ELSEVIER

Image and Vision Computing 20 (2002) 895–903

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Path detection in video surveillance

Dimitrios Makris*, Tim Ellis

Information Engineering Centre, City University, London EC1V OHB, UK

Abstract

This paper addresses the problem of automatically extracting frequently used pedestrian pathways from video sequences of natural outdoor scenes. Path models are learnt from the accumulation of trajectory data over long time periods, and can be used to augment the classification of subsequent track data. In particular, labelled paths provide an efficient means for compressing the trajectory data for logging purposes. In addition, the model can be used to compute a probabilistic prediction of the pedestrian's location many time steps ahead, and to aid the recognition of unusual behaviour identified as atypical object motion. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: People tracking; Learning paths; Scene labelling; Route detection; Track prediction; Video annotation

1. Introduction

We consider the problem of learning the routes or paths taken by pedestrians walking through outdoor scenes. The routes and paths are represented by a spatial model that identifies regions frequently used by pedestrians as they transit the scene [12]. The task is tackled in the context of a multi-camera video surveillance network, where video data is available from cameras with both overlapping and non-overlapping fields of view, though results in this paper will be presented for only single camera data.

The motivation for building such models is threefold. Firstly, we require an efficient method to encode and annotate individual tracks to construct a log of movement patterns over long periods of time (i.e. weeks). Object trajectories can be assigned to one of only a small number of detected pathways, resulting in significant compression for the logged data. Secondly, accumulating tracks over a long time period establishes a norm of typical movements to be established and this can support the recognition of atypical or unusual movement and patterns of behaviour. Finally, the information can be used to support the tracking process, enhancing the predictive capabilities of the system to correspond objects over successive image frames. This gives the system the opportunity to predict forward many frames, based on the current location and direction.

Routes and paths are distinguished in the following way: a route identifies a frequently used pathway followed by a pedestrian through the scene; a path identifies a scene-specific feature (e.g. a segment of pavement between road sections). In most cases, a route is associated with a matched pair of entry and exit points to the image, i.e. where an object first appears and where it disappears from the field of view. This might normally be at the borders of the image, but can also occur at occlusion boundaries, from which the tracked object does not reappear. The justification for making this distinction is that the raw input data is object-based, consisting of object trajectories extracted from image sequences. However, for the three tasks identified above (labelling, logging and prediction) we need to associate trajectories with scene features.

Section 2 gives a brief overview of some of the problems of extracting routes and paths from image sequences and examines previous reported work for this task. Section 3 describes the models that will be generated for this research, and then a description of the implementation of the algorithms for building these models. Then, results are presented for applying the model building from image sequences taken from two separate sites.

2. Previous work

In general, the first stage of path recognition requires the tracking of individual objects, establishing frame-to-frame correspondence and this task has been extensively studied by many researchers, and is widely described in the literature. However, much less effort has been directed at the path detection problem [3,4,8].

In extracting pedestrians routes, the method must be able to cope with a wide range of inconsistent motions, often resulting from a variety of interaction: people avoiding each

* Corresponding author.
E-mail addresses: d.makris@city.ac.uk (D. Makris), t.j.ellis@city.ac.uk (T. Ellis).

0262-8856/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0262-8856(02)00098-7

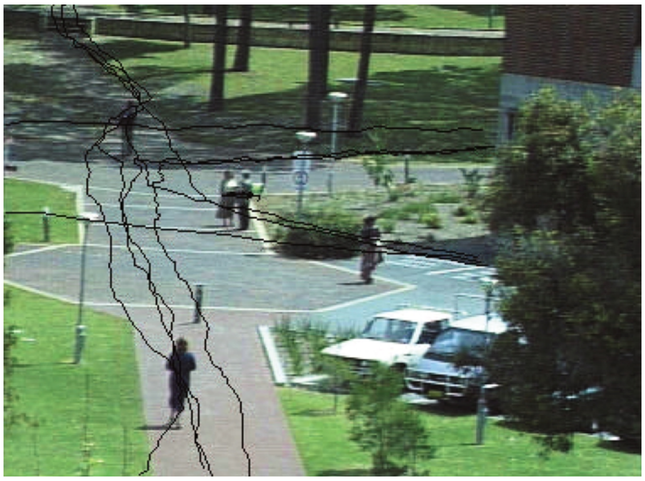

Fig. 1. Sample trajectories plotted in image plane.

other at busy times; avoiding static objects in the pathway; or just the casual meandering of a lone pedestrian. The deviation from a 'bee's line' track (i.e. straight) is, in part, dependant on the width of the pathway. However, the trajectories can be further complicated by excursions outside the pathway (e.g. onto the grass). Fig. 1 shows a small number of trajectories (generated from the blob centroids) extracted from an image sequence for one of the sites described in Section 5.

Fernyhough [3] built a database of object paths by accumulating the frequency of trajectory occurrences in the spatial domain. He derived image regions from the database using a classification proposed by Howarth and Buxton [5], dividing the space into leaf and composite regions, which can be used to represent areas of similar behaviours.

Johnson and Hogg [6] used a vector quantisation approach to model the distribution of trajectories using a neural network. An object's trajectory was represented by a set of smoothed position and velocity vectors. Many trajectories were accumulated to form a distribution in the image plane that was represented by a set of prototype vectors. The vector quantisation was implemented using two competitive learning networks, the first to model the distribution of flow vectors, and the second to model the trajectory distributions. A layer of leaky neurons connects the two networks, and introduces a memory element into the network architecture. The model supports the detection of untypical instantaneous motions for detecting atypical trajectories, and was extended to satisfy the requirements for prediction [7], but does not provide a mechanism for labelling the trajectories.

Sumpter and Bulpitt [9] adopt Johnson's methodology [6], but augment the network with feedback from the output into the input of the leaky neurons to learn the patterns of activations. They then apply this model to predicting behaviours of a flock of penned domesticated animals (ducks) in response to a predator threat.

Boyd et al. [1] classified movement flow between (manually) segmented image regions using a graphical model, with links representing the movement between adjacent regions, using an analysis based on network tomography, intended for statistically modelling data packet flow in computer communication networks. The method avoids extracting trajectories, registering only the changing density estimates of objects between adjacent regions to model the flow patterns. As such, it also fails to meet the main requirements for labelling and prediction that we require.

3. Scene model

We develop a spatial model for representing routes in the image. Each route is modelled with a central axis formed by a sequence of knot points (nodes), which represents some average of the route, bounded by an envelope that identifies the variation of the trajectories sampled for the route. The route nodes are spaced at equal separation distances equal to a resample distance. Each route has two terminator nodes (start and end) that typically correspond to entry/exit points in the image (Fig. 2).

A second level of model derives semantic descriptions of scene components inferred from the routes—the paths. The paths are described with the following features:

1. Entry/exit zones. Regions where pedestrians enter or exit the image
2. Junctions. Regions where routes cross each other.

Pedestrians enter and exit the scene in specific regions, following specific routes. A route describes the entire

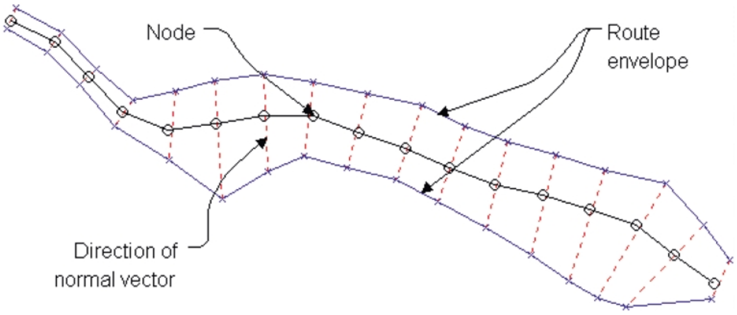

Fig. 2. Spatial model of a route.

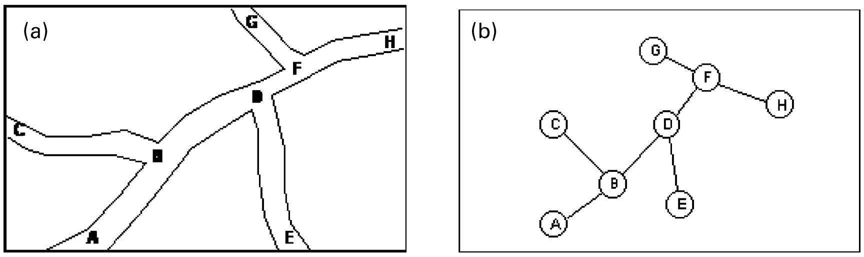

Fig. 3. (a) Spatial representation of paths. (b) Graph representation of paths.

trajectory of a pedestrian from the time that he enters the scene till the time that he exits and can be described as a curve with specific start and end points. Junctions are the areas where routes cross or bifurcate. We use a graph to represent the topology of the network of nodes (entry/exit points and junctions) for the paths.

The models are learnt from example trajectories extracted from an image sequence of pedestrian motion. Trajectories are grouped using a geometrical analysis that compares the separation distance between a trajectory and an evolving route description. A second stage extracts paths from the routes by detecting route cross over points. These are classified as junctions, and are combined with the route entry/exit points to construct the topological representation shown in Fig. 3(b). Fig. 3(a) shows the corresponding spatial representation of the paths.

4. Implementation

Routes are learnt by grouping sets of trajectories. The route description is stored in a database. Section 4.1 describes how trajectories are selected for grouping, the criteria for matching a new trajectory to a route, the updating process of routes in the database, and finally, how routes are merged.

A trajectory is derived from tracking an object across many frames extracted from an image sequence. It consists by a sequence of 2D points corresponding to a specific point on the target. The centroid of the target object seems to be a good choice, because its estimation from the tracking algorithms is more reliable than the estimation of other points like the top of the bottom of the object. To avoid learning with unrepresentative data, we eliminate short trajectories and trajectories of moving objects whose direction changes frequently over short time periods.

The distance between consecutive points of the trajectory varies considerably because of the speed of the target or/and the distance from the camera or/and the direction of the motion. For this reason, a valid and suitable trajectory is resampled over the space, using linear interpolation, to normalise the trajectories of high and low speed objects and to counter the effects of perspective. Because the aim of the proposed models is to represent spatially the physical extent of routes, velocity information is not interesting, so it can be discarded by resampling over space. But, even if velocity preservation is required, this can be achieved by keeping a velocity vector for each resampled point. The resample process also generates smoothed trajectories.

The route-learning algorithm takes the next trajectory and attempts to match it with all existing routes in the database. If a match is detected, then the matched route will be updated. If not, then a new route will be initialised. After a route has been matched with a trajectory, it becomes a candidate for merging with other routes in the database. The database builds up a typically a small number of routes, which represent the principal pathways taken by pedestrians in the scene. Routes with only a small number of updates are generally discarded.

Each route node i is characterised by:

- A 2D vector that represents the image coordinates of the node: $\mathbf{x_i} = [x_i, y_i]$
- A weight factor w_i that reflects the strength of the node, based on the number of times that it has been updated
- A normal vector $\mathbf{n_i} = [nx_i, ny_i]$, defined as the unit vector perpendicular to the local direction (defined by three consecutive nodes of the route main axis)
- A distribution of observations across the route (along the normal vector). Although this distribution could be modelled by a probability density function (PDF), we prefer to use the boundaries of this distribution, the left boundary $\mathbf{l_i} = [lx_i, ly_i]$ and the right boundary $\mathbf{r_i} = [rx_i, ry_i]$ as these points define the physical extent of the route on the image. The two boundary points lay along the normal vector of the node.

Table 1
Seven routes extracted from sequence 1 (left) and nine for sequence 2 (right)

Route	Nodes	Weight	Usage	Route	Nodes	Weight	Usage
1	31	7.5	0.1560	1	17	39.2	0.28
2	43	20.4	0.5880	2	17	26.0	0.186
3	26	9.0	0.1560	3	16	10.2	0.068
4	10	1.9	0.1300	4	16	19.9	0.134
5	34	2.6	0.0590	5	18	11.6	0.088
6	17	1.0	0.0110	6	15	6.3	0.04
7	24	1.0	0.0160	7	17	12.1	0.086
				8	14	7.2	0.042
				9	18	10.0	0.076

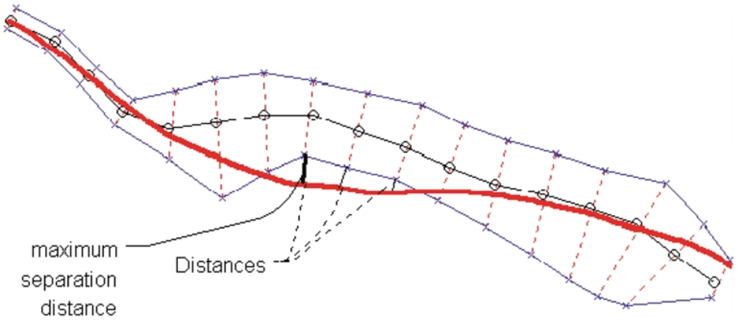

Fig. 4. Distances of a trajectory points from a route. The maximum separation distance is smaller than the threshold, so the trajectory matches the route.

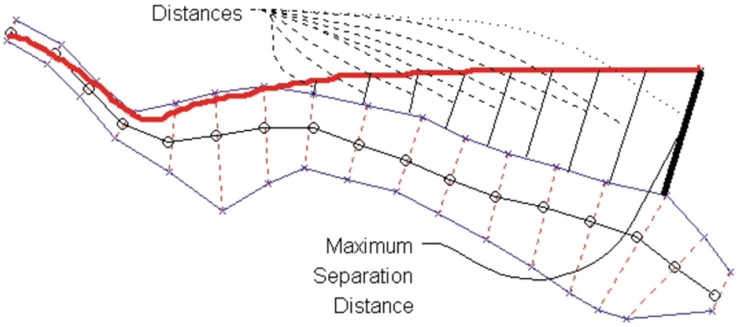

Fig. 5. Distances of a trajectory points from a route. The maximum separation distance is larger than the threshold, so the trajectory is unmatched.

4.1. Route-trajectory match

Each trajectory is compared to all existing routes in the database. The comparison of routes is based on a distance measure between the trajectory and a route. This distance measure is the maximum separation distance between the trajectory and the route.

Computation of the maximum separation distance requires the distances of the trajectory points from the envelope of the route. If the trajectory point is within the envelope, the distance is negative; otherwise it is positive and equal to the Euclidean distance of the point from the closest boundary of the envelope. The maximum of these distances for a given trajectory and a route is the maximum separation distance (Figs. 4 and 5). To cope with the case where the trajectory overlaps the route but is longer, the computation should not include the distances of the trajectory points beyond the terminators of the route (Fig. 6).

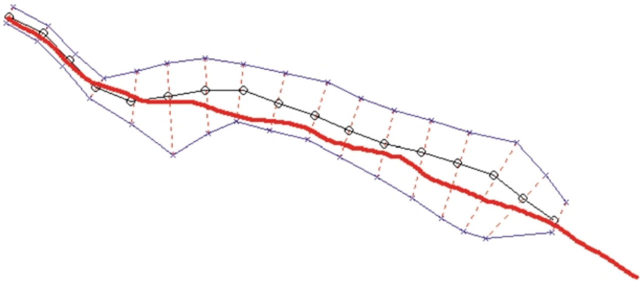

Fig. 6. Example of a route that must be extended.

After estimating the distance of the trajectory from all the routes, the trajectory is matched to the route with the minimum corresponding distance, under the condition that this distance is below a given threshold (Fig. 4). If the trajectory does not match to any existing route, a new route is created in the database, initialised with the trajectory.

4.2. Route updating

Route updating must also cope with varying degrees of overlap between route and the trajectory. In particular, if the trajectory is longer than the route at either end (Fig. 6), then the route must be extended.

A route is updated with a matched trajectory in the following three steps:

1. *Node updating*. Each node is updated using the intersection point of the trajectory with the normal direction line and the node weight for the new node position (x') :

   $$\vec{x}' = \frac{w}{w+1}\vec{x} + \frac{1}{w+1}\vec{x}_t$$

   The weight factor w is incremented; the trajectory is checked if it is beyond the envelope and if so, the associated boundary point becomes equal to the trajectory point that is on the normal direction.
2. *Route extension*. If there are trajectory points beyond the route end node, the route will be extended. In this case,

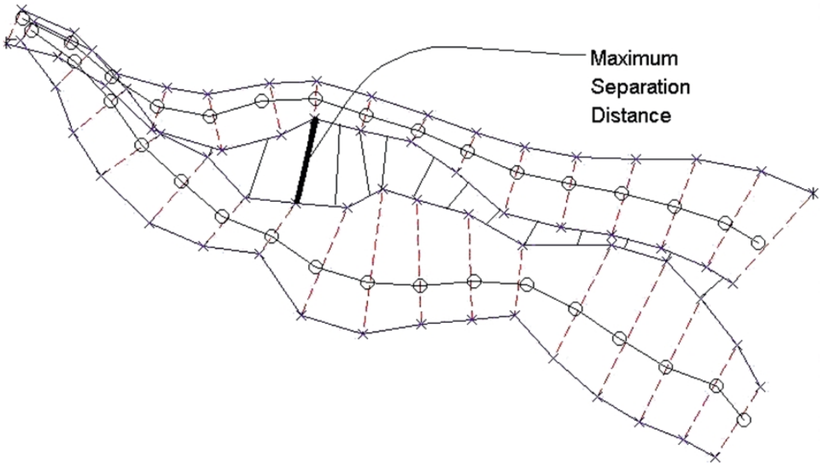

Fig. 7. Maximum separation distance of two paths. In this example, the distance is smaller than the threshold, so the paths will be merged.

the trajectory points extending beyond the end of the route are added to the route.

3. *Route resampling*. To maintain equal distances between the route nodes, the route is resampled (using linear interpolation) after updating, and the normal vectors are recalculated.

4.3. Route merging

Following updating, the updated route is compared with all the other routes in the database. If two routes are close enough, then they should be merged. The merging criterion is the maximum separation distance between the two routes.

To compute the maximum separation distance, the distance of each node from the other route is calculated. The maximum of these distances of the nodes of one route from the other route defines the maximum distance of one route from the other. Ideally, for a small resample distance, both distances are almost equal. In practice, the maximum separation distance between the routes can be defined as the mean of the two distances, or as the distance of the route with the smallest weight factor from the one with the highest weight factor (Fig. 7).

If the maximum separation distance of two routes is smaller than a defined threshold the routes are merged. The route merging is similar to route updating with a trajectory. First, the route with the highest weight is selected as the main route and then this is updated with the secondary route. Each node of the main route is updated with a virtual node of the secondary route that is estimated by the intersection of the secondary route with the normal direction line of the updated node. The main difference between updating and merging is that in the latter case, the weight factors (w_2) and the envelope of the secondary route must be considered. The envelope of the merged route is calculated from the combination of the envelopes of both routes and the weight and node position are calculated as follows:

$$w_1' = w_1 + w_2, \qquad \vec{x}' = \frac{w}{w+1}\vec{x} + \frac{1}{w+1}\vec{x}_t$$

At the terminators, if the secondary route has nodes that extend beyond the terminators of the main route, then the main route is extended. Finally, the main updated route is resampled.

The above algorithm requires only two parameters: (a) the resample distance between the route nodes that defines the accuracy of the route representation and (b) the distance threshold that represents the minimum allowed value of the maximum separation distance between different routes.

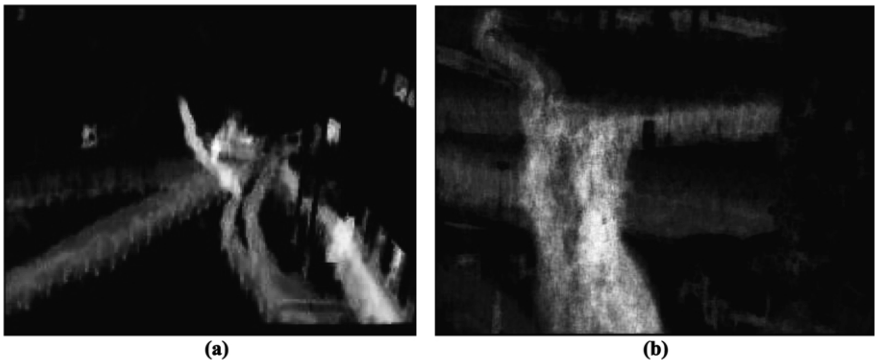

Fig. 8. (a and b) MHI (approximately 2 min of video) for the two data sets indicating the main regions of motion activity.

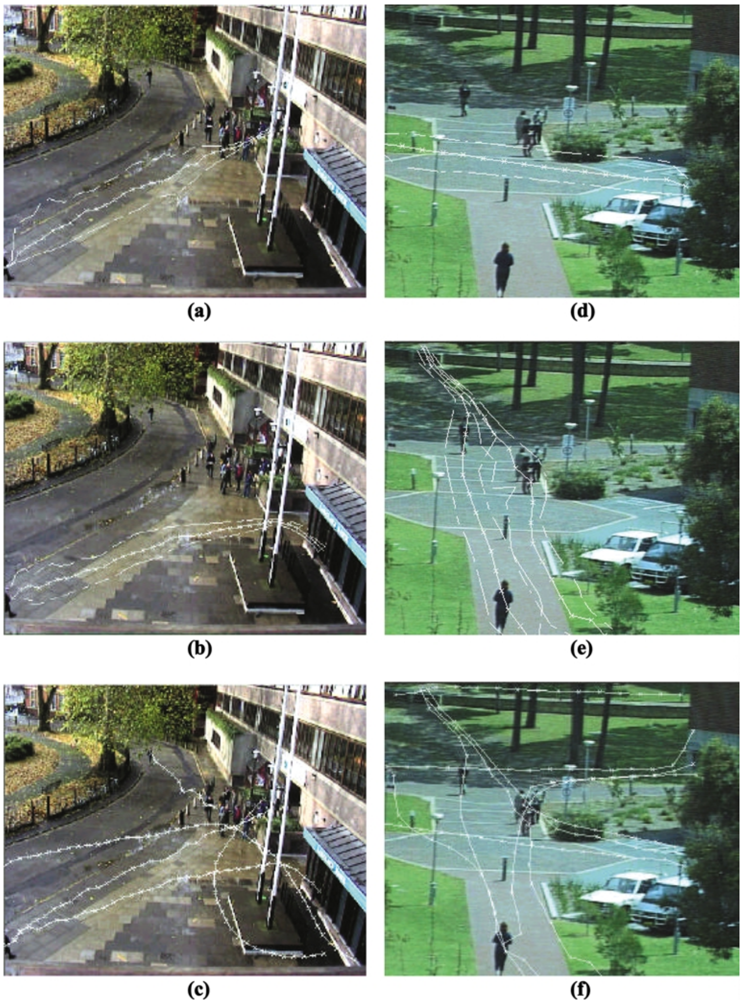

Fig. 9. (a and b) Two route models extracted from video 1. (c) The central axes of the route models of video 1. (d and e) Three route models extracted from video 2. (f) The central axes of the route models of video 2.

4.4. Constructing paths from routes

Paths are constructed by considering the following conditions: (i) grouping common sections of routes and creating a junction when the routes diverge, (ii) by finding where two routes cross each other, or (iii) by grouping route terminators.

To identify a common section of two routes, all the nodes of the routes are compared using a distance threshold based on the resample distance. To avoid forming large numbers of junctions at points where the two routes are only briefly close, a threshold ensures that we only perform grouping if the number of matched nodes is above some minimum.

To find out where two routes cross each other, we search for a pair of nodes from different routes within the minimum threshold distance and locate the crossing around these nodes. If we identify a node where several routes terminate and it is not close to the existing terminators, then this node defines a new terminator in the scene and therefore indicates that the route must split into two parts.

5. Results

The following results were obtained from video sequences from two separate sites. Approximately 25 min of video taken from a single camera were used to construct the route and path data from 340 object trajectories.

Fig. 8(a) and (b) shows the motion histogram image (MHI, [2]) from the two sites (sequence 1 and sequence 2, e.g. Fig. 9(a) and (d)) indicating the regions of the image where motion has been detected. In this case, the histogram is constructed from the binary blobs of tracked objects, which were extracted following background subtraction.

Table 1 identifies the main routes extracted from the two image sequences. The table indicates the number of nodes used to construct each path, the average number of trajectories that contributed to the route (weight), and finally the usage, which represents the probability of an object found on a particular path. Low weighted routes (i.e. 4, 6 and 7 of sequence 1) could be discarded, or may require more training data to determine if they represent frequently

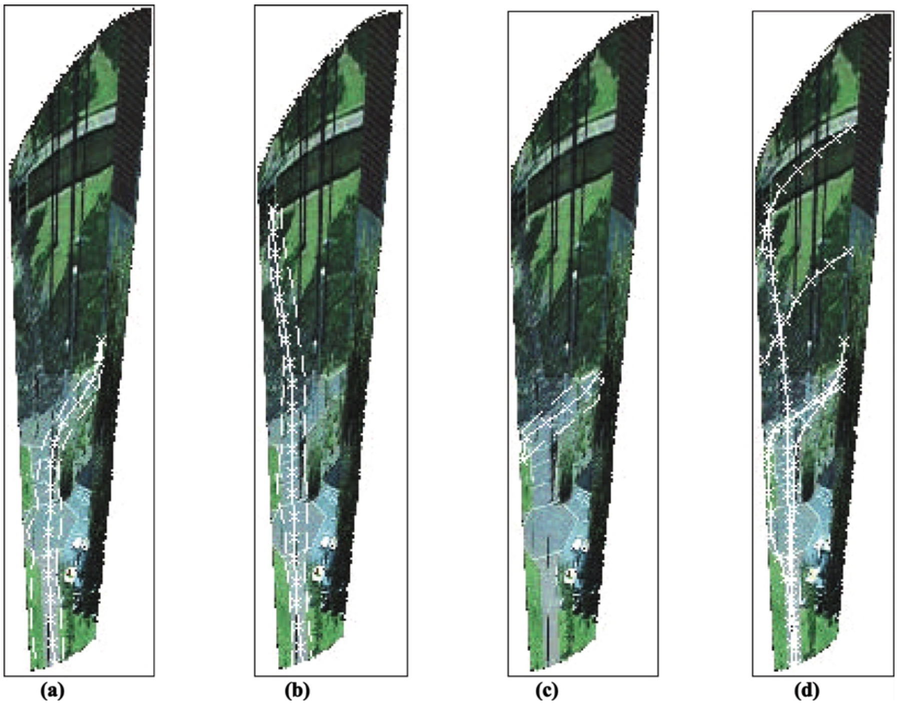

Fig. 10. Results of the algorithm using a ground plane coordinate system on the trajectories of video 2. (a–c) Three route models. (d) The central axes of all the route models.

used routes. Six routes with a low frequency of use have been discarded from the routes extracted from sequence 2.

Fig. 9(a) and (b) shows two routes extracted from the trajectories of sequence 1 (Table 1, routes 1 and 5). The routes were extracted from a video sequence (resolution 388 × 288) of 10 min, sampled at $2\frac{1}{2}$ frames per second, comprising 155 trajectories. Threshold values used for the trajectory and route merging were 20 pixels and the resample distance was 10 pixels. The routes show two pathways leading into (and out of) the main entrance, and the entrance to the University bank. As can be seen, the central axis and the envelope give a good match to the pathways visible in the image MHI shown in Fig. 8(a). Fig. 9(c) shows the central axis for all the routes extracted from the sequence.

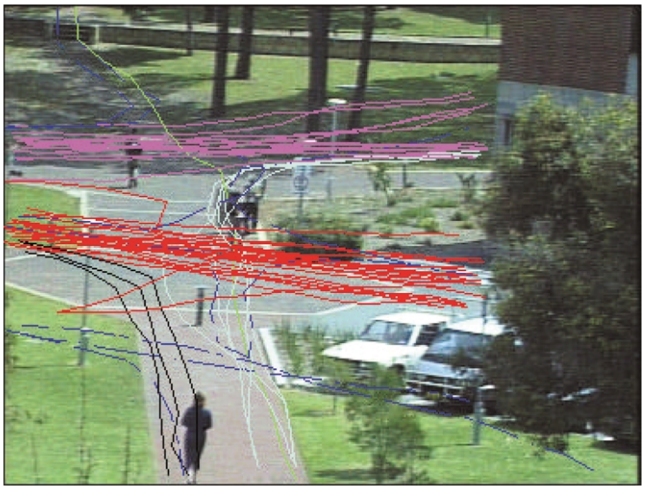

Fig. 11. Classification of trajectories using route model.

Fig. 9(d) and (e) shows three routes extracted from the trajectories of sequence 2 (Table 1, routes 7, 5 and 4). The routes were extracted from a video sequence (resolution 768 × 576) of 14 min, sampled at two frames per second, comprising 190 trajectories. Threshold values used for the trajectory and route merging were 60 pixels and the resample distance was 40 pixels. Again the central axes and the envelopes give a compact representation of the pathways visible in the image and can be favourably compared to the MHI shown in Fig. 8(b). Fig. 6(e) contains the two principal routes leading down from the steps at the back of the scene onto the path in the foreground. Two separate routes are detected as a result of the bollard in the centre of the pathway, which causes a bifurcation of the route. Fig. 9(f) shows the central axes for the nine principal routes detected in this sequence. Six low frequency routes were rejected, and are not shown.

Fig. 10 shows the result of the algorithm if ground plane coordinates are used instead on image plane coordinates. If we assume that the pedestrians walk on a plane surface (ground), then the image coordinates can be easily converted to ground plane coordinates, by projecting the trajectories on the appropriate plane. In this case, because the trajectories are formed by tracking the centroid of a human body, whose height is about 1.70–1.80 m, the trajectories should be projected on the plane, which is parallel to the ground and roughly 90 cm above it, which can give a satisfactory approximation of the real ground plane coordinates. Coordinate conversion is achieved by

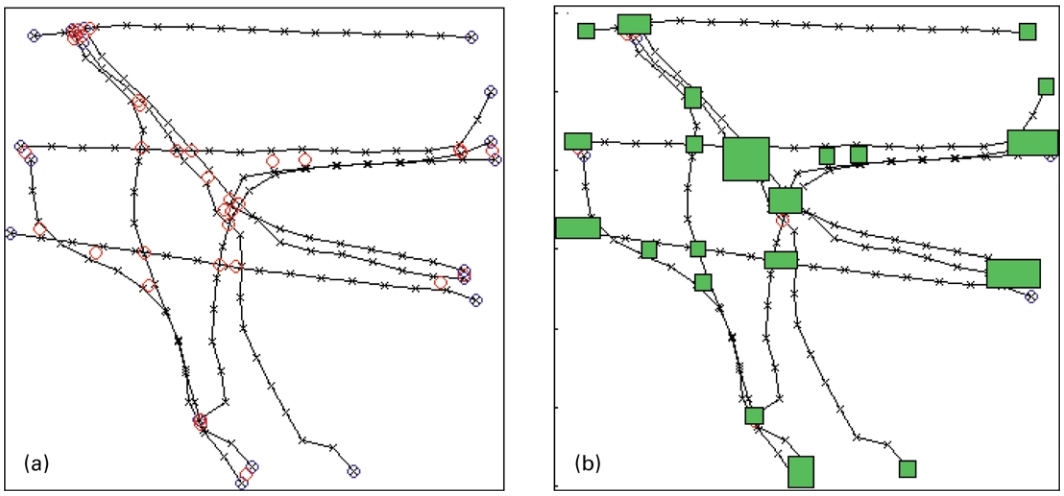

Fig. 12. (a) Detected junction (red circles) and entry–exit points (blue circles). (b) Green rectangles indicate the merged junctions and entry–exits.

using a geometric camera model as defined in Ref. [11]. Then, the method is applied with a resample distance 5000 mm and a threshold of 2000 mm.

The main benefit of representing the route models in ground plane coordinates is that in a multi-camera surveillance system, the ground coordinate system is common for all the cameras. Therefore, if some cameras have overlapped field of views, their route models can be merged. Another benefit of using a ground plane coordinate system is that the algorithm parameters are more meaningful and any perspective effects are minimised.

Fig. 11 shows the result of applying the route model to a set of 53 previously unseen trajectories. The colour coding uses solid colour lines (red, magenta, cyan, green and black) to identify trajectories that have been matched with known routes. Deep blue lines with gaps (see bottom left) indicate trajectories not matched with existing routes. It can be seen, for example, for trajectories in the top left, where pedestrians have been tracked coming down the steps and turning right at the end of the pathway. This pathway was not represented in the original training data. The algorithm identifies eight unknown routes from this set.

Fig. 12(a) shows the result of applying path detection to the route database. Junctions have been detected to indicate where routes cross or bifurcate (red circles). Entry and exit points are indicated by the blue circles. The threshold distance of 40 pixels is used (derived from the resample distance) for an image size of 768 × 576. Overlapped sections of routes are combined if they have two or more matched nodes. A pair of red circles defines the common section of the two routes, and two junctions will be formed as a result of merging this common section. In locations where the entry–exit points and junctions are found to be close, they are merged to construct the final network. Fig. 12(b) shows the final results of these merging operations.

Overlapped sections of routes are combined if they have two or more matched nodes. In locations where these entry and exit points are found to be close, a red circle indicates they will be merged to construct the final network. Fig. 12(b) shows the final results of this merging operation.

Finally, the grouping process that generates the path model provides a simple scheme to generate a probabilistic expectation of the route that will be taken by a pedestrian entering the scene at a particular location (if it is on a known route). If p_{ij} is the probability that an object will exit the scene from the node j, under the condition that it entered the scene from the node i, and if N_{ij} is the number of trajectories that have follow the route from node i to route j, then we can calculate the probability of possible exit nodes for a given entry node from the formula (Fig. 13):

$$p_{ij} = \frac{N_{ij}}{\sum_k N_{ik}}$$

Table 2 indicates these probabilities from the trajectory and route data used to create Fig. 12(b). So for some new object appearing at entry node 5, the probabilistic prediction is 0.73 that it will exit at node 8, and 0.27 that it will exit at node 6.

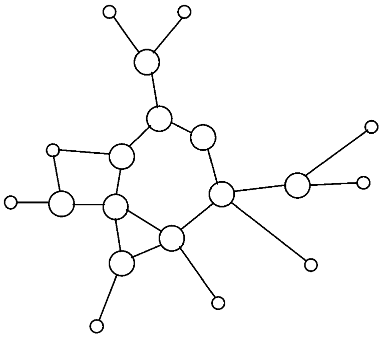

Fig. 13. Hand constructed graph of the network shown in Fig. 8(b). Leaf nodes (smaller circles) indicate entry–exit points, interior nodes (larger circles) are junctions.

Table 2
Probability of entry to exit node predictions estimated from frequency count of route usage

Entry node	Exit node probability									
	1	2	3	4	5	6	7	8	9	10
1			1.0							
2						0.17	0.58	0.25		
3	1.0									
4										1.0
5						0.27		0.73		
6		0.14			0.18				0.68	
7		1.0								
8		0.38			0.48					0.14
9						1.0				
10				0.89				0.11		

6. Conclusions—future work

This paper has demonstrated the practicality of building spatial models based on the analysis of trajectory data extracted from image sequences. The models have been shown to be valuable for economically encoding the route followed by an object in the scene, reducing the trajectory data down to a single label associated with each route. Although many surveillance tracking algorithms provide a local predictive step to aid the correspondence process in the next image frame, encoding the route and path data supports prediction over many time steps, and may be particularly useful for predicting across some types of occlusion in the scene (e.g. a parked vehicle).

The exit node predictions generated from the routes and paths are restricted by the number of trajectories available for learning and more reliable statistics would require much longer training periods (i.e. more trajectories). In fact, it is likely that we would need to partition the route learning into different time periods (e.g. each hour), as the statistics are not stationary over time.

The representation of the route models is based on sequences on knot points and linear interpolation is performed whenever is required. Although the accuracy of the results is satisfactory, we consider the use of cubic splines instead, that they will provide more accurate models.

The classification process presented in the results does classify trajectories to that fall outside the current learnt state of the model. However, to determine a reliable classification of such an event requires a longer time set of observations. We consider augmenting the probabilistic model of path usage to represent a Markov process, which encodes the track history.

The next step in this research will be to combine these models across multiple camera views, using a common world coordinate system like a ground plane coordinate system. One of the guiding reasons for using the spatial model is that the features that are extracted are easily identified with scene features, and when the models are combined across many views, this characteristic will substantially ease the integration of information.

Acknowledgments

The authors acknowledge financial support from the Engineering and Physical Sciences Research Council (EPSRC) under grant number GR/M58030. We are grateful to Ming Xu for supplying the trajectory data. Thanks for the helpful comments of the referees.

References

[1] J.E. Boyd, J. Meloche, Y. Vardi, Statistical Tracking in Video Traffic Surveillance, Proceedings of ICCV99, Corfu Greece, September, 1999.

[2] J.W. Davis, A.F. Bobick, The Representation and Recognition of Action Using Temporal Templates, IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97), 1997.

[3] J.H. Fernyhough, A.G. Cohn, D.C. Hogg, Generation of Semantic Regions from Image Sequences, Proceedings of ECCV96, Cambridge, England, April, 1996.

[4] W.E.L. Grimson, C. Stauffer, R. Romano, L. Lee, Using Adaptive Tracking to Classify and Monitor Activities in a Site, Proceedings of CVPR98, June, 1998.

[5] R.J. Howarth, H. Buxton, Analogical Representation of Spatial Events for Understanding Traffic Behaviour, Proceedings of the 10th European Conference on AI, Wiley, New York, 1992, pp. 785–789.

[6] N. Johnson, D.C. Hogg, Learning the Distribution of Object Trajectories for Event Recognition, Proceedings of BMVC95, Birmingham, England, 1995.

[7] N. Johnson, Learning object behaviour models, PhD Thesis, Leeds, England, September, 1998.

[8] R. Rosales, S. Sclaroff, 3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions, Proceedings of ICCV99, Corfu Greece, September, 1999.

[9] N. Sumpter, J. Bulpitt, Learning Spatio-temporal Patterns for Predicting Object Behaviour, Proceedings of BMVC98, Southampton, England, September, 1998.

[11] R. Tsai, An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, Proceedings of CVPR'86, June, IEEE, 1986.

[12] D. Makris, T. Ellis, Finding Paths in Video Sequences, BMVC 2001, Manchester, England, September, 2001.