Robust Multi-person Tracking for Real-Time Intelligent Video Surveillance

Jin-Woo Choi, Daesung Moon, and Jang-Hee Yoo

We propose a novel multiple-object tracking algorithm for real-time intelligent video surveillance. We adopt particle filtering as our tracking framework. Background modeling and subtraction are used to generate a region of interest. A two-step pedestrian detection is employed to reduce the computation time of the algorithm, and an iterative particle repropagation method is proposed to enhance its tracking accuracy. A matching score for greedy data association is proposed to assign the detection results of the two-step pedestrian detector to trackers. Various experimental results demonstrate that the proposed algorithm tracks multiple objects accurately and precisely in real time.

Keywords: Multiple-object tracking, particle filter, background modeling, pedestrian detection, real-time applications, video surveillance applications.

I. Introduction

Intelligent video surveillance systems are increasingly employed in the field of security. Among several core technologies of intelligent video surveillance, automatic and robust real-time tracking of multiple people is essential. Multi-person tracking in a complex, real-world environment remains a challenging problem. There are many challenges that make tracking a difficult problem, such as illumination changes, occlusions, background clutters, scale changes, shape variations, and fast motions [1]. To solve these problems, many tracking algorithms have been recently proposed.

There are many multiple-object tracking algorithms for the tracking of multiple targets through association with detection results [2]–[6]. These algorithms approach the problem of multi-target tracking by optimizing detection assignments over a temporal window [2]–[3], [5] or by using global optimization of data association [3]. However, these approaches are non-causal methods and are ones that require future frames to estimate the position of an object in a current frame. Therefore, local or global optimization–based methods are unsuitable for real-time surveillance applications such as intrusion detection.

One of the promising causal tracking methods is sequential Monte Carlo [7], also known as particle filter–based tracking. Particle filter–based tracking methods [8]–[15] represent tracking uncertainty in a Markovian manner; thus, they are suitable for online applications. In addition, a particle filter has the ability to handle non-Gaussianity and multi-modality. Consequently, particle filter–based tracking is robust to partial occlusions, rotations, and rapid motions. Nummiaro and others [11] proposed a color histogram–based particle filter that is robust to illumination changes and pose variations. Nevertheless, initialization of a tracker is somewhat naive for

Manuscript received May 22, 2014; revised Jan. 1, 2015; accepted Jan. 17, 2015.

This work was partly supported by the ICT R&D program of the MSIP, Rep. of Korea (No. 10039149, Development of Basic Technology of Human Identification and Retrieval at Distance for Active Video Surveillance Service with Real-time Awareness of Safety Threats) and (No. 13-912-06-006, Development of the Filtering Technology for Objectionable Streaming Contents on Smart Platform).

Jin-Woo Choi (jwc@etri.re.kr), Daesung Moon (daesung@etri.re.kr), and Jang-Hee Yoo (corresponding author, jhy@etri.re.kr) are with the SW & Contents Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.15.0114.0629

application to surveillance camera systems.

There are object tracking approaches utilizing both particle filtering and object detection techniques to automatically initialize each tracker and enhance the overall tracking performance. Okuma and others [12] proposed a color-based particle filtering method that initializes each tracker using the detector output. Breitenstein and others [13]–[14] extended this idea using a detector confidence term. Iwahori and others [16] proposed a particle filter with an adaptive-boosting (AdaBoost) classifier to restrict a particle distribution to a rectangular region of the classifier output. Xu and Gao [17] combined human detection and tracking based on a Histogram of Oriented Gradients - support vector machine (HOG-SVM) classifier and particle filter. Some multiple-object tracking methods use background modeling and subtraction before object detection to reduce the search region [15], [18]. This approach can reduce the computation time of the pedestrian detector and therefore the overall time required to track multiple persons.

In this work, we propose a novel real-time multi-person tracking method. By employing a two-step pedestrian detector; an iterative particle repropagation (IPR) method; and a matching score for data association and state update method, the proposed algorithm achieves real-time processing speed with a high tracking performance in a fixed-camera environment. The proposed method shows a comparable, and in some cases superior, tracking accuracy and precision in real time.

To summarize, our major contributions are as follows:

- A two-step detection approach to reduce the computation time of the pedestrian detector.
- A novel IPR method to reduce the chance of tracker drift or failure.
- A matching score for data association and state update method.
- Real-time processing speed with a high tracking accuracy and precision.

The rest of this paper is organized as follows. Section II describes the proposed multiple-object tracking algorithm in detail. Various experimental results from quantitative and qualitative analyses are shown in Section III. Finally, we provide some concluding remarks in Section IV.

II. Proposed Algorithm

The proposed multiple-object tracking method consists of the following three parts: pedestrian detection (Section II-1), tracker (Section II-2), and data association (Section II-3). An overall block diagram of the proposed multiple-object tracking algorithm is described in Fig. 1.

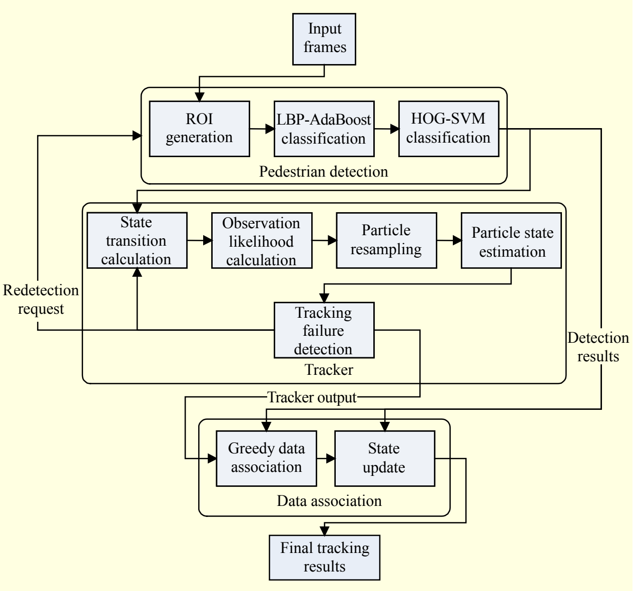

Fig. 1. Block diagram of proposed algorithm.

A basic assumption for the proposed algorithm is that of a fixed-camera environment. Multi-person tracking in a moving-camera environment is beyond the scope of this paper. In the installation (tuning) stage, the proposed algorithm models a background and generates an image of it. Instead of employing an advanced background modeling method, such as a Gaussian mixture model, we employ a simple background modeling method based on frame difference accumulation.

Differences of consecutive frames are accumulated for ten consecutive frame intervals. For each pixel, the background model is updated by a weighted summation of the current frame and the previous background model only if the accumulated frame difference for a given pixel is small enough. By using this model-update criterion, only the pixels that belong to the background are updated to the background model. Most of the pixels belong to the object and as such are not updated to the background model.

For each input frame, the region of interest (ROI) is simply generated by subtracting the background from the input frame. The pedestrian detection algorithm detects those persons that are in the ROI using two-step (local binary patterns (LBP)-AdaBoost and HOG-SVM) classifiers. The detection results are used to initialize and update each tracker. The particle filter–based tracking algorithm then estimates the state of each object. For each object, the particles are propagated by a dynamic model, and the observation likelihood of each particle is evaluated. Particle resampling is performed to prune unlikely hypotheses. After estimating the state of each object, the proposed "Tracking Failure Detection" method determines whether the tracking result is successful. The proposed IPR method (see Section II-2-C) retries to track an object to make a successful decision. Finally, the greedy data association algorithm assigns the optimal matching detection to each tracker and updates the state of each object.

1. Pedestrian Detection

Pedestrian detection is performed to initialize and update the state of each tracker. To achieve the real-time tracking performance, the pedestrian detector works only on the ROI instead of the whole image. The ROI is simply generated by subtracting the background from the input frame, and the background is modeled by the aforementioned background modeling method.

In this work, a two-step pedestrian detection algorithm is proposed to improve the processing speed of a pedestrian detector. As a result, the speed of the overall multi-person tracking algorithm can be improved while maintaining a high detection accuracy, in contrast to [16] and [17], which employ only AdaBoost and SVM, respectively. The first step is to detect candidate regions that may or may not contain human beings using the AdaBoost classifier with LBP features [19]. Since the LBP-AdaBoost algorithm has a high processing speed, it can rapidly find candidate regions from an input ROI image. The next step is that of human detection, which is carried out using an SVM classifier with HOG features [20]. HOG-SVM is used to verify the remaining few candidate regions because it is relatively more accurate than an LBP-AdaBoost classifier, despite the fact that it is also slow.

For each ROI, an image pyramid with 15 layers is generated by downsampling the original ROI. The size ratio between consecutive layers is 1.21; thus, the minimum size of the pyramid layer is 7% of the original ROI. Then, an LBP-AdaBoost classifier with a 32 × 64 pixel sliding window scans each pyramid layer to detect candidate regions. After selecting candidate regions, the HOG-SVM-based algorithm classifies each candidate region as either human or non-human.

Figure 2 shows the effectiveness of the proposed two-step detection approach. The generated ROI (yellow bounding box) in the scene is much larger than any region containing a human. Searching the whole ROI with the HOG-SVM detector requires an excessive amount of time. Instead, the relatively faster LBP-AdaBoost is employed to filter out the most unreliable candidate regions, as depicted in Fig. 2(a). The HOG-SVM detector then verifies the remaining few candidate regions and finally rejects the false positive error, as depicted in Fig. 2(b). Using the proposed two-step pedestrian detection approach, we can improve the speed of the detection compared to the single-step approach, which uses only the HOG-SVM (see Section III-5).

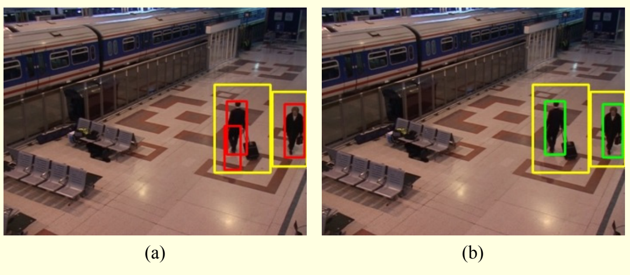

Fig. 2. Effectiveness of proposed two-step detection approach.

2. Tracker

Let us define a state **s** = (x, v_x, y, v_y) comprising the center position of an object and the object's corresponding velocity components. To achieve multiple-person tracking in real time while maintaining robust tracking accuracy, we employ a particle filter framework.

A. Particle Filter

A particle filter estimates the current probability distribution of a target for the given observations $\mathbf{z}_{1:t}$ as a weighted sum of N Kronecker delta functions (the particles), as described below [21]

$$p_t(\mathbf{s}_t \mid \mathbf{z}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{s}_t - \mathbf{s}_t^{(i)}), \tag{1}$$

where $\mathbf{s}_t$ is the state variable of an object and $w_t^{(i)}$ is the weight for the ith particle $\mathbf{s}_t^{(i)}$ at time step t. Using the *importance sampling* [21] and *bootstrap filter*, the weight for each particle can be expressed in such a way so that it is proportional to the likelihood of the corresponding observation, as described in the following:

$$w_t^{(i)} \propto p_t(\mathbf{z}_t \mid \mathbf{s}_t^{(i)}). \tag{2}$$

The proposed method for evaluating the observation likelihood (see Fig. 1) (2) is explained in the next subsection.

The size of the object is excluded from the state definition because the particle filter–based object tracking algorithms have weaknesses in object size estimation, especially when there is severe depth variation in a video sequence. However, the width and height of an object is essential in evaluating the observation likelihood of each particle and displaying the bounding box of an object. Therefore, the proposed algorithm employs an explicit size estimation method. The object-size estimation method estimates the current size of an object by

$$(\hat{w}_t, \hat{h}_t) = (w_{t-1}^{\mathrm{d}}, h_{t-1}^{\mathrm{d}}), \tag{3}$$

where w^{d}_{t-1} and h^{d}_{t-1} are the width and height of a detected object in the frame prior to frame t.

We use a first-order dynamic model to diffuse each particle.

$$\mathbf{S}_t = \mathbf{TS}_{t-1} + \mathbf{W}_{t-1}, \qquad (4)$$

where $\mathbf{S}_t$ is a 4 × N particle state matrix at time t, and the ith column of $\mathbf{S}_t$ is a particle state vector, $\mathbf{s}_t^{(i)}$. A state transition matrix, $\mathbf{T}$, propagates the particles with a first-order motion model, and $\mathbf{W}_{t-1}$ is a 4 × N random matrix in which each column is an independent and identically distributed multivariate Gaussian random vector that provides perturbations to the state components.

B. Observation Model

In our method, similarity between a target color histogram and a sample color histogram is used to evaluate an observation likelihood. To achieve robustness to illumination changes, the HSV color histograms are constructed using 6 × 6 × 6 bins. The observation likelihood of each particle can then be evaluated by

$$p(\mathbf{z}_t \mid \mathbf{s}_t^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{c}}} \exp\left\{ -\frac{1-\rho[l_{\mathbf{s}_t^{(i)}}, l_t]}{2\sigma_{\mathrm{c}}^{\,2}} \right\}, \qquad (5)$$

$$\rho[a,b] = \sum_{u=1}^{m} \sqrt{a^{(u)} b^{(u)}}, \qquad (6)$$

where $l_{\mathbf{s}_t^{(i)}}$ is the color histogram of the ith particle state vector $\mathbf{s}_t^{(i)}$, and l_t is a target histogram of a tracker. Here, $\rho[a, b]$ is the Bhattacharyya coefficient of two histograms, a and b, and σ_{c} is the standard deviation of the color noise.

To obtain an enhanced approximation of an object's color distribution, multiple color histograms are used. The tracking region is divided into m subregions. After a corresponding color histogram is constructed for each subregion, the m histograms, l_1, l_2, … , l_m, are then concatenated into one histogram, l_{c}. The concatenated color histogram is then used for calculation of (5) and (6). Therefore, the observation likelihood of each particle using multiple color histograms is constructed through (5) and (6) using the similarity between the concatenated sample histogram and concatenated target histogram.

Using multiple color histograms per object, both spatial information and global color distribution information are incorporated into the evaluation of an observation likelihood. As a result, improved object tracking precision can be achieved.

C. Iterative Particle Repropagation

To reduce the chance of tracker drift or tracking failure when there are some occlusions or fast motion, an IPR method is proposed. The IPR algorithm works as follows (see Table 1).

Table 1. Iterative particle repropagation.

```
For each object,
  1) Reset the counter variable $n = 0$.
  2) Save the particle state before particle propagation:
       $\mathbf{b}_{t-1}^{i} = \mathbf{s}_{t-1}^{i},\ i = 1, \dots, N.$
  3) Process the particle–filter based tracking method:
       $\mathbf{S}_t = \mathbf{TS}_{t-1} + \mathbf{W}_{t-1},$
       $p\left(\mathbf{z}_t \mid \mathbf{s}_t^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{c}}} \exp\left\{ -\frac{1-\rho[l_{\mathbf{s}_t^{(i)}}, l_t]}{2\sigma_{\mathrm{c}}^2} \right\}.$
  4) Decide the tracking is successful or not:
       $\mathrm{Status}(k) = \begin{cases} 0 & \text{if } \rho[l_{\hat{\mathbf{s}}_t^k}, l_t^k] < \rho_{\mathrm{th}}, \\ 0 & \text{if } \rho[l_{\hat{\mathbf{s}}_t^k}, l_t^k] - \rho[l_{\hat{\mathbf{s}}_{t-1}^k}, l_{t-1}^k] < \Delta_{\mathrm{th}}, \\ 1 & \text{otherwise.} \end{cases}$
  5) If Status($k$) = 0, restore the saved particle states and increase $n$ by 1:
       $\mathbf{s}_{t-1}^{i} = \mathbf{b}_{t-1}^{i},\ i = 1, \dots, N$ ,
       $n = n + 1$ .
     Else if Status($k$) = 1, go to step 8)
  6) If $n \geq N_{\mathrm{iter}}$, go to the step 7); else, go to step 2).
  7) Request redetection.
  8) Proceed to the next frame.
```

To determine whether the tracking of each object is successful or not, the following criterion is proposed:

$$\mathrm{Status}(k) = \begin{cases} 0 & \text{if } \rho[l_{\hat{\mathbf{s}}_t^k}, l_t^k] < \rho_{\mathrm{th}}, \\ 0 & \text{if } \rho[l_{\hat{\mathbf{s}}_t^k}, l_t^k] - \rho[l_{\hat{\mathbf{s}}_{t-1}^k}, l_{t-1}^k] < \Delta_{\mathrm{th}}, \\ 1 & \text{otherwise,} \end{cases} \qquad (7)$$

where $l_{\hat{\mathbf{s}}_t^k}$ is the color histogram constructed at the estimated position of the kth object, l_t^k is the target color histogram of the kth object, and ρ_{th} and Δ_{th} are predefined threshold values. The first condition of (7) is a failure condition of the color histogram similarity, and the second condition of (7) is a failure condition of the similarity difference between consecutive frames.

If a tracking failure is detected, then the proposed algorithm restores the distributed particles. To restore the particles, the particle states must be saved before propagation. The *save* and *restore* is performed by

$$\begin{aligned} \mathbf{b}_{t-1}^{i} &= \mathbf{s}_{t-1}^{i},\ i = 1, \dots, N, \\ \mathbf{s}_{t-1}^{i} &= \mathbf{b}_{t-1}^{i},\ i = 1, \dots, N. \end{aligned} \qquad (8)$$

The restored particles are then redistributed by the motion model, and the observation likelihood of each particle is evaluated. After the estimation of the object state, whether the tracking is successful or not is determined by (7). This procedure is repeated until the tracking result is determined as successful, or the number of iterations reaches a predefined value, N_{iter}. After the IPR step, if the final decision is one of

http://dx.doi.org/10.4218/etrij.15.0114.0629

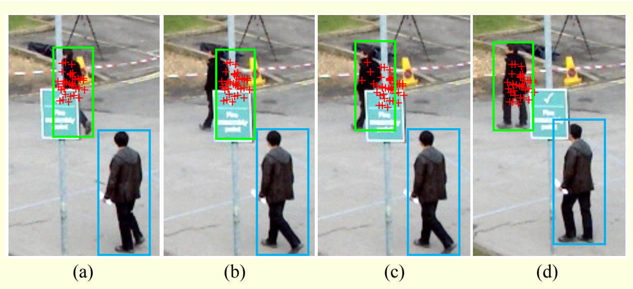

Fig. 3. Illustration of iterative particle repropagation: (a) frame t–1, (b) frame t before iterative particle repropagation, (c) frame t after iterative particle repropagation, (d) frame t+1.

tracking failure, then a redetection signal is transmitted to the pedestrian detection module. Otherwise, the proposed method proceeds to the next frame and tracks the objects as described above.

The proposed IPR method reduces the chance of a tracking failure from an unfortunate particle diffusion or color sampling when constructing a sample histogram. An illustration of this is given in Fig. 3 (the person within the green bounding box). At frame t–1, the particles are properly distributed, and the estimation of the object position is sufficiently accurate (Fig. 3(a)). At frame t, owing to an unreliable likelihood estimation or unfortunate particle diffusion, the particles are improperly distributed (Fig. 3(b)). If IPR is not employed, then improper particles will result in tracker drift.

However, in Fig. 3, using (7), a tracking failure is detected. After the detection of the tracking failure, the particles are restored and propagated again. Since the repropagation step iteratively retries to make a successful decision, the chance of a tracking failure is reduced. After the repropagation step, the tracking can be determined as successful, as shown in Fig. 3(c).

3. Data Association

To assign a detection result to a given tracker, a data association method is required. Instead of obtaining the optimal single-frame assignment using the Hungarian algorithm, we solve the assignment problem using a greedy data association algorithm. It is known that this shows similar results but at a lower computational complexity [14].

A. Matching Score

A matching score between the detector output and tracker consists of a distance term and a color similarity term. The matching score is defined as

$$M(\mathbf{s}_{\text{tr}}, \mathbf{s}_{\text{det}}) = \alpha p_N(\mathbf{s}_{\text{tr}} - \mathbf{s}_{\text{det}}) + \rho[l_{\text{tr}}, l_{\text{det}}], \tag{9}$$

where $p_N(\mathbf{s}_{\text{tr}} - \mathbf{s}_{\text{det}})$ is a Gaussian distribution with mean zero and standard deviation σ_{d} and α is a weighting factor for the distribution. The term $p_N(\mathbf{s}_{\text{tr}} - \mathbf{s}_{\text{det}})$ measures the affinity between the detection result $\mathbf{s}_{\text{det}}$ and the tracker $\mathbf{s}_{\text{tr}}$; hence, it gives weight to the detection result that is closer to the tracker. Here, $\rho[l_{\text{tr}}, l_{\text{det}}]$ is the Bhattacharyya coefficient (6) between the target color histogram of the tracker and the target color histogram of the detection result. The second term measures the similarity between the existing tracker and the detector output.

Combining the distance and similarity terms as a matching score is reasonable for the following reasons. A similar detector output far from the tracker should be penalized to remove an improper tracker jump between consecutive frames. The decision to combine a distance and a similarity term to give a matching score improves the tracking accuracy and precision remarkably, as shown in Section III-3-B.

The detector output with the maximum matching score is associated with a tracker. The associated detection result and tracker are excluded from the matching pool. The same procedure is processed until there is no valid pair available in the matching pool.

B. State Update

After the assignment, the proposed algorithm updates the state of each tracker using the information of the assigned detection result. In this update step, not only the state variable $\mathbf{s}_{\text{tr}} = (x, v_x, y, v_y)$ but also the width and height of the object w_t and h_t of each tracker are updated by assigning the associated detector output $\mathbf{s}_{\text{det}} = (x_{\text{det}}, y_{\text{det}}, w_{\text{det}}, h_{\text{det}})$ to x, y, w_t, and h_t, respectively. In addition, the target color histogram of the object is recalculated at the updated position. The recalculated color histogram makes the updated tracker more reliable in terms of MOTA.

The proposed algorithm counts the number of consecutive association failures of each tracker. If the number of consecutive association failures is greater than a predefined threshold, N^f_{th}, then the detector verifies whether the object is a human. The detector votes on three consecutive frames. If the tracker output is classified as a human for more than two frames, then the final decision is to classify the object as a human. Otherwise, the proposed algorithm classifies the object as a non-human and then terminates the tracker.

It is possible that a temporarily disappeared object reappears soon. Therefore, instead of terminating the tracker immediately, the proposed algorithm deactivates the tracker. If a disappeared object re-enters the scene within a predefined number of consecutive frames, N_{temp}, then the proposed algorithm recognizes the re-entering object and tracks it as a normal object. Otherwise, the deactivated tracker is permanently terminated.

III. Experimental Results

1. Implementation Details and Parameter Settings

The proposed algorithm was implemented in ANSI C and C++. In the implementation, no parallel processing technique, such as multi-threading or GPU processing, was used. We tested the proposed algorithm using a PC equipped with a 3.2 GHz 64-bit CPU and 8 GB of memory.

Background modeling was performed only for the first 200 frames of every test sequence except for *TUD Crossing*, since the *TUD Crossing* sequence is quite short (201 frames in total). To achieve a real-time processing performance, input frames were resized to 320 × 240 pixels for pedestrian detection. In the case of tracking, input frames were resized to 640 × 480 pixels to maintain a high tracking accuracy and precision while achieving real-time processing performance.

The LBP-AdaBoost and HOG-SVM classifiers were trained using the public INRIA DB [20]. A total of 2,416 positive samples from the INRIA Person dataset and 2,388 negative samples from natural images were used to train the classifiers. The negative samples were extracted from the same dataset manually.

Every algorithm parameter was fixed for all experiments in this study. The number of particles per tracker, N, was fixed at 100. Two histograms per object were employed to generate the concatenated color histogram since this was sufficient to approximate the observation likelihood. Standard deviations σ_c and σ_d were set to 0.2 and 10.0, respectively, and α was set to 10. Tracking failure detection thresholds ρ_{th} and Δ_{th} were set to 0.8 and –0.2, respectively, after testing a large number of experiments. This is a compromise between false-positive failure detection and missing failure detection. Here, N_{iter} was set to 2. Both N_{th} and N_{temp} were set to 40.

2. Test Sequences

For a fair comparison between the performance of the proposed algorithm and state-of-the-art algorithms, various test sequences were used. The test sequences include real-world surveillance videos LAB hallway normal, (Lab normal) and LAB hallway low illumination (Lab low), made by us, and PETS 2009 S2.L1 View 001 (S2L1), PETS 2009 S2.L2 View 001 (S2L2) [22], AVSS 2007 iLIDS AB-Easy (iLIDS Easy) AVSS 2007 iLIDS AB-Medium (iLIDS Medium) [23]–[24], PETS 2006 S4-T5-A View 004 (PETS 2006) [25], and TUD Crossing [3], all of which are publicly available.

3. Quantitative Analysis

CLEAR MOT [26] metrics were used to evaluate the objective tracking performance. Among the various measures, two metrics — multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) — were used.

A. Performance Comparison

We compared our tracking method with the state-of-the-art methods on sequences S2L1, S2L2, iLIDS Easy, iLIDS Medium, PETS 2006, and TUD Crossing. The comparison results are shown in Table 2. MOTP and MOTA scores of the compared algorithms were extracted from the corresponding researches [2], [14], [18], and [27]. MOTA scores solely were extracted from [23] and [24], because they did not provide MOTP scores.

For S2L1, the proposed algorithm most accurately and precisely tracked the targets among the compared state-of-the-art methods. GTO [2] is a multi-camera system that uses five camera views and scene-specific knowledge. However, our method outperforms state-of-the-art methods in terms of both precision and accuracy. Although DCPF [14] employs a powerful online machine learning algorithm, thus its computational complexity is much higher, the proposed algorithm shows better results.

Although the proposed algorithm did not yield the best results for the highly challenging S2L2 sequence, the accuracy of our method is comparable to the state-of-the-art method. DCPF [14] achieves about only a six percentage point higher accuracy than the proposed algorithm despite the much higher complexity (see Section III-5). CEM [27] is based on the

Table 2. Comparison of CLEAR MOT [26] evaluation results (figures in red indicate the best results).

Sequence	Method	MOTP	MOTA
S2L1	Proposed	67.47	84.21
	DCPF [14]	56.30	79.70
	PMPT [18]	53.79	75.97
	GTO [2]	60.00	66.00
S2L2	Proposed	49.02	44.41
	DCPF [14]	51.30	50.00
	CEM [27]	73.20	47.80
iLIDS Easy	Proposed	70.79	67.17
	DCPF [14]	67.00	78.10
iLIDS Medium	Proposed	62.30	53.95
	DCPF [14]	66.00	76.00
	HADR [23]	—	68.40
	EPD [24]	—	55.30
PETS 2006	Proposed	66.34	81.58
TUD Crossing	Proposed	61.10	67.18
	DCPF [14]	71.00	84.30

http://dx.doi.org/10.4218/etrij.15.0114.0629

global energy minimization of the trajectories, which is non-causal. Nonetheless, the accuracy gap between the proposed algorithm and CEM is only three percentage points. In terms of precision, the performance gap between the proposed method and DCPF [14] is only 2.3 percentage points.

For iLIDS Easy, the proposed algorithm yielded the best performance in terms of precision. In contrast, the proposed algorithm had about a nine percentage point lower MOTA score than DCPF [14] because our algorithm relies on the background model generated in the first 200 frames. The iLIDS Easy sequence contains the motion of a subway train, which is not present in the first 200 frames. For this reason, the proposed algorithm suffers from false positives at the train regions; thus, its accuracy score drops.

For iLIDS Medium, the proposed algorithm shows a comparable precision performance. In terms of accuracy, the proposed algorithm shows the worst performance against the three methods compared. The main reason for the accuracy gap is missing detections of several partially visible persons. Since the detector employed is not a part-based one, the proposed algorithm suffers from missing detections of partially occluded persons.

Because TUD Crossing contains considerable background changes throughout the whole sequence and no background modeling is performed for this sequence, the accuracy of the proposed algorithm is lower than that of DCPF [14]. Furthermore, a long-term inter-object occlusion occurs several times in this sequence. The proposed algorithm fails to track persons occasionally during such occlusions; thus, this causes the accuracy score to drop.

Although for some test sequences the state-of-the-art methods show better results, the compared algorithms are definitely not real-time algorithms. In contrast, the proposed algorithm runs in real-time while maintaining reasonable tracking precision and accuracy (see Section III-5). Therefore, the proposed algorithm is more suitable than the conventional algorithms for use in practical video surveillance applications.

B. *Effectiveness of Proposed Algorithm*

The effectiveness of the proposed algorithm is demonstrated in Table 3. As shown in the table, IPR remarkably enhanced the tracking accuracy. For the S2L1 sequence, the accuracy gap between "with IPR" and "without IPR" is about 4.5 percentage points. As illustrated in Section II-2-C, the proposed IPR effectively retried to make the tracking successful, and the chance of a tracking failure was reduced. As a result, the accuracy of the tracking was enhanced. Since the IPR method was designed to reduce the chance of a tracking failure, the influence on the tracking precision is insubstantial.

Without the proposed data association method, the tracking accuracy drops more owing to frequent identity switches. As described in Section II-3, the proposed data association method effectively matched the detection results and trackers; thus, the accuracy of the tracking was enhanced. In addition, using multiple histograms per object improved the tracking precision, as demonstrated in Table 3. The experimental results show that, as intended, the proposed IPR, data association, and observation model effectively enhance the accuracy and precision.

Table 3. Effectiveness of proposed algorithm. "IPR off" shows test results of proposed algorithm without IPR method. "IPR+DA+MH off" shows test results of proposed algorithm without IPR, data association, and multiple color histograms per object.

Sequence	Method	MOTP	MOTA
S2L1	Proposed	67.47	84.21
	IPR off	67.43	79.70
	IPR+DA+MH off	62.52	76.87
iLIDS Easy	Proposed	70.79	67.17
	IPR off	70.60	64.57
	IPR+DA+MH off	64.51	53.76
PETS 2006	Proposed	66.34	81.58
	IPR off	66.41	79.23
	IPR+DA+MH off	64.68	74.20

4. Qualitative Analysis

In Fig. 4, the qualitative tracking performance of the proposed algorithm is depicted. Each tracked object is highlighted using bounding boxes with different colors. Figure 4(a) illustrates the tracking results in S2L1. The third and fourth columns of Fig. 4(a) show that the proposed algorithm tracks objects without an identity switch when severe inter-object occlusions exist. In the first and second columns of Fig. 4(a), the identity of a person who re-entered a scene was successfully maintained.

S2L2 is highly challenging because of high-density crowds in the scene and variations in illumination. As demonstrated in the second and third columns of Fig. 4(b), some persons in the upper region of the scene were not detected owing to the excessive illumination conditions. Several identity switches occurred throughout the sequence because some people in the crowd were close and similar to each other. Although some missing detections and identity switches occurred, most of the people were tracked with reasonable quality.

Figures 4(c) and 4(d) show the tracking results in iLIDS Easy and iLIDS Medium. The proposed algorithm shows satisfying results despite the drastic scale change of the objects.

http://dx.doi.org/10.4218/etrij.15.0114.0629

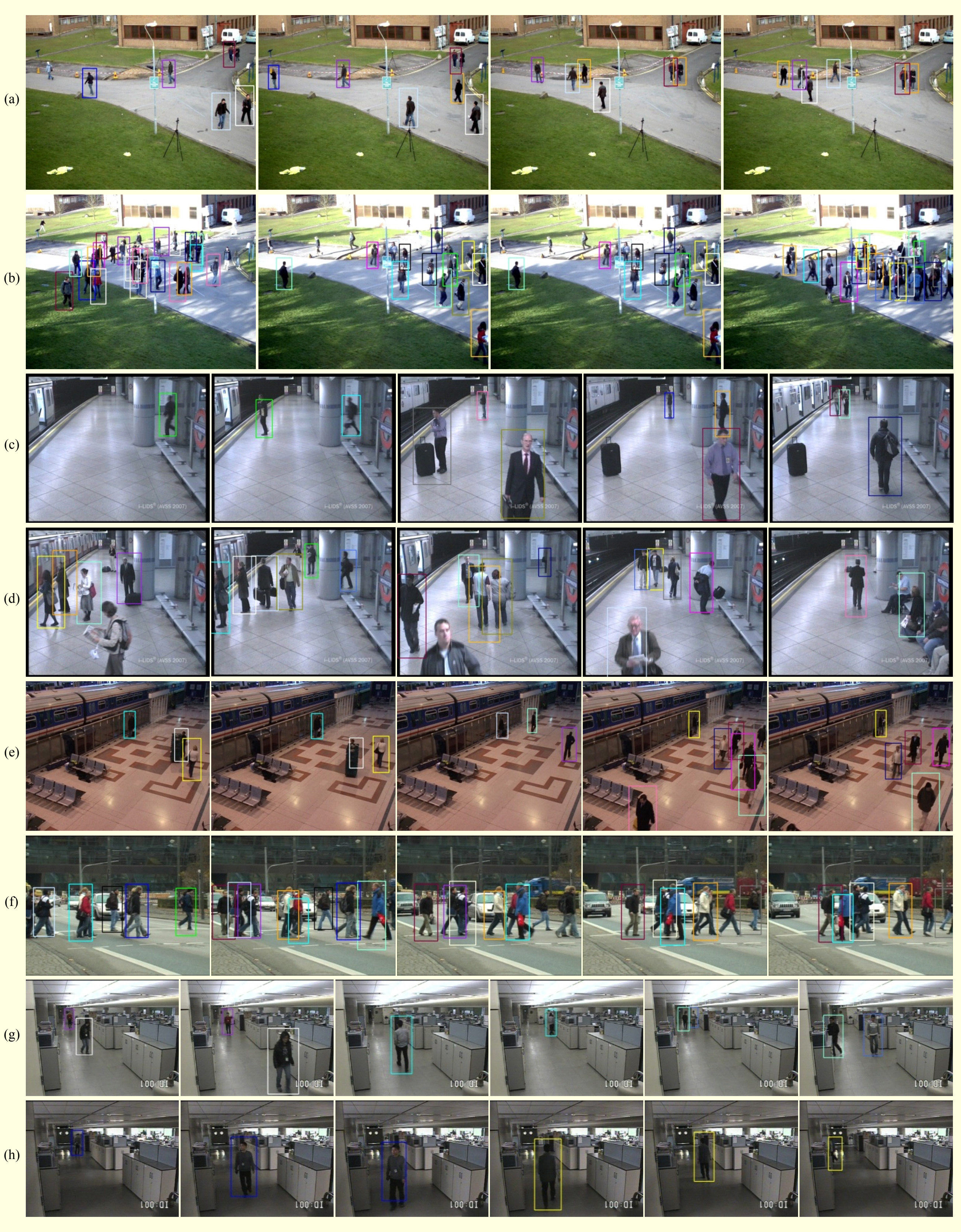

Fig. 4. Tracking results on (a) S2L1, (b) S2L2, (c) iLIDS Easy, (d) iLIDS Medium, (e) PETS 2006, (f) TUD Crossing, (g) Lab normal, and (h) Lab low sequences.

However, there are certain limitations. In the third column of Fig. 4(c), a man and his suitcase were detected as a single person. The reason for this error is the naive method of interaction between the background subtraction and the detector. Since the classifiers used in this work are not part-based, there were substantial missing detections of partially visible objects, as shown in the first, third, and fifth columns of Fig. 4(d).

Figure 4(e) presents the tracking results for PETS 2006. This sequence contains medium inter-object occlusions and long-time standing objects. The proposed method tracked the objects accurately. Nevertheless, the tracker occasionally failed to track the man standing next to the wall. Since the texture of the man was similar to that of the wall, the detector classified the man as a non-human over N^{f}_{th} frames. Consequently, the detector failed to detect the man over a few frames but then re-detected the object at a later stage. This caused a re-initialization of the object.

There are several long-term inter-object occlusions in the TUD Crossing sequence. As depicted in the third, fourth, and fifth columns of Fig. 4(f), the identities of the occluded objects were maintained (marked as an orange bounding box and a white bounding box). However, some persons to the right of the scene were missed owing to a non-static background.

The Lab normal and Lab low sequences contain severe depth variations. As shown in Figs. 4(g) and 4(h), the proposed algorithm successfully tracked the targets and robustly estimated the size of the objects despite the drastic scale change of the objects. Although Lab low contains a much lower illumination than typical surveillance sequences, the proposed algorithm tracked the objects with reasonable quality.

5. Processing Speed

Table 4 shows the processing speeds of the proposed and state-of-the-art algorithms. The processing speeds of the compared algorithms were extracted from the corresponding researches [14], [23]–[24] except for the processing speeds of PMPT [18], GTO [2], and CEM [27], which were not reported. Nonetheless, CEM [27] and GTO [2] definitely do not run in real-time owing to their global optimization structure.

To demonstrate the influence of the two-step pedestrian detection approach, we tested the proposed algorithm without two-step pedestrian detection (SVM only) on the two datasets, S2L1 and S2L2. As shown in the table, the detection speed drops drastically for S2L2 because the high-density crowd in the sequence caused the search region for the detector to be larger. Therefore, it required an excessive amount of time for detection. In contrast, employing a much faster LBP-AdaBoost detector, the proposed algorithm filtered out most of the unreliable candidate regions with a higher speed. HOG-SVM was performed only for a small number of remaining candidate regions. Consequently, the proposed two-step pedestrian detection approach effectively improved the detection speed, as can be seen from the table.

Table 4. Comparison of processing speeds (fps). Figures in red indicate the best results. Second column indicates the maximum number of targets in the sequence.

Sequence	Target count	Method	Total	Tr	Det
S2L1 (768 × 576)	7	Proposed	15.1	33.2	27.9
		SVM only	12.9	27.6	24.3
		DCPF [14]	0.4–2	—	—
		PMPT [18]	—	—	—
		GTO [2]	—	—	—
S2L2 (768 × 576)	35	Proposed	6.0	15.0	9.9
		SVM only	1.4	10.4	1.7
		DCPF [14]	0.4–2	—	—
		CEM [27]	—	—	—
iLIDS Easy (720 × 576)	4	Proposed	29.5	86.4	44.8
		DCPF [14]	0.4–2	—	—
iLIDS Medium (720×576)	9	Proposed	18.4	68.0	25.4
		DCPF [14]	0.4–2	—	—
		HADR [23]	—	50.0	—
		EPD [24]	1.0	—	—
PETS 2006 (720×576)	7	Proposed	31.2	66.7	58.8
TUD Crossing (640×480)	8	Proposed	14.2	38.4	22.6
		DCPF [14]	0.4–2	—	—
Lab normal (640×480)	2	Proposed	33.4	—	—
Lab low (640×480)	2	Proposed	31.4	—	—

Compared to the state-of-the-art method, the proposed algorithm yielded the fastest processing speed for every test sequence except for the test case of iLIDS Medium. Since the processing speed of HADR [23] was measured under the assumption that all detections were given, it is not fair to directly compare the HADR [23] results with the other results. In terms of the tracking speed, the proposed algorithm was 18 frames per second (fps) faster than HADR [23].

Since S2L2 contains a dense crowd of people, the processing speed of the proposed algorithm shows a relatively slower processing speed. Nonetheless, the tracker processing speed is 15.0 fps, and the processing time per object is 5 ms to 6 ms, on average, which is a reasonable speed for real-time applications. In addition, compared to DCPF [14], the proposed algorithm achieved a much faster processing speed.

The relationship between the number of targets per frame

http://dx.doi.org/10.4218/etrij.15.0114.0629

and the total processing speed is not trivial, because the processing time of the detector depends on the size and number of ROIs. For the tracker, the processing time is increased linearly as the number of targets increases. The average tracker processing time per target is 5 ms to 6 ms, as mentioned above.

IV. Conclusion

We have proposed a real-time multi-person tracking system for intelligent video surveillance. Under a fixed-camera environment assumption, the proposed algorithm robustly detects and tracks human objects without supervision. Various experimental results show that the quantitative performance of the proposed algorithm is comparable to state-of-the-art algorithms in terms of tracking precision and accuracy while maintaining a real-time processing speed. Furthermore, the proposed real-time multiple-object tracking algorithm outperformed the state-of-the-art algorithms in certain cases despite the fact that the compared algorithms are definitely not real-time algorithms. If parallelization techniques are used, then the proposed algorithm can be applied to an embedded environment and run in real time. The proposed multi-person tracking algorithm is expected to be applied to commercial smart surveillance camera systems.

References

[1] A. Yilmaz, O. Javed, and M. Shah, “Object Tracking: A Survey,” *ACM Comput. Surveys*, vol. 38, no. 4, Dec. 2006, pp. 1–45.

[2] J. Berclaz, F. Fleuret, and P. Fua, “Robust People Tracking with Global Trajectory Optimization,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, New York, USA, June 17–22, 2006, pp. 744–750.

[3] M. Andriluka, S. Roth, and B. Schiele, “People-Tracking-by-Detection and People-Detection-by-Tracking,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, Anchorage, AK, USA, June 23–28, 2008, pp. 1–8.

[4] Y. Li, C. Huang, and R. Nevatia, “Learning to Associate: Hybrid Boosted Multi-target Tracker for Crowded Scene,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, Miami, FL, USA, June 20–25, 2009, pp. 2953–2960.

[5] A. Perera et al., “Multi-object Tracking through Simultaneous Long Occlusions and Split-Merge Conditions,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, New York, USA, June 17–22, 2006, pp. 666–673.

[6] B. Benfold and I. Reid, “Stable Multi-target Tracking in Real-Time Surveillance Video,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, Providence, RI, USA, June 20–25, 2011, pp. 3457–3464.

[7] A. Doucet, N.D. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, New York, USA: Springer-Verlag, 2001.

[8] M. Isard and A. Blake, “Condensation-Conditional Density Propagation for Visual Tracking,” *Int. J. Comput. Vis.*, vol. 29, no. 1, Aug. 1998, pp. 5–28.

[9] H.-Y. Cheng and J.-N. Hwang, “Adaptive Particle Sampling and Adaptive Appearance for Multiple Video Object Tracking,” *Signal Process.*, vol. 89, no. 9, Sept. 2009, pp. 1844–1849.

[10] P. P´erez et al., “Color-Based Probabilistic Tracking,” *Conf. Comput. Vis.*, Copenhagen, Denmark, vol. 2350, May 28–31, 2002, pp. 661–675.

[11] K. Nummiaro, E. Koller-Meier, and L.V. Gool, “An Adaptive Color-Based Particle Filter,” *Image Vis. Comput.*, vol. 21, no. 1, Jan. 2003, pp. 99–110.

[12] K. Okuma et al., “A Boosted Particle Filter: Multitarget Detection and Tracking,” *Conf. Comput. Vis.*, Prague, Czech Rep., vol. 3021, May 11–14, 2004, pp. 28–39.

[13] M.D. Breitenstein et al., “Robust Tracking-by-Detection Using a Detector Confidence Particle Filter,” *IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, Sept. 29–Oct. 2, 2009, pp. 1515–1522.

[14] M.D. Breitenstein et al., “Online Multiperson Tracking-by-Detection from a Single Uncalibrated Camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, Sept. 2011, pp. 1820–1833.

[15] J.-W Choi and J.-H. Yoo, “Real-Time Multi-person Tracking in Fixed Surveillance Camera Environment,” *IEEE Int. Conf. Consum. Electron.*, Las Vegas, NV, USA, Jan. 11–14, 2013, pp. 125–126.

[16] Y. Iwahori et al., “Efficient Tracking with AdaBoost and Particle Filter under Complicated Background,” *Int. Conf. KES*, Zagreb, Croatia, vol. 5178, Sept. 3–5, 2008, pp. 887–894.

[17] F. Xu and M. Gao “Human Detection and Tracking Based on HOG and Particle Filter,” *Int. Congress Image Signal Process.*, Yantai, China, Oct. 16–18, 2010, pp. 1503–1507.

[18] J. Yang et al., “Probabilistic Multiple People Tracking through Complex Situations,” *CVPR*, *Performance Evaluation Tracking Surveillance*, Miami, FL, USA, June 25, 2009, pp. 79–86.

[19] L. Zhang et al., “Face Detection Based on Multi-block LBP Representation,” *Int. Conf. ICB*, Seoul, Rep. of Korea, vol. 4642, Aug. 27–29, 2007, pp. 11–18.

[20] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, San Diego, CA, USA, June 20–26, 2005, pp. 886–893.

[21] M.S. Arulampalam et al., “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, Feb. 2002, pp. 174–188.

[22] J. Ferryman and A. Shahrokni, “PETS: Dataset and Challenge,” *IEEE Int. Workshop Performance Evaluation Tracking Surveillance*, Miami, FL, USA, June 25, 2009, pp.1–6.

[23] C. Huang, B. Wu, and R. Nevatia, “Robust Object Tracking by Hierarchical Association of Detection Responses,” *Conf. Comput. Vis.*, Marseille, France, vol. 5303, Oct. 12–18, 2008, pp. 788–801.

http://dx.doi.org/10.4218/etrij.15.0114.0629

[24] B. Wu and R. Nevatia, “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet-Based Part Detectors,” *Int. J. Comput. Vis.*, vol. 75, no. 2, Nov. 2007, pp. 247–266.

[25] D. Thirde, L. Li, and J. Ferryman, “Overview of the PETS 2006 Challenge,” *IEEE Int. Workshop Performance Evaluation Tracking Surveillance*, New York, USA, June 18, 2006, pp. 47–50.

[26] K. Bernardin and R. Stiefelhagen, “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics,” *EURASIP J. Image Video Process.*, May 2008, pp. 1–10.

[27] A. Andriyenko and K. Schindler, “Multi-target Tracking by Continuous Energy Minimization,” *IEEE Conf. Comput. Vis. Pattern Recogn.*, Providence, RI, USA, June 20–25, 2011, pp. 1265–1272.

Jin-Woo Choi received his BS degree in electrical engineering and his MS degree in electrical engineering and computer science from Seoul National University, Rep. of Korea, in 2008 and 2010, respectively. From 2010, he has been working at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, as a researcher. His research interests include visual object tracking, object detection, object recognition, and computer vision applications.

Daesung Moon received his MS degree in computer engineering from Pusan National University, Rep. of Korea, in 2001. He received his PhD degree in computer science from Korea University, Seoul, Rep. of Korea, in 2007. He joined the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, in 2000, where he is currently a senior researcher. His research interests include network security, data mining, biometrics, and image processing.

Jang-Hee Yoo received his BS degree in physics and his MS degree in applied computer science from Hankuk University of Foreign Studies, Seoul, Rep. of Korea, in 1988 and 1990, respectively. He received his PhD degree in electronics and computer science from the University of Southampton, UK, in 2004. Since November 1989, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, as a principal researcher and is currently a visiting scientist at the University of Washington, Seattle, USA, until July 2015. He has also been a professor with the Department of Information Security Engineering, University of Science and Technology, Daejeon, Rep. of Korea. His current research interests include embedded computer vision, biometric systems, human motion analysis, intelligent video surveillance, HCI, and intelligent robots. He is a senior member of the IEEE and a member of the IEEK.

http://dx.doi.org/10.4218/etrij.15.0114.0629