Detection of Abandoned Objects in Crowded Environments

Medha Bhargava, Chia-Chih Chen, M. S. Ryoo, and J. K. Aggarwal
Computer and Vision Research Center
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712, USA
{medha_b,ccchen,mryoo,aggarwaljk}@mail.utexas.edu

Abstract

With concerns about terrorism and global security on the rise, it has become vital to have in place efficient threat detection systems that can detect and recognize potentially dangerous situations, and alert the authorities to take appropriate action. Of particular significance is the case of unattended objects in mass transit areas. This paper describes a general framework that recognizes the event of someone leaving a piece of baggage unattended in forbidden areas. Our approach involves the recognition of four sub-events that characterize the activity of interest. When an unaccompanied bag is detected, the system analyzes its history to determine its most likely owner(s), where the owner is defined as the person who brought the bag into the scene before leaving it unattended. Through subsequent frames, the system keeps a lookout for the owner, whose presence in or disappearance from the scene defines the status of the bag, and decides the appropriate course of action. The system was successfully tested on the i-LIDS dataset.

1. Introduction

Visual surveillance systems today consist of a large number of cameras, usually monitored by a relatively small team of human operators. Typically, each operator watches a set of screens that cycle through views of different locations every few seconds. Recent studies have shown that the average human can focus on tracking the movements of up to four dynamic targets simultaneously, and can efficiently detect changes to the attended targets but not the neighboring distractors [1]. It appears that there are spatial and temporal limits to the tracking capability of humans [2]. When targets and distractors are too close, it becomes difficult to individuate the targets and maintain tracking. This difficulty in selecting a single item from a dense array, despite clear visibility, has been attributed to the acuity of attention, or, alternatively, to obligatory feature averaging. Speed of the targets is another factor that limits the tracking accuracy of the average person [2]. Further, according to the classical spotlight theory of visual attention, people can attend to only one region of space (i.e. area in view) at a time, or at most, two [3]. Simply stated, the human visual processing capability and attentiveness required for the effective monitoring of crowded scenes or multiple screens within a surveillance system is limited. Thus, more often than not, camera footage at such locations finds greater use in post-event investigation than in crime prevention and security enforcement.

Intelligent video analysis offers a promising solution to the problem of active surveillance. Automatic threat detection systems can assist security personnel by providing better situational awareness, enabling them to respond to critical situations more efficiently. In this paper, we present a new methodology for detecting objects left unattended in public areas such as mass transit centers, sporting events and entertainment venues. The algorithm is general, and may be readily adapted for several related applications such as the detection of fallen rocks and other obstructions on roads, railway tracks and runways, and the monitoring of cargo. Here, we focus on the detection of abandoned baggage at train stations, where an object is defined as *abandoned* in a spatio-temporal context: when its owner has left a predefined detection area for longer than a certain time period t (60 seconds in our case).

Our system essentially emulates the behavior of a human operator. At the first sighting of unattended baggage, the system traces it back in time to look for its owner. The *owner* of the bag is conservatively defined as the person who brings it into the scene. Once a candidate owner has been associated with the bag, a search for the owner is initiated. If the owner is found to be missing from the detection zone for longer than t seconds, the bag is deemed as abandoned and an alarm is raised. If eventually the person returns to the bag, the alarm is stopped. The flow of events and the order of processing are illustrated in Figure 1.

The rest of the paper is arranged as follows: Section 2 discusses a few approaches explored by other researchers to solve the problem; Section 3 explains the technical details of our method, followed by its performance on the i-LIDS dataset [4] in Section 4. Section 5 wraps up the paper with a brief summary and discussion.

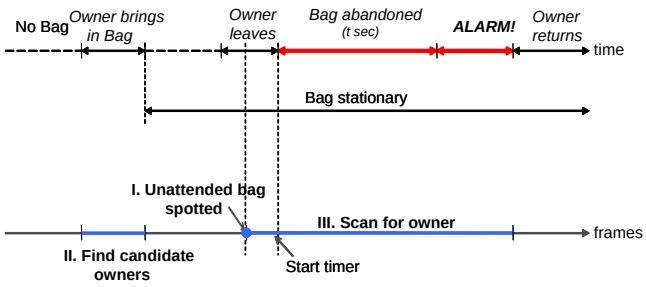

Figure 1: *The abandonment of baggage is described by four sub-events (top two axes), while the progression of the algorithm may be divided into three processing modules (lowest axis).*

2. Previous approaches

In recent years, much effort has been devoted to designing systems that automatically detect abandoned objects in public areas. Haritaoglu *et al* [5] described a method that exploits periodic motion and static symmetry of a person to determine if the person is carrying an object. Systems such as [6] employ adaptive background subtraction (ABS) techniques to detect unknown, changed, or removed objects. Spengler and Schiele [6] propose an approach for detecting abandoned objects and tracking people using the Condensation algorithm. Martinez-del-Rincon *et al* [7] and Aguilera *et al* [8] have explored the use of multiple cameras for similar surveillance tasks in different settings.

3. Proposed algorithm

Our method is designed to capture and exploit the temporal flow of events related to the abandonment of a bag. Figure 1 shows the formal representation of our task, adopted from Allen and Ferguson's classic temporal interval representation of events [10]. Their framework applies temporal interval logic to define the relationships between actions and 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. Here, we define the activity of the abandonment of a bag in terms of four sub-events that lead to it – entry of the owner with the bag, departure of the owner without the bag, abandonment of baggage and consequent timed alarm, and the possible return of the owner to (the vicinity of) the bag.

The algorithm is composed of three computational modules that operate to detect the four aforementioned sub-events that describe the activity: the detection of unattended baggage, reverse traversal through previous frames to discover the likely owner(s), and the continued observation of the scene. The process is preceded by a basic preprocessing stage as discussed below.

3.1 Low Level Processing

For efficiency and ease of computation, we perform background subtraction on each frame. To enhance the generality of our framework, the system is designed to automatically estimate the background from the image sequence. A background initialization algorithm adapted from [11] was used to build the background model [12]. In [11], at each pixel, stable intervals of time are identified and local optical flow is computed to help determine which interval is most likely to display the background. This method has been shown to yield impressive results when optimal parameters are selected. In our system, this critical process of parameter estimation is automated by analyzing the input sequence [12]. The static background thus extracted is impressively close to the true setting.

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, retaining only the most useful segments. Next, the mean-shift algorithm [13] is applied for color quantization and image segmentation. Subsequent processing deals exclusively with the resultant foreground segments, or blobs.

3.2 Detection of Unattended Baggage

The goal of the first module (as depicted in Figure 1) of the algorithm is the detection of any stationary baggage that seems to have been left by itself. Until such an event occurs, it is unnecessary to track and monitor all ongoing activities in the scene. Doing so not only cuts computational costs but also avoids ambiguities born of inaccuracies in tracking in the presence of much movement and occlusion.

The *k*-nearest neighbor classifier is used to classify foreground blobs in novel frames as belonging to the *bag* or *non-bag* class. Classification is based on the shape and size of binary blobs. The representation of bags is established using typical shape and size characteristics gleaned from a set of positive and negative examples provided to the system. Positive training samples were manually collected through Google Image Search and subsequently binarized. Negative samples include humanoid blobs and irregularly-shaped segments selected from the given data sequences. The classifier is trained off-line, using the following features:

- *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 an ellipse that envelopes the blob
- *Orientation*
- *Size*

The size of each binary blob is coarsely normalized (using weights determined empirically) to account for the effects of perspective projection. Blobs outside a predefined range of size are excluded from consideration as possible bags.

The performance of our baggage detection setup (using $k = 3$) is very good. Owing to the simplicity of the binary classifier and the features used, execution time is minimal. To ensure that the bag remains stationary while left alone as well as to reinforce the decision of the classifier, each suspect blob is tracked over a number of consecutive frames (usually, around 10) to check for the consistency of detection and position, before declaring it as *unattended* and moving on to look for its potential owner(s).

3.3 Reverse Traversal

In crowded scenarios where a bag appears to have been abandoned, a human operator is likely to rewind the video to around the 'drop-off' point when the bag was first brought to and placed at its detected position, and observe the movement and behavior of the owner from that point on to gauge the threat level of the situation. This module of our system acts in much the same way. Once the system latches onto an unattended bag, it traces it through previous frames to detect the event of the owner setting down the bag.

Most of the backtracking stage is implemented in a straightforward manner to facilitate speedy traversal to the frames of interest, i.e. when the bag 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 (or larger) occupying the same spot as the detected baggage is assumed to indicate the presence of the bag. This supposition may result in overshooting of the desired frames, which can occur in the event when the entry of the bag at the position is not clearly visible. This method of matching based only on positional overlap accounts for instances of severe occlusion of the bag, thereby reducing the chances of mistaking the wrong person(s) as the possible owner(s).

When no valid blob is found at the anticipated location, it is inferred that the bag was in motion and ought to be present elsewhere in the neighborhood. Note that while tracking in reverse time, the movement of the bag corresponds to the past event of the owner arriving at the location with the bag. The algorithm then performs template-matching using normalized cross-correlation [14] to search for the bag in the nearby region (using the previously stored appearance model of the abandoned bag).

Image matching is performed using normalized cross-correlation the patch level; i.e. each foreground segment is matched against the recorded template patches. Rectangular image patches are extracted from the desired neighbourhood, as shown in Figure 3, using sampling grids of different sizes, devised to coarsely account for perspective distortion. A comprehensive pool of patches is extracted, from which patches containing none or a very small fraction of the segments are discarded. A record of the centroid of the parent blob from which each patch was derived is also maintained, and serves as a simple way of incorporating some basic positional information into the model of each candidate owner, as will be clear shortly.

Normalized cross-correlation coefficients are computed according to Eq. (1) for every pair of corresponding planes in HSV space *i.e.* $p \in \{h, s, v\}$, where γ^p is the greatest

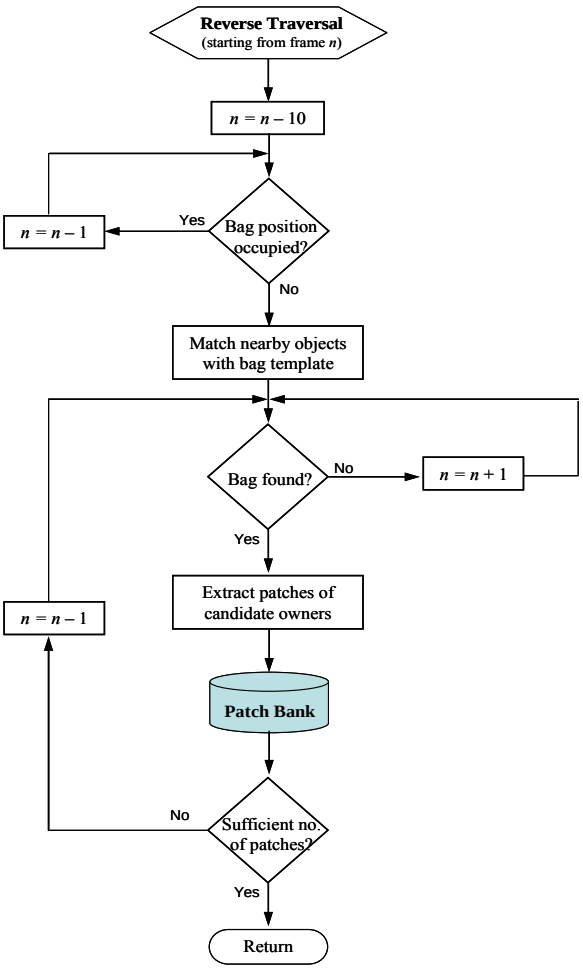

Figure 2: *Flowchart description of module*

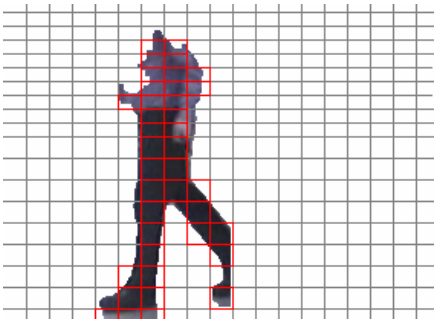

Figure 3: *The patch bank consists of informative image patches (red) in the neighborhood of the back-tracked baggage, which are collected the frames when candidate owner is expected to bring the baggage to the scene.*

cross-correlation coefficient in plane p. A weighted mean of the three coefficients is used to quantify the degree of matching. In Eq. (1), a template patch f^p is positioned at (u, v) of the sub-image of a colored foreground blob I^p. $\bar{f}^p$ and $\bar{I}^p_{u,v}$ represent the mean values of a template patch and a foreground sub-image under the region of the template patch respectively.

$$\gamma^p(u,v) = \frac{\sum_{x,y}\left[I^p(x,y) - \bar{I}^p_{u,v}\right]\left[f^p(x-u,y-v) - \bar{f}^p\right]}{\left\{\sum_{x,y}\left[I^p(x,y) - \bar{I}^p_{u,v}\right]^2 \sum_{x,y}\left[f^p(x-u,y-v) - \bar{f}^p\right]^2\right\}^{1/2}} \quad (1)$$

Two situations can arise from the outcome of correlation in a frame n: either the bag is found nearby or it is not. The methods used for handling the two possibilities are discussed below and outlined illustratively by Figure 2.

Situation 1: If the bag is found, it may be inferred that the bag was being moved or carried at the time, presumably by its owner. The extracted foreground blobs in its neighborhood can thus be considered as representative of the candidate owners. As can be seen in Figure 3, the patch bank forms the patch-based appearance model of the candidate owner(s), with patches extracted using a sampling grid that is normalized for size. Only the most informative patches (those with at least 50% non-zero values) are retained. The system continues backtracking (to frame n=n–1) until the beginning of the video stream is reached or a sufficient number of patches is collected (within the updated neighborhood of the bag).

Situation 2: In the case where the bag cannot be found, it may be inferred that the desired set of frames has been overshot. Such an eventuality may arise when the actual arrival of the bag on the scene occurs in the presence of occlusion, or if someone (or something) else was standing beside the bag when the real owner left. Conversely, in terms of reverse traversal, this is the situation where the movement of the bag under inspection from its detected location goes unnoticed by the simple blob tracker either due to severe obstruction by another sufficiently large object blob or its merging into another blob visually near it. In such a case, the system flips the direction of traversal (so as to be moving towards along the positive time axis) and attempts to locate the bag in frame (n+1).

An explanation of the working of this module is shown in Figure 4.

Pinpointing the true owner of each observed *unattended bag* blob with reliable precision in the presence of several people can be very difficult. The odds of making a mistake in assigning specific ownership in a crowded scenario are rather high, so any attempts to zero in on a single individual automatically are best avoided. It could be possible that the true owner came in alongside another person (or more), and given the view, it may not be possible (even humanly so) to discern the actual owner reliably. In fact, in such a case, it would probably be desirable to attribute possession to all possibly involved persons for later inspection and investigation, should any foul play be suspected.

```
main()
{ ...
 lookForUnattendedBaggage()
 if (suspicious bag detected in frame[n])
 {
   backtrack to frame[n-1] and check
   if (bag found)      ⇒ bag stationary
     Continue backtracking till not found

   if (bag not found)
     traverse(n)
 }...
}
traverse(n)
{
 attempt to match bag template in
   neighborhood
 if (matched)
   ⇒bag in motion, presumably with owner
   findCandidateOwners()
   while (more patches needed)
     continue iterative reverse traversal
     i.e. n = n-1
 if (not matched)
   ⇒bag not in scene
   ⇒point of entry overshot in traversal
   continue iterative traversal in
   forward direction    i.e. n = n + 1
   when (matched)
      findCandidateOwners()
      while (more patches needed)
       continue iterative forward traversal
        i.e. n = n + 1
}
```

Figure 4: *Pseudo-code for detecting candidate owners.*

The patch bank is expected to contain several redundant patches. This redundancy is intentional and useful to bias subsequent comparisons to match the actual owner with greater probability, since it is only reasonable to expect that the true owner would stay by the bag for at least a short period of time. It could also potentially serve as a means for handling viewpoint variation.

As a final step of creating owner hypotheses, the patch bank is consolidated by taking advantage of the positional relationships between patches. Since a few successive images in the sequence are considered for building appearance models of candidate owners, any large translation of blobs across the frames is highly improbable. Patches collected across different frames but originating from blobs that lie in the same locality are clustered together as potentially belonging to the same person(s). The redundancy of the patch bank together with this spatial clustering results in an inherent probabilistic distribution of likelihood of ownership per blob within the prescribed window of interest around the bag. The patch bank may finally be pruned by eliminating clusters that are too small to be meaningful for comparison.

3.4 Continued Scene Monitoring

The purpose of the third module is to monitor the departure and the return of candidate owners, which directly controls the activation of the alarm. After constructing a representative patch bank, we return to the point when the bag was identified as unattended i.e. the present frame. Looking forward in time from then on, our intention is to keep track of the presence and actions of all possible owners. The system maintains a watchful eye to detect the event of their departure from the neighborhood of the bag, observes the area for the possible eventuality of their return, and sounds the alarm if they are missing for longer than a predefined t seconds (here, $t = 60$).

In order to look for candidate owners, every color blob in the vicinity of the bag is cross-correlated with the complete patch bank. Only a fraction of the most similar patches is retained. The spatial coherence of the top hits for each color blob is then analyzed to see if the blob closely matches any single appearance model in the patch bank. If it does, then ownership of the bag is assigned to the corresponding blob. This step is taken to safeguard against the possibility that parts of the blob may match random patches in the patch bank that were originally extracted from different blobs. Adding the spatial configuration parameter adds to the uniqueness of each patch and the robustness of the system. Thus, the presence or absence of the owner is established based on both appearance and spatial constraints.

If a likely owner who meets both matching criteria is found in the detection area, no action is taken. For as long as the bag remains *unattended*, the system continues scanning for the owner in the area. However, if the owner steps outside the predefined detection zone or is not visible at all, a timer is set. To insure against inaccurate feature matching, the conclusion of possible abandonment is reached over a sequence of successive frames (usually, 3 frames). The system carries on its scrutiny of the scene for the possible event of the owner returning to the bag, in which case the timer is deactivated. Once again, this decision is made over several frames to add to its confidence. In the event that the timer ticks on for t seconds, an alarm is triggered, and persists until the owner returns to the bag or it is manually reset.

4. Experimental results

We tested our algorithm on the Imagery Library for Intelligent Detection Systems (i-LIDS) dataset [4], made available by the UK Government's Home Office Scientific Development Branch. i-LIDS is the Government's benchmark for evaluating video-based detection systems. It contains three video sequences featuring scenarios of temporarily abandoned baggage shot at the same location at Westminster metro station, labeled by their projected level of difficulty. Videos are recorded at a sampling rate of 25 fps, at a resolution of 720 x 576 pixels.

The three videos feature different degrees of scene density, baggage size, and type. The complexity of the problem arises from occlusion, changes in lighting, large perspective distortion, and the similarity in appearance of different people. Another significant problem is the possible similarity between the appearance of people and the background, which could lead to inaccurate segmentation. Our system is able to successfully overcome these difficulties to yield impressive results, as shown in Table 1. Alarm times match within one second of the ground truth. Figure 5 demonstrates the sequence of processing dataset *AB_medium*, which corresponds directly to the progression of sub-events as described in Figure 1.

Table 1: *Performance on the i-LIDS dataset.*

Sequence	Start time		Alarm Duration	
	Ground truth	Our Result	Ground truth	Our Result
easy_AB	00:03:00	00:02:59	00:00:12	00:00:12
medium_AB	00:02:42	00:02:42	00:00:18	00:00:18
hard_AB	00:02:42	00:02:42	00:00:24	00:00:25

(a) Frame 265 (b) Frame 297

(c) Frame 577 (d) Frame 616

(e) Frame 949 (f) Frame 1059

Figure 5: *Results of processing sequence AB_medium. Figure 5(d) shows the identified unattended bag, which initiated reverse traversal up to Figure 5(a) where the bag was not found. Tracking in forward direction to Figure 5(b) re-discovered the bag and candidate owners were recorded. Figure 5(c) shows the point where the timer was set. 60s later, the alarm goes off in Figure 5(e) and is eventually discontinued in Figure 5(f) when the owner re-enters the scene.*

5. Conclusion

This paper introduces a general framework to detect objects abandoned in a busy scene. The algorithm is, to the best of our knowledge, novel and unique. The proposed algorithm is appealing in its simplicity and intuitiveness, and is demonstrated experimentally to be conceptually sound. It is well-equipped to handle the concurrent detection of multiple abandoned objects swiftly, in the presence of occlusion, noise and affine distortion. The algorithm lends itself naturally to the recognition of a vast variety of related activities, ranging from surveillance and corridor observation to traffic management and cargo monitoring. Its modular structure allows the flexibility for integrating more functionality and sophisticating various sub-modules without disturbing the remaining framework.

The performance and success of our methodology is promising, but much remains to be done. The current MATLAB implementation is computationally sub-optimal. The system can certainly be easily parallelized. A binary search may be performed to look for the 'drop-off' point of the baggage. For more reliable segmentation, it would be worth exploring ways of periodically updating the background, or even using different backgrounds in different contexts (for example, a background with the train at the station). In order to handle more general kinds of objects that may be left around, the system would benefit from more advanced object detection and recognition techniques. Also, a device to separate merged or partially covered objects is needed for greater utility.

There is tremendous scope for experimentation and refinement of the current system. It is, nonetheless, a step towards effective, efficient monitoring of objects in challenging public environments.

References

[1]. C. Sears and Z. Pylyshyn, "Multiple Object Tracking and Attentional Processing," *Canadian Journal of Experimental Psychology*, Vol. 54, pp. 1-14, 2000.

[2]. P. Cavanaugh and G. Alvarez, "Tracking Multiple Targets with Multifocal Attention," *Trends in Cognitive Sciences*, Vol. 9(7), pp. 349-354, 2005.

[3]. F. Tong, "Splitting the Spotlight of Visual Attention," *Neuron*, Vol. 42, pp 524-526, 2004.

[4]. i-LIDS dataset for AVSS 2007.

[5]. I. Haritaoglu, R. Cutler, D. Harwood, L. Davis, "Backpack: Detection of People Carrying Objects using Silhouettes," *Proc. IEEE International Conference on Computer Vision*, Vol. 2, pp. 102-107, 1999.

[6]. H. Grabner, P. Roth, M. Grabner, "Autonomous Learning of a Robust Background Model for Change Detection," *Proc. IEEE International Workshop on PETS*, pp. 39-54, 2006.

[7]. M. Spengler, B. Schiele, "Automatic Detection and Tracking of Abandoned Objects," *Joint IEEE International Workshop on Visual Surveillance and PETS*, 2003.

[8]. J. Martinez-del-Rincon, J. Elías Herrero, Jorge Jómez and Carlos Orrite Uruñuela, "Automatic Left Luggage Detection and Tracking using Multi-Camera UKF," *Proc. IEEE International Workshop on PETS*, pp. 59-65, 2006.

[9]. D. Thirde, M. Borg, J.Ferryman, J. Aguilera, M. Kampel and G. Fernandez, "Multi-Camera Tracking for Visual Surveillance Applications," *11th Computer Vision Winter Workshop*, 2006.

[10]. J. Allen and G. Ferguson, "Actions and Events in Interval Temporal Logic," *Journal of Logic and Computation*, 4(5):531-579, 1994.

[11]. D. Gutchess, M. Trajkovic, E. Kohen-Solal, D. Lyons and A. K. Jain, "A Background Model Initialization Algorithm for Video Surveillance," *Proc. ICCV*, pp. 733-740, 2001.

[12]. Chia-Chih Chen and J. K. Aggarwal, "An Adaptive Background Model Initialization Algorithm for Video Surveillance," *submitted for publication*, 2007.

[13]. D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications," *Proc. IEEE International Conference on Computer Vision*, pp. 1197-1203, 1999.

[14]. J. Lewis, "Fast Normalized Cross-Correlation," *Vision Interface*, 1995.