Machine Vision and Applications (2009) 20:271–281
DOI 10.1007/s00138-008-0181-8

ORIGINAL PAPER

Detection of object abandonment using temporal logic

Medha Bhargava · Chia-Chih Chen · M. S. Ryoo · J. K. Aggarwal

Received: 24 January 2008 / Revised: 22 September 2008 / Accepted: 18 November 2008 / Published online: 15 January 2009
© Springer-Verlag 2009

Abstract This paper describes a novel framework for a smart threat detection system that uses computer vision to capture, exploit and interpret the temporal flow of events related to the abandonment of an object. Our approach uses contextual information along with an analysis of the causal progression of events to decide whether or not an alarm should be raised. When an unattended object is detected, the system traces it back in time to determine and record who its most likely owner(s) may be. Through subsequent frames, the system searches the scene for the owner and issues an alert if no match is found for the owner over a given period of time. Our algorithm has been successfully tested on two benchmark datasets (PETS 2006 Benchmark Data, 2006; i-LIDS Dataset for AVSS, 2007), and yielded results that are substantially more accurate than similar systems developed by other academic and industrial research groups.

Keywords Abandoned objects · Threat detection · Temporal logic · Public areas

M. Bhargava · C.-C. Chen (✉) · M. S. Ryoo · J. K. Aggarwal
Department of Electrical and Computer Engineering,
Computer and Vision Research Center,
The University of Texas at Austin,
Austin, TX 78712-0240, USA
e-mail: ccchen@mail.utexas.edu

M. S. Ryoo
e-mail: mryoo@mail.utexas.edu

J. K. Aggarwal
e-mail: aggarwaljk@mail.utexas.edu

Present Address:
M. Bhargava
CGGVeritas, Brentford, UK
e-mail: medha.bhargava@gmail.com

1 Introduction

In recent years, owing to the increasingly ubiquitous presence of cameras, the design of automatic surveillance systems for event recognition in crowded public areas has received much attention. The goal is to equip intelligent systems with the ability to reliably detect the possibility of danger. Such systems must prove their effectiveness in complex situations involving significant crowds, clutter and occlusion. They must be economically feasible and practically realizable in real-time, so as to be able to alert the authorities in a timely fashion to avert potential harm. Like all image processing frameworks, they must be able to successfully overcome the problems of lighting, viewpoint changes, noise and other distortions. The greatest challenge, perhaps, for such threat detection systems is to achieve a low rate of false positives and more importantly, a near-zero rate of false negatives.

This paper describes a powerful framework for a system that utilizes multiple spatio-temporal and contextual cues to detect a given sequence of events. Here, we tackle the specific threat posed by baggage abandoned in public areas. Our approach draws inspiration from the typical workings of a human operator. When a curiously unattended object becomes visible, the operator is likely to review the tape closely to determine how it came to be left there and to ascertain whether it has been abandoned or if its owner has simply stepped away momentarily. If the owner is still present in the scene, there may not be a reason to be concerned, but if he or she cannot be found, it is certainly a cause for the alarm.

Similarly, in our framework, if a lone object is discovered in the scene, the system tracks it backwards through recent video to look for its owner. The owner of the baggage is assumed to be the person who brings the object into the scene and sets it down at the location it is found. By inspecting the frames when the object was in contact with a human entity, distinctive features of its candidate owner(s) are acquired. These features are then used to search for the owner in subsequent frames. If no suitable match is found for a predefined period of time, the object is deemed as abandoned and an alarm is raised. If a match is eventually found (i.e. if the owner returns to the suspicious object), the alarm is defused.

The rest of this paper is organized as follows. We review recent works on detection of baggage abandonment in Sect. 2. A detailed description of our methodology is presented in Sect. 3. Strengths and shortcomings of our framework are demonstrated experimentally and some concerns are addressed in Sect. 4. Section 5 wraps up the paper with a summary of our work, its applicability and several interesting directions worth exploring in the future.

2 Previous works

The abandoned baggage problem has recently attracted considerable interests, and solutions have been attempted in many different ways, each inevitably with its own limitations. Several tracking models have been proposed based on a variety of techniques.

Lv et al. [11] combine a Kalman filter-based blob tracker with a shape-based human tracker to detect people and objects in motion. Event detection is set up in a Bayesian inference framework. Stauffer and Grimson [19] present an event detection module that classifies objects, including abandoned objects, using a neural network, but is limited to detecting only one abandoned object at a time. The probabilistic tracking model proposed by Smith et al. [18] is built of a mixed-state dynamic Bayesian network and a trans-dimensional Markov chain Monte Carlo (MCMC) method. Bhargava et al. [3] characterize the event of object abandonment by its constituent sub-events. Their algorithm verifies the sequence of foreground observations by pre-defined event representation and temporal constraints.

Adaptive background subtraction (ABS) has been a rather popular choice to detect unknown, changed or removed articles in the foreground. ABS methods, such as those described in [5,10], build and maintain a statistical model of the background, usually implemented in conjunction with an object tracker. Porikli [15] demonstrates static object detection using long-term and short-term backgrounds constructed using different adaptation rates. However, in general, ABS-based systems run the risk of integrating stationary foreground objects into the background before they are actually deserted. Their performance also suffers considerably from foreground clutter.

Much work has also been done on multi-view surveillance systems [2,12]. Such systems offer the significant merits of inferring the 3D spatial position of all objects, their depth, size and motion. Although such systems have been largely successful, the deployment of multiple cameras per location is usually not practical in wide spread public areas such as the railways. Our goal is to be able to utilize existing camera facilities for monitoring in public space, demanding little or no changes or additional expense. Thus, we limit our work to monocular image sequences.

3 Algorithm for event recognition

The proposed algorithm imitates the process flow of a human operator who decides whether someone has actually left an object at the scene or only stepped away momentarily. Our current approach to the problem is not based on individual tracking of all people and objects; instead, the system only searches for objects left by themselves. If an unattended object is detected, the system proceeds to look for its most likely owner and creates the owner's appearance model. An alarm is triggered if the owner is not found in the area for longer than a set period of time.

Our method is designed to capture and exploit the temporal flow of events related to the abandonment of an object. Adapted from Allen and Ferguson's classic temporal interval representation of events [1], the upper part of Fig. 1 illustrates the formal representation of the task. Their representational framework applies temporal interval logic to define the relationships between actions, events, and their effects. An event is defined as having occurred if and only if a given sequence of observations matches the formal event representation and meets the pre-specified temporal constraints. Allen's representation has been used extensively in a variety of applications, with some recent works by Ryoo and Aggarwal [16] and Nevatia et al. [13] for activity recognition using computer vision.

Likewise, we define the activity of abandonment of an object in terms of four sub-events that lead to it: the entry of the owner with the object, departure of the owner without the object, abandonment of the object and subsequent timed alarm, and the possible return of the owner to the (vicinity of the) object. Event inference follows from the detection of each of these sub-events or intervals, as depicted in Fig. 1. The sub-events to be recognized are causally related, i.e., the confirmation of one sub-event triggers the search for the next.

Our algorithm is composed of three computational modules which capture each of the sub-events and verify their temporal relationship as a whole. These modules are devised to perform the following sub-tasks: detection of unattended object(s), reverse traversal through previous frames to discover likely owner(s), and the continued observation of the scene. The process is preceded by a basic preprocessing stage that may vary depending on the dataset.

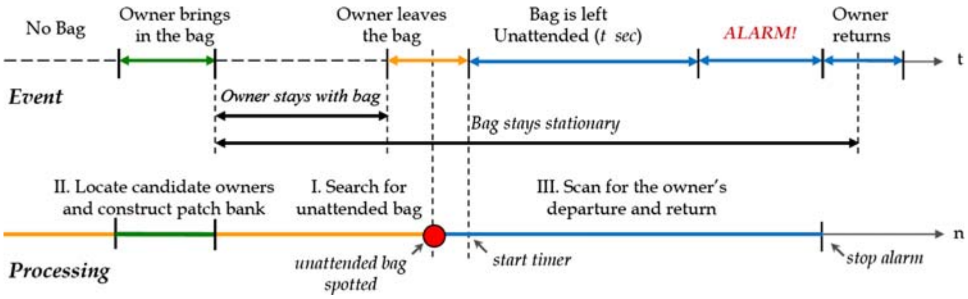

Fig. 1 Sequence of events (*top axis*) in time and progression of the system algorithm (*lowest axis*). Module I detects an unattended baggage and initiates Module II, which traverses through previous frames and collects appearance samples from candidate owners. Module III monitors the scene to verify the existence of the owner. A timer is set if the owner is not found, and an alarm is triggered if the owner fails to return within *t* s

To ensure clarity, the algorithm is described in terms of one abandoned object and one rightful owner. It must be noted that the framework can be extended to handle concurrently multiple abandoned objects and their corresponding owners. Also, since our unattended object detector is an independent module, it can be trained to pick out any kind of object by shapes, given a sufficient number of samples. Current experimental datasets were staged at a busy subway station and involve one baggage with one owner, or two people traveling together (in close proximity) with the baggage.

3.1 Low-level processing

Reliable low-level processing is crucial for any computer vision system. To acquire foreground blobs for later analysis, we perform background subtraction and several image preprocessing steps on each frame. To give the system a ready applicability in different scenarios, a background model is automatically estimated from the image sequence itself. The background initialization algorithm proposed by Chen and Aggarwal [4] is used to construct the background model. This algorithm is derived from [6], which identifies stable intervals of intensity values at each pixel, and determines which interval is most likely to display the true background based on local optical flow information. In [4], the critical process of parameter estimation is performed by approximating the scale of foreground activities under multiple resolutions. However, due to different foreground depths, objects closer to the camera will dominate the distribution of optical flow and the influence of background visibility from the farther objects can be overwhelmed. They discovered and solved this problem by equalizing the density of optical flow. Their method has been shown to yield impressive results indoors and outdoors.

Background subtraction is performed in the HSV color space, which inherently offers greater robustness to changes in illumination (such as the occurrence of shadows). A series of morphological operations is carried out to clean up the image, preserving only the blobs of interest (based on size and position). Subsequent processing deals exclusively with the resultant foreground blobs.

3.2 Detection of unattended objects

The goal of the first module of the algorithm is the detection of any object that seems to be unattended. The system does not track and monitor ongoing activities until the occurrence of such an event. The focus of this paper is on the detection of abandoned baggage in public places. Baggage may include suitcases, sports bags, rucksacks, backpacks, boxes, etc. The algorithm may be suitably tailored to identify other kinds of objects as well. The basis for anticipating the possible assortment of object classes is both site- and application-specific.

It is assumed that unattended baggage may be any baggage-like foreground blob that can be seen as distinctly separate from nearby blobs for at least a short period of time. The *k*-nearest neighbor (*k*-NN) classifier is used to classify foreground blobs in new frames as belonging to the baggage or non-baggage class. Baggage is defined as a solid, contiguous entity that usually does not exceed half the height of an average adult. Thus, classification is based on the size and shape of binary foreground blobs. However, the bag handle or handgrip poses a special problem by distorting the generic shape of the baggage. To avoid misclassification, morphological ‘open’ operations are performed on the binary foreground image using cross-shaped structural elements, which were found effective in removing the deforming handgrips while retaining the main body of the baggage.

The *k*-NN classifier is trained off-line using positive examples collected through Google Image Search. Negative examples used include non-baggage segments selected from the data sequences. The current system uses feature vectors from about 60 positive and 120 negative image samples. The following properties are used to characterize each training instance:

- Compactness: the ratio of area to squared perimeter (multiplied by 4π for normalization)
- Solidity ratio: the extent to which the blob area covers the convex hull area
- Eccentricity: the ratio of major axis to minor axis of the ellipse that envelopes the blob
- Orientation
- Size

The size of each binary blob is normalized to compensate for perspective distortion. To estimate the normalization weights from the training set, passengers' torso areas and the corresponding centroid y-coordinates are recorded. Using polynomial regression, a function of normalization weight versus y-coordinate is then computed.

A simple 3-nearest neighbor classifier is applied to detect baggage-like objects. Owing to the simplicity of the binary classifier and the features used, execution time is minimal. To verify the decision of the classifier, each suspicious baggage-like blob is tracked over a fixed number of consecutive frames (about eight) to ensure the consistency of detection and position. Temporary false classifications and moving baggage are rejected by employing temporal filtering. Thus, with the use of temporal filtering, sporadic false classification, especially due to moving baggage, is avoided. Once an object is identified as being unattended, the next module is initiated to search for its potential owner(s).

3.3 Reverse traversal for searching candidate owners

In crowded environments where baggage appears to have been abandoned, a human operator is likely to rewind the video to the 'drop-off' point. The operator then carefully observes the movements and behaviors of all passengers to gauge the most likely owner. This module of our system acts in much the same way. Once an unattended baggage is located, the system traces it through previous frames to search for the moment when the baggage was first brought to and placed at the detected location. The event of the owner setting down the baggage maximizes the likelihood of the owner's presence in the neighborhood of the baggage and, as a result, provides the system with the best timing for collecting the owner's appearance model.

Most of the backtracking stage is implemented in a straightforward manner to facilitate speedy traversal of the frames of interest, i.e. when the baggage was first visibly introduced in the immediate neighborhood of its detected location. Initial tracking is based solely on the location and size of the blob, regardless of its appearance. The presence of any blob of approximately the same size occupying the same spot as the detected baggage is assumed to indicate the presence of the baggage. This supposition may result in overshooting of the desired frames, which can occur in the event when the entry of the baggage at the position is not clearly visible. This method of matching based only on positional overlap accounts for instances of severe occlusion of the baggage, thereby reducing the chances of mistaking the wrong person(s) as the possible owner(s).

When no valid blob can be found at the anticipated area, it is inferred that the baggage was being moved and ought to appear elsewhere nearby. Note that while backtracking in time, the movement of the baggage corresponds to the past event of the owner arriving at the location with the baggage. The algorithm then performs template-matching using a similarity measure that combines grayscale image correlation with similarity of color to search for the baggage in the neighborhood region.

Normalized cross-correlation (NCC) [9] is extensively used in the field of image processing as a correlation measure for template matching, since it is invariant to changes in illumination. A map of correlation coefficients, γ, is computed according to Eq. (1) for every gray level image patch and a designated matching region of foreground. The template patch f is positioned at (k, l) of the foreground sub-image I. $I_{k,l}$ represents the foreground area covered by the patch. $\overline{f}$ and $\overline{I}_{k,l}$ are the mean intensities of the corresponding areas. $(k_{\max}, l_{\max})$ is the location of the peak in the cross-correlation matrix $|\gamma|$.

$$\gamma(k,l) = \frac{\sum_{x,y\in I_{k,l}} (I(x,y) - \overline{I}_{k,l})(f(x-k,y-l) - \overline{f})}{\left[\sum_{x,y\in I_{k,l}} (I(x,y) - \overline{I}_{k,l})^2 \sum_{x,y\in I_{k,l}} (f(x-k,y-l) - \overline{f})^2\right]^{\frac{1}{2}}} \tag{1}$$

The mean color of the sub-image window that corresponds to $I_{k_{\max},l_{\max}}$ is compared with the mean color of the patch in the HSV domain. The metric used to guage similarity, $c_{i,j}$, between any two HSV colors [17], $m_i = (h_i, s_i, v_i)$ and $m_j = (h_j, s_j, v_j)$, is shown in Eq. (2). Both $\left|\gamma(k_{\max},l_{\max})\right|$ and $c_{i,j}$ range between 0 and 1. The average of the two distances is used as a combined measure to quantify the degree of matching.

$$c_{i,j} = 1 - \frac{1}{\sqrt{5}}\Big[(v_i - v_j)^2 + (s_i\cos(h_i) - s_j\cos(h_j))^2 + (s_i\sin(h_i) - s_j\sin(h_j))^2\Big]^{\frac{1}{2}} \tag{2}$$

Two situations can arise from the outcome of correlation in a frame: either the baggage is found in the neighborhood or it is not. Detailed procedures for handling the two possibilities are discussed below.

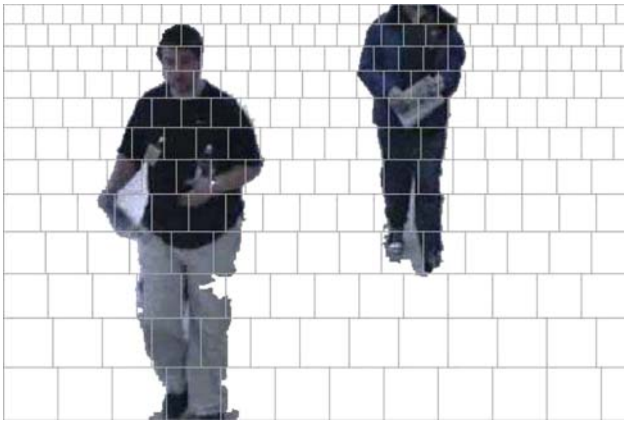

Fig. 2 The patch bank consists of informative image patches in the neighborhood of the back-tracked baggage, which are collected when the owner brings the baggage into the scene. The sampling grids are devised to coarsely account for perspective distortion

Situation 1: If the baggage is found, it may be inferred that the baggage was being moved or carried at the time, presumably by its owner. The foreground segments in its neighborhood can thus be considered as representatives of the candidate owners. To retain the appearance model of candidate owners, we define a patch bank comprising of a pool of foreground image patches. Rectangular patches are extracted from the desired foreground region, as shown in Fig. 2. Sampling grids are devised to account for perspective distortion so that each patch coarsely covers the same area. Patches contain only a small fraction of foreground area (less than 50% of the patch area) or are lack of texture (measured by entropy) are discarded. The system continues backtracking until the beginning of the video stream is reached or a sufficient number of patches are collected (within the updated neighborhood of the baggage).

Situation 2: In the case that the baggage cannot be found, it may be inferred that the time interval during which the owner set down the baggage has been overshot. Such an event may arise when the actual arrival of the baggage in the scene occurs in the presence of occlusion, or if someone (or something) else was beside the baggage when the real owner left. Conversely, in terms of reverse traversal, this is the situation when the movement of the baggage under inspection from its detected location goes unnoticed by basic template matching due to severe occlusion by another sufficiently large object. In such a case, the system attempts to relocate the baggage in the forward direction i.e. frame $(n+1)$.

In a crowded environment, pinpointing the exact owner of each unattended baggage blob can be a difficult task even for a human observer. Given that the chances of erroneous ownership assignment can be rather high in such a scenario, any attempts to zero in on a single individual automatically are best avoided. It is possible that the true owner comes in alongside one or more other persons, and given the view, the system may not be able to discern the actual owner accurately. In such a case, it would be more reasonable to attribute possession to all who could possibly be involved. For this purpose, it is important to estimate the individual who was the most likely originator of each patch and group patches accordingly. To establish connections between candidate owners and their corresponding patches, parent blobs from which each patch was derived are tracked. The patches are then labeled by tracked individuals.

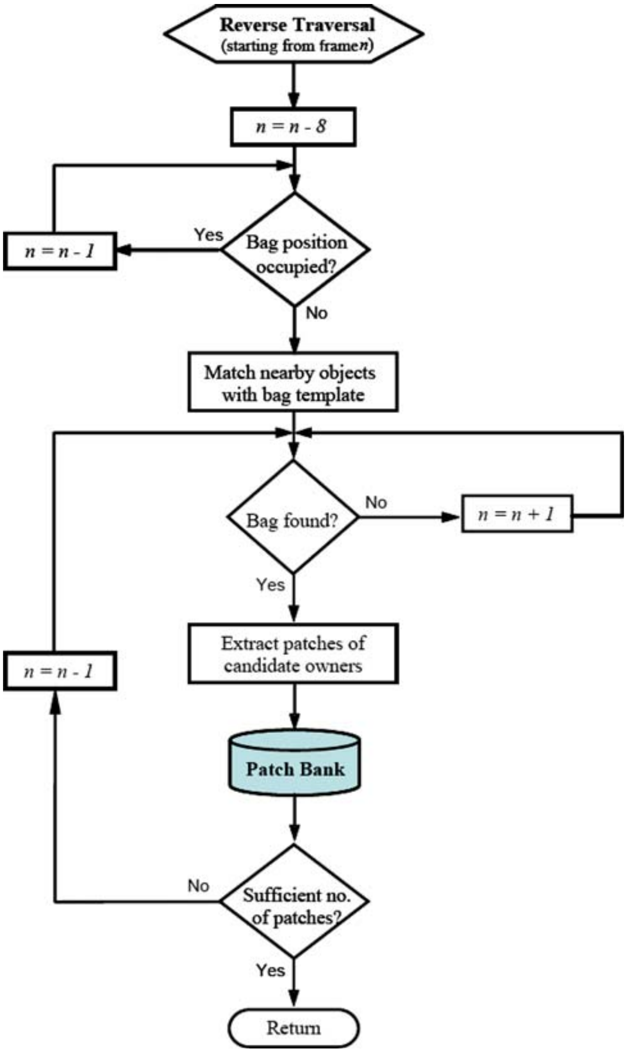

Fig. 3 Flowchart description of Module 2

3.4 Continued scene monitoring

The purpose of the third module is to monitor the departure and the possible return of the owner, which directly controls the switching on and off of the alarm. After constructing a representative patch bank, we return to the point when the baggage was identified as unattended, i.e. the present frame.

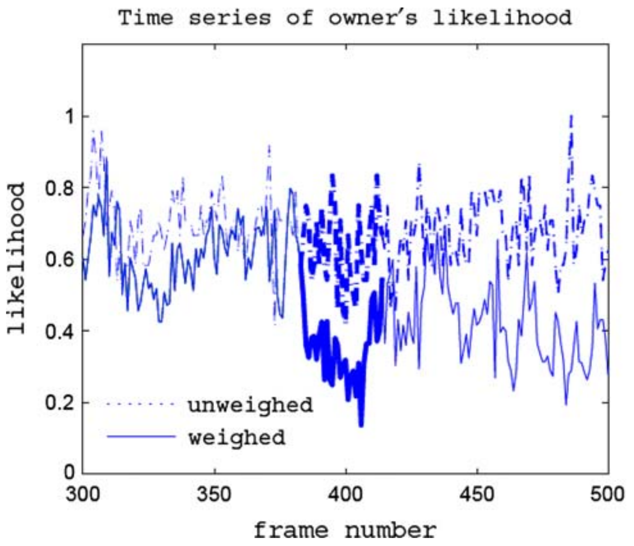

Fig. 4 Likelihood of the owner's existence as a function of time (after normalization). The *dashed curve* is generated by matching foreground images with unweighted patch bank while the *solid curve* is computed using the weighted patch bank as owner's template. The segments in *boldface* correspond to the interval of the owner's departure

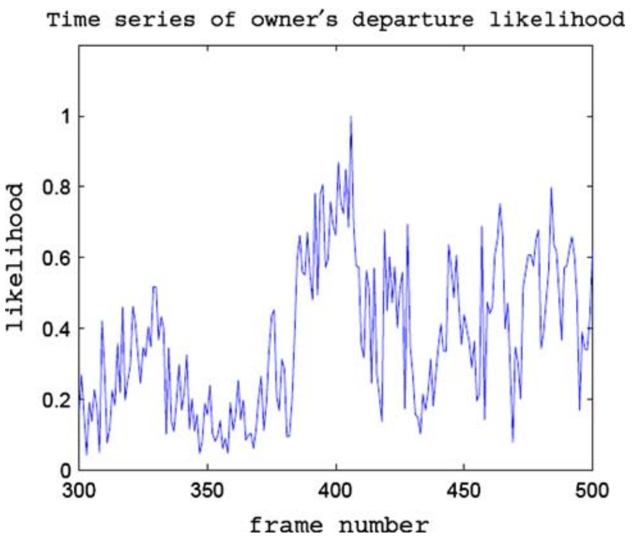

Fig. 5 Estimated by Eq. (3), the likelihood of the owner's departure is plotted as a function of time (normalized). The event of the owner leaving the scene happens between frames 383 and 414. The likelihood function reaches the maximum at frame 413

Looking forward in time from then on, the processing module tallies every foreground segment with the patch bank and computes the likelihood of the owner's existence in the area (referred to here as the owner's likelihood). By analyzing the likelihood as a function of time, the system is able to detect the owner's departure from the scene and sounds the alarm if he or she is missing for more than the predefined duration. The system continues monitoring the scene for the possible eventuality of the owner's return, and to consequently disable the alarm.

In order to search for candidate owners, every blob in the vicinity of the baggage is cross-correlated with the entire patch bank using the metric introduced in the previous section (the average of Eqs. (1) and (2)). We define the owner's likelihood in each frame by the maximum ratio of the match among all the appearance bins, where one appearance bin contains image patches extracted from the same individual. For each bin, the ratio of match is the weighted sum of the most similar patches in the bin to the total number of patches in the bin is evaluated. Patches are weighted according to their likelihood of being derived from the actual owner, as explained below.

The patch bank serves as a reservoir of candidate owners' appearance samples. However, to avoid the risk of designating specific ownership of the unattended baggage to one person, the patch bank may contain patches collected from different individuals who were in the neighborhood at the time. However, since each patch is treated as equally likely to be derived from the true owner, the presence of a false owner (but a candidate member) may generate a owner's likelihood value that is comparable to the value the actual owner. Therefore, in order to make the patch bank more responsive to the real owner, each patch is validated by the interval between two sub-events.

As shown in the top axis of Fig. 1, the owner stays with the baggage from the time it was set down to the time it was detected as unattended. To assign an appropriate weight to those patches which are more likely to have been extracted from the real owner, each patch in the bank is compared with the foreground sub-images in the neighborhood of the unattended baggage within that interval. It must be remembered that even patches derived from the actual owner may not find a hit in the bank due to change in effective viewpoint, arising from changes in the angular position of the person. Thus, the weight assigned to each patch is composed of a baseline score as well as a score that is proportional to the percentage of hits in the verification interval (Fig. 3).

Figure 4 shows two time series of the owner's likelihood, taken from the same interval of a testing sequence (Test6_i-LIDS). The dashed and solid function curves were generated using the unweighted and the weighted patch bank as the owner's appearance template respectively. The segment of the likelihood marked in boldface between frames 383 and 414 corresponds to the event of the owner leaving the scene. The departure of the owner is hidden behind intense fluctuations in the curve when the unweighted patch bank is used, while the same event corresponds to a consistent drop in the owner's likelihood using the weighted patch bank.

However, despite the weights, the presence of people who appear similar to the owner by virtue of clothing or viewpoint results in a significant amount of flux in the curve. Therefore, simple thresholding of the owner's likelihood using the weighted patch bank is, by itself, not reliable for the detection of the owner. Instead, the algorithm uses it to estimate the time of departure of the owner by analyzing the downward trend in the likelihood time series. The departure of the owner gradually reduces the number of hits in the weighted patch bank and results in a continuous fall in the owner's likelihood. To gauge the timing of the owner's complete absence from the scene, the likelihood decline in every fixed period is summed within a sliding window. The likelihood of the owner's departure, L_{dept}, is defined in Eq. (3). In this equation $L(n)$ is the owner's likelihood at frame n, w is the width of the sliding window, and t_u is the time the baggage found unattended. The owner's departure likelihood is characterized by two terms. The first term represents the amount of local likelihood decline, while the second compares the current likelihood with the average likelihood in the past. The departure likelihood is maximized when significant likelihood reduction is captured by both terms. The width of the sliding window determines the accuracy of the predicted timing. A very narrow window can result in greater sensitivity to false positives. The window length, w, is set to 2 s throughout our experiments. Figure 5 shows the owner's departure likelihood, which corresponds to the same interval highlighted in Fig. 4. In this example, the time of the owner's absence thus estimated is about 1 s earlier than the ground truth.

$$\begin{aligned} &L_{\text{dept}}(n) \\ &= \left| \sum_{k=n-w+1}^{n} \frac{\mathrm{d}}{\mathrm{d}k} L(k) \cdot \left\lfloor \frac{1}{2} \left(1 - \operatorname{sgn}\left(\frac{\mathrm{d}}{\mathrm{d}k} L(k)\right) \right) \right\rfloor \cdot \frac{1}{w} \right| \\ &\quad + \left| L(n) - \sum_{k=t_u-w+1}^{n-w} \frac{L(k)}{n-t_u} \right|, \quad \forall n > t_u \end{aligned} \tag{3}$$

Once the owner is detected as absent, a timer is set. If the timer lasts longer than t s, an alarm is switched on and the baggage is declared as abandoned. The system continues scanning for the owner in the scene, and computes the likelihood of the owner's return. We define the owner's return likelihood (L_{return}) in Eq. (4), where t_d is the time of the owner's departure estimated by Eq. (3). Similarly, the owner's return is detected when a substantial likelihood increment is observed in both terms of the equation. However, the reappearance of the owner does not necessarily imply that the baggage will be claimed. Therefore, the timer (and alarm) is not disabled until the displacement of baggage unveils the covering background. As soon as the timer is stopped, the processing module scans backward in time to locate the accurate timing of the owner's coming back by referring to the recorded return likelihood.

$$\begin{aligned} &L_{\text{return}}(n) \\ &= \left| \sum_{k=n-w+1}^{n} \frac{\mathrm{d}}{\mathrm{d}k} L(k) \cdot \left\lfloor \frac{1}{2} \left(1 + \operatorname{sgn}\left(\frac{\mathrm{d}}{\mathrm{d}k} L(k)\right) \right) \right\rfloor \cdot \frac{1}{w} \right| \\ &\quad + \left| L(n) - \sum_{k=t_d+1}^{n-w} \frac{L(k)}{n-w-t_d} \right|, \quad \forall n > t_d + w \end{aligned} \tag{4}$$

4 Experimental results

To demonstrate the performance of our system, 15 video sequences extracted from PETS 2006 dataset [14] and the Imagery Library for Intelligent Detection Systems (i-LIDS) [8] are tested. These datasets are made publicly available for research and educational purpose by the UK Information Commissioner and the UK Home Office Scientific Development Bran-ch, respectively. The testing set of our experiment includes six sequences from PETS 2006 and nine sequences from i-LIDS (the source dataset of AVSS 2007 special session). Each sequence contains one complete event of baggage abandonment except for S3C3 and S6C3 of PETS06. These public datasets were recorded from different subway stations in UK. The tested PETS sequences were all taken from the same camera (there are four viewpoints available). i-LIDS offers three training sequences and six competition sequences [7]. The training sequences are labeled by their projected level of difficulty and the contest ones are named in order of occurrence.

The 15 sequences tested involve different degrees of scene density, baggage size and type. The greatest challenges of the crowded subway setting are severe occlusion, considerable perspective distortion, and lighting changes. In addition, most people in the videos wear dark-colored clothes, jackets and coats, which pose extra difficulties for the system to discern between individuals. For example, the passing by of passengers with similar apparel to the owner in question interferes with the likelihood of the owner's existence. As a consequence, the the timing of the alarm was misled at times. Figures 6 and 7 demonstrate the sequential process of Test4_i-LIDS and S4C3_PETS06, respectively. The snapshots in both figures correspond directly to the progression of sub-events as described in Fig. 1.

In our experiment, we follow the rules defined by the individual conferences. For example, in PETS 2006, baggage is declared as abandoned when the owner is missing for 30 s, while the same duration is set to be 60 s in AVSS 2007. To facilitate faster processing, image sequences are downsampled by 4. The parameter settings for each benchmark dataset are fixed. Our results are compared with the ground truth data in Table 1. Except in sequences Test1_i-LIDS and

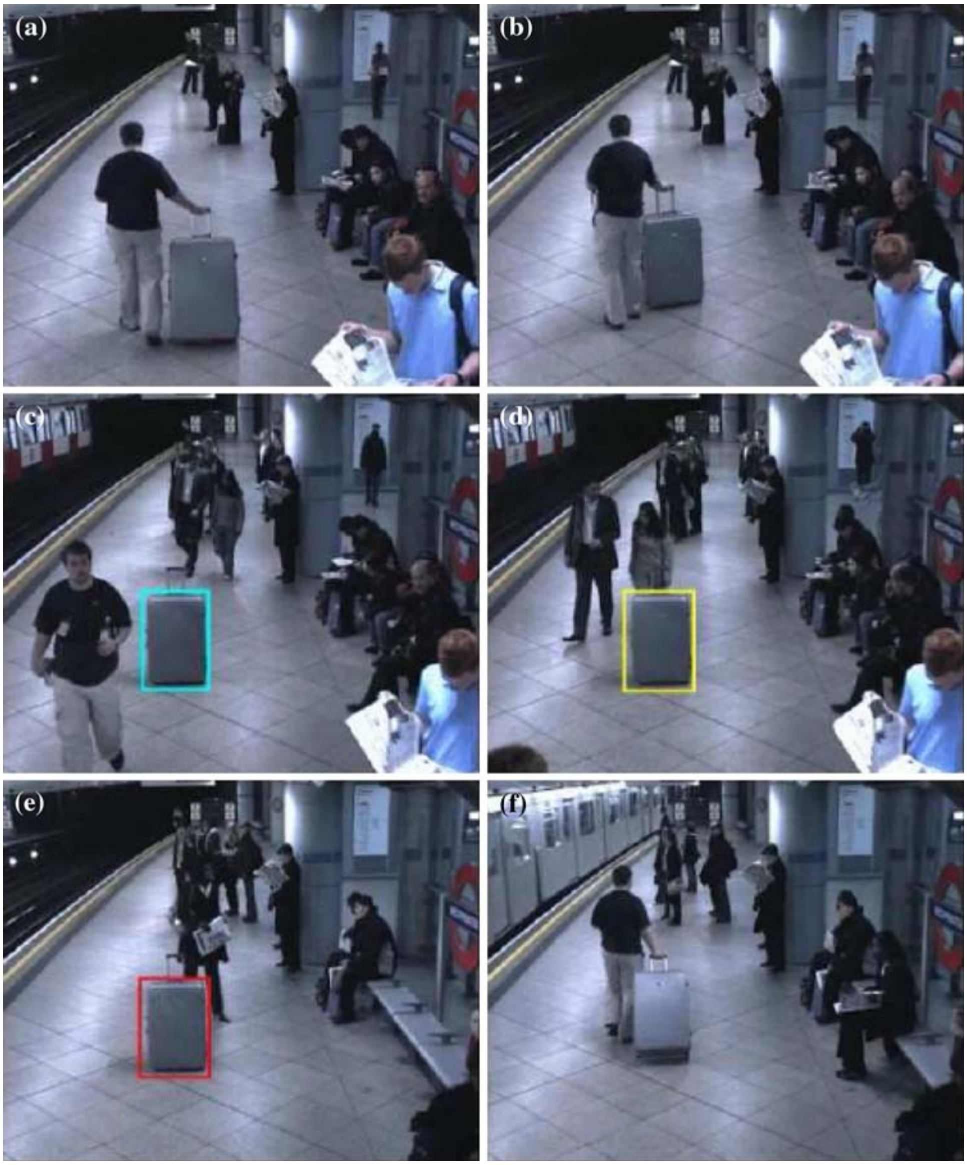

Fig. 6 Results of processing sequence Test4_i-LIDS. The unattended baggage is first identified in **c**, which initiates reverse traversal up to **b** where the baggage is not found at the detected location. Tracking the moved baggage backward in time to **a** where a sufficient number of the owner's appearance patch is collected. Returning to the point when unattended baggage is discovered, **d** shows the complete departure of the owner, which enables the timer. 60 s later, the alarm is triggered (**e**), and is eventually set off (**f**) by the owner's return

Test5_i-LIDS, our system is successful in estimating the timing when the baggage was left unattended. The activation of alarm and the owner's return are within 2–4 s of the ground truth on average.

There are certain circumstances under which our system fails. This happens largely due to the unavailability of crucial cues and segmentation problems. An instance of this can be seen in Test1_i-LIDS. Figure 8a is a snap shot of which shows the owner's departure (encircled in red) from the occluded baggage (outlined in blue). In this particular sequence, both the owner's entrance with the baggage and his exit without it can be barely detected even by a human observer. The baggage was not discovered until the complete dispersion of the crowd. Subsequent timing of the owner's departure was erroneously predicted due to inaccessibility of the actual owner's appearance patches. As a result, the alarm was not activated before the return of the owner. Similarly, in Test2_i-LIDS, the baggage was found unattended later than the ground truth because the baggage could not be segmented out until the departure of the passengers around it. Test5_i-LIDS, as shown in Fig. 8b, illustrates another special case. Here, the baggage and the person sitting next to it are both blended into the same foreground blob. This arrangement remains until the end of the sequence, and thus, the *k*-NN baggage classifier fails to detect it.

5 Conclusions and future work

Public safety is a critical issue in our world today. Through the assistance of automatic threat detection systems, security personnel may be equipped with instant and comprehensive

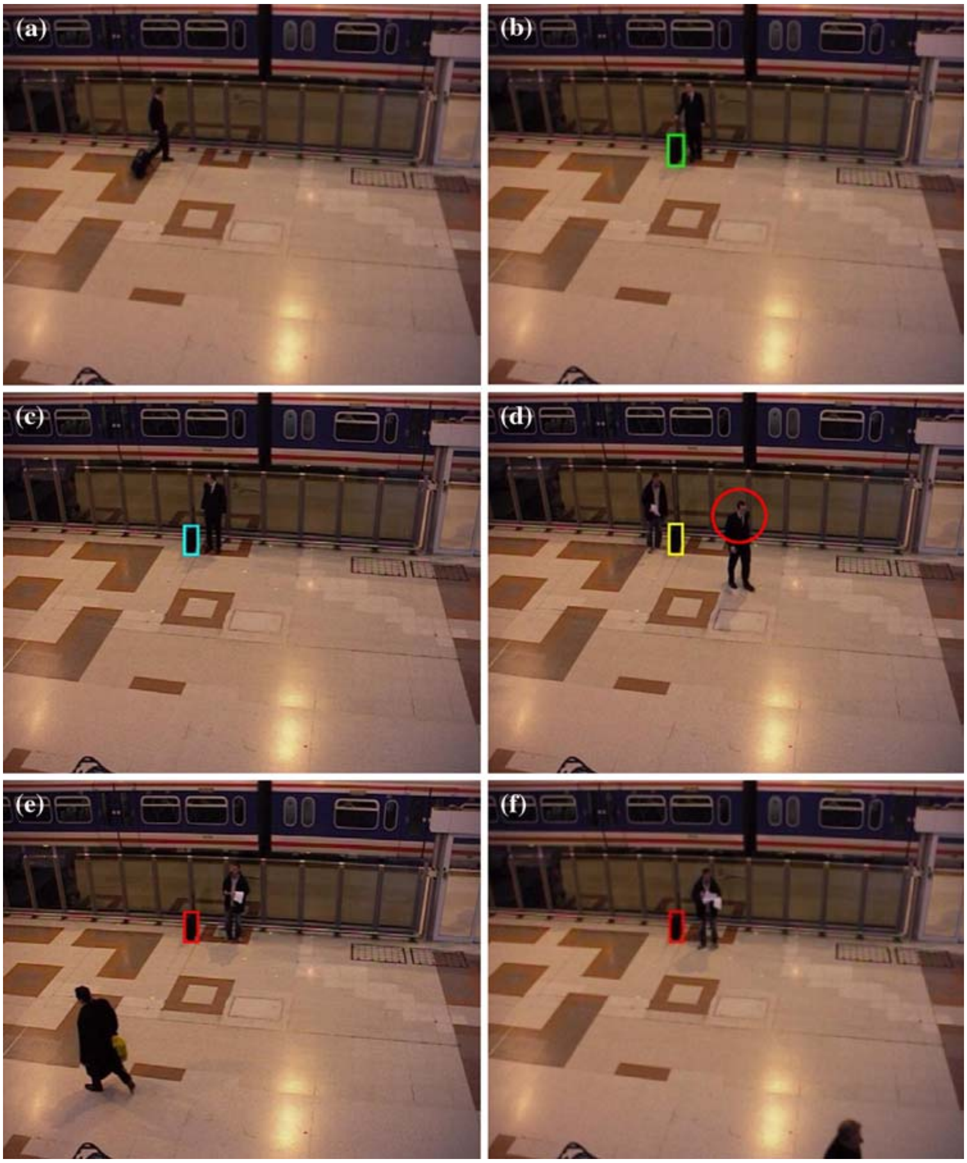

Fig. 7 Snapshots of the process of sequence S4C3_PETS06. In **c** the baggage is not in physical contact with the owner, and is locked on by the unattended object detector. The system traverses backward in time to search for the initial presence of the baggage. **b** to **a**, image patches are collected from the foreground area around the detected baggage. In **d**, the departure of the owner (marked by *red circle*) from the baggage neighborhood (defined in PETS 2006) starts the timer. Since the owner never comes back to the scene after his leaving, the alarm continues from **e** to **f** (the last frame) (colour in online)

awareness of potential crises. In this paper, we introduce a general framework to recognize the event of object abandonment in a busy scene. The proposed algorithm is characterized by its simplicity and intuitiveness, and is demonstrated to be highly effective on benchmark datasets. It is capable of handling concurrent detection of multiple abandoned objects, in the presence of substantial occlusion, and perspective distortion. The algorithm lends itself naturally to the recognition of a vast variety of related activities, ranging from surveillance of abnormal activities, corridor monitoring to traffic and cargo management. The modular structure allows the flexibility of integrating more functionality without remodeling the framework.

The performance of our algorithm is impressive; however, its robustness in certain problematic contexts must be improved. For example, if people all wear dark and textureless clothes in the monitored area, fewer patches would be extracted from the candidate owners, and would be less discriminating. It would be beneficial to use the spatial correlation between patches as another constraint in our patch-based matching scheme. The entire constellation of matched patches would then have to be justified by the owner's patch bank.

Segmentation is another challenging issue, including foreground as well as object segmentation. For better foreground segmentation, it would be worth exploring techniques of adaptive background modeling, or a mechanism for switching among pre-stored background models (backgrounds of the platform with and without the train, for example). The ability to separate merged or occluded foreground objects would further increase the accuracy of our system. With prior knowledge of the class of objects, class-based segmentation

Table 1 Abandoned baggage detection results

Sequence	Bag left unattended		Alarm starting time		Owner's return	
	Ground	Our	Ground	Our	Ground	Our
S2C3_PETS06 (00:00–01:25)	00:51	00:54	01:21	01:24	Never return	No return
S3C3_PETS06 (00:00–01:19)	Always attended	Always attended	Never started	Not activated	Never return	No return
S4C3_PETS06 (00:00–00:41)	01:01	01:03	01:31	01:33	Never return	No return
S5C3_PETS06 (00:00–00:53)	01:06	01:10	1:36	01:40	Never return	No return
S6C3_PETS06 (00:00–00:46)	00:27	00:29	Never started	Not activated	Never return	No return
S7C3_PETS06 (00:00–00:56)	00:26	00:26	00:56	00:56	Never return	No return
Easy_i-LIDS (00:00–04:08)	01:54	01:55	03:00	02:59	03:12	03:12
Med_i-LIDS (00:00–04:08)	01:40	01:46	02:42	02:46	03:00	03:00
Hard_i-LIDS (00:00–04:08)	01:40	01:43	02:42	02:43	04:06	04:07
Test1_i-LIDS (00:00–04:08)	02:27	03:11	03:29	Not activated	03:46	03:48
Test2_i-LIDS (04:08–07:10)	05:47	06:02	06:48	07:02	07:03	07:06
Test3_i-LIDS (07:10–10:57)	09:17	09:17	10:19	10:19	10:41	10:43
Test4_i-LIDS (10:57–14:33)	13:05	13:05	14:07	14:05	14:21	14:24
Test5_i-LIDS (14:33–18:13)	16:30	Not detected	17:34	Not activated	18:01	Not detected
Test6_i-LIDS (18:13–21:45)	20:03	20:03	21:07	21:09	21:36	21:40

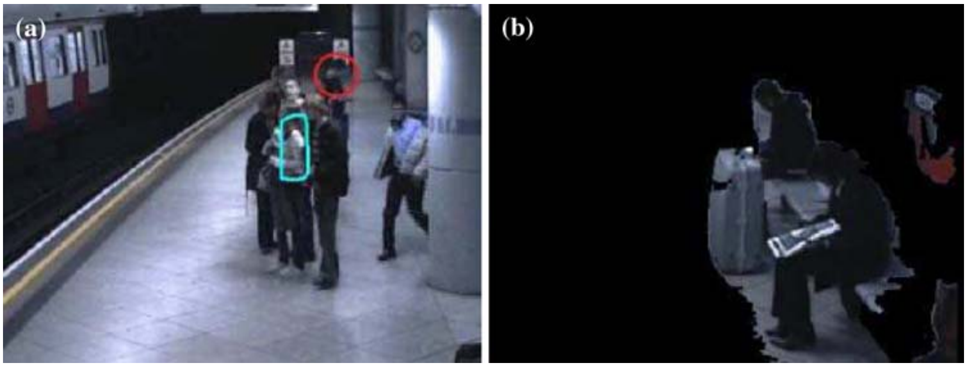

Fig. 8 Special scene contexts. **a** In Test1_i-LIDS, the owner's entrance with the baggage and the following departure without the baggage are not viewable. **b** Throughout Test5_i-LIDS, the foreground blob of the unattended baggage is merged with that of the person sitting behind it

techniques may be well-suited to this task. Other solutions include adopting fused information from multiple cameras to reduce positional ambiguity.

While there are many ideas that we will continue to test and explore, the basic system framework we present here provides a powerful solution towards effective, efficient recognition of unusual events in challenging public environments.

References

1. Allen, J., Ferguson, G.: Actions and events in interval temporal logic. J. Logic Comput. **4**(5), 531–579 (1994)
2. Auvinet, E., Grossmann, E., Rougier, C., Dahmane, M., Meunier, J.: Left-luggage detection using homographies and simple heuristics. In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 51–58 (2006)
3. Bhargava, M., Chen, C.-C., Ryoo, M.S., Aggarwal, J.K.: Detection of abandoned objects in crowded environments. In: Proceedings of 2007 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS), London (2007)
4. Chen, C.-C., Aggarwal, J.K.: An adaptive background model initialization algorithm with objects moving at different depths. In: IEEE International Conference on Image Processing (ICIP), San Diego (2008)
5. Grabner, H., Roth, P., Grabner, M.: Autonomous learning of a robust background model for change detection. In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 39–54 (2006)
6. Gutchess, D., Trajkovic, M., Kohen-Solal, E., Lyons, D., Jain, A.K.: A Background model initialization algorithm for video surveillance. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 733–740 (2001)
7. i-LIDS Bag and Vehicle Detection Challenge in Association with AVSS (2007)
8. i-LIDS Dataset for AVSS (2007)
9. Lewis, J.P.: Fast normalized cross-correlation. In: Industrial Light and Magic, pp. 1–7 (1995)
10. Li, L., Luo, R., Huang, W., Eng, H.: Context-controlled adaptive background subtraction. In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 31–38 (2006)

11. Lv, F., Song, X., Wu, B., Singh, V.K., Nevatia, R.: Left-luggage detection using Bayesian inference. In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 83–90 (2006)
12. Martinez-del-Rincon, J., Herrero-Jaraba, J., Gomez, J., Orrite-Urunuela, C.: Automatic left luggage detection and tracking using multi-camera UKF. In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 59–65 (2006)
13. Nevatia, R., Zhao, T., Hongeng, S.: Hierarchical language-based representation of events in video streams. In: Proceedings of IEEE Workshop on Event Mining (2003)
14. PETS 2006 Benchmark Data (2006)
15. Porikli, F.: Detection of temporal static regions by processing video at different frame rates. In: Proceedings of IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS), London (2007)
16. Ryoo, M.S., Aggarwal, J.K.: Recognition of composite human activities through context-free grammar based representation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, pp. 1709–1718 (2006)
17. Smith, J., Chang, S.-F.: VisualSEEk: a fully automated content-based image query system. In: Proceedings of ACM International Conference on Multimedia, Boston (1996)
18. Smith, K., Quelhas, P., Gatica-Perez, D.: Detecting abandoned luggage items in a public space. In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 75–82 (2006)
19. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **22**(8), 747–757 (2000)