Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 1165–1171

www.elsevier.com/locate/imavis

People tracking in surveillance applications

Luis M. Fuentes[a,*], Sergio A. Velastin[b]

[a]Departamento de Fisica Aplicada, Facultad de Ciencias, Universidad de Valladolid, 47071 Valladolid, Spain
[b]Digital Imaging Research Center, Kingston University, Penrhyn Road Kingston-upon-Thames, Surrey KT1 2EE, UK

Received 7 April 2003; received in revised form 6 June 2005; accepted 7 June 2005

Abstract

This paper presents a real-time algorithm that allows robust tracking of multiple objects in complex environments. Foreground pixels are detected using luminance contrast and grouped into blobs. Blobs from two consecutive frames are matched creating the matching matrices. Tracking is performed using direct and inverse matching matrices. This method successfully solves blobs merging and splitting. Some application in automatic surveillance systems are suggested by linking trajectories and blob position information with the events to be detected.

Keywords: CCTV Surveillance; Tracking; Automatic surveillance

1. Introduction

Video surveillance of human activity usually requires people to be tracked. Information about their behaviour can be obtained from characteristics of their trajectories and the interaction between them. The analysis of a single blob position or trajectory can determine whether the person is standing in a forbidden area, running, jumping or hiding. Combining such information from two or more people may provide information about the interaction between them. In the process leading from an acquired image to the information about objects in it, two steps are particularly important: foreground segmentation and tracking. In this paper we present a simplified foreground detection method based on luminance contrast and a straightforward tracking algorithm that relies only on blob matching information without having to use statistical descriptions to model or predict motion characteristics. The presented tracker is part of software developed in the UK's EPSRC founded project PerSec [14]. Originally designed to work with CCTV footage from London Underground stations (indoors), it places more emphasis on ensuring real-time execution and studying the interaction between blobs than obtaining a precise trajectory of them. Although background-updating techniques have not been used the algorithm has been tested with PETS2001 image sets to provide examples of simple trajectories. Further application to the detection of events is suggested based on the analysis of centroids and trajectories.

2. Related work

Foreground detection algorithms are normally based on background subtraction algorithms (BSAs) [1–4], although some approaches combine this method with a temporal difference [5]. These methods are based on extracting motion information by thresholding the differences between the current image and a reference image (background) or the previous image, respectively. BSAs are widely used because they detect not only moving objects but also stationary objects not belonging to the scene. The reference image is defined by assuming a Gaussian model for each pixel. BSAs are normally improved by means of updating their statistical description so as to deal with changing lighting conditions [6–8], normally linked with outdoor environments. Some authors present a different model of background, using pixels' maximum and minimum values and the maximum difference between two consecutive frames [4], a model that can clearly take advantage of the updating process. Pixels of each new frame are then classified as belonging to the

[*] Corresponding author.
E-mail address: luis.fuentes@computer.org (L.M. Fuentes).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.06.006

background or the foreground using the standard deviation to define a threshold. After the segmentation of the foreground pixels, some processing is needed to clean noisy pixels and define foreground objects. The cleaning process usually involves 3×3 median [7] or region-based [4] filtering, although some authors perform a filtering of both images-current and background-before computing the difference [3,6]. The proposed method is simpler. No model is needed for the background, just a single image. For outdoor applications this background image may be updated. Tracking algorithms establish a correspondence between the image structure of two consecutive frames. Typically the tracking process involves the matching of image features for non-rigid objects such as people, or correspondence models, widely used with rigid objects like cars. A description of different approaches can be found in Aggarwal's review, [9]. As the proposed tracking algorithm was developed for tracking people, we reduce the analysis of previous work to this particular field. Many approaches have been proposed for tracking a human body, as can be seen in some reviews [9, 10]. Some are applied in relatively controlled [3,8,11] or in variable outdoor [4,7] environments. The proposed system works with blobs, defined as bounding boxes representing the foreground objects. Tracking is performed by matching boxes from two consecutive frames. The matching process uses the information of overlapping boxes [7], colour histogram back projection [12] or different blob features such as colour or distance between the blobs. In some approaches all these features are used to create the so-called matching matrices [2]. In many cases, Kalman filters are used to predict the position of the blob and match it with the closest blob [11]. The use of blob trajectory [11] or blob colour [7] helps to solve occlusion problems.

3. Segmentation

Foreground pixels detection is achieved using luminance contrast [13]. This method simplifies the background model, reducing it to a single image, and it also reduces computational time using just one coordinate in colour images. The central points of the method are described below.

3.1. Definition of luminance contrast

Luminance contrast is an important magnitude in psychophysics and the central point in the definition of the visibility of a particular object. Typically, luminance contrast is defined as the relative difference between object luminance, L_O, and surrounding background luminance, L_B:

$$C = \frac{L_O - L_B}{L_B} \tag{1}$$

As can be seen, positive and negative values are possible, negative contrast meaning an object darker than the background. To apply this concept in foreground detection we propose an alternative contrast definition comparing the luminance coordinate in the YUV colour system 'y' of a pixel $P(i,j)$ in both the current and the background images:

$$C(i,j) = \frac{y(i,j) - y_B(i,j)}{y_B(i,j)} \tag{2}$$

Luminance values are in the ranges [0,255] for images digitised in YUV format or [16,255] for images transformed from RGB coordinate [13]. Null (zero) values for background 'y' coordinate are changed to one because the infinite contrast value they produce has no physical meaning. With these possible luminance values, contrast will be in the non-symmetrical range [−1,254]. Values around zero are expected for background pixels, negative values for foreground pixels darker than their corresponding background pixels and positive values for brighter pixels, Fig. 1. However, highest values are obtained under the unusual circumstances of very bright objects against very dark background and values bigger than 10 are not likely to be obtained.

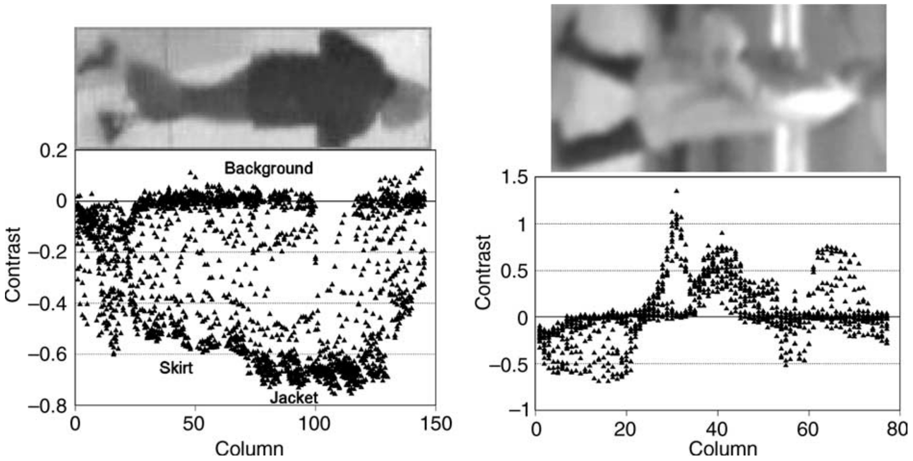


Fig. 1. Two examples of luminance contrast distribution. The right hand plot, showin and example of positive contrast, has been sub-sampled.

3.2. Foreground detection and blob selection

According to the non-symmetrical distribution of contrast around zero, the foreground detection algorithm should use two different thresholds for positive C_P and negative C_N values of contrast, depending on the nature of both the background and the objects to be segmented, although a single contrast threshold $C = C_P = -C_N$, may be used for the sake of simplicity. According to our experience, a contrast threshold between 0.15 and 0.20 works fine in most of the cases, and can deal even with the minor illumination changes that can occur indoors, [13]. So, a pixel $P(i, j)$ is set to foreground when the absolute value of its contrast is bigger than the chosen threshold C. Otherwise it is set to background. A median filter is applied afterwards to reduce noise and the remaining foreground pixels are grouped into an initial blob. This blob is divided horizontally and vertically using X–Y projected histogram, box size and height-to-width ratio. Resulting blobs are classified, according to their size and aspect, and characterised with the following features: bounding box, width, height and the centroid of foreground pixels in the box. In particular, box size, aspect and centroid's position are used to avoid undesirable divisions, like splitting a person into several blobs.

3.3. Tracking

The algorithm described here uses a two-way matching matrices algorithm (matching blobs from the current frame with those of the previous one and vice versa) with the overlapping of bounding boxes as a matching criterion-two blobs in two consecutive frames match when their bounding boxes overlap-, Fig. 2. This criterion has been found to be effective in other approaches [7] and does not require the prediction of the blob's position since the visual motions of blobs are normally small relative to their spatial extents. Due to its final application, the algorithm works with relative positioning of blobs and their interaction forming or dissolving groups and does not keep the information of blob's position when forming a group. However, the proposed system may be easily enhanced. Colour information may be used in the matching process and the predicted position may be used to track individual blobs.

3.3.1. Matching matrices

Let us take two consecutive frames, $F(t-1)$ and $F(t)$. Foreground detection and blob identification algorithms result in N blobs in the first frame and M in the second. To find the correspondence between both sets of blobs, two matching matrixes are evaluated: the matrix matching the new blobs, $B_i(t)$, with the old blobs, $B_j(t-1)$, called M_t^{t-1} and the matrix matching the old blobs with the new ones M_{t-1}^t (3). To clarify the matching, the concept of 'matching string' is introduced. Its meaning is clear, the numbers in column k show the blobs that match with the blob k, (4).

$$M_{t-1}^t(i,j) = \text{Matching}\{B_i(t-1), B_j(t)\}$$

$$M_t^{t-1}(i,j) = \text{Matching}\{B_i(t), B_j(t-1)\} \tag{3}$$

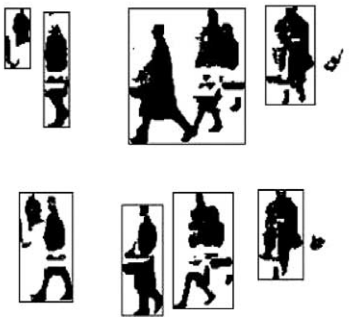
$$S_{t-1}^t(i) = \bigcup_j \frac{j}{M_{t-1}^t(i,j)} = 1 \tag{4}$$

It is possible for one blob to get a positive match with two blobs and, sometimes, with three. In this case, the corresponding string element has to store two or three values. An example of all these concepts appears below, Fig. 2.

3.3.2. Tracking

The algorithm solves the evolution of the blob from frame $F(t-1)$ to frame $F(t)$ by analysing the values of the matching strings of both frames. Simple events such as people entering or leaving the scenario, people merging into a group or a group splitting into two people are easily solved. The correspondence between some events in the temporal evolution of the blobs and the matching strings is easy to follow, merging (5), splitting (6), entering (7), leaving (8) and correspondence (9).

$$B_i(t-1) \cup B_j(t-1) \equiv B_k(t) \Rightarrow \begin{matrix} S_{t-1}^t(i) = S_{t-1}^t(j) = k \\ S_t^{t-1}(k) = i \cup j \end{matrix} \tag{5}$$



$$M_{t-1}^{t} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_t^{t-1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S_{t-1}^{t} = \begin{bmatrix} 1 & 1 & \begin{matrix} 2 \\ 3 \end{matrix} & 4 \end{bmatrix} \quad S_t^{t-1} = \begin{bmatrix} \begin{matrix} 1 \\ 2 \end{matrix} & 3 & 3 & 4 \end{bmatrix}$$

Fig. 2. An example of detected blobs in two consecutive frames, the matching matrices and strings.

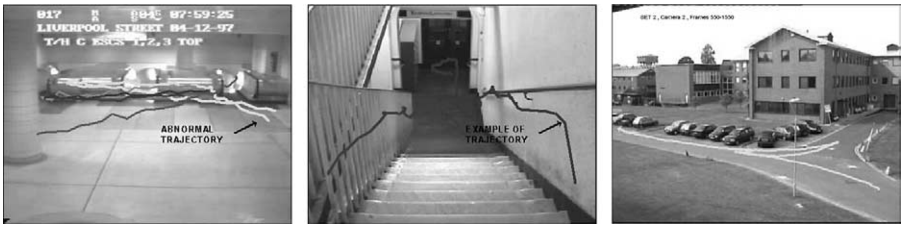


Fig. 3. An example of trajectories in different scenarios.

$$B_i(t-1) \equiv B_j(t) \cup B_k(t) \Rightarrow \begin{array}{l} S_{t-1}^{t}(i) = j \cup k \\ S_{t}^{t-1}(j) = S_{t}^{t-1}(k) = i \end{array} \tag{6}$$

$$B_i(t) \equiv \text{new} \Rightarrow \begin{array}{l} S_{t-1}^{t}(j) \neq i \quad \forall j \\ S_{t}^{t-1}(i) = \varnothing \end{array} \tag{7}$$

$$B_i(t-1) \equiv \text{leaves} \Rightarrow \begin{array}{l} S_{t-1}^{t}(i) = \varnothing \\ S_{t}^{t-1}(j) \neq i \quad \forall j \end{array} \tag{8}$$

$$B_i(t-1) \equiv B_j(t) \Rightarrow \begin{array}{l} S_{t-1}^{t}(i) = j \\ S_{t}^{t-1}(j) = i \end{array} \tag{9}$$

After classifying, the matching algorithm updates each new blob using the information stored in the old ones and keeps the position of the centroid to form a trajectory when the blob is being tracked. If two blobs merge to form a new one, this particular blob is classified as a group. This new group blob is tracked individually although the information about the two merged blobs is stored for future use. If the group splits again, the system uses speed direction and blob characteristics -colour may be used here, for example- to identify correctly the two splitting blobs. It is always possible an interpolation of the position of the tracked blob in the frames where ir was forming part of a group in order to obtain complete trajectories whenever it is necessary (Fig. 3).

Tracking blobs centroid from frame to frame, trajectories of single persons or cars are easily obtained. During occlusion, trajectories are completed with interpolated centroids position. When the occlusion is produced by an fixed obstacle, the median speed of previous frames is used; if the occlusion is due to blob merging, the centroid of the new blob is also considered.

4. Event detection

Blob detection provides 2D information allowing an approximate positioning of people in the 3D scenario -a more precise positioning requires either a geometric camera calibration or stereo processing analysing simultaneously images from two cameras. People position can be used to detect position-based events such as unattended luggage, intrusion in forbidden areas, falls on tracks, etc. Further analysis using position information from consecutive frames, tracking, allows a basic analysis of people interaction and the detection of dynamic-based events, unusual movements in passageways, vandalism, attacks, etc. The following paragraphs shows some examples of how event detection can be achieved using the position of the centroid, the characteristics of the blob and the tracking information, [14]. A low people-density situation is assumed.

(1) Unattended luggage. A person carrying luggage leave it and move away:
- Initial blob splits in two
- One (normally smaller and horizontal) presents no motion, the other moves away from the first.
- Temporal thresholding may be used to trigger the alarm.

(2) Falls.
- Blob wider than tall.
- Slow or no centroid motion
- Falls or tracks: centroid in forbidden area.

(3) People hiding. People hide (from the camera-from other people)
- Blob disappearing in many consecutive frames
- Last centroid's position no close to a 'gate' (to leave the scene)
- Last centroid's position very close to a previously labelled 'Hiding zone'
- Temporal thresholding may be used to trigger the alarm.

(4) Vandalism. People vandalising public property:
- Isolation: only one person/group present in the scene
- Irregular centroid motion
- Possible change in the background afterwards

(5) Fights. People fighting move together and break away many times, fast movements:
- Centroids of groups or persons move to coincidence
- Persons/Groups merging and splitting
- Fast changes in blobs characteristics

Frame by frame blob analysis and tracking provide enough information to detect some of the previous events and the possibility of others. This event detector is always

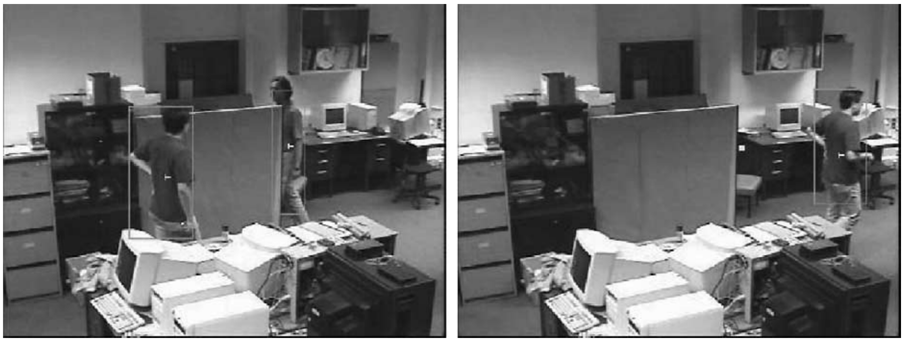


Fig. 4. Event detection: when a blob vanish away from any area where people can leve on enter the scene, the system marks the last position and raise a hidding event.

working and raise the alarm when the above conditions are fullfilled for one or more of the blobs under tracking. For instance, when a tracked person disappears from the image for more than a previously defined number of frames (may be just one for blobs or scenes with no segmentation problems) and the last centroid position is not near a gate -entry/exit part of the image-, a Hidding event may be raised, and the last point seen marked in the image, Fig. 4. In any case, the system only attracts the attention of the operator, who always decides whether an event is actually taken place.

5. Discussion

The system has to be a real alternative to existing CCTV instalations with up to hundred cameras per control room. In order to minimize costs, each computer has to be able to process video streams from at least four cameras. Therefore, a high processing speed is essential. Luminance contrast segmentation and its associated background model have been chosen because they provide an excellent performance with lower computational cost. Some important points concerning the influence of the chosen method in background subtraction and tracking are discussed below.

5.1. Illumination

There are always variations in the illumination parameters between two images of the same scene taken on different days, and even at different times of day. However, indoor backgrounds provide a relatively stable lighting configuration whereby variation is normally due to a lamp replacement or momentary lamp failure. There are many other factors, such as changes in voltage or obstruction of reflected light that can lead to minor illumination changes but their effect on the general illumination level is relatively small. These minor modifications produce a global shift in the contrast plot, with the ‘background contrast’ moving from zero to positive or negative values depending on whether the new scenario is darker or lighter than the stored background image. The observed shifts were not bigger than 15% and the selection of an appropriate contrast threshold can deal with these small illumination changes; Fig. 5(a) shows how the luminance contrast plot reflects a minor illumination variation. In outdoor applications these light variations are normally bigger, but any well-known background-updating algorithm should be able to deal with them [4–7,11].

5.2. Threshold dependence

For well-contrasted objects, according to visual standards those with luminance contrasts above 0.5, the segmentation algorithm is barely affected by the chosen contrast thresholds, Fig. 6(a). For low-contrasted object, the algorithm proved itself stable under threshold modifications, normally avoiding shadows with threshold values around 0.2. This value appears as good default value for all scenarios

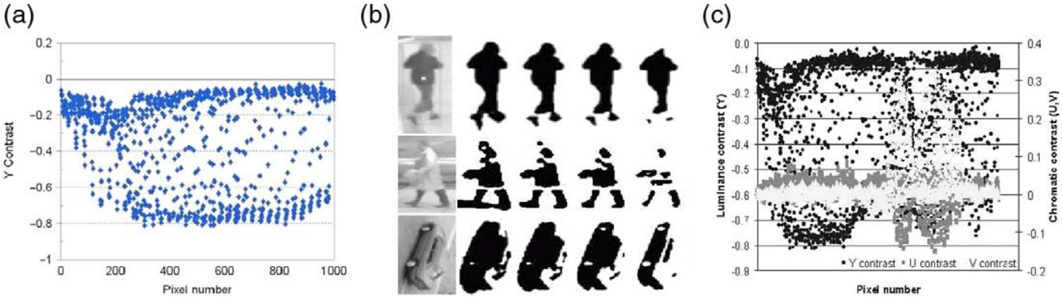


Fig. 5. (a) Luminance contrast plot showing illumination variation; (b) Effect of contrast threshold values −0.10, 0.15, 0.20 and 0.40- on foreground detection; (c) colour contrast plot.

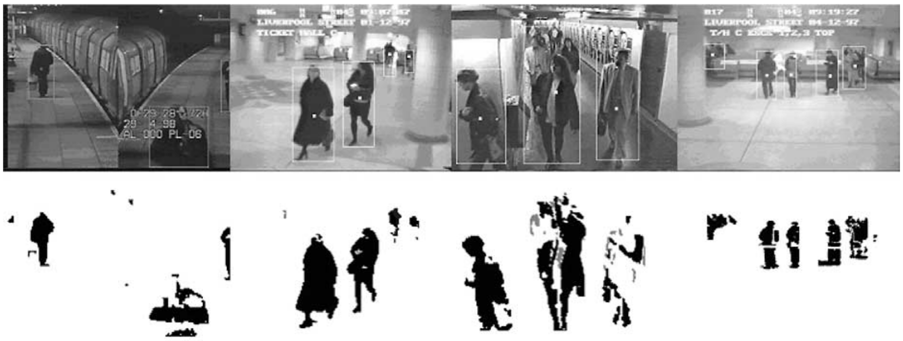


Fig. 6. Four examples of foreground detection and blob selection.

analysed, some examples are shown in Fig. 5(b). It is large enough to be unaffected by changes in lighting up to 20%, when the background is not updated, and the false positives rate, after cleaning and object detection, is lower than 5%.

5.3. Colour information

Disregarding colour information in foreground detection improves the speed of the processing with a minimal loss of information. This is especially true in the kind of images we are analysing here. Indoor environments are normally badly illuminated in terms of the sensitivity requirements of standard colour cameras used in CCTV surveillance systems. This fact produces poor colour reproduction. Together with the fact that people in northwestern countries wear, mostly, dark and plain clothes, colour information is not significant in most of the cases [13]. Colour information may always be used in a later stage to improve segmentation once the blob have been detected. Chromatic contrast can provide better results when luminance contrast is very low (isoluminance or colour contrast without luminance contrast is a very rare condition) or a joint analysis of chromatic and luminance contrasts helps to discriminate between luminance contrast produced by shadows and contrast produced by object borders [13].

5.4. Tracking algorithm

When two blobs merge forming a group, their information is kept but it is the group blob that is tracked through the following frames. That means that information about the centroid of individual blobs is not available while they are part of a group. For a more general application, the predicted position, using the stored values of position and velocity, may be used to complete the trajectory of the tracked blob while grouping with others. In this way, a temporally consistent list of blobs is kept together with their trajectories and their positions. During occlusions, the individual blobs merged into the group are always supposed to be forming that group. Therefore, an assumption has to be made: objects cannot disappear from the scene unless they exit through predefined borders. These borders are defined as image zones through which objects can leave or enter the scene. In this way, people hiding or appearing and objects left and picked up can be detected. However, this assumption has a strong dependence upon the foreground detection method, which has to be solid enough not to lose a blob due to its low contrast. Normally this means a lower contrast threshold, which has the effect of adding the shadow of the object to the foreground pixels leading to a less accurate positioning of the blob's centroid.

5.5. Results

According to PETS 2001 requirements, trajectories of tracked objects in image plane (2D) were provided in the required XML files. As pointed out before, the system does not store the centroid of the tracked object when it merges into a group and, therefore, only single object trajectories were provided. The detection of some events,

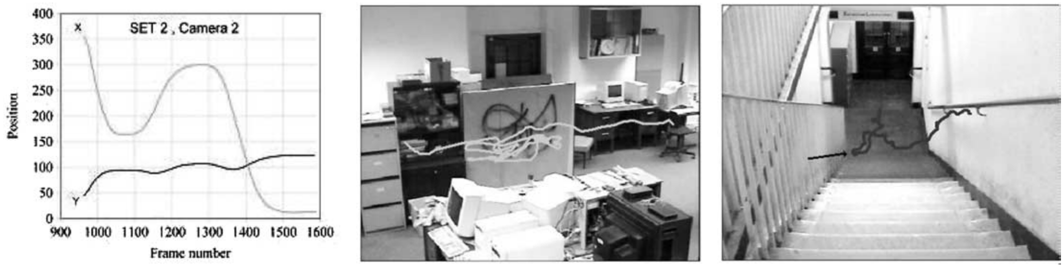


Fig. 7. Tracking and event detection results. From left to right, XY plot of the trajectory of a cars centroid on the rightmost image in Fig. 4 and centroid trajectories of people making graffiti and falling on the stairs.

falls on stairs and vandalism, is illustrated with the superimposed trajectories, Fig. 7.

6. Conclusions

The presented real-time tracking system was implemented on an 850 MHz compatible PC running Windows 2000. It works with colour images in half PAL format 384×288. It has been tested with live video and image sequences in BMP and JPEG formats. The minimum processing speed observed is 10 Hz, from disk images in BMP format. Working with a video signal there is no perceptible difference between processed and un-processed video streaming. The system can successfully resolve blobs forming and dissolving groups and track one of them throughout this process. It also can be easily upgraded with background updating, for outdoors applications, and tracking of multiple objects. Finally, the system shows it can be used in surveillance application to detect some predefined events - like graffiti scribbling or unattended luggage-, and therefore, helping existing systems with security and monitoring tasks. The event detector, however, still lacks of testing using an unsupervised video stream to check the detection and false positive ratios. We are currently refining it and planning the run of such tests.

References

[1] L.M. Fuentes, S.A. Velastin, Assessment of Digital Image Processing as a means of Improving personal security in public Transport, Proceedings of the Second European Workshop on Advanced Video-based Surveillance Systems, (AVBS2001) (2001).

[2] S.S. Intille, J.W. Davis, A.F. Bobick, Real-Time Closed-World Tracking, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97) (1997) 697–703.

[3] F. De la Torre, E. Martinez, M.E. Santamaria, J.A. Moran, Moving Object Detection and Tracking System: a Real-Time Implementation, Proceedings of the Symposium on Signal and Image Processing GRETSI 97, Grenoble (1997).

[4] I. Haritaoglu, D. Harwood, L.S. Davis, W4: Real-time surveillance of people and their activities, IEEE Transaction Pattern Analysis and Machine Intelligence 22 (8) (2000) 809–822.

[5] S. Huwer, H. Niemann, Adaptive Change Detection for Real-Time Surveillance Applications, Proceedings of the IEEE Workshop on Visual Surveillance, Dublin (2000) 37–43.

[6] N. Rota, M. Thonnat, Video Sequence Interpretation For Visual Surveillance, Proceedings of the IEEE Workshop on Visual Surveillance, Dublin (2000) 59–68.

[7] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, H. Wechsler, Tracking groups of people, Computer Vision and Image Understanding 80 (2000) 42–56.

[8] C.R. Wren, A. Azarbayejani, T. Darrel, P. Pentland, Pfinder: real-time tracking of the human body, Transactions Pattern Analysis and Machine Intelligence 17 (6) (1997) 780–785.

[9] J.K. Aggarwal, Q. Cai, Human motion analysis: a review, Computer Vision and Image Understanding 73 (3) (1999) 428–440.

[10] D.M. Gavrila, The visual analysis of human movement: A survey, Computer Vision and Image Understanding 73 (1999) 82–98.

[11] R. Rosales, S. Claroff, Improved Tracking of Multiple Humans with Trajectory Prediction And Occlusion Modelling, Proceedings of the IEEE Conference on Computer Vision and Pattern Recodgnition (1998).

[12] J.I. Agbinya, D. Rees, Multi-object tracking in video, Real-Time Imaging 5 (1999) 295–304.

[13] L.M. Fuentes, S.A. Velastin, Foreground Segmentation using Luminance Contrast, Proceedings of the WSES/IEEE Conference on Speech, Signal and Image Processing, Malta (2001) 2231–2235.

[14] L.M. Fuentes, Assessment of image processing techniques as a means of improving Personal Security in Public Transport. PerSec, EPSRC Internal Report, April 2002.