Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, in conjunction with the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007

Detecting Luggage Related Behaviors Using a New Temporal Boost Algorithm*

Pedro Canotilho Ribeiro Plinio Moreno José Santos-Victor
Instituto Superior Técnico
Instituto de Sistemas e Robótica
Lisboa, Portugal
{pribeiro, plinio, jasv}@isr.ist.utl.pt

Abstract

In this paper we propose an approach to recognize luggage related behaviors in public spaces. We model behaviors in a multiclass learning framework, defining four classes: (i) walking, (ii) not moving, (iii) picking up/leaving bag, and (iv) abandoned bag.

We rely on the output of a tracking algorithm to generate targets in each image. Then, we analyze each target separately, by computing three features: (i) optic flow, (ii) motion energy, and (iii) bounding box area. The features are fed into a novel boosting algorithm that adds temporal consistency in a short-term way. This temporal boosting algorithm considers time explicitly in the weak classifiers, leading to an improvement in noise robustness and performance.

We show that our approach, with very simple features and a time-based boosting algorithm, is able to generate properly alarms on suspicious behaviors in a sequence of PETS 2007 database.

1 Introduction

The protection of critical transportation places and infrastructure is a very important topic these days. Many of these facilities exist in areas of high pedestrian traffic, making them accessible to attack, while not well suited for monitoring by humans, as it requires careful concentration over long periods of time.

Surveillance systems able to detect some potentially suspicious situations are crucial to people safety [3]. In this paper we present such a system, that is able to track pedestrians and detect their behaviors at critical transportation places (e.g. airports). Suspicious behaviors are viewed as low-level detections, using up to one second (25 frames) of video, and are essential to perform higher level reasoning about more complex situations over longer periods of time.

We propose an approach that models the behavior of people and luggage in an airport scenario. In every frame of the video we model the behavior of the targets provided by a tracking algorithm. The targets detected can be of three types: (i) people, (ii) luggage, and (iii) people with luggage. Using the evolution of target position along time, we define four behaviors: (i) person walking, (ii) person not moving, (iii) person picking up/leaving bag, and (iv) abandoned bag. To learn those behaviors, we propose a new multiclass learning algorithm with two main characteristics: (i) learns the optimal temporal window for classification, and (ii) shares features among classes and groups of classes.

We present a new learning algorithm that considers the temporal evolution of the features. Our proposal models temporal consistency in boosting, by parameterizing time in weak classifiers. As the basis of our work, we consider GentleBoost algorithm, using regression stumps as weak classifiers. A regression stump is similar to a single node binary tree, that selects a branch for a given feature according to a threshold using a binary decision function.

Alternatively, we propose to compute a new decision function, by averaging the decision value in the learned temporal window. This procedure transforms the binary decision into a fuzzy one, according to the temporal window size. This new weak classifier based on a fuzzy decision function provides two main advantageous properties in relation to other boosting algorithms: (i) improved noise robustness, and (ii) better performance.

The temporal boost algorithm is further extended to cope with multi-class problems, following the approach by Torralba et. al. in the JointBoosting [5]. We can thus (i) learn the parameters for all classes at once, and (ii) share features among classes and groups of classes.

Along with this temporal boosting algorithm, we pick a very simple and adequate set of features to model the

*This work has been partially supported by Fundação para a Ciência e Tecnologia (Proj. 61911 - GESTINTERACT; plurianual funds through the POS_Conhecimento Program that includes FEDER funds) and EU Project URUS (IST-045062). Pedro Ribeiro also acknowledges the PhD grant SFRH/BD/18936/2004

1. Given: $(x_1, y_1), \ldots, (x_N, y_N)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$, set $H(x_i) := 0$, initialize the observation weights $w_i = 1/N$, $i = 1, 2, \ldots, N$
2. Repeat for $m = 1, \ldots, M$
   (a) Find the optimal weak classifier h_m over (x_i, y_i, w_i).
   (b) Update strong classifier $H(x_i) := H(x_i) + h^*_m(x_i)$
   (c) Update weights for examples $i = 1, 2, \ldots, N$, $w_i := w_i e^{-y_i h^*_m(x_i)}$

Figure 1. GentleBoost algorithm.

human-luggage interaction: (i) optic flow, (ii) motion energy, and (iii) bounding box area.

We selected some of the PETS 2007 benchmark videos to train the parameters of the boosting algorithm. In the testing stage we use a different sequence to classify behaviors of the targets detected in each frame. The results suggest that the classification output can be used to generate alarms, for instance, related to (i) attended luggage removal, and (ii) unattended luggage.

2 Boosting with temporal consistent learners

In this paper we propose a new classifier that considers explicitly short-time consistency, based on boosting techniques. Boosting builds a strong classifier by summing several weak classifiers trained on various distributions over the training data, and became extremely popular in computer vision due to the ability to perform feature selection and classification.

2.1 GentleBoost algorithm

The Boosting algorithm provides a framework to sequentially fit additive models in order to build a final strong classifier, outputting the log-odd of the class given a feature point x_i. The boosting algorithm of Figure 1 uses adaptive Newton steps, resulting in minimizing, at each round, a weighted squared error

$$J = \sum_{i=1}^{N} w_i (y_i - h_m(x_i))^2, \tag{1}$$

where $w_i = e^{-y_i H(x_i)}$ are the weights and N the number of training samples. The optimal weak classifier is then added to the strong learner and the data weights adapted. Finally, the algorithm increases the weight of the misclassified samples and decreases correctly classified ones.

The choice of the weak learner h_m depends on the application, but in the case of GentleBoost it is common to use simple functions such as regression stumps. They have the form $h_m(x_i) = a\delta\left[x_i^f > \theta\right] + b\delta\left[x_i^f \leq \theta\right]$, where f is the number of the feature and δ is an indicator function (i.e. δ[*cond*] is one if *cond* is *true* and zero otherwise). Regression stumps can be viewed as decision trees with only one node, where the indicator function sharply chooses branch a or b depending on threshold θ and feature x_i^f.

To optimize the stump one must find the set of parameters $\{a, b, f, \theta\}$ that minimizes J w.r.t. h_m. A closed form for a and b are obtained and the value of pair $\{f, \theta\}$ is found using an exhaustive search [5].

Next section shows how to modify the regression stumps in the boosting procedure, including temporal consistency in the classification decision.

2.2 Weak learners with temporal consistency

GentleBoost algorithm allows to fit any weak learner function, h_m, in order to obtain a final strong classifier. In this work we add temporal consistency to the learner response using a set of T consecutive data points to perform the classification. The rationale is to use as much as possible of the information available in order to improve classifier output, instead of using a single point. Although the temporal window size is problem dependent, in most of the works the window size remains constant.

We propose to include the temporal window size T as an additional parameter of the regression stump. In this way the regression stump will only use advantageous information at each round, by choosing how many points to use depending on the feature values. Thus, consistency is included by defining the new Temporal Stumps as the mean classification output of the regression stump, in a temporal window of size T,

$$h^*_m(x_i) = \frac{1}{T} \sum_{t=0}^{T-1} \left(a\delta\left[x_{i-t}^f > \theta\right] + b\delta\left[x_{i-t}^f \leq \theta\right] \right). \tag{2}$$

The particular information extracted within the temporal window become clear if we put a and b in evidence,

$$h^*_m(x_i) = a\left(\frac{1}{T} \sum_{t=0}^{T-1} \delta\left[x_{i-t}^f > \theta\right]\right) + b\left(\frac{1}{T} \sum_{t=0}^{T-1} \delta\left[x_{i-t}^f \leq \theta\right]\right). \tag{3}$$

The new temporal weak learner of Eq. 3 can be viewed as the classic regression stump with a different "indicator function". If $T = 1$ it becomes the original regression stump, and for $T > 1$ the weak learner changes the indicator functions. The new indicator functions compute the percentage of points above the threshold θ

$$\Delta^T_+(f, \theta, T) = \frac{1}{T} \sum_{t}^{T-1} \delta\left[x_{i-t}^f > \theta\right], \tag{4}$$

and below the threshold θ

$$\Delta^T_-(f, \theta, T) = \frac{1}{T} \sum_{t}^{T-1} \delta\left[x_{i-t}^f \leq \theta\right], \tag{5}$$

in the considered temporal window T and for the feature number f. The indicator functions with temporal consistency in Eqs. 4/5, can take any value in the interval $[0\ 1]$, depending on the length of the temporal window used. For example, if $T = 2$ the functions can take 3 different values, $\Delta_{+}^{T} \in \{0,\ 1/2,\ 1\}$, if $T = 3$ can take four values, $\Delta_{+}^{T} \in \{0,\ 1/3,\ 2/3,\ 1\}$ and so on.

The fuzzy output of the new "indicator function" ,Δ, represents the confidence of threshold choice to use the data with temporal support T. Thus, at each boosting round, we use a weighted confidence of both branches, instead of choosing only one branch. We will show that optimizing this fuzzy regression stump brings additional resistance to noise, thus increasing the generalization capabilities.

During classification of unseen data, the algorithm has the possibility to decrease the confidence measure, Δ, for instance if the new data is noisy when compared to the training data. This differs from the usual boosting binary decision and can be compared to what fuzzy trees brought to decision trees.

Replacing the weak learner with temporal consistency of Eq. 3, in the cost function of Eq. 1, we compute the optimal temporal stump parameters a and b,

$$a = \frac{\bar{\nu}_{+}\bar{\omega}_{-} - \bar{\nu}_{-}\bar{\omega}_{\pm}}{\bar{\omega}_{+}\bar{\omega}_{-} - (\bar{\omega}_{\pm})^2},\ b = \frac{\bar{\nu}_{-}\bar{\omega}_{+} - \bar{\nu}_{+}\bar{\omega}_{\pm}}{\bar{\omega}_{+}\bar{\omega}_{-} - (\bar{\omega}_{\pm})^2}, \tag{6}$$

with

$$\bar{\nu}_{+} = \textstyle\sum_{i}^{N} w_i y_i \Delta_{+}^{T}, \quad \bar{\nu}_{-} = \sum_{i}^{N} w_i y_i \Delta_{-}^{T},$$
$$\bar{\omega}_{+} = \textstyle\sum_{i}^{N} w_i \Delta_{+}^{T}, \quad \bar{\omega}_{-} = \sum_{i}^{N} w_i \Delta_{-}^{T},$$

and

$$\bar{\omega}_{\pm} = \textstyle\sum_{i}^{N} w_i \Delta_{-}^{T} \Delta_{+}^{T}.$$

Note that all the above variables are functions of $\{f, \theta, T\}$ that we dropped for notation simplicity. To find the optimal f,θ and T we use exhaustive search. Comparing the temporal weak learner with the original GentleBoost weak learner, we have an additional parameter T to optimize. The algorithm is summarized in figure 1 similarly to GentleBoost, but optimizing the new temporal stump, h_m^*.

2.3 Going to the multi class problem

A multi-class categorization problem is usually solved as a set of multi-binary problems where separate classifiers are trained and applied independently. As pointed by Torralba *et al* [5], for the object detection problem, this is a waste of resources because many of the features used can be shared among several classes.

There are some datasets where computational complexity issues and amount of training data are even more critical; for instance, when working with video (temporal data) instead of single frames. In the case of boosting, sharing helps to: i) reduce computational complexity by sharing weak classifiers among classes and ii) reduce the amount of training data needed in order to attain the same classification performance.

In the same manner as Torralba *et al* extended GentleBoost to multi-class problems, we generalize temporal GentleBoost to the multi-class problem.

2.3.1 The Temporal-JointBoosting algorithm

The idea behind JointBoosting [5] is to share weak classifiers (and features) across classes. At each round the algorithm chooses a weak classifier that share a feature among the subset of classes. The optimal subset of classes is chosen by minimizing the error cost function for all possible combinations.

The optimization to be solved has now one more variable, the classes, thus one must solve the following weighted least squares problem in each iteration:

$$J = \sum_{c=1}^{C} \sum_{i=1}^{N} w_i^c (y_i^c - h_m(c, x_i))^2, \tag{7}$$

where $w_i^c = e^{-y_i^c h_m(c,x_i)}$ are the new class-specific weights. Note that each training sample has now C weights, one for each class, and the weights are a probability distribution only over the $C \times N$ space.

Shared stumps uses the data that belongs to the optimal subset of classes as positive samples, and remaining data as negative samples. In the case of classes that belongs to the optimal subset $S(n)$, the stump function is similar to the binary case. For classes outside the optimal subset, the stump function is a class-specific constant, k^c. This ensures that adding the class to the subset improves the classification more than just using a constant classifier (see [5] for details). The shared temporal stump has the following form:

$$h_m(c, x) = \begin{cases} a_S \Delta_{+}^{T} + b_S \Delta_{-}^{T} & \text{if } c \in S(n) \\ k_S^c & \text{if } c \notin S(n), \end{cases} \tag{8}$$

where $\Delta_{+/-}^{T}$ are the temporal consistency function defined in Eqs. 4 and 5. The minimization of the cost function in Eq. 7, using the shared temporal stumps at each iteration provides the optimal parameters,

$$a_S = \frac{\sum_{c\in S(n)} \bar{\nu}_{+}^c \sum_{c\in S(n)} \bar{\omega}_{-}^c - \sum_{c\in S(n)} \bar{\nu}_{-}^c \sum_{c\in S(n)} \bar{\omega}_{\pm}^c}{\sum_{c\in S(n)} \bar{\omega}_{+}^c \sum_{c\in S(n)} \bar{\omega}_{-}^c - \left(\sum_{c\in S(n)} \bar{\omega}_{\pm}^c\right)^2}, \tag{9}$$

$$b_S = \frac{\sum_{c\in S(n)} \bar{\nu}_{-}^c \sum_{c\in S(n)} \bar{\omega}_{+}^c - \sum_{c\in S(n)} \bar{\nu}_{+}^c \sum_{c\in S(n)} \bar{\omega}_{\pm}^c}{\sum_{c\in S(n)} \bar{\omega}_{+}^c \sum_{c\in S(n)} \bar{\omega}_{-}^c - \left(\sum_{c\in S(n)} \bar{\omega}_{\pm}^c\right)^2}, \tag{10}$$

and

$$k^c = \frac{\sum_i w_i^c y_i^c}{\sum_i w_i^c},\ \ c \notin S(n). \tag{11}$$

To obtain the optimal θ, f, T and class subset, $S(n)$, exhaustive search is performed. The final algorithm is described in figure 2.

1. Given: $(x_1, y_1), \dots, (x_N, y_N)$ where $x_i \in X$, $y_i^c \in Y = \{-1, +1\}$, set $H(c, x_i) := 0$, initialize the observation weights $w_i^c = 1/(N \times C), i = 1, 2, \dots, N, c = 1, 2, \dots, C$
2. Repeat for $m = 1, 2, \dots, M$
   (a) Repeat for $n = 1, 2, \dots, 2^C - 1$
      i. Find the optimal joint temporal weak classifier:
      $$h_m^n(c, x) = \begin{cases} a_S \Delta_+^T + b_S \Delta_-^T & \text{if } c \in S(n) \\ k_S^c & \text{if } c \notin S(n) \end{cases}$$
      ii. Evaluate the error:
      $$J = \sum_{c=1}^{C} \sum_{i=1}^{N} w_i^c (y_i^c - h_m(c, x_i))^2$$
   (b) Find the subset of classes with less error, n^*.
   (c) Update strong classifier
   $$H(c, x_i) := H(c, x_i) + h_m^{n^*}(c, x_i)$$
   (d) Update weights for examples $i = 1, 2, \dots, N$, $w_i^c = e^{-y_i^c h_m^{n^*}(c, x_i)}$

Figure 2. Temporal-JointBoost algorithm.

2.3.2 Efficient computation of temporal shared stumps

As mentioned in the previous section, the optimal subset of classes is provided by exhaustive search. This becomes a slow computation since it involves scanning over all features and all N thresholds (where N is, in general, the number of training samples) and all temporal windows T. However, if one look carefully to this process, it can be easily verified that the computations used for the subsets with only one class (each $c \in C$) can be reused for all the remaining subsets (with more than one class).

At each boosting round the process is the following:

1) For the one class subsets: Compute the parameters a_c and b_c so as to minimize the weighted square error:

$$a_c(f, \theta, T) = \frac{\bar{\nu}_+^c \bar{\omega}_-^c - \bar{\nu}_-^c \bar{\omega}_\pm^c}{\bar{\omega}_+^c \bar{\omega}_-^c - \left(\bar{\omega}_\pm^c\right)^2}, b_c(f, \theta, T) = \frac{\bar{\nu}_-^c \bar{\omega}_+^c - \bar{\nu}_+^c \bar{\omega}_\pm^c}{\bar{\omega}_+^c \bar{\omega}_-^c - \left(\bar{\omega}_\pm^c\right)^2}. \quad (12)$$

Keep each one of the terms $\{\bar{\nu}_+^c, \bar{\nu}_-^c, \bar{\omega}_+^c, \bar{\omega}_-^c, \bar{\omega}_\pm^c\}$, and compute $\bar{\omega}^c = \sum_i^N w^c$ and $\bar{\gamma}^c = \sum_i^N w^c y_i^c$.

2) For the remaining subsets: The optimal a_S and b_S can be computed as a combination of each one of the a_c and b_c terms that belong to the subset S using equations 9 and 10.

2) Weighted least square error: The weighted regression error, for each triplet $\{f, \theta, T\}$ and for the set of classes $S(n)$, can also be computed from the previous equations using:

$$J = \sum_c \bar{\omega}^c + a_s^2 \sum_{c \in S(n)} \bar{\omega}_+^c + b_s^2 \sum_{c \in S(n)} \bar{\omega}_-^c$$
$$+ 2a_s b_s \sum_{c \in S(n)} \bar{\omega}_\pm^c - 2a_s \sum_{c \in S(n)} \bar{\nu}_+^c - 2b_s \sum_{c \in S(n)} \bar{\nu}_-^c$$
$$+ \sum_{c \notin S(n)} k^{c2} \bar{\omega}^c - 2 \sum_{c \notin S(n)} k^c \bar{\gamma}^c.$$

2.4 Results on Synthetic data

We perform tests with synthetic data in order to illustrate the advantages of our algorithm over other boosting approaches: (i) improved noise robustness, and (ii) improved performance in data with large overlapping in the feature space. The aim is to learn five elliptic trajectories using point location as features, and then classify new points as belonging to one of the trajectories. We create synthetic data and apply three boosting algorithms: (i) GentleBoost [2], (ii) TemporalBoost [4] and (iii) our optimal temporal boosting. The recently proposed TemporalBoost [4] also introduced temporal consistency in a boosting procedure, by averaging previous AdaBoost weak classifiers sequentially, while the classification error decreases. However, temporal support is considered in a heuristic procedure only after training the weak classifiers, and the TemporalBoost averaged output is mapped to a binary value.

The data for the learning algorithms comprise five elliptical trajectories were generated for 10 levels of overlapping between classes, each one corresponding to 5 class classification problem . The input features (x_i^1, x_i^2) are noisy observations of the actual ellipses, generated accordingly to: $x_i^1 = a \cos t + \mathcal{N}(0, \sigma)$ and $x_i^2 = b \sin t + \mathcal{N}(0, \sigma)$, where a and b are the major and minor axis, t represents time, and $\mathcal{N}(\mu, \sigma)$ is Gaussian noise. We vary the problem complexity, starting from the simplest case (easily separable, e.g. at the left side of Figure 3), and increasing the proximity among classes towards a more complex problem (e.g. at the right side of Figure 3). In Figure 3 we see that our temporal version of JointBoost performing almost 20% better than TemporalBoost, and 30% better than the original JointBoost. The following step is to select adequate features for detecting behaviors included in the PETS 2007 dataset.

3 Behaviors and Feature computation

In human environments with surveillance cameras, it can be defined a wide range of possible human behaviors. In this paper we focus on behaviors present in environments like transport networks, town centers and public facilities such as schools, hospitals and sport grounds. We consider four activities potentially interesting to identify suspicious behaviors: *1 - walking, 2 - not moving, 3 - picking up or leaving bag* and *4 - abandoned bag*. The idea is to detect

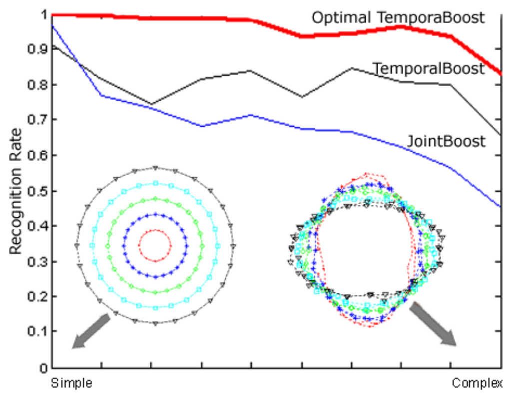

Figure 3. Recognition rate in multi-class synthetic datasets, using 10 levels of increasing overlap among classes

these behaviors from less than one second of input video and using information only from the detected target optical flow (displacement between frames of the detected pixels) and target size.

From the optical flow measurements we compute two features: i) the vertical component of the mean optical flow and ii) the motion energy (averaged over 25 frames). To model the target size we use the detected bounding box area. These three features are fed into the optimal Temporal-JointBoost that will jointly train all the four classes, choosing the optimal temporal window to use.

In the next section we will show the results on the PETS 2007 dataset, recorded in an airport hall.

4 Human activity recognition

To demonstrate the performance of the proposed algorithm for detecting the four behaviors we used the PETS 2007 datasets. To perform the tests we selected three sequences and used only the third view in each one: *left luggage 1* and *theft 3* for training and *left luggage 2* for testing.

For each sequence we track the interesting targets using the LOTS tracker [1] and then compute the features in the targets bounding box. To train the algorithm we generate a label for all the targets with the desired behaviors. Training features of four targets were computed in the sequences *left luggage 1* and *theft 3*, using a total of 814 frames. The testing stage was done in 1340 frames, from four targets present in the sequence *left luggage 2*.

Figure 4 illustrates the test scenario, *left luggage 2*, and the tracker output for the frames 1785, 1870 and 1920. Tracker output along frames was used to compute the features and, jointly with the label, fed them to the optimal Temporal-JointBoost algorithm.

(a)

(b)

(c)

Figure 4. Example images of one scenario from PETS 2007 dataset used to test the algorithm. The images represent frames 1785 4(a), 1870 4(b) and 1920 4(c), with the detected bounding boxes and correspond to the *left luggage 2* scenario.

Figure 5 shows sample images and the corresponding classification results (*1 - walking*, *2 - not moving*, *3 - picking up or leaving bag* and *4 - abandoned bag*). Figure 5(a) corresponds to an individual that enters into the scene carrying a large bag which is then placed on the ground. Example 5(b) corresponds to an individual that picks up an abandoned bag walking away with it. In the last example 5(c) and individual enters into the scene talking on his cellular phone, either walking or not moving.

From the examples plotted in Figure 5, we see that our algorithm is capable of generating meaningful alarms. In these images only three frames were misclassified. For the remaining frames of the *left luggage 2* sequence we found a similar behavior that is shown in Figure 6. Most of the frames misclassified correspond to an ambiguous and faltering *walking*, while the most critical behaviors, *picking up or leaving bag* and *abandoned bag*, are classified correctly in most of the frames, having a delay of less than a second.

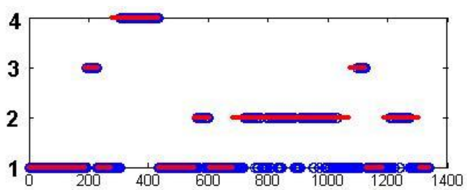

Figure 6. Classification results of four targets tracked and extracted from sequence *left luggage 2* **(PETS 2007 database). The numbers on the left correspond to classes:** *1 - walking*, *2 - not moving*, *3 - picking up or leaving bag* **and** *4 - abandoned bag*. **The results correspond to 1340 frames and are plotted on blue behind a red, hand generated, ground truth. Most of the misclassified frames result from an ambiguous and faltering** *walking* **(from frame 500 to frame 1100), for simplicity labeled as** *not moving*.

5 Conclusions

In this paper we have proposed a methodology to recognize lugagge related behaviors. Our proposal has two main components: (i) a novel learning algorithm, and (ii) a very simple set of features.

The learning algorithm is able to handle with temporal data consistency in non-sequential single-frame classification problems. Although the temporal evolution of the data might be crucial for certain classification problems (e.g. human activity recognition), even when performing single-frame classification, it is rarely introduced at the level of the classifier training. We extend the GentleBoost algorithm by modeling time as an explicit parameter to optimize. As a consequence of this time consistency model, we obtain a fuzzy decision function, depending on the temporal support of the data, which brings additional robustness to noise. Additionally, we allow our temporal boosting algorithm to cope with multi class problems, following the JointBoosting approach by Torralba et. al [5].

We consider four target activities related to the suspicious behaviors: *walking*, *not moving*, *picking up or leaving bag* and *abandoned bag*. We choose three features computed in target's bounding box to distinguish between these behaviors: (i) the vertical component of the mean optic flow, (ii) motion energy, and (iii) bounding box area. The results suggest the suitability of our approach for detection of suspicious activities. We obtain very good results taking into account that we perform single-frame classification without modeling the activities dynamics.

As future work we aim to include spatially dependent higher level models, like conditional random fields, that will model the activities dynamics and will take into account temporal dependencies at various levels. The sequential models should act as contextual restrictions during the transition between activities that add long-term temporal consistency to the recognition, rather than the short-term consistency presented here.

In future applications, those systems with high level models could alert authorities if a pedestrian displays suspicious behavior, e. g. entering a secure area, dropping a bag or loitering.

References

[1] T. Boult, R. Micheals, X. Gao, and M. Eckmann. Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. *Proceedings of the IEEE*, 89(10), October 2001. 5

[2] T. H. J. Friedman and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics*, 28(2):337407, 2000. 4

[3] N. P. R. Bodor, B. Jackson. Vision-based human tracking and activity recognition. *Proc. of the 11th Mediterranean Conf. on Control and Automation*, June 2003. 1

[4] P. Smith, N. da Vitoria Lobo, and M. Shah. Temporalboost for event recognition. In *International Conference on Computer Vision*, volume 1, pages 733– 740, October 2005. 4

[5] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transacions On Pattern Analysis and Machine Intelligence,In press*, 2007. 1, 2, 3, 6

(a)

(b)

(c)

Figure 5. Sample classification results for the test sequence *left luggage 2* **correspond to:** *1 - walking*, *2 - not moving*, *3 - pick up or left bag* **and** *4 - abandoned bag*. **In 5(a) an individual enters into the scene carrying a large bag which is then placed on the ground. It corresponds to the frames: 429, 465, 519, 771, 814, 844, 874, 889 and 906. In all plotted frames the activity was correctly classified,** *walking* **in the first two,** *not moving* **in the following three,** *walking* **again for two frames and finally** *pick up or left bag* **in the rightmost two. Example 5(b) corresponds to an individual that picks up an abandoned bag walking away with it. Images correspond to frames: 705, 723, 795, 849, 889, 907, 925, 943 and 961. In this case the two images on the left are misclassified as** *walking*, **the following two being correctly classified as** *abandoned bag*. **The remaining frames are correctly classified as** *walking* **followed by** *pick up or left bag* **and then** *walking* **again. In the last example 5(c) and individual enters into the scene talking on his cellular phone, either walking or not moving. The frames presented are: 717, 735, 753, 771, 789, 807, 825, 843 and 861. In all plotted frames the activity was correctly classified.**