Optical Engineering

SPIEDigitalLibrary.org/oe

Human tracking in thermal images using adaptive particle filters with online random forest learning

Byoung Chul Ko
Joon-Young Kwak
Jae-Yeal Nam

SPIE

Human tracking in thermal images using adaptive particle filters with online random forest learning

Byoung Chul Ko
Joon-Young Kwak
Jae-Yeal Nam
Keimyung University
Department of Computer Engineering
1000 Shindang-Dong
Dalseo-Gu, Daegu 704-701, Republic of Korea
E-mail: niceko@kmu.ac.kr

Abstract. This paper presents a fast and robust human tracking method to use in a moving long-wave infrared thermal camera under poor illumination with the existence of shadows and cluttered backgrounds. To improve the human tracking performance while minimizing the computation time, this study proposes an online learning of classifiers based on particle filters and combination of a local intensity distribution (LID) with oriented center-symmetric local binary patterns (OCS-LBP). Specifically, we design a real-time random forest (RF), which is the ensemble of decision trees for confidence estimation, and confidences of the RF are converted into a likelihood function of the target state. First, the target model is selected by the user and particles are sampled. Then, RFs are generated using the positive and negative examples with LID and OCS-LBP features by online learning. The learned RF classifiers are used to detect the most likely target position in the subsequent frame in the next stage. Then, the RFs are learned again by means of fast retraining with the tracked object and background appearance in the new frame. The proposed algorithm is successfully applied to various thermal videos as tests and its tracking performance is better than those of other methods. [DOI: 10.1117/1.OE.52.11.113105]

Subject terms: human tracking; thermal image; particle filters; random forest; local intensity distribution; oriented center-symmetric local binary patterns, online learning.

Paper 131054 received Jul. 15, 2013; revised manuscript received Oct. 17, 2013; accepted for publication Oct. 18, 2013; published online Nov. 18, 2013.

1 Introduction

Human tracking is an essential work for human gesture recognition, surveillance applications, augmented reality, and human–computer interfaces. Therefore, the tracking of humans in videos has received considerable attention in the computer vision field, and many successful human tracking approaches have been proposed in recent years.

Human tracking researches can be divided into two categories according to the sensors. Electro-optical (EO) sensors such as charge-coupled devices (CCDs) are the most widely used cameras for human detection and tracking. Human tracking based on the input images captured by RGB-EO sensors has already been producing reliable performance using color information when the illumination is constant and the target image quality is good.[1] However, much of the human tracking research based on EO sensors is not applicable to certain tasks in dark indoor and outdoor environments because of the changeable illumination, existence of shadows, and cluttered backgrounds. In contrast to EO sensors, thermal sensors allow the robust tracking of a human body in outdoor environments during day or night, regardless of poor illumination conditions and the body posture.

In general, thermal sensors can detect relative differences in the amounts of thermal energy emitted or reflected from different parts of a human body in a scene.[2] That is, the temperature of the background is largely different from that of the human being. Moreover, the price of a thermal camera has fallen significantly with the development of infrared technology, and thermal cameras have been used in many industrial, civil, and military fields.[3,4] However, there are still many problems to solve for reliable human tracking with thermal sensors.

- Nonhuman target objects such as buildings, cars, animals, and light poles having intensities similar to those of humans.[4]
- Persons overlapping while crossing paths.[5]
- Low signal-to-noise ratios and white-black or hot-cold polarity changes.[6]
- Halos appearing around very hot or cold objects.[6]
- Differences in temperature intensity between humans and backgrounds depending on weather and season.

Other important disadvantages of many successful tracking methods are the assumptions that the background is static, the target appearance is fixed, the image quality is good, and the illumination is constant.[7] However, in practice, the appearances of humans and the lighting conditions are changing constantly. Further, the background is not static, especially in the case of a camera that is installed in a mobile platform.

In our work, we use a long-wave infrared (LWIR) thermal camera instead of EO sensors to track humans, particularly at night and in outdoor environments, under the assumption that the camera is freely oriented for a mobile platform application.

1.1 *Related Work*

Object tracking has been studied widely in the field of video surveillance. In tracking approach, there are two types of

object tracking: multiple target tracking and single target tracking. In this paper, we focus on single target tracking. The purpose of our research is to automatically track the object's bounding box in every frame by using a given bounding box provided by the user including the object of interest in a first frame.

In the current research, object tracking can be classified as follows.

First, deterministic methods[8] typically track an object by performing an iterative search for the local maxima of a similarity cost function of the template image and the current image. Jurie and Dhome[9] employed the color distribution, with a metric derived from the Bhattacharyya coefficient as the similarity measure, and used the mean-shift procedure to perform the optimization. The mean-shift algorithm[10] is a popular algorithm for deterministic methods.

Second, the statistical methods solve tracking problems by taking the uncertainties of the measurements and model into account during object state estimation.[11] The statistical correspondence methods use the state space approach to model object properties such as position, velocity, and acceleration. Kalman filters[12] are used to estimate the state of a linear system when the state is assumed to have a Gaussian distribution. One limitation of the Kalman filters is the assumption that the state variables are normally distributed. Thus, the Kalman filters will give poor estimates of state variables that do not follow a Gaussian distribution. This limitation can be overcome by using particle filters,[13] also known as condensation algorithms or sequential Monte Carlo methods, which are efficient statistical methods to estimate target states. Most recent studies[7,8,14–16] have attempted to apply particle filters to the tracking systems so that dependable object tracking results can be achieved. Yang et al.[8] proposed hierarchical particle filters for tracking fast multiple objects by using integral images for efficiently computing the color features and edge orientation histograms. The observation likelihood based on multiple features is computed in a coarse-to-fine manner. Deguchi et al.[14] employed the mean-shift algorithm to track the target and incorporate the particle filters into the mean-shift result in order to cope with a temporal occlusion of the target and reduce the computational cost of the particle filters. Khan et al.[15] also employed particle filters and mean shift jointly to reduce computational cost and detect occluded objects by estimating the dynamic appearances of objects with online learning of a reference object. Sidibe et al.[16] presented an object tracking method based on the integration of visual saliency information into the particle filter framework to improve the performance of particle filters against occlusion and large illumination variations.

For online learning, Klein et al.[7] proposed a visual object tracking method using a strong classifier that comprises an ensemble of Haar-like center-surround features. This classifier is learned from a single positive training example with AdaBoost and quickly updated for new object and background appearances with every frame. Saffari et al.[17] and Shi et al.[18] proposed the online random forest (RF) for the object tracking by continuous self-training of an appearance model while avoiding wrong updates that may cause drifting.

The first challenge in object tracking is to build an observation model. The color histogram is a well-known feature for object tracking because it is robust against noise and partial occlusion. However, it becomes ineffective in the presence of illumination changes or when the background and the target have similar colors.[16] A combination of color and edge features is also used for mutual complements.[8,13]

The second challenge in object tracking is to design an estimation of the likelihood (distance) between the target object and candidate regions. Several types of distances, such as histogram intersection or Euclidean distance, are used to compute the similarity between feature distributions.[8] The most popular method to estimate likelihood is using the Bhattacharyya coefficient as a similarity measure.[14,16]

The third challenge in object tracking is to recognize and track objects in images taken by a moving camera, such as one mounted on a robot or a vehicle, because this is much more challenging than real-time tracking with a stationary camera. In moving camera applications, the background is not static and the appearance, pose, and scale of a human vary significantly. To track humans in a moving environment, Jung and Sukhatme[19] proposed a probabilistic approach for moving object detection when using a single camera on a mobile robot in outdoor environments. Klein et al.[7] proposed an object tracking method based on particle filters by adapting new observation models for object and background appearances changing over time in moving camera. Leibe et al.[20] integrated information over long time periods to revise its decisions and recover from mistakes by considering new evidence from different camera environments (such as static or moving cameras) and large-scale background changes. Kalal et al.[21] proposed a tracking framework (TLD) that explicitly decomposes the long-term tracking task into tracking, learning, and detection. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates detector's errors and updates it to avoid these errors in the future.

However, since much of the human tracking research based on CCD cameras has many limitations, especially for dark indoor and outdoor environments owing to poor illumination, a few algorithms for tracking humans in thermal images have been tried.

Li and Gong[4] constructed the regions-of-interest histogram in an intensity-distance projection space model with a particle filter to overcome the disadvantage of insufficient intensity features in thermal infrared images. Padole and Alexandre[5] used two types of spatial and temporal data association to reduce false decisions for motion tracking with thermal images alone. Xu et al.[22] proposed a method for pedestrian detection and tracking with a single night-vision video camera installed on a vehicle. The tracking phase for the heads and bodies of pedestrians is a combination of Kalman filter prediction and mean shift.

Fernandez-Caballero et al.[23] proposed an approach to real-time human detection and tracking through the processing of thermal images mounted on an autonomous mobile platform. This method simply used static analysis for the detection of humans through image normalization and optical flow for enhancing the human segmentation in moving and still images.

However, there exist nonhuman target objects, such as buildings, cars, animals, and light poles, which have intensities similar to that of humans in thermal images.[4]

Therefore, it is very difficult to maintain correct tracking when humans overlap while crossing paths. To solve these problems, some recent tracking systems use additional information from color CCD cameras.[24] Leykin and Hammoud[1] proposed a system to track pedestrians by using the combined input from RGB and thermal cameras. First, a background model is constructed with color and thermal images. Then, a pedestrian tracker is designed using particle filters. Han and Bhanu[25] proposed an automatic hierarchical scheme to find the correspondence between the preliminary human silhouettes extracted from synchronous color and thermal image sequences for image registration without tracking. Cielniak et al.[24] proposed a method for tracking multiple persons with a combination of color and thermal vision sensors on a mobile robot. To detect occlusion, they proposed a machine learning classifier for a pairwise comparison of persons using both the thermal and color features provided by the tracker.

However, these human tracking methods based on thermal sensors or thermal and color sensors have the following typical disadvantages:

- A few algorithms[5,23] use the conventional background subtraction and intensity threshold to detect a candidate object for tracking.
- Many tracking methods[1,4,5,22–24] assume that the background is static and the target appearance is fixed.
- Even though a combination of thermal and color images aid human tracking in daylight, color images are useless in darkness.
- Combinations of thermal and color sensors impose additional costs for camera equipment and computation time.

To improve the human tracking performance for moving cameras while minimizing the computation time for darkness, this study proposes a novel human tracking approach for thermal videos that are based on online RF learning and combination of a local intensity distribution (LID) with oriented center-symmetric local binary patterns (OCS-LBP). As shown in Fig. 1, we design a real-time RF, which is the ensemble of decision trees for confidence estimation, and confidences of the RF are converted into a likelihood function of the target state. In the initial stage, the target model is selected by the user and particles are sampled. In the second and third stages, subblock-based RFs are generated using the long-term positive and negative examples with LID and OCS-LBP features by online learning. The learned RF classifiers are used to detect the most likely target position in the subsequent frame in the fourth and fifth stages. Then, the RFs are learned again by means of fast retraining with the tracked object and background appearance in the new frame.

This human tracking method based on RF combined with an LID and OCS-LBP allows human tracking to be performed in near real-time with a mobile thermal camera. Moreover, the tracking accuracy increases compared with that of a conventional human tracking method for thermal images.

The remainder of this paper is organized as follows. Section 2 describes the target representation method using LID and OCS-LBP features. Section 3 introduces the basic human tracking method using particle filters. Section 4 introduces the proposed human tracking method that incorporates online RF learning to avoid tracking drift caused by pose variations, illumination changes, and occlusion. Section 5 presents an experimental evaluation of the accuracy and applicability of the proposed human tracking method. Section 6 summarizes our conclusions and discusses the scope for future work.

2 Target Representation Using LID and OCS-LBP

To track a human, a feature space should be chosen for the target. Choosing an optimal feature of the target model is a more critical step because a thermal image has different characteristics than a color image. Therefore, we combine two appearance features: LID and OCS-LBP.

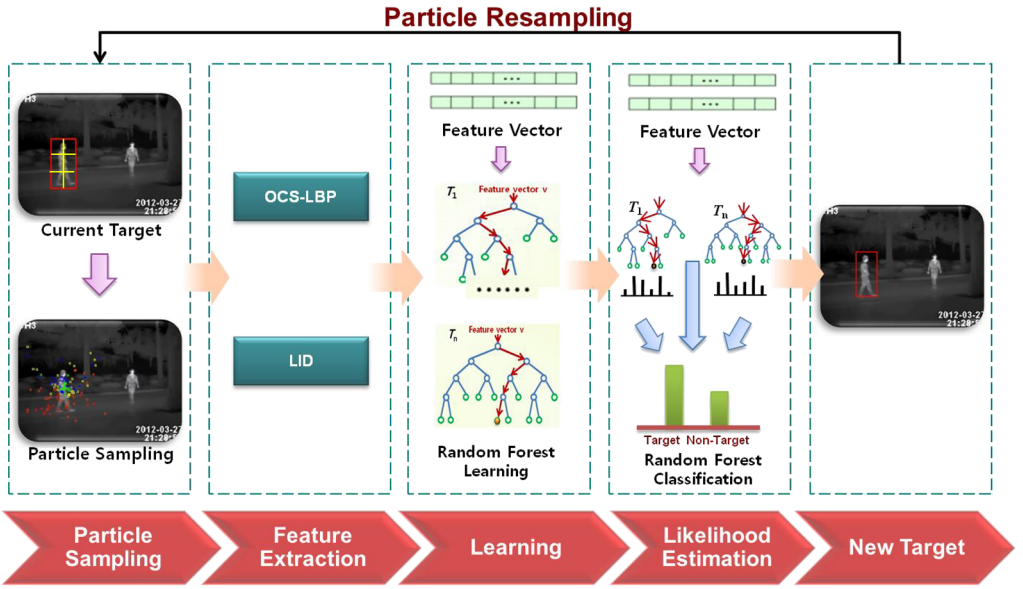


Fig. 1 Block diagram of the human tracking procedure using thermal images.

2.1 *Local Intensity Distribution*

A color histogram based on distance is a frequently used feature for object tracking.[4,10,14] However, the major characteristic of a human body in a thermal image is high intensity without color information, and individual humans exhibit distinct temperatures. Therefore, intensity is a better feature than color to distinguish humans from a background and other objects. In this research, we divide the bounding boxes of a target model and a target candidate into adjacent 3×2 subblocks to create a robust feature model for object occlusion as shown in Fig. 2. Partitioned subblocks are beneficial if the size of a box is relatively large. This was justified by the experiment of Khan et al.,[15] which revealed that an object often contains multiple local modes and that partitioned subblocks track objects more correctly than a single box when occlusion occurs.

In accordance with the research of Deguchi et al.[14] and Comaniciu et al.,[10] let $\{\mathbf{x}_j^i\}_{j=1\ldots n}$ be the normalized pixel locations in the i'th subblock defined as the target model. The normalized LID is represented by m-component (m-bin) histograms and the LID of the i'th subblock of the target model is denoted by $\mathbf{q}^i = \{q_u^i\}_{u=1\ldots m}$, where q_u^i is the u'th histogram component of i'th subblock. Since pixels will be more in the peripheral region than in the center, a normal histogram is affected by occlusions and interference from the background. Therefore, we use the Epanechnikov kernel $k(\mathbf{x})$, which is an isotropic kernel that assigns greater weights to pixels at the central points of the subblocks as follows:[13]

$$K(\mathbf{x}^i) = k(\|\mathbf{x}^i\|^2) = \begin{cases} 1 - \|\mathbf{x}^i\|^2 & \|\mathbf{x}^i\|^2 < 1 \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where the distance $\|\mathbf{x}^i\|^2$ is measured from the center point of the i'th subblock, $\mathbf{y}^i = (x_c^i, y_c^i)$. Each point $\mathbf{x}^i = (x_j^i, y_j^i)$ included in the i'th subblock is given by

$$\|\mathbf{x}^i\|^2 = \left\| \frac{\mathbf{y}^i - \mathbf{x}_j^i}{\mathbf{h}^i} \right\|^2 = \begin{cases} \left\| \frac{x_j^i - x_c^i}{h_x} \right\|^2 \\ \left\| \frac{y_j^i - y_c^i}{h_y} \right\|^2 \end{cases}, \tag{2}$$

where the bandwidth $\mathbf{h}^i$ means (h_x^i, h_y^i), the x and y radii of the subblock, respectively.

Finally, the probability of the features $u = 1, \ldots, m$ in the target model of i'th subblock is estimated as

$$q_u^i = C \sum_{\mathbf{x}_t^i \in R} k(\|\mathbf{x}^i\|^2) \delta[b(\mathbf{x}_t^i) - u], \tag{3}$$

where δ is the delta function, $b(\mathbf{x}_t^i)$ is the intensity component at $\mathbf{x}_t^i$, and R is the set of normalized coordinates within the subblock. For details on the normalization parameter C, refer to Ref. 10.

The LID of the i'th subblock of the target candidate centered at y in the current frame is denoted by $\mathbf{P}^i(\mathbf{y}) = \{p_u^i(y)\}_{u=1\ldots m}$, where $p_u^i(y)$ is the u'th histogram component of the i'th subblock. Using the same Epanechnikov kernel $k(\mathbf{x})$ and different bandwidth, depending on the size of the candidate box, the probability of the i'th subblock in the target candidate is estimated as

$$p_u^i(y) = C \sum_{\mathbf{x}_t^i \in R} k(\|\mathbf{x}^i\|^2) \delta[b(\mathbf{x}_t^i) - u]. \tag{4}$$

2.2 *Oriented Center-Symmetric LBP*

In human detection, texture features such as the histogram of oriented gradient (HOG)[26] and LBP (Ref. 27) are popular features to discriminate humans from backgrounds. Recently, the LBP texture operator has been successfully used in various computer vision applications, such as face recognition,[28] human detection,[29] and human tracking,[30] because it is robust against illumination changes, very fast to compute, and does not require many parameters.[31] LBP describes the gray-scale local texture of the image with low computational complexity by using a simple method. The original LBP descriptor forms different patterns based on the number of pixels by thresholding a specific range of neighboring sets with the central gray-scale intensity value. Even though LBP are widely used as a texture operator, they produce rather long histograms. Ma et al.[32] combined HOG and LBP to compute oriented LBP feature. First, they define the arch of a pixel as all continuous "1" bits of its neighbors. Then, the orientation and magnitude of a pixel is defined as its arch principle direction and the number of "1" bits in its arch, respectively.

CS-LBP (Ref. 33) uses a modified scheme comparing the neighboring pixels of the original LBP to simplify the computation while keeping the characteristics such as tolerance against illumination changes and robustness against monotonic gray-level changes. CS-LBP is different from LBP in that differences between pairs of opposite pixels in a

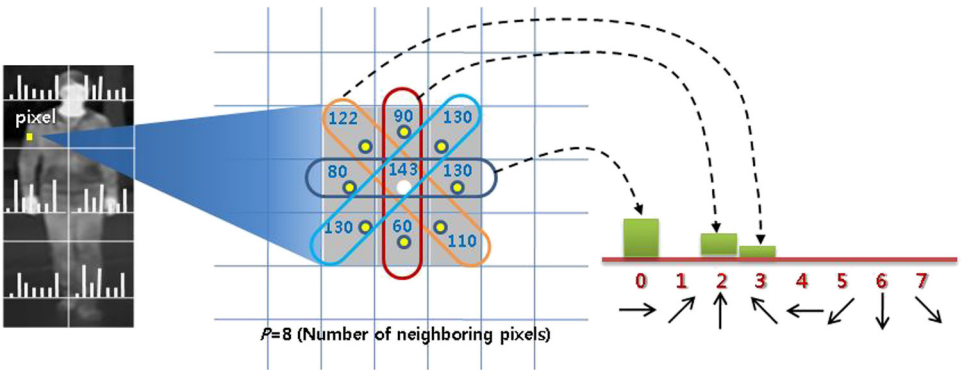


Fig. 2 Representation of oriented center-symmetric local binary patterns (OCS-LBP) histogram generation. Local OCS-LBP histograms generated from each subblock and their gradient orientation histograms.

neighborhood are calculated, rather than comparing each pixel with the center. This halves the number of comparisons for the same number of neighbors and produces only 16 (2^4) different binary patterns. However, since the original CS-LBP lose the orientation and magnitude information, we introduce a new lower-dimensional feature-oriented CS-LBP (OCS-LBP) using a different approach of oriented LBP.[32]

In order to extract an oriented histogram of OCS-LBP from a subblock, gradient orientations are estimated at every pixel and a histogram of each k'th orientation in a neighborhood is binned using Eqs. (5) and (6). Each pixel influences the gradient magnitude for an orientation according to the closest bin in the range from 0 to 360 deg at 45 deg intervals. In Eq. (6), robustness is maintained in flat image regions by thresholding the intensity-level differences using a small value T in Eq. (5), as follows:

$$s(x) = |x| \quad \text{if } |x| > T, \tag{5}$$

$$\begin{aligned} &\text{OCS} - \text{LBP}^{k}_{R,N}(x,y) = s(n_i - n_{i+(N/2)}) \\ &\quad [0 < i < (N/2) - 1, \quad k = 0 \ldots 7], \end{aligned} \tag{6}$$

where n_i and $n_{i+(N/2)}$ correspond to the intensity values of the center-symmetric pairs of pixels for N equally spaced pixels in a circle with radius R. Further, k is the bin number of the gradient orientation.

In Fig. 2, gradient orientation is confirmed when the differences between pairs of opposite pixels in a neighborhood are over the threshold. For example, the absolute difference between the values of n_0 (130) and n_4 (80) is over the threshold T and n_0 is greater than n_4, so the absolute difference (magnitude) is assigned to the zero bin. The gradient orientation histogram for each orientation k of a subblock is obtained by summing all the gradient magnitudes whose orientations belong to bin k. After that, the final set of k OCS-LBP features of a single subblock is normalized by the min-max normalization.

Using the same method with the LID, the bounding box of a target model and a candidate are divided into 3×2 adjacent subblocks and OCS-LBP histograms are extracted from each subblock. The number of subblock is decided according to the experiment results of Ref. 7. In Ref. 7, the target object was divided by nonoverlap 2×2 subblocks to make a robust target model about occlusion. However, we change the number of subblocks as 3×2 based on the human body ratio. All local OCS-LBP histograms are then used for online learning of the RF classifier.

3 Particle Filters

An object tracking algorithm based on particle filters[13] has drawn much interest over the last decade. This is a sequential Monte Carlo method, which recursively approximates the posterior distribution using a finite set of weighted samples. In addition, it weights particles based on a likelihood score and then propagates these particles according to a motion model.

Originally, particle filters consisted of the following three steps.[34]

Prediction Step

Given all available observations $y_{1:t-1} = \{y_1, \ldots, y_{t-1}\}$ up to time t–1, the prediction state uses the probabilistic system transition model $p(x_t|x_{t-1})$ to make a posterior prediction at time t.

$$p(x_t|y_{t-1}) = \int p(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1}. \tag{7}$$

Updating Step

At time t, the observation y_t is available, so the state can be updated using Bayes' rule.

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t) p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}. \tag{8}$$

The candidate samples $\tilde{x}_t^i$ are drawn from an importance distribution $q(\tilde{x}_t|x_{1:t-1}, y_{1:t})$ and the weights of the samples.

$$\omega_t^i = \omega_{t-1}^i \frac{p(y_t|\tilde{x}_t^i) p(\tilde{x}_t^i|x_{t-1}^i)}{q(\tilde{x}_t|x_{1:t-1}, y_{1:t})}. \tag{9}$$

In the case of bootstrap filters, particle weights are iteratively estimated from the observation likelihood.

$$\omega_t^i = \omega_{t-1}^i \cdot p(y_t|x_t^i). \tag{10}$$

Resampling Step

Since the probability of most samples is negligible, K particles having large weights are statistically selected more often than others, and the posterior state vector $\hat{x}_t$ is updated as the weighted average over the states of the particles.

$$\hat{x}_t = \sum_{j=1}^{K} \omega_t^j \cdot x_t^j. \tag{11}$$

4 Human Tracking Based on Online RF Learning

To estimate the observation likelihood $p(y_t|x_t^i)$ for the weighting of the particles, the Bhattacharyya distance is generally used by calculating the object appearance similarity.[10] In this paper, we estimate the observation likelihood for each particle by using an RF classifier instead of normal distance measures. Even though Saffari et al.[17] and Shi et al.[18] proved the robustness of object tracking by using online RF learning, two methods cannot avoid the template drift problem when images are taken by a moving camera because they only used the positive and negative samples from the current frame. In addition, because two methods only train one RF by considering full body region regardless of the extent of occlusion, they cannot track an object correctly in the case where an object has a severe occlusion.

Therefore, we design a new classifier with online RF learning with long-term samples as well as subblock-based RFs to avoid tracking drift caused by pose variation, illumination changes, and long-term occlusion.

4.1 *Initialization of Target Model*

Particle filters are sequential Monte Carlo methods that recursively approximate the posterior distribution using a

finite set of particles over time t. In this paper, we define the set of particles as $Pt_t = \{pt_t^i\}_{i=1..N}$, where the i'th particle pt_t^i at time t consists of its weight ω_t^i and state vector

$$\mathrm{pt}_t^i = [cx_t^i, cy_t^i, w_t^i, h_t^i, \omega_t^i]^T$$

where (cx_t^i, cy_t^i) is the center position of the tracked object, while w_t^i and h_t^i are the width and height of the bounding box of a target.

In the initial stage, the position and bounding box are manually selected and the state vector of the initial target $\mathrm{Tr}_1 = [cx_1, cy_1, w_1, h_1, RF_1]^T$ is then set automatically according to the user selection. RF_t^i is the classifier determined by online learning at time t.

4.2 State Prediction

In the prediction stage of the particle filters, N particles are propagated through the second-order autoregressive motion model[35] to predict the particle positions. The center position of the i'th particle is interpolated from the previous position (cx_{t-1}, cy_{t-1}), the average velocities at times t–2 and t–1, and white Gaussian noise $[G(0, \sigma_x^2), G(0, \sigma_y^2)]$.

$$\begin{aligned} vx_{t-2} &= cx_{t-2} - cx_{t-3},\ vy_{t-2} = cy_{t-2} - cy_{t-3} \\ vx_{t-1} &= cx_{t-1} - cx_{t-2},\ vy_{t-1} = cy_{t-1} - cy_{t-2}, \end{aligned} \tag{12}$$

$$\begin{aligned} cx_t^i &= cx_{t-1}^i + \left\lfloor \frac{vx_{t-2} + vx_{t-1}}{2} \right\rfloor + G(0, \sigma_x^2) \\ cy_t^i &= cy_{t-1}^i + \left\lfloor \frac{vy_{t-2} + vy_{t-1}}{2} \right\rfloor + G(0, \sigma_y^2). \end{aligned} \tag{13}$$

In the case of the second frame, only the velocity is linearly combined with the previous position and white Gaussian noise.

The box size of the i'th particle is linearly interpolated from the previous box size of the target object (w_{t-1}, h_{t-1}) and white Gaussian noise $[G(0, \sigma_w^2), G(0, \sigma_h^2)]$.

$$w_t^i = w_{t-1} + G(0, \sigma_w^2) \quad h_t^i = h_{t-1} + G(0, \sigma_h^2), \tag{14}$$

where $\sigma_x = \sigma_y = 6.4$ and $\sigma_w = \sigma_h = 0.64$ are used according to experimental results.[7]

4.3 Subblock-Based Random Forest Learning

An RF proposed by Breiman[36] is a decision tree ensemble classifier, with each tree grown using some type of randomization. This RF has a capacity for processing vast amounts of data, with high learning speeds, based on a decision tree.

For the learning of the initial RF, training data are constructed using a positive example that is selected by the user and two negative examples that are randomly sampled from the background of the first frame. In the second frame, the training data are increased to two positive examples and four negative examples. Negative samples are randomly selected from outside of a tracked object regardless of background cluttering. Training data are increased in the ratio 1:2 until 15 frames. The memory capacity is 15 for positive examples and 30 for negative examples. Every new target is added to positive memory and the RF is learned using the limited number of positive examples until the 15th frame. In contrast, negative examples are updated as the background at each frame according to the increase in frames. After the 15th frame, we always keep the five positive examples from the 1st through 15th frames in order to avoid the template drift problem by modifying the idea of Klein et al.[7] Moreover, the reminder of the positive memory is occupied by the new example and the oldest example is discarded, like in a queue, because the more similar history of the positive examples produces more confident classifiers.

In this research, each particle is divided into six subblocks as mentioned in Sec. 2.2, and two types of RF classifiers for the i'th subblock are learned using the LID and OCS-LBP extracted from the corresponding blocks in the 45 training examples.

Let F be the set of RFs $\{(rf_i)\}_{i=1...S}$, where S is the number of subblocks. The i'th RF, rf_i, is represented as $\mathrm{rf}_i = (\mathrm{srf}_i^{\mathrm{lid}}, \mathrm{srf}_i^{\mathrm{ocs}})$. Here, we construct two RFs, $\mathrm{srf}^{\mathrm{lid}}$ and $\mathrm{srf}^{\mathrm{ocs}}$, for each subblock: one uses only the LID feature ($\mathrm{srf}^{\mathrm{lid}}$) and the other uses only the OCS-LBP feature ($\mathrm{srf}^{\mathrm{ocs}}$), rather than combining these into one feature vector according to the experiments of Ko et al.,[37] because the basic characteristics of the LID and OCS-LBP are different. Therefore, the total number of RFs at time t is 12 (2 RFs × 6 subblocks).

The learning of the RFs in i'th subblock at time t is summarized below.

1. Set the number of decision trees T for two RFs.
2. Choose the number of variables for $\mathrm{srf}^{\mathrm{lid}}$ and $\mathrm{srf}^{\mathrm{ocs}}$. These variables are used to split each node from eight LID input variables and eight OCS-LBP input variables. By using different i'th variables, the split function $f(v_i)$ iteratively splits training data into left and right subsets.
3. Each tree for an individual RF is grown according to the following steps:
   a. Select n new bootstrap samples from training set B_n and grow an unpruned tree using the n bootstrap samples.
   b. At each internal node, each node selects m variables randomly and determines the best split function using only these variables.
   c. Grow the tree within the maximal tree depth.

When the class label set is denoted by $C = \{\mathrm{Positive}, \mathrm{Negative}\}$, a leaf node n has a posterior probability, and the class distributions of l trees, $p(C_i|l)$, are estimated empirically as a histogram of leaf nodes on a class label, C_i.

The depth of the trees is set at 20 according to the results of Ko et al.,[31] and the number of trees is four each for $\mathrm{srf}^{\mathrm{lid}}$ and $\mathrm{srf}^{\mathrm{ocs}}$. The experimental results for deciding the appropriate number of trees are described in Sec. 5.3.

4.4 Likelihood Estimation Using RFs

After a set of RFs is learned on positive and negative training examples of frame t, the observation likelihoods for each particle of frame t are estimated using RF classifiers. The reference feature histogram, the LID, and the OCS-LBP of the i'th subblock of a test particle are applied to the corresponding $\mathrm{rf}_i = (\mathrm{srf}_i^{\mathrm{lid}}, \mathrm{srf}_i^{\mathrm{ocs}})$. The likelihood of the i'th subblock is estimated by combining the probabilities of

srf_i^{lid} and srf_i^{ocs}. The test image is used as input to the learned RF, and the probability distribution (likelihood) of the i'th subblock in the positive class is generated by ensemble (arithmetic) averaging of each distribution of all trees $L = (l_1, l_2, \ldots l_T)$ using Eqs. (15) and (16).

$$P^i_{\text{lid}}(C_{\text{Positive}}|L) = \frac{1}{T}\sum_{t=1}^{T} P(C_{\text{Positive}}|lt), \tag{15}$$

$$P^i_{\text{ocs}}(C_{\text{Positive}}|L) = \frac{1}{T}\sum_{t=1}^{T} P(C_{\text{Positive}}|lt). \tag{16}$$

Hence, the final likelihood of a particle is estimated from Eq. (17).

$$P^j = \frac{1}{S}\sum_{i=1}^{S} (P^i_{\text{lid}} + P^i_{\text{ocs}})_n, \tag{17}$$

where S is the number of subblocks.

This process is continued iteratively until the likelihoods of all particles are computed.

Once the final likelihood (P^j) of the j'th particle is estimated, the weight (w_t^j) of the j'th particle at time t is replaced by using the likelihood P^j obtained from the RF and each weight is normalized.

$$\hat{w}_t^j = \frac{w_t^j}{\sum_{j=1}^{J} w_t^j}. \tag{18}$$

The state of the current target is updated as the top $J(= 15)$ particles having greater weight.

$$\bar{\text{Tr}} = [\bar{cx}, \bar{cy}, \bar{w}, \bar{h}]^T = \sum_{j=1}^{J} \hat{w}_t^j \cdot s^j. \tag{19}$$

4.5 *Online Relearning of RF Classifiers*

When a tracking human target is detected in a current frame, the RFs should be relearned using the updated history including positive and negative examples. The purpose of online RF learning is to avoid tracking drift caused by pose variation, illumination changes, and occlusion.

The basis of the proposed online RF learning is to compute the difference in target state between the current and previous targets and only relearn the RF for the current frame if this difference and the probability of the target satisfy the conditions. In this paper, the learning condition is adaptively changed by using Eq. (21) according to the variance in thermal intensity of a target region.

Online RF learning consisted of the procedures described below.

c_t^k: Center of a tracked target specified by the current state vector $\bar{\text{Tr}}$ at time t.

OC: Counter for duration check of full occlusion (OC = 0)

1. Compute the difference between centers of previous and current target regions.

$$\text{diff} = \sqrt{(c_t^k - c_{t-1}^k)^2}. \tag{20}$$

2. If diff < $T1$ // normal tracking.

   2.1 Compute the probability P_t of current target region by using RF.

   Where threshold $T1$ is the half width of the current target.

   2.2 Compute learning condition $U_t^{\max}$ using intensity variance σ^2 of target region and Eq. (21).

$$U_t^{\max} = \min\{0.72, [1/\log(\sigma^2)] \cdot \alpha\}, \tag{21}$$

$$U_t^{\min} = U_t^{\max} - 0.4, \tag{22}$$

   where 0.72 and 0.4 is the minimum probability for the learning condition. 0.4 is the control parameter for occlusion and it gives even large $U_t^{\min}$ values when it has a smaller value than 0.4. This value can be increased when camera is moving or multiple persons are walking. It gives even small $U_t^{\min}$ values when it has a larger value than 0.4. This value can be decreased when camera is static or only one person is walking. α is a constant learning rate decided empirically ($\alpha = 5.4$ in our tests).

   2.3 If $P_t > U_t^{\max}$

      2.3.1 Update 15 positive data and 30 negative training data.

      2.3.2 Learn the RF using training data.

      2.3.3 Resample particles using Eqs. (12), (13), and (14), except for 15 particles having high probability.

   2.4 If $P_t > U_t^{\min}$ // full occlusion

      2.4.1 Use previous RF without updating training data and learning RF.

      2.4.2 Replace current particles with previous particles.

      2.4.3 If full occlusion is continuous.

      Increase the number of full occlusion counter: OC = OC + 1;

      Else OC = 0;

   2.5 If $U_t^{\min} < P_t < U_t^{\max}$ // partial occlusion

      2.5.1 Use previous RF without updating training data and learning RF.

      2.5.2 Resample particles using Eqs. (12), (13), and (14), except for 15 particles having high probability.

3. If diff $\geq T1$ // abnormal tracking

   3.1 Use previous RF without updating training data and learning RF.

   3.2 Replace current particles with previous particles.

4. If OC $\geq T2$ // tracking terminal condition

   4.1 Tracking is terminated.

In procedure 2.2, we use the intensity variance of the target region to determine the learning condition. This condition is based on the fact that the probability of the RF on the current target is low as the intensity variance of the target is high. The test result for a minimum RF learning threshold of 0.72 is described in Sec. 5.1. In procedure 2.4.3, we check the duration of full occlusion and occlusion counter (OC) increases its number whenever continuous full occlusion is occurring. Then, if the total number of OC is over the terminal condition, tracking is terminated in procedure 4. The terminal condition T2 is a changeable threshold according to the application and we set it as 30 frames.

5 Experimental Results

To evaluate the performance of the proposed algorithm, we used four types of LWIR thermal videos containing moving object with background clutter, sudden shape deformation, unexpected motion change, and long-term partial or full occlusion between objects at night.

- Type I: Four thermal videos captured by a static camera in a dynamic background (OTCBVS benchmark dataset[38]).
- Type II: Four thermal videos captured by a static camera in a dynamic background.
- Type III: Two thermal videos captured by a moving camera.
- Type IV: Two thermal videos captured by moving and static cameras.

The frame rates of the video data varied from 15 to 30 Hz, while the size of the input images was 320×240 pixels. All test videos were captured in outdoor environments. Table 1 lists the detailed descriptions of the 12 test videos.

To evaluate the performance of the proposed method, we use the spatial overlap metric defined in Ref. 39. Let us define the concepts of spatial and temporal overlap between tracks as ground-truth (GT) tracks and system (ST) tracks in both space and time. After the ground truth and the estimated bounding box of the target in the i'th frame of a sequence are determined, the spatial overlap is defined as the amount of overlap $\mathrm{Area}(\mathrm{GT}_i, \mathrm{ST}_j)$ between GT_i and ST_j tracks in a specific frame k.

$$A(\mathrm{GT}_{ik}, \mathrm{ST}_{jk}) = \frac{\mathrm{Area}(\mathrm{GT}_{ik} \cap \mathrm{ST}_{jk})}{\mathrm{Area}(\mathrm{GT}_{ik} \cup \mathrm{ST}_{jk})}. \tag{23}$$

The initialization of the rectangle including the tracking object is manually selected by the user. The proposed human tracking system has been implemented in Visual C++ and tested using a PC with an Intel Core 2 Quad processor.

5.1 *Tests on Minimum Threshold and Condition for RF Learning*

In our study, the most appropriate minimum threshold to use in Eq. (21) for updating training data and RF learning was found to be 0.72 on the basis of several experiments. To determine the proper threshold for RF learning, four test data were selected from the test dataset shown in Table 1, namely, Videos 1 and 2 (OTCBVS data) and Videos 5 and 6 (our data). We selected the two videos from the OTCBVS data (i.e., Videos 1 and 2) because in these videos

Table 1 Properties of 12 test videos (S, static camera; M, moving camera).

Video type	Video sequence	Total frames	Description	Season
Type I (OTCBVS)	Video 1	300	Two persons walking in the woods (S)	Unknown, outdoors
	Video 2	274	Multiple persons walking in the street (S)	Unknown, outdoors
	Video 3	209	Multiple persons walking in the street (S)	Unknown, outdoors
	Video 4	733	One person walking in the yard (S)	Unknown, outdoors
Type II (our data)	Video 5	550	Two persons walking in the street (S)	Winter night, outdoors
	Video 6	400	Two persons walking in the yard (S)	Summer night, outdoors
	Video 7	197	Two persons walking in the street (S)	Winter night, outdoors
	Video 8	500	One person walking in the yard (S)	Summer night, outdoors
Type III (our data)	Video 9	338	Multiple persons walking in the yard (M)	Summer night, outdoors
	Video 10	880	Multiple persons walking in the yard (M)	Summer night, outdoors
Type IV (YouTube data)	Video 11	371	Multiple persons walking in the same direction (M)	Unknown, outdoors
	Video 12	196	One person walking in the cluttered background (S)	Unknown, outdoors

Note: Video (MPEG, 16.2 MB) [URL: http://dx.doi.org/10.1117/1.OE.52.11.113105.1].

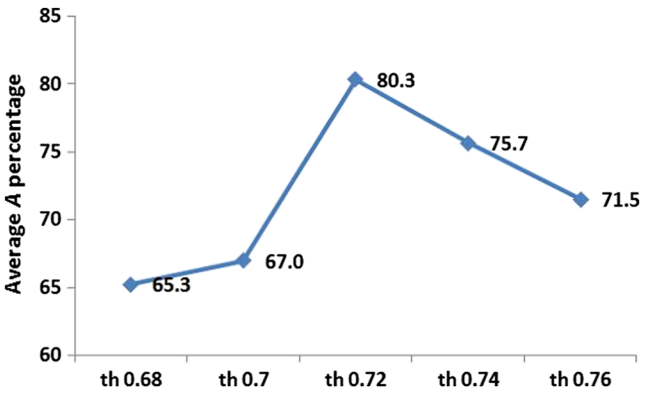


Fig. 3 Experimental results for five possible pairs to determine minimum threshold for updating training data and random forest (RF) learning.

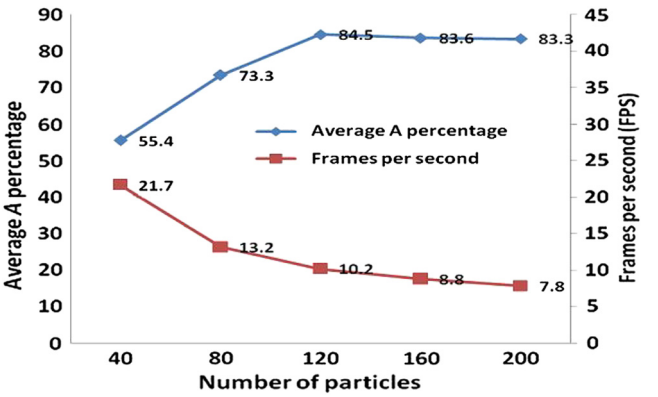


Fig. 5 Five possible pairs of experimental results to determine number of particles.

two persons walk in a cluttered background and become fully occluded by each other. We selected Videos 5 and 6 of our data because two persons walk in different directions and become fully occluded by a tree and each other. In the first experiment, the minimum threshold for RF learning was estimated by changing the value of the static threshold. As shown in Fig. 3, a minimum threshold of 0.72 for RF learning exhibited the best performance, with an average $A(\mathrm{GT}_{ik}, \mathrm{ST}_{jk})$ value of 80.3%. Therefore, 0.72 was adopted in Eq. (21) as the minimum threshold for RF learning.

For RF learning, we imposed the learning condition [Eq. (21)] using the minimum threshold. The purpose of the condition [Eq. (21)] is to design an adaptive RF classifier depending on the variation of intensity. To verify the performance of the learning condition, we compared the average A values of the static threshold determined in Fig. 3 with those of the learning condition [Eq. (21)] for the same four test data. As shown in Fig. 4, the adaptive learning condition exhibited the better performance for all four videos, with an average of 84.5 versus 80.3%.

5.2 *Determination of Optimal Number of Particles*

The main disadvantage of particle filters is the computational cost of using a large number of particles, even though particle filters are known to be robust in visual tracking through occlusions and cluttered backgrounds.[15] Therefore, it is essential to find the proper number of particles by considering the computational cost. Figure 5 shows the results of experiments using five possible values for the number of particles. As shown in Fig. 5, even though 40 particles gave the shortest processing time, the tracking performance was the worst. In contrast, 120 particles gave the best tracking performance and relatively good processing time, so 120 was adopted as the number of particle filters.

5.3 *Determination of Optimal Number of Trees*

The RF is known to be very fast in learning and testing as compared to other classifiers, i.e., the multiclass support vector machines.[31] The important parameters of the RF are the depth of the trees and the number of trees, T. Although increasing the depth of the trees and the number of trees improves the performance, the runtime cost depends on the depth of each tree and the number of trees. In our study, we set the maximum depth of the trees at 20 according to the experiments of Ref. 31.

To determine the proper number of trees for a local RF, we used the same four test data and compared the tracking performance by changing the number of trees. As shown in Fig. 6, when the number of trees for a local RF was four, the tracking performance was the best and the processing time was relatively good. Therefore, we adopted four trees for a local RF. In this study, we constructed two RFs per subblock: one uses only the LID feature and the other uses only the OCS-LBP feature, so the total number of trees for a target is 48 $(2 \times 4 \times 6)$.

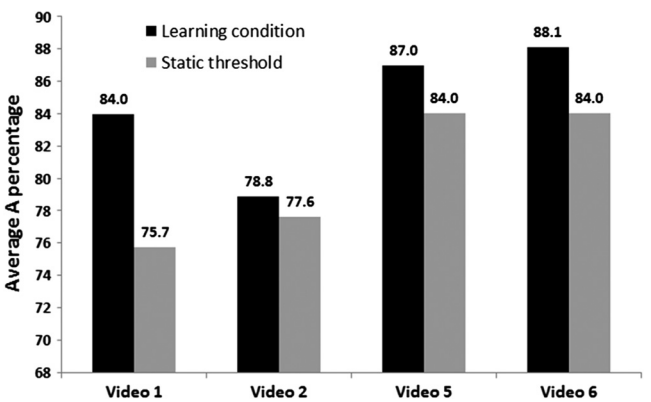


Fig. 4 Verification of performance of learning condition by comparing the values of the proposed updating condition and a static threshold.

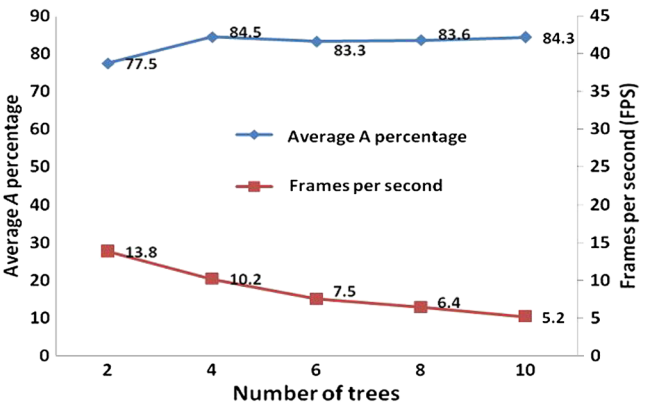


Fig. 6 Five possible pairs of experimental results to determine number of trees per local RF.

5.4 *Performance Comparison for Online RF Learning Versus Static RF*

Online RF learning is the main technique to track occluded humans and avoid tracking drift caused by pose variation, illumination changes, and occlusion in a cluttered background captured by a moving camera. To evaluate the effectiveness of the proposed online RF learning, we compared the tracking performance with online learning to that without online learning (static RF). A static RF is learned but once, when the user selects the human rectangle, and human tracking is performed by the static RF classifier without relearning.

Figure 7 shows a performance comparison of human tracking methods for the same four test data. As shown in Fig. 7, the online RF learning produced a better tracking performance with an average tracking success rate of 84.5% compared to 75.1%. The main reason for the higher tracking success rate of the proposed online RF learning is that the long-term full or partial occlusion between persons and tree is reflected in the training history and the RF learning.

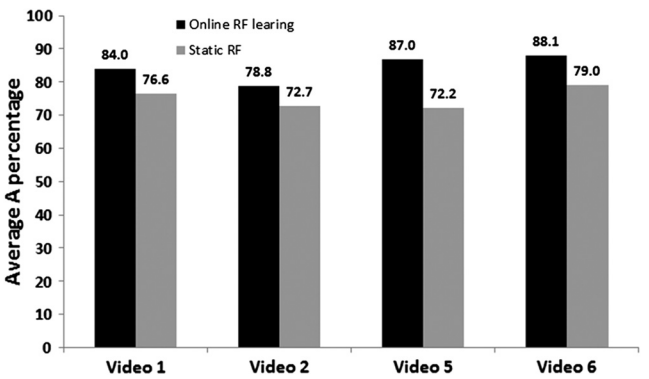


Fig. 7 Tracking performance using the proposed method improved by 9.4% when compared to that using the static RF method.

5.5 *Comparisons Between Different Algorithms*

To evaluate the performance of the proposed algorithm, the proposed method was compared with OCS-LBP with RF (OCS-LBP + RF) and LID with RF (LID + RF). In addition, we evaluate three different types of related works: (1) LID with particle filters using thermal image[4] (LID+ particle filters), (2) simple online RF learning using Haar-like feature,[17] (3) TLD tracker[21] that is known as a robust object tracking algorithm in a moving camera. The experiments were performed using the same dataset as described in Table 1. As shown in Fig. 8, the overall performance of our proposed approach exceeded that of the other two combinations, the particle filters,[4] simple online RF,[17] and TLD tracker,[21] based on the *A* percentages of 81.9, 69.6, 57.2, 69.9, 70.9, and 62.2%. From the results, we can infer that an individual intensity feature is not a distinguishing feature for human tracking in thermal images, particularly for cases of human occlusion. In contrast, the OCS-LBP feature produced reasonable tracking results even in thermal images. Even though simple online RF and particle filter produced the second and third best tracking performance of the other three methods, they still showed a few missing or false detection results when occlusions occurred. TLD tracker showed the worst tracking results, showing that learning and detection algorithm of TLD is not appropriate for human tracking in thermal image. The test results showed that for robust and practical tracking, the combination of two features is superior to the individual feature-based human representation model in thermal video.

For a more detailed evaluation of tracking performance, Fig. 9 shows comparisons between the proposed method and the two methods [particle filters[4] with LID and simple online RF (Ref. 17)] in terms of the *A* performance versus the frame number for Videos 1, 5, and 9. The ground truth of the target object is marked manually.

In Videos 5 and 9, the tracking method based on particle filters[4] and simple online RF (Ref. 17) lost the target object when occlusions occurred or camera is moving. In case of

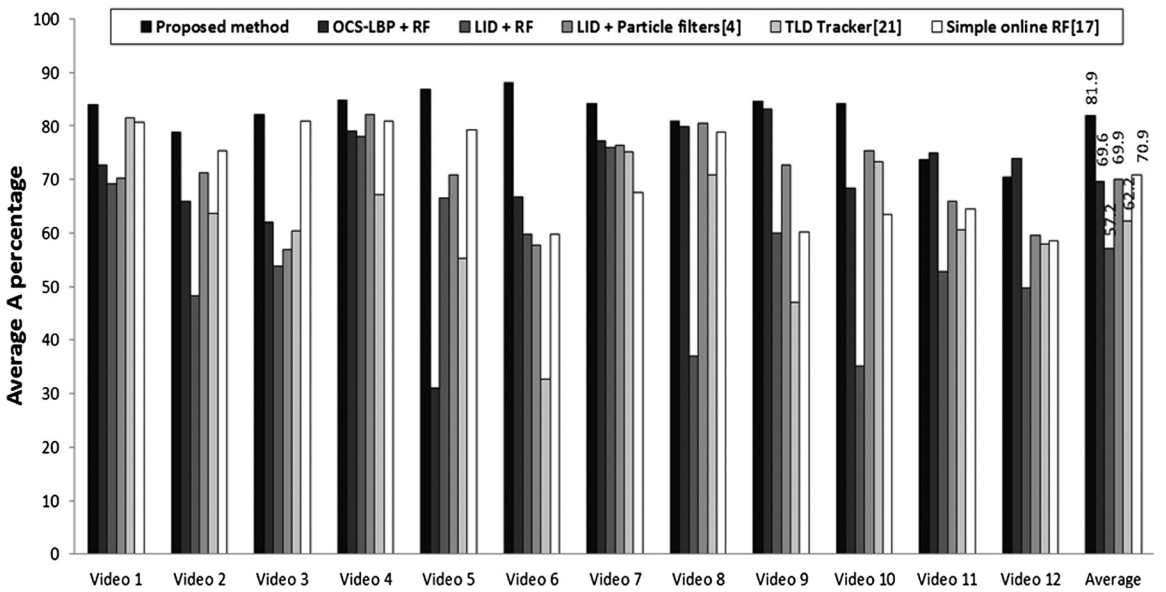


Fig. 8 Overall performance comparison of the proposed tracking algorithm with five other methods using the same dataset.

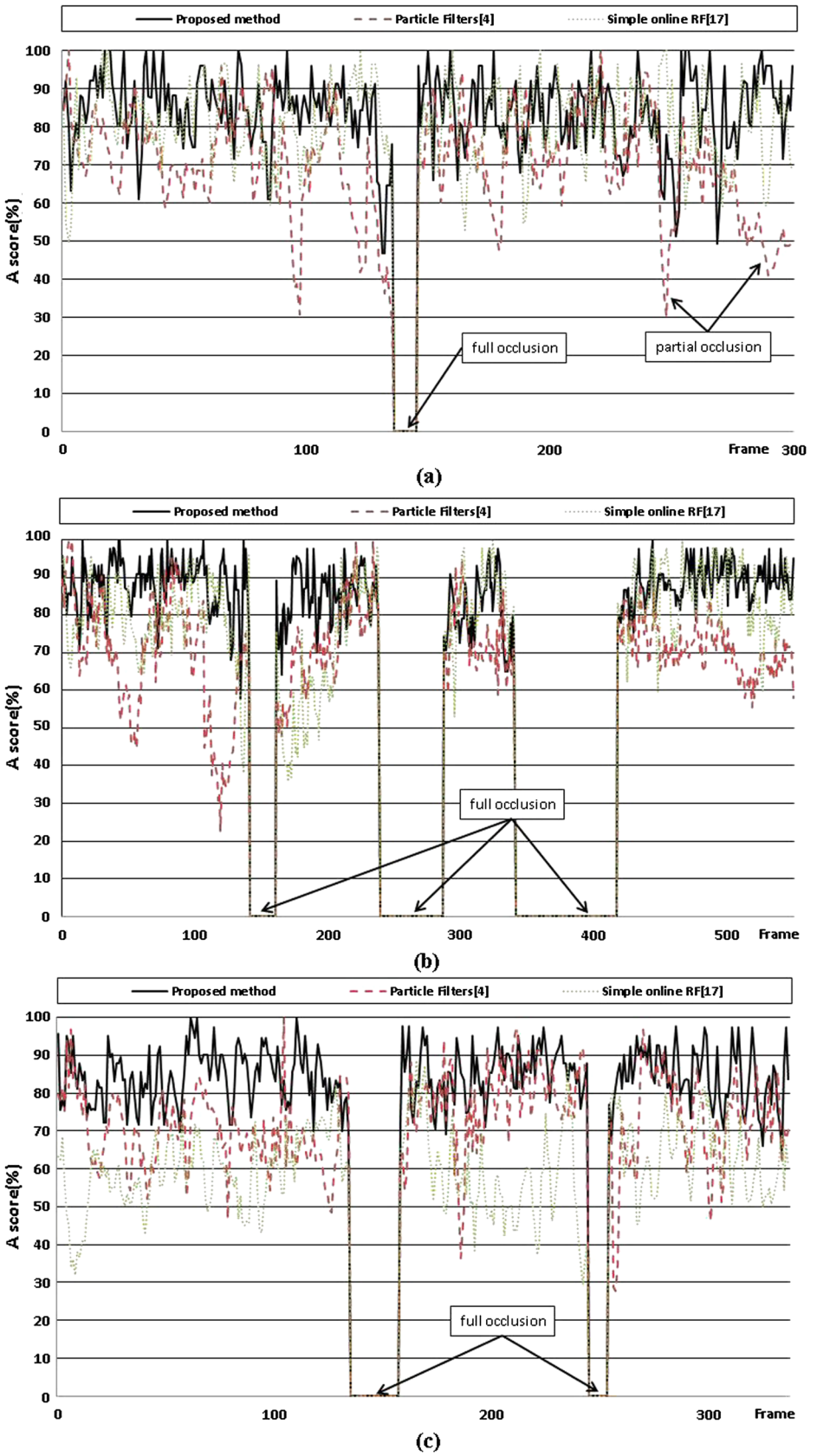


Fig. 9 Results of tracking the target object with three different methods using A score for (a) Video 1, (b) Video 5, and (c) Video 9.

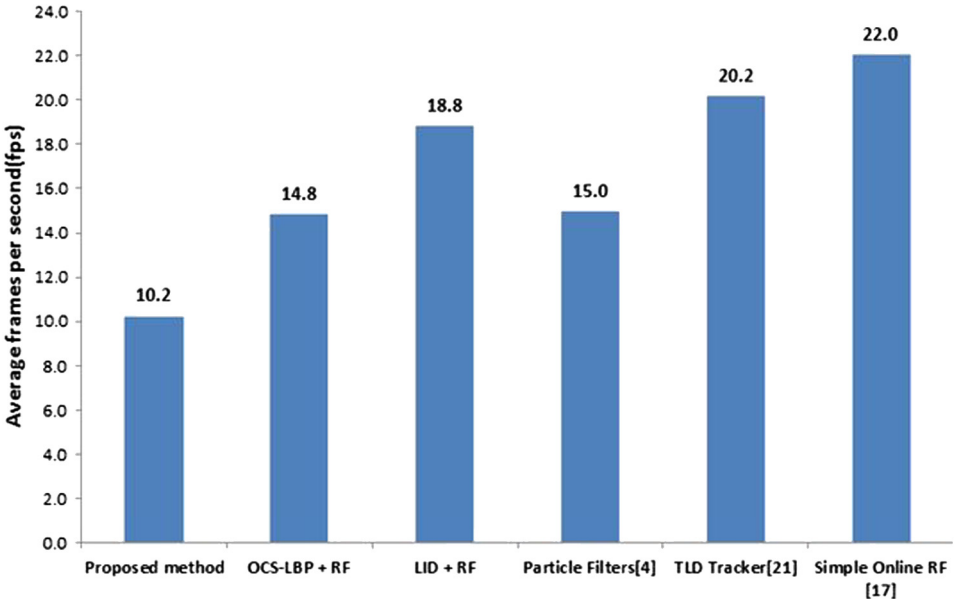


Fig. 10 Comparison between computational speeds of five methods.

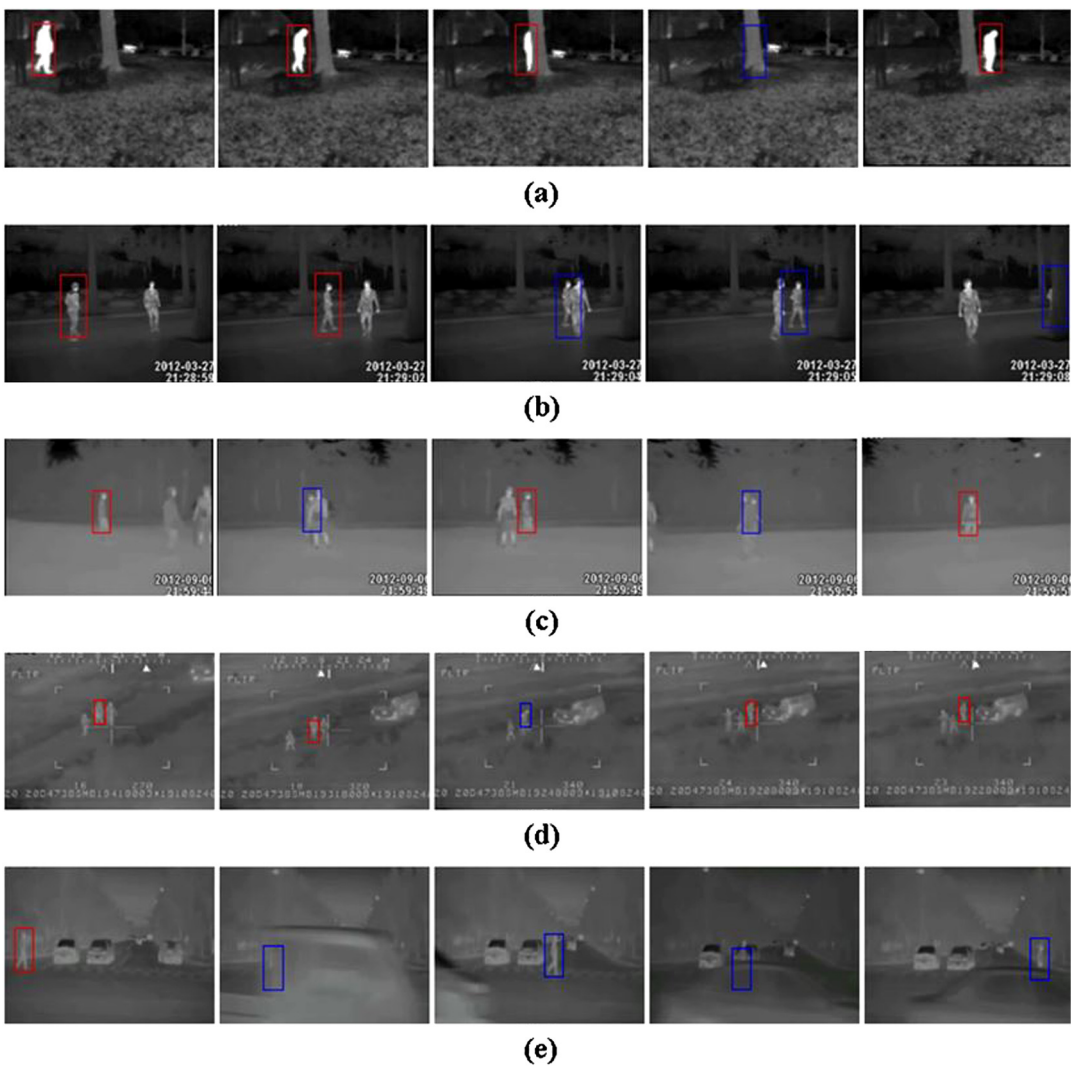


Fig. 11 Human tracking results obtained using the proposed method. Red boxes are regions tracked by online RF learning, and blue boxes are regions tracked by the static RF due to the partial or full occlusion.

simple online RF,[17] it showed the worst performance in Video 9 because it could not distinguish the real target from background when occlusions occurred. However, for all three videos, the proposed scheme had a significantly smaller error and more robust results than the other methods, regardless of the full or partial occlusion and the camera movement.

Figure 10 shows the computational speeds of the six methods. As shown in Fig. 10, the proposed method (at 10.2 fps) requires more computation time than the other methods (at 14.8, 18.8, 15, 20.2, and 22 fps) because it uses subblocks of particles, online RF learning, and two types of RF. When the online learning was not applied (static RF), the tracking speed was approximately the same as that of the LID + RF (~19 fps) and faster than that of the CS-LBP + LID using conventional particle filters. Simple online RF (Ref. 17) showed highest computational speed as 22 fps because it used simple learning RF with Haar-feature. Because the main reason for computational delay is the online RF learning for individual subblocks, optimization of the real-time learning may be considered for the next version.

Figure 11 shows the tracking results obtained for Videos 4, 5, 9, 11, and 12 by using our proposed method. From the results in Figs. 11(a) to 11(e), we deduce that our proposed method accurately and robustly tracks moving objects, despite background clutter with similar intensity distributions [(b), (c), and (e)], object intersections [(a) to (d)], long-term full (or partial) occlusion [(a) to (d)], and camera movement [(c) and (d)].

The complete video sequences can be viewed at the following webpage: http://cvpr.kmu.ac.kr.

6 Conclusions

In this paper, we have demonstrated that the proposed online RF learning method with particle filters improves human tracking performance for thermal videos, especially in cases of poor illumination, object occlusion, background clutter, and moving cameras.

To track a human region, an RF is relearned using the updated history, including positive and negative examples, whenever a new target is detected. Once a set of RFs is learned, the observation likelihood for each particle of frame t is estimated using RF classifiers. The proposed online RF learning computes the difference in target state between the current and previous targets, and the RF for the current frame is updated only if the difference and probability of the target satisfy the conditions. In this study, the learning condition was adaptively changed according to the variance of the thermal intensity of a target region.

This paper also proposed a new lower-dimensional OCS-LBP feature and proved that a combination of the OCS-LBP feature with the LID produces robust and practical tracking results from the individual feature-based human representation model, especially for thermal videos.

In the future, we plan to improve our human tracking algorithm to track multiple persons in dynamic environments by designing a faster learning algorithm with a small portion of particles and a robust feature model appropriate to thermal images.

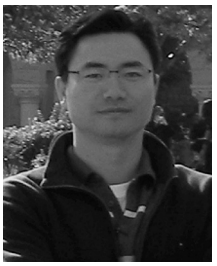
Acknowledgments

This research was partially supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology, and it was also financially supported by the Ministry of Education, Science Technology and National Research Foundation of Korea through the Human Resource Training Project for Regional Innovation.

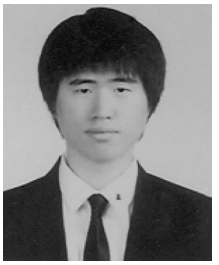
References

1. A. Leykin and R. Hammoud, "Pedestrian tracking by fusion of thermal-visible surveillance videos," *Mach. Vis. Appl.* **21**(4), 587–595 (2010).
2. A. Fernández-Caballero et al., "Real-time human segmentation in infrared videos," *Expert Syst. Appl.* **38**(3), 2577–2584 (2011).
3. M. Correa et al., "Human detection and identification by robot using thermal and visual information in domestic environments," *Int. J. Intell. Rob. Syst.* **66**(1–2), 223–243 (2012).
4. J. Li and W. Gong, "Real time pedestrian tracking using thermal infrared imagery," *J. Comput.* **5**(10), 1606–1613 (2010).
5. C. Padole and L. Alexandre, "Wigner distribution based motion tracking of human beings using thermal Imaging," in *IEEE Computer Vision and Pattern Recognition Workshops*, San Francisco, California, pp. 9–14, IEEE (2010).
6. J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Comput. Vis. Image Underst.* **106**(2–3), 162–182 (2007).
7. D. A. Klein et al., "Adaptive real-time video-tracking for arbitrary objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, pp. 772–777, IEEE (2010).
8. C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filters," in *IEEE Int. Conf. on Computer Vision*, Beijing, China, pp. 212–219, IEEE (2005).
9. F. Jurie and M. Dhome, "A simple and efficient template matching algorithm," in *IEEE Int. Conf. on Computer Vision*, Vancouver, Canada, pp. 544–549, IEEE (2001).
10. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 564–577 (2003).
11. A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys* 38(4), 1–45 (2006).
12. T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 90–99 (1986).
13. M. Isardm and A. Blake, "Condensation—conditional density propagation for visual tracking," *Int. J. Comput. Vis.* **29**(1), 5–28 (1998).
14. K. Deguchi, O. Kawanaka, and T. Okatani, "Object tracking by the mean-shift of regional color distribution combined with the particle-filters algorithm," in *Int. Conf. on Pattern Recognition*, Cambridge, England, pp. 506–509, IEEE (2004).
15. Z. H. Khan, I. Y. H. Gu, and A. G. Backhouse, "Robust visual object tracking using multi-mode anisotropic mean shift and particle filters," *IEEE Trans. Circuits Syst. Video Technol. Arch.* **21**(1), 74–87 (2011).
16. D. Sidibe, D. Fofi, and F. Meriaudeau, "Using visual saliency for object tracking with particle filters," in *European Signal Processing Conf.*, Aalborg, Denmark, pp. 1–5, IEEE (2010).
17. A. Saffari et al., "On-line random forest," in *IEEE Int. Conf. on Computer Vision Workshops*, Kyoto, Japan, pp. 1393–1400, IEEE (2009).
18. X. Shi et al., "Multi-cue based multi-target tracking using online random forests," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Prague, Czech, pp. 1185–1188, IEEE (2011).
19. B. Jung and G. S. Sukhatme, "Real-time motion tracking from a mobile robot," *Int. J. Social Rob.* **2**(1), 63–78 (2010).
20. B. Leibe et al., "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(10), 1683–1698 (2008).
21. Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409–1422 (2012).
22. F. Xu, X. Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Trans. Intell. Transp. Syst.* **6**(1), 63–71 (2005).
23. A. Fernández-Caballero et al., "Optical flow or image subtraction in human detection from infrared camera on mobile robot," *Rob. Auton. Syst.* **58**(12), 1273–1281 (2010).
24. G. Cielniak, T. Duckett, and A. Lilienthal, "Data association and occlusion handling for vision-based people tracking by mobile robots," *Rob. Auton. Syst.* **58**(5), 435–443 (2010).
25. J. Han and B. Bhanu, "Fusion of color and infrared video for moving human detection," *Pattern Recognit.* **40**(6), 1771–1784 (2007).
26. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, California, pp. 886–893, IEEE (2005).

27. T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognit.* **29**(1), 51–59 (1996).
28. T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European Conf. on Computer Vision*, Prague, Czech, pp. 469–481, Springer (2004).
29. D. Y. Kim et al., "Human detection using wavelet-based CS-LBP and a cascade of random forests," in *IEEE Int. Conf. on Multimedia and Expo*, Melbourne, Australia, pp. 362–367, IEEE (2012).
30. J. Ning et al., "Robust object tracking using joint color-texture histogram," *Int. J. Pattern Recognit. Artif. Intell.* **23**(7), 1245–1263 (2009).
31. B. C. Ko, S. H. Kim, and J. Y. Nam, "X-ray image classification using random forests with local wavelet-based CS-local binary patterns," *J. Digital Imaging* **24**(6), 1141–1151 (2011).
32. Y. Ma, X. Chen, and G. Chen, "Pedestrian detection and tracking using HOG and oriented-LBP features," in *Int. Conf. on Network and Parallel Computing*, Changsha, China, pp. 176–184, IEEE (2011).
33. M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.* **42**(3), 425–436 (2009).
34. M. S. Arulampalam et al., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002).
35. H. R. Byun and B. C. Ko, "Robust face detection and tracking for real-life applications," *Int. J. Pattern Recognit. Artif. Intell.* **17**(6), 1035–1055 (2003).
36. L. Breiman, "Random forests," *Mach. Learn.* **45**(1), 5–32 (2001).
37. B. C. Ko, J. Y. Kwak, and J. Y. Nam, "Wildfire smoke detection using temporal-spatial features and random forest classifiers," *Opt. Eng.* 51(1), 017208 (2012).
38. J. W. Davis, "OTCBVS benchmark dataset collection," Jan 2005, http://www.cse.ohio-state.edu/otcbvs-bench (16 September 2010).
39. F. Yin, D. Markris, and S. Velastin, "Performance evaluation of object tracking algorithms," in *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, Rio, Brazil, pp. 1–8, IEEE (2007).



Byoung Chul Ko received his BS degree from Kyonggi University, Korea, in 1998 and his MS and PhD degrees in computer science from Yonsei University, Korea, in 2000 and 2004. He was a senior researcher of Samsung Electronics from 2004 through 2005. He is currently an assistant professor in the Department of Computer Engineering, Keimyung University, Daegu, Korea. His research interests include content-based image retrieval, fire detection, and robot vision.



Joon-Young Kwak received his BS and MS degrees from Keimyung University, Korea, in 2011 and 2013. He is currently a PhD student of Keimyung University, Korea. His research interests include fire detection and human tracking.

Jae-Yeal Nam received his BS and MS degrees from Kyongbuk National University, Korea, in 1983 and 1985. He received his PhD degree in electronic engineering from University Texas at Arlington in 1991. He was a researcher of ETRI from 1985 through 1995. He is currently a professor in the Department of Computer Engineering, Keimyung University, Daegu, Korea. His research interests include video compression and content-based image retrieval.