Relative Attributes For Large-scale Abandoned Object Detection

Quanfu Fan
IBM T. J. Watson Research Center
qfan@us.ibm.com

Prasad Gabbur
ID Analytics
pgabbur@gmail.com

Sharath Pankanti
IBM T. J. Watson Research Center
sharat@us.ibm.com

Abstract

Effective reduction of false alarms in large-scale video surveillance is rather challenging, especially for applications where abnormal events of interest rarely occur, such as abandoned object detection. We develop an approach to prioritize alerts by ranking them, and demonstrate its great effectiveness in reducing false positives while keeping good detection accuracy. Our approach benefits from a novel representation of abandoned object alerts by relative attributes, namely staticness, foregroundness and abandonment. The relative strengths of these attributes are quantified using a ranking function[19] learnt on suitably designed low-level spatial and temporal features.These attributes of varying strengths are not only powerful in distinguishing abandoned objects from false alarms such as people and light artifacts, but also computationally efficient for large-scale deployment. With these features, we apply a linear ranking algorithm to sort alerts according to their relevance to the end-user. We test the effectiveness of our approach on both public data sets and large ones collected from the real world.

1. Introduction

We present a robust and efficient approach to prioritize alerts in abandoned object detection (AOD) for large scale video surveillance. AOD is one of the most important video surveillance applications and has been well studied [4] in the literature. However, the issue of false alarms, although being well-known in industry, has been little addressed in research. Consider for instance that a well designed system yields an extremely low false alarm rate of 2 alerts/day/camera only. It will produce a total of $2,000$ alerts/day on $1,000$ cameras. Suppose that each alert can be verified quickly in 2 minutes on average, then as many as 66.67 hours are still needed for human inspection of all the alerts. This is equivalent to the workload of a team of more than 8 full-time security officers! Hence, there is an urgent need for development of highly scalable AOD approaches with low false positive rates (FPRs) for urban surveillance.

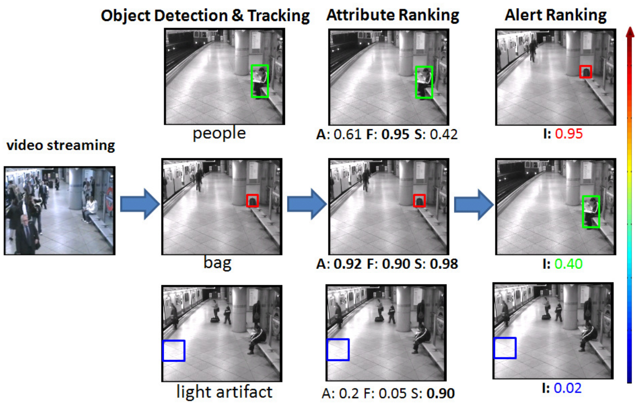

Figure 1: An overview flow chart of our system (best view in color). An abandoned object tends to indicate high staticness (S), foregroundness (F) and abandonment (A). Our approach applies the technique [19] to learn the strengths of these visual attributes (shown below each image). These high-level attributes are then fed to a second level ranker to prioritize the importance (I) of an object. Alerts are sorted (3^{rd} column) by importance and finally triggered if their relevance scores are high. (image courtesy of iLIDS [2])

Making AOD deployable on a large scale raises many new technical challenges. Firstly, scalability does not only mean being able to work in real time. It also requires that a system scale up to a large number of cameras with various view angles and be robust to issues such as quick lighting changes and weather condition changes. Secondly, many other things can be confused with abandoned objects. Among them, people who move around for a while and then become stationary (Fig. 2) exhibit close resemblance to abandoned objects. These types of errors cannot be easily reduced by a pedestrian detector because of varied human poses and occlusions. Finally, harmful or suspicious drops occur rarely . In practice natural drops such as parked bikes and traffic cones (Fig. 2) are considered as true drops, which account for only a tiny portion of the total number of alerts triggered in a system. Such an extremely high imbalance between true and false alarms demands the system to have good hit rates while at the same time working at low FPRs.

In this paper, we address the challenges aforementioned by prioritizing abandoned object alerts using ranking techniques. Ranking is well suited for the problem of AOD , where false alarms dominate detection results. It has the ability to move up alerts of higher importance to the top of the adjudication process while significantly suppressing false alarms.

In order to make alert prioritization feasible, we first propose a novel representation of abandoned objects by visual attributes, namely, *staticness*, *foregroundness* and *abandonment*. In general, abandoned objects are essentially foreground objects that remain motionless over a certain period of time in the scene. Compared to temporally static people who often exhibit slight movement (internal motion), they indicate much higher *staticness*. On the other hand, they are more dissimilar to background than spurious foreground objects such as lighting artifacts and shadows, i.e. with high *foregroundness*. One more attribute that abandoned objects possess uniquely is *abandonment*, which is referred to in [9] as some associated human activity or behavior around an object just before it is dropped and left in isolation. Motivated by the recent work of relative attributes by Parikh and Grauman [19], we further specify the relative strengths of these attributes on different types of alerts raised by various objects in the scene, and apply the technique of [19] to score the attributes. As demonstrated later, these high-level semantic features are intuitively discriminative in separating abandoned objects from other types of alerts. They are also invariant to camera view changes and can be computed fast, making them well-suited for large-scale video analysis.

To learn these attributes, we build an efficient lightweight tracker to track objects in the scene. Since static objects are of primary interest in AOD, we integrate the tracker with the approach of [8] which models temporarily static objects by a finite state machine. Doing so provides rich information about the history of a static object, from where and when it originates to where and when it becomes static, even under occlusion. This information enables effective extraction of spatio-temporal features for staticness and abandonment analysis. As we show later, these low-level features, when combined together, can be transformed into powerful high-level features for alert ranking.

We finally use these learnt attributes as input to a ranker to sort alerts by importance. The degree of importance for an alert is given in the order of *bags* $>$ *people* $>$ *other alerts*. Here *bags* refer to true abandoned objects or true alerts. We enforce such a relationship of ordering between alerts in the ranker largely because people are the most confusing alerts to bags and other alerts such as light artifacts and shadows are of the least interest to the users. We again adopt the technique of [19] for alert ranking due to its simplicity and efficiency. In our experiments, it demonstrates good generalization capability on new data.

An overview of our system is illustrated in Fig. 1. To the best of our knowledge, this work is the first to propose a general representation of abandoned objects by quantifiable visual attributes. While some of these attributes (or concepts) have been tried in previous work [9] for false alarm reduction, they were used qualitatively and mostly in a heuristic way. Our approach is also one of the very few that endeavor to make large-scale video surveillance practical, with significant focus on scalability and robustness. In our experiments, we thoroughly validated the effectiveness and robustness of our approach under various challenging urban scenarios, using both public data sets with staged drops and a data set collected from deployed cameras including natural drops.

2. Related Work

Abandoned object detection has received extensive research attention recently due to its relevance to antiterrorism [4]. However most previous works [6, 13, 16] focus on improving detection accuracy in crowded scenarios and are only evaluated on some small public data sets such as [3] and [1].

Most of these data sets contain a small number of short test sequences captured in relatively simple scenarios. As a consequence, the issue of false positives, though a notoriously known problem in industry, may not stand out as an urgent issue to resolve in research. Some work such as [25, 7] focus on detection of abandoned and removed objects, but these approaches usually do not handle lighting changes very well or are susceptible to low texturedness and cluttered background. The idea of tracking has been applied to abandoned objection detection in [23, 6, 13] for owner identification. While these approaches may work well in simple scenarios , they can possibly miss many objects in crowded scenarios where tracking fails easily. Another line of work is to use object recognition techniques to detect bags or luggage directly ([18, 16]), but training robust detectors across cameras remains challenging. A particular limitation of these approaches is that they can only detect a few specific types of objects. Recently, some works have attempted to address the issue of false positives in a more systematic way to meet the requirement of large-scale deployment of abandoned object detection. For example, in [9], a sequence of robust filters were developed to address different types of false alarms by doing foreground and abandonment analysis.

Attributes have been widely studied for a variety of multimedia and vision tasks [17, 12, 5, 20] such as multimedia retrieval, face verification and object classification. Due to limited space, we refer readers to [19] for a good review of these techniques.

3. Abandoned Object Alerts

In the context of *PETS2006* [3], an abandoned object is defined as an item of luggage that has been left behind by its owner. In this work, we consider abandoned objects as stationary objects that are physically isolated from other foreground objects in the scene for some time. We relax the requirement of ownership into a more realistic setting where the owner of an object may not be trackable or even not be visible. In practice, in addition to bags or luggage, interesting drops picked up by a system include natural items such as bikes, garbage cans and traffic cones (Fig. 2). For convenience, we refer to all of them as *bags* (or *true drops*) in this paper as opposed to false alarms described below.

Among false alarms, people and quick lighting changes are two dominant sources (Fig. 2), followed by shadows and ghosts (spurious foreground objects detected after temporarily static objects move again in a scene). Sometimes adverse weather conditions such as rain and snow could cause a sudden significant increase of false alarms. For these types of alerts, we place them into the same category as that of quick lighting changes due to their similarity.

Abandoned objects are primarily static items. We detect them using the method of [8], a technique based on background subtraction (BGS). This technique features a finite state machine (FSM) that tracks temporarily static objects robustly even under occlusion. The FSM provides the background model with object-level information that allows for region-level background modeling and updating, thus improving the model's capabilities in handling crowds and occlusions and in detecting static objects. A static object is identified after a large portion of its pixels are observed motionless.

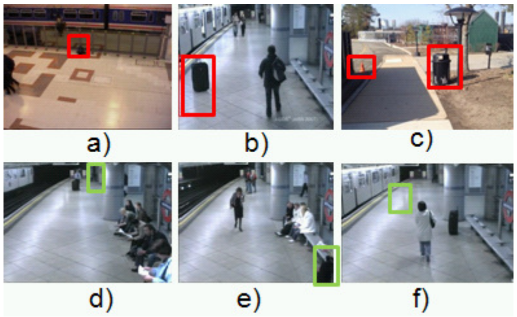

Figure 2: Typical abandoned object alerts in video surveillance. a) a sample staged drop from *PETS2006* b) a sample staged drop from *i-LIDS* c) two natural drops (trash cans and traffic cones) d) a non-occluded sitting person e) an occluded sitting person f) a light artifact

Alerts	ST	FG	AB
B^+	High	High	High
P^-	Low	High	Medium
L^-	High	Low	Low
S^-	High	Medium	Low
G^-	High	Low	Low
Relative Order	$B^+ > P^-$ $L^-, S^-, G^- > P^-$	$B^+, P^- > S^-$ $S^- > L^-, G^-$	$B^+ > P^-$ $P^- > S^-, L^-, G^-$

Table 1: Staticness (ST), foregroundness (FG) and abandonment (AB) attributes and their relative strengths for different types of abandoned object alerts: Bags (B), People (P), Light artifacts (L), Shadows (S) and Ghosts (G). The superscript '+' denotes the class of true drops and '-' denotes false alarms. The bottom row shows the relative orderings of the different objects w.r.t. the attribute of the corresponding column. $X > Y$ implies X exhibits a higher degree of a particular attribute than Y, while X, Y implies that X and Y possess a similar degree of that attribute.

4. Attributes of Alerts

Many computer vision tasks have been traditionally addressed with by a supervised learning approach using a training dataset with low-level features and ground truth labels. The hope is that the classifier can learn the underlying semantic structure in the data. More recently, designing semantic features that are also physically interpretable by humans has been seen to yield promising results [17, 12, 5, 20]. Further, obtaining ground truth labels for pairs of points indicating the relative degree of such features in the points is easy. In other words, the pairwise relationships between points can be fully established by only defining the relative ordering between objects, which are much fewer. We take a similar approach here and design three physically expressible features (attributes) that seem plausible for abandoned object detection.

Specifically, our attributes are called *staticness*, *foregroundness* and *abandonment*, as mentioned previously. *Staticness* is designed to refer to the degree of immobility or stillness of an object across multiple video frames. Similarly, *foregroundness* refers to the distinctiveness of the object relative to the background based on its appearance. Finally, *abandonment* expresses the notion of the object being left in isolation after remaining in possession or vicinity of some other entity. In our work, the level of abandonment for an object is related to the magnitude of external motion around the object right before it is left in isolation. In such a way, we bypass the problem of solving the challenge of owner identification and tracking in crowded scenes and instead focus on analyzing the motion around the abandonment of an object.

It is possible to describe the relative strengths of different kinds of objects associated with alerts in terms of the above attributes (Table 1). We expect that a truly abandoned object (B^+) such as a bag or a piece of luggage remains static in the scene for a long time (high *staticness*), is very different from the background (high *foregroundness*) and has been previously in the possession of its owner (high *abandonment*). On the other hand, an object associated with a false alarm is not expected to exhibit high degrees of all the three attributes. For instance, a person (P^-) is highly distinctive from the background (high foregroundness) but may indicate slight movement occasionally (low staticness). In addition, he can be part of a group initially in the scene and isolated later exhibiting abandonment somewhere between a bag and a static background (medium abandonment). Similarly other situations associated with false alarms such as lighting changes (L^-), shadows (S^-) and ghosts (G^-) exhibit different degrees of the proposed attributes and hence different relative rankings as shown in Table 1. The ordering and similarity labels for the attribute ranking algorithm (Section 5) are derived from these hypotheses.

Fig. 3 shows a small sample of our data points represented in the relative attributes space (3D) as learned by attribute rankers (Section 5). Clearly, bags, which are our objects of interest, are separated well from other objects corresponding to false alarms.

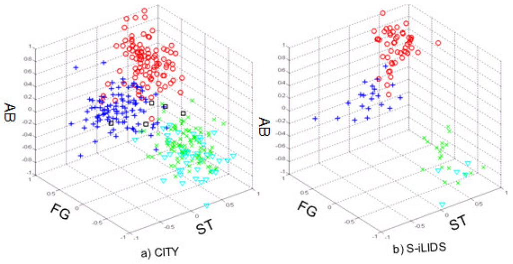

Figure 3: Attribute scores of staticness (ST), foregroundness (FG) and abandonment (AB) learned from two data sets *CITY* and *S-iLIDS* for bags(○), people(+), lighting artifacts(×), shadows (□), and ghosts (▽).

5. Ranking Using Relative Attributes

We adopt the relative attribute framework [19] to rank order our data points in terms of their degree of *staticness*, *foregroundness* and *abandonment*. It is an extension of the SVM ranking formulation [11] by incorporating known similarities between pairs of points in addition to relative orderings. Given a set of data points $\boldsymbol{x}_i$ in $\mathbb{R}^n$ with known pairwise rank orderings and similarities between some pairs of points in terms of a set of attributes $a \in A$, the ranking function f_a is a linear combination of the measurements parametrized by a weight vector $\boldsymbol{w}_a$

$$f_a(\boldsymbol{x}_i) = \boldsymbol{w}_a^T \boldsymbol{x}_i, a \in A \tag{1}$$

The weight vector $\boldsymbol{w}_a$ for a particular attribute a is learnt by optimizing an SVM-like objective function. Let $(i > j) \in O_a$ represent known pairwise rankings between points w.r.t. attribute a and similarly $(k \approx l) \in S_a$ represent known pairwise similarities on the same attribute. The optimum weight vector $\boldsymbol{w}_a^*$ is obtained by minimizing the following objective function

$$\boldsymbol{w}_a^* = \underset{\boldsymbol{w}_a}{\operatorname{argmin}} \left(\frac{1}{2} \boldsymbol{w}_a^T \boldsymbol{w}_a + C \left(\sum_{(i,j) \in O_a} \xi_{ij}^2 + \sum_{(k,l) \in S_a} \gamma_{kl}^2 \right) \right) \tag{2}$$

$$\xi_{ij} = \max\left(0, 1 - (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{w}_a\right)$$
$$\gamma_{kl} = |(\boldsymbol{x}_k - \boldsymbol{x}_l)^T \boldsymbol{w}_a|$$

where $\xi_{ij} \geq 0$ and $\gamma_{kl} \geq 0$ denote $slack$ variables that penalize wrong rank orderings and dissimilarities respectively between the labeled pairs in the training data set. The coefficient C emphasizes the ordering and similarity errors relative to the margin. Note that the margin here refers to the difference between the two nearest ranking scores among all pairs of known rankings in the training data. The above objective function is convex and is optimized using Newton's method with a line search for the optimum step size.

6. Object Tracking And Low-level Features

One of the main challenges is to deal with alerts raised by people, which often exhibit high similarity to abandoned objects. Two useful clues for separating people from bags are how an object arrives at the current location and how it remains static in the same location. This requires understanding the history of an object in the scene, and thereby object tracking, which is challenging in crowded scenes. Fortunately, tracking does not need to be perfect in our case. Even if one can only track an object for a short period of time prior to its being static, such information turns out to be helpful for staticness and abandonment analysis when combined with other BGS-related information, as described later. Different from other tracking-based approaches [23], tracking in our approach is not intended to identify the owner of an abandoned object. Instead it aims to provide sufficient evidence for differentiating people from truly static objects for the purpose of suppressing false alarms.

6.1. Mini-tracker

Our tracker is a simplified version of the one used in [22] that does blob association with bipartite matching. While appearance has proved useful, we only use size and location information for computational efficiency. The mini-tracker also keeps track of only the start and end positions of a track for space efficiency. It is understandable that such a tracker is by no means expected to perform well in a crowd. However, as demonstrated later, the tracking information enables extraction of low-level features that can be turned into powerful high-level features by attribute learning.

Occlusions occur frequently in typical urban environments with human activity. Losing an object due to occlusion will not allow us to fully leverage the information from the tracker. We thus enhance the mini-tracker's capability of maintaining a static object for a longer term, especially under occlusion. This is achieved by enabling interactions between the tracker and the FSM in the background model (see section 3 for details about FSM), which understands when a static object is occluded. Specifically, we place on hold an object marked as "occluded" by FSM, and check how well it matches the original blob right after it re-appears in the scene. The track of the object is reactivated in the case of a good match being found. It should be mentioned that identity switchings still occur on moving people, but fortunately they are usually not harmful to the system performance. Fig. 4 shows a few tracks in a crowded scene. Note that each of the two bag tracks in Fig. 4a) and 4b) starts and ends almost at the same location, indicating possibly high staticness. On the other hand, the long track of the person in Fig. 4c) implies a strong movement of the object, i.e low staticness.

6.2. Low-level Features

The tracker provides the start and end locations of an object as well as its size at each location (bounding box), denoted here by (L_s, R_s) and (L_e, R_e) respectively. For the purpose of abandonment analysis, we further search for the blob R_a that maximally overlaps with R_e right before the object gets tracked by the mini-tracker. For a truly static object, this region is supposed to be a larger external motion either associated with the owner (Fig. 4a)) or a crowd (Fig. 4b)) in the case of high-level activity.

Based on the tracking results, we extract the following features that are more relevant to abandonment:

1. the time taken for an object to get static;
2. the distance between R_e and R_s, i.e. $||L_e - L_s||$;
3. the total length of the track;
4. the aspect ratio of the static region;
5. the ratio of the area of the static region over that of the start region, i.e. $\max A(R_s)/A(R_e), A(R_e)/A(R_s))$ where $A(.)$ represents the area of a region;
6. the height of the static region over that of the start region, i.e. $\max(h_{R_s}/h_{R_e}, h_{R_e}/h_{R_s})$;
7. repeat 4 and 5 for region R_a and R_e if R_a exists.

The following features are more relevant to staticness and mostly extracted from BGS,

1. the maximum and average movements of the object since its being static;
2. the total residence time of the object in FSM
3. the percentages of frames above a good matching threshold

Note that the FSM starts to track an object if it remains static for more than 1 second in the scene. The level of matching between the object R_i at frame i and its original region R_0 are measured using area matching for efficiency, i.e. $r = \frac{R_i \cap R_0}{R_i \cup R_0}$. A matching is considered "good" if $r \geq 0.85$ in our system.

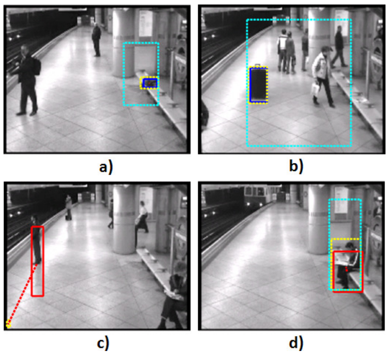

Figure 4: Examples of tracks for people and bags. solid box (red or blue): abandoned object; yellow box: the start position of an object; cyan box: the foreground object from which the object is split. It is expected that the track of a true drop is extremely short (a & b) while the track of a person is sufficiently long to indicate his movement (c).

For foreground analysis, we directly adopt the feature set developed in [9], which has demonstrated superior performance in separating foreground objects and background artifacts related to lighting changes. These features include several edge/texture-based measures that are invariant or robust to illumination changes.

The features described above are used as input for the attribute ranker discussed in Section 5 to compute the ranking scores of staticness, foregroundness and abandonment. As seen from Fig. 3, bags are separated reasonably well from other types of objects even though these features are learned from data with large variations in terms of camera views and human activity.

7. Alert Ranking

We design a second level ranker to sort alerts using the attribute scores learnt previously in Section 5. In practice, some types of false alarms are more important than others to the end user. It is found that investigating irrelevant alerts caused by shadows and lighting artifacts leads to wasteful utilization of a security officer's time and effort. While alerts raised by activities of people in the scene are also less interesting, investigating such alerts sometimes can be useful in detecting potentially harmful situations. Moreover, people alerts present more ambiguity to true drops than other alerts (see Fig. 3). This suggests a relative ordering of alerts themselves based on both their relevance to the end user and their separability, i.e. *bags* > *people* > *other alerts*. We enforce such a relationship of ordering between alerts in a ranker. Due to its simplicity and efficiency, we adopt the technique of [19] again for alert ranking by treating relevance as one single attribute.

Data	#camera	duration	#drops	Bag	People	Light	Shadow	Ghost	Total
		(hrs)		(B^+)	(P^-)	(L^-)	$(S^-$	(G^-)	
PETS2006	1	0.15	6	5	0	1	0	0	6
AVSS-AB	1	0.01	3	3	1	3	0	0	7
i-LIDS	2	3.8	60	48	21	19	0	5	93
CITY	30	70.5	255	196	203	187	9	83	678
NATS	2	96	19	19	139	238	9	107	512

Table 2: evaluation data and alert distributions.

A binary SVM classifier with probabilistic output can also be used for ranking. In our experiments, we compare such a SVM-based approach to ours and demonstrate the superiority of the latter.

8. Experimental Results

8.1. Evaluation Data and Annotations

We tested our approach on two public data sets commonly used in AOD evaluation: *PETS2006* [3] and *AVSS-AB* [1]. *PETS2006* consists of 7 different scenarios captured by 4 cameras from different viewpoints. We chose View 3, which has been extensively tested in previous work. *AVSS-AB* includes 3 drops selected from *i-LIDS* [2], one of the most challenging AOD data sets that was captured in two subway scenarios at different levels of activity.

Apparently both *PETS2006* and *AVSS-AB* are too small for model training, so we included another 3 larger data sets in our evaluation. The first one is *S-iLIDS*, a subset of *i-LIDS* where we picked two video clips from each scenario in the public training data [1]. The data set has a total of 60 staged drops, and was selected in a way to ensure that the baseline approach used for comparison can detect a reasonably good portion of the drops in the video. The second is a challenging data set(*CITY*) used in [9], containing 255 staged drops within over 70 hours of video footage captured from 30 cameras in typical urban scenarios such as streets and parks. It covers almost all kinds of confounding issues known in video surveillance. For instance, high activity, occlusions, low contrast, confusing people and weather/lighting changes. To further validate the effectiveness of our approach in challenging realistic environments, we collected data for 2 days in a row from 2 cameras monitoring busy streets with a lot of loitering people. This data set, which we call *NATS*, only contains natural drops such as bikes and trash cans.

We detected static objects using the approach in [8] on all data sets (See Section 3), and manually classified all the objects into 5 categories, i.e. bags, people, light artifacts, shadows and ghosts (Table 2). This forms the ground truth of our evaluation. Note that the number of bags may be smaller than the number of drops in Table 2 due to detection failures in [8].

8.2. Evaluation

Not many currently existing approaches are suitable for our evaluation as most of them only focus on improving detection accuracies. We use a recently developed approach [9] (*FSM-AOD*) as our baseline for comparison, which has demonstrated good capability in handling light artifacts on *CITY* by robust foreground analysis.

We developed three approaches based on the low-level and high-level features and compared them against the baseline. The first one is an alert ranker using high-level attributes (*HL-RANK*) as described in Section 7, and the other two are basically binary SVMs using low-level features (*LL-SVM*) and high-level attributes (*HL-SVM*) respectively. The two SVMs treat bags as positive labels and other alerts as negative and are trained with a linear kernel. While parameter tuning generally leads to better performance, we set all the model parameters to default in our evaluation.

Results on *CITY* and *S-iLIDS*. We first evaluated our approaches on *CITY* and *S-iLIDS* in two ways, *general test* and *cross-data validation*. The former is a traditional test, i.e. splitting a data set into half for training and half for testing. We conducted 10 runs for each model, and reported the averages of the runs here. The latter further validates the generalization ability of an approach to new data, i.e. training on one data set and testing on another. We show the results as ROCs in Fig. 5 and 6, respectively. The independent axis is the False Positive Rate (FPR=#FPs/#Total Alerts) as a function of the classification or ranking score. The dependent axis is the True Positive Rate (TPR=#TPs/#Total TPs). We do not adopt the widely used performance metrics such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) here [14] because all our approaches rank true positives well on these two data sets, leaving these metrics less effective in measuring ranking quality.

General Test. First of all, we observe that all the proposed approaches outperform the baseline on the two datasets (Fig. 5), suggesting both low-level and high-level features enable better modeling capability. At the same recall, all our approaches show a significant reduction of false alarms in comparison with the baseline.

HL-RANK achieves the best performance on *S-iLIDS*, but is not as good as *LL-SVM* and *HL-SVM* on *CITY*. We speculate that *LL-SVM* is overfitting on *CITY* as it is known that high-dimensional features are more prune to overfitting. We will confirm this later in the cross-data validation where *LL-SVM* fails to yield consistently comparable performance over other data sets.

Cross-data Validation. In this test, *HL-RANK* outperforms *HL-SVM* and *LL-SVM* on both data sets (Fig. 6). Especially, when trained on the larger data set (*CITY*), *HL-RANK* provides a big improvement over other approaches on the smaller data set (*S-iLIDS*). In addition, while *LL-*

[1] Scene 1: *ABTEA104a,ABTEA105a*; Scene 2: *ABTEA201a, ABTEA201b*

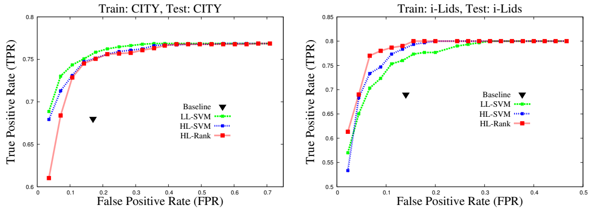

Figure 5: TPR v.s. FPR on different approaches (general test)

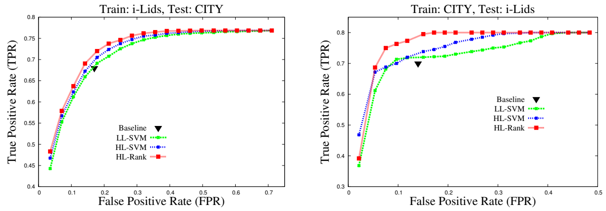

Figure 6: TPR v.s. FPR on different approaches (cross-data validation)

Methods	*PETS2006*		*AVSS-AB*	
	P	R	P	R
[24]	0.05	1.0	0.01	1.0
[10]	0.6	1.0	0.1	1.0
[15]	0.5	1.0	0.03	1.0
[13]	0.75	1.0	0.33	1.0
[21]	0.37	1.0	0.05	1.0
FSM-AOD [9]	0.83	0.83	0.5	1.0
LL-SVM	1.0	0.26	1.0	0.40
HL-SVM	1.0	0.42	1.0	0.90
HL-RANK	0.95	0.80	0.97	1.0

Table 3: Precision (P) and Recall (R) of different approaches on *PETS2006* and *AVSS-AB*.

Data	MAP			NDCG		
	LL-SVM	*HL-SVM*	*HL-RANK*	*LL-SVM*	*HL-SVM*	*HL-RANK*
Cam #1	0.20	0.16	**0.22**	0.46	0.41	**0.53**
Cam #2	0.15	0.15	**0.18**	0.42	0.47	**0.51**

Table 4: Ranking quality comparisons on *NATS* using MAP and NDCG.

Data	*FSM-AOD* [9]		*HL-RANK*	
	#TPs	#FPs	#TPs	#FPs
Cam #1	8	110	8	49
Cam #2	8	100	6	56

Table 5: #TPs and #FPs of different approaches on *NATS*.

SVM built upon *CITY* gives the best results in the general test above, apparently it has difficulty generalizing well to new data.

Fig. 7 shows some examples of true and false positive alerts detected by our system(*HL-RANK*). Our approach is able to eliminate difficult people alerts such as those two illustrated in the top of the figure. Also shown in the bottom of the figure are two very challenging false positives, which are mis-detected possibly due to failures of the tracker.

Results on *PETS2006* and *AVSS-AB*. We cross-validated the effectiveness of our approaches on *PETS2006* and *AVSS-AB* using the models directly trained on *CITY*. In order to compare our results with those reported in other work([4]), we normalize the rank scores from *HL-RANK* into $[0, 1]$ and set the detection threshold to 0.5.

As illustrated in Table 3, while all previous works have a perfect recall of 1.0, most of them, due to lack of attention to false alarms, suffer greatly from this issue even on video data of just a few minutes in relatively simple scenarios. As a comparison, all our proposed approaches yield very few false alarms (high precision). *HL-RANK* continues to demonstrate good generalization ability on *PETS2006*, which has a different camera view from others. However *HL-RANK* fails to detect one bag in *PETS2006*. A closer examination of the results reveals that the missed bag is long and skiny, resembling a person. *LL-SVM* produces the worst recall performance again, confirming our speculation of model overfitting.

Results on *NATS*. Natural drops vary from staged ones in many aspects. It raises a question whether or not our models trained on staged data are still effective in realistic environments. We thus further evaluated our approaches using challenging natural data. Note that *NATS* only includes few natural drops and we do not expect that our approaches can rank them as high as those staged drops, so it makes sense in this case to evaluate the ranking quality of our approaches with MAP or NDCG [14]. For NDCG, we treat all true drops as *relevant* (i.e. a relevance of 1) and all false positives as *irrelevant* (i.e. a relevance of 0). As shown in Table 5, *HL-RANK* consistently provides the best performance on this data set, suggesting that a proper ranking technique is promising for large-scale surveillance.

To better understand how much our system can benefit from a ranking technique, we turned *HL-RANK* into a classifier by thresholding the ranking scores by 0.5, as the same as what has been done on *PETS2006* and *AVSS-AB*. In Table 5, *HL-RANK* demonstrates clear advantages over the baseline by reducing half of the false alarms while still achieving a comparable detection rate with the baseline.We also notice that even with our best approach, the false alarm rate on this data set is still quite high ($20-30$/day/camera). However this data set probably represents the most complex scenario we could imagine in urban surveillance, including rainy weather, high loitering activity and various day-night light artifacts.

Computational Scalability. We benchmarked our system based on *HL-RANK* on a 4-core VM with 2.93 GHz CPU and 4G RAM. The system can process 12 i-LIDS video files at 25 FPS by downsampling the video to 180×144 (half of the original size). This is 3 times faster than real time in a busy subway scenario.

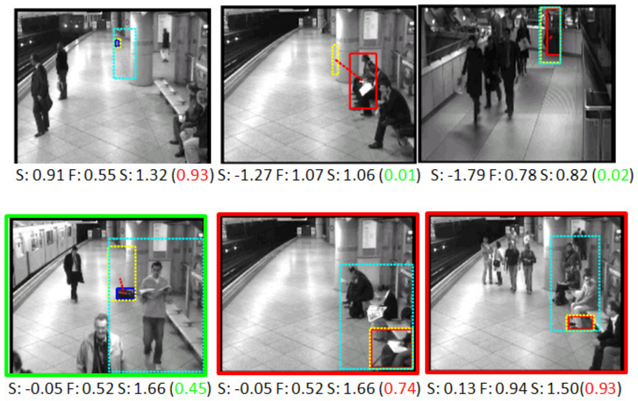
S: 0.91 F: 0.55 S: 1.32 (0.93)
S: -1.27 F: 1.07 S: 1.06 (0.01)
S: -1.79 F: 0.78 S: 0.82 (0.02)
S: -0.05 F: 0.52 S: 1.66 (0.45)
S: -0.05 F: 0.52 S: 1.66 (0.74)
S: 0.13 F: 0.94 S: 1.50(0.93)

Figure 7: Example alerts provided by our system. The top row shows correct detections while the bottom row illustrates false detections (a false positive is highlighted with a red bounding box around the image while a green box indicates a false negative). Shown in the brackets are the ranking scores of the alerts. Also see Fig. 4 for explanation of the bounding boxes.

9. Conclusions

We propose a novel approach to abandoned object detection using the framework of relative attributes. Specifically, we design three physically interpretable attributes (staticness, foregroundness and abandonment) to characterize different kinds of alerts raised by various objects in the scene. We learn ranking functions for each of the attributes to rank order the alerts based on their strengths on the corresponding attributes. The attributes are used as input to an alert prioritization method which performs a ranking using alert importance. Our results suggest that ranking is a promising technique for large-scale video surveillance.

References

[1] AVSS-AB. In *www.eecs.qmul.ac.uk/andrea/avss2007.html*. 2, 6

[2] i-LIDS. In *ftp://motinas.elec.qmul.ac.uk/pub/iLids*. 1, 6

[3] PETS2006. In *www.cvg.rdg.ac.uk/PETS2006/data.html*. 2, 3, 6

[4] A. Bayona, J. Miguel, and J. Martinez. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. In *AVSS*, pages 25–30, 2009. 1, 2, 7

[5] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010. 2, 3

[6] M. Bhargava, C. Chen, M. Ryoo, and J. Aggarwal. Detection of abandoned objects in crowded environments. In *AVSS*, 2007. 2

[7] L. Campos, J. SanMiguel, and J. Martinez. Discrimination of abandoned and stolen object based on active contours. In *AVSS*, 2011. 2

[8] Q. Fan and S. Pankanti. Modeling of temporarily static objects for robust abandoned object detection in urban surveillance. In *AVSS*, 2011. 2, 3, 6

[9] Q. Fan and S. Pankanti. Robust foreground and abandonment analysis for large-scale abandoned object detection. In *AVSS*, 2012. 2, 5, 6, 7

[10] S. Guler and K. Farrow. Abandoned object detection in crowded places. In *PETS*, pages 18–23, 2006. 7

[11] T. Joachims. Optimizing search engines using clickthrough data. In *ACM KDD*, KDD '02, 2002. 4

[12] N. Kumar and A. Berg et al. Attribute and smile classifiers for face verification. In *ICCV*, 2009. 2, 3

[13] H. Liao, J. Chang and L. Chen. A localized approach to abandoned luggage detection with foreground-mask sampling. In *AVSS*, pages 132–139, 2008. 2, 7

[14] T. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*. 6, 7

[15] R. Mathew, Z. Yu, and J. Zhang. Detecting new stable objects surveillance video. In *Multimedia Signal Processing*, pages 1–4, 2005. 7

[16] R. Miezianko and D. Pokrajac. Detecting and recognizing abandoned objects in crowded environments. In *ICVS*, 2008. 2

[17] M. Naphade, J. Smith and J. Tesic et. al. Large-scale concept ontology for multimedia. In *IEEE Multimedia*, 2006. 2, 3

[18] A. Otoom, H. Gunes and M. Piccardi. Feature extraction techniques for abandoned object classification in video surveillance. In *ICIP*, 2008. 2

[19] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 1, 2, 4, 5

[20] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012. 2, 3

[21] F. Porikli, Y. Ivanov and T. Haga. Robust abandoned object detection using dual foregrounds. *Euras. J. Adv. Sig. Proc*, 2008. 7

[22] A. Senior. Tracking people with probabilistic appearance models. In *PETS*, 2002. 4

[23] M. Spengler and B. Schiele. automatic detection and tracking of abandoned objects. In *CVPR*, 2005. 2, 4

[24] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*,1999. 7

[25] Y. Tian, R.Feris and A. Hampapur. Real-time detection of abandoned and removed objects in complex environments. In *IWVS*, 2008. 2