

Finding Paths in Video Sequences

Dimitrios Makris and Tim Ellis
Information Engineering Centre
City University
Northampton Square
London EC1V 0HB, UK
d.makris@city.ac.uk

Abstract

This paper investigates the task of identifying frequently-used pathways from video sequences of natural outdoor scenes. Path models are adaptively learnt from the accumulation of trajectory data over many image frames. Labelled paths are used as an efficient means for compressing the trajectory data for logging purposes. In addition, the path models are used to predict the object's location many timesteps ahead, and to aid the recognition of unusual behaviour identified as atypical object motion.

Keywords: people tracking, learning paths, scene labelling, route detection, track prediction, video annotation, video surveillance

1 Introduction

In this paper we investigate the task of learning the routes or paths taken by pedestrians walking through outdoor scenes. The motivation for this objective is three-fold. Firstly, we require an efficient method to encode and annotate individual tracks to construct a log of movement patterns over long periods of time (e.g. weeks). Object trajectories will be assigned to one of only a small number of detected pathways, resulting in significant compression for the logged data. Secondly, the information can be used to support the tracking process giving the system the opportunity to predict forward many frames, based on the current location and direction. Finally, accumulating tracks over a long time period establishes a pattern of typical movements and this can support the recognition atypical or unusual movements.

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 to 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 re-appear. The justification for making this distinction between routes and paths 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, prediction and atypicality) we wish to associate trajectories with scene features and hence we maintain two representations.

The following section 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. The next section describes the models that will be generated for this research, and then a description of the implementation of the algorithms for building these models. Results are then presented for model building from image sequence taken from a campus site.

2 Previous work

To extract pedestrians routes requires a method that is able to cope with a wide range of inconsistent motions, often resulting from a variety of interaction: people avoiding each other at busy times; avoiding static objects in the pathway; or just the casual meandering of a lone pedestrian. The deviation from a straight-line 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). Figure 1 shows an example of a small number of trajectories (generated from the blob centroids) projected back into the image, and a motion histogram image [2] accumulated from a longer sequence of the same scene, showing regions which are cues for motion in the scene.



Figure 1a. Sample trajectories plotted in image plane and b) motion histogram image.

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 et al [6] used a neural network to learn typical motions of pedestrians through a scene. Many trajectories were accumulated to form a distribution in the image plane that was represented by a set of prototype vectors. This 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 un-typical instantaneous motions for detecting atypical trajectories, but does not satisfy the requirements for prediction or provide a mechanism for labelling the trajectories.

Sumpter and Bulpitt [8] adopted Johnson's methodology [6], adding feedback to the network to learn different patterns of activation. 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 propose a spatial model for representing routes in the image. A route describes the entire trajectory of an object from the time it enters the scene to when it exits and can be described as a curve with specific start and end points. Each route is modelled with a central spline axis (a set of equi-distant nodes that form a polyline) that represents some average of the route, bounded by two extrema, that bound the variation in trajectories sampled for the route (see figure 2). A route has two terminator nodes (start and end) that typically correspond to entry/exit points or regions in the image.

Specifically, each node i is characterized 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 $\mathbf{n}_i = [n_{x_i}, n_{y_i}]$, defined as the unit vector perpendicular to the local spline direction (defined by three consecutive nodes of the route)
- a set of observations across the route (along the normal vector). We use the extremes of this set, the left $\mathbf{l}_i = [l_{x_i}, l_{y_i}]$ and the right $\mathbf{r}_i = [r_{x_i}, r_{y_i}]$ which lie along the normal vector of the node.

The advantages of such a model are that the routes are closely tied to specific scene features and can be used to support spatial reasoning. The spline provides a simple representation for the route that is computationally simple and can be dynamically updated for on-line learning. Using the extrema for quantifying the width emphasises the spatial bounds for the routes and can cope with asymmetries.

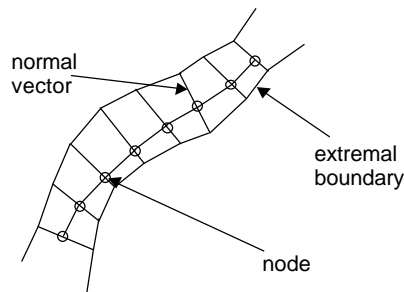


Figure 2. Spline and extrema representation of a route

The route models are learnt from example trajectories extracted from an image sequence of object motion. Trajectories are grouped using a geometrical analysis that compares the separation distance between a trajectory and an evolving route description.

A second level model derives a semantic description of scene components inferred from the routes based on paths. The paths have a similar geometric representation as the routes and are constructed by detecting junction regions where routes cross each other, or where sections of route lie closely parallel. We use a graph to represent the topology of the network of nodes (entry/exit points and junctions) for the paths.

4 Implementation

4.1 Route Learning

Routes are learnt by grouping sets of geometrically adjacent trajectories and the route description is stored in a database. The following section describes how trajectories are selected for grouping, the criteria for matching a new trajectory to a route, the updating 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 set of 2D coordinates corresponding to some specific point on the target, e.g. the centroid. To avoid learning with un-representative data, we eliminate short trajectories and trajectories of slowly moving objects whose direction changes frequently over short time periods.

A valid and suitable trajectory is then resampled over the space to normalize the trajectories of high and low speed objects and to counter the effects of perspective. The resampling process also generates smoothed trajectories. Next, the trajectory is compared to all existing routes in the database, according to a defined distance measure between a trajectory and a route. This distance measure computes the maximum distance between the trajectory and the route. 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 threshold. If the trajectory does not match to any existing route, a new route is created in the database, initialised with the trajectory data (see appendix).

When a trajectory matches a route, the route must be updated with the new trajectory. The update task is performed in three steps:

- i) node updating: each node is updated using the trajectory closest point coordinates (\vec{x}_t) and the weight factor w of the node:

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

In addition the weight factor w is incremented, the trajectory is checked if it is beyond the width distribution extremes and if so, the associated extreme become equal to the projection of the point on the node normal direction.

- ii) route extension: if there are trajectory points beyond the route terminator nodes, the route will be extended, using these points.

- iii) route resampling: to maintain equal distances between the route nodes, the route is re-sampled (using linear interpolation) after updating, and the normal vectors are re-calculated.

Following updating, the updated route is compared with all the other routes in the database to determine if it should in turn be merged. Merging compares the two routes along their total length computing their minimum separation and are merged if this is less than a defined threshold.

The route merging is similar to route updating with a trajectory. First, the route with the highest weight is selected as main and then this route is updating with the other (secondary route). Each node of the main route is updated with the closest node of the secondary route, but this time the weight factor (w_2) and the width distribution of the secondary route must be considered. The extremes of the merged route are calculated from the combination of the extremes of both routes and the weight and the node position is calculated as follows:

$$w_1' = w_1 + w_2 \quad x_1' = \frac{(w_1 * x_1 + w_2 * x_2)}{w_1 + w_2}$$

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 two parameters: a) the resampling factor between the route nodes and b) the distance threshold. It is sensible to set the distance threshold equal to or larger than the resample factor.

4.2 Deriving semantic description from routes

Although the route model allows us to distinguish different activities in the scene, it does not explicitly assign any semantics to the scene features. A semantic description of the scene would refer to paths and junctions. Paths and junctions can be derived by the set of the routes, by considering the following scenarios:

- i) Two or more routes are combined to create a path where they satisfy the route merging criteria (see section 4.1) over a sub-section of the route. A junction is formed at the location where the overlapped routes diverge.
- ii) When the routes cross each other, a junction is set at the cross area and each route is divided into two paths

5 Results

Figure 3a and 3b show three routes extracted from the trajectories of the sequence (table 1, routes 7, 5, 4). The routes were extracted from a video sequence (resolution 768x576) of 14 minutes, sampled at 2 frames per second, comprising 190 trajectories. Threshold values used for the trajectory and route merging were 60 pixels and the resampling value was 40 pixels. As can be seen, the spline axis and extrema give a compact representation of the pathways visible in the image and can be favourably compared to the MHI shown in figure 1b.

Figure 3b 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. Figure 3c shows the splines for the 9 principal routes detected in this sequence. Table 1 identifies the main routes extracted from the image sequence, showing 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. 3, 6 and 7) could be also potentially be discarded, or may require more training data to determine if they represent frequently-used routes. Six routes with a very low frequency of use have been discarded from the routes extracted from the sequence.

Route	Nodes	Weight	Usage
1	17	39.2	0.28
2	17	26.0	0.186
3	16	10.2	0.068
4	16	19.9	0.134
5	18	11.6	0.088
6	15	6.3	0.04
7	17	12.1	0.086
8	14	7.2	0.042
9	18	10.0	0.076

Table 1. Nine main routes extracted from sequence.

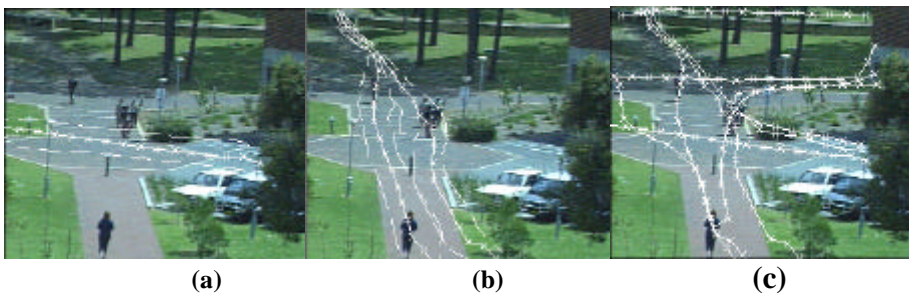


Figure 3. a) Route 7, b) routes 4 and 5, c) 'mean' trajectory of all major routes.

Figure 4 shows the result of applying the route model to a set of 53 previously unseen trajectories. The trajectories shown in figure 4(a-c) have been recognised and classified from the learnt models. Figure 4d) indicate trajectories not matched with existing routes. It can be seen, for example for trajectories in the top left of figure 4d), 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 a total 8 unknown routes from this set.

Figure 5a shows the result of applying path detection to the route database. Junctions have been detected to indicate where routes cross or converge (red unfilled circles). Entry and exit points are indicated by the blue circles. The threshold distance of 40 pixels is used (derived from the re-sampling distance) for an image size of 768x576. 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



Figure 4. a-c) Classification of trajectories using route model.
 d) un-classified routes not recognised by the system.

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. Figure 5b shows the final results of these merging operations.

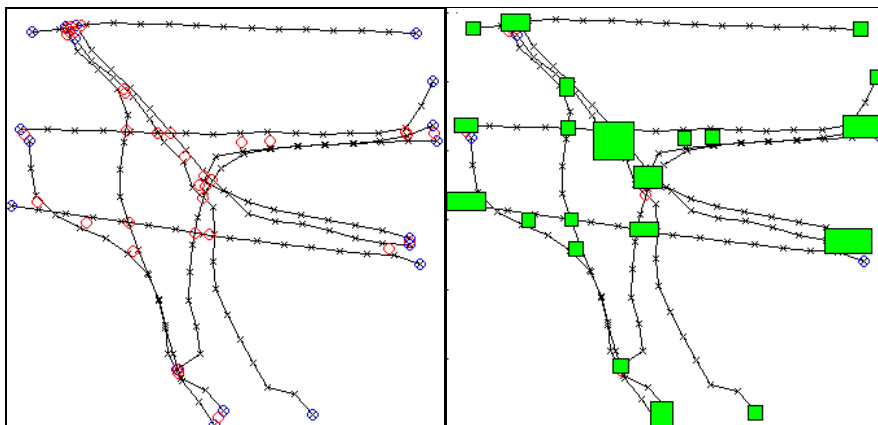


Figure 5a. Detected junction (red circles) and entry-exit points (blue circles).

Figure 5b. Green rectangles indicate the merged junctions and entry-exits.

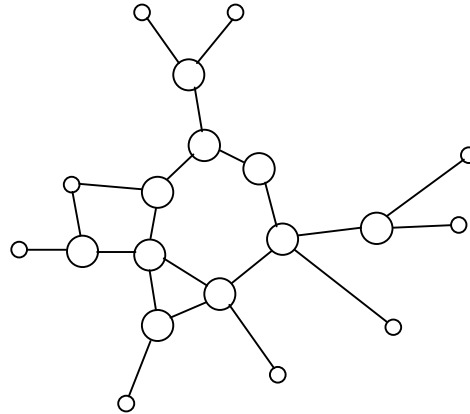


Figure 6. Hand constructed graph of the network shown in figure 5b. Leaf nodes (smaller circles) indicate entry-exit points, interior nodes (larger circles) are junctions.

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). The frequency usage statistics record the known destinations (exit points) for each entry. The table below indicates these probabilities from the trajectory and route data used to create figure 5b. 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.

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		

Table 2. Probability of entry to exit node predictions estimated from frequency count of route usage. (Note: nodes are numbered clockwise, starting from top left corner.)

6 Conclusions

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. 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. However, the spatial model is simple to update, and on-line adaptation to new trajectories is a straight-forward task.

We have not yet considered how that models will be used to identify atypical motions. Although the classification process presented in the results does classify trajectories that fall outside the current learnt state of the model, to determine a reliable classification of such an event really requires a longer time set. The next step in this research will be to combine these models across multiple camera views. 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.

Acknowledgements

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.

References

- [1] Boyd, J E, Meloche J and Vardi, Y “Statistical tracking in video traffic surveillance”, Proc. ICCV99, Corfu Greece, Sept. 1999.
- [2] Davis, J W and Bobick, A F, “The representation and recognition of action using temporal templates” IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97).
- [3] Fernyhough, J H, Cohn, A G, Hogg, D C “Generation of semantic regions from image sequences”, Proc. ECCV96, Cambridge, England, April 1996.
- [4] Grimson, W E L, Stauffer, C, Romano, R and Lee, L, “Using adaptive tracking to classify and monitor activities in a site”, Proc. CVPR98, June 1998.

- [5] Howarth, R J and Buxton, H “Analogical representation of spatial events for understanding traffic behaviour”, Proc. 10th European Conference on AI, John Wiley, pp785-789, 1992.
- [6] Johnson, N and Hogg, D C, “Learning the Distribution of Object Trajectories for Event Recognition”, Proc. BMVC95, Birmingham, England, 1995.
- [7] Rosales, R and Sclaroff, S, “3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions”, Proc. ICCV99, Corfu Greece, Sept. 1999.
- [8] Sumpter, N and Bulpitt, J “Learning spatio-temporal patterns for predicting object behaviour”, Proc. BMVC98, Southampton, England, Sept. 1998.

Appendix: Definitions of various distances

Distance of a point from a route node.

This is described by the following equation:

$$d = \|\mathbf{x} - \mathbf{x}_t\| - \min(\|\mathbf{l} - \mathbf{x}_t\|, \|\mathbf{r} - \mathbf{x}_t\|)$$

for trajectory coordinate \mathbf{x}_t and route node \mathbf{x} with extrema \mathbf{l} and \mathbf{r} .

Distance point from path.

If the path had been represented by a simple curve (spline), the distance of a point from the path would have been just the Euclidean distance from the curve, e.g. the length of the line segment that connects the point with the curve and it is perpendicular to the curve. But because our definition of the path includes not only a curve, but an associated width function along the curve, we should specially define this distance:

When the point is inside the path, his distance should be negative, and when it is outside positive and equal to the distance from the curve that is defined by the closest extrema, from the two extreme curves of the path. An approximation of this can be the minimum distance of the point from all the nodes of the path.

Distance of a node from trajectory can be defined as the minimum distance of the node from all the points of the trajectory.

Distance of trajectory from path: It can be defined as the maximum distance over all the distances of the trajectory points from the path.

Distance between two nodes of two different paths: it should be defined as the distance between the two closest extremes of the two nodes. It can be negative, in the case that the nodes are overlapped. An approximation of this distance can be estimated by the following equation:

$$d = \|\mathbf{x}_1 - \mathbf{x}_2\| - \min(\|\mathbf{l}_1 - \mathbf{x}_2\|, \|\mathbf{r}_1 - \mathbf{x}_2\|) - \min(\|\mathbf{l}_2 - \mathbf{x}_1\|, \|\mathbf{r}_2 - \mathbf{x}_1\|)$$

Distance of a node from another path is the minimum distance of the node from all the nodes of the other path.

Distance between one path from another is the maximum distance over the distances of the nodes of the first path from the second path. Because this definition is not symmetrical, the distance can be calculated as the mean average of the two distances.