People tracking in surveillance applications

Luis M. Fuentes and Sergio A. Velastin

Department of Electronic Engineering
King's College London (University of London)
London WC2R 2LS, UK
luis.fuentes@computer.org, sergio.velastin@iee.org

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. Results from indoor and outdoor scenarios are shown.

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 people.

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 [1], "Assessment of Image Processing Techniques as a means of Improving Personal Security in Public transport". It was originally designed to work with CCTV footage from London Underground stations (indoors) placing more emphasis on 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. Results on both indoor and outdoor tracking are presented.

2. Related work

Foreground detection algorithms are normally based on background subtraction algorithms (BSAs) [2,3,4], 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 [6,7.8], 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 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 3x3 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, [10]. 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 [10,11]. Some are applied in relatively controlled [3,8,12] 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 [9] 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 [12]. The use of blob trajectory [12] 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 luminances of the object, L_O, and the surrounding background, L_B

$$C_L = \frac{L_O - L_B}{L_B} \qquad (1)$$

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:

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

Luminance values are in the ranges [0,255] for images digitised in YUV format or [16,255] for images transformed from RGB coordinate. 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. An example is shown in Figure 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 usually obtained.

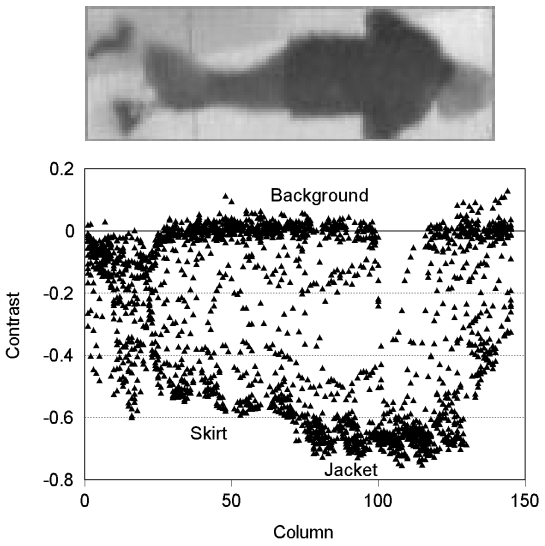


Figure 1. Values of luminance contrast for individual pixels

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. To simplify the discussion, we assume from now onwards a single contrast threshold C, that is C_P = - C_N = C. 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.

4. 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. 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 were always 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 while forming a group.

4.1. 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}.

$$\begin{aligned} M_{t-1}^{t}(i,j) &= \text{Matching}\left\{B_i(t-1), B_j(t)\right\} \\ M_{t}^{t-1}(i,j) &= \text{Matching}\left\{B_i(t), B_j(t-1)\right\} \end{aligned} \quad (3)$$

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*.

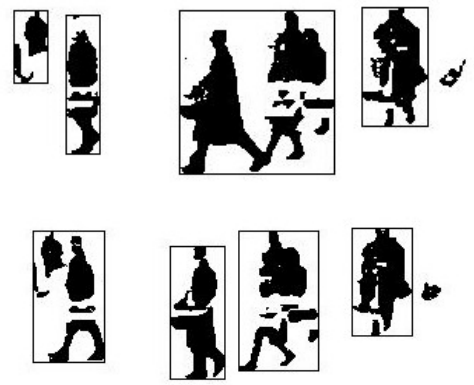
$$S_{t-1}^{t}(i) = \bigcup_j j \quad such\,that \quad M_{t-1}^{t}(i,j) = 1 \quad (4)$$

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

4.2. 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 matrices 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, Figure 3.

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 and the information about the two merged blobs is stored for future use, keeping the tracking after splitting, as shown in the example given in Figure 4.



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

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

Figure 2: An example of detected blobs in two consecutive frames the matching matrices and strings

Merging : $B_i(t-1) \bigcup B_j(t-1) \equiv B_k(t) \quad \Leftrightarrow \quad \begin{cases} S_{t-1}^{t}(i) = S_{t-1}^{t}(j) = k \\ S_t^{t-1}(k) = i \bigcup j \end{cases}$

Splitting : $B_i(t-1) \equiv B_j(t) \bigcup B_k(t) \quad \Leftrightarrow \quad \begin{cases} S_{t-1}^{t}(i) = j \bigcup k \\ S_t^{t-1}(j) = S_t^{t-1}(k) = i \end{cases}$

Entering : $B_i(t) \equiv New \quad \Leftrightarrow \quad \begin{cases} S_{t-1}^{t}(j) \neq i \quad \forall j \\ S_t^{t-1}(i) = \varnothing \end{cases}$

Leaving : $B_i(t-1) \equiv Leaves \quad \Leftrightarrow \quad \begin{cases} S_{t-1}^{t}(i) = \varnothing \\ S_t^{t-1}(j) \neq i \quad \forall j \end{cases}$

Correspondence : $B_i(t-1) \equiv B_j(t) \quad \Leftrightarrow \quad \begin{cases} S_{t-1}^{t}(i) = j \\ S_t^{t-1}(j) = i \end{cases}$

Figure 3: Correspondence between some events in the temporal evolution of the blobs and the matching strings.

5. Discussion

Due to final system requirements, 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. In outdoor applications these light variations are normally bigger, but any well-known background-updating algorithm should be able to deal with them [4,5,6,7,12].

5.2. Effect of 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. 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.

5.3. 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.

5.4. Tracking results

Trajectories of tracked objects in image plane (2D) are provided along with the required XML files. An indoor scenario with an abnormal trajectory detected by the system and two cars in two sequences in PETS 2001 data set 2 - it presents small background variations- have been chosen. As pointed out before, the system does not store the centroid of the tracked object when it merges into a group. Thus only single object trajectories are provided, see Figure 5.

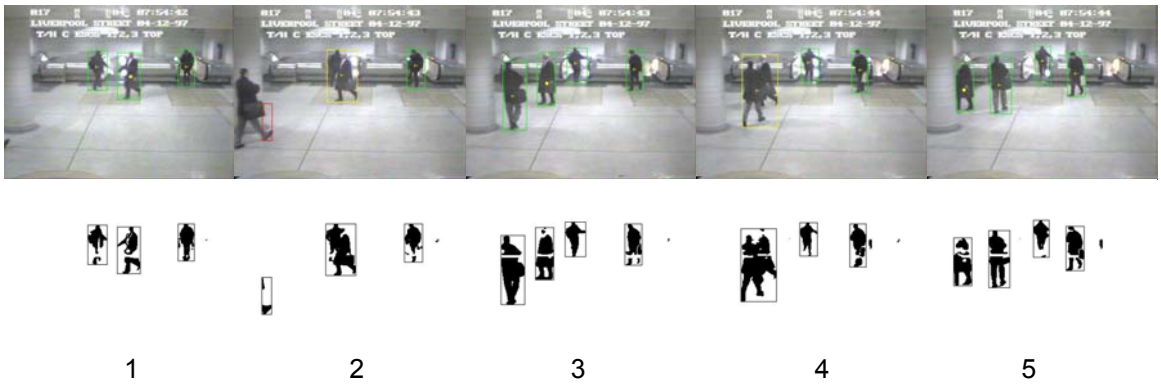


Figure 4: Tracking example, movie 1. The algorithm is tracking the man in the middle in frame 1, marked as tracked with a dot, through two consecutive grouping processes. A tracking example involving a 3-person group is shown in movie 2.

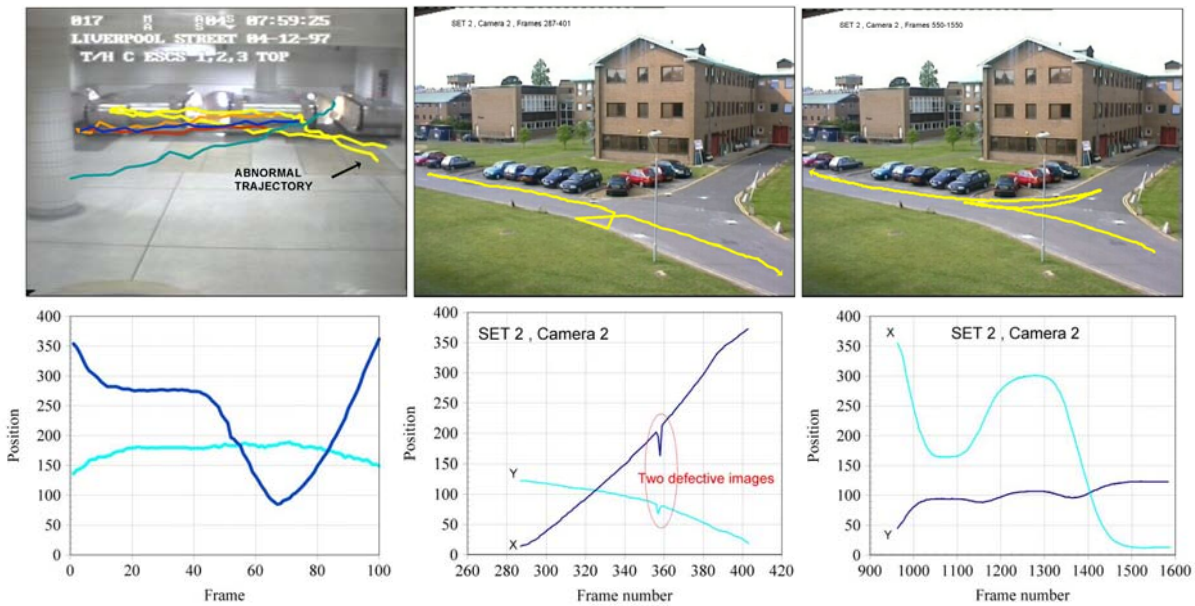


Figure 5: Examples of trajectories of the centroid of the bounding box. An abnormal trajectory in a London Underground station and trajectories of two cars in PETS 2001 Image set 2, XML files 1 and 2.

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 384x288. 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 and tracking of multiple objects.

7. Acknowledgements

The work reported here has been carried out as part of UK's Engineering and Physical Sciences Research Council project "PerSec: Assessment of image processing techniques as a means of improving personal security in public transport" in collaboration with the Centre for Transport Studies, University College London. The authors are also grateful to London Underground Limited for access to their sites and advice.

8. References

[1] L.M. Fuentes and S.A. Velastin, "Assessment of Digital Image Processing as a means of Improving Personal Security in Public Transport", *Proceedings of the 2nd European Workshop on Advanced Video-based Surveillance Systems*, September 2001.

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

[3] F. De la Torre, E. Martinez, M. E. Santamaria and 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 and L.S. Davis, "W4: Real-Time Surveillance of People and Their Activities", *IEEE Trans. Pattern Analysis and Machine Intelligence,* **22**(8), 2000, pp. 809-822.

[5] S. Huwer and H. Niemann, "Adaptive Change Detection for Real-time Surveillance Applications", Proceedings of The IEEE Workshop on Visual Surveillance, Dublin, 2000, pp. 37-43.

[6] N. Rota and M. Thonnat, "Video Sequence Interpretation for Visual Surveillance", Proceedings of The IEEE Workshop on Visual Surveillance, Dublin, 2000, pp. 59-68.

[7] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler, "Tracking Groups of People", *Computer Vision and Image Understanding* **80**, 2000, pp. 42-56.

[8] C. R. Wren, A. Azarbayejani, T. Darrel and P. Pentland, "Pfinder: Real-Time Tracking of the Human Body", *Trans. Pattern Analysis and Machine Intelligence,* **17**(6), 1997, pp. 780-785.

[9] J.I. Agbinya and D. Rees, "Multi-Object Tracking in Video", *Real-Time Imaging* **5**, 1999, pp. 295-304.

[10] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review", *Computer Vision and Image Understanding,* **73**(3), 1999, pp. 428-440.

[11] D. M. Gavrila, "The visual analysis of human movement: A survey", *Computer Vision and Image Understanding,* **73**, 1999, pp. 82-98.

[12] R. Rosales and S. Claroff, "Improved Tracking of Multiple Humans with Trajectory Prediction and occlusion Modelling", *Proceedings of the IEEE Conf. On Computer Vision and Pattern Recognition,* 1998

[13] L.M. Fuentes and S.A. Velastin, "Foreground segmentation using luminance contrast", *Proceedings of the WSES/IEEE Conference on Signal XXXX and Image Processing,* September 2001.