IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 2, 2018, accepted September 23, 2018, date of publication October 1, 2018, date of current version October 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872684

Real-Time People Counting System for Customer Movement Analysis

SUNG IN CHO[1], (Member, IEEE), AND SUK-JU KANG[2], (Member, IEEE)
[1]Department of Electronics and Electrical Engineering, Daegu University, Gyeongsan 38453, South Korea
[2]Department of Electrical Engineering, Sogang University, Seoul 121-742, South Korea

Corresponding author: Suk-Ju Kang (sjkang@sogang.ac.kr)

This work was supported by the Daegu University Research under Grant 20170311.

ABSTRACT We propose a real-time people-counting system that can be applied in a retail store to estimate the number of people entering and exiting. The proposed method consists of three main procedures: foreground extraction based on the average picture level (APL), a dilated motion search based on the maximum a posteriori probability (MAP), and flow analysis based on multiple touching sections (MTS). We first produce a background model to extract the foreground by using line-wise APL. A dilated motion search with the MAP-based approach is then used to estimate the motions on the line of interest. Next, the flow generated by the foreground on the MTS is analyzed. Finally, the results of the motion estimation and flow analysis are incorporated to produce the number of people entering and exiting the store. We used a low-cost microcomputer to implement the system, which is capable of wireless transmission and is easy to install in a retail environment. Experimental results show that the proposed method provided the best F1 score and accuracy values for the people count results with much lower computational complexity than benchmark methods. In addition, it successfully estimated the number of people entering and exiting the store in real time.

INDEX TERMS Flow analysis, MAP-based motion search, LOI-based people counting, microcontroller.

I. INTRODUCTION

The goal of people counting is to estimate the number of people entering and exiting a specific area. People counting can be used for various practical applications, such as video surveillance, urban planning, resource management, and customer profiling. Information from people counting is very useful and highly desirable for retail shops or restaurants because it can be used for real-time profiling analysis of customer visit patterns, which could enable efficient management and target marketing. In addition, with the recent expansion of on-line to off-line (O2O) services [1], the profiling results of customer visit patterns can be very important information for online and offline customer matching.

The management of a small retail store is very sensitive to changes in customer visit patterns. Therefore, analyzing these patterns could have a significant impact on efficient offline store management. Furthermore, regional commercial analysis can be done using the patterns obtained from a number of small retail stores in a particular area, as shown in Fig. 1. Therefore, there is a need for a low-cost system that is easy to install and capable of wireless transmission while accurately deriving statistical information for the number of people in retail stores.

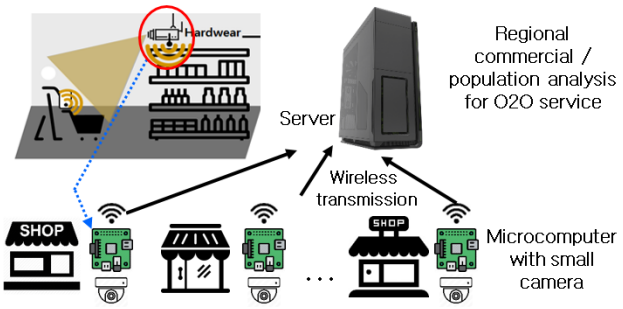

FIGURE 1. Vision-based people counting system with network structure for profiling analysis of customer visit patterns, which enables O2O service.

Various approaches have been proposed for people counting [2]–[10]. Most approaches generally use a framework based on a region of interest (ROI) or line of interest (LOI). ROI-based frameworks estimate the number of people in the ROI, whereas LOI-based frameworks estimate the number of people crossing the LOI, as shown in Fig. 2.

People-counting methods are broadly divided into methods based on detection with tracking [2]–[4] and methods based on segmentation with regression [5]–[10].

FIGURE 2. Example of people counting approach based on the line of interest (red solid line). (a) Frame from the Grand Central dataset [6]. (b) Frame from the University of California, San Diego (UCSD) dataset [2].

Tracking-based methods detect people's locations and track them for counting. One method extracts people locations and tracks them with unsupervised Bayesian clustering to detect individual entities [2]. In another method, the foreground is first extracted, and people's heads are detected using the 2-D correlation between a bank of annular patterns and the extracted foreground regions [4]. The detected heads are tracked using a Kalman filter, which is widely used for tracking systems [11]–[13], and the people count is estimated using an LOI-based approach.

However, these detection with tracking-based methods are very vulnerable to changes in light and occlusion, where a number of people overlap each other in the image or appear and disappear. Thus, the accuracy of people counting can be significantly degraded in such cases. Moreover, the methods generally require high computational complexity for the detection and tracking.

Methods based on segmentation with regression generally use foreground and motion information for crowd segmentation, which is the most fundamental feature for the estimation of a number of people. Flow mosaicking [5] is a representative motion-based approach for people counting. In this method, the motion vectors (MVs) on a selected LOI are extracted. A temporal slice image (TSI) is then generated by stacking LOI pixels over time. The moving segment is stacked to produce a blob, and the size of each segment stacking is determined by its velocity. The final cumulative people count is estimated by a regression function with the volume of the extracted blob.

In another approach [6], the crowd is segmented into subcomponents of distinct motion flow, and then a low-level feature is extracted from each segment. The number of people is calculated from the extracted feature and Bayesian Poisson regression (BPR). In Ma and Chan's method [7], a TSI is generated by stacking LOIs over time as in flow mosaicking [5], but a fixed line width is used for stacking, whereas flow mosaicking uses the velocity of a moving segment as the size of the line stacking. Next, the number of people is estimated in a set of overlapping sliding windows called the temporal ROI (TROI). People counts in each TROI are estimated using BPR with a local histogram of oriented gradients (HOG) and various features. In addition, per-pixel temporal normalization is used to improve the accuracy of people counting for a large blob. These approaches can improve the accuracy of

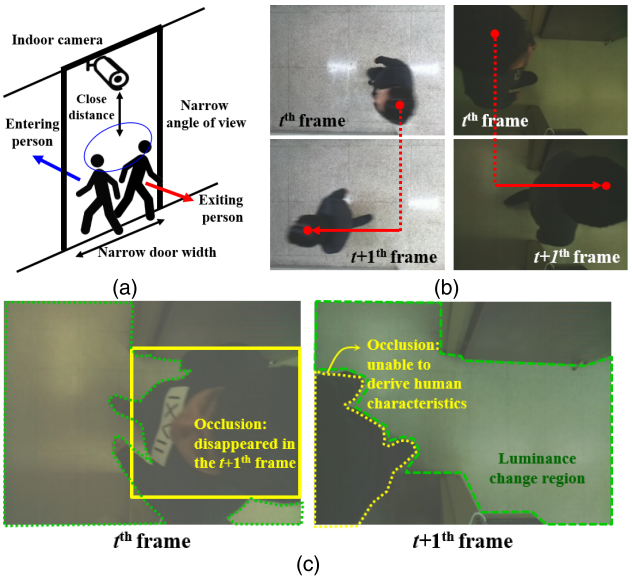

FIGURE 3. A practical environment for estimating the number of people entering and leaving a retail store. (a) Configuration example of camera installation at a practical retail store. (b) Example of large motion of two consecutive frames. (c) Occlusion and luminance change in a real environment of people counting.

instantaneous counts. This method was later improved upon by using a multiple-window-length TROI and outlier-robust objective function for people counting [8].

Methods based on segmentation with regression generally provide the more accurate results of people counting than methods based on detection with tracking. These methods successfully estimate the number of people in outdoor test sets where the distance between the people and the camera is large enough, as shown in Fig. 2 (a), as well as in large indoor spaces, as shown in Fig. 2 (b).

However, these methods have some drawbacks for indoor environments, such as small retail stores or restaurants. As shown in Fig. 3 (a), the camera and the people are very close to each other in these environments, in contrast to other datasets for people counting shown in Figs. 2(a) and (b). Thus, the motion of a person is generally much larger, as shown in Fig. 3 (b), and luminance variation and occlusion occur very frequently, as shown in Fig. 3 (c). As a result, the accuracy of detecting people, motion, and foreground can be significantly degraded. In addition, other methods have very high computational complexity to operate in real time on a low-cost device with poor computing resource that can be installed in a real retail store.

We propose a real-time people counting method for use on low-cost and easy-to-install devices in a real small retail store. We assume that a camera is installed on the ceiling of a doorway over the heads of visitors and that the frame rate of input images is very low (less than four frames per second). Thus, input images would generally include a smaller field of view and larger people motions than in existing datasets. The proposed method uses flow volume analysis based on

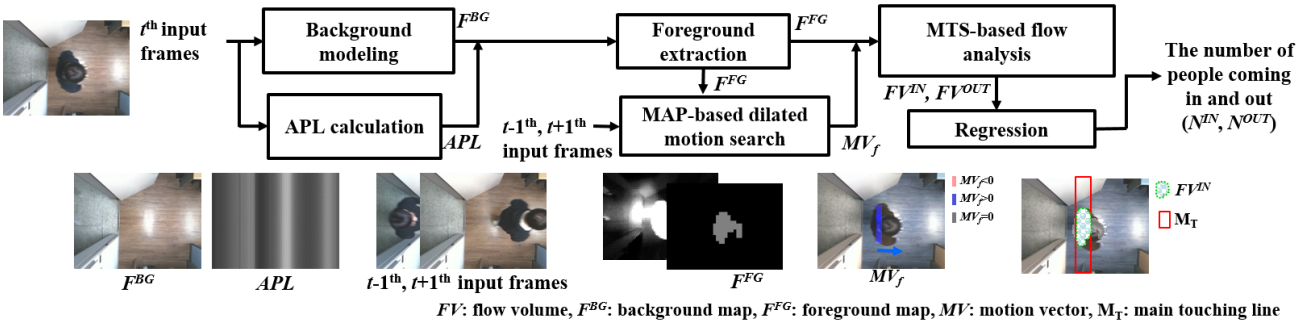

FIGURE 4. The overall architecture of the proposed method.

multiple touching sections (MTS) and dilated motion estimation based on the maximum a posteriori probability (MAP) to improve the accuracy of people counting while minimizing the computational complexity. In addition, we present an easy to install and the low-cost hardware system for people counting by using the microcomputer with the small hardware resource.

The main contributions of this work are summarized as follows:

1) We propose a line-wise background modeling and foreground extraction, which utilizes the line-wise average picture luminance (APL). This method successfully extracts foreground under the various illumination environments.
2) A dilated motion search that utilizes the foreground information as the prior knowledge and a multiple touching section (MTS)-based flow volume analysis are proposed to provide the higher accuracy of people counting with much simpler operations, compared with conventional methods.
3) The proposed method is successfully implemented in a general-purpose and low-cost microcomputer, Raspberry Pi, in real time as a test platform.

II. PROPOSED METHOD

A. OVERVIEW

Fig. 4 shows the overall architecture of the proposed people counting method. The background is first modeled to extract the foreground. Then, a MAP-based dilated motion search is conducted on the down-sampled input frame. The volume and direction of moving flow are estimated using three pre-defined touching sections, and then the count and moving direction of people are determined by incorporating the results of the flow analysis and the dilated motion search. The proposed method is implemented using a Raspberry Pi microcomputer as a test-bed, which is widely used for the implementation of various real-time systems related to image processing [14], [15]. In the following explanations, we assume that the columns of the image are the LOIs. When the rows are the LOIs, as shown in Fig. 2 (b), all the explanations can be understood by rotating the image by 90 degrees.

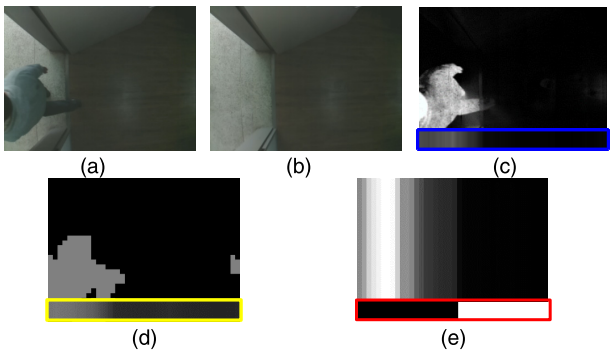

FIGURE 5. Example of F^{FG} generation. (a) Current input frame t. (b) F^{BG}. (c) Δ_M with AD (box with blue line). (d) F^{FG} with a line APL (box with yellow line). (e) Δ_L with the mask (box with red line) for line-wise semi-global F^{BG} update (only the white area is updated).

B. BACKGROUND MODELING FOR FOREGROUND EXTRACTION

Various approaches for background modeling have been used in people counting methods [16]–[18]. Existing methods provide excellent results with people-counting datasets [3], [7], [19], but they have high computational cost and are vulnerable to variations in illumination. To alleviate these problems, we propose a simple approach for foreground extraction that considers the APL of a given frame. As shown in Fig. 5, the foreground is derived based on the difference Δ_M between the input frame and the background model:

$$F^{FG}_{x,y,t} = \begin{cases} 1, & \Delta_M(x,y) > \alpha \cdot APL^2(x) + \beta \cdot AD(x) \\ 0, & otherwise \end{cases}$$
$$\Delta_M(x,y) = \left| F_t(x,y) - F^{BG}_{x,y,t} \right|, \quad (1)$$

where $F^{FG}_{x,y,t}$ and $F^{BG}_{x,y,t}$ are the foreground map and background map of a pixel (x, y) at time t, respectively. F_t is the current input frame. $APL(x)$ and $AD(x)$ are the APL and average Δ_M of column x, respectively. α and β are mixing parameters for $APL(x)$ and $AD(x)$.

The foreground (Fig. 5 (d)) is extracted by thresholding Δ_M (Fig. 5 (c)), which is the most common approach for foreground extraction [20]–[22]. Generally, it is difficult for the foreground objects described by Δ_M to appear clearly in an image with low illumination. Therefore, the APL of a given

input frame is considered when thresholding Δ_M. In addition to APL, $AD(x)$ can provide the average strength of Δ_M in column x of a given input frame and is considered to reflect the extent to which the foreground is highlighted from the background in various environments. When the APL and AD are derived, the global and local information can be considered together by using line-wise calculation. α and β were set to 0.5 and 0.2, respectively based on extensive experiments using test sets with various illumination conditions.

At the system startup, F^{BG} in (1) is set to the first input frame. Then, F^{BG} is updated globally and semi-globally. The global update is performed by averaging F^{BG} and the current input frame only when the number of pixels with $F^{FG} = 1(N_{FG})$ in the previous input frame ($F^{FG}_{x,y,t-1}$) is less than 0.1% of the total number of pixels of the current input frame:

$$F^{BG}_{x,y,t+1} = \begin{cases} \left(F^{BG}_{x,y,t} + F_t(x,y)\right)/2, & if\ N_{FG} < N_R \cdot N_C \cdot 0.001 \\ F^{BG}_{x,y,t}, & otherwise, \end{cases} \tag{2}$$

where N_C and N_R are the number of rows and columns of a given input frame. N_{FG} is the total number of foreground pixels. In the case of the global update, it cannot properly reflect background changes caused by the temporal changes in illumination. Also, errors can occur if the foreground stops moving. To alleviate these problems, we use line-wise updates based on the average difference ($\Delta_L(x)$) of vertical lines of consecutive input frames (F_{t-1} and F_t):

$$F^{BG}_{x,t+1} = \begin{cases} \left(F^{BG}_{x} + F_t(x)\right)/2, & if\ \Delta_L(x) < 1 \\ F^{BG}_{x,t}, & otherwise, \end{cases}$$

$$\Delta_L(x) = \frac{1}{N_R}\sum_{y=1}^{N_R} |F_t(x,y) - F_{t-1}(x,y)|. \tag{3}$$

By using Δ_L in rather than Δ_M, as shown in Fig. 5 (e), we can accurately extract the foreground even in the event of background changes caused by temporal illumination variation and when the foreground stops moving. In addition, this line-wise update is robust to image noise components and changes in local illumination. Fig. 5 (d) shows the final result of the foreground extraction. The foreground is scaled down in accordance with MV search that will be explained the later section. Details related to this scaling down will be described in the following section.

C. MAP-BASED DILATED MOTION SEARCH

After the foreground extraction, simplified motion extraction is performed, which enables real-time operation on low-cost hardware. The proposed motion search uses three consecutive frames: F_{t-1}, F_t, and F_{t+1}, as shown in Fig. 6 (a). This is done to help explain the algorithm. For the real implementation, F_{t-2}, F_{t-1}, and F_t are used. MV is estimated using the following MAP-based framework:

$$P\left(MV_C \left| F^L_t, F^L_{t-1}, F^L_{t+1}\right.\right) = \frac{P\left(F^L_t, F^L_{t-1}, F^L_{t+1} | MV_C\right) P(MV_C)}{P\left(F^L_t, F^L_{t-1}, F^L_{t+1}\right)}, \tag{4}$$

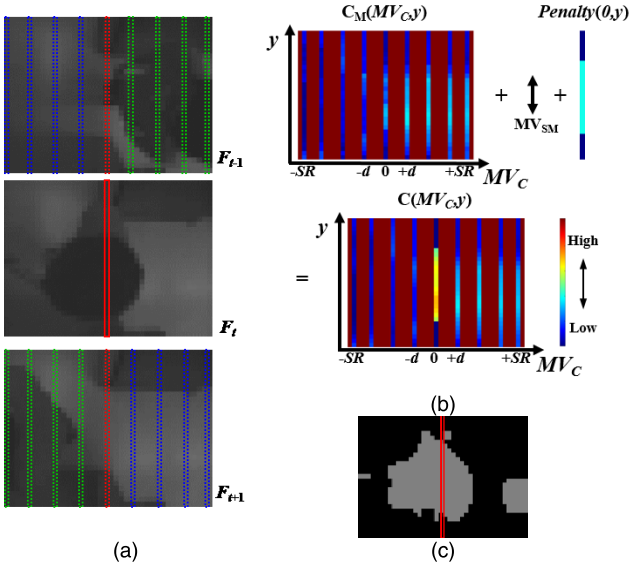

FIGURE 6. Motion extraction in the proposed method. (a) Input frames F_{t-1}, F_t, F_{t+1} and LOI_t (box with red solid line at t frame) with $F_{t-1}(x_{LOI}\text{-}MV_C)$ and $F_{t+1}(x_{LOI} + MV_C)$ for positive MV_C (box with blue dotted line), negative MV_C (box with green dotted line), and zero MV_C (box with red dotted line). (b) Matching cost calculation by the dilated search. (c) F^{FG}.

where P is the probability, and MV_C is a candidate MV. In (4), $P(F^L_t, F^L_{t-1}, F^L_{t+1})$ can be ignored because it does not change with respect to MV_C. Our MAP-based motion search finds a value of MV_f that maximizes the following a posteriori probability with respect to MV_C:

$$MV_f = \arg\max_{MV_C} P\left(F^L_t, F^L_{t-1}, F^L_{t+1} | MV_C\right) P(MV_C), \tag{5}$$

where $P(F^L_t, F^L_{t-1}, F^L_{t+1}|MV_C)$ is the likelihood of MV_C and is generally modeled as the matching cost based on the sum of the absolute difference (SAD) [23]. For $P(MV_C)$, which represents the prior knowledge about the motion search, we use the smoothness and foreground information (F^{FG}). The matching cost vector $\mathbf{C_M}(MV_C, y)$ is shown in Fig. 6(b) for MV_C displacements from the pre-defined LOI at the y-th pixel in the current t^{th} frame. The vector is calculated as follows:

$$\begin{aligned} \mathbf{C_M}(MV_C, y) = &\sum_{MV_C=-SR}^{SR} |\mathbf{LOI_t}(y) - \mathbf{F_{t-1}}(x_{LOI} - MV_C, y)| \\ &+ \sum_{MV_C=-SR}^{SR} |\mathbf{LOI_t}(y) - \mathbf{F_{t+1}} \\ &\times (x_{LOI} + MV_C, y)|, \\ \mathbf{LOI_t} = &[F_t(x_{LOI}, yl), F_t(x_{LOI}, yl+1), \\ &\ldots, F_t(x_{LOI}, yh)]^T, \\ \mathbf{F_{t-1}}(x) = &[F_{t-1}(x, yl), F_{t-1}(x, yl+1), \\ &\ldots, F_{t-1}(x, yh)]^T, \end{aligned} \tag{6}$$

where $F_t(x, y)$ is the pixel at (x, y) in the current t^{th} input frame. $\mathbf{F_{t-1}}(x)$ and $\mathbf{F_{t+1}}(x)$ are the column vectors of the

$(t-1)^{\text{th}}$ and $(t+1)^{\text{th}}$ input frames at column x. $\mathbf{LOI_t}$ is the column vector of the selected LOI in the current t^{th} input frame. x_{LOI} is the column position of the pre-defined LOI. yl and yh are the lower and upper coordinates of the pre-defined LOI.

Fig. 6 (a) shows the configuration of the terms used in (6). SR is the maximum search range along the x-axis and is set to half the width of an input frame. As shown in (6), line-wise bilateral motion search is used with the bi-directional SADs calculated from $\mathbf{F_{t-1}}(x)$, $\mathbf{F_{t+1}}(x)$, and $\mathbf{LOI_t}$. To reduce the computational complexity of the MV search, dilated search is used by setting MV_C as follows:

$$MV_C \in [-SR, \ldots -2\times d, -d, 0, +d, +2\times d, \ldots, +SR], \tag{7}$$

where d is the interval of dilated search and can be set chosen by considering the width of an input frame and the computational complexity of the motion search. We set it to 5 pixels, which allows real-time operation in our target system.

After calculating $\mathbf{C_M}$, the MV smoothness (MV_{SM}) between adjacent pixels is used for the modeling of prior knowledge $P(MV_C)$ by sharing their $\mathbf{C_M}$ values. In addition, F^{FG} is used for the penalty assignment. For the foreground pixels ($F^{FG}_{x,y,t}=1$) on the LOI shown in Fig. 6 (c), a positive penalty value (κ) is added to the matching cost of the zero MV ($MV_C=0$). For the background pixels ($F^{FG}_{x,y,t}=0$) on the LOI, a negative κ is added to the matching cost of the zero MV. Fig. 6 (b) shows the final matching cost $\mathbf{C}(MV_C, y)$ for MV_C at the y-th pixel of the LOI. This cost reflects $P(MV_C)$ and is calculated as follows:

$$\begin{aligned}\mathbf{C}(MV_C, y) = {} & \mathbf{C_M}(MV_C, y) + Penalty(MV_C, y) \\ & + \mathbf{C_M}(MV_C, y-1) + \mathbf{C_M}(MV_C, y+1). \end{aligned} \tag{8}$$

MV_C at the y-th pixel on the LOI that provides the lowest $\mathbf{C}$ is selected as the final MV ($MV_f(y)$). The penalty value κ was empirically set to 100. The confidence of the selected $MV_f(y)$ ($MV_{conf}(y)$) is estimated by its lowest $\mathbf{C}(MV_f, y)$ ($C_L(y)$) and the size of the MV cluster as follows:

$$MV_{conf}(y) = \begin{cases} 0, & if\ C_L(y) > TH_C\ or\ N_{cluster}(MV_f, y) = 0 \\ 1, & otherwise, \end{cases} \tag{9}$$

where N_R is the number of columns of a given input frame. $N_{cluster}(MV_f, y)$ is the number of adjacent MV_fs ($MV_f(y-1)$ and $MV_f(y+1)$) with the same direction as the current $MV_f(y)$. $MV_f(y)$ is ignored for the determination of people movements if $C_L(y)$ is higher than TH_C or $N_{cluster}(MV_f, , y)$ is equal to zero, which means that the y-th pixel of the LOI in the current frame does not have a similar pixel to itself in F_{t-1} and F_{t+1} or $MV_f(y)$ is separated by itself. C_L generally increases in proportion to the luminance of an input frame. Thus, TH_C for C_L is determined according to the overall APL(APL/4) of a given image.

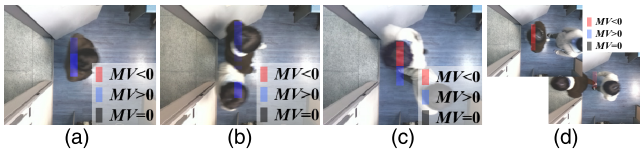

FIGURE 7. Example results of MV_f for the various situations: (a) single entry, (b) multiple entries, (c) crossing entries, and (d) consecutive entries.

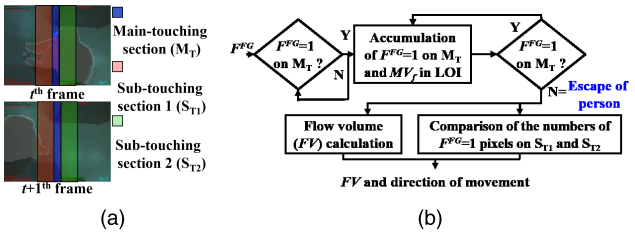

FIGURE 8. Flow analysis: (a) configuration of multiple touching sections (MTS) and (b) the procedures of the flow analysis.

To reduce the computational complexity and influence of image noise, low-pass filtering (LPF) and down sampling (1/8) are applied to three consecutive frames. In the real implementation, F_{t-2} and F_{t-1} can be re-used, so LPF and down sampling are applied only once to F_t. The down-scaling of input frames enables the exploitation of structural information of the input frame during the extraction of the matching cost. This down-scaling is also applied to the foreground extraction in accordance with the MV extraction. Fig. 7 shows the MV_f results for various cases of people entries after using the down-scaling process.

D. MULTIPLE TOUCHING SECTIONS (MTS)-BASED FLOW ANALYSIS

In addition to the MV search, we use MTS-based flow analysis to estimate the direction and size of the foreground movement, as shown in Fig. 8 (a). The main touching section (M_T) involves a pre-defined LOI. In addition to M_T, two sub-touching sections (S_Ts) are located next to M_T. The target of the proposed flow analysis is to derive the number of foreground pixels ($F^{FG}=1$) generated in M_T. The flow volume (FV) is derived by accumulating the number of foreground pixels in M_T (N^{FG}_{MT}) from the time when the foreground pixels appear in M_T until they are no longer present in the region, as shown in Fig. 8 (b). By using this approach, it is possible to estimate the people counts generated by multiple or consecutive entries and exits.

The final inward FV (FV^{IN}_t) and outward FV (FV^{OUT}_t) in the current frame are derived by incorporating the information of movement direction that has already been estimated by the MAP-based dilated MV search as follows:

$$\begin{aligned} FV^{IN}_t &= \sum_{x,y\in\mathbf{M_T}} \left(F^{FG}_{x,y,t}\right) \times \left(N^{MV}_{IN} / \left(N^{MV}_{IN} + N^{MV}_{OUT}\right)\right), \\ FV^{OUT}_t &= \sum_{x,y\in\mathbf{M_T}} \left(F^{FG}_{x,y,t}\right) \times \left(N^{MV}_{OUT} / \left(N^{MV}_{IN} + N^{MV}_{OUT}\right)\right), \end{aligned} \tag{10}$$

where N_{IN}^{MV} and N_{OUT}^{MV} are the numbers of accumulated MV_fs indicating the inward and outward directions, respectively. There can be cases where it is impossible to derive the direction of the flow by the MAP-based dilated motion search because the confidences of all extracted MV_fs in (9) are zero due to severe occlusion or abrupt illumination change. In such cases, we use the flow direction derived from an auxiliary approach that estimates the flow direction by considering the distribution of foreground pixels in S_{T1} and S_{T2} at the moment when the foreground pixels escape M_T.

For example, in the $t+1^{\text{th}}$ frame shown in Fig. 8(a), when the foreground pixels escape M_T, the number of foreground pixels in S_{T1} (N_{ST1}^{FG}) is higher than the number of foreground pixels in S_{T2}(N_{ST2}^{FG}). For this case, the flow direction is determined to the left (out), and in the opposite case, the flow direction is determined to the right (in). This approach is very simple, but it can cause errors for complex movements of flow. Therefore, it is used only when there is no derived MV information from the MAP-based dilated MV search. This MTS-based analysis can be a very effective way of deriving the movement direction since the movements of visitors are not generally complicated in the target retail store.

The *FV*-based method for the estimation of total people counts can have lower instantaneous accuracy than conventional methods that use the results of complex motion extraction and various human characteristics with per-person regression [3]–[8]. This occurs because there are differences in the moving speed and size of individual people, which affect the results of the *FV* estimation. However, in the case of an actual retail store, people counts for a certain period of time are required for profiling customer visit patterns rather than the instantaneous people count. Since the differences in moving speed and size for each person are generally random, they can be offset in the estimation results of people count for a certain period of time. Furthermore, to increase the accuracy of people counts for a certain period of time, the moving speed and size for each person are considered when the final people counts are determined by using the proposed regression approach that utilizes the moving speed and size of each person and is explained in the next section.

The *FV*-based method can provide robust accuracy in people counting for even severe occlusion or illumination changes. In contrast, conventional methods can cause severe errors in the extraction of MV or human characteristics due to severe occlusion or illumination changes, which can frequently occur in real environments, as shown in Fig. 3(c). Consequently, the proposed *FV*-based approach is more suitable for such applications.

E. REGRESSION FOR FINAL PEOPLE COUNT

The *FV* needs to be normalized according to the distance between the camera and the person. The average *FV* per person (FV_{avg}^{N}) is derived from the average value of the N minimum *FV*s for a certain period of time. The extracted FV_{avg}^{N} is regarded as the *FV* generated by one person's movement and is used for the normalization of *FV*. In a real system, training for deriving FV_{avg}^{N} can be conducted by performing several movements of one person after starting the system. In the experiments, 10 minimum *FV*s are gathered and averaged except for the smallest and the largest *FV*s to produce FV_{avg}^{N} for each dataset. As a result, the normalization of flow is performed separately for each dataset as follows:

$$N_{EST}^{IN} = FV^{IN}/FV_{avg}^{N}, \quad N_{EST}^{OUT} = FV^{OUT}/FV_{avg}^{N}. \quad (11)$$

In addition to the normalization, multivariate linear regression for extracting the final people count is done using N_{EST}^{IN}, N_{EST}^{OUT}, FV_{avg}^{N}, and the average magnitude of MV_f for each dataset (MV_f^{MAG}):

$$N^{IN(OUT)} = w_1 \cdot N_{EST}^{IN(OUT)} + w_2 \cdot FV_{avg}^{N} + w_3 \cdot MV_f^{MAG} + B, \quad (12)$$

where w_1, w_2, w_3, and B are the weights and bias values for the multivariate linear regression. These can be obtained from off-line regression using a training dataset. N^{IN} and N^{OUT} are the final numbers of people entering and leaving the LOI for each dataset.

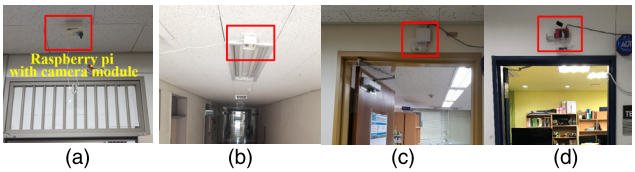

FIGURE 9. Installation of the proposed system located above a door (box with red solid line). (a) At Daegu Univ. (DU) 1. (b) At DU2. (c) At Sogang Univ. (SU) 1. (d) At SU2.

TABLE 1. Specifications of test sets.

Test sets	DU1	DU2	SU1	SU2	Total
Number of sub-sets	22	11	17	5	55
Number of frames	8324	4020	6610	2033	21017
Number of sub-sets for off-line regression	6	3	4	1	14
Number of frames for off-line regression	2295	1052	1581	413	5341

TABLE 2. Specifications of system for people counting.

Location	Height of camera	APL	Camera tilt angle	Image resolution	Frames per second (FPS)
DU1	2.57M	150	90 degree	640×480	4
DU2	2.60M	163			
SU1	2.17M	67			
SU2	2.15M	50			

III. EXPERIMENTAL RESULTS

For the experiments, we used four kinds of test sets that were obtained in four different environments, as shown in Fig. 9 and Table 1. Each test set consists of 5-22 sub-test sets which contain 14-30 incoming and outgoing people. The total number of sub-test sets is 55, and each sub-test set consists of 240-900 frames. Among the 55 sub-test sets, 14 sub-test sets (5341 frames) were used for off-line regression training, and the final experimental results

TABLE 3. In and out people counts with error of benchmark and proposed methods on test sets (E_{IN}: $|N^{IN}\text{-}N^{IN}_{GT}|$, E_{OUT}: $|N^{OUT}\text{-}N^{OUT}_{GT}|$, E: $E_{IN}+E_{OUT}$).

Test sets	Ground-truth		FM					DPC					CPC					Proposed				
	N^{IN}_{GT}	N^{OUT}_{GT}	N^{IN}	N^{OUT}	E_{IN}	E_{OUT}	E	N^{IN}	N^{OUT}	E_{IN}	E_{OUT}	E	N^{IN}	N^{OUT}	E_{IN}	E_{OUT}	E	N^{IN}	N^{OUT}	E_{IN}	E_{OUT}	E
SU1 (1~16)	271	272	253	250	18	22	40	233	217	38	55	93	289	288	18	16	34	277	274	6	2	8
SU2 (17~24)	129	127	130	137	1	10	11	139	130	10	3	13	110	112	19	15	34	136	140	7	13	20
DU1(25~37)	195	195	209	208	14	13	27	116	135	79	60	139	197	204	2	9	11	195	201	0	6	6
DU2 (38~41)	60	60	22	22	38	38	76	19	21	41	39	80	58	58	2	2	4	55	54	5	6	11
Total	655	654	614	617	71	83	154	507	503	168	157	325	654	662	41	42	83	677	681	18	27	45

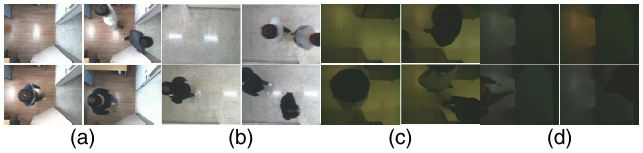

FIGURE 10. Example test images. (a) DU1. (b) DU2. (c) SU1. (d) SU2.

are described for the remaining 41 sub-test sets, as shown in Table 1.

Fig. 10 shows example test images from four places. The test images were obtained under various luminance conditions that are similar to the conditions in an actual retail store. The precision, recall, and F1 score were used for objective evaluation, along with an accuracy calculation from a previous study [5].

For the benchmark methods, we used flow mosaicking (FM) [5], directional people counter (DPC) [4], and counting people crossing a line (CPC) [8], which are the most popular methods for people counting. All of the benchmark methods were implemented using MATLAB. For CPC, motion segmentation was performed using C code that is distributed by the author, and the rest of the processing was implemented using MATLAB.

Adjustable parameters were optimized to produce the highest accuracy based on extensive experiments for fair comparison. Training or off-line regression of the benchmark methods was performed using the same conditions as the proposed method. The software of the proposed system was developed using Python 2 with OpenCV 3.2, which are supported by our target hardware system (a Raspberry Pi microcomputer).

A. SYSTEM CONFIGURATION OF PROPOSED PEOPLE COUNTING

A Raspberry Pi was used to implement the proposed method. Using the proposed people counting system, the estimated people counts can be used not only in the management of each retail store, but also in regional commercial analysis through wireless communication as shown in Fig. 1.

A Raspberry Pi camera module v2 was used for capturing input frames that were used as a test set. Fig. 9 shows the proposed system installed in four environments. The locations are similar environments to a retail store which is indoors and has a low ceiling and small door. Table 2 shows the specifications of the proposed system.

B. EVALUATION OF THE ACCURACY OF PEOPLE COUNTING

For the evaluation of the methods, the absolute difference between $N^{IN(OUT)}$ and the ground-truth number of people entering or leaving the LOI was used. In addition, the precision, recall, and F1 score (F1) were used for the evaluation:

$$\begin{aligned}\text{Precision} &= \frac{\min\left(N^{IN}, N^{IN}_{GT}\right)+\min\left(N^{OUT}, N^{OUT}_{GT}\right)}{N^{IN}+N^{OUT}},\\ \text{Recall} &= \frac{\min\left(N^{IN}, N^{IN}_{GT}\right)+\min\left(N^{OUT}, N^{OUT}_{GT}\right)}{N^{IN}_{GT}+N^{OUT}_{GT}},\\ \text{F1} &= 2\times((\text{Recall}\times\text{Precision})\,/\,(\text{Recall}+\text{Precision})),\end{aligned} \tag{13}$$

where N^{IN}_{GT} and N^{OUT}_{GT} are the ground-truth numbers of people entering and leaving LOI. Lastly, the accuracy that was used in a previous study [5] was also used for the evaluation:

$$\text{Accuracy} = 1-\frac{\left|N^{IN}-N^{IN}_{GT}\right|+\left|N^{OUT}-N^{OUT}_{GT}\right|}{N^{IN}_{GT}+N^{OUT}_{GT}}. \tag{14}$$

Table 3 shows the in and out people counts (N^{IN} and N^{OUT}) with E_{IN}, E_{OUT}, and E, which are $|N^{IN}\text{-}N^{IN}_{GT}|$, $|N^{OUT}\text{-}N^{OUT}_{GT}|$, and $E_{IN}+E_{OUT}$, respectively. The detailed results and errors for each sub-test set are shown in Figs. 11 (a) and (b). The left axis is E_{IN} (E_{OUT}), and the right axis is N^{IN}_{GT}(N^{OUT}_{GT}). Among the benchmark methods, DPC showed the highest E_{IN} and E_{OUT} values, whereas CPC showed the lowest values. Compared with CPC, the proposed method showed lower E_{IN} and E_{OUT} values for all test sets except for DU2, even though it has much simpler operations. The total E of the proposed method for all test sets was approximately half that of CPC.

Table 4 and Figs. 11 (c)-(f) show the precision, recall, F1 score, and accuracy values for the test sets. CPC and the proposed method generally provided higher average recall and precision values than the other benchmark methods. For the average F1 score, which incorporates the recall and precision values, the proposed method provided the best results. The proposed method also showed the best overall average accuracy values. For each test set, we compared the minimum values of the precision, recall, F1 score, and accuracy to evaluate the lower performance bound of each method, as shown in Table 4. The proposed method showed robust people counting performance by providing the highest minimum values of the various evaluation metrics.

C. EVALUATION OF COMPUTATIONAL COMPLEXITY

For the comparison of computational complexity, the processing time per frame (P_T) was compared using C, MATLAB, and Python on a PC with an Intel I7 7700 processor at

TABLE 4. Average (Avg.) Recall (Rec.), Precision (Pre.), F1 score (F1), and Accuracy (Acc.) with Minimum (Min.) F1 and Acc. of benchmark and proposed methods on test sets.

Test sets	FM						DPC						CPC						Proposed					
	Avg. Rec.	Avg. Pre.	Avg. F1	Avg. Acc.	Min F1	Min Acc.	Avg. Rec.	Avg. Pre.	Avg. F1	Avg. Acc.	Min F1	Min Acc.	Avg. Rec.	Avg. Pre.	Avg. F1	Avg. Acc.	Min F1	Min Acc.	Avg. Rec.	Avg. Pre.	Avg. F1	Avg. Acc.	Min F1	Min Acc.
SU1	0.92	0.93	0.91	0.85	0.60	0.43	0.92	0.95	0.93	0.87	0.80	0.67	0.98	0.91	0.94	0.87	0.88	0.73	0.98	0.96	0.97	0.93	0.89	0.80
SU2	0.97	0.93	0.95	0.90	0.90	0.80	1.00	0.85	0.91	0.81	0.87	0.70	0.87	1.00	0.93	0.87	0.82	0.70	0.99	0.92	0.95	0.91	0.93	0.84
DU1	0.98	0.92	0.95	0.90	0.93	0.87	0.76	0.98	0.84	0.74	0.60	0.43	0.99	0.97	0.98	0.96	0.92	0.83	0.99	0.98	0.98	0.96	0.97	0.93
DU2	0.37	1.00	0.54	0.37	0.50	0.33	0.47	1.00	0.63	0.47	0.57	0.40	0.97	1.00	0.98	0.97	0.97	0.93	0.91	1.00	0.95	0.91	0.89	0.80
Overall	**0.90**	**0.94**	**0.89**	**0.83**	-	-	**0.84**	**0.94**	**0.87**	**0.78**	-	-	**0.96**	**0.96**	**0.96**	**0.91**	-	-	**0.98**	**0.96**	**0.97**	**0.94**	-	-

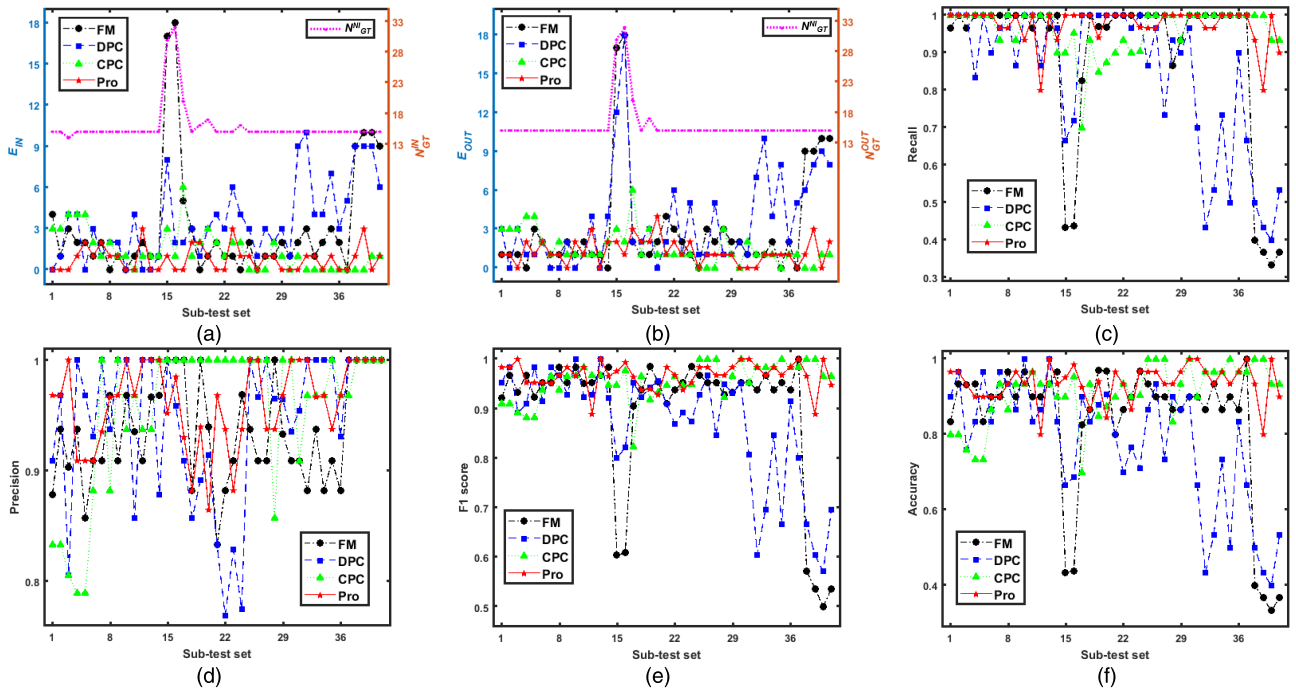

FIGURE 11. Comparisons of people counting results of the benchmark and the proposed method on all sub-test image sets. (a) E_{IN} $|N^{IN}\text{-}N^{IN}_{GT}|$ with N^{IN}_{GT} (left axis for E_{IN} and right axis for N^{IN}_{GT}). (b) E_{OUT} $|N^{IN}\text{-}N^{OUT}_{GT}|$. (c) Recall. (d) Precision. (e) F1 score. (f) Accuracy.

TABLE 5. Average processing time per frame (P_T).

Methods	FM on PC	DPC on PC	CPC on PC	Proposed method on PC	Proposed method on raspberry pi
Avg P_T (sec)	0.23	0.08	0.79	0.05	0.26

3.60 GHz with of 16 GB DDR3 RAM. Depending on the coding style, the processing time of the program can change, but in general, a MATAB-based program is faster than a Python-based program for the same operations. However, the Raspberry Pi only supports Python, so the proposed method was implemented in Python. Nevertheless, as shown in Table 5, the proposed method had a very small P_T on a PC compared to the benchmark methods. The proposed method required only 1/18 of the P_T on a PC compared with a CPC, which has the best accuracy among the benchmark methods. DPC showed a slightly higher P_T than the proposed method, but its accuracy was much lower.

The Raspberry Pi is a microcomputer with poor computing power compared to a PC. The proposed method required only 0.26 seconds for each frame on the Raspberry Pi. Hence, the proposed people-counting system can operate in an environment of 3-4 fps. The main bottleneck for increasing the fps is the capture rate of the camera module with Raspberry Pi.

IV. CONCLUSION

We proposed a novel approach for estimating the people counts in a retail store. The method can operate in real time on hardware that is low cost, easy to install, and capable of wireless transmission. Computationally efficient people counting was accomplished using line-wise APL-based background modeling, MAP-based dilated motion search, and MTS-based flow analysis.

By using the APL-based background modeling, the foreground was successfully extracted under various illumination environments, and then the MAP-based dilated motion search extracted the MV of people movements by incorporating the extracted foreground. After the motion search, MTS-based flow analysis estimated the flow volume caused by foreground movement. The final number of people entering or leaving the LOI was calculated using multivariate

linear regression with the flow volume, flow velocity, and flow direction. The benefits of the proposed method were verified through experiments using several test sets produced in various environments that are similar to a real retail store. The proposed method provided the best accuracy with much smaller processing time compared with the benchmark methods.

REFERENCES

[1] T.-M. Tsai, P.-C. Yang, and W.-N. Wang, "Pilot study toward realizing social effect in O_2O commerce services," in *Social Informatics* (Lecture Notes in Computer Science), vol. 8238. Cham, Switzerland: Springer, 2013, pp. 268–273.
[2] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian detection of independent motion in crowds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 594–601.
[3] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 705–711.
[4] J. García, A. Gardel, I. Bravo, J. L. Lázaro, M. Martínez, and D. Rodríguez, "Directional people counter based on head tracking," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3991–4000, Sep. 2013.
[5] Y. Cong, H. Gong, S.-C. Zhu, and Y. Tang, "Flow mosaicking: Real-time pedestrian counting without scene-specific learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1093–1100.
[6] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.
[7] Z. Ma and A. B. Chan, "Crossing the line: Crowd counting by integer programming with local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2539–2546.
[8] Z. Ma and A. B. Chan, "Counting people crossing a line using integer programming and local features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 10, pp. 1955–1969, Oct. 2016.
[9] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2871–2878.
[10] H. Xu, P. Lv, and L. Meng, "A people counting system based on head-shoulder detection and tracking in surveillance video," in *Proc. ICCDA*, vol. 1, Jun. 2010, pp. V1-394–V1-398.
[11] T. Bucher *et al.*, "Image processing and behavior planning for intelligent vehicles," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 62–75, Feb. 2003.
[12] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.
[13] S. J. Lee, Y. Motai, and M. Murphy, "Respiratory motion estimation with hybrid implementation of extended Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4421–4432, Nov. 2012.
[14] M. Eriksson, M. Armendariz, O. O. Vasilenko, A. Saleem, and L. Nordström, "Multiagent-based distribution automation solution for self-healing grids," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2620–2628, Apr. 2015.
[15] P. Cosimo *et al.*, "An embedded vision system for real-time autonomous localization using laser profilometry," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3482–3495, Dec. 2015.
[16] C.-H. Yeh, C.-Y. Lin, K. Muchtar, H.-E. Lai, and M.-T. Sun, "Three-pronged compensation and hysteresis thresholding for moving object detection in real-time video surveillance," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4945–4955, Jun. 2013.
[17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1999, pp. 246–252.
[18] M. Heikkila and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.
[19] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks," in *Proc. 6th Int. Workshop Energy Minimization Methods Comput. Vis. Pattern Recognit. (EMMCVPR)*, 2007, pp. 169–183.
[20] B.-F. Wu, C.-C. Kao, C.-L. Jen, Y.-F. Li, Y.-H. Chen, and J.-H. Juang, "A relative-discriminative-histogram-of-oriented-gradients-based particle filter approach to vehicle occlusion handling and tracking," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4228–4237, Aug. 2014.
[21] M. I. Chacon-Murguia and S. Gonzalez-Duarte, "An Adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3286–3298, Aug. 2012.
[22] T. Celik and H. Kusetogullari, "Solar-powered automated road surveillance system for speed violation detection," *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3216–3227, Sep. 2010.
[23] D. Wang, L. Zhang, and A. Vincent, "Motion-compensated frame rate up-conversion—Part I: Fast multi-frame motion estimation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 131–141, Jun. 2010.

SUNG IN CHO (S'10–M'17) received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2010, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology in 2015. From 2015 to 2017, he was a Senior Researcher at LG Display. He is currently an Assistant Professor of electronic engineering with Daegu University. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, deep learning algorithm, and circuit design for LCD and OLED systems.

SUK-JU KANG (S'08–M'11) received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2006, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology in 2011. From 2011 to 2012, he was a Senior Researcher at LG Display, where he was a Project Leader for Resolution Enhancement and Multi-View 3-D System Projects. From 2012 to 2015, he was an Assistant Professor of electrical engineering with Dong A University, Busan. He is currently an Associate Professor of electronic engineering with Sogang University. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, and circuit design for LCD, OLED, and 3-D display systems.