第29卷 第3期 自 动 化 学 报 Vol. 29, No. 3
2003年5月 ACTA AUTOMATICA SINICA May, 2003

Occlusion Tolerent Tracking Using Hybrid Prediction Schemes[1]

E. Corvee S. Velastin G. A. Jones
(Digital Imaging Research Centre, Kingston University, Kingston, U. K.)
(E-mail: {E. Corvee, Sergio. Velastin, G. Jones}@Kingston. ac. uk)

Abstract A method of combining multiple moving objects prediction schemes is presented that allows a tracking framework to select and identify the best observation evidence in occlusion scenarios. The underlying framework tracks any objects in monocular image sequences taken from stationary uncalibrated cameras with fixed focal length. A mixture model method is deployed to estimate the static background reference image. The tracking algorithm simply uses a constant acceleration motion model to track objects in the simplest scenarios. However, the main contribution is the use of three simultaneous predictors with a least square correlation stage to select the most likely object position. The three prediction schemes are an $\alpha-\beta$ tracking scheme, a Kalman filtering method, and a region segmentation and matching method. The tracker is evaluated against different image sequences each offering different occlusion problems.

Key words Hybrid prediction, least square correlation, occlusion

1 Introduction

The presented work is part of the PRISMATICA[1] project which is concerned with automatic visual surveillance of public transport environments by image processing methods. PRISMATICA aims to integrate technical systems and operational processes to develop innovative security management systems for transport operators, and to develop and integrate tools to automatically detect a range of events, such as violence, trespass, congestion, fires, suspect packages and suspect individuals, theft, vandalism, ticket fraud. As such, the project contributes to more general efforts to make public transport systems more attractive to passengers, and more secure both for passengers and staff. The project specifically aims to assess the pedestrian behaviour in stations: an event is triggered when a pedestrian or object remains stationary in unusual locations or for an unusual period of time, or when the scene is congested or overcrowded.

By far the most common approach to monitoring scene objects in typical mid-range surveillance imagery uses pixel differencing to detect moving regions in static scenes[1], blob analysis to extract observations of moving objects, and trajectory tracking to establish the temporal history of individual scene events. The most significant challenge to this otherwise successful approach is the frequent problem of occlusion and fragmentation where the shape, dimensions and colour signature of the merged or fragmented observations do not correlate well with the actual object observation. This can lead to either loss of correspondence or mis-association particularly where the projected 2D width and height are also derived from the dimensions of the observations. Such problems are usually addressed by embedding appearance models which improve tracking accuracy by comparing the width and height, shape or colour of observations with a model of the object.

A number of contributions to the tracking knowledge are made in this work. Primarily, a hybrid tracking scheme is presented which integrates several sources of prediction: i) an $\alpha-\beta$ and ii) a Kalman filter operating on a linear motion model, and iii) an appearance

[1] Supported by the EU IST ADVISOR Project 'Annotated Digital Video for Surveillance and Optimised Retrieval' (IST-1999-11287)

[2] http://www. prismatica. com/

Received December 2, 2002; in revised form April 8, 2003
收稿日期 2002-12-02; 收修改稿日期 2003-04-08

model which is used to predict the most likely position in the next frame given the appearance of the object in the last. In addition, occlusion handling rules are introduced to prevent the motion tracker being confused during occlusion processes. Finally a novel but computationally efficient connected components-like technique is described which extracts objects (or blobs) from the motion detection image.

Overview of Approach

The tracking algorithm tracks blobs from frame to frame and consists of performing the following consecutive steps for each new current frame.

Maintenance of Reference Image Mixture of Gaussian methods based on Staffer and Grimson[1]—see section 3. 1.

Moving Object Detection Detection of foreground pixels (see section 3. 2), and extraction of candidate regions of moving objects i. e. blobs—see section 3. 3.

Blob Tracking—see section 4. Tracking algorithm is decomposed in the following steps:

1) Prediction: Each prediction method is employed to identify the most likely location of the appropriate observation in the current frame.

Method A A Constant Acceleration $\alpha-\beta$ Tracker—see section 4. 2.

Method B Region Segmentation and Matching—see section 4. 3.

Method C Constant Velocity Kalman Tracker—see section 5.

2) Matching: Each of these prediction schemes returns i) a blob representing the best match from the set of moving regions in the current image, and ii) an estimate of the position of the current blob. The best is evaluated using a cross-correlation method comparing pixels from the previous and current moving regions—see 4. 1.

3) Updating: The motion models are updated using the locations of the returned observations.

Creating New Objects Detection and creation of the new tracks for the new blobs in the scene which have not been matched to any previous blob.

2 Related work

In most systems the first step in tracking objects is to separate the foreground from the background or to detect motion *i. e.* to detect the regions (apparent shape) of independently moving objects regardless of their speed, direction or texture. The majority of the established frameworks track objects against a background captured from a single and stationary CCD camera with fixed focal length, as in this study. We build a background reference image using a mixture of Gaussian models. Then the foreground objects are segmented from the background reference image by using a simple thresholding method on a luminance contrast criteria. The implementation of the foreground object segmentation is presented in more details in section 3.

Most of the tracking methods that use pixel-differencing employ either a mixture of Gaussian models or a Kalman filtering method to model the background scene. Stauffer and Grimson [1] have introduced the concept of multi-Gaussian mixture model and it is widely used, e. g. [2~4]. The approach uses EM (Expectation Maximisation) to fit a Gaussian mixture model efficiently to each incoming pixel stream algorithm to determine which mixture model is the most likely to result from the background process. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively is considered part of the background model. This technique has for example been very successful in vehicle identification and tracking[5] in which tracking has been facilitated by the fact that the pixels of foreground objects are modelled as the weighted sum of three selected distributions: the road, shadows and vehicle. The other major technique to model the background image is the Kalman filtering based method[6]. As in the mixture model method technique, the Kalman filtering technique will also adapt to the changing illumination occurring in the background. An interesting alternative way to estimate the background is presented in [7] where Robust Statistical filters model the background pixels. The approach is using L-filters (i. e. a linear combination of the ordered samples of the image sequence).

The Tracker module is implemented within a hypothesize, validate and update framework—see Figure 1. Each active scene Object has an associated Trajectory Model which describes the current position, velocity and possibly acceleration of the object in image coordinates. (An alternative ground plane space may be a more appropriate space within which to track[4]). In addition, each Object has an associated Appearance Model which may be used to identify those blobs with the most similar shape and/or chromatic structure. Such appearance models may simply describe the expected width and height of the object's bounding box, or record the pixel greylevels within the last bounding box. More sophisticated models may record the contour[8], binary pixel shape[9,10], or spatio-chromatic structure[11,12]. More dynamic variants of the appearance model may simply describe the rate of change of these bounding box dimensions[13,14] while active or statistical appearance models may attempt to learn the allowable variation in object appearance[15,16]. In the hypothesis phase of the procedure, the object position and appearance are predicted from the Trajectory and Appearance models. Each active scene object is then validated by locating an appropriate corresponding observation from the list of candidate observations—Data Association[17]. Greedy matching is a common local approach to establishing correspondences in which the observation closest (using the Mahalanobis distance metric) to the predicted position of an object is selected. In addition to incorporating appearance information, more sophisticated global approaches attempt to enforce the uniqueness constraint by considering all possible object-observation pairings[18,19]. Unmatched observations may be used to hypothesise new objects appearing within the scene. In the update phase, the position and appearance of each corresponding observation are used to update the trajectory and appearance model of validated scene objects. Typical update mechanisms include the $\alpha-\beta$ filter and the Kalman filter. Fundamentally, the tracker maintains the temporal coherence of object identities.

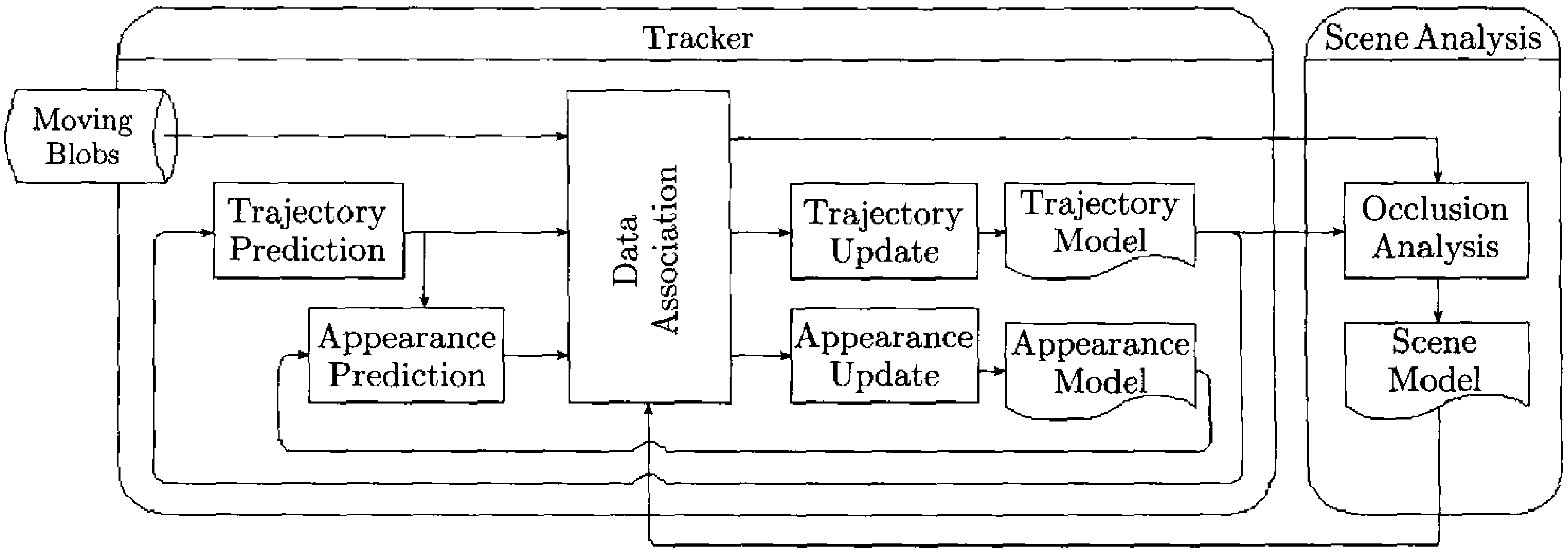

Fig. 1 Tracking architecture

For a tracking algorithm to be successful, it has to be robust in all scenarios, including the specific case of occlusion. All existing trackers can only cope with moderate levels of occlusion, and most of them cannot cope well when moving objects leave the group of merging objects in different directions in which they entered. The longer an object merges into a group, the more difficult it is to be tracked. Some system can have additional constraint by assuming for example to have the objects moving on a ground plane only; enabling a depth ordering and a better estimation of the tracks. While the use of Appearance Models such as shape or chromatic texture models are vital to establish temporal coherence of object identity, robust real time implementations are not currently available for even the typically high-specification computing platforms used in visual surveillance research. Sometimes the tracking of objects through inter-frame correspondence of features breaks down, because of significant shape feature variations or due to occlusion of objects by one another. Different solutions to the occlusion problem in tracking have been proposed. Rosales and Sclaroff[1] for example use Kalman filter, Khan and Shah[20] segment object into similar colour classes and Colins *et al.* use the normalised colour histogram of each objects[21]. In Anzalone and Machi two combined methods are applied[22] (the first method uses a mixed parametric and fuzzy logic approach to compute distances among objects in the features space and to assign to each association and affinity index. The second method is based on a Kalman filter approach). Recently, Haritaoglu *et al.* implemented a real-time human-tracking system and suggested using a multi-camera system to analyse the occlusions[23]. In here single static camera are used and several methods are combined altogether to predict the best estimate. These techniques compose of grey level segmentation, geometry match and a simple Kalman filter. Where implemented occlusion reasoning stages can consider the longer term history of each track to appropriately introduce merge and split operations and re-establish correspondence caused by occlusion or fragmentation[2].

3 Segmenting moving regions

In most systems the first step in tracking objects is to separate the foreground from the background or to detect motion. This means to detect the regions (apparent shape) of independently moving objects regardless of their speed, direction or texture. Moving objects are assumed to occlude a background captured from a single and stationary CCD camera with fixed focal length. In common with most current implementations, we build a background reference image using a mixture of Gaussian models[1]. Foreground objects are then segmented from the background reference image by using a simple thresholding method on a luminance contrast criteria.

3.1 Building a reference image

Stauffer and Grimson[1] model the values of each pixel as a mixture of a Gaussian. Based on the persistence and the variance of each of the Gaussians of the mixture, they determine which Gaussians may correspond to the background grey level. Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it.

The system adapts to deal robustly with lighting changes, repetitive motions of scene elements, tracking through cluttered regions, slow moving objects, and introducing or removing objects from the scene. Slowly moving objects take longer to be incorporated into the background. Also, repetitive variations are learned, and a model of the background distribution is generally maintained even if it is temporarily replaced by another distribution which leads to faster recovery when objects are removed. This method requires two significant parameters: the learning constant and the proportion of the data that should be accounted for the background.

If each pixel resulted from a particular surface under particular lighting, a single Gaussian would be sufficient to model the pixel value while accounting for acquisition noise. And if only lighting changed over time, also a single, adaptive Gaussian per pixel would be sufficient. In practice, as previously explained, multiple surfaces often appear in the view of a particular pixel and the lighting conditions change. Thus, multiple, adaptive

Gaussians are necessary——typically 5.

3.2 Foreground pixel detection

New moving objects are located by comparing each new frame against a reference image that contains only stationary objects. Foreground pixels detection is achieved in this project by using a luminance contrast measurement[24]. 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, C, is defined as the relative difference between object luminance, I_c, and local background luminance, I_b.

$$C(i,j) = \frac{I_c(i,j) - I_b(i,j)}{I_b(i,j)} \tag{1}$$

where (i, j) is the pixel location in the image plane. Values of luminance around 0 are expected for background pixels while negative and positive values are expected to occur at foreground pixels with brighter and darker intensities respectively. A threshold is used to either classify the pixels as foreground or background. An empirical value of the order of 0.15～0.20 is chosen: if the absolute value of the luminance contrast is greater than this threshold then the pixels are labelled as foreground pixel, else as background. An example of foreground/background objects pixels are shown in Figure 2.

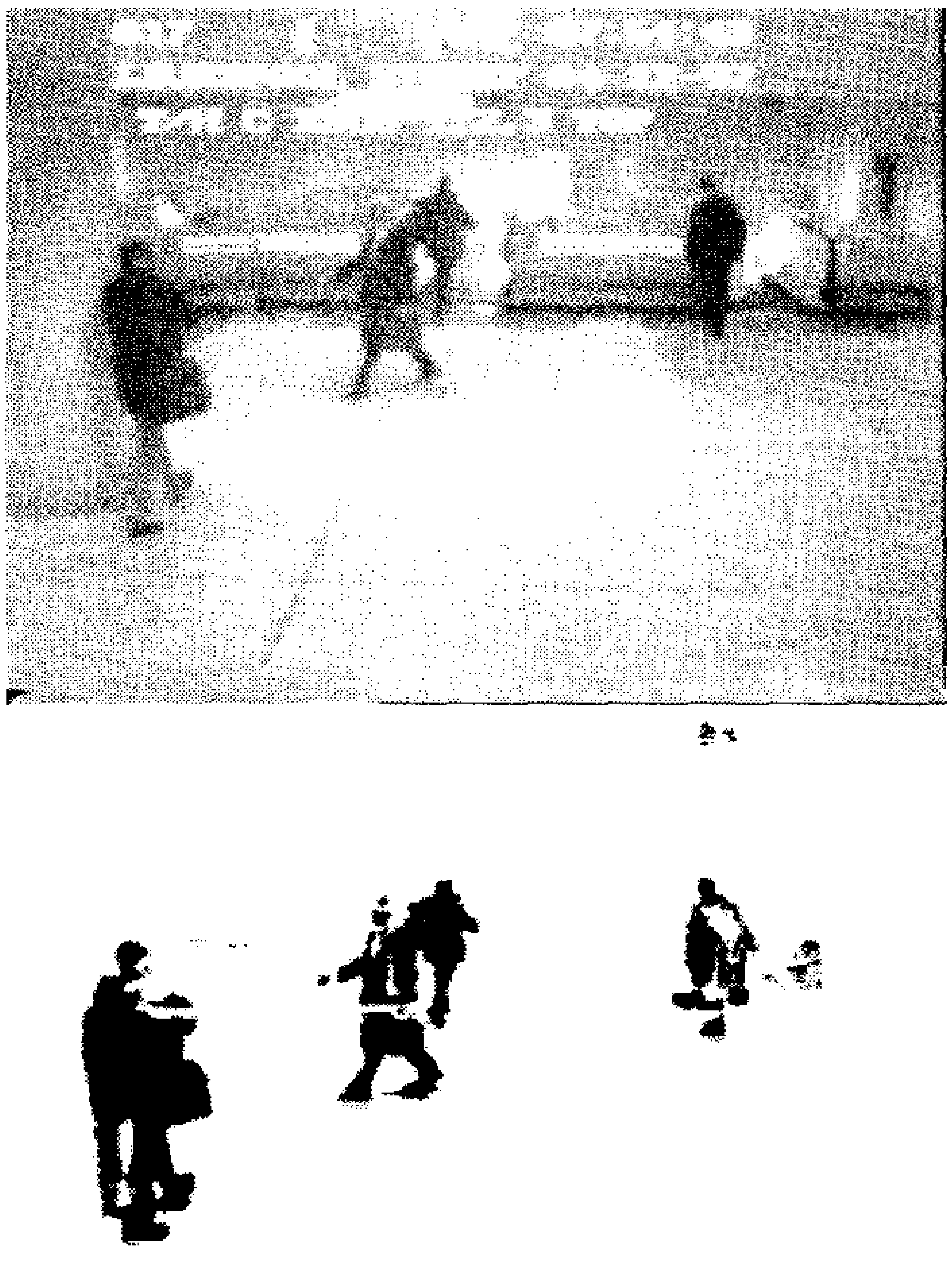

Fig. 2 Detected foreground pixels

3.3 Extracting moving regions

After all pixels have been flagged as either foreground or background from the foreground detection module, the region finding module processes them in order to create blobs of the foreground objects. The task of separating foreground pixels into different blobs is achieved using a projected histogram method on the current foreground data[24]. The algorithm searches along the histogram all the possible spatial segments where group of foregrounds exist, where each segment create a blob area as shown in Figure 3. This procedure of histogram followed by blob formation is firstly achieved on the horizontal axis on one single area, the frame formed by the whole image. Once a first set of blobs is created the same procedure is repeated on the vertical axis and on each blob area delimited in the previous procedure. This will lead to a more detailed blobs segmentation. Finally for more accuracy and to segment each possible group of foreground pixel into blobs, an additional horizontal and vertical histogram formation is achieved on each updated blob areas. In order to reduce false foreground object segmentation, each blob should contain at least 30% of foreground pixels and the bounding box should have a height and width of minimum pixels, of the order of 5. Examples of segmented blobs in metro scenes are shown in Figure 4.

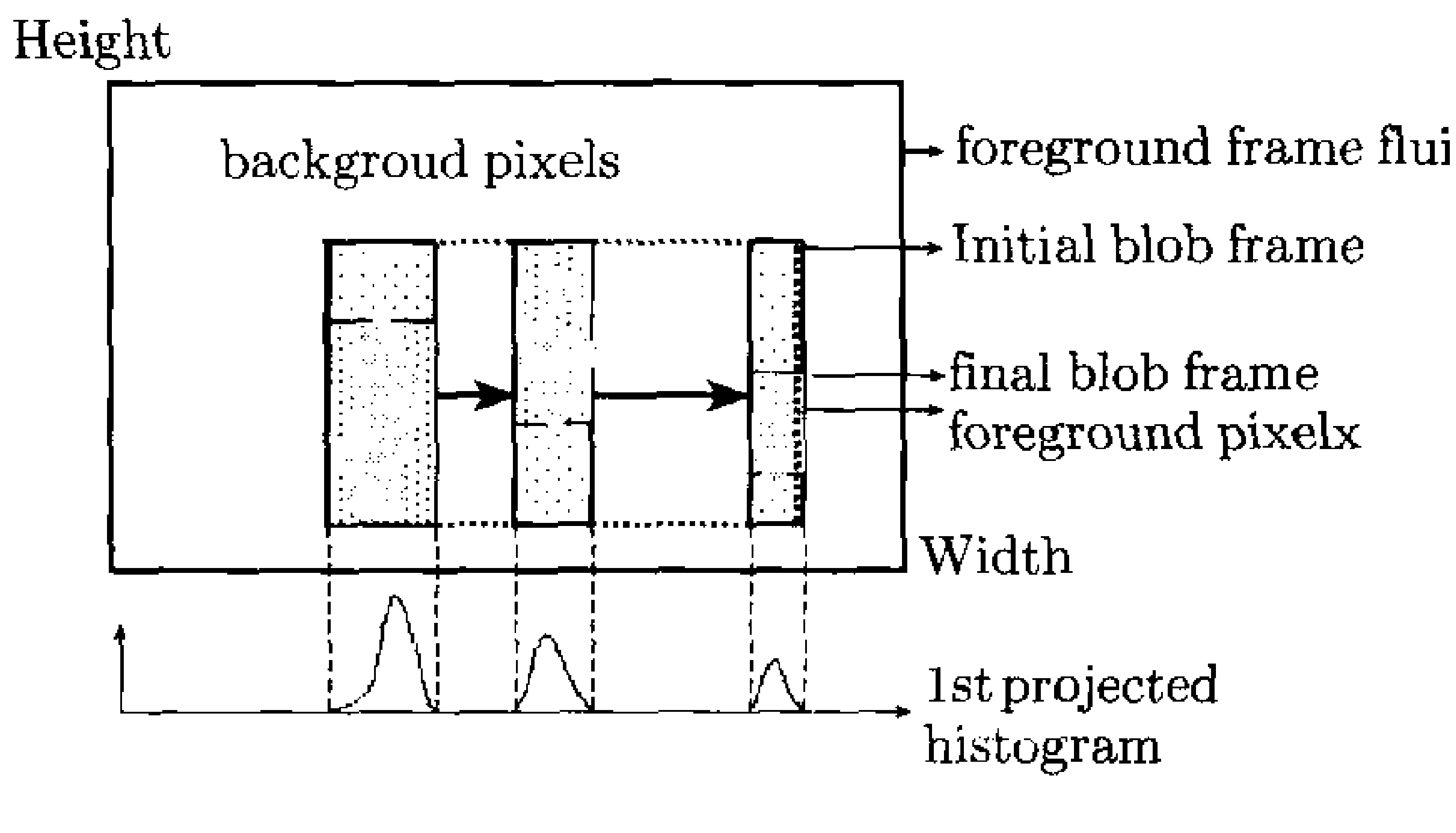

Fig. 3 Finding blob frames

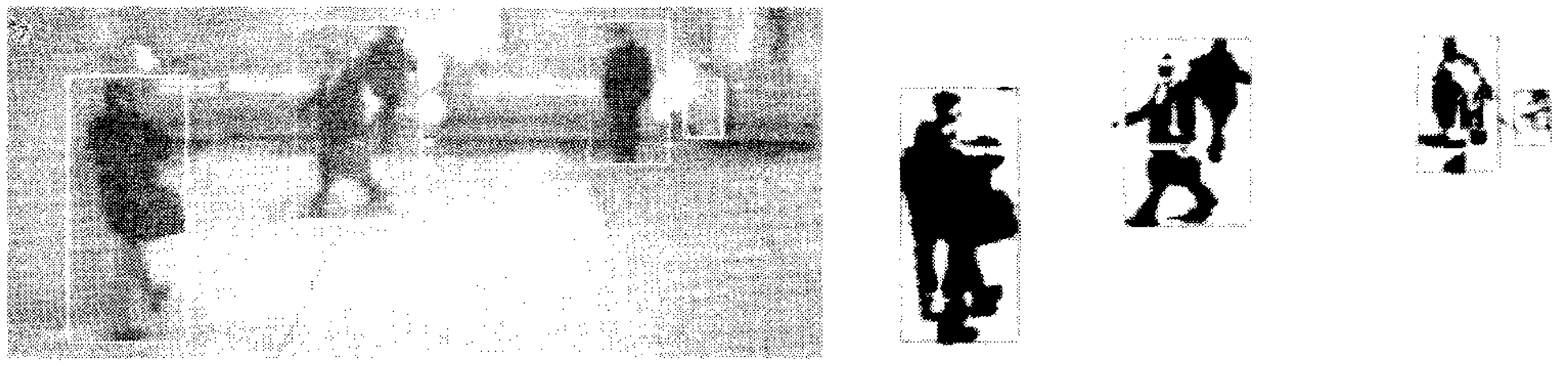

Fig. 4　Detected foreground blobs

4　Blob tracking

The general aim of a tracking algorithm is to establish the temporal history of an object with reference to the set of feature observations (*i. e.* blobs) extracted from the image sequence over time——ideally from the moment each blob appears in the scene until it disappears. The most significant challenge to this otherwise successful approach is the frequent problem of occlusion and fragmentation where the shape, dimensions and colour signature of the merged or fragmented observations do not correlate well with the actual object observation.

4.1　Occlusion reasoning

The tracking algorithm developed here tracks the blobs between two successive frames by detecting the overlapping bounding boxes between the new untracked detected blobs, given by the current frame, and the predicted objects from previous frame. Many scenarios can occur: one blob occluded by another, merging of many blobs into a single one, blob disappearance (due to occlusion by background objects or simply due to foreground detection failure). In this tracking algorithm the following rules have been chosen:

Correspondence between predicted object position and candidate blob from current image is signalled by overlap of bounding boxes.

Ambiguity If a predicted blob simultaneously intersects with two current blobs then the current blob with the largest intersection area will be considered as the best candidate.

Occlusion If the bounding boxes of several predicted objects overlap one single current blob then the blob is assigned to object which returns the best correlation match—see below.

Match Testing To assess the quality of a match, the pixels within the bounding box of the previous blob is projected into the current image and compared with the corresponding pixels in the current image using a squared error measure. Alternate predictions are differentiated by seeking the most similar match and hence yield the minimum squared error.

Static Occlusion If a predicted object intersects no candidate blobs from the current frame and is predicted to remain with the image, the object is considered to be temporally occluded by some static object *e. g.* a car, sign, etc. (It is also common in situations where the pixel intensities of the predicted object are very similar to the background. Such situations may be effectively differentiated using a pre-learnt semantic landscape of the scene[2]—though this has not been implemented in this work.) In such static occlusion cases the object is continuously predicted until either it is located, exceeded an unvalidated TTL, or is expected to have left the scene.

Initialisation After attempting to match all the previous blobs with the new current blobs, if a current blob remains un-matched with a previous blob then this new blob is simply considered as a new appearing object.

4.2　Method A—trajectory prediction using $\alpha-\beta$ with constant acceleration model

Each moving object usually follows a certain trajectory which means that studying all of the positions of a tracked objects with time could allow us to predict relatively accurately where this object would be situated in the current new frame. In this study we consider that each object is expected to have an acceleration of the order of the previous acceleration computed *i. e.* constant acceleration. If the trajectory and the dynamic of each moving object were smoothly varying then a constant acceleration model would be satisfactory (it wouldn't be necessary to update the acceleration term every frame). Nonetheless acceleration needs to adapt though regularised by the previously computed acceleration.

The prediction from the previous blob into the new frame is required to locate the appropriate new blob observation from the candidate list of new blobs extracted from the current image. Consequently a predicted velocity vector is also needed. Considering an inter-frame time of unity, $\Delta t = 1$ and a constant acceleration model, we obtain the predicted acceleration term $\tilde{a}_t = a_{t-1}$ which in turn gives us then the predicted velocity $\tilde{v}_t = \tilde{a}_t + v_{t-1}$.

This predicted velocity allows the prediction of where approximately the blob situates in the new frame. Once it has matched with a new blob (if it does), this new blob repositions the now tracked blob by its new position $\boldsymbol{x}_t$, giving hence the updated new observed velocity $\boldsymbol{v}_t' = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$ and the new regularised acceleration term is:

$$\begin{aligned} \boldsymbol{a}_t &= \alpha(\boldsymbol{v}_t - \boldsymbol{v}_{t-1}) + (1-\alpha)\,\boldsymbol{a}_{t-1} \\ \boldsymbol{v}_t &= \alpha(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) + (1-\alpha)\boldsymbol{v}_{t-1} \end{aligned} \tag{2}$$

A choice of the regularising factor α of (2) makes the model independent of the previous acceleration results while a choice of 0 would make the model totally dependent on the previous result as this will represent the case of constant acceleration model. A choice of 50% for α has been chosen.

4.3 Method B-segmentation tracking

When several separate objects merge under occlusion to create a single blob then the problem of separating those object within the occluding box arises. There have been many attempts to resolve this problem. The simplest method is assume the occluding object will subsequently split and to simply wait for the merged blob to split. In the meantime the tracker suppresses the validation and update stage and blindly continues to predict the likely object position in each subsequent frame. Even where 2D height and width dimensions are stable, such an approach is obviously dependent on the occluding objects having distinct and unchanged motion trajectories—a risky strategy where visually adjacent objects are often adjacent in the real world. Neither $\alpha-\beta$ filters nor Kalman trackers are immune to this problem. The approach here relies on a more sophisticated appearance model which is matched between views. More specifically, as this appearance model is built from spatially distinct object features, it is likely that some inter-frame matching is possible during the occlusion process itself.

The i^{th} and j^{th} blobs in the previous and current frames are denoted by B_{t-1}^i and B_t^j respectively. A region segmentation algorithm divides the these previous and current blob features into uniform texture regions, $R_{t-1}^i = [\gamma_1^i, \cdots, \gamma_{N_i}^i]$ and $R_t^j = [\gamma_1^j, \cdots, \gamma_{N_j}^j]$ respectively. Each region γ is represented by two values: its mean intensity μ_γ and its standard deviation term σ_γ. An iterative grass fire-based region segmentation technique is deployed to extract uniform textured regions. Typical results are illustrated in Figure 5.

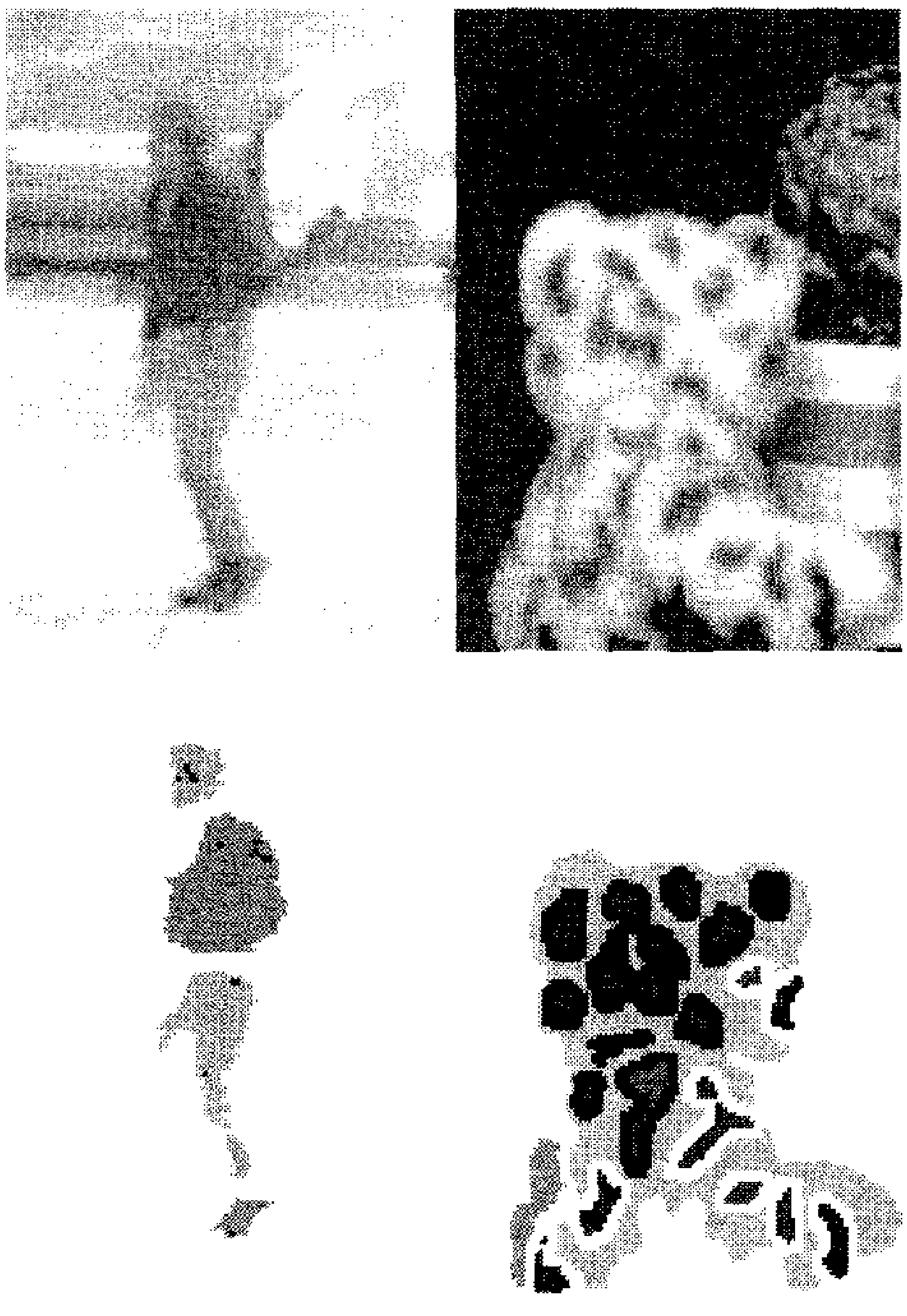

Fig. 5 Region segmentation

Once the segmentation process is achieved, a match error functional $E(\gamma_k^j, \gamma_k^i)$ which measures the dissimilarity between regions is performed for every pair of previous and current region features between R_{t-1}^i and R_t^j. The match algorithm selects the current greylevel region γ_l^j which returns the minimum match error. A region is said to be similar to another if both have similar mean and standard deviation values. Thus the error functional is defined as follows:

$$E(\gamma_k^i, \gamma_l^j) = w_\mu E_\mu(\gamma_k^i, \gamma_l^j) + w_\sigma E_\sigma(\gamma_k^i, \gamma_l^j) \tag{3}$$

where

$$E_\mu(\gamma_k^i, \gamma_l^j) = 1 - \frac{\min(\mu_{\gamma_k^i}, \mu_{\gamma_l^j})}{\max(\mu_{\gamma_k^i}, \mu_{\gamma_l^j})}; \quad E_\sigma(\gamma_k^i, \gamma_l^j) = 1 - \frac{\min(\sigma_{\gamma_k^i}, \sigma_{\gamma_l^j})}{\max(\sigma_{\gamma_k^i}, \sigma_{\gamma_l^j})}$$

Region Matching Algorithm

The region matching algorithm operates on the sets of previous and current blobs $\{B_{t-1}^i; \forall i\}$ and $\{B_t^j; \forall j\}$ respectively. Figure 6 illustrates the result between a pair of typical blobs extracted using the algorithm described below.

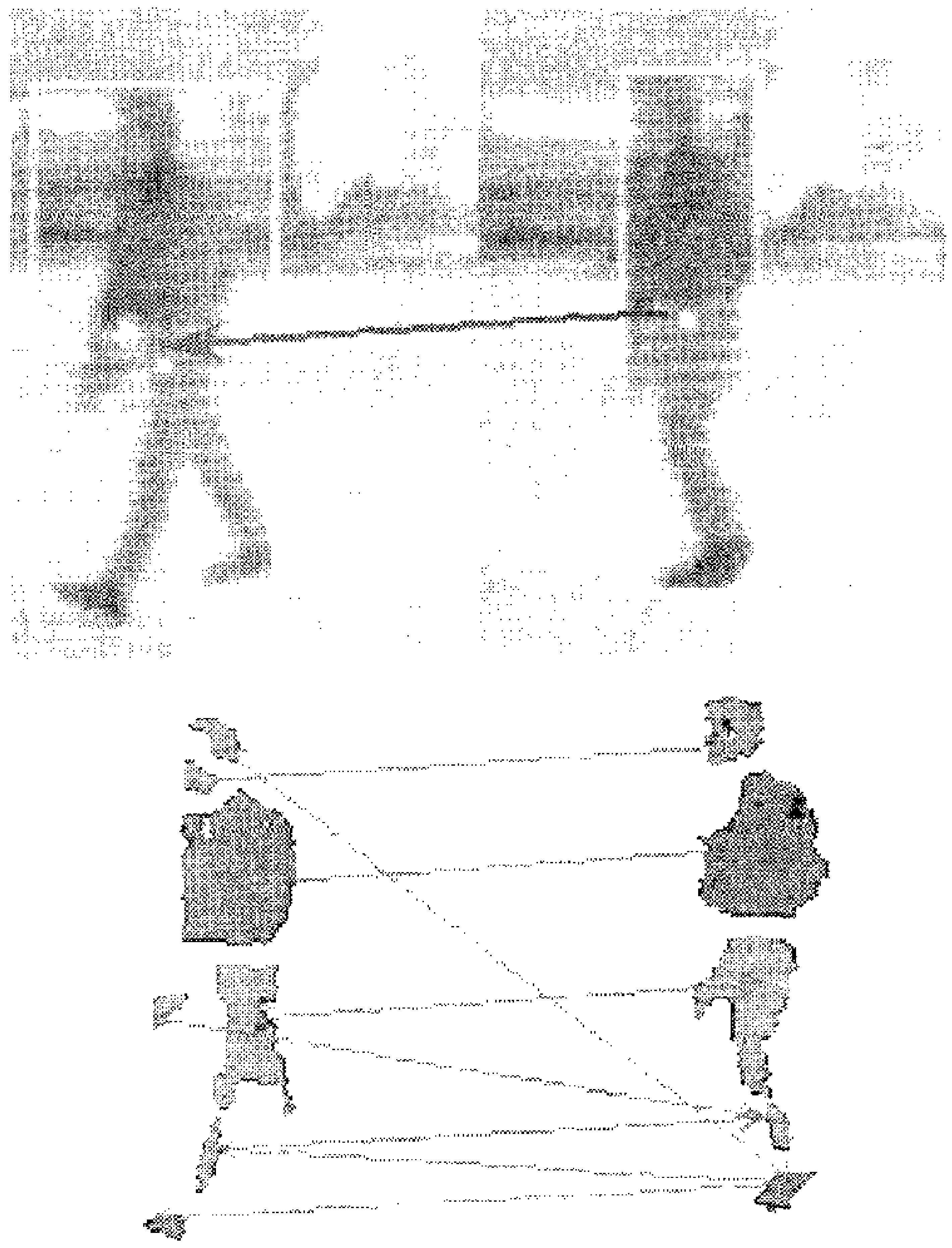

Fig. 6 Matching results

For each predicted object: B_{t-1}^i

- Retrieve the regions $[\gamma_1^i, \cdots, \gamma_{N_i}^i]$ of predicted blob i (segmented in previous cycle).
- Predict location of corresponding current blob, and recover each current blob overlapping the predicted blob.

For each current overlapping blob B_t^j

1) Segment the corresponding current blob B_t^j into its regions $[\gamma_1^j, \cdots, \gamma_{N_j}^j]$
2) Perform a greedy search using equation (3), select the match $(\gamma^i\ \gamma^j)$ with minimum match error over all match pairs $(\gamma_k^i, \gamma_l^j) \in B_{t-1}^i \times B_t^j$.

- If minimum functional value of this best match is greater than some validation threshold, object is assumed to be totally occluded.

Finally, the predicted velocity for blob B_{t-1}^i from this scheme is computed as the vector separating the centroids of the best match regions γ^i, γ^j.

5 Kalman tracking

The Kalman filtering method relies on two main equations: an equation on the state process and another on the measurement process[25]. Once estimated, the state parameters are updated and fed forward for the next iteration phase of the filtering. Each iteration involves 4 steps to be performed along which predictions are done and matrices are updated before jumping to the next iteration.

$$\boldsymbol{x}_{k+1} = \Phi_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{4}$$

where $\boldsymbol{x}$ is of dimension N, Φ_k is the $N \times N$ state transition matrix that relates the states of the process at time k and $k+1$, and $\boldsymbol{w}_k$ is the noise vector associated with the process model between time k and $k+1$.

The second observation equation governing the Kalman filters is the equation that linearly relates the measurements (the observations) to the state vector:

$$\boldsymbol{z}_k = H_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{5}$$

where $\boldsymbol{z}_k$ is the M dimensional noisy and distorted observation vector at time k, $\boldsymbol{v}_k$ is an M

dimensional noise model vector and H_k is an $M \times N$ dimensional square matrix relating the state and measurement parameters in a non noisy process.

The two noise processes w_k and v_k associated with the state and measurement equations respectively, are assumed to be white uncorrelated noise and independent of each other, hence we can write

$$E[w_k w_k^T] = Q_k, \quad E[v_k v_k^T] = R_{z_k} \tag{6}$$

After further development, the method to implement the Kalman equations is described in the four following steps[21].

Step. 1 State and Observation Prediction

$$\hat{x}_k^- = \Phi_k x_{k-1} \tag{7}$$

$$P_k^- = \Phi_k P_{k-1} \Phi_k^T + Q_k \tag{8}$$

$$\hat{z}_k^- = H_k \hat{x}_k^- \tag{9}$$

$$Z_k^- = H_k P_k^- H_k^T \tag{10}$$

where $\hat{x}_k^-$, P_k^- are the predicted uncertain states, and $\hat{z}_k^-$, Z_k^- are the uncertain locations of the predicted observation. For system noise Q_k, see section 5. 2.

Step. 2 Recovery of Observation

The appropriate new observation z_k and uncertainty R_{z_k} must be identified. In cluttered scenes a Greedy Search algorithm typically selects the closest observation using the Mahalanobis distance based on predicted and observation uncertainty. In our algorithm it is identified by a correlation process described in section 4. 1.

Step. 3 Compute Kalman Gain

$$K_k = P_k^- H_k^T (Z_k^- + R_{z_k})^{-1} \tag{11}$$

H_k remains constant throughout the Kalman filtering process. At time $k=0$, we need to have a prior estimate of P_k^- in order to start the system. It is often estimated from the covariance of the first observation. In our study, the first P was replaced by a scaled identity matrix. For details on R_{z_k} implementation, see section 5. 2.

Step. 4 Kalman Update

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - z_k^-) \tag{12}$$

The term $z_k - z_k^-$ represents the error between actual and predicted observation.

$$P_k = (I - K_k H_k) P_k^- \tag{13}$$

The covariances P_k represents the new current uncertainty in the updated state vector, while I is the identity matrix and K_k the Kalman Gain.

5. 1 Method C-constant velocity Kalman Tracker

The Kalman filtering process implemented in this study is based on the Kalman phases described above. The computational time required to perform a Kalman filtering system depends directly on the size of the state and measurement vectors, x and z respectively. More precisely, it depends on the time required to make a matrix inversion operation in the first step (see Equation (11)). And the time required to perform a matrix inversion operation increases exponentially, this places a real time constraint on the choice in the vector models. As the overall tracking algorithm involves many sub-algorithms, like background estimation, foreground detection and segmentation, the state and measurement vectors have been chosen to be of the simplest form: z contains only the foreground object positions and x will involve both position and velocity of the blobs.

$$z_k = [x_k \quad y_k]^T \tag{14}$$

$$x_k = [x_k \quad v_{x,k} \quad y_k \quad v_{y,k}]^T$$

From here, the matrixes Φ and H of equations (4) and (5) respectively can be constructed as follows:

$$\Phi = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{16}$$

5.2 Determining the noise processes

The determination of the process noise covariance Q is difficult to estimated as there is no ability to observe and measure directly the process. Sometimes a relatively simple process model (as assumed in this study) could produce acceptable results if there is enough uncertainty into the process via the selection of Q. Moreover, Q is the parameter which controls the adaptation capacity of the algorithm to adapt its tracking to noisy measurements. In other words, Q fixes the fluctuation of the coefficients of the filter. A choice of $Q=0.33I$ has been chosen empircally and is satisfying for all tracking scenarios.

The component R_{z_k} of the covariance matrix of equation (6) represents the uncertainty on the measurement which is in this study the position of the centroid of the foreground objects. Estimation of R_{z_k} will depend on the accuracy of the foreground detection method. In this work R_{z_k} is estimated based on the second order shape matrices of the region—the scatter matrix. In this study this scatter measure is computed and a more realistic centroid uncertainty derived from the scaled Scatter matrix. The Scatter matrix S of a region is defined as follows:

$$S = \frac{1}{N}\sum_i [(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_\mu)(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_\mu)^{\mathrm{T}}] \tag{17}$$

where N is the number of pixels in the region, and the centroid $\hat{\boldsymbol{x}}_\mu = (\mu_x, \mu_y)^{\mathrm{T}}$ is defined as

$$\mu_x = \frac{1}{N}\sum_i x_i, \quad \mu_y = \frac{1}{N}\sum_i y_i \tag{18}$$

and $\{\boldsymbol{x}_i = (x_i, y_i)^{\mathrm{T}};\ 1 \leqslant i \leqslant N\}$ are the locations of pixels within the moving region.

The estimation of the two noise processes using the Q and R parameters has been satisfactory for the tracker to be robust and reliable in most of the situations. However, in the cases where the object motion is not well modelled by the constant velocity motion model, the filter can easily loose the track *i. e.* the predicted object locations deviate significantly from observations. Consequently a fading factor η_k has been introduced in order to enable the filter to adapt[26] which is multiplied to the matrix P_k to increase the state uncertainty which in turn increases the Kalman Gain. This fading factor depends on the difference of the magnitudes of the velocity between two iterations *i. e.*

$$\eta_k = \log(\| \boldsymbol{v}_k - \boldsymbol{v}_{k-1} \|) + 10 \tag{19}$$

where $\boldsymbol{v}$ is extracted from the state vector defined in equation (14).

6 Results

In this section, we will evaluate tracking performance on four difficult image sequences containing one or more occlusion events. Each image sequence is illustrated by six frames in Figures 7, 8, 9 and 10 whose sizes are 192×144, 384×288, 288×216 and 192×144 respectively. Sequences 1, 3 and 4 are taken from cameras placed in public transport systems while Sequence 2 consists of an outdoor sequence taken from the PETS[1] database.

In Figures 7 to 10 the results of the hybrid tracking Method ABC are displayed for each of the selected frames. Each tracked object is assigned a unique colour. The method relies on the three independent algorithms. Each predicts the location of the observation

[1] vısualsurveıllance. org

and the one which gives the minimal correlation error is assigned to the tracker.

Since we are only interested in evaluating the tracking performance *i. e.* assignment of consistent labels over time) in the case of occlusions, the accuracy of the bounding boxes positions is not assessed—only the accuracy of the labelling. The evaluation methodology for accessing the accuracy of labelling counts the number of times $N_{D,i}$ an object i has been assigned the correct label relative to the number of times that object is in the view volume N_i. For each sequence S, labelling accuracy for each method can be computed as

$$\text{Accuracy}_S = \frac{\sum_{\text{for each object i in sequence S}} N_{D,i}}{\sum_{\text{for each object i in sequence S}} N_i}$$

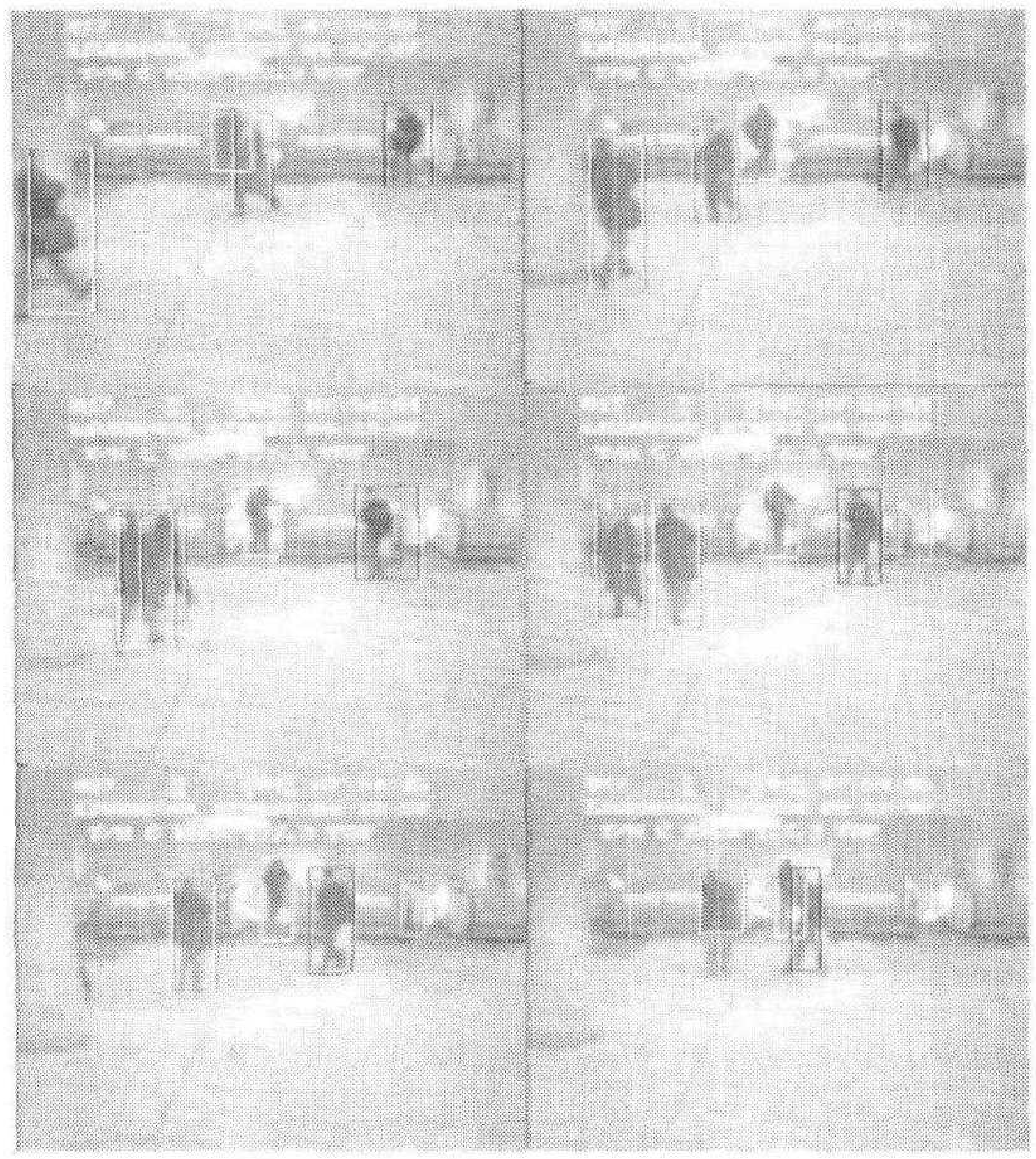

Fig. 7 Sequence 1

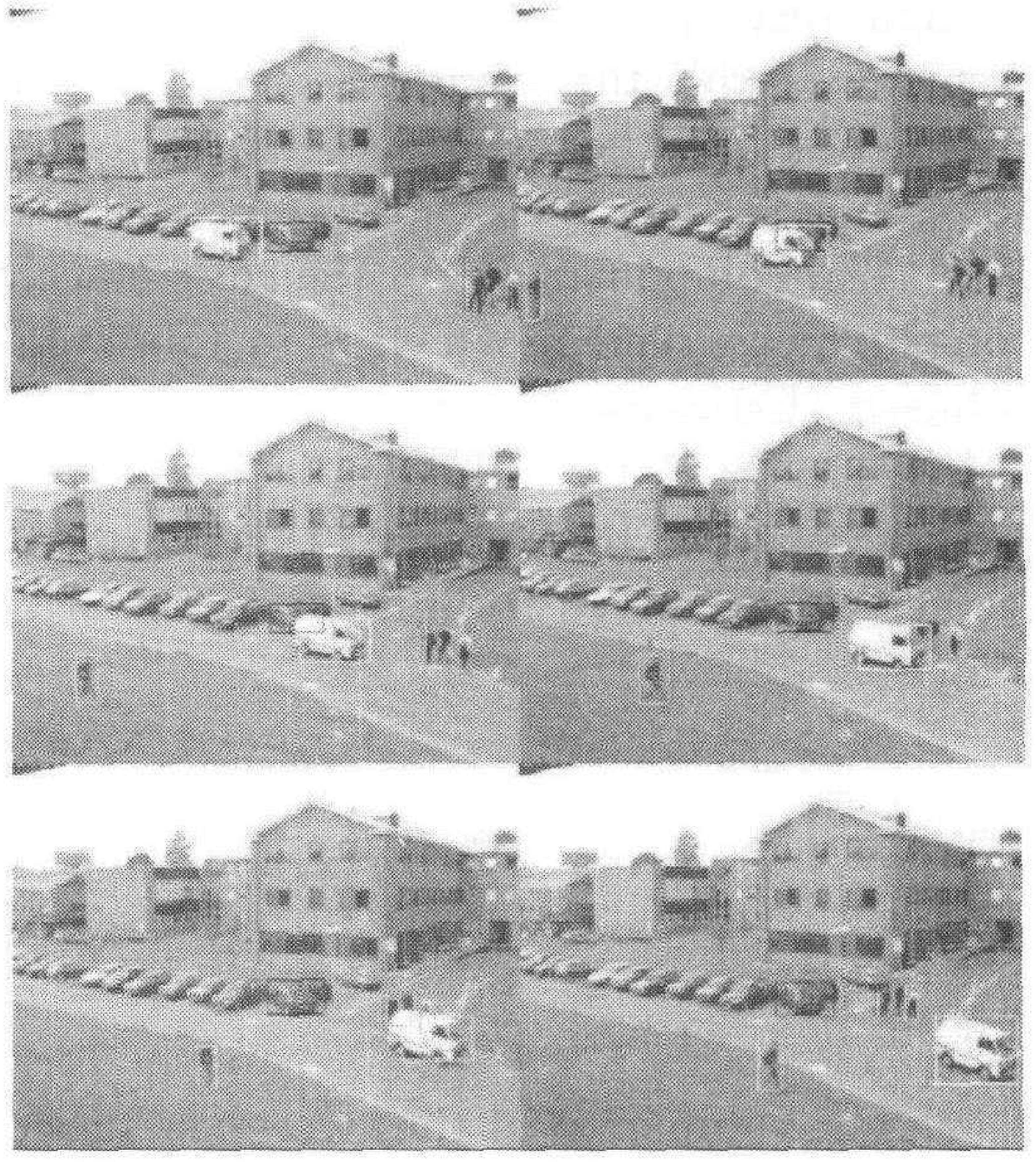

Fig. 8 Sequence 2

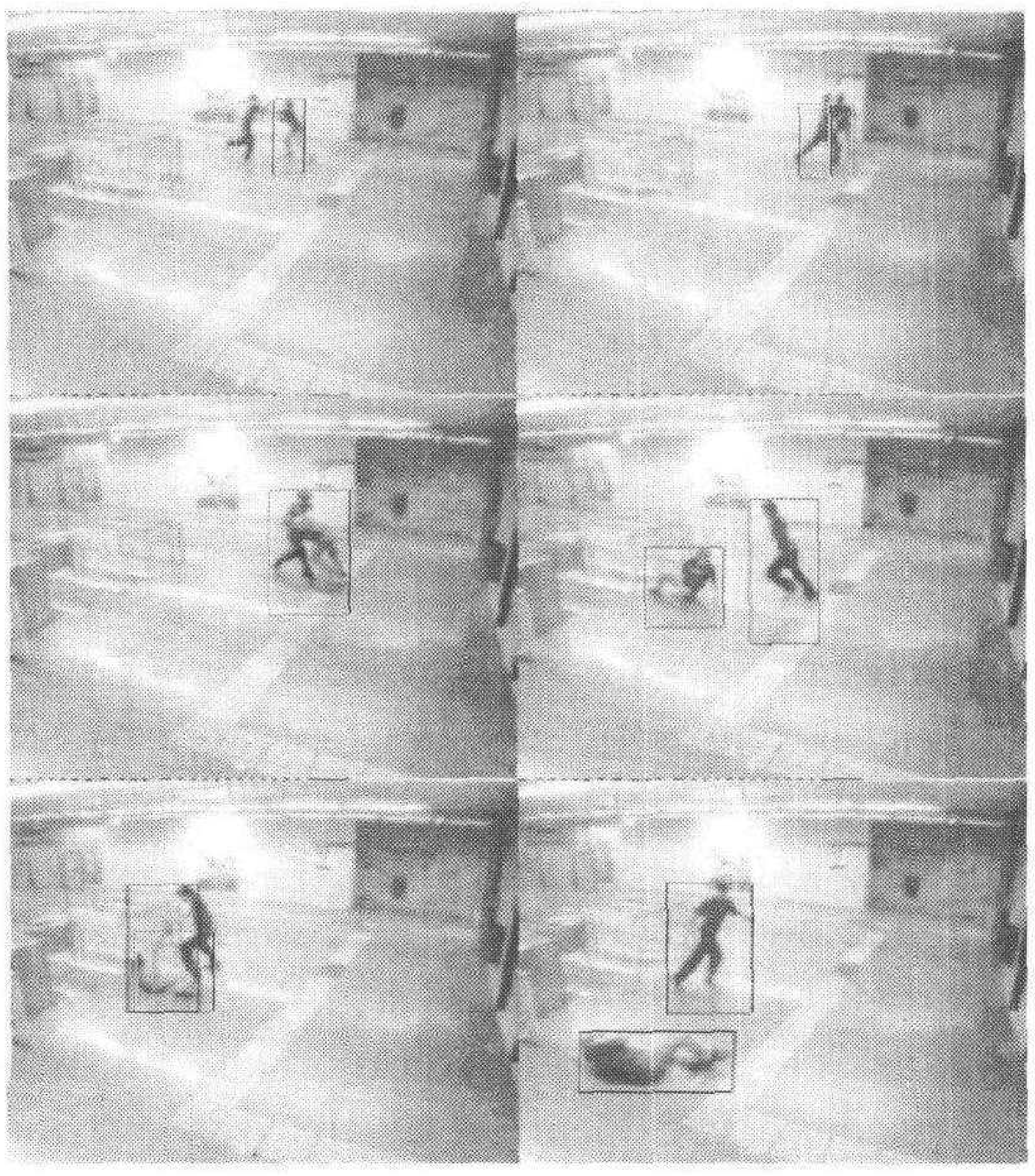

Fig. 9 Sequence 3

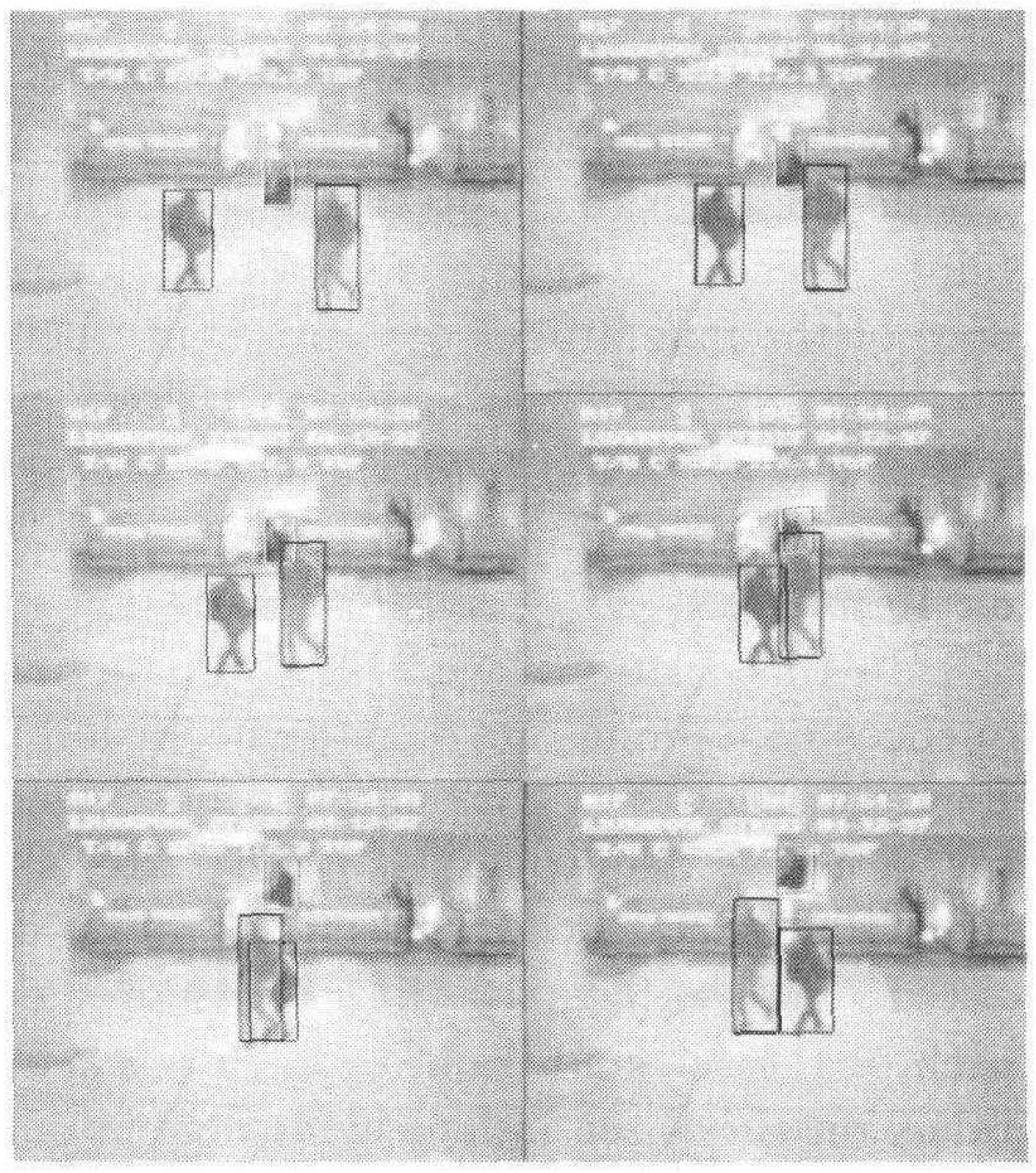

Fig. 10 Sequence 4

Table 1 presents the accuracy of the three methods A, B and C run independently against the accuracy of the hybrid ABC.

Table 1 Accuracy of methods

Method	Accuracy (%)			
	Seq. 1	Seq. 2	Seq. 3	Seq. 4
A	94	82	21	60
B	3	32	2	90
C	92	90	10	40
ABC	96	89	24	98

Sequence 1 presents multiple small and dark objects of pedestrians occluding each other while walking with a relatively small 2D plane speed and with a relatively constant acceleration, making it difficult for Method B to segment distinct textures. On the another hand Methods A and C very successful. Sequence 2 contains very similar occlusion scenarios as Sequence 1 except that some objects like cars are big enough to allow Method B to make a contribution. Sequence 4 differs from Sequences 1 and 2 in the fact that the objects appear very quickly in the view with high velocities, making method C hard to adapt and Method A inaccurate. On another hand, this sequence has been made up of similar objects for the whole sequence. Those objects have been selected big enough with non-distinctive textures to show that Method C can overcome occlusion issues when other methods fail. Finally Sequence 3 presents two people simulating a fight, where their velocities, shapes and appearances are changing constantly for the whole sequence. Consequently both motion model methods cause mistrack or simply fail to track all present objects in the scene.

7 Conclusion

An investigation of the issues encountered in all occlusion scenarios is presented here. Numerous tracking algorithm use motion models of different complexity hoping that the objects will behave according to a predicted model. Two have been implemented here. However, despite the predictive accuracy of the two motion model based tracking schemes, they cannot deal with all situations. To support these traditional approaches, we have implemented an appearance-model based technique which decomposes the moving region (or blob) into homogeneous greylevel regions, and matches these between frames. Three simple and quick algorithms have been combined together to maximise the robustness in the face of complex occlusion scenarios. Although, the results presented cannot cope with all scenarios, a better handling of the occlusion process has been achieved.

Future work will address a better segmentation process particularly where there is a very high level of contrast between object and background, and the improvement of motion model. In addition, a greater level of object localisation would be desirable using an improved foreground detection scheme which include with shadow detection and removal.

References

1 Stauer C, Grimson W E L. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000,**22**(8):747～757

2 Ellis T, Xu M. Object detection and tracking in an open and dynamic world. In:Proceedings of the 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Hawaii:IEEE Press, 2001

3 Mittal A, Huttenlocher D. Scene modeling for wide area surveillance and image synthesis. In:Computer Vision and Pattern Recognition (CVPR'00), IEEE Computer Society,2000. 2160～2167

4 Renno J R, Orwell J, Jones G A. Learning surveillance tracking models for the self-calibrated ground plane. In: British Machine Vision Conference, Cardi:British Machine Vision Association, 2002

5 Nir Friedman,Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In:Proceedings of the 13rd Conference on Uncertainty in Artificial Intelligence, 1997. 175～181

6 Dieter Koller,Joseph Weber,Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In: ECCV (1),

Sprnger-Verlag, 1994. 189～196

7 Rosin P L, Ellis T. Image difference threshold strategies and shadow detection. In:Proceedings of British Machine Vision Conference BMVC 1995, Birmingham:Morgan Kaufmann Publishers, 1995. 347～356. 43

8 Siebel N T, Maybank S J. Real-time tracking of pedestrians and vehicles. In:Proceedings of the 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Hawaii:IEEE Press, 2001

9 Haritaoglu I, Harwood D, Davis L S. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, **22**(8):809～830

10 Marcenaro L, Marchesotti L, Regazzoni C S. Tracking and counting multiple interacting people in indoor scenes. In:Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Copenhagen, 2002. 56～61

11 Brock-Gunn S A, Dowling G R, Ellis T J. Tracking using colour information. In:Technical Report TCU/CS/1994/7, City University, Department of Computer Science, 1994. 207～216

12 Chang Ting-Hsun, Gong Shao-Gang. Bayesian modality fusion for tracking multiple people with a multi-camera system. In:Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, Kingston:Kluwer Academic Publishers, 2001. 79～87

13 Justus H Piater, Stephane Richetto, James L Crowley. Event-based activity analysis in live video using a generic object tracker. In:Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Copenhagen, 2002. 1～8

14 Renno J, Orwell J, Jones G A. Towards plug-and-play visual surveillance: Learning tracking models. In:Proceedings of IEEE International Conference on Image Processing, Rochester, New York:IEEE Press, 2002

15 Haritaoglu I, Harwood D, Davis L S. W4: Who? When? Where? What? A real time system for detecting and tracking people. In:Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, 1998

16 Andrew Senior. Tracking people with probabilistic appearance models. In: Proceedings of 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Copenhagen: IEEE Press, 2002. 48～55

17 Bar-Shalom Y, Fortmann T. Tracking and data association. Mathematics in Science and Engineering, Academic Press, 1988

18 Bakowski A, Jones G A. Video surveillance tracking using colour adjacency graphs. In: IEE Conference on Image Processing and Its Applications, Manchester:Institution of Electrical Engineers, 1999. 794～798

19 Orwell J, Remagnino P, Jones G A. From connected components to object sequences. In:Proceedings of the 1st IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, IEEE Press,2000. 72～79

20 Rosales R, Sclaroff S. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In: IEEE International Conference on Computer Vision and Pattern Recognition, 1998

21 Khan S, Shah M. Tracking people in presence of occlusion. In: Asian Conference on Computer Vision, Taiwan, 2000

22 Collins R, Lipton A, Kanadeand T, Fujiyoshi H, Duggins D, Tsin Y. A System for Video Surveillance and Monitoring: VSAM Final Report. Carnegie Mellon University,CMU-RI-TR-00-12, 2000

23 Anzalone A, Machi A. Video-based management of traffic light at pedestrian road crossing. In: Advanced Video—Based Surveillance Systems, Kluwer Academic Publishers, 1999. 49～57

24 Fuentes L M, Velastin S A. People tracking in surveillance applications. In:Proceedings of the 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance,Hawaii:IEEE Press, 2001

25 Brown R G, Hwang P Y C. Introduction to Random Signals and Applied Kalman Filtering. 3rd Edition. Canada: John Wiley and Sons, 1985

26 Huwer S, Niemann H. 3D model based detection and tracking of people in monocular video sequences. In:Proceedings of the IASTED International Conference on Signal and Image Processing, Las Vegas: Nevada, 2000. 19～23

E. Corvee Received his bachelor degree in Physics with Electronic and Computing (1999) from Kingston University, Surrey, UK, and master degree in Imaging and Digital Image Processing (2000) from King' s College, London, UK. Since then, he has been a researcher in Computer Vision studying for a Ph. D. degree in Motion Analysis for Post-Production Applications with primary interest in dynamics of image sequences and the development of automated tracking for surveillance applications.

S. Velastin Led the Video Research Laboratory at Kings College London until September 2001 before he joined the Digital Imaging Research Centre (DIRC) at Kingston University. He has worked for a number of years on visual monitoring of crowds and the detection of incidents related to personal security (EPSRC grants GR/H78511/01(P), GR/J46005/01(P), GR/M29436/01-02(P), Framework 4 EU grant CHROMATICA (TR1016)). He currently holds two EU grants-ADVISOR (IST-1999-11287) and PRISMATICA (GRD1-2000-10601)-related to these topics, and is collaborating with a number of public transport operators, NGOs, research institutions and industrialists on industrially funded projects. Recently he has organised and is chair of the IEE Workshop on Intelligent distributed surveillance systems and has published over 40 publications in the fields of computer vision and distributed systems. His main research interests include the development of computer vision technologies for crowd analysis and personal security, and the development of distributed computer systems that integrate intelligent detection devices to support the operational needs of public security organisations.

G. A. Jones Director of the Digital Imaging Research Centre at Kingston University. Received his Ph. D. degree in computer vision from Kings College London, and has over 15 years experience in image and video sequence analysis, intelligent systems and multimedia data communications, and has managed a number of industrial and government funded projects in these areas. Recently he has been responsible for video/image analysis projects supported by the film and special effects industry (Computer Film Company Ltd., UK and Dynamic Digital Depth Pty. Ltd., Australia) and by the video security industry (Primary Image Vision Systems Ltd.). He is currently co-investigator on the EU INMOVE (IST-2001-37422) project developing an expandable set of software tools enabling the provision of a new range of intelligent video based services to end users in various mobile/wireless networks. In 2000, Dr. Jones chaired the British Machine Vision Workshop on Visual Surveillance and was co-chair of the IAPR Workshop on Advanced Video-based Surveillance Systems in 2001.

综合多种预测方案实现遮挡情况下的目标跟踪

E. Corvee S. Velastin G. A. Jones

(*Digital Imaging Research Centre*, *Kingston University*, *Kingston*, 英国)

(E-mail:{E. Corvee,Sergio. Velastin,G. Jones}@Kingston. ac. uk)

摘 要 由于采用了多种运动预测方案,本文提出的目标跟踪方法能选择最佳的观测结果,实现对非标定固定焦距的静止摄像机的单目图像序列中的目标跟踪.静态背景参考图像由混合模型法估计,在最简单的环境下,跟踪算法则采用匀加速运动模型对目标完成跟踪.本文主要贡献是采用了三个预测器和最小方差相关时选择目标最可能的位置.三个预测器分别是:$\alpha-\beta$跟踪方案、卡尔曼滤波和区域分割匹配方案.本跟踪方案通过具有不同遮挡情况的序列图像得到了验证.

关键词 混合预测,最小方差相关,允许遮挡

中图分类号 TP391.41