Automated Real-Time Detection of Potentially Suspicious Behavior in Public Transport Areas

Mohannad Elhamod, *Member, IEEE*, and Martin D. Levine, *Life Fellow, IEEE*

***Abstract*—Detection of suspicious activities in public transport areas using video surveillance has attracted an increasing level of attention. In general, automated offline video processing systems have been used for post-event analysis, such as forensics and riot investigations. However, very little has been achieved regarding real-time event recognition. In this paper, we introduce a framework that processes raw video data received from a fixed color camera installed at a particular location, which makes real-time inferences about the observed activities. First, the proposed framework obtains 3-D object-level information by detecting and tracking people and luggage in the scene using a real-time blob matching technique. Based on the temporal properties of these blobs, behaviors and events are semantically recognized by employing object and interobject motion features. A number of types of behavior that are relevant to security in public transport areas have been selected to demonstrate the capabilities of this approach. Examples of these are abandoned and stolen objects, fighting, fainting, and loitering. Using standard public data sets, the experimental results presented here demonstrate the outstanding performance and low computational complexity of this approach. We also discuss the advantages over other approaches in the literature.**

***Index Terms*—Abandoned luggage, behavior recognition, blob matching, fainting, fighting, interobject motion, loitering, meeting, object tracking, occlusion, real time, semantics based, surveillance, theft of luggage, transport, walking together.**

I. Introduction

INCREASINGLY, police and security staff rely on video surveillance systems to facilitate their work. This practice is most evident in large public transportation areas such as metro stations and airports. However, these systems remain largely labor intensive, and the personnel monitoring the video displays find it extremely difficult to be attentive to randomly occurring incidents [1], [2]. Although automated video surveillance systems do exist, they have been used mainly for offline video analysis after an event has occurred, most notably in the case of riot investigations and forensics. At present, these surveillance systems are of marginal help for real-time alerts. Moreover, contrary to the false image created by the media and film industry, research in this young but promising field has made little advancement so far. For example, in the latest riots in the U.K., automated face recognition was not found to be useful due to low camera feed resolution and imperfect lighting conditions [3].

The function of an automated surveillance system is to draw the attention of monitoring personnel to the occurrence of a user-defined suspicious behavior *when it happens*. Two challenges stand in the face of developing fully automated behavior recognition. First, objects of interest, such as people and luggage in a scene, must be found robustly, classified, and tracked through time. Second, a *stable* means of describing events must be found. This is particularly an issue for complex types of events having many different possible variations, such as fighting. Undeniably, in many cases, they are extremely difficult to describe.

What are the contributions of this paper? The majority of researchers to date have invoked machine learning to detect suspicious behavior. To our knowledge, we uniquely propose here a *complete semantics-based* solution to the behavior detection problem that addresses the whole process from pixel to behavior level. Furthermore, the processing is achieved in real time. Although much of the lower level processing stages in this paper are not original, part of our contribution in this regard was to carefully select and integrate them. This proved to be critical for ultimately making correct high-level inferences, which is an issue seldom addressed in the field.

The primary disadvantage of machine learning is that the learned classifiers depend on having reliable standard data sets for training and testing. These are extremely difficult to obtain, particularly for anomalous types of behaviors. This issue is of utmost importance when determining classifier parameters and thresholds. In contrast, the semantic approach replaces this need for training with a more straightforward process based on human reasoning and logic. We claim that this is a more feasible and viable method. For example, it eliminates the specification of complex learning parameters such as decision-tree-pruning thresholds, which are not intuitive to tune and require the intervention of experts in the field. In the semantic approach, more intuitive and meaningful parameters replace these. This paper assumes that foreground *blobs* are extracted in each frame using a conventional background subtraction method. These blobs represent the silhouettes of animate (e.g., people) and inanimate (e.g., luggage) *objects* in the scene, which are the semantic entities associated with the events described. However, in practice, we note that a single blob will often represent multiple objects occluding or standing next to each other. After all blobs have been extracted, inferences are made to segment, track, and classify the objects that they represent. Finally, the anomalous events must be labeled.

Manuscript received April 1, 2012; revised August 10, 2012; October 15, 2012; and October 23, 2012; accepted November 1, 2012. Date of publication December 20, 2012; date of current version May 29, 2013. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The Associate Editor for this paper was M. Á. Sotelo Vázquez.

The authors are with McGill University, Montreal QC H3A 2A7, Canada (e-mail: mndhamod@cim.mcgill.ca; levine@cim.mcgill.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2228640

One contribution of this paper is to elaborate a mathematical framework based on the abstract descriptions depicted by Fuentes and Velastin [4] for detecting potentially suspicious and anomalous activities. These are related to public transport surveillance areas and deal with such issues as abandoned and stolen luggage, loitering, fighting, and fainting. The proposed approach employs high-level motion features, such as velocity of and distance between the semantic entities (objects) in the scene. These objects are defined in 3-D, rather than the usual 2-D. We note that crowd behavior is outside the scope of this paper.

This paper is an expansion and elaboration of a previous conference paper by the same authors [5]. Unlike the former, the conference paper only describes behavior detection, without delving into the details of object detection and tracking.

II. Previous Work

Behavior recognition is a broad term that covers a number of categories of activities, which require different means of detection. For example, crowd behavior, such as crowd movement [6], requires techniques that capture the overall characteristics of the crowd rather than the individuals in it. On the other hand, short-term human actions, such as gymnastic exercises [7] and gestures [8], are often relatively simpler and even periodic. These are of a different nature and therefore require different detection techniques, involving body models [9] and space–time shapes [10].

This paper focuses on automatically flagging suspicious behavior in public transportation systems. Examples are loitering [11], abandoned objects [12], [13], and fighting [14], [15].[1] These types of behavior may occur over a significant period of time. They often involve more than one object; therefore, such matters as finding trajectories, identity tracking, and object classification must be addressed.

One important concern in the field is that most published research focuses on detecting a single type of behavior rather than providing a generic framework. For example, "abandoned luggage detection" is usually handled using low-level background subtraction methods [16], [17]. This method is useful for detecting stationary foreground objects but hardly so for other types of behavior such as loitering or fighting. In addition, most papers that describe a generic behavior detection framework only provide high-level descriptions of a layered framework, without providing a detailed methodology for its implementation.

Grammar-based detection is one of the dominant concepts in the area of behavior detection. It generally attempts to interpret actions in terms of temporal state transitions and conditions [9]. Grammar requires the training of classifiers such as hidden Markov models (HMMs) [14] and temporal random forests [8]. However, using such machine learning techniques suffers from drawbacks related to the classifier itself and the activity in question. Each classifier usually has its own peculiarities and weaknesses. For example, HMMs are of a highly sequential nature and cannot capture parallel and subevents [9]. Moreover, the scarcity of standard labeled data sets that can be employed for reliable training is a major disadvantage [18]. In addition, the high-dimensional feature space associated with extremely variable activities, such as fighting, makes this even more difficult.

As an alternative to such a learning approach, researchers have used semantics-based recognition. This enables a more meaningful definition of events, which facilitates understanding by humans. In this case, actions are described using natural language. Semantics-based recognition permits both manual [18], [19] and trainable definitions of events [20]. The case-frame representation, or $CASE$, which was proposed by Fillmore in 1968 [21], is a leading example. This representation permits writing natural language statements in terms of case frames, where each is made up of cases such as agents, predicates, locations, and objects [18]. Hakeem *et al.* [18], [21] were the first to propose an extended version of $CASE$ $(CASE^E)$, which integrated temporal logic in the form of *interval algebra* [22] into $CASE$. This permitted modeling activities that involved nonsequential subevents. Later, Hakeem and Shah [20] also introduced a probabilistic $CASE$ representation that replaced the deterministic tree representation with a learning-based probabilistic transition model.

Perhaps, the most recent work closest to ours is that conducted by Fernandez *et al.* [23]. They proposed a multilevel architecture with knowledge taxonomy. This approach belongs to a class of methods that does not require any training at all but rather is based solely on a logical description of events. Fernandez *et al.* [23] address uncertainties by using fuzzy metric-temporal Horn logic, which is a technique considerably more involved than ours. However, we note that their work lacks an appropriate and sufficient level of investigation of low-level feature processes. This is particularly significant since their experimental results incorporate a variety of human motion representations, such as body pose, motion, and face tracking, and a large taxonomy of descriptors, entities, and events. Such a high degree of complexity that entails detecting subtle behaviors, such as kicking vending machines, requires nontrivial low-level processing.

In this paper, we elaborate on the simpler and better suited approach to real-time performance provided by Fuentes and Velastin [4], which, similar to the work of Fernandez *et al.* [23], is not dependent on training or learning. In their paper, Fuentes and Velastin [4] argue that events in a transport environment can be described in terms of position, trajectory, and split/merge events. These lower level descriptors provide an initial point for the semantic implementation in this paper. However, unlike [4], we extensively discuss the algorithms and features used, starting with object detection and tracking, and ending with flagging suspicious behavior.

For a further and broader overview and discussion, readers are referred to [1], [9], and [24].

III. Object Tracking and Classification

Given an RGB video frame, we use a *Lab*-based codebook background subtraction method to segment the blobs of all foreground silhouettes. Obviously, as is well known, each blob

[1] Referred to as "analytics" in industry.

does not necessarily represent a single semantic entity. For example, a number of these might occlude each other in the scene and form a single blob from the camera's point of view. *Objects* representing semantic entities in the scene are found and tracked by matching these blobs in consecutive frames.

Traditionally, object tracking has been performed by searching the neighborhood of a previous position to predict the current one. Examples are *particle filtering* [25], [26] and *mean-shift tracking* [27]. However, as some research has pointed out [27]–[29], high-level information obtained from a preprocessing stage, such as background subtraction, can enhance the tracking task and reduce its complexity [24], [30]. We use a form of this concept, namely blob matching, to provide a feasible *real-time* solution.

The majority of object tracking methods require comparing features of two entities in two consecutive frames to compute their similarity. As would be expected, the choice of features used has a major impact on performance. Simple spatial information, such as blob overlap in consecutive frames [4], [31]–[33], has been employed extensively. However, this feature is inadequate for high-speed motion. More sophisticated features such as texture, motion, and edges [34], [35] have also been used. Nonetheless, because of its high saliency, the most reliable feature is undoubtedly *color*, whether standalone or in combination with others. Different color representations have been used in the literature, such as an object's average color [30], first-order statistical models [36], and spatiospectral *mixtures of Gaussians* [28], [31], [33]. However, (nonparametric) color histograms [31] are known for their relatively low computational complexity and ability to produce more accurate models compared with (parametric) Gaussian approximations. The former are robust to nonrigid motion [37] of the kind observed in surveillance videos, but are susceptible to clutter, noise, and fast pose changes. Other more involved similarity measures have also been proposed, such as the structural similarity index measure [26] and color spatiograms [38], [39]. Overall, color histograms seem to provide low complexity while simultaneously dealing with constantly occluding patterns. For this reason, we use them in this paper.

A. *Object Modeling and Blob-to-Object Matching*

Our object tracking approach is based on the work of Tavakkoli *et al.* [36]. At each frame, a list of objects is updated by matching blobs in the current frame with objects from the previous one. This matching process is not necessarily one-to-one. Cases of object splits, merges, one-to-one matches, creation, and deletion are all examined to ensure a correct update.

To minimize the confusion caused by the creation of false blobs by background subtraction, a notion of *reliability* is adopted from [31]. This concept dictates the inhibition and immediate discarding of objects that do not persist long enough (approximately 1–3 s) after first being detected because it is assumed that they correspond to noise or clutter.

To match blobs and objects in two consecutive frames, color histograms and spatial information are used. The color histograms are adaptively updated at 5 fps using

$$\begin{aligned}\text{Histogram}_{\text{object},\,t} \\ &= \alpha\text{Histogram}_{\text{object},\,t-1} + (1-\alpha)\text{Histogram}_{\text{blob},\,t} \quad (1)\end{aligned}$$

where α is the learning rate (empirically set by experimentation to approximately 0.6).

The intersection of the color histograms, which are based on the *Lab* color system, is computed to measure the spectral similarity between an object and a blob. This histogram distance is the fastest measure to compute, is robust to partial occlusion [40], and has good discriminative power [41]. A basic form of histogram intersection is given by the following:

$$\text{Intersection}(\text{Hist}_1,\,\text{Hist}_2) = \sum_{i=1}^{n} \min(\text{Hist}_{1,\,i},\,\text{Hist}_{2,\,i}). \quad (2)$$

Other forms of histogram intersection that include a normalization factor have also been cited [42]. However, it has been argued that such a normalization factor can be changed or omitted without any loss of generality [41]. In this paper, we devised a new normalization factor that is advantageous for the purposes of blob-to-object matching. We normalize the intersection by the largest of the histograms. The formula for histogram intersection then becomes:

$$\text{Intersection}(\text{Hist}_1,\text{Hist}_2) = \frac{\sum\limits_{i=1}^{n} \min(\text{Hist}_{1,\,i},\text{Hist}_{2,\,i})}{\max\left(\|\text{Hist}_1\|,\|\text{Hist}_2\|\right)}. \quad (3)$$

Our rationale is that the bigger the difference in size between the histograms, the less likely they are to represent the same object. Therefore, division by the largest of the histograms acts as a penalty for size mismatching.

The computed value of histogram intersection is compared with a threshold to determine whether the blob and the object are similar. Empirically, the threshold was set to values between 0.45 and 0.6 (out of 1). This threshold is a relatively high value to account for cases of background and foreground occlusion.

Many occlusion and blob-to-object assignment resolution methodologies have been used in the literature. One of the most common is *matching matrices* [4], [30], where a matrix is populated with matching scores and used to find the best blob-to-object mapping. Another approach is *iterative matching* [32], [43], where the need for a matrix is relaxed, and the object list is searched to assign the blob with the highest association probability to each object. However, by relaxing the need for a matrix, the order of matching becomes of consequence.

In this paper, we adopt the concept of *staged matching* [30], [36] to resolve this problem. According to this concept, several passes through the object list are run. The matching criterion is changed at each pass, going from the most reliable to the least. This permits the more reliable blob-to-object associations to take priority by being executed first. In our implementation, given the list of objects that spectrally match a blob, we use three consecutive spatial matching stages. The first and most reliable one is where the list is limited to recent objects whose location predicted by Kalman filtering overlaps considerably with the blob's location. In the next stage, if this is not the case, the overlap condition is relaxed, allowing for recovery in case the predicted position was erroneous. Such incorrect predictions might be due to abrupt pose changes, motion nonlinearity, or partial occlusion by the background. Finally, as a last resort, the algorithm looks for a match in even older frames whose prediction greatly overlaps with the blob. This particular stage allows for recovery from lost tracks.

B. Occlusion Handling

Occlusion handling is a critical task because it bears on the robustness of object tracking and coherence. If occlusion is resolved incorrectly, inferences following from this will most likely lead to a false understanding of the scene. In concordance with [30] and [33], we argue that finding the exact location of objects participating in occlusion within a single blob is an exhaustive search that is computationally expensive and actually unnecessary. This is because localization at the blob level provides sufficient spatial information for determining the object location. *Thus, we consider the location of a blob to be the actual location of all its constituent objects.* In this paper, the issue of which objects are occluding which is completely ignored, and we adopt the position that all merged objects form a *pool* (the blob) with no particular occluding/occluded relationships being noted. We also create a dummy object for the pool that exhibits the adaptive appearance model necessary for blob matching. In a nutshell, we render the phenomenon of occlusion into a split/merge problem. In addition, we adopt the concept of *potential occlusion* [28], which permits an object that has not *yet* been *conclusively* associated with any of the splitting blobs to be associated with *all* the accompanying splitting blobs until such time that resolution becomes conclusively possible. A video that illustrates this concept can be found at [**?**]. Of course, this may give rise to false temporary data describing an object's whereabouts. To prevent the contamination of an object's color model during occlusion, adaptive updating of the color appearance model is inhibited during this period [30], [36].

A merge is detected by checking whether a blob in the current frame covers the majority of the bounding boxes of two or more objects (or their predictions) in the previous frame. Similarly, splits are detected by finding blobs in the current frame that cover two or more centroids of the previous frame's objects (or their predictions). Merges and splits are checked before any one-to-one associations are made.

C. Object Creation and Removal

After all blob-to-object matching cases of merging, splitting, and one-to-one matches have been processed, some of the remaining blobs and objects may still remain unmatched. An unmatched blob is ideally a new object that has just appeared in the scene. Therefore, a new object is created for each unmatched blob. On the other hand, an unmatched object could also be either an object that has just left the scene or one whose blob has been falsely undetected due to some failure in background subtraction. Therefore, a grace period of a few seconds is provided to allow for the object's recovery.

D. Evaluation

Using the aforementioned techniques, our object tracking system was found to be highly reliable. This is evident in a number of tests that were performed on public data sets. References [44] and [46] are examples that demonstrate smooth tracking and occlusion handling. Videos cited in Section VII also demonstrate that our system yields reliable behavior recognition, although it is not based on using learning methods.

It is worth noting that, in spite of the robustness of our approach, failures such as lost tracks and object confusion are inevitable. However, in the majority of tests performed on a number of standard data sets, this approach was able to successfully track people and their luggage, even in circumstances that involved three or four occluding objects.

In spite of the advantages cited earlier, similar to others, the approach in this paper exhibits a few weak points. First, since the method is based mainly on color features, the tracking of objects having the same color profile could possibly fail when they jointly participate in an occlusion event. This could be mitigated using additional high-level features, such as face or clothing descriptors. However, due to the resolution and the distance at which cameras are commonly installed in such settings, we found face recognition generally inapplicable in this context. For the same reason, objects that exit the scene and re-enter later cannot be re-identified, and are considered new entities. Another shortcoming is the phenomenon of occlusion behind objects in the background. In the same vein, as discussed in Section III-C, an object is considered to have disappeared after a grace period of being unmatched. Thus, when it remains hidden behind a background entity, such as a pillar, for longer than the grace period and then re-appears, this object is not associated with the original one, but rather considered new. Again, face or clothing recognition might be invoked, but this is beyond the scope of this paper.

A more detailed qualitative and quantitative evaluation can be found in [47].

IV. Feature Calculation

After determining the objects of interest in the video, their 3-D *motion features* are calculated, and an historical record is created. Based on this record, objects are classified as being either animate (persons) or inanimate. This classification process is important because it is an integral component of the definition of semantic behavior. There are many potential features discussed in the literature [14], [15], [48]. These can be split into single-object features, such as position, and interobject features, such as the alignment between two objects. Table I lists the features selected that were found to be most relevant for the types of behavior investigated in this paper. These features are measured in real-world 3-D spatial coordinates, which can be calculated from the image (pixel) coordinates by means of any traditional camera calibration method. The position of an object, in terms of which almost all the rest of its features are calculated, is obtained by applying the transformation to the pixel locations of the feet. These are simply designated as the lowest pixels of the 2-D blob to which the object belongs.

TABLE I
THREE-DIMENSIONAL OBJECT FEATURES. SUBSCRIPTS AND SUPERSCRIPTS DESCRIBE OBJECT ID(S) AND TIME, RESPECTIVELY

	Feature	Definition	Formula
Single-object features	Position	Coordinates of the object after applying camera calibration.	$\overrightarrow{p_i^t} = (x_i^t, y_i^t)$
	Speed	A scalar representing the magnitude of the change in position in 3D coordinates with respect to a chosen origin.	$s_i^t = \frac{\left\Vert \overrightarrow{p_i^t} - \overrightarrow{p_i^{t-\Delta t}} \right\Vert}{\Delta t}$
	Direction	A unit vector in the direction of motion.	$v_i^t = \frac{\overrightarrow{p_i^t} - \overrightarrow{p_i^{t-\Delta t}}}{\left\Vert \overrightarrow{p_i^t} - \overrightarrow{p_i^{t-\Delta t}} \right\Vert}$
	Merged	A Boolean feature set to true when an object is merged with another.	More than one object mapped to the same blob.
Two-object features	Distance	Distance between two objects.	$d_{i,j}^t = \left\Vert \overrightarrow{p_i^t} - \overrightarrow{p_j^t} \right\Vert$
	Alignment	Cosine of the angle between the velocity vectors of two objects.	$al_{i,j}^t = v_i^t . v_j^t$
	Speed difference	A scalar representing the difference in speed between two objects.	$\Delta s_{i,j}^t = \left\vert s_i^t - s_j^t \right\vert$

The 3-D positions of the objects are computed by finding the real-world coordinates of the lowest pixel in the blob as seen in the 2-D image. Because blobs can have irregular shapes, the calculated position of the object is often erratic if sampled at high frame rates. This erratic movement generally dominates the detection of the true motion of the object [15]. To stabilize the computation of motion, features are computed at a rate of 1 fps, which is five times slower than the object segmentation and tracking rate.[2]

In addition to the single-object features, the inter-object features between *every* combination of two objects are also stored in historical sequence.

Next, the object category is determined as being either animate (e.g., human) or inanimate (e.g., luggage) based on a blob's ability to seemingly move about independently. The most commonly used feature for this purpose is geometrical shape [13], [49], [50]. However, this type of feature often fails due to the high variability of an object's silhouette caused by noise, pose variations, and complex occlusion scenarios.

In this paper, we follow the concept in [12], which uses *motion features* to classify objects into four categories: *unknown* U, *abandoned object* O, *person* P, and *still person* SP. Exploiting motion features prevents errors resulting from the use of shape-based features. Fig. 1 shows the state diagram for the implemented algorithm, which is largely adapted from [12] with a few minor modifications. The algorithm distinguishes between an inanimate object and a still person, a subtlety highly important for its consequences in understanding the scene. The threshold is chosen to tolerate small errors and perturbations in the measured position of an object. In this paper, $v_{\text{Threshold}} = 0.5$ m/s was used.

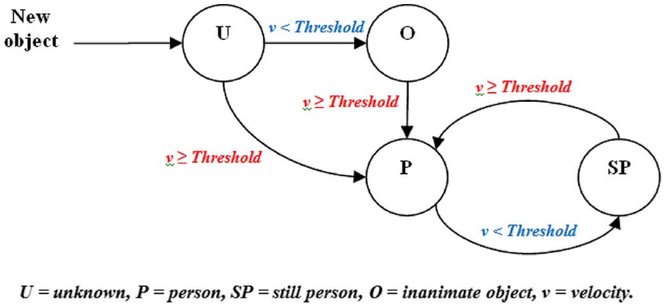


Fig. 1. Object classification state diagram. When a new object occurs in the scene, it is classified as unknown. When its motion features are sampled, the velocity is used to determine whether it is a person or an inanimate object. Using this transition model ensures that a still person is not misclassified as luggage.

Finally, the "owners" of inanimate objects are also identified. This is critical for determining whether a bag is attended, abandoned, or stolen. When an object is first detected, the closest person is considered to be the potential owner. This becomes effective when the object is actually classified as inanimate.

V. BEHAVIOR SEMANTICS

The extracted features discussed in Section IV are used to semantically define and detect activities in the scene. The historical record that encapsulates the features of each object and pair of objects is updated every second. Based on these, a set of conditions for each activity of interest is retrospectively tested (in this paper, once every second). If the conditions in question are satisfied, that behavior is flagged.

In the following, behaviors of interest are discussed individually in depth. Appropriate examples and discussions are given.

VI. DEFINING BEHAVIORS OF INTEREST

A. Abandoned and Stolen Objects

A major concern in the literature to date has been the detection of abandoned luggage. Generally, detection has been performed using only background subtraction methods, such as [16] and [17], without other forms of reasoning such as object classification and tracking. The problem with this is that such an approach cannot discriminate between a stationary person and an abandoned object. Other methods use features such as color, edges, shape completeness, and histogram contrast [51]. In our experience, none of these was found to be sufficiently robust to noise and pose changes. Moreover, the issue of finding the object's owner is still inadequately addressed. This is crucial, for example, to the distinction between stolen and retrieved luggage.

This paper addresses the aforementioned shortcomings using a semantic definition. We use the definition in [12], which defines an abandoned object as "a stationary object that has not been touched by a person for some time threshold." Integrating the object ownership into this statement, Table II shows our refined definitions of abandoned and stolen luggage, which are based on the motion features introduced in Section IV.

[2] All the simulations and examples in this paper were run on an Intel Core i5 CPU @ 3.47 GHZ with 4 GB of RAM.

TABLE II
DEFINITIONS OF ABANDONED AND STOLEN LUGGAGE

Relation	Semantic Description
obj_1 abandoned obj_2	$classification(obj_1) \in \{P, SP\}$ $classification(obj_2) \in O$ $obj_1 = owner(obj_2)$ for all $t \geq t_{\mathbf{abandoned}} : d_{obj_1, obj_2} > d_{\mathbf{abandoned}}$
obj_1 stole obj_2	$classification(obj_1) \in \{P, SP\}$ $classification(obj_2) \in O$ $obj_1 \neq owner(obj_2)$ obj_1 takes obj_2
obj_1 takes obj_2	$merged(obj_1, obj_2) = true$ $d_{P^{t=0}_{obj_1}, P^{t=t_{now}}_{obj_2}} > d_{\mathbf{abandoned\_luggage}}$

(P = person, SP = still person, O = inanimate object.)

For a real-life situation, [52] suggests using $d_{\text{abandoned}} = 3$ m and $t_{\text{abandoned}} = 30$ s.

B. Loitering

Loitering is useful for detecting a number of public transit situations such as drug dealing [11]. In [24], it is defined as "the presence of an individual in an area for a period of time longer than a given time threshold." This is semantically translated into

$$\text{Obj}_1 \text{ is loitering} \equiv \text{classification}(\text{Obj}_1) \in \{P, SP\} \wedge \text{lifetime}(\text{Obj}_1) \geq t_{\text{loitering}}. \quad (4)$$

In a real-life scenario, the threshold should be set in terms of the public transportation waiting time. PETS 2007 [53] uses $t_{\text{loitering}} = 60$ s, a figure which might seem a little low in current circumstances.

C. Fighting

Fighting is one of the most challenging activities to characterize. Some researchers use the "black box" approach by training a classifier on a number of "believed-to-be" event characteristics, such as shape-related features [54]. However, the descriptive power of these features is often inadequate and may lead to overfitting, particularly when a sufficient number of training videos is unavailable, which is usually the case. Motion-related features have also been used to train a classifier [15]. Semantically speaking, fights are defined by groups of blob centroids "moving together, merging and splitting, and overall fast changes in the blobs' characteristics." [4] This definition can be translated into a set of conditions involving the motion features of the objects (see Section IV).

Our experiments indicate that the most reliable means for defining fights is in terms of the frequency of object splitting and merging. A *"minimal speed in different directions"* condition was also added to ensure the presence of a highly dynamic level of activity. Thus, we defined the semantics for fighting, as shown in equation (5).

$$\begin{aligned} \text{obj}_1 \text{ fights with obj}_2 \equiv\ & \text{classification}(\text{obj}_1) = P \\ & \wedge \text{classification}(\text{obj}_2) = P \\ & \wedge \left(\frac{\#\text{ splits and merges}}{t_{\text{fight}}} > f_{\text{split/merge}_{\min}} \right) \\ & \wedge \text{for all splits} : \Big(\text{al}_{\text{obj}_1, \text{obj}_2} < \cos(\theta_{\text{fight}_{\min}}) \\ & \qquad \wedge (s_{\text{obj}_1} > s_{\text{fight}_{\min}}) \\ & \qquad \wedge (s_{\text{obj}_2} > s_{\text{fight}_{\min}}) \Big). \end{aligned} \quad (5)$$

The thresholds were empirically set to:

$$f_{\text{split/merge}_{\min}} \in \{3, 4\},\ t_{\text{fight}} \in [3, 7]$$
$$s_{\text{fight}_{\min}} \in [0.5, 1] \text{ m/s},\ \theta_{\text{fight}_{\min}} = 40^\circ$$

D. Meeting and Walking Together

Although generally not considered to be suspicious, meeting and walking together may be useful in certain surveillance scenarios. This would be particularly the case were face recognition included as a feature. For example, it might be pertinent for security purposes to flag individuals that meet with a suspicious individual. Table III defines both events semantically in terms of each person's speed, the distance between them, and their alignment.

The values of the thresholds were experimentally determined as

$$\begin{aligned} s_{\text{walking}_{\min}} &= 0.5 \text{ m/s},\ t_{\text{walking\_together}} \in [0.5, 3] \text{ s} \\ \Delta s_{\text{walking}_{\max}} &\in [0.5, 1] \text{ m/s},\ \theta_{\text{walking\_together}_{\max}} = 40^\circ \\ d_{\text{meeting}_{\max}} &\in [1.5, 2] \text{ m}. \end{aligned}$$

E. Fainting

The aspect ratio of a person's silhouette is perhaps the most widely used feature for fainting detection [4], [55], [56]. Despite its popularity and simplicity, this approach is neither view invariant [55] nor robust to camera nonlinearities. Fig. 2 shows an example.

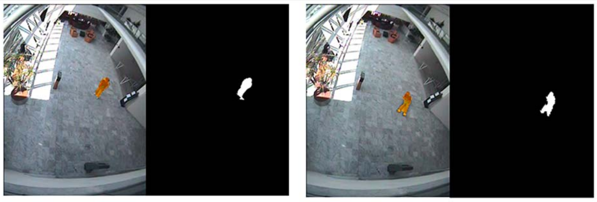
Shoaib *et al.* [55] solve this problem in three dimensions, using the distance between the detected head location and the mean head location in 2-D, based on the average human height. The decision is then made using this distance.

In this paper, we use the camera calibration method in [58] to resolve the alignment issue in 3-D and account for any nonlinearities in the camera parameters. Assuming the person to be standing, the hypothesized 2-D location of the feet on the floor is computed. To verify this assumption, this location is compared with the actual detected location of the feet in 3-D. The actual real-world coordinates $(x_{\text{feet}}, y_{\text{feet}})$ and $(x_{\text{head}}, y_{\text{head}})$ are determined using the 3-D image coordinates of the lowest and highest pixels of the 2-D blob in the image, respectively. The location of the third world coordinate in 3-D space

TABLE III
DEFINITIONS OF MEETING AND WALKING TOGETHER

Relation	Semantic Description
obj₁ meets obj₂	$classification(obj_1) \in \{P, SP\}$ $classification(obj_2) \in \{P, SP\}$ Obj_1 is close to obj_2 $(s_{obj_1} \leq s_{\mathbf{walking}_{min}})$ $(s_{obj_2} \leq s_{\mathbf{walking}_{min}})$
obj₁ walks with obj₂	$classification(obj_1) \in \{P, SP\}$ $classification(obj_2) \in \{P, SP\}$ obj_1 is close to obj_2 for $t < t_{\mathbf{walking\_together}} : ((s_{obj_1} > s_{\mathbf{walking}_{min}})$ $\wedge (s_{obj_2} > s_{\mathbf{walking}_{min}})$ $\wedge (\Delta s_{obj_1,\, obj_2} \leq \Delta s_{\mathbf{walking\_together}_{max}})$ $\wedge (al_{obj_1,\, obj_2} > \cos(\theta_{\mathbf{walking\_together}_{max}}))$
obj₁ is close to obj₂	$(d_{obj_1,\, obj_2} < d_{\mathbf{meeting}_{max}})$ $\vee (merged(obj_1, obj_2) = true)$

P = person, SP = still person, O = inanimate object.



The standing person (at left, in orange) and the one that fainted (right) have almost the same silhouette aspect ratio, proving that this property is insufficient for faint detection. Frames were taken from PETS 2004 [57]

Fig. 2. Insufficiency of a blob's aspect ratio for faint detection.

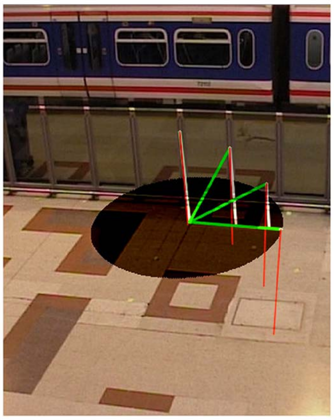
is assumed to be given by $z_{\text{feet}} = 0$ m and $z_{\text{head}} = 1.5$ m. If a person is not actually standing, z_{head} would be less than 1.5 m, and consequently, $(x_{\text{head}}, y_{\text{head}})$ would be incorrect. Therefore, the computed distance between the planar coordinates of the head and feet would be zero for a perfectly standing person and would grow larger as the posture deviated from the vertical. Fig. 3 demonstrates this concept visually.

Consequently, the definition of *fainting* was determined as follows:

$$\begin{aligned} \text{obj}_1 \text{ falls} &\equiv d(p_{\text{feet}_{\text{detected}}}, p_{\text{head}_{\text{assuming } z_{\text{head}}=1.5\text{ m}}}) > d_{\text{faint}} \\ \text{obj}_1 \text{ faints} &\equiv \text{classification}(\text{obj}_1) \\ &= SP \wedge \text{obj}_1 \text{ falls for } t \geq t_{\text{faint}}. \end{aligned} \tag{6}$$

The value of d_{faint} was found experimentally to be approximately the height of an average person and was fine-tuned so that a fainting person was detected for all possible yaw angles.

The green lines represent a person at different pitch angles. At each angle, the white lines represent the correct head height. However, since this information cannot be obtained using only the extracted blobs (2D information only), it is necessary to make the additional assumption that the head is at the elevation of an average person (1.5 meters). The red lines represent head heights under this assumption.



Notice that the actual projection (white) and the presumed projection (red) diverge as the pitch angle decreases, in other words, as the person falls. The black circle represents a chosen threshold. If the assumed projection of the head lies outside this threshold, the person is considered to be falling.

Reference [59] shows that this approach is valid for all yaw angles by illustrating that, for all possible yaw angles, the red line for a standing person is inside the circle, while the red line for a fainting person is outside the circle.

Fig. 3. Faint detection using geometrical distances at different pitch angles.

In addition, t_{faint} must be chosen to take into account the local situation. Fig. 3 shows four pitch angles captured from [59].

VII. EXPERIMENTAL RESULTS

The *evaluation* of behavior recognition experiments is challenged by a number of difficulties at several levels [60]. First, most activities of interest are of high complexity, which becomes an issue in the presence of clutter in the test scenario. Another issue is the inadequacy of professional and challenging high-quality data sets currently available for testing. Moreover, criteria for performance evaluation, such as a standard metric, hit-and-miss weighting, and the construction of the ground truth, are still subject to controversy. These challenges lead to inconsistencies among the experimental results in different papers in the literature.

In this paper, carefully selected standard public data sets were used to test the proposed framework. These data sets are BEHAVE [14], CAVIAR (PETS 2004) [57], and PETS 2006 [52]. To our surprise, very little work other than ours uses standard data sets. The trend toward focusing on private and often nonprofessional data sets is a major limiting factor in the field's development.

Our framework successfully detects all of the events discussed earlier, performing object tracking in an average time of 11 ms per object per frame, whereas behavior recognition averages just about 1 ms per frame. These components, along with background subtraction, constitute the total processing time required per frame, which is approximately 200 ms. Fig. 4 shows the key frames of some of the tested scenarios from the selected data sets. The indicated web links are to the complete videos.

In the following, Tables IV and V demonstrate the performance of our framework compared with (1) ground truth and (2) other results, respectively.

Table IV compares the performance of our framework with a data set's ground truth by providing the precision and recall

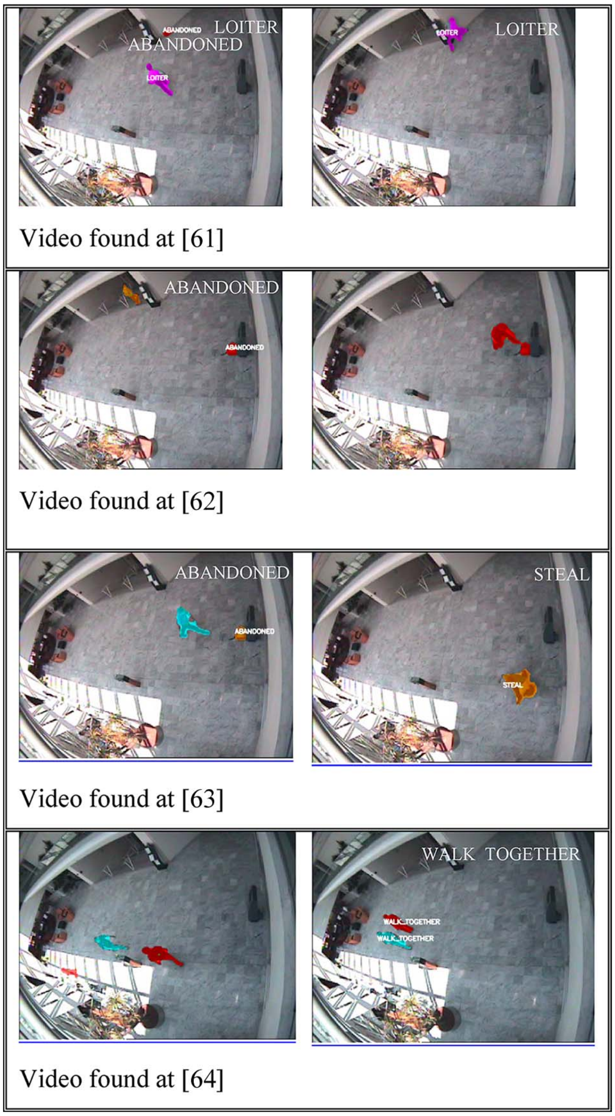


Fig. 4. Snapshots of the experimental results on different standard data sets.

scores for the selected scenarios. To calculate these metrics for each frame, an incorrectly raised alarm is considered to be a false positive, whereas a missed detection is considered a false negative. Detection of abandoned luggage, stolen luggage, loitering, and fainting is highly reliable. In addition, for the most part, the precision is usually higher than the recall. This means that the framework is biased toward raising less false alarms, which is usually considered to be more favorable than obtaining less missed detections [1]. The low performance of "meeting" and "walking together" in the CAVIAR data set [57] is caused by the very short duration of these events and the extreme nonlinearity of the camera calibration model.

To further demonstrate the performance of our framework, it was also compared with the relatively few quantitative results *based on publicly available data sets in the literature*. Table V compares our results with three of the publications that provide such results. Judged against [4] and [79], it is evident that our framework yields higher precision while still maintaining a relatively high recall. It is difficult to explain why this is so, particularly because of the inaccessibility of the videos used to obtain their results. However, we surmise that the relatively greater complexity of our approach to blob detection and merging is the reason.

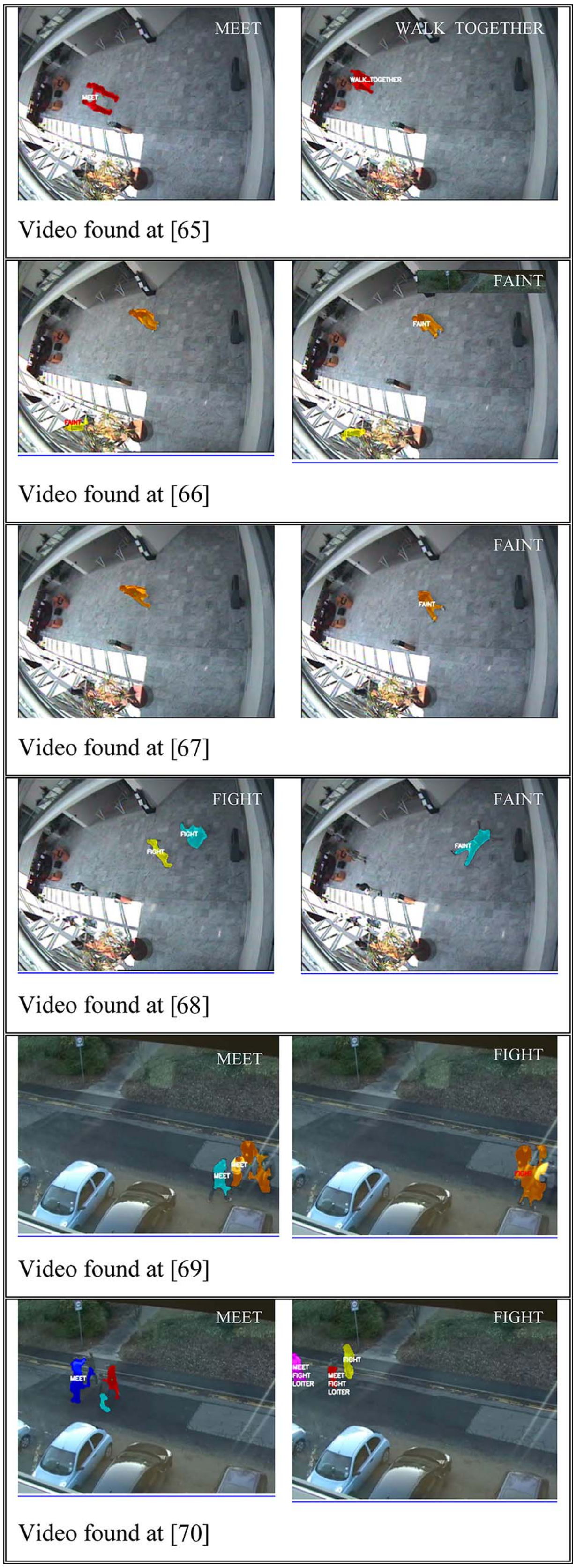
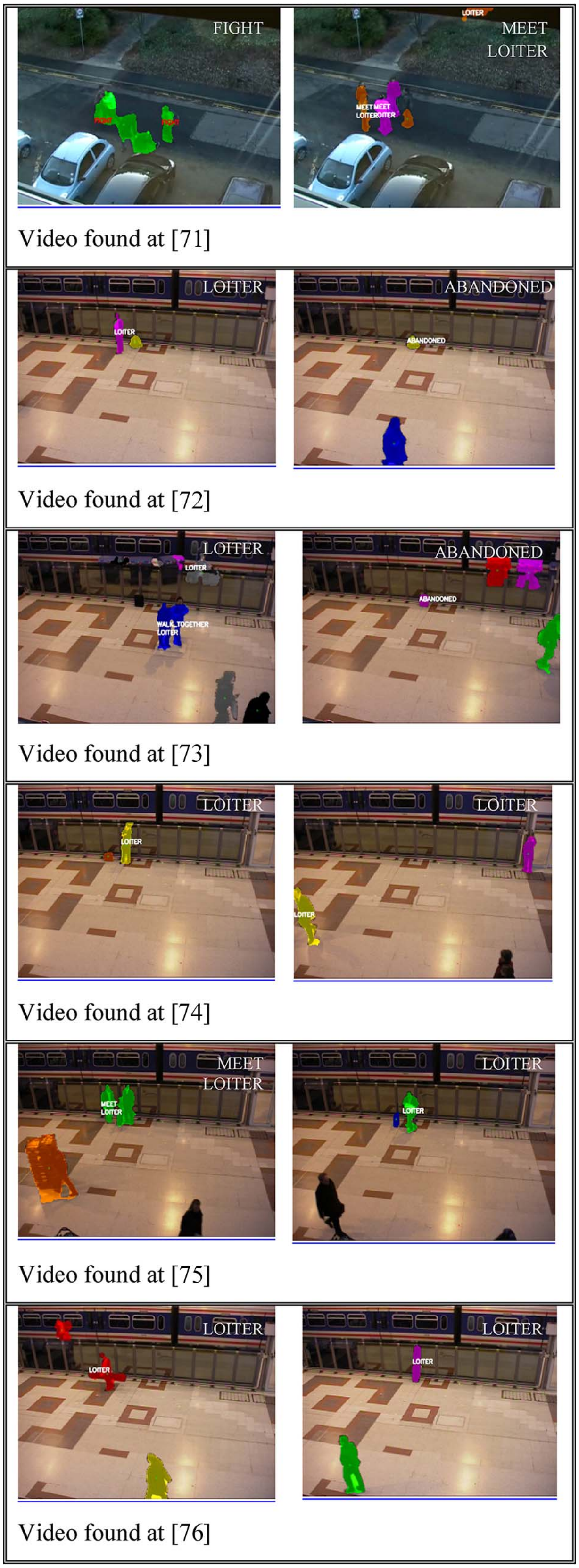


Fig. 4. *(Continued.)*



Video found at [71]

Video found at [72]

Video found at [73]

Video found at [74]

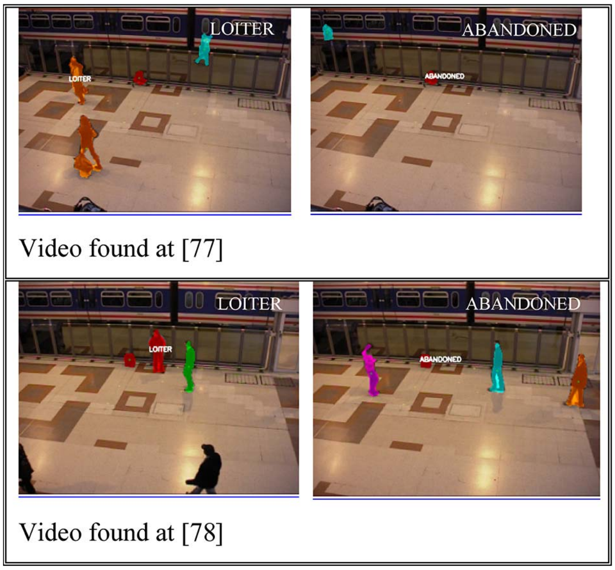
Video found at [75]

Video found at [76]

Fig. 4. *(Continued.)*

VIII. Threshold and Parameter Sensitivity

Setting thresholds and parameters is always a controversial issue, often generating suspicion. We note that most of the high-level parameters that define the semantic behaviors, such as acceptable walking speeds and meeting distances, were set using physical and logical reasoning. However, certain sensitive parameters describing complex types of behavior, such as fighting, required experimentation and fine-tuning. This is because such behaviors manifest a broad range of variations and are very difficult to model, even if based on reasoning. Furthermore, somewhat more than other types of behavior, they depend on a number of environmental factors, such as the physical setup and lighting conditions. Therefore, as might be expected in this case, it was necessary to adjust the values of the parameters for each individual database.



Video found at [77]

Video found at [78]

Fig. 4. *(Continued.)*

We observe that parameter tuning can be interpreted as being analogous to the problem of undertraining in machine learning since both represent a certain deficiency of knowledge. However, the semantic approach has the advantage of permitting human reasoning to easily model parameter values (e.g., speeds, distances, and angles). This is contrasted to the difficulty of finding sufficiently large and meaningful data sets for training machine learning systems. Of course, learning also requires fine-tuning of parameters, such as neural network size, connections, as well as learning parameters. Ultimately, machine-learning approaches in the current literature seem to be unable to generalize and systems based on semantics.

IX. Conclusion

In this paper, a complete semantics-based behavior recognition approach that depends on object tracking has been introduced and extensively investigated. Our approach begins by translating the objects obtained by background segmentation [47] into semantic entities in the scene. These objects are tracked in 2-D and classified as being either animate (people) or inanimate (objects). Then, their 3-D motion features are calculated and recorded in the form of historical records. Ultimately, behaviors are semantically defined and detected by continuously checking these records against predefined rules

TABLE IV
EXPERIMENTAL RESULTS FOR BEHAVIOR RECOGNITION USING DIFFERENT STANDARD PUBLIC DATA SETS. AL = ABANDONED LUGGAGE, TL = THEFT OF LUGGAGE, L = LOITERING, M = MEETING, WT = WALKING TOGETHER, FA = FAINTING, FI = FIGHTING

Dataset/Scene	Behavior of Interest	Quantitative Evaluation (Precision / Recall)	Qualitative Evaluation
CAVIAR/ Leftbag_pickup	AL	89%/ 77%	Successful detection.
	L	98%/100%	
CAVIAR/ leftbag (retrieved)	AL	93%/82%	Successful detection.
	M	73%/47%	
	WT	70%/35%	
CAVIAR/ leftbag (theft)	AL	81%/ 100%	Successful detection.
	TL	100%/ 100%	
	M	36%/ 40%	
	WT	100%/ 19%	
CAVIAR/ meet_walk_split	M	100%/ 0%	Successful detection of walking together. Failed detection of meeting due to its very short time.
	WT	92%/ 79%	
CAVIAR/ meet_walk _together	M	27%/ 40%	Successful detection of walking together. Failed detection of meeting due to its very short time.
	W	100%/ 60%	
CAVIAR/ rest_fallonfloor	FA	100%/ 80%	Successful detection
CAVIAR/rest_ wiggleonfloor	FA	80%/ 69%	Successful detection
CAVIAR/fight_o ne_man_down	FI	63%/ 31%	Successful detection
	FA	100%/ 67%	
BEHAVE/ frames 67210-76800	M	100%/ 86%	Successful detection
	FI	62%/ 55%	
BEHAVE/ frames 46200- 46323	FI	100%/ 32%	Successful detection
BEHAVE/ frames 50375- 50932	M	69%/ 45%	Successful detection
	FI	50%/ 46%	
PETS2006/ S1C3	AL	97%/ 100%	Successful detection
	L	100%/ 98%	
PETS2006/ S2C3	AL	100%/ 98%	Successful detection
	L	100%/100%	
	M	100%/92%	
PETS2006/ S3C3	L	100%/ 83%	Successful detection
PETS2006/ S4C3	L	100%/ 93%	Successful detection
	M	100%/ 93%	
PETS2006/ S5C3	AL	100% / 0%	Detection failed because of a failure in tracking and classifying the objects in question
	L	25%/ 61%	
PETS2006/ S6C3	AL	95%/ 100%	Successful detection
	L	98%/ 100%	
PETS2006/ S7C3	AL	100%/ 100%	Successful detection
	L	100%/ 100%	
AVSS 2007 i-LIDS/ AVSS_AB_Easy	AL	100%/ 68%	Abandoned luggage detected successfully. However, it fails after a while due to a failure in tracking.
	L	96%/ 81%	

TABLE V
COMPARING OUR RESULTS OF BEHAVIOR RECOGNITION FOR PUBLICLY AVAILABLE DATA SETS FOUND IN THE LITERATURE WITH OTHER FRAMEWORKS. AL = ABANDONED LUGGAGE, TL = THEFT OF LUGGAGE, L = LOITERING, M = MEETING, WT = WALKING TOGETHER, FA = FAINTING, FI = FIGHTING, P = PRECISION, R = RECALL

Paper	Dataset/ Scene	Behavior of interest	Evaluation – Their work	Evaluation -Our work
Tracking-based event detection for CCTV systems [4]	CAVIAR/ leftbag	AL	P = 57%	P = 93%
	CAVIAR/ Leftbag _pickup	AL	P = 81%	P = 89%
	CAVIAR/ fight_one_ man_down	FI	P = 50%	P = 63%
	CAVIAR/ meet_walk _split	M, WT	P = 100%	P = 94%
An Abandoned Object Detection	PETS2006/ S1C3	AL	Correct detection (0 seconds difference)	Correct detection (0 seconds difference)
System Based on Dual Background Segmentation [17]	PETS2006/ S2C3	AL	Late detection (1 second difference)	Late detection (2 seconds difference)
	PETS2006/ S5C3	AL	Correct detection (0 seconds difference)	Failed
	PETS2006/ S6C3	AL	Early detection (1 seconds difference)	Correct detection (0 seconds difference)
	PETS2006/ S7C3	AL	Correct detection (0 seconds difference)	Correct detection (0 seconds difference)
	AVSS 2007 i-LIDS/ AVSS_AB_E asy	AL	Early detection (2 seconds difference)	Correct detection (0 seconds difference)

Comparative Evaluation of Stationary Foreground Object Detection Algorithms Based on Background Subtraction Techniques [79]	PETS2006 (overall performance) / Approach 1	AL	P = 5%
			R = 100%
				P = 99%
				R = 80%

	PETS2006 (overall performance) / Approach 2	AL	P = 60%
			R = 100%
				P = 99%
				R = 80%

	PETS2006 (overall performance) / Approach 3	AL	P = 50%
			R = 100%
				P = 99%
				R = 80%

	PETS2006 (overall performance) / Approach 4	AL	P = 75%
			R = 100%
				P = 99%
				R = 80%

	PETS2006 (overall performance) / Approach 5	AL	P = 37%
			R = 100%
				P = 99%
				R = 80%

and conditions. This approach ensures real-time performance, adaptability, robustness against clutter and camera nonlinearities, ease of interfacing with human operators, and elimination of the training required by machine-learning-based methods. Experimentation was carried out on multiple standard publicly available data sets that varied in terms of crowd density, camera angle, and illumination conditions. The experimental results demonstrated successful detection of the various activities of interest.

REFERENCES

[1] H. Weiming, T. Tieniu, W. Liang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[2] G. L. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis, "Active video-based surveillance system: The low-level image and video processing techniques needed for implementation," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 25–37, Mar. 2005.

[3] N. Firth, Face recognition technology fails to find U.K. rioters, New-Scientist, London, U.K. [Online]. Available: http://www.newscientist.com/article/mg21128266.000-face-recognition-technology-fails-to-find-uk-rioters.html

[4] L. M. Fuentes and S. A. Velastin, "Tracking-based event detection for CCTV systems," *Pattern Anal. Appl.*, vol. 7, no. 4, pp. 356–364, Dec. 2004.

[5] M. Elhamod and M. D. Levine, "A real time semantics-based detection of suspicious activities in public scenes," in *Proc. 9th Conf. CRV*, Toronto, ON, Canada, 2012, pp. 268–275.

[6] N. T. Siebel and S. J. Maybank, "The ADVISOR visual surveillance system," in *Proc. ECCV Workshop ACV*, 2004, pp. 103–111.

[7] Z. Zhang, T. Tieniu, and H. Kaiqi, "An extended grammar system for learning and recognizing complex visual events," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 240–255, Feb. 2011.

[8] D. Demirdjian and C. Varri, "Recognizing events with temporal random forests," in *Proc. Int. Conf. Multimodal Interfaces*, Cambridge, MA, 2009, pp. 293–296.

[9] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Underst.*, vol. 115, no. 2, pp. 224–241, Feb. 2011.

[10] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space–time shapes," in *Proc. 10th IEEE ICCV*, 2005, vol. 2, pp. 1395–1402.

[11] N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs, "Detection of loitering individuals in public transportation areas," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 167–177, Jun. 2005.

[12] N. Bird, S. Atev, N. Caramelli, R. Martin, O. Masoud, and N. Papanikolopoulos, "Real time, online detection of abandoned objects in public areas," in *Proc. IEEE ICRA*, 2006, pp. 3775–3780.

[13] L. Sijun, Z. Jian, and D. Feng, "A knowledge-based approach for detecting unattended packages in surveillance video," in *Proc. IEEE AVSS*, 2006, p. 110.

[14] S. Blunsden and R. B. Fisher, "The BEHAVE video dataset: Ground truthed video for multi-person behavior classification," *Annu. BMVA*, vol. 2010, no. 4, pp. 1–11, 2010.

[15] S. Blunsden, E. Andrade, and R. Fisher, "Non parametric classification of human interaction," in *Proc. 3rd Iberian Conf. Pattern Recog. Image Anal., Part II*, Girona, Spain, 2007, pp. 347–354.

[16] P. Fatih, "Detection of temporarily static regions by processing video at different frame rates," in *Proc. IEEE Conf. AVSS*, 2007, pp. 236–241.

[17] A. Singh, S. Sawan, M. Hanmandlu, V. K. Madasu, and B. C. Lovell, "An abandoned object detection system based on dual background segmentation," in *Proc. 6th IEEE Int. Conf. AVSS*, 2009, pp. 352–357.

[18] A. Hakeem, Y. Sheikh, and M. Shah, " CASE$^{\mathrm{E}}$: A hierarchical event representation for the analysis of videos," in *Proc. 19th Nat. Conf. Artif. Intell.*, San Jose, CA, 2004, pp. 263–268.

[19] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 171–184, Nov. 2002.

[20] A. Hakeem and M. Shah, "Learning, detection and representation of multi-agent events in videos," *Artif. Intell.*, vol. 171, no. 8/9, pp. 586–605, Jun. 2007.

[21] C. J. Fillmore, The Case for Case, 1967. [Online]. Available: http://linguistics.berkeley.edu/~syntax-circle/syntax-group/spr08/fillmore.pdf

[22] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, Nov. 1983.

[23] C. Fernandez, P. Baiget, X. Roca, and J. Gonzalez, "Interpretation of complex situations in a semantic-based surveillance framework," *Image Commun.*, vol. 23, no. 7, pp. 554–569, Aug. 2008.

[24] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi, "Understanding transit scenes: A survey on human behavior-recognition algorithms," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 206–224, Mar. 2010.

[25] Y. Changjiang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Proc. 10th IEEE ICCV*, 2005, vol. 1, pp. 212–219.

[26] A. Loza, W. Fanglin, Y. Jie, and L. Mihaylova, "Video object tracking with differential Structural SIMilarity index," in *Proc. IEEE ICASSP*, 2011, pp. 1405–1408.

[27] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[28] V. Papadourakis and A. Argyros, "Multiple objects tracking in the presence of long-term occlusions," *Comput. Vis. Image Underst.*, vol. 114, no. 7, pp. 835–846, Jul. 2010.

[29] F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveil.*, 2003, pp. 37–45.

[30] S. S. Intille, J. W. Davis, and A. F. Bobick, "Real-time closed-world tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 1997, pp. 697–703.

[31] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Comput. Vis. Image Underst.*, vol. 80, no. 1, pp. 42–56, Oct. 2000.

[32] A. Tesei, A. Teschioni, C. Regazzoni, and G. Vernazza, "'Long-memory' matching of interacting complex objects from real image sequences," in *Proc. Conf Time Varying Image Process. Moving Objects Recog.*, 1996, pp. 283–288.

[33] M. Dahmane and J. Meunier, "Real-time video surveillance with self-organizing maps," in *Proc. 2nd Can. Conf. Comput. Robot Vis.*, 2005, pp. 136–143.

[34] R. T. Collins, L. Yanxi, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[35] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Mach. Vis. Appl.*, vol. 8, no. 3, pp. 187–193, 1995.

[36] A. Tavakkoli, M. Nicolescu, and G. Bebis, "A spatio-spectral algorithm for robust and scalable object tracking in videos," in *Proc. 6th Int. Conf. Adv. Vis. Comput.—Volume Part III*, Las Vegas, NV, 2010, pp. 161–170.

[37] M. H. Y. Liao, C. Duan-Yu, S. Chih-Wen, and T. Hsiao-Rang, "Real-time event detection and its application to surveillance systems," in *Proc. IEEE ISCAS*, Island of Kos, Greece, 2006, pp. 4 pp.-512.

[38] S. T. Birchfield and R. Sriram, "Spatial histograms for region-based tracking," *ETRI J.*, vol. 29, no. 5, pp. 697–699, Oct. 2007.

[39] S. T. Birchfield and R. Sriram, "Spatiograms versus histograms for region-based tracking," in *Proc. IEEE CVPR*, 2005, vol. 2, pp. 1158–1163.

[40] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, Nov. 1991.

[41] I. R. Khan and F. Farbiz, "A new similarity measure and back-projection scheme for robust object tracking," in *Proc. ISCIT*, 2010, pp. 412–417.

[42] A. Vadivel, A. K. Majumdar, and S. Shamik, "Performance comparison of distance metrics in content-based image retrieval applications," in *Proc. Int. Conf. Inf. Technol.*, Bhubaneswar, India, 2003, pp. 159–164.

[43] O. Oreifej, R. Mehran, and M. Shah, "Human identity recognition in aerial images," in *Proc. IEEE CVPR*, 2010, pp. 709–716.

[44] M. Elhamod. (2012). Potential Occlusion. [Online]. Available: http://cim.mcgill.ca/~mndhamod/BehaviourPaper/PotentialOcclusion.pdf

[46] M. Elhamod. (2011). Object Tracking Demo: Bag_sequence from Papadourakis and Argyros. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/Bag_sequence_Papadourakis_and_Argyros.avi

[46] M. Elhamod. (2011). Object Tracking Demo: Video8 from VISOR dataset. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/VISOR_video8.avi

[47] M. Elhamod, "Real-time automated annotation of surveillance scenes," M.S. thesis, Dept. Elect. Comput. Eng., McGill Univ., Montreal, QC, Canada, 2012.

[48] K. Terzic, L. Hotz, and B. Neumann, "Division of work during behaviour recognition—The SCENIC approach," in *Proc. BMI*, 2007, pp. 144–159.

[49] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Proc. 4th IEEE WACV*, 1998, pp. 8–14.

[50] L. Weilun, H. Jungong, and P. H. N. De With, "Automatic video-based human motion analyzer for consumer surveillance system," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, pp. 591–598, May 2009.

[51] W. Jianting, G. Haifeng, Z. Xia, and H. Wenze, "Generative model for abandoned object detection," in *Proc. 16th IEEE ICIP*, 2009, pp. 853–856.

[52] PETS 2006 Benchmark Data. (2006). [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2006/data.html

[53] PETS 2007 Benchmark Data. (2007). [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2007/data.html

[54] Y. Jie, C. Jian, and L. Hanqing, "Human activity recognition based on the blob features," in *Proc. IEEE ICME*, 2009, pp. 358–361.

[55] M. Shoaib, R. Dragon, and J. Ostermann, "View-invariant fall detection for elderly in real home environment," in *Proc. 4th PSIVT*, 2010, pp. 52–57.

[56] V. Vaidehi, K. Ganapathy, K. Mohan, A. Aldrin, and K. Nirmal, "Video based automatic fall detection in indoor environment," in *Proc. ICRTIT*, 2011, pp. 1016–1020.

[57] PETS-ECCV. (2004). [Online]. Available: http://www-prima.imag.fr/PETS04/caviar_data.html

[58] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.

[59] M. Elhamod. (2011). Faint Detection Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/FaintDemo.avi

[60] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *Proc. 37th IEEE AIPR*, 2008, pp. 1–8.

[61] M. Elhamod. (2011). CAVIAR Dataset: Left Bag and Pick up Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_LeftBag_PickedUp.avi

[62] M. Elhamod. (2011). CAVIAR Dataset: Left Bag and Retrieved Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_LeftBag(retrieved).avi

[63] M. Elhamod. (2011). CAVIAR Dataset: Left Bag and Stolen Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_LeftBag(stolen).avi

[64] M. Elhamod. (2011). CAVIAR Dataset: Meet, Walk, and Split Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_Meet_WalkSplit.avi

[65] M. Elhamod. (2011). CAVIAR Dataset: Meet and Walk Together Demo 1. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_Meet_WalkTogether1.avi

[66] M. Elhamod. (2011). CAVIAR Dataset: Rest and Wiggle on the Floor Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_Rest_WiggleOnFloor.avi

[67] M. Elhamod. (2011). CAVIAR Dataset: Rest and Fall on the Floor Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_Rest_FallOnFloor.avi

[68] M. Elhamod. (2011). CAVIAR Dataset: Fight and One Man Down Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/CAVIAR_Fight_OneManDown.avi

[69] M. Elhamod. (2011). BEHAVE Dataset: Frames 46200-47080 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/BEHAVE_46200-47080.avi

[70] M. Elhamod. (2011). BEHAVE Dataset: Frames 50375-54200 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/BEHAVE_50375-54200.avi

[71] M. Elhamod. (2011). BEHAVE Dataset: Frames 67210-76800 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/BEHAVE_67210-76800.avi

[72] M. Elhamod. (2011). PETS 2006 Dataset: S1C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S1C3.avi

[73] M. Elhamod. (2011). PETS 2006 Dataset: S2C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S2C3.avi

[74] M. Elhamod. (2011). PETS 2006 Dataset: S3C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S3C3.avi

[75] M. Elhamod. (2011). PETS 2006 Dataset: S4C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S4C3.avi

[76] M. Elhamod. (2011). PETS 2006 Dataset: S5C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S5C3.avi

[77] M. Elhamod. (2011). PETS 2006 Dataset: S6C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S6C3.avi

[78] M. Elhamod. (2011). PETS 2006 Dataset: S7C3 Demo. [Online]. Available: http://www.cim.mcgill.ca/~mndhamod/ThesisVideos/PETS2006_S7C3.avi

[79] A. Bayona, J. C. SanMiguel, and J. M. Martinez, "Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques," in *Proc. 6th IEEE Int. Conf. AVSS*, 2009, pp. 25–30.

Mohannad Elhamod (M'12) was born in Damascus, Syria, in 1984. He received the B.Sc. degree in electrical and computer engineering from Jordan University of Science and Technology, Irbid, Jordan, in 2007 and the M.Eng. degree in electrical engineering (field of computer vision) from McGill University, Montreal, QC, Canada, in 2012.

From 2007 to 2009, he was a Project Engineer in the field of networking and telecommunications with Syrian Data Systems, Damascus. From 2010 to 2012, he was a Teaching Assistant for a number of computer science and engineering courses with McGill University. He is currently a Software Integration Specialist with Miranda Technologies, Montreal. His research interests include computer vision and artificial intelligence.

Mr. Elhamod received the Provost's Graduate Fellowship and Principal's Graduate Fellowship from McGill University in 2010.

Martin D. Levine (S'59–M'66–SM'74–F'88–LF'04) received the B.Eng. and M.Eng. degrees in electrical and computer engineering from McGill University, Montreal, QC, Canada, in 1960 and 1963, respectively, and the Ph.D. degree in electrical engineering from Imperial College of Science and Technology, University of London, London, U.K., in 1965.

From 1979 to 1980, he was a Visiting Professor with the Department of Computer Science, Hebrew University, Jerusalem, Israel. From 1972 to 1973, he was a Member of the Technical Staff with the Image Processing Laboratory, Jet Propulsion Laboratory, Pasadena, CA. From 1986 to 1998, he served as the founding Director of the McGill Center for Intelligent Machines, McGill University. He is currently a Professor with the Department of Electrical and Computer Engineering, McGill University. He has worked as a Consultant for various government agencies and industrial organizations. He was a founding partner of AutoVu Technologies Inc. and VisionSphere Technologies Inc., for which he was the Chief Scientific Officer. He was also a member of the Scientific Board of ART Advanced Research Technologies Inc. He is the author of the book *Vision in Man and Machine* (McGraw-Hill College, 1985) and is a coauthor of *Computer Assisted Analyses of Cell Locomotion and Chemotaxis* (CRC, 1986). He was the Editor of the Plenum Book Series on Advances in Computer Vision and Machine Intelligence. His research interests include computer vision, image processing, and artificial intelligence.

Dr. Levine is a Fellow of the Canadian Academy of Engineering and the International Association for Pattern Recognition. He was also the founding President of the Canadian Image Processing and Pattern Recognition Society. He was the General Chair of the Seventh International Conference on Pattern Recognition held in Montreal during the summer of 1984. He was also elected as a Fellow of the Canadian Institute for Advanced Research (CIAR) in 1984. He served as the President of the International Association of Pattern Recognition from 1988 to 1990. He served as a CIAR/PRECARN Associate from 1990 to 1996. He is a member of the Editorial Board of the journal *Computer Vision and Understanding* and is responsible for face recognition. He has also served on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and *Pattern Recognition*. He received the 1997 Canadian Image Processing and Pattern Recognition Society Service Award for his outstanding contributions to research and education in computer vision.