Background Subtraction With Neighbor-based Intensity Correction Algorithm

Thien Huynh-The *, Oresti Banos *, Ba-Vui Le *, Dinh-Mao Bui *, Sungyoung Lee *, Yongik Yoon †, Thuong Le-Tien ‡
* Department of Computer Engineering
Kyung Hee University, Gyeonggi-do, 446-701, Korea
Email: thienht,oresti,lebavui,sylee@oslab.khu.ac.kr, mao.bui@khu.ac.kr
† Department of Multimedia Science
Sookmyung Women's University, Seoul, 140-172, Korea
Email: yiyoon@sookmyung.ac.kr
‡Department of Electric and Electronics Engineering
Hochiminh University of Technology, Hochiminh City, Vietnam
Email: thuongle@hcmut.edu.vn

***Abstract*—An efficient foreground detection algorithm is presented in this work to be robust against consecutively illuminance changes and noise, and adaptive with dynamic speeds of motion in the background. The scene background is firstly modeled by a novel algorithm, namely Neighbor-based Intensity Correction, which identifies and modifies motion pixels extracted from the difference of the background and the current frame. Concretely the first frame is assumed as an initial background to be updated at each new coming frame based on the mechanism of the standard deviation value comparison. Two pixel windows used for standard deviation calculation are generated surrounding a corresponding motion pixel from the background and the current frame. The steadiness of the current background at the pixel-level is measured by a constantly updating factor to decide the usage of the algorithm or not. In the next stage, the foreground of the current frame are detected by the background subtraction scheme with an optimal Otsu threshold. This method is evaluated on various well-known datasets in the object detection and tracking area and then compared with recent approaches via some common quantitative measurements. From experimental results, the proposed method achieves the better results (approximately 5−20%) in term of the foreground detection accuracy.**

I. Introduction

Despite the take-up of the use of background subtraction technique in visual surveillance systems, this approach presents some crucial issues including the performance in term of computational cost and accuracy of the background estimation. In the background subtraction techniques, an observed image is compared with an estimated background image which does not contain objects and can be achieved by the background modeling algorithm. This comparison process separates the image into two sets of pixels: the foreground contains the object area with 1-bit presentation and the background is a complementary set with 0-bit presentation. Fundamentally the background is defined as a reference frame with pixel values visible most of the time. Although existing models are capable to enhance the accuracy by using more frames for estimation, they are ineffective for most practical situations. The limitations in accuracy and the important role in the foreground detection motivated the authors to discuss issues related to the background estimation.

A simple scheme to model the background of a scene is use of the statistical approach [1], [2]. The limitation of statistical methods can wrongly engage the foreground objects into the background in the case of non-motion objects remaining for a long time. Gaussian Mixture Models (GMM), the most commonly used technique for the background estimation, is firstly introduced by Stauffer et al. [3]. In this technique, each pixel value is estimated using the separated Gaussian mixture and continuously learned by an online approximation. Several improved versions [4]–[9] have been proposed as the main contribution in the object detection method using the background subtraction technique. For example, Zivkovic [4] considered an improved adaptive GMM, in which parameters and components of the mixture model are constantly chosen for each adaptive pixel. Improvement of the convergence rate without comprising model stability, presented by Lee et al. [5], is another development of GMM. A recent method, suggested by Elqursh et al. [7], describes tracking content in the low-dimensional space and then synthesizes by GMM at each coming frame. In order to eliminate the illumination change and noise in intelligent video-based surveillance systems, a novel GMM-based solution was proposed by Li et al. [8]. The approach has three key contents: an explicit spectral reflection model analysis, an online expectation-maximization algorithm, and a two-stage foreground detection algorithm. Followed by probabilistic regularisation, a method based on Dirichlet process GMM [9] was proposed to estimate per-pixel background distributions.

Although GMM-based improvements have been proposed to grow upon the detection performance in difficult scenes, they still have general limitations, such as fail detection in high speed movement and parameter estimation issue. To avoid the mission of finding an appropriate shape for the probabilistic model, researchers pay their attention to nonparametric approaches for the background modeling. The real-time algorithms in [10] quantizes the background pixel values into codebooks that describe a compressed form of background model for a number of frames. Although obtaining the high performance in the real time environments, the drawback of codebook approaches comprises the long period of time to

978-1-4673-8374-5/15/$31.00 ©2015 IEEE

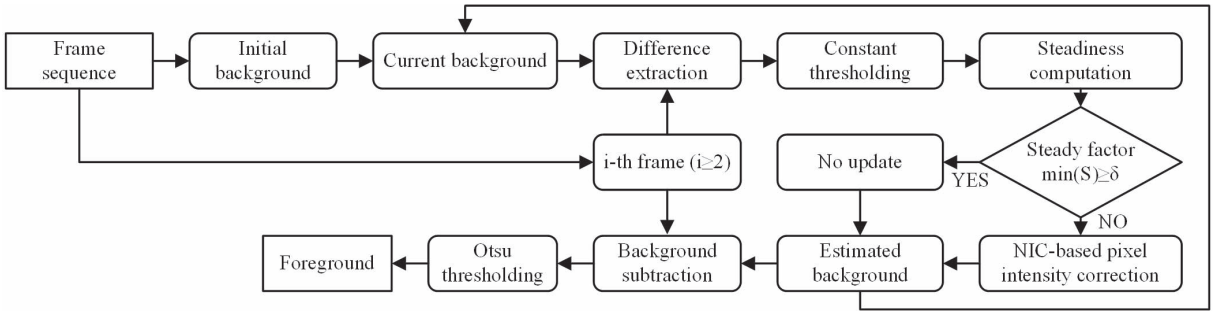

Fig. 1. The workflow of the background estimation using the NIC algorithm.

construct the model and also the high memory use to store codewords. Another highly used nonparametric approach in the background subtraction is the Kernel Density Estimation (KDE) [11]. This technique estimates the probability density function with the histogram to redefine the values of current background pixels. Liu et al. [12] presented a hybrid model, integrated by KDE and GMM, to construct the probability density function of the background and moving object model. Although KDE-based methods can provide fast responses to the high speed movement scenes, handling concomitant events at various speeds of this approach is restricted due to the first-in first-out manner. In recent years, a background modeling technique, namely Visual Background Extraction (ViBE), proposed by Barnich [13], which determines whether a pixel belongs to the background by randomly comparing its intensity with neighborhoods. Although ViBE can provides satisfactory detection results when compared with existing approaches, it has the problematic issue with harsh conditions such as scenes with darker background, shadows, and frequent background change. Standing on the aspect of minimizing the memory requirement, a dual-layer background model, one for low adaptation speed with the long-term background and another for high adaptation speed with the short-term background, was presented in research of Gruenwedel et al. [14]. The Radial Basis Function (RBF) through artificial neural networks was described in [15] as an unsupervised learning process for multi-background generation.

In this paper, the authors proposed a novel method for foreground detection based on the background subtraction scheme, in which the initial background is assumed to the first frame and consecutively updated at each new coming frame. In the background estimation stage, the motion pixels extracted from the difference frame are adjusted to background truth intensity based on considering the intensity patterns of two windows from the background and the current frame. The rule for correction is developed through the comparison of two standard deviation values. The updated background is then used to detect and segment the foreground with an optimal threshold determined from the Otsu method.

II. Neighbor-based Intensity Correction (NIC) for Foreground Detection

A. Neighbor-based Intensity Correction for Background Estimation

In this study, the background is consecutively modelled and updated from motion information of the current frame during the foreground extraction process. The workflow of the proposed algorithm is concretely represented in Fig. 1 with the input as the frame sequence and the output is the estimated background. As the prior knowledge, the first frame from the input video sequence is assumed as the initial background:

$$B_1(x,y) = F_1(x,y) \tag{1}$$

where $B_1(x,y)$ and $F_1(x,y)$ are the intensity values of a pixel at the coordinate (x,y) with $x \leq P, y \leq Q$, in the initial background and the first frame, where P and Q are the horizontal and vertical size of the input frame.

For the i^{th} coming frame ($\forall i \geq 2$), the background image B_i used for the background subtraction will be estimated from B_{i-1}. In the first step, the difference between the current background and the current frame, denoted D_i, is calculated through the following equation:

$$D_i(x,y) = |F_i(x,y) - B_{i-1}(x,y)| \quad ;\forall i \geq 2 \tag{2}$$

Before extracting the difference image, the background and the current frame are necessarily converted from the color to grayscale image.

The difference image D_i contains information about the moving objects and noise, therefore, D_i needs to be segmented into the background and moving object areas by constant thresholding:

$$\mathcal{D}_i(x,y) = \begin{cases} 1 & ;\forall D_i(x,y) \geq \tau \\ 0 & ;\forall D_i(x,y) < \tau \end{cases} \tag{3}$$

where τ is the constant value. The binary image $\mathcal{D}_i$ has 0-bit pixels representing the non-motion areas and 1-bit pixels representing the motion areas. In principle, the moving objects have greater difference when compared with the light change or shadow artifact. If using a high value of τ, the noise is eliminated as well, but some motion pixels can be unexpectedly misidentified. In contrast, the noise pixels are sometimes recognized as the motion pixels in the case of a small value of τ. Therefore, it can be seen that parameter τ has an influence on the results of $\mathcal{D}_i$ and needs to be carefully selected through experimental evaluations hereafter.

In the proposed algorithm, the steadiness of each pixel, denoted $S(x,y)$, is suggested to calculate the number of intensity changing times and updated at each coming frame. If value of a pixel is changed in two consecutive frames, it is possible to say that this pixel is less steady than non-change intensity pixels. The steady factor is utilized to evaluate the robustness of the current estimated background. Concretely, the background

will become close to the true background after a number of frames, hence the estimation process may be ignored and the current background is maintained for foreground extraction. The steadiness of each pixel is computed and accumulated in a sequence of frames by the following equation:

$$S_i(x,y) = \begin{cases} S_{i-1}(x,y) - 1 & ;\forall \mathcal{D}_i(x,y) = 1 \\ S_{i-1}(x,y) + 1 & ;\forall \mathcal{D}_i(x,y) = 0 \end{cases} \quad (4)$$

where S_i is the steady matrix at the i^{th} frame. It is initialized with zero value, i.e. $S_1(x,y) = 0$, and has the same size with the input frames. It can be seen that the steady value of a non-motion pixel is greater than the value of a motion pixel in accumulating frame by frame. For instance, an arbitrary pixel p is detected as the motion pixel in t_1 frames, while as the non-motion pixel in t_2 frames by (3) after $T = t_1 + t_2$ frames. Calculated by (4), the steady value $S_T(x_p, y_p) = (-1)\,t_1 + (1)\,t_2$ is negative if $t_1 > t_2$, and positive if $t_1 < t_2$.

The intensity correction algorithm is applied or not for the current background based on considering the steady matrix S. Concretely, if the minimal value of steady factor is greater than the steady threshold, denoted $\delta > 0$, i.e. $\min_{(x,y)}(S_i(x,y)) \geq \delta$, the intensity modification is ignored since the current background is provisionally robust. The current background is therefore maintained for consideration of the next frame, i.e. $B_i = B_{i-1}$ for the $(i+1)^{th}$ coming frame, and then directly support to the foreground extraction stage. In the opposite case, $\min_{(x,y)}(S_i(x,y)) < \delta$, the intensity modification is implemented by the NIC algorithm on the current background. This process is briefed as follows:

$$\begin{array}{ll} \text{Implement\_NIC} & ;if \min_{(x,y)}(S_i(x,y)) < \delta(i) \\ \text{None} & ;if \min_{(x,y)}(S_i(x,y)) \geq \delta(i) \end{array} \quad (5)$$

Through the steady factor of each pixel, it is capable to assess the robustness of the current estimated background. Typically, the scene consisting of more movements, represented through the object density, requires more frames to converge to the background truth since the pixel intensities are successively changed. Therefore, the steady threshold δ should be set to a great value for the dense scenes and a small value for the sparse scenes. However, to be capable in the dynamic environments, the value of δ needs to be automatically identified based on the moving object area. In this work, the authors describe the relationship between the steady threshold and the object density by a monotonically increasing function:

$$\delta(i) = 10^{-\log_2(1-r)} \quad ;0 \leq r < 1 \quad (6)$$

where $\delta(i)$ presents the steady threshold at the i^{th} frame, r is the ratio of number of foreground pixel over the total of pixels, i.e. $r = {num(D_i(x,y) = 1)}/{PQ}$ according to (3).

To select pixels in the motion set $\mathcal{D}_i(x,y) = 1$ for the NIC algorithm, the authors additionally consider the condition of steady factor, concretely, with motion pixels carrying negative steady value. Re-identification of candidate pixels will reduce the computation cost for inappropriate points generated from the sudden light change. The pixels are filtered by two conditions:

$$\mathcal{P}_i = \{(x,y) \,|\, [\mathcal{D}_i(x,y) = 1] \cap [S_i(x,y) < 0]\} \quad (7)$$

where $\mathcal{P}_i$ is a set of filtered pixels.

The intensity correction algorithm is executed for pixels in the set $\mathcal{P}_i$. The main idea of the algorithm is illustrated in Fig. 2, in which the motions are simply described. A first case for the single-pixel shifting is represented in the first row consisting of the background image in Fig. 2(a) and the current frame in Fig. 2(b). The pixels belong to the set $\mathcal{P}$ are shown in Fig. 2(c). In this step, eight pixels in the set $\mathcal{P}$ need to be modified to the correct value, concretely, four pixels on the left side should be adjusted to the values of the corresponding pixels in the current frame and four pixels on the right side should be adjusted to the background pixel value. Accordingly, two windows are constructed from the background image and the current frame surrounding a filtered pixel, denoted $W^B_{(x,y)}$ and $W^F_{(x,y)}$, respectively. For instance with the pixel p_1 in Fig. 2(c), two windows are identified in Fig. 2(d) and (e). The intensity patterns of two windows are different, i.e. the ratio between the number of motion and non-motion pixels are dissimilar. This difference is exploited for correction, consequently, the standard deviation values of samples in two sets (known as pixels in the windows) are calculated. A small value indicates that the pixel values tend to be closer to the average while a greater value points out that the values of pixels are dispersedly spread.

Fundamentally, the standard deviation σ of a square window is generally calculated by the following equation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{p=1}^{n}\sum_{q=1}^{n}(I(p,q) - \mu)^2} \quad (8)$$

where n is the size of a square window and $N = n^2$ is the number of pixels. The mean of intensity μ of the window is determined from the pixel intensity $I(p,q)$ in a sample image I by:

$$\mu = \frac{1}{N}\sum_{p=1}^{n}\sum_{q=1}^{n} I(p,q) \quad (9)$$

For each pixel in $\mathcal{P}_i$, two standard deviation values, denoted $\sigma^B_{(x,y)})$ and $\sigma^F_{(x,y)}$, are calculated from two windows of the current background and the current frame. The rule for correction algorithm is employed based on the result of comparison process between two standard deviation values as follows:

$$B_i(x,y) = \begin{cases} B_{i-1}(x,y) & ;\forall (x,y) \notin \mathcal{P}_i \\ B_{i-1}(x,y) & ;\forall (x,y) \in \mathcal{P}_i \,\big|\, \sigma^F_{(x,y)} \geq \sigma^B_{(x,y)} \\ F_i(x,y) & ;\forall (x,y) \in \mathcal{P}_i \,\big|\, \sigma^F_{(x,y)} < \sigma^B_{(x,y)} \end{cases} \quad (10)$$

where B_i is the estimated background image at the i^{th} frame.

In the window construction, the size of $(n \times n)$ can be adjusted based on the speed of object movement. To investigate the influence of window size on the result of intensity correction rule, some particular values have been assumed for several cases. Let denote n_0 and n_1 are the number of non-motion and motion pixels with the intensity g_0 and g_1, respectively. So there are $(n_0 + n_1)$ pixels contained in the window. The mean of pixel values is calculated by (8) is:

$$\mu = \frac{g_0 n_0 + g_1 n_1}{n_0 + n_1} \quad (11)$$

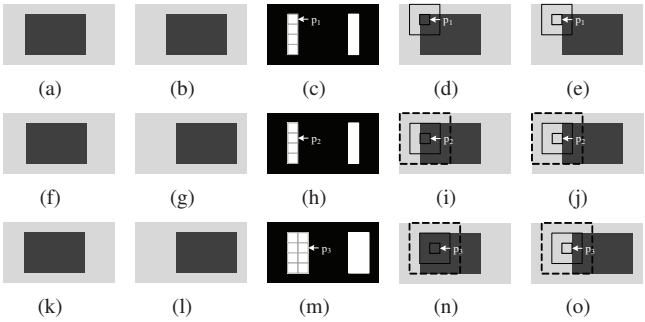

Fig. 2. An illustration of NIC operation in different cases of moving objects: the single-pixel shifting (first and second row) and the multi-pixel shifting (third row). From the left to right: the background image, the current frame, the difference image, the background and current frame with (3×3) and (5×5) windows captured surrounding a motion pixel.

The standard deviation can be derived under the variant developed from (7):

$$\begin{aligned}\sigma &= \sqrt{\frac{1}{n_0+n_1}\left[n_0(g_0-\mu)^2 + n_1(g_1-\mu)^2\right]} \\ &= \frac{\sqrt{n_0 n_1}}{n_0+n_1}\left|g_0-g_1\right|\end{aligned} \tag{12}$$

Through (12) for the standard deviation calculation, the term $|g_0 - g_1|$ and $(n_0 + n_1)$ are constant components, so the result is decided by the term $\sqrt{n_0 n_1}$, i.e. the standard deviation value depends on the number of motion and non-motion pixels. Let us consider the first case represented in the first row of Fig. 2 with the window size of (3×3). The window $W^B_{p_1}$ captured from the current background contains five non-motion pixels and four motion pixels in Fig. 2(d), while seven non-motion pixels and two motion pixels are covered by the window $W^F_{p_1}$ captured from the current frame in Fig. 2(e). Due to $\sigma^F_{(x,y)}\left(=\sqrt{2\times 7}\right) < \sigma^B_{(x,y)}\left(=\sqrt{4\times 5}\right)$, the p_1-pixel intensity is modified from g_1 to g_0 by referring (10).

Let us investigate the next example shown in the second row of Fig. 2, the problem occurs when the number of motion and non-motion pixels in two (3×3) windows, denoted by the solid line boudary, established at the p_2-pixel are respectively antithetic. This lead to the erroneous background correction, i.e. p_2 will be preserved by its intensity instead of modifying from g_1 to g_0. This problem can be overcome if the window size is changed. Particularly in this example, the size is modified from (3×3) to (5×5), denoted by the hashed line boundary. In the third case, the multi-pixel shifting motion which usually occurred in the practice environment is illustrated in Fig. 2(m). The intensity modification for p_3 is incorrect if uses the (3×3) window. Similar to the second case, this drawback is solved if the window size is changed to (5×5). The terms of $\sqrt{n_0 n_1}$ are re-calculated to bring the correct modification. Through above examples, the window size effects to the quality of the estimated background via computation of the standard deviation. Some experimental evaluations hereafter prove an influence of the window size on performance in term of background estimation accuracy.

After the intensity correction process with NIC algorithm, the estimated background is then entered to the foreground extraction as an input and stored to continuously process in the $(i+1)^{th}$frame.

B. Foreground Extraction using Background Subtraction Scheme

In this stage, the foreground is detected based on the background subtraction scheme with an adaptive threshold. The difference is extracted from the current frame F_i and the estimated background image B_i by re-using (2):

$$D^*_i(x,y) = \left|F_i(x,y) - B_i(x,y)\right| \quad ;\forall i \geq 2 \tag{13}$$

The foreground is then segmented based on the difference image D^*_i by an optimum value identified from the Otsu method [16]. The Otsu method is fundamentally formulated to perform clustering-based image thresholding for segmentation, in which two pixel classes (foreground pixels and background pixels) are assumed to be sufficiently distinguishable. As the principle concept, the Otsu method is to calculate the optimal threshold to separate an image into the background area, denoted G_0, and the foreground area, denote G_1. Through the minimization of the intra-class variance (the variance within the class), the threshold can reduce the error in classification. The threshold is exhaustively sought as an intensity based on the weighted sum of variance of two classes:

$$\sigma^2_\omega(g) = \omega_{G_0}(g)\,\sigma^2_{G_0}(g) + \omega_{G_1}(g)\,\sigma^2_{G_1}(g) \tag{14}$$

where $\omega_{G_0}(g)$ and $\omega_{G_0}(g)$ are the class probabilities at the intensity g. Corresponding two pixel classes, $\sigma^2_{G_0}$ and $\sigma^2_{G_1}$ are the individual class variances. The formulas for element calculation are defined in [16]. The threshold with minimum of weighted sum of variance is defined as:

$$\tau_{opt} = \arg\min_g\left(\sigma^2_\omega(g)\right) \tag{15}$$

The thresholding process is employed similar to (3) except replacing τ by τ_{opt} :

$$\mathcal{D}^*_i(x,y) = \begin{cases} 1 & ;\forall D^*_i(x,y) \geq \tau_{opt} \\ 0 & ;\forall D^*_i(x,y) < \tau_{opt} \end{cases} \tag{16}$$

The foreground sometimes consists of disconnected edges due to the drastic luminance change in dynamic scenes. In order to fuse narrow breaks and long thin gulfs, eliminate small holes, and fill gaps in the contour, some morphological operations [16] may be used during the post-processing.

III. Experimental Results and Discussion

A. Experimental Setup

Some parameters in the NIC algorithm need to be set up as default values, which comprises $\tau = 20$ of the constant threshold and (5×5) of the window size. All of the experiments are performed on the desktop PC operating Windows 7 with a 2.67 GHz Intel Core i5 CPU and 4GB RAM. MATLAB R2013a was the software for simulation.

In this paper, the foreground detection method is evaluated on several video sequences which are selected from public datasets consisting of the PETS 2009 [17] and PETS 2014 [18]. Video sequences including indoor and outdoor scene which represent typical situations in the video surveillance system and widely used in the object detection and tracking domain.

B. Evaluation Metrics

The proposed method is performed and compared with state-of-the-art methods in the performance of foreground detection. The accuracy analysis is undertaken with some quantitative measurements including $Recall$, $Precision$, $F1$, and $Similarity$ metric [19]. All metric values range from 0 to 1, with higher value pointing out the higher accuracy.

C. Object Detection Performance

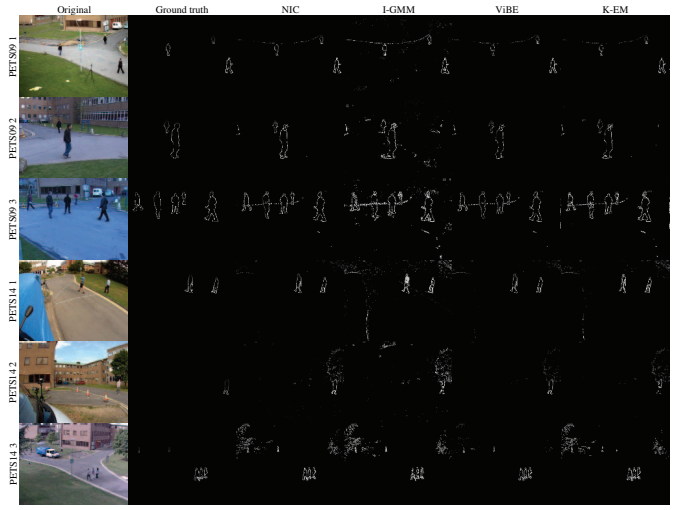

Fig. 3. Foreground detection results in visualization of of the proposed method and three state-of-the-art methods. Top to bottom in row: PETS09 1 (795 frames), PETS09 2 (795 frames), PETS09 3 (795 frames), PETS14 1 (1413 frames), PETS14 2 (1414 frames), and PETS14 3 (890 frames). Left to right in column: Original frame, Ground truth, NIC foreground, I-GMM foreground, ViBE foreground, and K-EM foreground.

TABLE I. QUANTITATIVE METRICS COMPARISON BETWEEN THE PROPOSED METHOD AND THE STATE-OF-THE-ART METHODS.

Method	PETS09 1				PETS09 2			
	Recall	Precision	F1	Similarity	Recall	Precision	F1	Similarity
NIC	0.947	0.953	0.950	0.905	0.871	0.917	0.894	0.808
I-GMM	0.981	0.644	0.778	0.636	0.893	0.773	0.829	0.707
ViBE	0.913	0.956	0.934	0.876	0.797	0.870	0.832	0.712
K-EM	0.912	0.917	0.915	0.843	0.865	0.842	0.853	0.744
Method	PETS09 3				PETS14 1			
	Recall	Precision	F1	Similarity	Recall	Precision	F1	Similarity
NIC	0.934	0.880	0.906	0.828	0.945	0.946	0.945	0.896
I-GMM	0.936	0.812	0.870	0.769	0.852	0.745	0.795	0.660
ViBE	0.943	0.956	0.950	0.904	0.947	0.917	0.932	0.872
K-EM	0.921	0.943	0.932	0.873	0.833	0.826	0.829	0.708
Method	PETS14 2				PETS14 3			
	Recall	Precision	F1	Similarity	Recall	Precision	F1	Similarity
NIC	0.956	0.614	0.748	0.597	0.895	0.796	0.842	0.728
I-GMM	0.675	0.417	0.515	0.347	0.580	0.535	0.557	0.386
ViBE	0.778	0.632	0.698	0.536	0.914	0.750	0.824	0.700
K-EM	0.579	0.656	0.615	0.444	0.895	0.756	0.845	0.731

The object segmentation results of three PETS 2009 sequences (denoted as PETS09 1, 2, and 3) are represented in the first three rows of Fig. 3. Although objects in sequences are successfully detected, the accuracies of them are different due to some objective reasons and the nature of background subtraction technique. For example, the motions of ribbon (in PETS09 1 and 4) due to the wind negatively affect to the accuracy. The object shape is sometimes wrongly segmented because of the intensity analogy between the background and the object pixel (in PETS09 4). Another objective challenge is further listed here is the overall chroma when the videos are captured in different parts of the day. Three sequences from PETS 2014 (denoted as PETS14 1, 2, and 3) also explain the walking activity with medium crowd in a vehicle parking area. The segmented foregrounds corresponding to three sequences are shown in last three rows of Fig. 3. A problem found in these videos is the perspective projection, in which the objects are smaller in their distance from the observer increases. Therefore, it is difficult to detect and segment objects along so far the line of sight. Furthermore, the foreground quality is also degraded by constantly unexpected motions of plants in the scene. The system in some cases can be mistaken whenever objects are considered as noise and removed out of a foreground.

The $Recall$, $Precision$, $F1$, and $Similarity$ results corresponding to considered sequences are reported in Table. I. The evaluation metrics are calculated thanks to the ground truth with the binary classification. The proposed method performs quite well on the most sequences. However, more background pixels are erroneously detected and classified leading to the high value of term fp. Although obtaining the high $Recall$ result, the $Precision$, $F1$, and $Similarity$ values can be still diminished. This tendency is sometimes appeared in the results of the PETS14 2 and 3.

D. Comparison and Discussion

In this section, the proposed method is compared with three state-of-the-art methods: the improved adaptive GMM (denoted as I-GMM) in [4], the original ViBE in [13], and the spherical K-means Expectation-Maximization method (denoted as K-EM) in [8]. For the I-GMM algorithm, the authors implemented with the the number of components $M = 4$. The radius of the sphere $R = 20$, the time subsampling factor $\phi = 16$, the cardinality of the set intersection $\#_{\min} = 2$, and the number of distances $N = 20$ as the default values proposed in the ViBE algorithm are utilized for testing in this paper. Finally, the K-EM algorithm is evaluated with following parameters comprising $K_{\max} = 5$, $\eta = 0.005$, $d = 2$, $\beta = 10^{-6}$, $T \in [2, 5]$, $TH_P = 0.1$, $TH_I = 70$, $TH_D = 18$, and $\tau = 0.5$ for the outdoor case. The qualitative and quantitative comparison are readily represented in Fig. 3 and Table. I, respectively. In the I-GMM method, parameters are constantly updated and the appropriate number of components for each pixel are simultaneously selected by using recursive equations. This method is so weak under the strong light change in the CAVIAR dataset and the dynamic movements in the PETS 2014 dataset. It is not evident to observe the objects from the result of I-GMM in the sequence PETS14 3. Although getting high $Recall$ results in some testing sequences as CAVIARs (over 90%), the $Precision$ metric values of I-GMM are too low (from 30−50 %) because more detected object pixels are wrongly classified. The number of false positive pixels fp is considerably greater than the number of true positive pixels tp ($fp = 2tp$ in the case of CAVIAR 2), hence, it leads to reduce not only $Precision$ but also $F1$ and $Similarily$ value significantly. This fact is appropriate with the visual results in Fig. 3 with more artifacts detected as foreground pixels. The ViBE method utilizes the random neighbor selection to correct pixel intensity and the lifespan policy to update model over time in the background modeling. The background estimated

by ViBE is robust to be against noise when compared to GMM-based models. As a combination of the spherical K-means clustering and the expectation-maximization algorithm based on updating GMM in the effort to against illumination changes, K-EM is efficient in shadow removal challenge. Compared with I-GMM, ViBE and K-EM therefore show the higher performance of evaluation metrics in most of sequences. Similar to ViBE in the use of the neighbor information, NIC processes more preferable than ViBE in the background estimation due to utilizing neighbor pixels in a appropriate window instead of random pixels. Nevertheless, the accuracy is slightly improved (approximately 5% in average of $Similarity$ for all sequences). In the competition with I-GMM, NIC wins at all of the benchmarked datasets with considerably high accuracy (over 20% in average of $Similarity$).

TABLE II. COMPARISON OF AVERAGE DETECTION TIME (MS/FRAME)

Sequence	Resolution	NIC	I-GMM	ViBE	K-EM
PETS09 1	768×576	116	148	108	125
PETS09 2	768×576	123	151	134	121
PETS09 3	768×576	141	167	115	119
PETS14 1	1280×960	298	389	221	299
PETS14 2	1280×960	363	402	293	357
PETS14 3	1280×960	267	391	209	218

E. Computational Measurement

This section analyzes and compares the computational cost of the proposed method and others through the term of average detection time (ms/frame). Concretely, the invested time for the background estimation and foreground detection is computed through a profiling tool included in the MATLAB 2013a. The average times of testing sequences are listed in Table. II. It can be seen that a larger size frame generally requires more time for processing. The proposed method detects the foreground faster than I-GMM in the most of sequences and and achieve an equivalent speed of K-EM. Compared with ViBE, NIC requires more time than ViBE for computing the standard deviation instead of the Euclidean distance. Although the proposed method is not the fastest one, it exhibits the highest detection performance under dynamic illumination environments.

IV. CONCLUSIONS

In this work, the efficient background subtraction is presented based the Neighbor-based Intensity Correction (NIC) algorithm to improve the foreground detection accuracy through estimating a robust background. The main contribution is the use of standard deviation, calculated from neighboring pixel blocks, to identify the center pixel belongs to the background or object for intensity correction. Compared with existing background modeling algorithms, NIC is more flexible and adaptive with medium and high speed motion. In the detection phase, the background generated by NIC is then provided to background subtraction scheme to extract the foreground with an Otsu threshold. Compared with the state-of-the-art methods comprising I-GMM,ViBE, and K-EM, the proposed method outperforms in most of testing datasets from $5-20\%$ of $Similarity$. However, the limitations are the poor detection performance in repetitive movements as noise in a background and high computational cost in the estimation stage. In the future, we continuously focus on the unexpected background motion identification and the computational optimization issue.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0101-15-1282-00010002, Suspicious pedestrian tracking using multiple fixed cameras). This work was also supported by the Industrial Core Technology Development Program, funded by the Korean Ministry of Trade, Industry and Energy (MOTIE), under grant number #10049079.

REFERENCES

[1] N. A. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, "A background subtraction algorithm for detecting and tracking vehicles," *Expert Syst. Appl.*, vol. 38, pp. 1619–1631, Mar 2011.

[2] H. Zhou, Y. Chen, and R. Feng, "A novel background subtraction method based on color invariants," *Comput. Vis. Image Und.*, vol. 117, no. 11, pp. 1589–1597, Nov 2013.

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

[4] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," *Proc. IEEE Int. Conf. Pattern Recognition (ICPR)*, vol. 2, pp. 28–31, Aug 2004.

[5] D.-S. Lee, "Effective gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[6] Z. Wang, H. Xu, L. Sun, and S. Yang, "Background subtraction in dynamic scenes with adaptive spatial fusing," *Proc. IEEE Int. Workshop Multimedia Signal Processing (MMSP)*, pp. 1–6, Oct 2009.

[7] A. Elqursh and A. Elgammal, "Online moving camera background subtraction," *Proc. European Conf. Computer Vision (ECCV)*, vol. 4, pp. 228–241, 2012.

[8] D. Li, L. Xu, and E. D. Goodman, "Illumination-robust foreground detection in a video surveillance system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1637–1650, Oct 2013.

[9] T. S. F. Haines and T. Xiang, "Background subtraction with dirichlet process mixture models," *IEEE Trans. Image Process.*, vol. 36, no. 7, pp. 670–683, Apr 2014.

[10] J.-M. Guo, C.-H. Hsia, Y.-F. Liu, M.-H. Shih, C.-H. Chang, and J.-Y. Wu, "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.

[11] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," *Proc. European Conf. Computer Vision (ECCV)*, vol. 2, pp. 751–767, 2012.

[12] Z. Liu, K. Huang, and T. Tan, "Foreground object detection using top-down information based on em framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 4204–4217, Sep 2012.

[13] O. Barnich and M. V. Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun 2011.

[14] S. Gruenwedel, N. I. Petrovic, L. Jovanov, A. J. O. Nino-Casta-neda, and W. Philip, "Efficient foreground detection for real-time surveillance applications," *IET Electron. Lett.*, vol. 49, no. 18, pp. 1143–1145, Aug 2013.

[15] B.-H. Do and S.-C. Huang, "Dynamic background modeling based on radial basis function neural networks for moving object detection," *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 1–4, Jul 2011.

[16] R. C. Gonzalez and R. E. Woods, "Digital image processing," *3rd Edition, Prentice Hall*, 2007.

[17] PETS 2009 Benchmark Data. [Online]. Available: http://ftp.pets.rdg.ac.uk/pub/PETS2009/

[18] PETS 2014 Benchmark Data. [Online]. Available: http://ftp.pets.rdg.ac.uk/pub/PETS2014/

[19] 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.