Left-luggage detection using homographies and simple heuristics

Edouard Auvinet, Etienne Grossmann, Caroline Rougier, Mohamed Dahmane and Jean Meunier

Department of Computer Science and Operations Research
University of Montreal
Montreal, CANADA H3C 3J7

Abstract

Today, video surveillance is commonly used in security systems, but requires more intelligent and more robust technical approaches. Such systems, used in airports, train stations or other public spaces, can bring security to a higher level. In this context, we present a simple and accurate method to detect left luggage in a public area, which is observed by a multi-camera system and involving multiple actors. We first detect moving objects by background subtraction. Then, the information is merged in the ground plane of the public space floor. This allows us to alleviate the problem of occlusions, and renders trivial the image-to-world coordinate transformation. Finally, a heuristic is used to track all objects and detect luggage items left by their owners. An alarm is triggered when the person moves too far away from his luggage during a too long period of time. Experimental results prove the efficiency of our algorithm on PETS 2006 benchmark data.

1. Introduction

Faced with the increasing need of security in public spaces, public and commercial interest pushes research to develop active prevention solutions, capable of detecting suspicious events while they occur, rather than just recording them. For example, iOmniscient [4] claims to provide intelligent video surveillance software that detects objects left in crowded or busy areas, using a Non-Motion Detection (NMD) technique.

Surveillance applications developed nowadays are part of third generation surveillance systems [12], that cover a wide area using a multi-camera network. Typical watched areas are sensitive public places and infrastructures, that are susceptible of being crowded. Tracking people in a crowded environment is a big challenge, since, in image space, we must deal with merging, splitting, entering, leaving and correspondence. The problem is more complicated when the environment is observed by multiple cameras. To deal with this, approaches have been proposed which can be classified in two categories : uncalibrated and calibrated.

An interesting example of the uncalibrated method is proposed by Khan and Shah [5]. They take advantage of the lines delimiting the field of view of each camera, which they called *Edges of Field of View*. Similarly, Calderara *et al.* [1] introduce the concept of *Entry Edges of Field of View* to deal with false correspondences.

Among calibrated methods, we can cite the work of Yue *et al.* [13] who use homographies to solve occlusions. A second method is proposed by Mittal and Davis [7] which is based on epipolar lines.

The advantage of having calibrated cameras is that it greatly facilitates the fusion of visual information produced by many cameras.

A partial calibration, in which only camera-to-ground-plane homographies are known, is often used. Indeed, homographies are much easier to obtain than general calibration, while still providing a very useful image-to-world mapping.

In this paper, we will present an algorithm to detect abandoned luggage in a real world public environment. This is a typical challenge of nowadays surveillance systems. For testing purposes, we will use the PETS[1] datasets [10], described below in section 2, that provides multi-camera sequences containing left-luggage scenarios. We will exploit the fact that the PETS datasets provides calibration and sufficient data to estimate homographies.

The method we developed here is similar to the technique recently proposed by Khan and Shah [6]. They present a planar homography constraint to resolve occlusions and detect the locations of people on the ground plane corresponding to their feet.

Our video surveillance process is described in Figure 1. First, we perform a background subtraction in the image plane of each camera (see Section 3.1). Then, a homographic transformation is performed to merge information from all cameras in the scene floor homographic image (see Section 3.2). Finally, we work in the homographic image to track people using a heuristic method to detect suspect events (see Section 3.3).

[1]Performance Evaluation of Tracking and Surveillance

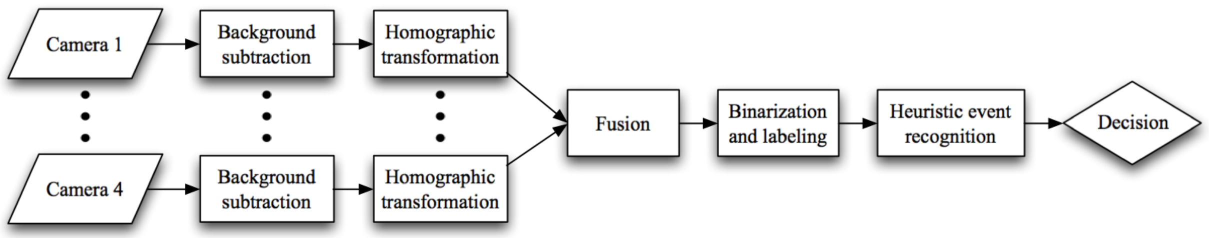

Camera 1
Camera 4
Background subtraction
Background subtraction
Homographic transformation
Homographic transformation
Fusion
Binarization and labeling
Heuristic event recognition
Decision

Figure 1: Scheme of the proposed algorithm.

Our main contribution is to present results obtained by a simple modular system. Its principal merit is that it has few parameters, most of them being easily identified physical quantities (e.g. minimum detected object size). In the technical description below, and in the conclusion, we will discuss principled ways of reducing even further the number of parameters.

2. Datasets

To test our algorithm, we used the datasets provided by the PETS 2006 workshop organization. These videos were taken in a real world public setting, a railway station, and made possible by the support and collaboration of the British Transport Police and Network Rail. There is a total of seven multi-camera sequences containing left-luggage scenarios with increasing scene complexity. Luggage items are of several different types (briefcase, suitcase, rucksack, backpack and even a ski gear carrier). Briefly stated, in the context of PETS 2006, a luggage has been left abandoned if the owner is farther than a given distance from the luggage (300cm) for a certain period of time (30 seconds). For these benchmark videos, calibration data for each individual camera are given and were computed from specific point locations (also given) taken from the geometric patterns on the floor of the station. Ground-truth information such as luggage locations and abandoned-luggage detection times are provided with each datasets. The scenarios involve up to 6 persons with a left-luggage occurrence in each one of them. They were filmed with four DV cameras with PAL standard resolution of 768 x 576 pixels and 25 frames per second.

3. Method

3.1. Motion Detection

Moving visual objects are segmented using a typical simple background subtraction with shadow removing. We construct the background image and classify as foreground any pixel that deviates from it by more than a certain threshold. This threshold is set so that at most ~ 1 % of background pixels are misclassified.

Automatic background model estimation techniques could have been used, but in the present work, a simple semi-automatic method was sufficient.

The background model consists of the median value C_{med} of each color component, for each pixel. The median is computed from ten images taken one second apart at the beginning of the sequence.

We now explain how the threshold is set. We consider the residues of RGB color components with respect to the background (the median) and the threshold is set so that 1 % of the background pixels are misclassified.

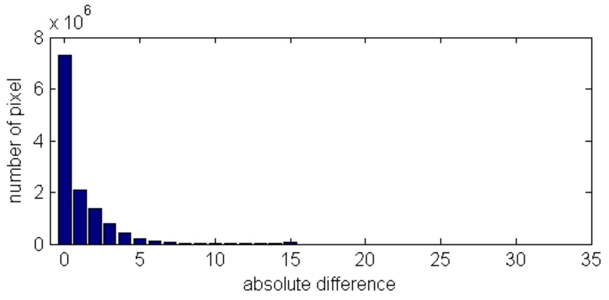


Figure 2: Values of the residues computed as in Eq. 1.

Figure 2 shows the residues computed from the ten images in one of the provided sequences. In the abscissa is the value:

$$\frac{\sum_{n=1}^{N} \left| C_n\left(i,j\right) - C_{med}\left(i,j\right) \right|}{N}, \tag{1}$$

where (i,j) is the pixel coordinate, and N the number of images (N=10 in our case).

The general shape of the histogram varies little from camera to camera, and sequence to sequence. A threshold of 15 gray levels is appropriate for all sequences and cameras, and used everywhere in this work. Potential foreground pixels are thus those that verify:

$$\left| I\left(i,j\right) - C_{med}\left(i,j\right) \right| \geq 15.$$

To obtain the final foreground silhouettes, shadows must be removed. For this purpose, the pixels with a reasonable darkening level and a weak chromatic distortion [2], below a certain threshold, considered as shadow, are therefore ignored. Isolated spurious pixels are removed by a single erosion operation. We then perform 5 dilations with a 3x3 kernel. We will justify this dilation step in the Section 3.2.1.

Once the silhouette binary images are obtained, we are ready to fuse them in the floor homographic plane or orthoimage.

3.2. Information Fusion in the Orthoimage

To obtain coherence information from the different views, we work in the ground orthoimage obtained with homographic transformations. This allows us to overcome the problem of occlusions. Indeed, this orthoimage provides a birds-eye view of the ground plane with information on moving objects in contact with the floor.

3.2.1 Homographic transformation

We use PETS 2006 benchmark data which provides corresponding points on the floor in each field of view of the camera. Two methods could be used to remap the images to the ground plane (i.e. build orthoimages). The first is to use the camera calibration parameters provided in the PETS dataset (computed with the Tsai algorithm [11]). The second method is to compute just the homographic transformations from that maps corresponding points from the image plane to the ground plane. The homographic matrix is a 3x3 matrix and the homographic transformation is:

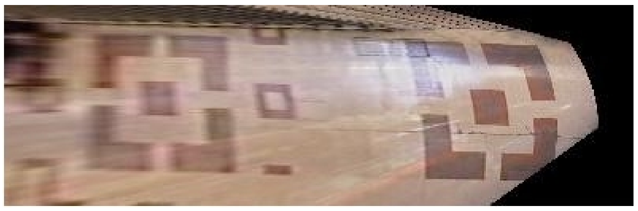
$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}$$

with (x_1, y_1) and (x_2, y_2) a pair of corresponding points.

Using at least four pairs of corresponding points, we are able to compute the homographic matrix using a least-squares method. Since more than four pairs are provided, we use the linear (non-optimal) least-squares estimation method.

Figure 3 shows an example of the result obtained for the two methods for one of the cameras. We notice that the provided Tsai calibration parameters yield a less good orthoimage: apparently the radial distortion correction goes wrong (Figure 3 (a)). In comparison, the homographic transformation computed from the provided point correspondences is relatively free of distortion, so we decided to use only the homographic transformations.

After that, each silhouette binary image is transformed to the orthographic plane using the corresponding homographic matrix. The fusion in the orthoimage is made by adding the pixel values of all four images. In the resulting birds-eye view image, Figure 4 (e), the intersection of each person's silhouette is at the level of the feet.


(a)

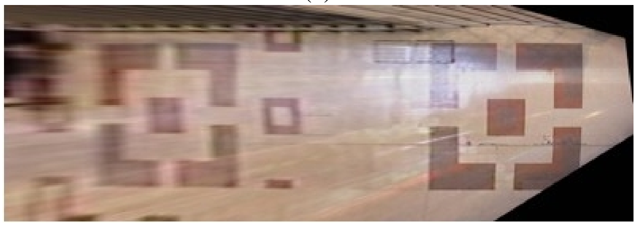

(b)

Figure 3: The birds-eye view obtained with (a) the Tsai camera model and from (b) homographic transformation estimated from the PETS corresponding points. These views correspond to camera #1.

Figure 5 shows the fusion without silhouette dilatation. We observe that there is a mapping error since the foot in each camera are not exactly overlapping after fusion in the floor homographic plane. This error is about 5-10 pixels and is mostly due to camera optical distortion that was not taken into account in our methodology. This is why, to overcome this problem and keep the algorithm as simple as possible, we decided to perform five silhouette dilatations to improve foot overlapping. Another advantage of dilatation is that it ensures the fusion of each foot blob into a unique more stable blob representing the person.

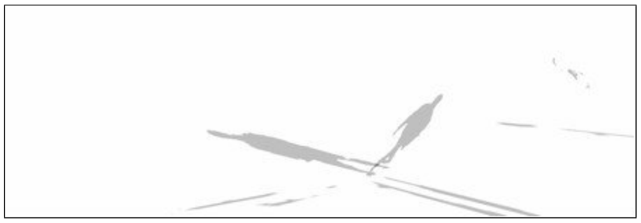


Figure 5: Mapping error in the orthoimage.

3.2.2 Extraction of the blobs

The orthographic image is used to detect the objects that touch the ground. In order to extract the corresponding blobs, we use the information of the overlapping field of

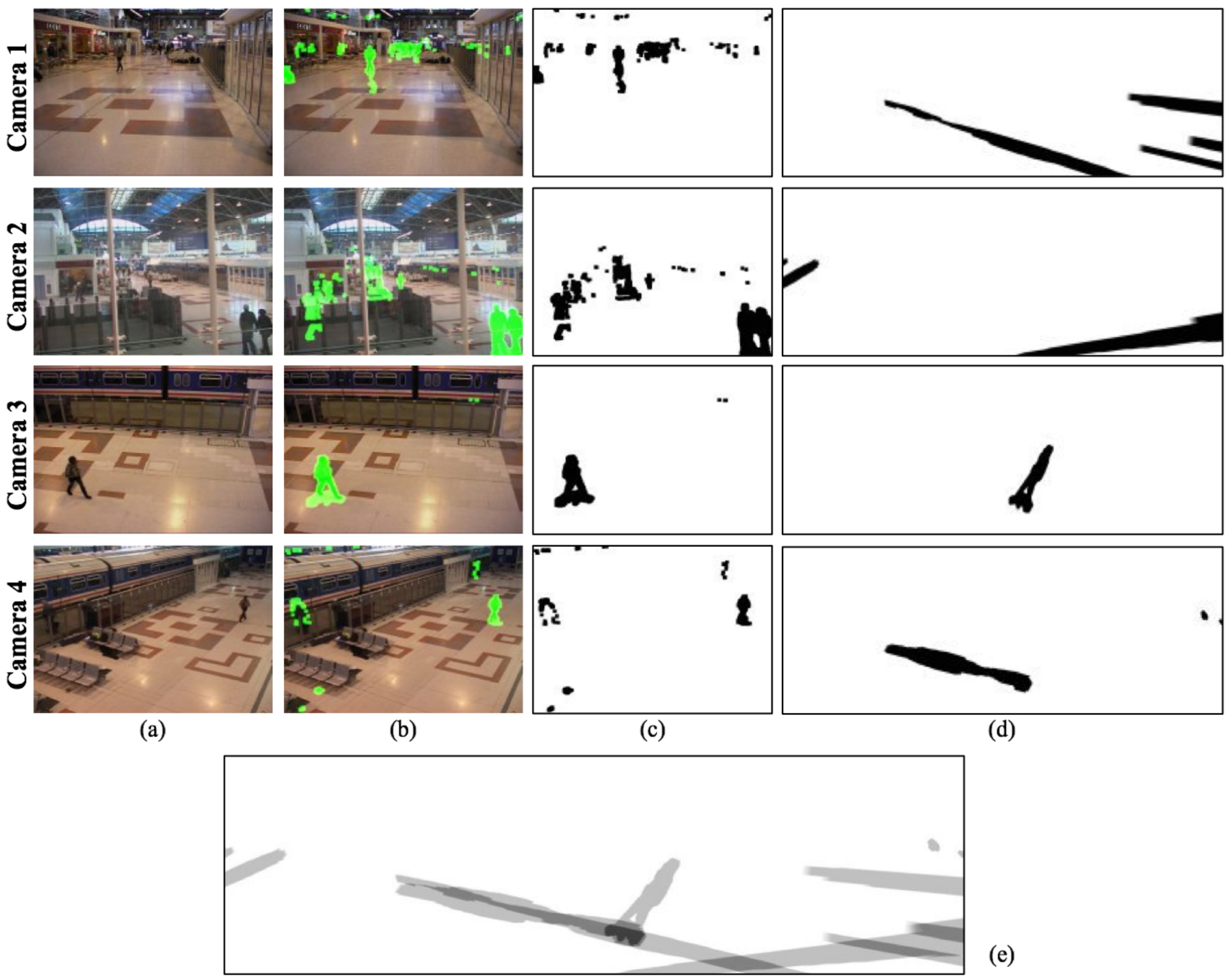


Figure 4: Silhouette extraction and fusion in the orthographic plane for frame #389 sequence 3 of the PETS 2006 dataset. (a) Original images, (b)(c) extracted silhouettes, (d) silhouette mappings and (e) silhouette fusion in the homographic plane.

view of the cameras (Figure 6). For instance, in a four-camera field of view (white area), the threshold for blob detection is set to three. This means that an overlapping of at least three silhouettes is necessary to create blobs: blobs can be detected in the white or light gray area of Figure 6.

3.3. Heuristic Event Recognition

Once the segmentation and blob labeling are performed in the orthoimages, only blob centroid positions (x,y), number of blob pixels N_{pix} and bounding boxes are passed to the event detection heuristic. The event detection heuristic has three main components:

1. Tracking of blobs from frame to frame, thus forming *spatio-temporal entities*.
2. Detection of spatio-temporal forks.
3. Detection of immobile blobs.

Based on the output of these components, a warning is raised when a fork has been detected and one branch (the "luggage" or "immobile object") is immobile, while another branch (the "owner") is more than $b = 300\ cm$ away. An alarm is raised when, in addition, the owner of the immobile object stays at a distance greater than b during 30 seconds. These are the definitions used in our results reported below.

We now describe each of the components in detail.

3.3.1 Tracking

For the purpose of tracking, we model blobs as circles of radius $\rho = \sqrt{N_{pix}/\pi}$, where N_{pix} is the known number of pixels in the blob. Two blobs are said to touch if their circles intersect, that is, if $\| (x_1, y_1) - (x_2, y_2) \|_2 \leq (\rho_1 + \rho_2)$

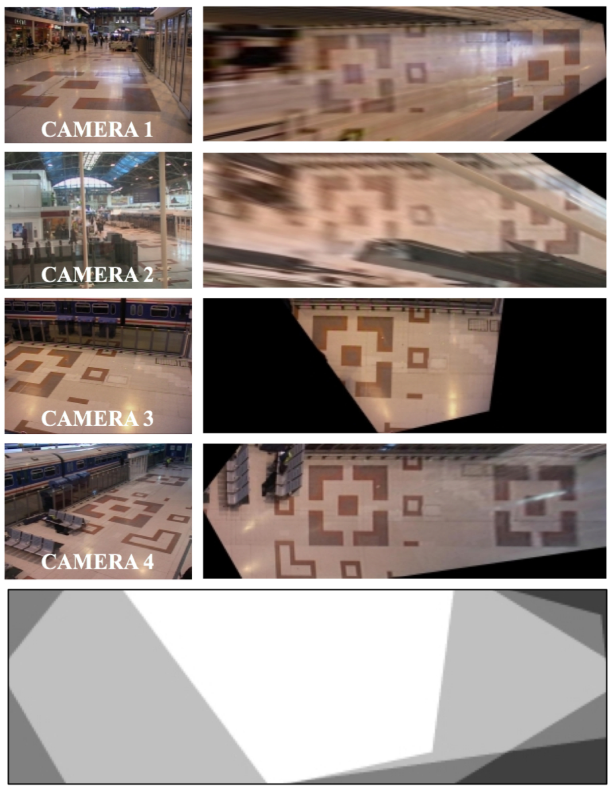


Figure 6: Original images and their homographic transformations, and the overlapping fields of view of all cameras.

where (x_1, y_1), (x_2, y_2) are the blob centroid positions, and ρ_1, ρ_2 are their radii.

If two blobs in consecutive frames touch, then they will be said to pertain to the same spatio-temporal entity (Figure 7).

We record the history of each entity, so that, when two distinct entities touch the same blob in a new frame, that blob will be said to pertain to the entity that has the largest number of previous blobs (Figure 8).

Figure 12 shows all the detected blobs, each colored according to its given label. Note that, with our definition of tracking, two blobs, in the same frame, that do not touch, may still pertain to the same entity (Figure 9).

3.3.2 Detection of spatio-temporal forks

Spatio-temporal forks correspond to objects that separate after having been in contact. Recognizing such patterns is fundamental to detect abandoned objects. A (possibly multi-pronged) fork is said to occur whenever two blobs that do not touch pertain to the same entity. In particular, we are interested in detecting forks in which one branch moves away, while the other remains immobile.

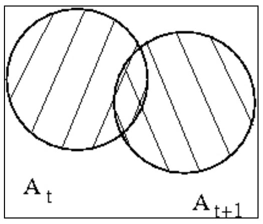


Figure 7: Tracking: blobs A_{t+1} and A_t, observed at times $t + 1$ and t are considered to pertain to the same spatio-temporal entity, because they intersect in the orthoimage.

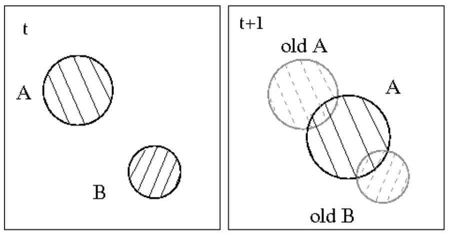


Figure 8: Merging of spatio-temporal entities: when two spatio-temporal entities A and B meet, only the label of the one with most previous blobs is kept.

3.3.3 Detection of immobile objects

Ideally, an immobile object would be characterized by a foreground blob that remains constant in time. In practice, blobs are subject to misdetections, spurious detections and poor localization.

We represent an immobile object as a ground position such that there exists a blob at less than 30cm in each frame, during more than 3 seconds. The immobile object exists as long as there is a blob at less than 30cm from its position. The position y_t of the immobile object (after t frames of existence) is defined as the mean of the closest blobs, at each frame:

$$y_t = \sum_{s=1}^{t} x_s,$$

where x_s is the blob, amongst all blobs x detected in frame s, that is closest to y_s: $x_s = \underset{x}{\arg\min} \ |x - y_s|$.

The distance of 30 cm corresponds to the uncertainty in the localization of the blob centroid. It has been chosen by examining the distribution of distances of blobs around potential immobile objects. The 3 second delay serves only to limit the number of potential immobile objects. Also, the position of a newly-abandoned object is unstable, due to the proximity of the owner. The 3 second delay also gives time for the object position to stabilize. Figures 10 and 11 further

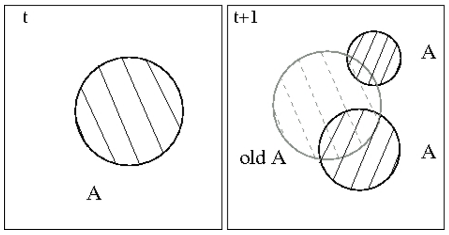


Figure 9: Forking, or branching, of spatio-temporal entities: blobs that are not connected may pertain to the same spatio-temporal entity, as a result of a fork

justify the choice of these quantities. Figure 10 shows the localizations of all detected blobs.

The local maximum of blob density would be the "ideal position" of the detected blob. The ground-truth position of the luggage (given in the PETS dataset), identified by a cross in the zoomed image, is slightly aside the maximum due to imprecision in the orthoimage construction.

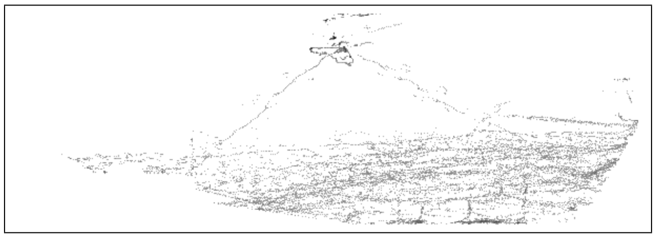
Although the local maximum appears well localized, there are frames in which no blob is in fact detected. In these frames the blob nearest to the object is much farther. These frames form spikes in Figure 11, which shows the distance between the ideal position (maximum density) and nearest detected blob, in Sequences 1 and 5. Given the amplitude of the spikes in this last sequence, a distance of 30 pixels is a safe choice.

We now explain how the three components above are combined to detect left luggage. At each time frame, we identify forks in which one branch is immobile and another (the "owner") is more than $b = 300\ cm$ away from the immobile branch. In such forks, we raise a warning and change the entity label of the owner, to be able to identify it later. If the luggage remains in place for 30 seconds, during which time the owner does not move closer than $b = 300\ cm$ from the luggage, then the object will be considered to be abandoned, and an alarm is raised.

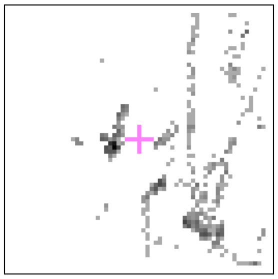
4. Results

In this section, we report our results on all seven PETS datasets. Two sets of results are given, one without the shadow suppression method, the other with shadow suppression. All parameters are otherwise exactly the same in all reported results.

An important information in video surveillance is the computation time. The pixel treatment is compiled in C++ with the OpenCV library [9], and the tracking runs under the Octave software [8]. On a Centrino, 2.26 GHz, the image treatment takes about 0.4 s per frame for 1200x400 homographic images. The tracking of objects with Octave takes 0.02 seconds per frame on a 1.4 GHz Celeron M.



Density of blob occurrences PETS 2006 in Sequence 1



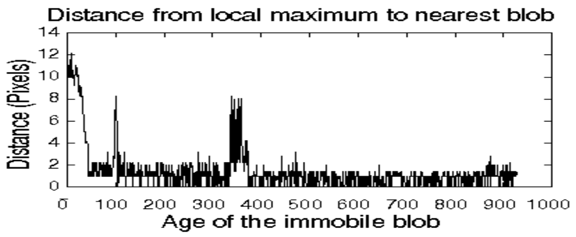
Zoom around the true abandoned object position (cross)

Figure 10: Density of blob occurrences in Sequence 1. The gray level represents the number of blobs detected at that pixel during the sequence. Top: Complete surveilled area; note the high density near the top, corresponding to the object, with the streaks left by the object carrier coming and going. Bottom: Zoom around the true object (location, marked with a cross), and the nearby local maximum of the number of detected blobs (dark mass).

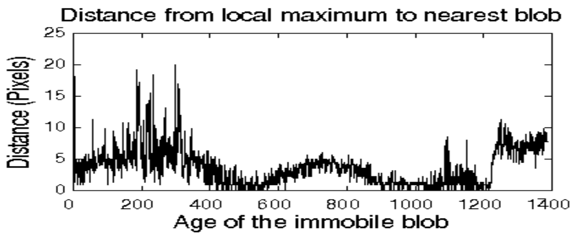
Seq.	TP	FP	Spatial error for TP (cm)	Temporal error for TP (s)	Subjective difficulty
1	1	0	25.8	+0.1	*
2	1	0	16.9	+2.5	***
3	0	0	-	-	*
4	1	0	63.9	+1.7	****
5	1	13	43.8	+0.2	**
6	1	0	43.9	+12.2	***
7	1	3	59.3	+0.5	*****

Table 1: Left-luggage detection without shadow removal. TP : True Positive (correct detection), FP : False Positive (incorrect detection). * : very easy, ***** : very difficult.

Tables 1 and 2 show the results of our algorithm on the seven datasets given by the PETS workshop organization. As expected the errors are usually larger when the shadows are not removed. Figure 13 shows the results in 3D: Warnings and alarms are represented by yellow and red spheres, respectively. Considering the simplicity of our methodology, the results are very satisfactory.



Blob occurrences in Sequence 1



Blob occurrences in Sequence 1

Figure 11: Distance, in cm, between local maximum of blob density and nearest blob, for the time interval during which the object is present there. The spikes in these curves justify a tolerance of 30cm during the localization of immobile objects.

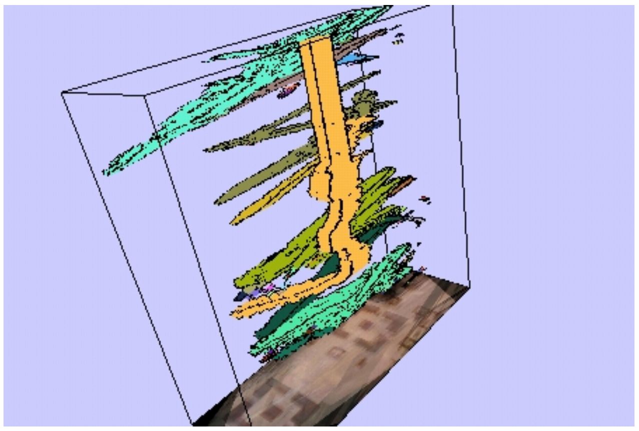
Seq.	TP	FP	Spatial error for TP (cm)	Temporal error for TP (s)	Subjective difficulty
1	1	0	11.8	+0.0	*
2	1	0	15.6	+0.2	***
3	0	0	-	-	*
4	1	0	37.7	+1.0	****
5	1	5	48.4	+0.2	**
6	1	0	10.3	+2.3	***
7	1	0	70.9	+0.7	*****

Table 2: Left-luggage detection with shadow removal. TP : True Positive (correct detection), FP : False Positive (incorrect detection). * : very easy, ***** : very difficult.

5. Conclusion

The proposed algorithm has the important advantage of being very simple. It has few parameters and we have shown how to set these parameters based on the input data.

As a consequence of its simplicity, our algorithm has some limitations. For instance, the tracking algorithm exploits the fact that the motions of the blobs of interest is typically small with regard to the blobs' spatial extents. This allows simple correspondence of blobs by means of bounding circles. However, if the motion becomes larger, this simple tracking methodology will fail and more complex motion description or prediction will be necessary. Another weakness of the tracking algorithm is the somewhat limited supervision of the temporal evolution of the blobs. For instance when two blobs merge and, after a while, split again into two blobs, there is no way in the proposed algorithm to identify each one of them (or to make the correspondence between the blobs before and after the merging). For in-



Sequence 1

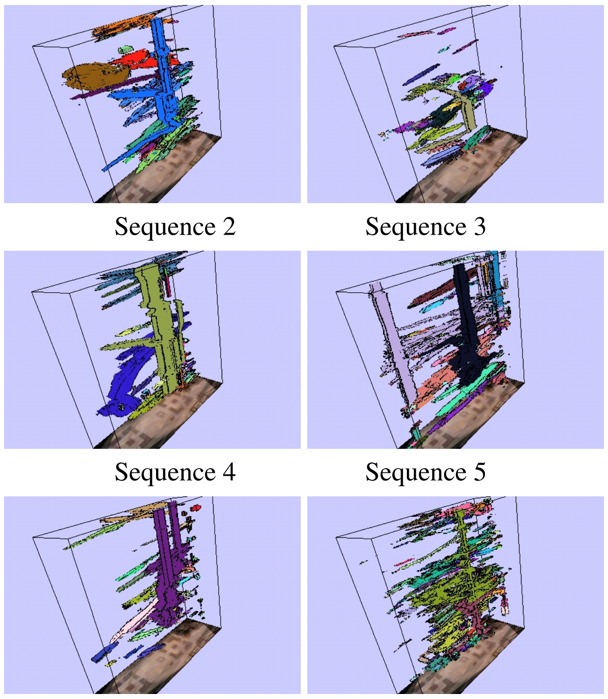


Figure 12: Spatio-temporal entities, identified by the heuristic of Section 3.3.1. Each entity is given a distinct color. Here, each blob is represented by its rectangular bounding box. One may identify some individual trajectories by a single color, while others are merged into a single entity.

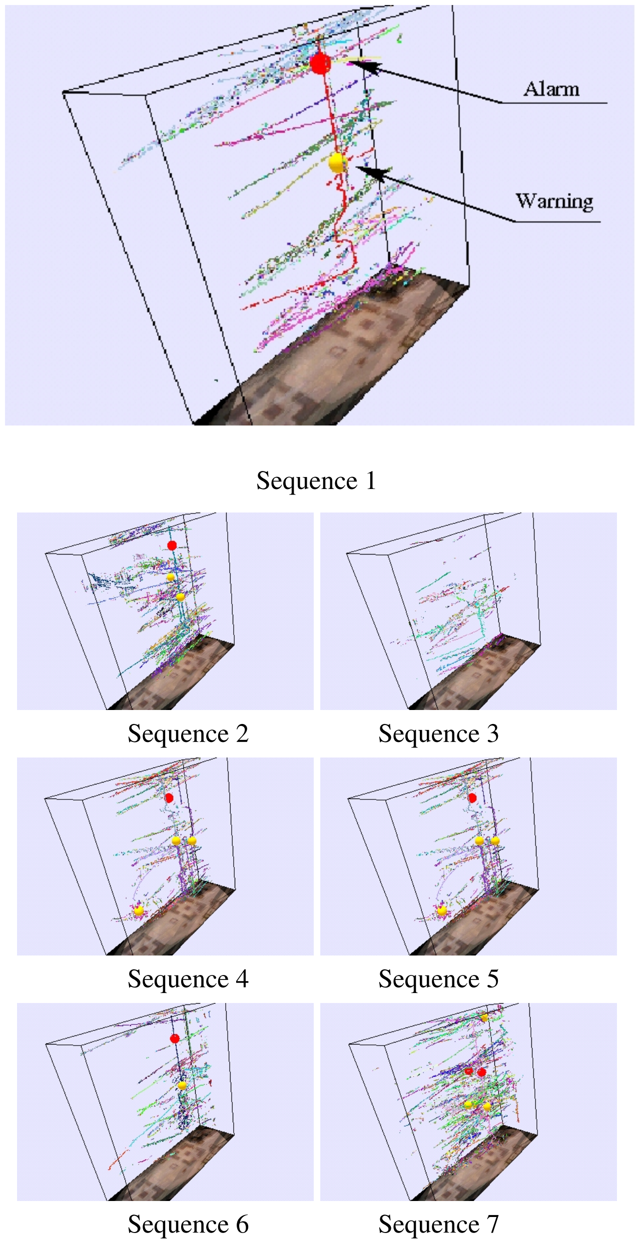


Figure 13: 3D visualization of the blobs and events detected in each of the seven sequences. These results are from the experiment reported in Table 2. Here, each blob is represented by a single colored point, while warnings and alarms are represented by yellow and red spheres, respectively.

stance, this could be a problem if the owner of a luggage meets another person forming a unique blob. After a few moments, if the owner leaves the scene, the algorithm will not be able to identify the leaving blob (the owner or the visitor?). Blob correspondence could be implemented based on the color histogram of the individual blobs (and corresponding silhouettes) before their grouping; this would allow the determination of the corresponding persons after an eventual future splitting but at the price of a more complex algorithm.

References

[1] S. Calderara, R. Vezzani, A. Prati and R. Cucchiara, "Entry edge of field of view for multi-camera tracking in distributed video surveillance", In *Proc. of IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 93-98, Sept. 2005

[2] M. Dahmane and J. Meunier, "Real-Time Video Surveillance with Self-Organizing Maps", In *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pp. 136-143, 2005.

[3] L.M. Fuentes and S.A. Velastin, "People Tracking in Surveillance Applications", In *Proc. of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS2001)*, 2001

[4] `http://www.iomniscient.com`

[5] S. Khan and M. Shah, "Consistent labelling of tracked objects in multiple cameras with overlapping fields of view", In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, Issue 10, pp. 1355-1360, Oct. 2003

[6] S. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint", In *European Conference of Computer Vision*, May 2006

[7] A. Mittal and L. Davis, "Unified multi-camera detection and tracking using region-matching", In *Proc. of IEEE Workshop on Multi-Object Tracking*, pp. 3-10, July 2001

[8] J.W. Eaton, "GNU Octave Manual", *Network Theory Limited*, isbn : 0-9541617-2-6, 2002

[9] `http://www.intel.com/technology/computing/opencv/index.htm`

[10] `http://pets2006.net`

[11] R.Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision", In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, pp. 364-374, 1986

[12] M. Valera and S.A. Velastin, "Intelligent distributed surveillance systems: a review", In *IEEE Proceedings Vision, Image and Signal Processing*, Vol. 152, Issue 2, pp. 192-204, April 2005

[13] Z. Yue, S.K. Zhou and R. Chellappa, "Robust two-camera tracking using homography", In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 1-4, May 2004