Background Subtraction Using Background Sets With Image- and Color-Space Reduction

Hasup Lee, HyungSeok Kim, *Member, IEEE*, and Jee-In Kim, *Member, IEEE*

***Abstract*—Background subtraction is a basic step for a variety of multimedia applications such as live video, traffic monitoring, communication system, interactive learning space, etc. Many approaches have been proposed for this problem, but the need for lower cost approaches still exists. In this paper, a relatively low-cost background-subtraction method is proposed, using background sets with image- and color-space reduction. The background sets are used to detect objects from dynamic backgrounds, which contain waves, trees, and fountains. The image space is reduced to deal with jittered and unsteady frames, e.g., the input from handheld mobile devices. The color space is reduced to compensate for color noise, e.g., the scattered RGB values of a digital camera. To reduce the cost, a combination of color-space reduction and hash-table look-up operations are used. The results compared with other methods show the feasibility of our method; moreover, it can be useful in mobile or embedded environments.**

***Index Terms*—Background subtraction, change detection, foreground segmentation, video signal processing.**

I. Introduction

A. Background Subtraction

BACKGROUND subtraction, or foreground detection using background modeling, is a basic step for various multimedia applications such as live video [1], traffic monitoring [2], communication system [3], interactive learning space [4], etc. Many approaches have been proposed for background subtraction in more than three decades, but the need for lower-cost approaches still exists. Several works have been provided well-structured surveys [5]–[8]. The approaches are categorized into basic, statistical, fuzzy, neural network, wavelet background modeling, background clustering, and background estimation [5].

In basic modeling, a simple average [9], median [10], luminance [11], or histogram analysis over time [12] is used. These values are used to determine whether a pixel is foreground or background. Also an interactive image segmentation is presented [13].

Manuscript received February 2, 2016; revised July 4, 2016; accepted July 17, 2016. Date of publication July 27, 2016; date of current version September 15, 2016. This work was supported in part by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT, and Future Planning under Grant 2012M3C4A7032185, and in part by the Bio-Synergy Research Project under Grant 2013M3A9C4078140 of the Ministry of Science, ICT, and Future Planning through the National Research Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yonghong Tian. *(Corresponding author: HyungSeok Kim.)*

The authors are with the Department of Internet and Multimedia Engineering, Konkuk University, Seoul 143701, South Korea (e-mail: hasups@gmail.com; hyuskim@konkuk.ac.kr; jnkm@konkuk.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2595262

In statistical modeling, single Gaussian [14], Mixture of Gaussians [15], Kernel Density Estimation [16], Principal Component Analysis (PCA) [17], $(2D)^2$PCA [18], PCA-based RBF network [2], Independent Component Analysis [19], or Support Vector Machine [20] techniques are used to classify the pixels. Many approaches have been developed from the Mixture of Gaussians [6], in particular.

Fuzzy-based modeling techniques include using a fuzzy Running Average [21] or a Fuzzy Mixture of Gaussians [22]. In neural-network modeling, backgrounds are represented using General Regression Neural Networks [23] or Self-Organizing Neural Networks [24]. In background clustering, pixels in the frame are clustered, and new pixels are classified as foreground or background using these clusters. Pixels are clustered using the K-means algorithm [25] or Codebook [26]. In background estimation, backgrounds are estimated using filters like the Kalman Filter [27].

Recently there have been challenges for specific background problems. Several approaches were proposed for solving the dynamic background problem [34]–[36]. To deal with variable bit rates over real world networks with limited bandwidth, several method was presented [37], [38]. There are background methods proposed to cope with turbulence [39], sudden luminance changes [40], and bootstrapping [41]. Other method improves efficiency by eliminating the examination of non ROI region [42]. There are recent improvements in method using matrix decomposition like robust PCA [43], [44] and Manhattan nonnegative matrix factorization [45]. As the higher order generalization of matrix decomposition, tensor based approaches have been proposed recently [46]–[50].

To solve the problem of the dynamic background, block-based codebook methods were proposed [34], [35]. A hierarchical codebook was employed [34] but the suitable size of block is not fixed for each video. To solve this, a multilayer codebook-based background model is proposed [35] with three adaptive block-based layers for coarse detection and a pixel-layer to classify pixels as foreground, shadows, and highlights. Another motion detection approach for the dynamic background based on radial basis function artificial neural networks is developed [36]. It generates a flexible probabilistic background model through an unsupervised learning then detects moving objects by only processing blocks that are highly likely to contain them. Our method also challenges the dynamic background but our goal is fast and lightweight algorithm for mobile devices or embedded system environments with reasonable results.

To overcome the limitations presented in video streams of variable bit rates over real world networks with limited bandwidth, an automated motion detection for traffic surveillance systems was proposed [37], [38]. It employs a Fisher's

linear discriminant-based radial basis function network [37]. Or a motion detection approach that is based on the cerebellar-model-articulation-controller through artificial neural networks is presented [38]. The results they showed are superior to others in video streams of both low and high bit rate of traffic surveillance systems.

For the turbulence videos, a simultaneous turbulence mitigation and moving object detection approach was proposed [39]. It produces an object confidence map using a turbulence model which utilizes both the intensity and the motion cues. Then it decomposes the sequence into the background, the turbulence, and the moving objects by solving three-term rank optimization.

When sudden luminance changes occur, background model re-initialization technique for motion detection was developed [40]. It contains a morphology-based temporal difference computation and a temporal difference-based entropy estimation step to detect sudden luminance changes.

In background subtraction, a bootstrapping process is necessary when there are no initial video data without moving objects. A hybrid method with detection of background and foreground candidates [41] are proposed to solve this. It quickly initializes the background model using background candidate detection method and eliminates unnecessary regions containing only background pixels using foreground candidate detection method.

A motion detection for the automatic video surveillance system was proposed [42]. It produces high-quality background model first, then eliminates the unnecessary examination of non ROI region. During background subtraction it performs rapid matching followed by accurate matching to produce optimum background pixels. The background model contains neither noise pixels nor artificial ghost trails.

In robust PCA approaches [43], [44], a large data matrix is decomposed into the low-rank matrix and the sparse matrix. Background subtraction is a natural application for this model. Video frames are stacked as columns of the source matrix, then the low-rank component corresponds to the background and the sparse component captures the foreground objects. A weighted combination of the nuclear norm and of the l1 norm is minimized to the decomposition of a low-rank matrix and an overall sparse matrix in Principal Component Pursuit [43]. This is one of robust PCA approaches and can recover the principal components of a data matrix even though a positive fraction of its entries are arbitrarily corrupted. Thus, objects can be detected in a cluttered background. But this cannot handle entire columns where every entry is corrupted. Outlier Pursuit recovers the optimal low-dimensional subspace and identifies the corrupted points (outliers) exactly [44]. It also uses convex optimization for recovering low-dimensional structure.

Another matrix decomposition framework applied to background subtraction is Manhattan non-negative matrix factorization (MahNMF) [45]. Non-negative matrix factorization (NMF) approximates a non-negative data matrix by the product of two non-negative low-rank factor matrices. MahNMF robustly estimates the low rank part and the sparse part of a non-negative matrix. It minimizes the Manhattan distance between a data matrix and its low-rank approximation for modeling the heavy-tailed Laplacian noise. It is robust to outliers including both occlusions and several types of noises.

Low-rank tensor recovery or decomposition is the higher order generalization of low-rank matrix decomposition like robust PCA. This decompose data into the low-rank and the sparse tensor. This is especially suitable for analyzing multi-linear data with gross corruptions, outliers and missing values [50].

Tensor based approaches need a priori tensor rank estimates and a low rank approximation computation of tensor. A framework for finding low rank approximation of a given tensor was proposed [46] and it used the adaptive Lasso with coefficient weights for sparse computation in tensor rank detection. The alternating direction method of multipliers is employed for minimization problem of low-rank tensor recovery [50]. Previous tensor based methods were sensitive to outliers and needed huge memory usage and computational issues because of the batch optimization methods. To tackle this, a stochastic optimization on tensor for robust low-rank and sparse tensor recovery was proposed [47]. In tensor based method, the memory consumption is increased when data size is very large. To solve this, an incremental tensor subspace learning was proposed [48] and it uses only a small part of the entire data and updates the low-rank model incrementally. When an observed data is formed by the superposition of the two tensors, a truncated and smoothed Schatten-p function based method is effective [49]. Authors derived this function using the augmented Lagrangian multiplier optimization algorithm.

These approaches are appropriate in different cases. However, for mobile-device or embedded-system conditions, the required processing cost is often too high to be feasible. Basic modeling methods have a low-enough time complexity but the results are not satisfactory. Other methods have acceptable performance, but the time costs are higher. Our approach began with basic background-modeling methods for low complexity, but has acceptable performance in a mobile environment.

B. Related Works

There are previous methods developed for mobile-device or embedded systems. They are estimation or tracking based approaches for moving camera [51], [53], [57], [59], region based approaches for embedded camera [52], [54], [55], [58] and energy saving analysis of conventional methods [56].

Estimation or tracking based approaches produce superior results for moving camera but they have more sophisticated algorithms than other background subtraction methods. Ours cannot deal with freely moving camera input but can applied as basic steps for application for this input.

In online moving camera background subtraction [51], pixel-based models for foreground and background regions are learned and long term trajectories with a Bayesian filtering framework are used to estimate motion and appearance models of the scene. This method is applied to freely moving camera input but it is not lightweight for mobile-device. Our algorithm is straightforward and has less complexity in time.

A real-time moving object detection by nonparametric modeling is presented [53]. Nonparametric background and

foreground models are obtained by a combination of chromaticity and gradients. A Bayesian classifier with Priors is applied to this models and a tracking strategy over previously detected foreground regions based on a particle filter is used. The title of this method contains 'lightweight' but this term is not quantitative. The experiment of this method is implemented under desktop PC environment and the processing rate is 3–11 frame per second [53]. The comparison of our method with others under desktop PC environment is presented in result section.

Another method for freely moving camera is developed [59]. It discriminates image motion induced by camera's motion and influenced by moving objects. RANSAC is used to estimate the background trajectory basis and the inliers are classified as background and the outliers are foreground. This method is applied for the video from freely moving cameras but difficult for online applications. Our approach focus on input from camera with limited motion like jittered, pan-tilt-zoom (PTZ) and dynamic background like fall, waving lake, whiffling leaves, etc.

Robust background subtraction using approximate matching of cell-phone videos is proposed [57]. This method use two videos, one with and the other without foreground objects. As a result, a video with only the foreground objects shown is generated. The camera paths are not identical and some amount of partial path is overlapped. It is efficient matching of foreground and background videos tracing significantly different camera trajectories. The difference of this method and ours is that this is for video from mobile device but ours runs on it.

The idea of region based approaches for embedded system is that the energy can be saved by allocating less resources for non ROI. They focus on the energy savings because it is critical for embedded systems. The quality of results is less important.

A chromatic clustering-based background model is proposed for embedded system [52]. The pixels belonging to the background colors are classified as 'stable' and others are 'adaptable'. Physically large and stable objects with similarly large clusters of chromaticity is 'stable'. A palette-based background model matching is used for 'stable' regions and Multimodal Mean [58] for 'adaptable' regions. Palette colors are chosen by histogram analysis. Multimodal Mean used a running sum of observed RGB values, an observation count, and two recency counts for background model matching. Multimodal Mean is extended for dealing with the movement of an existing stationary object in background [58]. This method provides time speedup and storage savings over MMean and is implemented in embedded vision system. But this is for fixed camera and our method can be applied in 'adaptable' region because of its simplicity.

Other method take advantage of savings in processing time by sending the microprocessor to idle state at the end of processing a frame [54]. The sending rate is adaptively changed based on the amount of activity and speeds of tracked objects. Salient and non-salient motion based on the history of a pixel's location is used for saving memory access [55]. A pixel with difference more than threshold in the region of salient motion is classified as foreground. But in the region of non-salient motion, it is checked by the neighborhood information. This is also focus on energy savings and implemented on embedded cameras. This method is also for fixed camera and low-cost system but ours can be applied for limited movement camera. This method is not designed for spatial and color noise and have too many constant values in algorithm.

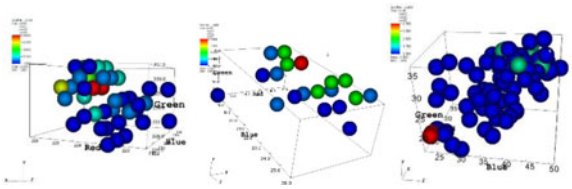


Fig. 1. Plotting a point in RGB color space over 100 background frames.

Backgroundsubtraction algorithms for Android devices deployed in wireless multimedia sensor networks are evaluated [56] in the aspect of the energy consumptions. It assume harsh environments such as earthquakes, so distorted frame and packet loss are simulated in experiments. The algorithms are not designed for mobile environment and common methods of BS algorithms in the literature. We compared our method with common methods in the literature and they are chosen by dataset provider [31]. For comparison, changedetection.net implement and provide the results of them. We applied the same conditions, processor, and post processing parameters they applied.

C. Background Set

The idea of background sets comes from the oscillation of pixels in background images. Each pixel of a background frame can have various values because of digital camera noise, the flicker of lights, etc. [62]. If the background contains waves, trees, fountains, fans, etc., the pixel values are also dynamic.

Background sets contain all the values for each pixel on the training frames for foreground detection. If a pixel value on the current frame is an element of the background set for its position, it can be categorized as background.

The input frames of mobile devices are frequently jittered and unsteady because they are handheld devices. Our approach deals with this condition using image-space reduction. A block of the image space is reduced to one point and the jittered movement within the block can be tolerated.

As shown in the examples of plotting RGB values of the same point over 100 still frames in Fig. 1, the pixels in the background of camera images are scattered. The color spectrum indicates the number of appearances of each RGB value. Red has the highest number (8, 9, and 4 times each) and blue has the lowest (only once). These scattered values are approximated using color-space reduction cubes.

The color space is reduced for fewer comparisons of the current points with a background, and the speed of our method can be improved using hash-table-based look-up operations. In addition, less memory can be used than with no reduction, but there is a trade-off between speed and accuracy. Parameter selection is described in Section III.

There have been cluster and codebook based background subtraction methods similar to our approaches. Here is the highlight of our model against other previous approaches.

There is a background modeling that represents each pixel in the frame by a group of clusters [60]. Each cluster consists of a weight and an average pixel value called the centroid. It defines a matching cluster as one which has a Manhattan distance between its centroid and the incoming pixel below a threshold. The boundary formed using Manhattan distance is an octahedron.

Other clustering based background subtraction classifies the pixel intensity based on A Modified Basic Sequential Clustering (MBSC) [61]. It runs clustering procedure then merging if two of the formed clusters are very closely located. After that, the appearance frequency of all clusters are calculated. Using this frequency, a single or multi-images are chosen as the background images. This model forms the spherical boundaries.

In codebook approach [26], a codebook consisting of one or more codewords is built for each pixel. Samples at each pixel are clustered into the set of codewords. The codewords consists of a color distortion metric and a brightness bound. The decision boundary formed using the codeword is a cylinder that its height defines the intensity boundary and radius does the color distortion.

In our approach, the decision boundary is the set of cubes. The decision boundary of Manhattan distance clustering [60] consists of an octahedron, MBSC [61] does spheres and codebook [26] does cylinders. Cylindrical boundary considering the intensity changes are not effective because there are so many noises under real environment like Fig. 1. All methods including ours can deal with multiple backgrounds. Clusters [60], [61], codebook [26] and our method keep multiple backgrounds for foreground detection.

Clustering procedures are needed for previous cluster based approaches [26], [60], [61] but our method do not. Only space reduction operation—subdivision or bitwise shift—is needed and hash table operation for boundary decision. The drawback is that our method uses more memory than others. Details about time and space complexity are described in experimental section.

This paper is organized as follows. The proposed background model is explained with mathematical expressions and images in Section II. The basic model is explained, and then extended using reductions in the image and color spaces. Our implementation is described and some experimental results are shown in Section III. Then, we discuss and conclude our approach in Sections IV and V, respectively.

II. Background Model

In this section, the background modeling is described using mathematical notations and abstract images. The basic background model is defined first, and then the image- and color-space reductions are applied. The notations of this paper are show in Table I.

TABLE I
NOTATIONS

Notation	Type	Meaning
X, Y	Constant number	Width, height of image frame
U, V	Constant number	Width, height of reduced image space
D_x , D_y	Constant number	Reduction ratio of the *X, Y* direction of image space
R, G, B	Constant number	RGB dimension of color space
R′, G′, B′	Constant number	RGB dimension of reduced color space
D_r , D_g , D_b	Constant number	Reduction ratio of the RGB direction of color space
N	Constant number	Number of training frame
e	Vector	Some pixel value
p[*x, y*]	Vector	Pixel value at (x,y) on current image frame
$T_i\,[x, y]$	Vector	Pixel value at (x,y) on *i*-th training frame
$B\,[x, y]$	Set of vector	Background set containing all pixel values at (x,y) in all training frames
B′ [*u, v*]	Set of vector	Background set at (u,v) in reduced image space; containing all pixel vales in the rectangle of (uD_x … $(u+1)\,D_x$) ∗ (vD_y … $(v+1)\,D_y$) in all training frames
B″[*u, v*]	Set of vector	Background set at (u,v) in reduced image and color space; same as *B′* [*u, v*] but all elements are divided by D_r , D_g , D_b respectively.

A. Basic Background Model

For a training frame *T*, a pixel value at [*x, y*] on the *i-th* frame of *T* is as follows. The number of training frames is *N,* and the width and height of the frames are *X* and *Y*, respectively

$$T_i\,[x, y] \quad \begin{pmatrix} 1 \le i \le N \\ 1 \le x \le X \\ 1 \le y \le Y \end{pmatrix}. \tag{1}$$

For the background sets *B*, the background set on [*x, y*] is defined in (2). *B*[*x, y*] is a background set containing all the pixel values at [*x, y*] in all training frames

$$B\ [x, y] = \mathrm{U}_{i=1}^{N} \{T_i\,[x, y]\}. \tag{2}$$

Pixel values, like RGB (vector) or intensity (scalar), can be elements of the background sets. The pixel values of the same position on the training frames are collected into one corresponding background set using the set-union operation. The image frames and the background-set space are the same dimension as the basic background model.

Using the basic background model, a background pixel is detected as follows. The pixel *p*[*x, y*] of the current image frame is background, if there exists a pixel value *e* that is an element of *B*[*x, y*], and the difference between *p*[*x, y*] and *e* is within a threshold

$$\exists e \in B\,[x, y]$$
$$|p\,[x, y] - e| < Threshold \tag{3}$$

The background will be overestimated using the basic background model, and it cannot deal with jittered and unsteady input. Further, the pixel values of the vibrating part are redundantly stored.

B. Image-Space Reduction

An image-space reduction is applied to the basic background model to compensate for spatial noise. Spatial noise comes from jittered and unsteady input, e.g., the images from handheld mobile devices. A dynamic background, with rustling trees or flowing water, can also cause spatial noise.

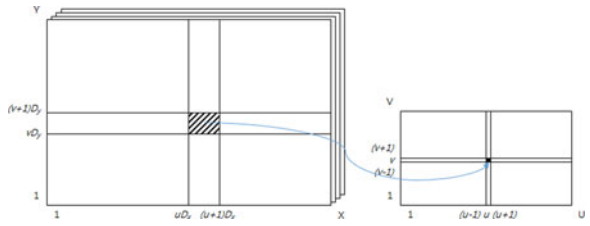


Fig. 2. Image-space reduction.

Image-space reduction can reduce the spatial noise. The pixels within the striped block of the training images in Fig. 2 are mapped into one background set. Each pixel within the block of the current image is classified using the same background set. Thus, the spatial noise within the block of the image space can be tolerated.

The dimension of the image space is reduced from $X * Y$ to $U * V$. Let the ratio of the *X* direction be D_x and the *Y* direction be D_y, as follows:

$$X \times Y \rightarrow U \times V$$
$$D_x = \frac{X}{U} \qquad D_y = \frac{Y}{V}. \tag{4}$$

The background sets B' are defined on the reduced image space, as follows. $B'[u, v]$ is the background set at [*u, v*] and defined in (5)

$$B'\ [u,v]\ =\ \mathrm{U}_{i=1}^{N}\mathrm{U}_{j=uD_x}^{(u+1)D_x}\mathrm{U}_{k=vD_y}^{(v+1)D_y}\ \{T_i\,[j,k]\} \times \begin{pmatrix} 1 \le u \le U \\ 1 \le v \le V \end{pmatrix}. \tag{5}$$

All pixel values in the rectangle of $(uD_x \ldots (u+1)D_x) * (vD_y \ldots (v+1)D_y)$ from all training frames are merged into the background set $B'(u, v)$, as in Fig. 2. The dimension of the reduced image space (background set space) is $U * V$.

A background test using the background sets B' is similar to the previous method. A pixel *p*[*x, y*] of the current image frame is in the background if there exists a pixel value *e* that is an element of $B'[x/D_x, y/D_y]$, and the difference between *p*[*x, y*] and *e* is within a threshold

$$\exists e \in B'\left[\frac{x}{D_x}, \frac{y}{D_y}\right]$$
$$|p\,[x,y] - e| < Threshold. \tag{6}$$

C. Color-Space Reduction

Color-space reduction is applied to the background sets to compensate for color noise. It is caused by the digital camera itself, the flicker of lights, etc. The pixel values of a fixed position on background image frames are not constant and form a non-uniform cluster, as in Fig. 1. These clusters are approximated with piles of striped cubes, as in Fig. 3.

The RGB values within the striped cubes are mapped onto one value; thus, the noise within this cube can be absorbed. However, too much reduction may cause aliasing artifacts.

The size of the cubes can be used as a threshold parameter, and thus can simplify the background test. The larger the cube size, the larger the tolerance permitted. A pixel on the current frame can be classified using only look-up operations in the reduced-color space. In this paper, RGB values are used as pixel values, but other values, such as intensity, YUV, and HSV, can be easily applied.

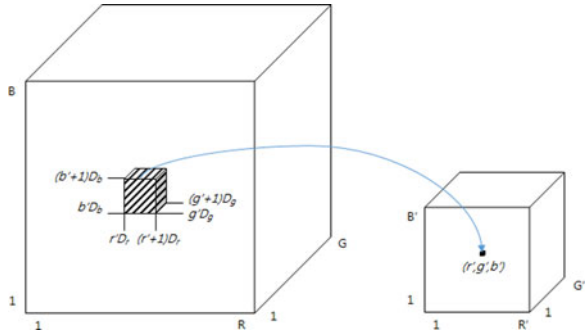


Fig. 3. Color-space reduction.

The color space is mapped from $R * G * B$ to $R' * G' * B'$. Let the ratio of the *R, G,* and *B* directions be D_r, D_g, and D_b, respectively, as follows:

$$R \times G \times B \rightarrow R' \times G' \times B'$$
$$D_r = \frac{R}{R'} \qquad D_g = \frac{G}{G'} \qquad D_b = \frac{B}{B'}. \tag{7}$$

The definition of the background sets B'' is modified from (5) by adding the pixel value part. The background set at [*u, v*] on the reduced image space is defined as follows, using D_r, D_g, and D_b:

$$B''\ [u,v]\ =\ \bigcup_{i=1}^{N}\ \bigcup_{j=uD_x}^{(u+1)D_x}\ \bigcup_{k=vD_y}^{(v+1)D_y}\ \left\{T_i'^{[j,k]}\right\}\ \begin{pmatrix} 1 \le u \le U \\ 1 \le v \le V \end{pmatrix} \times T_i\,[j,k] =\ (r,g,b)$$
$$T_i'[j,k] =\ \left(\frac{r}{D_r}, \frac{g}{D_g}, \frac{b}{D_b}\right). \tag{8}$$

The pixel's RGB values are mapped onto the reduced color space. In this case, RGB values are used and the dimensions of the color space are changed from $R * G * B$ to $R' * G' * B'$. The pixel values within the cube of $(r'D_r \ldots (r'+1)D_r) * (g'D_g \ldots (g'+1)D_g) * (b'D_b \ldots (b'+1)D_b)$ are mapped onto one value (r', g', b'), as in Fig. 3.

A pixel can be classified as foreground or background with only the existence test, if the background set B'' is used. The current pixel p[*x, y*] is background if there exists a pixel value *e* that is an element of $B''[x/D_x, y/D_y]$, and *e* is as below

$$\exists e \in B''\left[\frac{x}{D_x}, \frac{y}{D_y}\right]$$
$$p\ [x,y] =\ (r_p, g_p, b_p)$$
$$e\ =\ \left(\frac{r_p}{D_r}, \frac{g_p}{D_g}, \frac{b_p}{D_b}\right). \tag{9}$$

D. Post Processing

Post processing can enhance the results of background subtraction [30]. The most beneficial methods are a median filter

and a combination of the Opening and Closing morphological operations. The median is used to reduce the noise on an image or signal in signal processing.

Morphological operation can also clean noise. The Opening operation removes a blob and the closing operation fills a hole. A combination of these two operations is used with the small Opening and large Closing structure elements [30]. In this paper, the morphological operation is used as a post-processing method.

III. Experiment

A. Implementation

Our background-subtraction method was implemented using hash-table-based functions. The functions of a hash-table-based algorithm are often provided in computer-vision or system libraries. The time complexities of the search, insert, and delete hash-table functions are O(1) in the average case and O(n) in the worst case [33]. The space complexities are O(n) for both the average and the worst cases [33]. Our method is low cost because its algorithm is simple and only hash-table-based functions are used. Following code is the pseudocode of our method.

```
Boolean HashTable[Max_width/Dx][Max_height/Dy]

//Training Phase
for (i=0; i is Training_frame; i++) {
   frame ← Current_frame
   for (h=0; h < Height; h++) {
      for (w=0; w < Width; w++) {
         r, g, b ← RGB of frame[w][h]
         key ← f(r/Dr, g/Dg, b/Db)
         set_value(HashTable[w/Dx][h/Dy], key, true)
      }
   }
}

//Background detection
do {
   frame ← Current_frame
   for (h=0; h < Height; h++) {
      for (w=0; w < Width; w++) {
         r, g, b ← RGB of frame[w][h]
         key ← f(r/Dr, g/Dg, b/Db)
         If get_value(HashTable[w/Dx][h/Dy], key) is true then
            Mask[w][h] ← Background
         Else
            Mask[w][h] ← Foreground
      }
   }
   //Code using Mask below
} until ends.
```

We implemented our method on the *Microsoft Windows 7* OS for the desktop environment and on the *Android* OS for the mobile. The *SparseMat* functions of *OpenCV for Windows* 3.0 were used as the hash-table implementation for the desktop application. However, we could not use *SparseMat* for the mobile application because *OpenCV for Android* 3.0 does not provide hash-table functionality. The *SparseBooleanArray* of the *Android* 5.1 API was used instead. The *SparseMat* allows three parameters (RGB) as a key, but *SparseBooleanArray* only allows one parameter. Thus, a hash-key function with three parameters is necessary when implemented on a mobile platform.

TABLE II
Experimental Environments

	Desktop	Mobile
Software	Windows 7 OpenCV for Windows 3.0	Android 5.1 OpenCV for Android 3.0 Android 5.1 API
Hardware	8GB RAM Intel Core™ i7-3770 CPU (3.4GHz)	3GB RAM 1.9 GHz Octa Core (1.9 GHz Quad + 1.3 GHz Quad Core) Process (Galaxy Note4)
Hash Table function	SparseMat (OpenCV)	SparseBooleanArray (Android API)

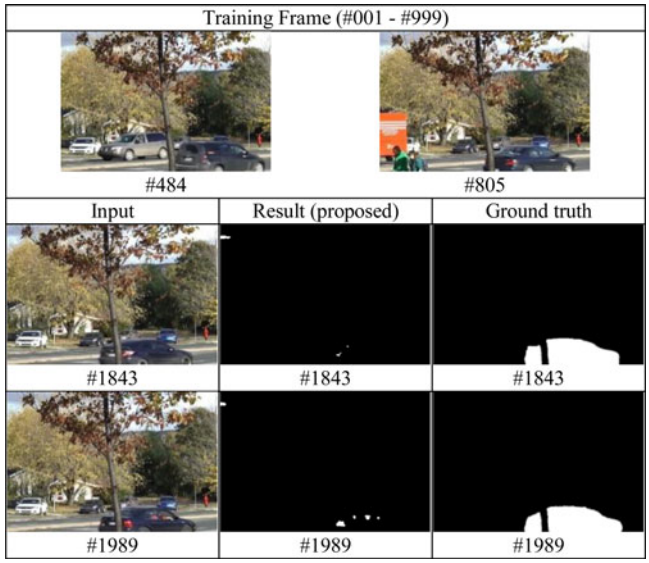


Fig. 4. Similar objects appearing in training frames and input frames.

The platforms, libraries, and hardware we used are described in Table II.

Regarding performance, there was no difference in speed (frames per second (FPS)) for the Windows application between only the camera capturing and using background subtraction. The mobile application ran at 5–6 FPS with no objects (the best case) and 2–3 FPS with no background (the worst case) for a 320 $\times$ 240-pixel camera capture image. Detailed processing times are described in the results section.

B. Datasets

The *CDNET* datasets were used for the experiment [31], [32]. They provided extensive datasets of 11 categories, with four to six video sequences in each containing indoor, outdoor, night, thermal, turbulence, etc. conditions. The *CDNET* datasets provide ground-truth data and evaluation tools on their homepage.

Let *TP* be True Positive, *FP* False Positive, *FN* False Negative, and *TN* True Negative. Then, the following metrics can be computed using their evaluation tools [31].

1) *Recall: TP / (TP + FN)*
2) *Specificity: TN / (TN + FP)*
3) *FPR (False Positive Rate): FP / (FP + TN)*

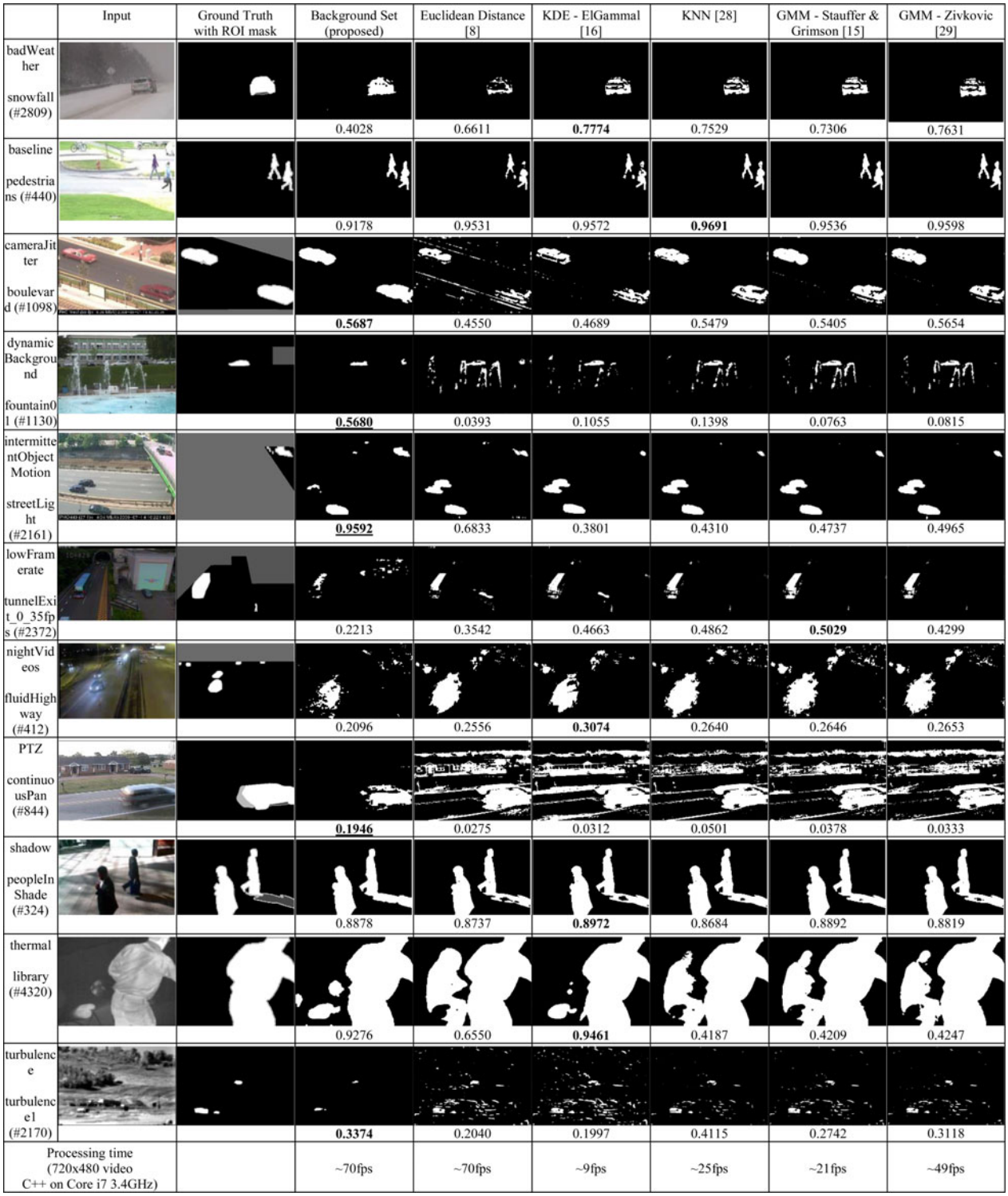


Fig. 5. CDNET dataset result images, F-Measure values, and processing times. First column means category, video title, and frame number. For example, *badWeather* is the category, *snowfall* is the video title, and *#2809* is the frame number in the first item of the second row. In Ground Truth, white (grayscale 255) regions mean motion, black (0) mean static, light gray (170) mean unknown, medium gray (85) mean non-ROI, and dark gray (50) mean shadow [31], [32]. The numerical values below each results are F-Measure values.

4) *FNR (False Negative Rate): FN / (TP + FN)*
5) *PWC (Percentage of Wrong Classifications): 100 ∗ (FN + FP) / (TP + FN + FP + TN)*
6) *F-Measure: (2 ∗ Precision ∗ Recall) / (Precision + Recall)*
7) *Precision: TP / (TP + FP)*

In addition, the *CDNET* homepage provides a table of submitted background-subtraction methods with their results for the above metrics [31]. In that table, we can sort the methods with respect to each metric. Our method's approximate performance among recent approaches can be estimated.

C. Results

The results of our background-set approach, as applied to the *CDNET* datasets, are shown in Fig. 5. One video sequence of each category was selected for comparison with other approaches: *Euclidean Distance* [8], *KDE - ElGammal* [16], *KNN* [28], *GMM - Stauffer & Grimson* [15], and *GMM - Zivkovic* [29]. The results of these approaches using the *CDNET* datasets were provided on the *CDNET* homepage [31]. The results of *KNN* and *GMM - Zivkovic* were obtained by the organizing committee of the *CDNET* datasets using the authors' original code. The results of *Euclidean Distance* and *KDE - ElGammal* were obtained by the organizing committee of the *CDNET* datasets, using the authors' own implementation or *OpenCV*. The results for *GMM - Stauffer & Grimson* were uploaded by the authors to the *CDNET* homepage.

The results in Fig. 5 show the feasibility of our method. In terms of quality, the F-Measure values of the given video sequences were compared. Our method is superior to the others in the categories of *dynamicBackground*, *intermittentObjectMotion,* and *PTZ*. For the *badWeather* and *lowFramerate* categories, ours is not strong. Similar results are shown for other categories.

The processing times of all methods were compared under the same conditions. The programming language is C++ and results were collected using 720×480 videos on a Core i7 3.4 GHz CPU. The processing time of our approach is similar to *Euclidean Distance* [8], which is the best of the others.

D. Parameters

Our method has two explicit and one implicit parameter. The explicit parameters are the ratio of the image-space .0 directions are D_r, D_g, and D_b, respectively, but $D_{\mathrm{rgb}} = D_r = D_g = D_b$). The implicit parameter is the number of training frames. In the experiment with the *CDNET* datasets, the frames from the *first frame* to the *test start frame* were used as training frames.

To show the results according to the parameter variations, the video sequences in the *dynamicBackground* category of the *CDNET* datasets were used, because these sequences are similar to the input under mobile environments. The evaluation results, according to the variations of parameters D_{xy} and D_{rgb}, are shown in Tables IV and V. The cases in the tables were selected to show the variety of the tendencies of the result graphs. 'Category average' means the average of all six cases of the *dynamicBackground* category.

Which evaluation value is important depends on the application, but we focused on the F-Measure value in this experiment to show the feasibility of our method. The F-Measure value is the local maximum (gray row) when D_{rgb} is 16 in the Category average of Table IV. Then, D_{rgb} is fixed as 16 and D_{xy} is changed in Table V. The local maximum of the F-Measure value (gray row) appears when D_{xy} is 4; its evaluation values have been compared to the results of other methods in the *CDNET* homepage [31]. Thirty methods were compared in the *CD.net12* page and 40 methods in the *CD.net14* page. *CD.net14* is a superset of *CD.net12*. Our method was ranked among these methods, and the result is shown in Table III.

TABLE III
COMPARISON WITH METHODS IN THE CDNET HOMEPAGE

Evaluations	Specificity	Recall	Precision	F-Measure	PWC	FNR	FPR
Our result ($D_{\mathrm{xy}} = 4, D_{\mathrm{rgb}} = 16$)	0.9990	0.6881	0.7999	0.7229	0.4676	0.3119	0.0010
CD.net14 rank (total 32)	9th	30th	13th	15th	12th	30th	9th
CD.net12 rank (total 43)	4th	40th	12th	17th	13th	40th	4th

For the post processing, we chose a 5×5 structuring element for the Opening operation and 7×7 for the Closing, based on the experimental results [11].

IV. DISCUSSION

Our background set based method shows slightly better results in the videos of the dynamic background and the limited movement of camera like PTZ. Our method contains the knowledge of all history of pixel value during the training phase. The other methods produce not so good results in the videos of PTZ category because they do not consider camera's movement. For dynamic background, other methods approximate the movement of background such as fall, waving lake, whiffling leaves, etc. But our method memorizes the entire pixel values of dynamic part of background of training frames. Some pixel values are considered as outliers in other methods but it may be valid for background.

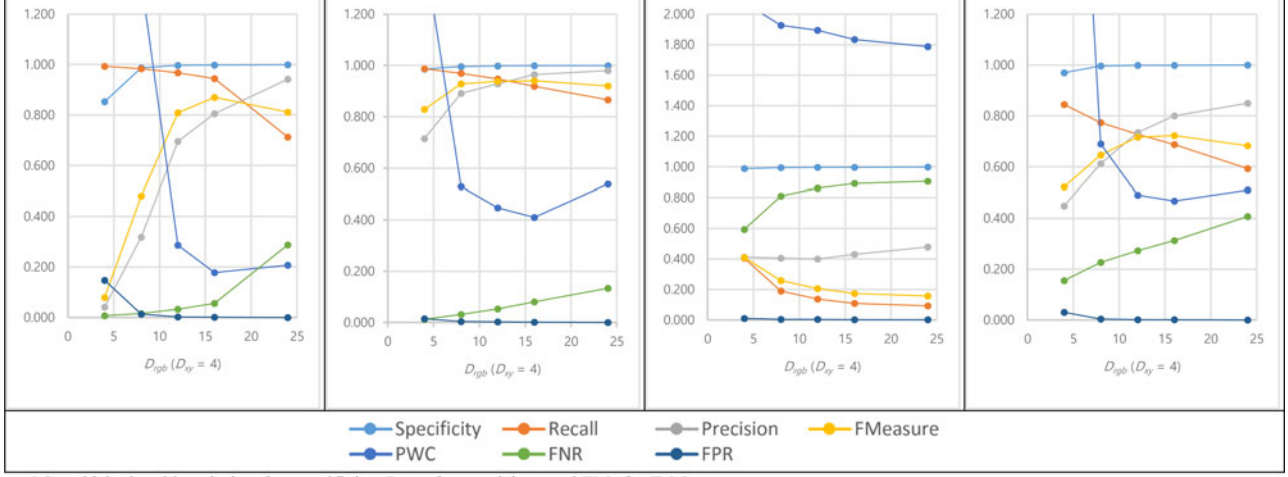
Proposed method has limitation to deal with the appearance of the intermittent objects during training phase. Our method cannot detect the objects that object with similar color already appeared in the training frames at same position as in Fig. 4. The navy-blue cars appeared in the training frames; however, the results of the scene are not good where similar color cars appear, as shown in Fig. 4. Our method does not use input factors, e.g., the object's size, appearance interval, changes in background intensity, etc. In addition, our method does not update the background set for speed. In a future study, for the intermittent objects during training phase, we could count the number of the pixel values and delete them if the number is less than a threshold.

In our method, the parameters must be chosen manually. The graphs in Tables IV and V show no general tendencies or patterns. The maximum or minimum evaluation values are not related to explicit fixed factors, like the image size. The ratio of the image-space reduction (D_{xy}) can be affected by the size of an object or the complexity of the object's boundaries. Further, the color-space reduction ratio (D_{rgb}) can be affected by the sharpness of the contrast between a background and an object. In future work, our method will be developed to adjust the parameters without loss of high performance.

No update policy of background for performance cause weakness to gradual changes of background. It is trade-off between the quality of result and the performance of algorithm. Because we emphasize the performance that our method can applied to mobile device or embedded system, we do not contain background updating mechanism. But we have a future plan to improve our method with background updating

TABLE IV
RESULTS FOR THE D_{RGB} VARIATIONS (*DynamicBackground* CATEGORY, $D_{\mathrm{XY}} = 4$)

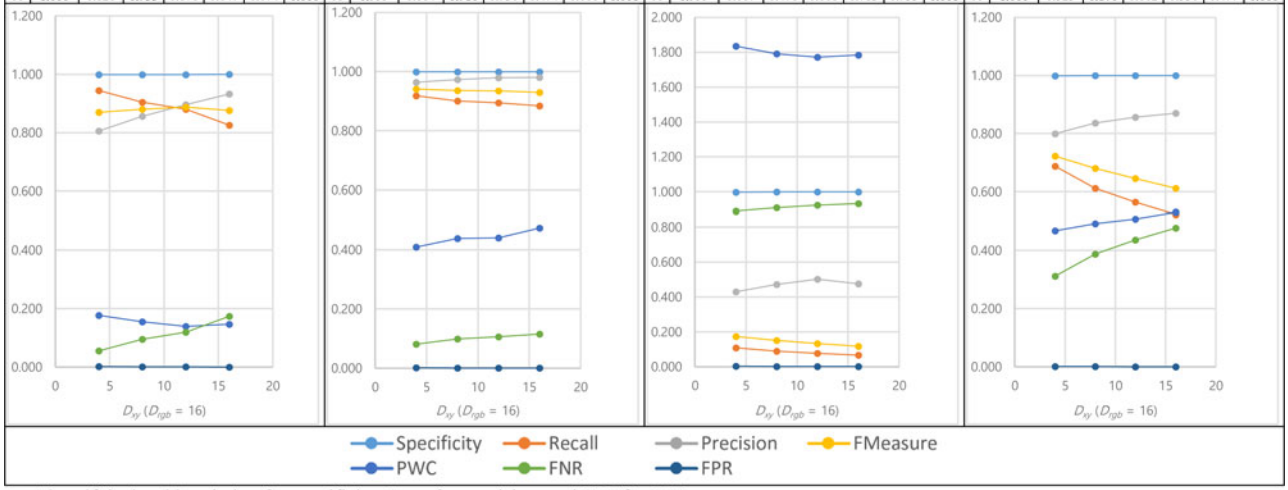
Boats (320x240)							Canoe (320x240)							Fall (720x480)							Category Average						
D_{rgb}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR	D_{rgb}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR	D_{rgb}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR	D_{rgb}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR
4	0.853	**0.993**	0.041	0.079	14.616	**0.007**	0.147	4	0.985	**0.987**	0.714	0.829	1.445	**0.013**	0.015	4	0.990	**0.406**	0.413	**0.409**	2.077	**0.594**	0.010	4	0.970	**0.845**	0.448	0.521	3.189	**0.155**	0.030
8	0.987	0.984	0.318	0.481	1.334	0.016	0.013	8	0.996	0.969	0.891	0.928	0.529	0.031	0.004	8	0.995	0.189	0.406	0.258	1.927	0.811	0.005	8	0.996	0.773	0.612	0.646	0.689	0.227	0.004
12	0.997	0.967	0.696	0.809	0.286	0.033	0.003	12	0.997	0.947	0.928	0.938	0.447	0.053	0.003	12	0.996	0.138	0.400	0.205	1.894	0.862	0.004	12	0.998	0.728	0.735	0.717	0.490	0.272	0.002
16	0.999	0.945	0.806	**0.870**	**0.178**	0.055	0.001	16	**0.999**	0.919	0.964	**0.941**	**0.410**	0.081	**0.001**	16	0.997	0.109	0.430	0.174	1.834	0.891	0.003	16	**0.999**	0.688	0.800	**0.723**	**0.468**	0.312	**0.001**
24	**1.000**	0.713	**0.942**	0.811	0.208	0.287	**0.000**	24	**0.999**	0.866	**0.980**	0.919	0.538	0.134	**0.001**	24	**0.998**	0.094	**0.478**	0.157	**1.787**	0.906	**0.002**	24	**0.999**	0.593	**0.850**	0.683	0.509	0.407	**0.001**



[a] Specif. is the abbreviation for specificity, Prec. for precision, and FM. for F-Measure.

TABLE V
RESULTS FOR THE D_{XY} VARIATIONS (*DynamicBackground* CATEGORY, $D_{\mathrm{RGB}} = 16$)

Boats (320x240)								Canoe (320x240)								Fall (720x480)								Category Average							
D_{xy}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR	D_{xy}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR	D_{xy}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR	D_{xy}	Specif.	Recall	Prec.	FM.	PWC	FNR	FPR
4	0.999	**0.945**	0.806	0.870	0.178	**0.055**	0.001	4	**0.999**	**0.919**	0.964	**0.941**	**0.410**	**0.081**	**0.001**	4	0.997	0.109	0.430	**0.174**	1.834	**0.891**	0.003	4	0.999	**0.688**	0.800	**0.723**	**0.468**	**0.312**	0.001
8	0.999	0.905	0.856	0.880	0.155	0.095	0.001	8	**0.999**	0.901	0.973	0.936	0.439	0.099	**0.001**	8	0.998	**0.090**	0.472	0.151	1.791	0.910	0.002	8	0.999	0.612	0.837	0.681	0.491	0.388	0.001
12	0.999	0.880	0.896	**0.888**	**0.139**	0.120	0.001	12	**0.999**	0.894	**0.980**	0.935	0.441	0.106	**0.001**	12	**0.999**	0.076	**0.501**	0.132	**1.771**	0.924	**0.001**	12	**1.000**	0.564	0.857	0.645	0.507	0.436	**0.000**
16	**1.000**	0.826	**0.933**	0.876	0.147	0.174	**0.000**	16	**0.999**	0.884	**0.980**	0.930	0.474	0.116	**0.001**	16	**0.999**	0.067	0.476	0.118	1.783	0.933	**0.001**	16	**1.000**	0.523	**0.870**	0.612	0.531	0.477	**0.000**



[a] Specif. is the abbreviation for specificity, Prec. for precision, and FM. for F-Measure.

algorithm. For the gradual changes of background, we could consider the neighbor's values of the pixel values of background in future work. If new pixel value is the neighbor's values with 6 or 26-connectivity, it could be added to background set.

In our approach, more memory needed than other methods when it deals dynamic backgrounds and the video of limited movement of camera. Without approximation of background, we use more memory but run faster than other methods. In future study, we could save memory more than current method with deletion of deprecated background subset.

V. CONCLUSION

A low-cost background modeling technique was proposed in this paper. It used background sets defined on reduced image and color spaces. Because the background sets contained all the values of each pixel on the training frames, dynamic backgrounds, like waves, trees, fans, etc., could be subtracted. For the same reason, our method showed better results for the pan-tilt-zoomed camera input.

The image-space reduction could deal with the jittered and unsteady input from handheld mobile devices. Because a block of the image space was reduced to a single

point, the jittered movement within that block could be tolerated.

Color noise, like the scattered RGB values of a digital camera, could be compensated with the color-space reduction. The scattered values were approximated using the color-space reduction cubes. If the size of the cubes were used as the threshold, the noise within the cubes would be tolerated.

Our method showed the high-speed results, compared with the other approaches. Using the color-space reduction, the background test could be completed using only look-up operations, instead of comparisons with a threshold. Our method had a low cost because it was implemented with a simple algorithm and low-complexity hash-table-based functions.

The proposed method in this paper had a low cost and an advantage in dynamic backgrounds. The results compared with other methods showed the feasibility of our method. It can be useful in mobile or embedded environments.

REFERENCES

[1] M. Hirose, T. Ogi, and T. Yamada, "Integrating live video for immersive environments," *IEEE Multimedia Mag.*, vol. 6, no. 3, pp. 14–22, Jul./Sep. 1999.

[2] C. Bo-Hao and H. Shih-Chia, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 837–847, Apr. 2014.

[3] C. Shao-Yi, M. Shyh-Yih, and C. Liang-Gee, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 577–586, Jul. 2002.

[4] F. El Baf, T. Bouwmans, and B. Vachon, "Comparison of background subtraction methods for a multimedia application," in *Proc. 14th Int. Workshop Syst., Signals Image Process./6th EURASIP Conf. Focused Speech Image Process., Multimedia Commun. Services*, Jun. 2007, pp. 385–388.

[5] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection—A systematic survey," *Recent Patents Comput. Sci.*, vol. 4, no. 3, pp. 147–176, 2011.

[6] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection—A Survey," *Recent Patents Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.

[7] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2004, vol. 4, pp. 3099–3104.

[8] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Comparative study of background subtraction algorithms," *J. Electron. Imaging*, vol. 19, no. 3, pp. 1–12, 2010.

[9] B. Lee and M. Hedley, "Background estimation for video surveillance," in *Proc. Image Vis. Comput. New Zealand*, 2002, pp. 315–320.

[10] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Mach. Vis. Appl.*, vol. 8, no. 3, p. 187–193, 1995.

[11] C. R. Jung, "Efficient background subtraction and shadow removal for monochromatic video sequences," *IEEE Trans. Multimedia*, vol. 11, no. 3, pp. 571–577, Apr. 2009.

[12] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, "Extracting roadway background image: Mode-based approach," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1944, pp. 82–88, 2006.

[13] X. Shiming, P. Chunhong, N. Feiping, and Z. Changshui, "Interactive image segmentation with multiple linear reconstructions in windows," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 342–352, Apr. 2011.

[14] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[15] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Conf. Comput. Vis. Pattern Recog.*, Jun. 1999, vol. 2, pp. 246–252.

[16] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. 6th Eur. Conf. Comput. Vis.*, 2000, pp. 751–767.

[17] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[18] S. Ja-Won and K. Seong Dae, "Recursive on-line $(2D)^2$ PCA and its application to long-term background subtraction," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2333–2344, Dec. 2014.

[19] M. Yamazaki, G. Xu, and Y. W. Chen, "Detection of moving objects by independent component analysis," in *Proc. 7th Asian Conf. Comput. Vis.*, 2006, pp. 467–478.

[20] H. Lin, T. Liu, and J. Chuang, "A probabilistic SVM approach for background scene initialization," in *Proc. Int. Conf. Image Process.*, Jun. 2002, vol. 3, pp. 893–896.

[21] M. H. Sigari, N. Mozayani, and H. Pourreza, "Fuzzy running average and fuzzy background subtraction: concepts and application," *Int. J. Comput. Sci. Netw. Security*, vol. 8, no. 2, pp. 138–143, 2008.

[22] F. El Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of Gaussians model: Application to background modeling," in *Proc. Int. Symp. Adv. Vis. Comput.*, Dec. 2008, pp. 772–781.

[23] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1614–1627, Nov. 2007.

[24] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.

[25] D. Butler, S. Sridharan, and V. M. Bove Jr., "Real-time adaptive background segmentation," *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, vol. 3, pp. III-349–III-352.

[26] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *J. Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, Jun. 2005.

[27] S. Messelodi, C. M. Modena, N. Segata, and M. Zanin, "A kalman filter based background updating algorithm robust to sharp illumination changes," in *Proc. Image Anal. Process.*, 2005, pp. 163–170.

[28] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recog. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[29] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recog.*, Aug. 2004, vol. 2, pp. 28–31.

[30] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1937–1944.

[31] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Workshop Change Detection (CDW-2012)*, pp. 1–8. (Jun. 2012). [Online]. Available: http://www.changedetection.net/

[32] Y. Wang *et al.*, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 23–28, 2014, pp. 393–400.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Hash tables," in *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009, pp. 253–285.

[34] J. M. Guo *et al.*, "Hierarchical method for foreground detection using codebook model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 804–815, Jun. 2011.

[35] J. M. Guo *et al.*, "Fast background subtraction based on a multilayer codebook model for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1809–1821, Oct. 2013.

[36] S. C. Huang and B. H. Do, "Radial basis function based neural network for motion detection in dynamic scenes," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 114–125, Jan. 2014.

[37] S. C. Huang and B. H. Chen, "Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1920–1931, Dec. 2013.

[38] S. C. Huang and B. H. Chen, "Automatic moving object extraction through a real-world variable-bandwidth network for traffic monitoring systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2099–2112, Apr. 2014.

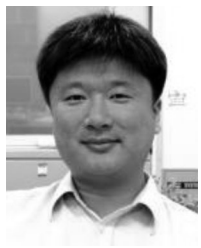
[39] O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 450–462, Feb. 2013.

[40] F. C. Cheng, B. H. Chen, and S. C. Huang, "A background model reinitialization method based on sudden luminance change detection," *Eng. Appl. Artif. Intell.*, vol. 38, pp. 138–146, 2015.

[41] F. C. Cheng, B. H. Chen, and S. C. Huang, "A hybrid background subtraction method with background and foreground candidates detection," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 1, 2015, Art. no. 7.
[42] S. C. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 1–14, Jan. 2011.
[43] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.
[44] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2496–2504.
[45] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, "MahNMF: Manhattan non-negative matrix factorization," *CoRR*, 2012. [Online]. Available: http://arxiv.org/abs/1207.3438
[46] C. Navasca and X. Wang, "Adaptive low rank approximation of tensors," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 99–105.
[47] S. Javed, T. Bouwmans, and S. Jung, "Stochastic decomposition into low rank and sparse tensor for robust background subtraction," in *Proc. 6th Int. Conf. Imag. Crime Prevention Detect.*, Jul. 2015, pp. 1–6.
[48] A. Sobral, C. Baker, T. Bouwmans, and E. Zahzah, "Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction," in *Proc. Int. Conf. Image Anal. Recog.*, Oct. 2014, pp. 94–103.
[49] M. Al-Qizwini and H. Radha, "Truncated and smoothed Schatten-p function for robust tensor recovery," in *Proc. 49th Annu. Conf. Inf. Sci. Syst.*, Mar. 2015, pp. 1–4.
[50] J. Shi, Q. Yin, X. Zheng, and W. Yang, "Alternating direction method of multipliers for generalized low-rank tensor recovery," *Algorithms*, vol. 9, no. 2, pp. 1–16, 2016.
[51] A. Elqursh and A. Elgammal, "Online moving camera background subtraction," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 228–241.
[52] B. Valentine, S. Apewokin, L. Wills, and S. Wills, "An efficient, chromatic clustering-based background model for embedded vision platforms," *Comput. Vis. Image Understand.*, vol. 114, no. 11, pp. 1152–1163, 2010.
[53] C. Cuevas and N. Garcia, "Efficient moving object detection for lightweight applications on smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 1–14, Jan. 2013.
[54] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1438–1452, Oct. 2011.
[55] M. Casares, S. Velipasalar, and A. Pinto, "Light-weight salient foreground detection for embedded smart cameras," *Comput. Vis. Image Understand.*, vol. 114, no. 11, pp. 1223–1237, 2010.
[56] P. Sarisaray-Boluk and K. Akkaya, "Performance evaluation of background subtraction algorithms for android devices deployed in wireless multimedia sensor networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, 2014, pp. 779–784.
[57] R. Hamid, A. D. Sarma, D. DeCoste, and N. Sundaresan, "Fast approximate matching of cell-phone videos for robust background subtraction, " *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1404.5351
[58] S. Azmat, L. Wills, and S. Wills, "Spatio-temporal multimodal mean," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, Apr. 2014, pp. 81–84.
[59] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep.-Oct. 2009, pp. 1219–1225.
[60] D. E. Butler, V. M. Bove Jr., and S. Sridharan, "Real-time adaptive foreground/background segmentation," *EURASIP J. Adv. Signal Process.*, vol. 2005, no. 14, pp. 1–13, 2005.
[61] M. Xiao and L. Zhang, "A background reconstruction algorithm based on modified basic sequential clustering," in *Proc. ISECS Int. Colloq. Comput., Commun., Control, Manage.*, 2008, pp. 47–51.
[62] Cambridge in Colour, "Digital Camera Image Noise - Part 2." (Feb. 2016). [Online]. Available: http://www.cambridgeincolour.com/tutorials/image-noise-2.htm

Hasup Lee received the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2007.

He is a Research Fellow with the VR Laboratory, Department of Internet and Multimedia Engineering, Konkuk University, Seoul, South, Korea, and a Researcher with the SDM Research Institute, Graduate School of System Design and Management, Keio University, Tokyo, Japan. He was previously a Senior Assistant Professor with the Visual Simulation Laboratory, Graduate School of System Design and Management, Keio University. His research interests include virtual reality, computer graphics, computer vision, and their applications.



HyungSeok Kim (S'08–M'09) received the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in February 2003.

He is a Professor with the Department of Internet and Multimedia Engineering, Konkuk University, Seoul, South Korea. Before joining Konkuk University, he was a Senior Researcher with the MIRA Laboratory, University of Geneva, Geneva, Switzerland. His research interests include real-time interaction in virtual environments and multimodal interaction mechanisms, shape modeling for real-time rendering, and evoking believable experiences in virtual environments.

Jee-In Kim (M'13) received the B.S. degree in computer engineering from Seoul National University, Seoul, South Korea, in 1982, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1984, and the Ph.D. degree in computer science and information from the University of Pennsylvania, Philadelphia, PA, USA, in May 1993.

He is a Professor of internet and multimedia and advanced technology fusion with Konkuk University, Seoul, South Korea. He is also the Director of the Digital Contents Research Center Konkuk University. His current research interests include ubiquitous computing, visualization, and HCI.