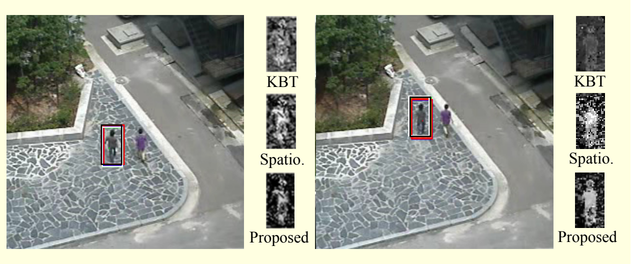


Fig. 2. Comparison of mean shift weight map obtained by KBT (white rectangle), spatiogram (red), and proposed method (black) on "noisy ground."

4. Target Model Update

It is necessary to update the target model since the appearance of a target tends to change during a tracking process. To update the target model adaptively, we consider three factors: the similarity between the histogram models in (9), the scale update ratio $\boldsymbol{\pi}$, and the color similarity $\Psi_b(\mathbf{y})$. We describe a method for updating elements of the proposed target model in Algorithm 2. The initial target size and initial σ'_b are saved and maintained to be used in the scale prediction scheme.

IV. Experiment Results

To compare the performance between the proposed algorithm and the existing algorithms, we challenge the problem of tracking the target object that bears similar colors to those in the background. The dataset contains seven sequences, including those of public datasets and our dataset: "PETS09_S2L1_V1" (PETS2009) and "PETS-09_S2L1_V6" (PETS2009) (of public dataset PETS2009 [15]), "OneStopMoveNoEnter2front" (of public dataset CAVIAR [16]), and "crowd," "summer beach," "white lane," and "noisy ground" (of our dataset). The initialization of tracking is set by manually marking the target object box in the first frame, as in [17]-[19], and the same initialization is used for all the trackers. For the object modeling, a 16×16×16 bin histogram is used for the RGB color image in the object region.

1. Comparison of Mean Shift Weight Maps

Figure 2 shows the tracking results of the KBT algorithm (white rectangle) [2], the spatiogram-based method (red) [5], and the proposed method (black). The mean shift weight of each pixel in the candidate region is shown right next to the input image. The pixels with higher weights are represented by

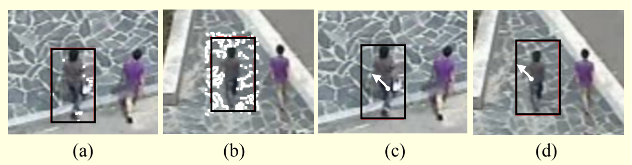


Fig. 3. (a), (b) Representation of pixels that have similar colors in background region; (c), (d) reference vector in 10th and 120th frames.

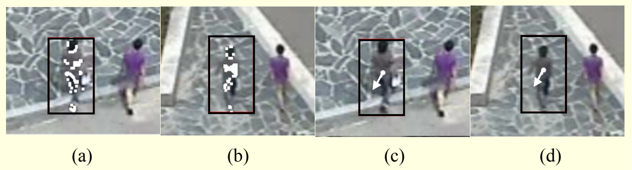


Fig. 4. (a), (b) Representation of pixels that have no similar color in background region; (c), (d) reference vector in 10th and 120th frames.

higher intensity, indicating higher pixel probability to be the target object. As shown in Fig. 2, the distinction between the target object and the background in the weight map for the KBT algorithm is not clear because of the pixels that bear similar colors to those in the background. Spatiogram-based tracking also produces high weighting pixels in the background. On the other hand, the proposed method clearly discriminates target and background pixels. The imprecise weight maps of the existing algorithms eventually lead to tracking failure.

Figures 3(a) and (b) contain white dots, which show the locations of the pixels that belong to the 8th bin for each RGB channel in the different frames, respectively. In addition, the reference vectors from the mean locations for the 8th bin are shown in Figs. 3(c) and (d). Figure 3(c) represents the mean location as a bold circular point and the reference vector as an arrow from the mean. The color on the region pointed to by the reference vector from the mean in Fig. 3(d) is very different from that of the target model in Fig. 3(c) because the background pixels are quantized to the same bin as the target.

Figure 4 shows the result to the contrary. The positions of the pixels quantized to the 4th bin for each RGB channel are shown in Figs. 4(a) and (b). The color similarity between the bin of the target in Fig. 4(c) and that of the candidate region in Fig. 4(d) is very strong. By utilizing the color similarity between Φ'_b and $\Phi_b(\cdot)$ for each bin between the two models, we can give different weights to each bin depending on not only the pixel counts but also the color arrangement of the target and candidate model.

2. Comparison of Tracking Performance

For quantitative performance evaluation, the tracking results are compared with those produced by the KBT [2], the spatiogram-based method [5], online feature selection [7], extended online feature selection [8], the LBP-based method [9], the particle-filter-based method [20], and the multiple-fragments-based method [10].

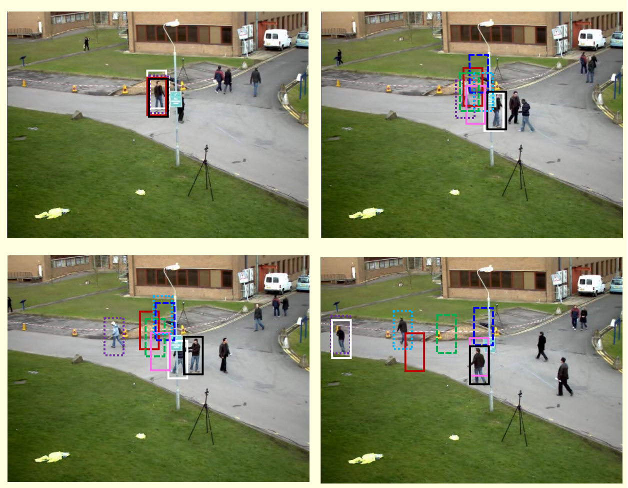


Fig. 5. Tracking results on “PETS09_S2L1_V1” in PETS2009 dataset with seven experiments using KBT [2] (white), spatiograms [5] (red), online feature selection [7] (blue dashed), extended feature selection [8] (violet dotted), LBP [9] (green dashed), MF [10] (cyan dotted), PF [20] (pink), and proposed method (black).

Figure 5 shows the tracking results on “PETS09_S2L1_V1” using seven existing methods and the proposed algorithm. As discussed in the subsection IV.1, the tracker [2] fails when a similarly colored object passes by the target object, and the tracker [5] misses the target in the case of partial occlusion due to its sensitivity to spatial differences. The tracker [7] leads to a tracking failure, as analyzed in [8], since the selected feature space dimension is limited to a low dimension. Note that the features in [7] are not totally independent and may be correlated, so it is relatively sparser than the proposed feature space dimension. The tracker [8] causes tracking failure since the number of features set (only seven features) is not enough to handle the distraction. In the case of [9], the number of samples tends to be decreased significantly in the target model since only the pixels that correspond to the nine uniform patterns are used. In our experiments, only about 6.6% of the pixels of the target region are used for target modeling. This reduced number can cause low discriminability and result in unstable tracking. Although the tracker [10] uses a multiple-fragments framework, the structure of the fragments in [10] is not updated during the whole process, it is not tolerant to position error, and it changes in object size. Furthermore, a one-dimensional histogram is computed for each R, G, and B channel so that discriminability is lower than our three-

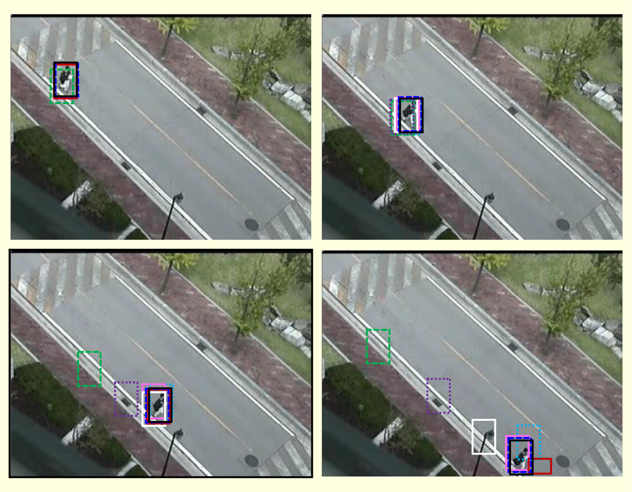


Fig. 6. Tracking results on "white lane," where the overlaid rectangles represent each algorithm in the same manner of Fig. 5.

dimensional model.

Figure 6 shows the experiment results for "white lane," where the pants of the person being tracked bear a similar color to that of the white lane. The trackers [2], [8], and [9] are distracted by the white lane and ultimately miss the target object.

In Table 1, the number of frames tracked successfully and the Euclidean distance are used to measure the performance of the tracking results for the proposed scheme and the seven existing trackers in six test videos. Tracking is considered to be a failure if the bounding box of the target object has no overlap with the ground truth box. The Euclidean distance is computed between the center of the tracking box and the ground truth box. Experiment results indicate that the proposed scheme is significantly better than five schemes ([2], [5], [8], [9], and [10]) and slightly better than [7] and [20] in terms of consistently having considerably less errors. Reference [20] shows slightly better performance than our method in videos "noisy ground" and "OneStopMove"; however, the background scene is required to model the background appearance. In other words, the method cannot be applied to videos captured from a moving camera, such as in the video "crowd."

3. Comparison of Computational Complexity

All of our experiments run on a standard PC (Intel i5 3.3 GHz with 4.00 GB RAM), and the resolution of test videos is 640×480. The comparison results of the computation speed in terms of average time required to track one video frame are shown in Table 2. The proposed method takes three to five times longer than the tracker [2], but the computation time is sufficiently acceptable for real-time applications. The trackers [5], [9], and [10] have similar complexity to the proposed method. Even the tracker [7] has a similar error rate compared to the proposed method for localization; its computation time takes almost 20 to 30 times (depending on the target object size) longer than that of the proposed scheme for all the test videos. The feature selection process using Gaussian filtering

Table 1. Quantitative performance evaluation measures of tracking simulations.

Frames tracked (frames)								
Sequence	KBT [2]	Spat. [5]	OFS [7]	EFS [8]	LBP [9]	MF [10]	PF [20]	Proposed
PETS09_S2L1_V1	64/127	56/127	57/127	57/127	74/127	57/127	62/127	**127/127**
PETS09_S2L1_V6	59/94	7/94	48/94	82/94	12/94	15/94	90/94	**95/95**
crowd	171/225	19/225	**225/225**	170/225	170/225	**225/225**	107/225	**225/225**
white lane	180/225	**225/225**	**225/225**	115/225	73/225	**225/225**	**225/225**	**225/225**
noisy ground	**301/301**	247/301	**301/301**	**301/301**	298/301	**301/301**	**301/301**	**301/301**
OneStopMoveNo.	255/613	253/613	573/613	612/613	335/613	285/613	**613/613**	**613/613**
Euclidean distance (pixels)								
Sequence	KBT [2]	Spat. [5]	OFS [7]	EFS [8]	LBP [9]	MF [10]	PF [20]	Proposed
PETS09_S2L1_V1	143.1	86.5	80.7	161.7	68.9	95.1	64.2	**5.2**
PETS09_S2L1_V6	41.9	160.2	66.1	47.9	135.0	127.9	27.0	**9.3**
crowd	63.8	310.1	**7.1**	65.9	65.7	10.9	138.3	7.7
white lane	29.0	10.5	4.5	85.2	188.6	11.1	9.2	**2.5**
noisy ground	4.5	22.1	2.5	5.3	4.8	**1.9**	**1.9**	3.0
OneStopMoveNo.	128.3	126.9	35.4	36.2	50.2	106.4	**15.4**	25.6

Table 2. Comparison of computation time required to track one video frame.

(ms)

Sequence	KBT [2]	Spat. [5]	OFS [7]	EFS [8]	LBP [9]	MF [10]	PF [20]	Proposed
PETS09_S2L1_V1	0.72	2.69	95.57	75.23	4.01	4.21	85.87	3.56
PETS09_S2L1_V6	2.57	8.84	255.3	82.16	7.54	9.99	85.99	8.96
crowd	5.23	17.65	395.71	95.66	11.57	32.96	88.85	15.33
white lane	1.17	4.22	102.38	77.60	4.62	8.60	88.97	5.15
noisy ground	0.25	0.96	31.55	20.72	1.38	1.26	23.71	1.12
OneStopMoveNo.	1.04	4.03	141.70	77.04	5.15	3.62	85.58	4.46
Average	1.83	6.40	170.37	71.40	5.71	10.11	76.50	6.43

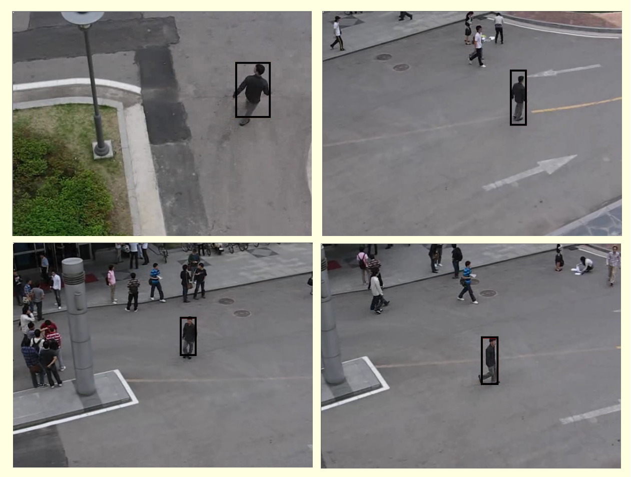


Fig. 7. Experiment results for the proposed scale prediction and target model update scheme on “crowd” captured by moving camera.

for 49 feature candidates [7] is mainly responsible for the heavy complexity in our analysis. The trackers [8] and [20] have relatively lower complexity compared to [7] but still take 10 to 20 times longer than the proposed method.

4. Comparison of Scale Prediction

The result of the proposed scale prediction scheme with the target model update is shown in Fig. 7. In Fig. 7, for the video “crowd” captured by a moving camera, the size of the target is successively tracked. Using our scale prediction scheme, we can control a zoom module in a real-time pan-tilt-zoom tracking system. To evaluate the robustness of the proposed scale prediction scheme, the proposed method is compared with two existing size prediction methods in KBT [2] and level sets [21], [22]. We use the Dice coefficient [23] to quantitatively compare the prediction results obtained by the three methods. Given ground truth object region Ω_1 and tracked region Ω_2, the Dice coefficient is defined as

$$d_v(\Omega_1,\Omega_2)=\frac{2Area(\Omega_1\cap\Omega_2)}{Area(\Omega_1)+Area(\Omega_2)}. \tag{22}$$

The Dice coefficient varies from 0 to 1 and measures the degree of similarity between the two different regions. Figure 8 shows the Dice coefficient versus the frame number of the proposed scheme and the two existing methods. As can be seen, the proposed predictor gives the best results in the six test sequences. The level sets method tends to have the weakness that the final contour expands to the similarly colored background region or shrinks when tracking fails. Likewise, the output of KBT approximately finds the scale of the target but accuracy is relatively lower compared to the proposed method.

V. Conclusion

We proposed a new color histogram model suitable for object tracking. The proposed model includes the color arrangement information for each bin of the histogram. Based on the proposed histogram model, we derived the mean shift procedure using the modified Bhattacharyya coefficient. In addition to target modeling and localization, we proposed the scale prediction and target model update methods to cope with changes in the target appearance.

Experiment results showed that the proposed tracking algorithm produces good results even when similar colors exist in the background region. In addition, the proposed scale predictor with the target model update scheme gives accurate and smooth prediction results for test sequences. The proposed tracking system is performed in real-time with an average computational time of 6.43 ms for objects of various sizes on 640×480 test videos.

As a future work, we will adopt a failure detection method to improve the robustness of the proposed algorithm. When the target tracking fails due to sudden changes of illumination or a long duration of heavy occlusions, we plan to identify the

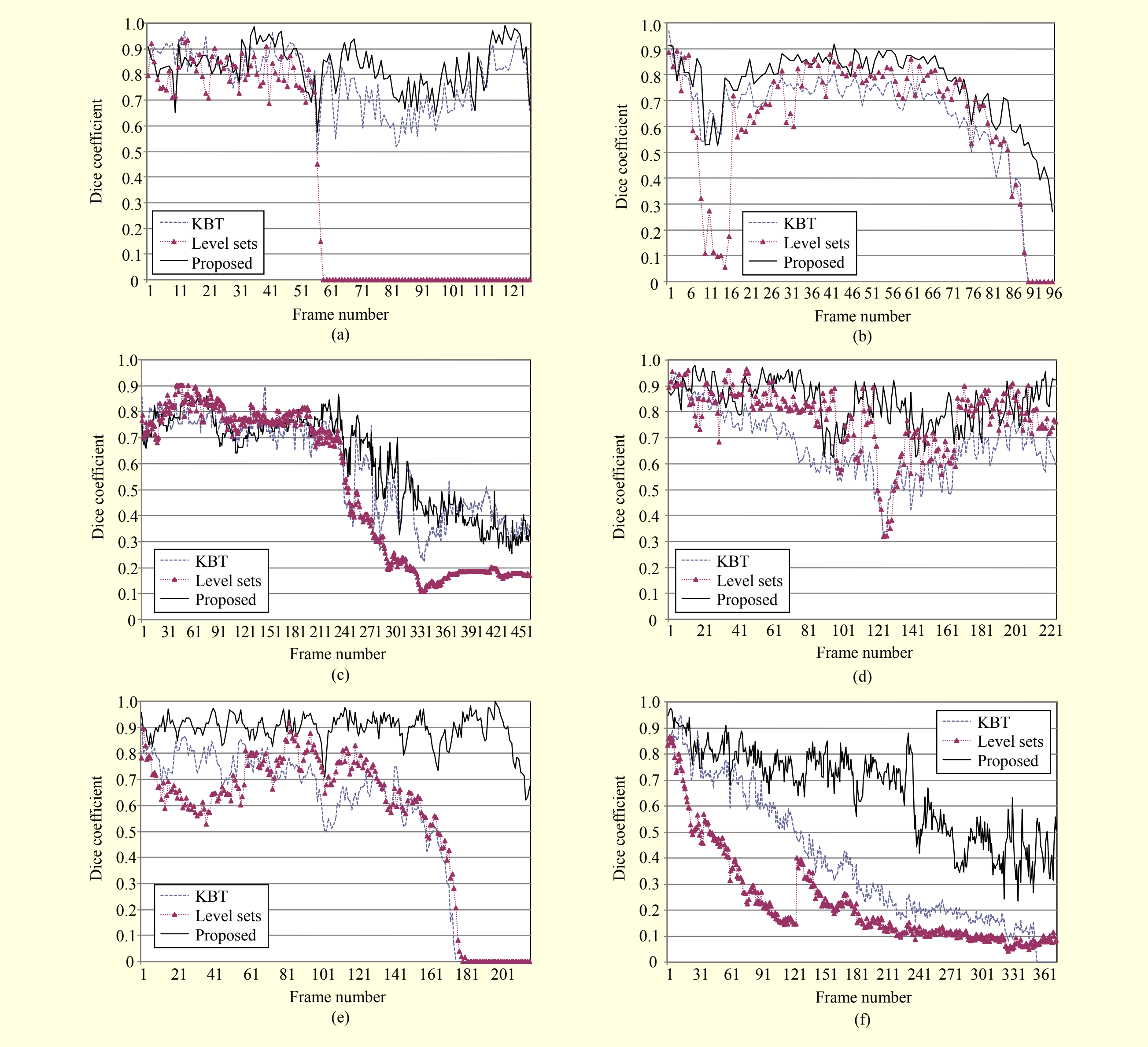


Fig. 8. Dice coefficients between tracked and ground truth regions for the six videos: (a) PETS09_S2L1_V1, (b) PETS09_S2L1_V6, (c) OneStopMoveNoEnter2front, (d) crowd, (e) white lane, and (f) summer beach.

tracking failure and recover the tracking process.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, Dec. 2006, pp. 1-45.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, May 2003, pp. 564-577.

[3] R.T. Collins, "Mean-Shift Blob Tracking through Scale Space," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 2, 2003, pp. 234-240.

[4] Z. Zivkovic and B. Krose, "An EM-Like Algorithm for Color-Histogram-Based Object Tracking," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 1, 2004, pp. 798-803.

[5] S.T. Birchfield and S. Rangarajan, "Spatiograms versus Histograms for Region-Based Tracking," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 2, 2005, pp. 1158-1163.

[6] S.T. Birchfield and S. Rangarajan, "Spatial Histograms for Region-Based Tracking," *ETRI J.*, vol. 29, no. 5, Oct. 2007, pp. 697-699.

[7] R.T. Collins, Y. Liu, and M. Leordeanu, "Online Selection of Discriminative Tracking Features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, Oct. 2005, pp. 1631-1643.

[8] J. Wang and Y. Yagi, "Integrating Color and Shape-Texture Features for Adaptive Real-Time Object Tracking," *IEEE Trans.*

Image Process., vol. 17, no. 2, Feb. 2008, pp. 235-240.

[9] J. Ning et al., "Robust Object Tracking Using Joint Color-Texture Histogram," *Int. J. Pattern Recog. Artificial Intell.*, vol. 23, no. 7, 2009, pp. 1245-1263.

[10] F. Wang, S. Yu, and J. Yang, "Robust and Efficient Fragments-Based Tracking Using Mean Shift," *Int. J. Electron. Commun.*, vol. 64, 2010, pp. 614-623.

[11] C. O'Conaire, N.E. O'Connor, and A.F. Smeaton, "An Improved Spatiogram Similarity Measure for Robust Object Localization," *IEEE Conf. Acoustics, Speech Signal Process.*, vol. 1, 2007, pp. 1069-1072.

[12] J. Davis et al., "Spatial-Depth Super Resolution for Range Images," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 1, 2007, pp. 1-8.

[13] G. Welch and G. Bishop "An Introduction to the Kalman Filter," Technical Report, UNC-CH Computer Science Technical Report 95041, 1995.

[14] R.G. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd ed., John Wiley & Sons, Inc, 1991.

[15] http://www.cvg.rdg.ac.uk/PETS2009/a.html

[16] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[17] S. Avidan, "Ensemble Tracking," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 2, 2005, pp. 494-501.

[18] A. Yilmaz, "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 10, Jun. 2007, pp. 1-6.

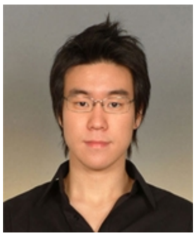
[19] H. Grabner and H. Bischof, "On-line Boosting and Vision," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 1, Jun. 2006, pp. 260-267.

[20] S.L. Zhao et al., "Using Cumulative Histogram Maps in an Adaptive Color-Based Particle Filter for Real-Time Object Tracking," *Advanced Material Research*, vol. 121, 2010, pp. 585-590.

[21] Y. Shi and W. Karl, "A Real-Time Algorithm for the Approximation of Level-Set-Based Curve Evolution," *IEEE Trans. Image Process.*, vol. 17, no. 5, May 2008, pp. 645-656.

[22] Y. Shi and W. Karl, "Real-Time Tracking Using Level Sets," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol. 2, Jun. 2005, pp. 34-41.

[23] K. Boesen et al., "Quantitative Comparisons of Four Brain Extraction Algorithms," *NeuroImage*, vol. 22, no. 3, Jul. 2004, pp. 1255-1261.



Dae-Hwan Kim received his BS in electronics engineering from Korea University, Seoul, Rep. of Korea, in 2008. He entered the Multimedia Image Processing Lab with the Department of Electrical Engineering of Korea University in March 2008. His research interests include video signal processing and visual surveillance.

Seung-Won Jung received his BS and PhD in electronics engineering from Korea University, Seoul, Rep. of Korea, in 2005 and 2011, respectively. He is currently a research professor in the Research Institute of Information and Communication Technology, Korea University. His research interests include image enhancement, image restoration, video compression, and computer vision.

Suryanto received his BS in electrical engineering from Bandung Institute of Technology, Indonesia, in 2005. He joined the CMCP laboratory in 2006 under an integrated MS/PhD program. His research interests include computer vision and multimedia signal processing.

Seung-Jun Lee received his BS in electronics engineering from Korea University, Seoul, Rep. of Korea, in 2010. He entered the Multimedia Image Processing Lab in the Department of Electrical Engineering of Korea University, in March 2010. His research interests include video signal processing and visual surveillance.

Hyo-Kak Kim received his BS in electronics engineering from Korea University. Seoul, Rep. of Korea, in 2005. He entered the Multimedia Image Processing Lab in the Department of Electrical Engineering at Korea University, in March 2005. His research interests include video signal processing and visual surveillance.

Sung-Jea Ko received his BS in electronics engineering from Korea University, Seoul, Rep. of Korea, in 1980, and his MS and PhD in electrical and computer engineering from the State University of New York at Buffalo, Buffalo, USA, in 1986 and 1988, respectively. From 1988 to 1992, he was an assistant professor of the Department of Electrical and Computer Engineering at the University of Michigan-Dearborn, Dearborn, USA. In 1992, he joined the Department of Electronical Engineering at Korea University where he is currently a professor. He has published over 150 international journal articles. He also holds over 50 patents for video signal processing and multimedia communications. He received the Best Paper Award from the *IEEE Asia Pacific Conference on Circuits and Systems* in 1996, the Hae-Dong Best Paper Award from the IEEK in 1997, the LG Research Award for the Outstanding Information Technology and Communication Researcher in 1999, and the Research Excellence Award from Korea University in 2004. He has served as a TPC member for the *IEEE Conference on Consumer Electronics (ICCE)* since 1997 and received a 10-year service award from the TPC of ICCE in 2008. He is also a TPC member and an international advisor of ICCE-Berlin and a member of the editorial board of the *IEEE Transactions on Consumer Electronics.*