Generation of Semantic Regions from Image Sequences

Jonathan H. Fernyhough, Anthony G. Cohn and David C. Hogg

Division of Artificial Intelligence, School of Computer Studies
University of Leeds, Leeds, LS2 9JT
{jfern,agc,dch}@scs.leeds.ac.uk

Abstract. The simultaneous interpretation of object behaviour from real world image sequences is a highly desirable goal in machine vision. Although this is rather a sophisticated task, one method for reducing the complexity in stylized domains is to provide a context specific spatial model of that domain. Such a model of space is particularly useful when considering spatial event detection where the location of an object could indicate the behaviour of that object within the domain. To date, this approach has suffered the drawback of having to generate the spatial representation by hand for each new domain. A method is described, complete with experimental results, for automatically generating a region based context specific model of space for *strongly* stylized domains from the movement of objects within that domain.
Keywords: spatial representation, scene understanding.

1 Introduction

Event recognition provides a significant challenge for high-level vision systems and explains the impetus behind the work described in this paper. Nagel (1988) outlines several previous applications that connect a vision system to a natural language system to provide retrospective descriptions of analysed image sequences. Typically the vision system is used to provide a 'geometric scene description' (GSD) containing a complete description of the spatial structure within the domain (i.e. the area in view of the camera) and the spatial coordinates of the objects in the scene at each instance of time. A generic event model (Neumann & Novak 1983), characterizing a spatio-temporal representation for that event, can be matched against the GSD in order to recognize instances of that event which can then be expressed in natural language.

More recent work demonstrates a simultaneous analysis of image sequences to provide the incremental recognition of events within a football game (Retz-Schmidt 1988, André, Herzog & Rist 1988). This enables the system to provide a running commentary of the actions within the domain including perceived intentions. A model of the world representing the static background of the scene is supplied manually so that the system can recognize situated events, for example realizing the difference between passing the ball and attempting to score a goal.

Although not necessary for *all* event recognition tasks, a spatial model providing a context specific representation of the domain is certainly beneficial. In

strongly stylized domains, such as road traffic environments where vehicles' movements are governed by strict constraints, a spatial model containing semantic information would allow the interpretation of object behaviour from the sequenced position of objects within the domain, for example areas where vehicles turn or where pedestrians cross the road. Fig. 1 shows an example to illustrate how a context specific region based model of space can be used to facilitate the recognition of a vehicle waiting to turn right. The region occupied by the vehicle in fig. 1*b* is an area of behavioural significance representing the location where vehicles must await oncoming traffic before turning right.

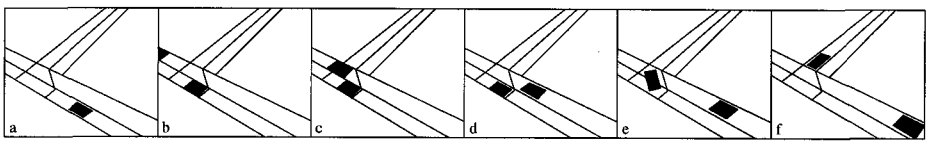

Fig. 1. A simplified spatial model of a road junction showing a sequence of object locations. A vehicle approaches a junction (*a*), reaches it (*b*) and then awaits oncoming traffic (*c* & *d*) before turning right into the new road (*e* & *f*).

Howarth & Buxton (1992) introduced such a spatial model for spatial event detection in the domain of traffic surveillance. This representation of space is a *hierarchical* structure based on *regions,* where a region is a spatial primitive defined as a (closed) two-dimensional area of space with the spatial extent of a region controlled by the continuity of some property.

There are two kinds of regions:

- *Leaf regions* are the finest granularity of region. They are areas of space that tile the entire scene and do not overlap. Leaf regions are used to structure space and are completely defined by how *composite regions* overlap.
- Concatenations of adjacent leaf regions form *composite regions* expressing areas sharing the same significance, for example region types (i.e. roads and footpaths) and regions with similar behavioural significance (i.e. give-way zones.) It is possible for different composite regions to share leaf regions (i.e. they may overlap) providing the hierarchical structure to the spatial layout. In terms of the domain context, a composite region represents the area described by the movement of objects within the domain (i.e. a *path.*)

Howarth (1994) produced such representations of space manually for each new domain: a time consuming and painstaking process. This paper demonstrates a method to generate such a spatial structure automatically for strongly stylized domains through the monitoring of object movement over extended periods.

Li-Qun, Young & Hogg (1992) describe a related method of constructing a model of a road junction from the trajectories of moving vehicles. However, this

deals only with straight road lanes and is unable to handle the fine granularity of region required for a detailed behavioural analysis – such as regions where a vehicle turns left. The method described here is not limited in this way.

Johnson & Hogg (1995) demonstrates a related approach in which the distribution of (partial) trajectories in a scene is modelled automatically by observing long image sequences.

2 Outline of the Method

The system accepts live video images from a static camera to produce shape descriptions corresponding to moving objects within the scene. This dynamic scene data is then analysed, in real-time, to build a database of paths used by the objects, before being further processed to generate the regions required for the spatial model. A diagram outlining this system is shown in fig. 2.

There are three main stages:

- A *tracking* process obtains shape descriptions of moving objects (§3).
- *Path generation* builds a model corresponding to the course taken by moving objects and subsequently updates the database of paths (§4).
- *Region generation* accesses the database of paths so that leaf and composite regions can be constructed for the spatial model within the domain (§5).

Latter sections will provide implementation details and results for the test domains as well as a discussion of future applications.

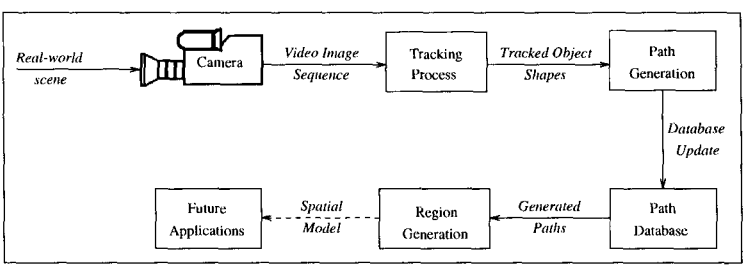

Fig. 2. Overview of the system

3 Tracking

The first step in automatically generating the spatial representation is the analysis of *dynamic scene* data. Visual information is provided through live video images from a static camera. The current test domains include an elevated view of a busy junction containing both pedestrians and vehicles (fig. 5*a*) as well as a predominantly pedestrian scene (fig. 5*b*). A list of objects is provided on a

frame by frame basis using the tracking process described in Baumberg & Hogg (1994*b*). A combination of background subtraction, blurring and thresholding is used to obtain object silhouettes for each frame. The outline of each silhouette is then described by a number of uniformly spaced control points for a closed cubic B-spline and assigned a label by considering object size and proximity in the previous frame. Although this method does not handle occlusion and is not particularly robust, it provides sufficient information for our purposes and it proves significantly faster than the active shape model described in Baumberg & Hogg (1994*a*).

4 Path Generation

A *path* is defined as the course that an object takes through the domain. More specifically, a path is represented by all pixels covered by the object along its course through the domain. To enable real-time processing from the tracking output and to reduce storage requirements, a list of active paths is maintained from frame to frame. With each new frame, the latest location of each object is combined with its respective existing active path.

Object location can be taken just from the outline of the object provided from the tracking process. However, these outlines are subject to various forms of noise. In particular, light reflections can alter the object silhouette dramatically (fig. 3*a*.) When combined, such object locations may produce a jagged path (fig. 3*b*).

Under ideal conditions, an object moving along a straight line trajectory will produce a convex path (except possibly at the ends) and although an object with a curved trajectory will obviously not have a convex path it will be 'locally convex'. The state of a path becomes important during database update – two objects following the same course should have approximately the same path which may not be the case without preprocessing them. Image smoothing techniques (such as averaging or median smoothing) enhance the condition of the path by filling in some of the gaps. However they are, in real-time terms, computationally expensive.

Instead of using smoothing techniques, path condition is enhanced by generating the convex hull of the object outline (fig. 3*c*.) Such calculations are *not* computationally expensive – the convex hull of any polygon can be found in linear time, $O(n)$ (see Melkman 1987). Although Baumberg & Hogg's (1994*b*) tracking program supplies a cubic B-spline representation of the object outlines, it is relatively easy to convert them to a polygonal representation (Sonka, Hlavac & Boyle 1993, Chapter 6.2.5, pp. 212–214).

The convex hulls combine to give a significantly smoother path (fig. 3*d*,) that is more likely to be correctly matched during database update.

Once an active path becomes complete it is merged into the database of existing paths. There are two possibilities when merging a new path into the database :

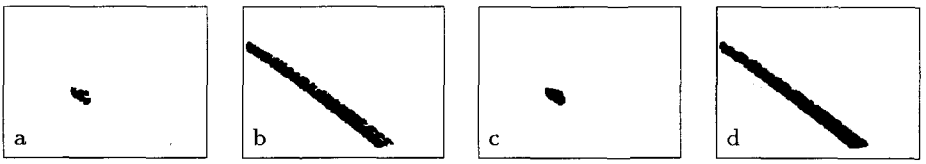

Fig. 3. (a) Object outline, (b) Path generated using object outline, (c) Convex hull of object outline, (d) Path generated using convex hull of object outline.

- an equivalent path already exists and should be updated to accommodate the new path.
- no equivalent path is found and the new path should be allocated a unique identity.

Equivalence is based on the percentage overlap between the new path and the paths contained within the database. *Path overlap* occurs when the constituent pixels of two paths coincide. Two paths are considered to be equivalent if a specified proportion of their paths overlap. When the specified percentage overlap is too low it is possible that two different paths will be found equivalent – for example, two adjacent road lanes may be matched and seen as just one wide lane. Alternatively, if the overlap is too high there may be no equivalences identified within a satisfactory time scale. Experimental results within the test domains have shown that a tolerable compromise appears to be an overlap of 80% – this allows a sufficient duration for the training period without undesirable equivalences being identified. Of course, this value is scene specific and will be discussed more in § 6.

When updating the database, a new path could be merged with an existing database path using a function analogous to the binary *or* operation – the value of each pixel representing a database path would indicate if any equivalent path has occupied that pixel. However, the update function used is analgous to arithmentic *addition* – allowing the value of each pixel for a database path to indicate the number of equivalent paths sharing that pixel.

At the end of the training period, each path held in the database will contain frequency distribution information for that path, fig. 4*a*. This representation has two benefits :

- "noise" can easily be identified from low distribution areas.
- it is possible to extract the most "common" path by thresholding the distribution, fig. 4*b*.

5 Region Generation

At *any* time during the training period it is possible to generate regions for the spatial model. Effectively this halts the database generation process (although it may be resumed) and uses that information to build the regions. A new region

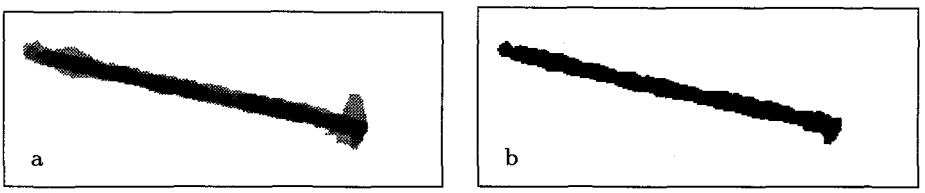

Fig. 4. (a) Path displaying a grey scale representation of the frequency distribution (b) Path obtained from most "common" usage

model can be created during the path generation stage each time a path becomes complete and is merged into the database. However, it is unclear how useful this continuous region generation may be as the model is constantly changing and the most recent state is unlikely to have any connection to the previous state.

When regions are generated only as required, path verification may also be accomplished. Each database path is tested against all other paths in the database to verify that no path equivalences have been created through the database update process – the merging of equivalent paths may alter the original shape enough that a previously unmatched path may now be found equivalent. Should any 'new' equivalences be discovered they are merged together as before.

Although this step is not entirely necessary, it has the advantage that a previously "weak" path may be strengthened by a 'new' equivalence. Without this operation, such paths will be strengthened with extra training – essentially, this step allows a shorter training period and as such provides an advantage over continous region generation.

To reduce "noise", any path with a uniformly low frequency distribution is discarded. Although low frequency distribution may represent infrequent object movement rather than "noise", it is also possible that abnormal or unusual behaviour is being displayed. In some applications this information may be useful; however, the method described in this paper relies on behavioural evidence and it is safe to reject these paths as they are not statistically frequent enough.

The remaining paths are then processed to obtain a binary representation of the 'best' or most 'common' route used – this depends on the database path update function being 'addition' rather than 'or' (see previous section). Thresholding is used to provide a binaray representation where the threshold is selected from the *cumulative* frequency histogram of each database path and the percentage overlap value employed in the test for path equivalence. An 80% overlap value is required to merge a path into the database and indicates the percentage of pixels shared by equivalent paths. This is reflected in the cumulative frequency histogram where the 'common' path forms the highest 80% of the histogram. So, the frequency value found at 20% of the histogram provides the value for the threshold operation.

These binary path representations express the composite regions for the spatial model – they describe each area of similar behavioural significance from objects following the same course through the domain. From §1, the leaf regions

can be completely defined by how the binary path representations overlap. Each binary path is allocated a unique identification before being added to the region map. Overlapping segments form separate leaf regions and are reassigned a new unique identification. When all the paths have been processed each *leaf region* will have been identified and labelled.

Occasionally, adjacent paths may share small areas of common ground – perhaps from shadows or the occasional large vehicle. This can generate very small regions that are not actually useful and the last step in leaf region generation is to remove such small regions by merging them with an adjacent region – selected by considering the smoothness of the merge. Smoothness is checked by considering the boundary of the small region and the proportion shared with the adjacent leaf regions. The adjacent region sharing the highest proportion of the small region's boundary is selected for the merge, e.g. if the small region has a border length of seven pixels and shares five with region A and only two with region B, the combination with region B would form a 'spike' whereas region A may have a 'local concavity' filled and subsequently be smoother. Fig. 5 displays the leaf regions obtained for the test domains.

To complete the spatial model, it is necessary to discover the union of leaf regions which make up each composite region (based on the binary representations of the database paths.) A complication in this process results from the previous merge of small "useless" regions which may now be part of a larger leaf region that should not be a member of the composite region for the path under consideration. Each composite region should contain only those leaf regions that are completely overlapped by the path it represents. A selection of composite regions is displayed in fig. 5 along with the identified leaf regions.

6 Experimental Results

The tracking program processes about 5 frames/second on a regular UNIX platform. The video image sequence used for the traffic junction is about 10 minutes in length and averages 5 or 6 objects each frame. In comparison, the pedestrian scene is roughly double the length with at most 3 objects in any frame and often with periods of no object movement.

At the end of the training period the traffic junction has entered 200 paths into the database which reduces to 70 after checking for equivalences. Of these paths, 28 prove frequent enough to be used in region generation so giving 28 composite regions and initially over 400 leaf regions. The removal of small regions reduces this number to around 150. After only 2 minutes, many of the significant routes have already been identified with 16 paths strong enough to be considered composite regions and generating a total of 87 leaf regions. For the pedestrian scene about 120 leaf regions are generated from 23 recognized paths.

These results rely on three threshold paramaters we were unable to eliminate from the system. Thresholds remain necessary for the overlap value in the path equivalence test, the actual threshold operation and the size of leaf regions that are to be merged into an adjacent region. As previously indicated, the overlap

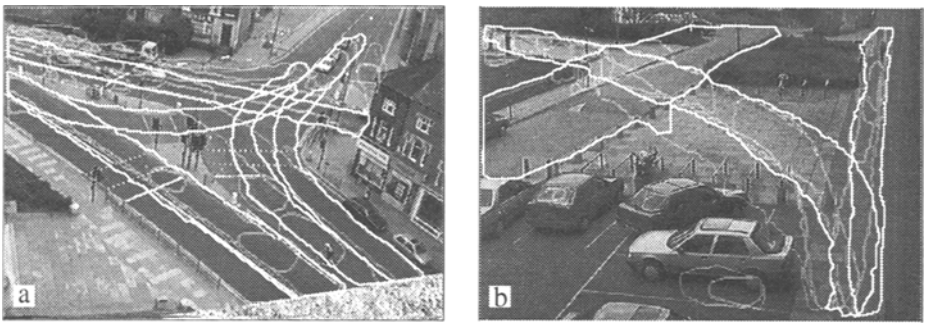

Fig. 5. (a) Road junction and (b) pedestrian scene displaying identified leaf regions along with a selection of composite regions.

value for path equivalence and the threshold operation are linked – one being the dual of the other. Experimental results indicated that an overlap value of 80% was suitable for both test domains. It is possible that the percentage overlap value is related to the camera angle for the scene. As the angle is reduced, objects in adjacent lanes will naturally overlap more. This means that when attempting the path equivalence test a higher overlap percentage value will be needed to distinguish equivalent paths from those that are actually adjacent lanes. The value used to determine small regions is passed on from the tracking program – here the minimum tracked object size is 10 pixels otherwise problems can arise. Ten pixels is less than 0.02 percent of the total image area size which does not seem too excessive.

7 Further Work / Work in Progress

The process as described is real-time as far as the training period is concerned and is able to generate the regions at any time during the training sequence. However, once generated the spatial model becomes a static entity which may cause problems in a changing world. For instance, if the model is used for traffic surveillance and road works subsequently alter traffic flow, the spatial model becomes inaccurate. In such situations it is desirable to have an adaptive model of space that is able to learn during use. Such an adaptive model may also prove useful for robot navigation with a non-static camera where the domain is constantly changing and in need of updating. It should be possible to enhance the method described here to provide an adaptive model of space.

The driving force behind the development of this technique to automatically generate semantic regions for a scene was the desire to provide a spatial model to assist event recognition procedures. Dynamic scene analysis has traditionally been quantitative and typically generates large amounts of temporally evolving data. Qualitative reasoning methods (Zimmermann & Freksa 1993, Cui, Cohn & Randell 1992) would be able to provide a more manageable way of handling

this data if a formal framework for the given situation exists. The spatial model described here, being topologically based, is able to provide such a qualitative formal framework. This allows generic event models to be provided using qualitative logic descriptions. Typically such generic event models are provided as part of the *a priori*. However, it is our intention to provide a method to determine these event models automatically from a statistical analysis of object location, movement and the relationships to other objects.

This representation of space could provide control for a tracking process by reducing the search space for moving objects – the spatial representation contains the paths followed by objects. The spatial model could also identify the potential location of new objects in the scene, again reducing the search space.

Other possible areas where such a spatial layout could be used are stereo image matching and fusing of multiple overlapping views. The topology of the spatial model is largely invariant to small changes in the viewing angle and provides sets of corresponding regions.

8 Conclusion

By using an existing tracking program that produces (2D) shape descriptions for tracked objects from a real image sequence, we have demonstrated an effective method for the real-time generation of a context specific model of a (2D) area of space. The domain is required to be strictly stylized for this method to be suitable; for example in the traffic surveillance domain there is typically a constrained set of possibilities for the movement of vehicles. This may not be the case for less stylized domains like the movement of fish in a pond. However, the extent of such stylized domains is sufficiently frequent for the method to be widely applicable.

The spatial model can be considered to be "data-centered" due to its construction from real image data. This means that an alternative tracking application could be used that provides object outlines projected onto the ground plane rather than the image plane to produce a spatial model representing a ground plane projection of the viewed scene which could prove useful[1]. Howarth & Buxton (1992) use a ground plane projection of the image plane to "better facilitate reasoning about vehicle interactions, positions and shape." Similarly, by using a tracking process that provides 3D shape descriptions the method would require relatively few changes to provide a complete 3D spatial model.

Previous contextually relevant spatial models have been generated by hand and as a consequence the domain is subject to human interpretation and occasionally misconception so the generated spatial model may not be entirely accurate. Our method relies only on observed behavioural evidence to describe the spatial model. As long as a sufficiently broad representation of object behaviour occurs throughout the training period the derived spatial model should be accurate without being prone to any misconceptions.

[1] A ground plane projection could also be obtained by back projection of the derived spatial model.

Statistical analysis allows the most used routes to be extracted from the database. This means that the length of the training period depends on the volume of object movement as well as representative object behaviour – for a quiet scene, a much longer image sequence will be necessary than with a busy scene. As long as the image sequence is of a sufficient length and demonstrates typical behaviour it is possible to obtain a reasonable representation of a (2D) area of space that is contextually relevant to the viewed scene.

References

André, E., Herzog, G. & Rist, T. (1988), On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer, *in* 'Proc. ECAI-88', Munich, pp. 449–454.

Baumberg, A. M. & Hogg, D. C. (1994*a*), An efficient method for contour tracking using active shape models, *in* 'IEEE Workshop on Motion of Non-rigid and Articulated Objects', I.C.S. Press, pp. 194–199.

Baumberg, A. M. & Hogg, D. C. (1994*b*), Learning flexible models from image sequences, *in* 'European Conference on Computer Vision', Vol. 1, pp. 299–308.

Cui, Z., Cohn, A. & Randell, D. (1992), Qualitative simulation based on a logical formalism of space and time, *in* 'Proceedings of AAAI-92', AAAI Press, Menlo Park, California, pp. 679–684.

Howarth, R. J. (1994), Spatial Representation and Control for a Surveillance System, PhD thesis, Queen Mary and Westfield College, The University of London.

Howarth, R. J. & Buxton, H. (1992), 'An analogical representation of space and time', *Image and Vision Computing* **10**(7), 467–478.

Johnson, N. & Hogg, D. (1995), Learning the distribution of object trajectories for event recognition, *in* D. Pycock, ed., 'Proceedings of the 6th British Machine Vision Conference', Vol. 2, BMVA, University of Birmingham, Birmingham, pp. 583–592.

Li-Qun, X., Young, D. & Hogg, D. (1992), Building a model of a road junction using moving vehicle information, *in* D. Hogg, ed., 'Proceedings of the British Machine Vision Conference', Springer-Verlag, London, pp. 443–452.

Melkman, A. V. (1987), 'On-line construction of the convex hull of a simple polyline', *Information Processing Letters* **25**(1), 11–12.

Nagel, H. H. (1988), 'From image sequences towards conceptual descriptions', *Image and Vision Computing* **6**(2), 59–74.

Neumann, B. & Novak, H.-J. (1983), Event models for recognitions and natural language description of events in real-world image sequences, *in* 'Proceedings of the Eigth IJCAI Conference', pp. 724–726.

Retz-Schmidt, G. (1988), A replai of soccer: Recognizing intentions in the domain of soccer games, *in* 'Proc. ECAI-88', Pitman, Munich, pp. 455–457.

Sonka, M., Hlavac, V. & Boyle, R. (1993), *Image Processing, Analysis and Machine Vision*, Chapman & Hall.

Zimmermann, K. & Freksa, C. (1993), Enhancing spatial reasoning by the concept of motion, *in* A. Sloman, D. Hogg, G. Humphreys, A. Ramsay & D. Partridge, eds, 'Prospects for Artificial Intelligence', IOS Press, pp. 140–147.