A Localized Approach to Abandoned Luggage Detection with Foreground-Mask Sampling

Huei-Hung Liao, Jing-Ying Chang, and Liang-Gee Chen

DSP/IC Design Lab.,
Graduate Institute of Electronics Engineering and Department of Electrical Engineering,
National Taiwan University
b93901125@ntu.edu.tw, {jychang, lgchen}@video.ee.ntu.edu.tw

Abstract

In this paper we propose a novel approach to the detection of abandoned luggage in video surveillance. Candidates of abandoned luggage items which may pose potential security threats are first identified and localized by our proposed foreground-mask sampling technique. Our approach can deal with luggage pieces of arbitrary shape and color without the need for prior learning, and it works well under crowded and highly-cluttered situations. This localization of suspicious luggage items in the scene enables us to focus attention and subsequent processing solely on their neighborhoods. The owner of the luggage is then located and tracked to determine whether or not the luggage has been abandoned deliberately. A probability model using the MAP principle is employed to calculate a posteriori confidence score for the luggage-abandonment event, and an alarm will be automatically triggered if the certainty of luggage abandonment is higher than a pre-defined threshold. We show our results on the video datasets provided by the 2007 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007) and the 2006 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006).

1. Introduction

Intelligent and automated security surveillance systems have become an active research area in recent time due to an ever-increasing demand for such systems in public domains. The ability to reliably detect suspicious items and identify their owners in a scene is of urgent need in a large number of places such as airports and train stations. In this work, we propose a novel approach that can automatically localize suspicious and possibly abandoned luggage items within camera view and track their owner(s). The system is capable of locating an abandoned luggage item even in a highly cluttered scene, for example in a crowded subway station; while at the same time it automatically ignores other irrelevant moving objects in the foreground. A probabilistic framework is then employed to calculate a posterior confidence for such a luggage-abandonment event based on tracking information. An alarm will be triggered if the luggage is declared to be abandoned with sufficiently high confidence.

We follow three similar but slightly different rules to those used in [3] in our definition of the luggage-abandonment event: (1) Contextual rule: A luggage is owned and attended by a person who enters the scene with it until such point that the luggage and the person are no longer in close proximity. (2) Spatial rule: A luggage is unattended when its owner is outside a small neighborhood around the luggage. (3) Temporal rule: If the owner of a luggage leaves the scene without the luggage, or if a luggage has been left unattended by the owner for a period of more than 30 consecutive seconds, in which time the owner has not re-attended to the luggage, the luggage is declared to be abandoned and an alarm should be triggered.

The task of abandoned luggage detection in surveillance video can generally be split into three stages: The first stage takes each video frame and localizes candidates of abandoned luggage items. The second stage then locates the luggage owner(s) and performs tracking on them, providing a trajectory for later probabilistic reasoning. The final stage evaluates a probability, or a confidence score, for the luggage-abandonment event based on information gathered in previous stages.

1.1. Related work

The three stages outlined above are all distinct research areas with rich literature of their own. Various existing algorithms may employ different methods for different stages.

The first stage of locating candidates of abandoned luggage items in the frame can generally be divided into two categories: Those that utilize the technique of

978-0-7695-3341-4/08 $25.00 © 2008 IEEE
DOI 10.1109/AVSS.2008.9

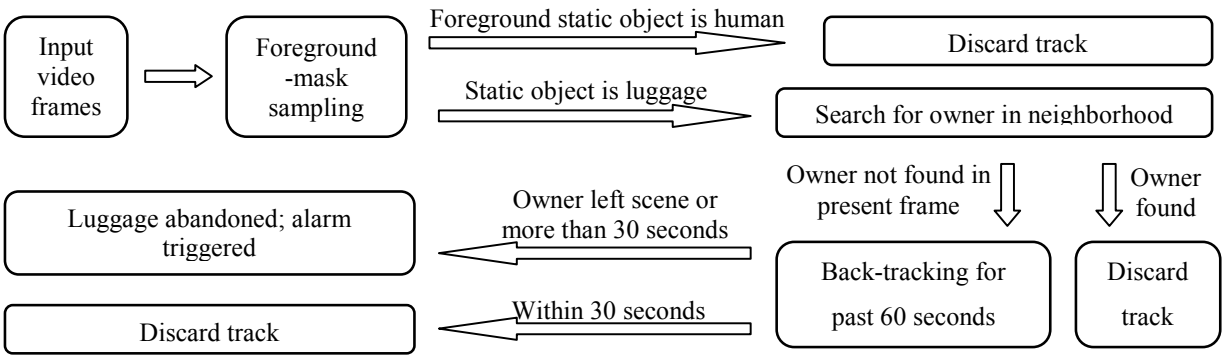

Figure 1. Flow chart of our proposed approach

background subtraction [6], [7] and [8], and those that don't [1] and [2]. Background subtraction works reasonably well when the camera is stationary and the change in ambient lighting is gradual, if at all. For those that do without background subtraction, a set of discriminative features of the objects of interest has to be learned beforehand through machine learning algorithms in order to be able to detect these objects in subsequent stages.

The majority of existing event detection methods incorporates tracking algorithm of some form in their system, as in [2], [3], [6] and [7]. In most cases tracking is performed on all detected moving objects or blobs in the foreground. However, due to occlusion and a fixed camera angle, this kind of comprehensive tracking often results in errors such as identity switch (when two nearing objects switch their identities), which is difficult to avoid and can be seen in many PETS 2006 demonstration sequences such as those in [5].

In most cases surveyed, the final stage of determining whether an alarm should be issued is done in a deterministic fashion. In a deterministic system an event is declared to have occurred if some criteria are satisfied. A minority employs a probabilistic framework [3] to model events, in which case an event is deemed to have occurred if its confidence score exceeds a certain threshold. The probabilistic approach gives users more flexibility to set thresholds and hence system sensitivity, as well as a better understanding of how *real* the situation might actually be.

1.2. Outline of our approach

Our proposed approach employs a novel technique, which we shall refer to as foreground-mask sampling, to localize the candidates of abandoned luggage items in the scene. As the first stage of our system, the foreground-mask sampling technique computes the intersection of a number of background-subtracted frames which are sampled over a period of time. Abandoned luggage items are assumed to be static foreground objects and therefore will show up in this intersection. Since our approach requires no prior learning of luggage appearance in any form, we can successfully localize luggage of all shapes, sizes, orientations, viewing angles and colors with no need for and no constraints from any training data. Once a suspicious luggage item is identified and localized, our algorithm attempts to search for its owner within a neighborhood around the detected luggage. If the owner is found within this neighborhood, the luggage is assumed to be attended by its owner and no further processing is required.

However, if no owner is found in proximity to the luggage in this present frame at time t, our tracking algorithm then goes back in time (for a pre-defined length of Δt seconds) to the frame at time $t - \Delta t$ when the owner was still attending the luggage, and it starts tracking the owner from then. The tracking algorithm utilizes motion prediction in conjunction with (1) skin color information and (2) an improved version of generalized Hough Transform on human body contour as feature. Rather than comprehensively tracking all moving foreground objects, which is normally done in most existing event detection systems, we track *only* the owner of the suspicious luggage item which has been localized in the first stage; other irrelevant moving objects in the foreground are simply ignored. We call our method selective tracking, as opposed to conventional comprehensive tracking.

The tracking module provides a trajectory of the luggage owner from frame $t - \Delta t$ to frame t, and this information is used for probabilistic reasoning in the third stage. For the luggage to become abandoned, its owner has to leave the scene without it, or it has to remain unattended for at least 30 consecutive seconds. A probability score will be given by the tracking algorithm to represent the reliability of the owner's tracked trajectory, and this probability score is used in the subsequent evaluation of the overall confidence score of the luggage-abandonment event. The event detection is formulated as a Maximum A Posteriori (MAP) problem. Finally an alarm will be

triggered if the overall confidence score of the luggage-abandonment event is above a given threshold, which is adjustable by the user to achieve varying levels of system sensitivity. See Figure. 1 for system work flow.

2. Foreground-mask sampling

In the first stage of our system, we attempt to localize static and possibly abandoned luggage items in camera view. This is done by what we call foreground-mask sampling.

As is generally acknowledged, object detection and recognition is an instinctive and spontaneous process for human visual system; however, implementing a robust and accurate computer vision system capable of detecting relevant objects in the surroundings has proved rather challenging. The main difficulty lies in the fact that the appearance of an object can undergo significant variation due to viewpoint changes, scene clutter, ambient lighting changes, and in some cases even changes in shape (for non-rigid objects such as human body). As a result, the same object may give widely different images under various viewing conditions.

Our proposed foreground-mask sampling technique attempts to emulate the natural human ability to direct attention *only* to the objects of interest to us, whatever shape or viewing angle it might have. We propose an elegant algorithm that identifies the objects (in this case luggage items abandoned) that we are looking for by logical foreground-background reasoning, while ignoring all other irrelevant objects in the scene. We do not use appearance-based model in locating suspicious luggage items, so our proposed method can deal with luggage of any color and shape and does not suffer from different viewing angles.

Since it is assumed that luggage abandoned on the scene shall remain static for a period of time, we collect a number of sample video frames from the past 30 seconds, as the temporal rule dictates; the number of frames is empirically chosen to be 6 in this case, evenly distributed among the 30 seconds sampled. In our experiment, changing the number of sample frames does not significantly alter detection performance. Background subtraction is then performed on these 6 sample frames to produce 6 corresponding foreground images. Specifically, let F_1 to F_6 be the 6 sample frames, B be the background image and Std be the standard deviation image, with (i, j) denoting pixel position in the image, we state that

F_k(i, j) is a foreground pixel if and only if $\left| F_k(i, j) - B(i, j) \right| > w(i, j) * Std(i, j)$

where k = 1 to 6 and w (i, j) is a weight on the standard deviation at pixel (i, j), which is smaller in value when i is small (upper part of the image) and larger when i is large (lower part). The weight w (i, j) is implemented as a function of image row i to take into account the gradual change in image resolution in the row-wise, vertical direction. This gradual variation in image resolution is caused by a tilting camera angle looking down the scene from top, which is a shared characteristic among a majority of surveillance cameras. Images produced by these cameras have higher resolution in the lower part where objects appear larger and the camera is closer to the scene; but as the objects move away from the camera they move upward in the image and become smaller, resulting in lower resolution and decreased image quality, see Figure 2. The use of the w (i, j) weighting raises the foreground threshold for lower part of the image where resolution is better and lowers that where resolution is not as good, compensating for the resolution change caused by a tilting camera angle. Our modification here using the variable weight w (i, j) works reasonably well in the absence of specific camera parameters. Although this may lack the precision offered by a meticulous calibration using camera parameters (when available), for our purpose here (which is simply to tell foreground from background) it should be an adequate substitute.

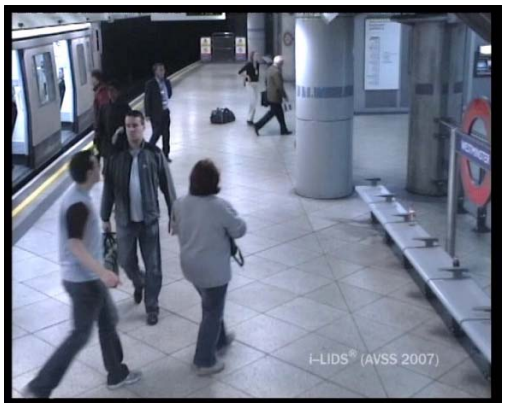

Figure 2.Image captured by a typical surveillance camera looking down, where objects appear larger in lower part and smaller in upper part. AVSS 2007 video dataset.

The 6 foreground images thus obtained are binarized to produce 6 foreground masks, where a foreground pixel is 1 in value and a background pixel 0; let M_k(i, j) be the 6 foreground masks, k = 1 to 6, and we state that

M_k(i, j) is 1 if and only if F_k(i, j) is a foreground pixel; otherwise, M_k(i, j) is 0 (background).

These 6 foreground masks are then merged and the intersection of them is taken as the static foreground object mask S, as

$$S = M_1 .* M_2 .* M_3 .* M_4 .* M_5 .* M_6$$

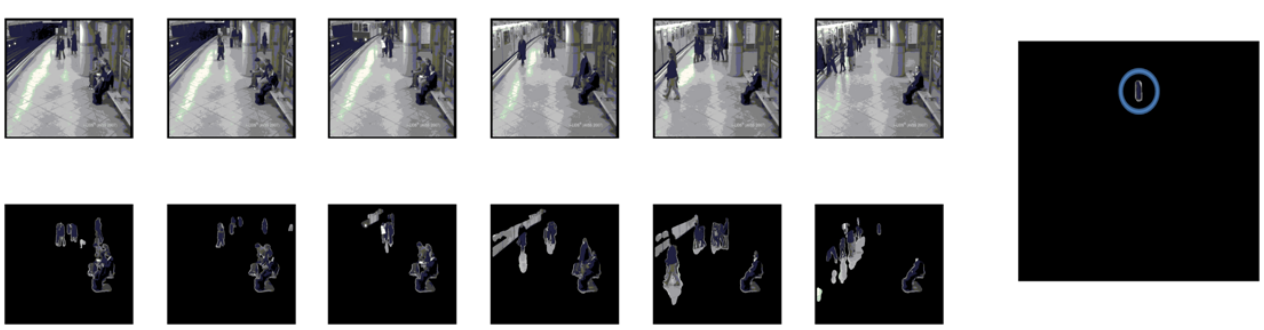

Figure 3. Foreground-mask sampling. The first row shows input frames; the second row shows corresponding foreground images. Image on right is the intersection of the 6 foreground images sampled over a period of 30 seconds, which contains the abandoned luggage item. AVSS 2007 video dataset.

Binarization allows the intersection to be taken through simple point-wise multiplication of the 6 foreground masks, as indicated by the operator '.*'. Filtering operation is then carried out on this static foreground object mask S to remove irrelevant and sporadic noisy pixels, and connected component analysis is performed afterwards. A white (valued 1) block on the static foreground object mask S indicates a region that has remained foreground in all of the 6 sample frames over the past 30-second period, and therefore this region should very likely correspond to either a static abandoned luggage item or a non-moving human being. Our tracking module, which is to be detailed in the next section, will then analyze the region and further localize it if it is determined to be a static luggage item, see Figure 3.

The static foreground object mask S thus obtained by the foreground-mask sampling gives possible candidates for abandoned luggage items. The approach is elegant and robust in that it manipulates directly such low-level features as foreground and background image pixels. This provides us with a localized target and allows us to focus on a localized search region for later tracking and higher-level event reasoning.

3. Selective tracking module

With the static foreground object mask S obtained, our system has localized information on where the suspicious objects are in the scene. It should also be pointed out that here we assume all static *foreground* objects to be either human or luggage item. For each white region (valued 1) in the static foreground object mask S, our algorithm checks if it is a human or luggage (i.e. not human) by a combination of skin color information and human body contour that shall be explained in detail shortly. If the region is determined to be a human, it is discarded because what we are looking for is abandoned luggage items. If it is a luggage item, a local search region is constructed around the detected luggage's neighborhood to see if its owner is in close proximity in this present frame at time t. If the owner is found, the region is again discarded because the owner exhibits no present intention of abandoning the luggage. If, however, the owner is not found around the luggage in the present frame, our algorithm then goes back in time for a pre-defined Δt seconds, 60 in this case, to the frame at time $t - \Delta t$ when the owner was still attending the luggage and starts tracking the owner from here (at time $t - \Delta t$). The tracking algorithm again employs skin color information and human body contour as features.

Because suspicious luggage has already been localized by foreground-mask sampling in the first place, we are able to perform tracking solely and selectively on the owner of this static luggage item. This mechanism closely mimics the human ability to notice and track only the object that is of interest to us even under a highly cluttered background, for example humans' natural ability to identify familiar faces in such crowded space as an airport pick-up area. Our ability to track only the object we are interested in also reduces the risk of identity switch that is difficult to avoid if tracking is performed on a comprehensive, full-frame scale.

The details on the implementation of detection and tracking using skin color information and human body contour are described below, as well as their integration into the motion prediction part of the tracking module.

3.1. Cr color channel with human skin

Human skin signal response is significantly larger in the YCbCr color space than in the commonly used RGB color space. Due to a large amount of blood flow, human skin gives high response to the Cr channel in YCbCr space, irrespective of skin color or race [4]. We propose to utilize skin color as given by the Cr channel for human face localization because in situations of severe occlusion (crowded scenes with people overlapping one another), human face is the most visible body part under a typical surveillance camera with a tilting angle looking down from top.

A search region is first constructed around the

Figure 4. Left: input video frame with localized search region indicated by red circle. Right: the Cr detection result within the search region. AVSS 2007 video dataset.

neighborhood of the suspicious static luggage item detected by our foreground-mask sampling technique. Background subtraction is then performed on all three R, G and B channels within the search region, using three sets of backgrounds and standard deviation images for the three color channels. An RGB foreground of the search region is obtained and then converted to the YCbCr color space, and the Cr channel is retained, see Figure 4. Conversion from RGB to YCbCr is straightforward through matrix multiplication. The reason we perform background subtraction within the search region before conversion to YCbCr color space is that since Cr is a color difference channel in red, the face signal is stronger with background clutter removed. The Cr channel response is then used to locate the luggage owner's face, in conjunction with human body contour information that is explained in the next section.

3.2. Improved Hough Transform on body contour

Cr channel responds to the color of red within the search region, which in some cases may include other reddish objects in addition to the owner's face. For this reason, we impose a second mechanism to reliably detect the presence of the luggage owner. This second feature used is the human upper-body contour which consists of the head-shoulder silhouette. The head-shoulder contour, as inspired by [1], is used under Hough Transform to detect the presence of human upper-body within the search region. The contour is given in Figure 5.

Hough Transform (HT) is a morphological tool which, in its simplest form, maps a straight line in normal space to a point in parameter space. We employ in our work a more sophisticated, generalized version of the Hough Transform that is capable of localizing contour of arbitrary shape. The whole algorithm consists of two stages: (1) Template generation and (2) Contour matching.

In the template generation stage, as in Figure 5, the HT algorithm first establishes a center point (x_C, y_C) for the contour template, which we assign to the face center in this case. A reference table of 180 bins is created. The algorithm then runs through all edge points (x, y) on the contour template and records on each point its ψ (angle with respect to the horizontal direction), r (distance with respect to the center point) and α (angle with respect to the center point). The ψ lies between 0 and 180 degrees and thus serves as the bin-index with which the (r, α) pair is recorded into the reference table. Multiple pairs of (r, α) may be recorded under the same ψ-angle bin in the reference table. When all the points on the contour template have been traversed, the template generation is finished and the reference table is complete. In mathematical form, the (r, α) pair is computed using the following relationships

$$r = [\,(x-x_C)^2 + (y-y_C)^2\,]^{(1/2)} \quad (1)$$

$$\alpha = \tan^{-1}[\,(y-y_C) / (x-x_C)\,]. \quad (2)$$

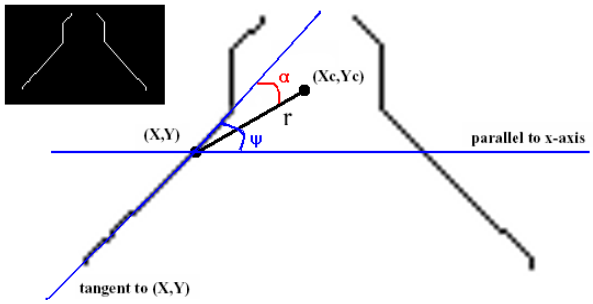

Figure 5. Head-shoulder contour under Hough Transform. In template generation, relative position of (X_C,Y_C) to (X,Y) is recorded in (r,α); in contour matching, the pixel on the assumed position of (X_C, Y_C) is incremented by 1 on the detection map, which is calculated from the start point (X,Y) by extending a distance of r and an angle of α. On the upper-left corner is the actual contour used.

In the next stage, contour matching is performed on the edge image of the input video frame. We use a 3x3 sobel kernel in obtaining our input edge image. First a detection map of the same size as input video frame is created. Initially it contains all zeros. The Hough Transform algorithm again travels all edge points on the input edge image, calculating the ψ angle of each edge point. For an edge point E (x, y) on the input edge image with ψ angle of m degrees, all (r, α) pairs under that specific ψ-angle bin, which is the $(m+1)^{th}$ bin, will be accessed. And for each of the (r, α) pairs under this bin, we use E (x, y) as the start point and calculate a coordinate pair as

$$(x_{C'}, y_{C'}) = (\,x + r\cos(\alpha), y + r\sin(\alpha)\,) \quad (3)$$

and increment the pixel value on the detection map at location ($x_{C'}$, $y_{C'}$) by 1. Once all (r, α) pairs under this bin have been processed, the algorithm moves on to the next edge point on the input edge image.

Here we proposed an improved implementation of the Hough Transform technique based on that originally introduced by [10]. In addition to accessing all (r, α) pairs under this specific $(m+1)^{th}$ bin of the reference table, we employ a Gaussian-weighting system centered on the

$(m+1)^{th}$ bin with a width of Δm = 5. Specifically, we access a total of 11 (=1+2Δm) bins centered on the $(m+1)^{th}$ bin with their respective weights given by a Gaussian distribution g, where g = 1 for the center $(m+1)^{th}$ bin as before and 0 < g < 1 for other 10 neighboring bins, decreasing with respect to distance from center bin. For these other 10 bins, ($x_{C'}$, $y_{C'}$) is also computed for each (r, α) pair under these bins and the corresponding location on the detection map is incremented by the bin's given weight g (smaller than the center weight of 1). See Figure 6 below for an illustration.

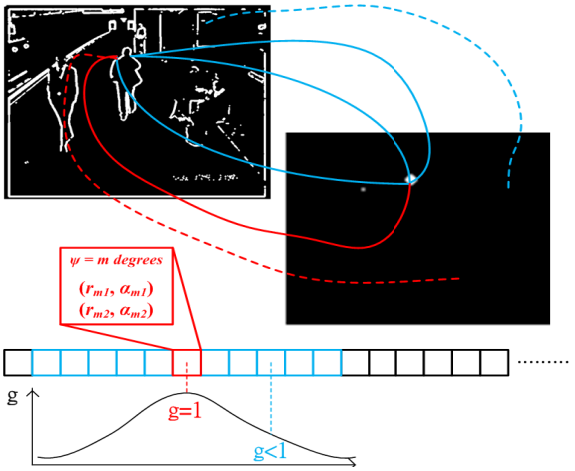

Figure 6. The origin point of the two red lines has a ψ angle of m degrees; it corresponds to the red, $(m+1)^{th}$, bin which contains two (r, α) pairs. Solid lines are correct matches, while dashed lines indicate noise. As shown, different points on the head-shoulder contour in the input edge image converge to a local maximum on the detection map on the right. At the bottom is the 180-bin reference table; a Gaussian weighting g is shown below it with the center bin labeled in red and neighboring 10 bins in blue.

The reason for making this modification is that for a ψ angle that is computed from an edge point on the input edge image, there is an inherent error due to pixel quantization and angle quantization, and thus the ψ angle obtained is at best indicative only of a small range of neighboring angles. We model this range by applying a Gaussian-weighting system on a range of ψ-angle bins, the range being specified by Δm = 5. Also, by allowing ψ angle to vary within a limited range, our system can handle human head-shoulder contours that are slightly out of alignment with perfectly frontal image.

Following the above procedure, once all edge points on the input edge image have been traversed, the supposed center point of the contour of interest in the input edge image will correspond to a local maximum in the detection map. See Figure 7 for the detection results given by our improved version of Hough Transform, in comparison with the original implementation of Hough Transform introduced in [10] and a simple normalized correlation

Figure 7. Upper row: (left) input video frame; (middle) the input edge image by 3x3 sobel convolution; (right) input frame with HT detection map superimposed on it using our improved implementation. Lower row: (left) Hough Transform detection result with our improved implementation, which gives one large response at the correct location of the most visible head-shoulder contour in the input edge image; (middle) detection result with original implementation of HT, which gives more false positives; (right) simple implementation of normalized correlation between the input edge image and contour template, which gives the most false positives, as expected.

technique for feature detection. Our improved method is shown to be superior to the latter two. And since this contour detection operation is solely performed within the search region, interference from irrelevant contours is reduced to a minimum. The generalized version of Hough Transform that we use here effectively maps an arbitrary contour of our choice (head-shoulder silhouette in this case) to a large-valued point (or a small region of scattered bright points due to pixel quantization).

3.3. Integration into motion prediction

For detection of luggage owner in a single frame, the color information from Cr channel and the upper-body contour information from our improved Hough Transform algorithm are combined to pin-point the head location of the owner.

To further exploit the temporal relationship between successive frames, motion prediction is employed. Prediction of the owner's location in the next frame is based on its location in the current frame and in the past frames with exponentially-decaying weights. Specifically, if we denote r(t) as the position vector of the owner at time t, the prediction for time t+1 can be formulated as

$$r(t+1) = r(t) + \Delta r \qquad (4)$$

where Δr is generated recursively by motion prediction and is given by

$$\Delta r = \alpha\, \Delta r + \beta(r(t) - r(t\text{-}1)) \quad \text{and} \qquad (5)$$

$$\alpha + \beta = 1. \qquad (6)$$

The fact that Δr is calculated recursively ($\alpha \neq 0$) ensures that (1) past information is taken into account and also that (2) past influences decay exponentially with time, so the prediction can follow the object as it changes speed. In our case, the exponential smoothing coefficients α and β are empirically determined to be 0.4 and 0.6, respectively.

Our approach employs three measures in calculating the probability score for the owner's tracked trajectory, which will be used in the final probabilistic reasoning to evaluate an overall confidence score for the luggage-abandonment event. These three measures are the location, size and color histogram of the luggage owner from the prediction by last frame and from the detection made on the present frame, as in [2]. The closer the prediction and the detection are, the higher the probability score is. Let P_{TOTAL} denote the probability score combining the three measures; P_{POS}, P_{SIZE} and P_{CH} denote the scores of the position measure, size measure and color-histogram measure, respectively; r represents the position vector, s the size (in pixel area) and c the color histogram. They are defined as follows with subscript P corresponding to prediction and D to detection:

$$P_{\text{POS}}(r_{\text{P}}, r_{\text{D}}) = exp\left(-\frac{(x_P - x_D)^2}{\sigma_x^2}\right) exp\left(-\frac{(y_P - y_D)^2}{\sigma_y^2}\right)$$

$$P_{\text{SIZE}}(s_{\text{P}}, s_{\text{D}}) = exp\left(-\frac{(s_P - s_D)^2}{\sigma_s^2}\right)$$

$$P_{\text{CH}}(c_{\text{P}}, c_{\text{D}}) = \frac{1}{\sqrt{2\pi}\sigma} exp(-D^2/2\sigma^2)$$

where D is in fact the Bhattacharyya distance between the two color histograms, as in $D^2 = 1 - \sum_{i=1}^{256} \sqrt{c_{\text{P}}(i) c_{\text{D}}(i)}$; σ is the standard deviation. The total probability score is given by combining the above three measures, each with a scale factor λ so that they sum up to 1, as

$$P_{\text{TOTAL}} = \lambda_{\text{POS}} P_{\text{POS}} + \lambda_{\text{SIZE}} P_{\text{SIZE}} + \lambda_{\text{CH}} P_{\text{CH}}. \quad (7)$$

4. Probabilistic event model

The tracking module provides a trajectory and its associated probability score. It computes the distance from the owner's feet to the luggage in question for each incoming frame; this distance is used to determine (1) at which time point the owner leaves the scene or the luggage and (2) for how long the luggage has been left unattended. In our definition, a luggage is formally declared abandoned when its owner leaves the scene without it or when it has been left unattended for 30 consecutive seconds.

Our definition of the luggage-abandonment event follows a probabilistic framework [3]. We use the symbol A for the event and O for the observation; here O represents the owner's tracked trajectory in the case when the owner has either left the scene without the luggage or left the luggage for more than 30 seconds. The probability of A given the observation O, i.e. $\text{P}(A \mid O)$, is what we call the overall confidence score of the event and is what we are seeking. However this conditional probability is difficult to come by directly, and therefore we formulate the problem as a Maximum A Posteriori (MAP) problem. The probability of A occurring independently is given to be 0.5, i.e.

$$\text{P}(A) = 0.5,$$

and we define the conditional probability that given an luggage-abandonment event that has already happened, the owner must be observed to have left the scene without the luggage or left the luggage for 30 consecutive seconds, with a certainty of 0.95, which translates into

$$\text{P}(O \mid A) = 0.95.$$

The probability of O can be obtained from (7). In (7) the probability P_{TOTAL} is computed between two successive frames, and here we model $\text{P}(O)$ as the mean value of P_{TOTAL} over all frames processed, as

$$\text{P}(O) = \frac{\sum_n P_{\text{TOTAL}}}{n} = \frac{1}{n} \sum_n (\lambda_{\text{POS}} P_{\text{POS}} + \lambda_{\text{SIZE}} P_{\text{SIZE}} + \lambda_{\text{CH}} P_{\text{CH}}). \quad (8)$$

Under the MAP principle, we can then evaluate the posteriori probability of A given O as

$$\text{P}(A \mid O) = \frac{P(A \cap O)}{P(O)} = \frac{P(A \cap O)}{P(A)} \frac{P(A)}{P(O)} = \text{P}(O \mid A) \frac{P(A)}{P(O)}, \quad (9)$$

which can be conveniently evaluated once $\text{P}(O)$ is known. This posteriori probability is the overall confidence score of the luggage-abandonment event defined above. If this confidence score $\text{P}(A \mid O)$ exceeds a pre-determined threshold ρ, the luggage is declared abandoned with a certainty of $\text{P}(A \mid O)$ and an alarm is triggered accordingly.

One of the advantages of using a probabilistic framework in event modeling is that it offers a user greater flexibility. Users of the system are able to adjust the value of ρ to achieve varying levels of system sensitivity. Another advantage is that it conveys a better idea of how *real* a potentially dangerous situation might be by reporting a confidence score along with the event.

5. Experimental results

We have tested our proposed method on surveillance video sequences from datasets provided by the 2007 IEEE International Conference on Advanced Video and Signal based Surveillance and the 2006 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. The AVSS 2007 dataset contains three

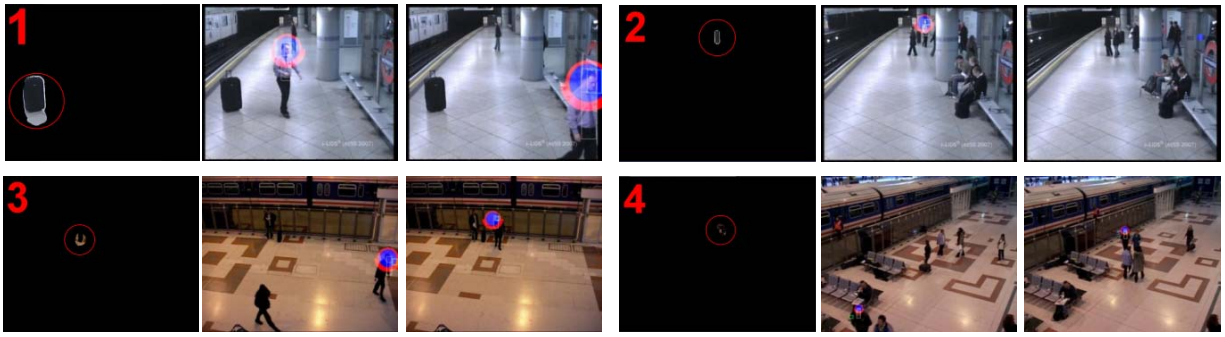

Figure 8. Sequence 1, 3 and 4: (from left) static luggage detected; owner tracking starts; owner leaves the scene, alarm triggered. Sequence 2: static luggage detected; owner tracking starts; owner lost due to occlusion, alarm triggered. Sequence 1 and 2 are from AVSS 2007 dataset; Sequence 3 and 4 from PETS 2006 dataset.

sequences recorded with different difficulty levels: easy, medium and hard. The easy sequence contains objects of larger appearance, activities which are closer to the camera and less scene clutter; as the difficulty level rises, objects become smaller and clutter more serious. Our proposed 3-stage approach has successfully detected the abandoned luggage in all three sequences from the AVSS 2007 dataset. We have been able to track the owner in the easy sequence all the way until he leaves the scene without the luggage, hence resulting in an alarm event. In the medium and hard sequences, however, the owner passes behind a large pillar before leaving the scene without the luggage, and therefore is occluded for about 1.5 seconds, which translates into around 40 frames under a frame rate of 25 fps. Our tracking engine has not been able to follow the owner through the occlusion and the owner is deemed to be lost; therefore alarms are also triggered for these two sequences. The PETS 2006 datasets contains seven sequences. In video 1, 2, 4, 5 and 6 the luggage owner leaves the scene without the luggage, and our method has successfully issued an alarm in all these 5 cases while tracking the owner all the way until the owner is no longer within camera view. In video 3 the owner stays with the luggage all the time, and therefore no alarm is issued. In video 7, the owner wanders about for some time before finally leaving the scene without luggage; the trajectory of the highly-maneuvering owner, however, contains too many abrupt changes in speed and direction for our present motion prediction algorithm to successfully follow. The owner is lost 34 seconds after he leaves the luggage, while an alarm is triggered at 30 seconds. In Figure 8, some labeled scene shots are provided.

6. Conclusion and future work

In this paper, we have proposed a novel approach to left-luggage detection in surveillance video. Through the use of foreground-mask sampling, we are able to emulate the human vision capability of localizing and focusing on solely the object of interest to us, while filtering out all other irrelevant, interfering agents. We are therefore able to apply tracking in a selective, more localized manner. We have also proposed an improved implementation of the Hough Transform for detecting the human upper-body contour from the video frames. And we have incorporated a probabilistic framework and employed the MAP principle in our modeling of the luggage-abandonment event and subsequent reasoning. In the future, we plan to extend our proposed approach to a multi-camera network where coordination of an array of cameras will allow cues to be gathered from multiple views and information to be relayed from one to another.

7. References

[1] B. Wu and R. Nevatia, "Detection of Multiple, Partially Occluded Humans in a Static Image by Bayesian Combination of Edgelet Part Detectors", *ICCV 2005, IEEE*, Vol I: 90-97

[2] B. Wu and R. Nevatia, "Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection", *CVPR 2006, IEEE*, Vol I: 951-958

[3] F. LV, X. Song, B. Wu, V. K. Singh, R. Nevatia, "Left-Luggage Detection using Bayesian Inference", *9th PETS, CVPR 2006, IEEE*, pp. 83-90

[4] Kumar, C. N. Ravi and Bindu. A, "An Efficient Skin Illumination Compensation Model for Efficient Face Detection", *IECON 2006, IEEE*, pp. 3444-3449

[5] K. Smith, P. Quelhas and D. Gatica-Perez, "Detecting Abandoned Luggage Items in a Public Space", *9th PETS, CVPR 2006, IEEE*, pp. 75-82

[6] J. Martínez-del-Rincón, J. Elías Herrero-Jaraba, J. Raúl Gómez, and C. Orrite-Uruñuela, "Automatic Left Luggage Detection and Tracking Using Multi-Camera UKF", *9th PETS, CVPR 2006, IEEE*, pp. 59-66

[7] L. Li, R. Luo, R. Ma, W. Huang, K. Leman, "Evaluation of An IVS System for Abandoned Object Detection on PETS 2006 Datasets", *9th PETS, CVPR 2006, IEEE*, pp. 91-98

[8] J. Zhou, J. Hoang, "Real Time Robust Human Detection and Tracking System", *CVPR 2005, IEEE*, Vol III: 149-149

[9] P.V.C Hough Method and Means for Recognizing Complex Patterns, US Patent 3,069,653, December 1962

[10] R. O. Duda, R. E. Hart, Use of the Hough Transform to Detect Lines and Curves in Pictures, CACM(15), No. 1, January 1972, pp. 11-15