Generation of Qualitative Spatio-temporal Representations from Visual Input

by

Jonathan Hedley Fernyhough

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Computer Studies
January 1997

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Jon Fernyhough
School of Computer
Studies
University of Leeds

Doctor of Philosophy
January 1997

Generation of Qualitative Spatio-temporal Representations from Visual Input

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. An algorithm, complete with experimental results, is described for the automatic generation of a hierarchical region based context specific model of space for *strongly* stylized domains from the observation of objects moving within that domain over extended periods.

The highest (hierarchical) level of region describes areas of behavioural significance or the paths followed by moving objects. An extension to the region generation algorithm allows these regions to be further sub-divided into *equi-temporal regions* (where it takes an object approximately the same time to traverse each sub-division) that can be used by an attention control mechanism to identify interacting objects.

By using a region based model, it becomes possible to convert the quantitative object locations into *qualitative* locations which then enables the use of the rich family of qualitative logics for real-world surveillance. To demonstrate the effectiveness of the spatio-temporal model combined with qualitative object representations, an event learning strategy is demonstrated that allows the automatic generation of contextually relevant event models, which are usually provided as part of the *a priori* system knowledge.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:-

Fernyhough J.H., "Context Specific Models of Space", Proceedings Cosit'95 - Doctoral Consortium, Semmering, Austria, September 1995.

Fernyhough J.H., Cohn A.G. and Hogg D.C., "Generation of Semantic Regions from Image Sequences", Proceedings European Conference on Computer Vision, Cambridge, UK, April 1996.

Fernyhough J.H., "Qualitative Reasoning for Automated Traffic Surveillance", Proceedings Tenth International Workshop on Qualitative Reasoning, Stanford Sierra Camp, California, May 1996.

Acknowledgments

First, I would like to thank my supervisor Tony Cohn for introducing me to QSR as well as for all his help, insight, enthusiasm and encouragement over the course of my research. I would also like to thank David Hogg for all his help and advice. If not for both Tony and David I would never have completed this work.

I would like to express my gratitude to all the members of the AI lab at Leeds. In particular, Adam for all his help on Computer Vision and Neil who was always willing to listen and comment on my ideas as well as proof-read my work.

My gratitude belongs to each member of the SpaceNet project. After discussions at each workshop I found new directions to explore.

I would also like to thank Richard Howarth for his advice and comments on an earlier draft of this thesis.

Finally, I wish to thank my friends and family who have helped me to complete my academic research with all their support and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The work described in this thesis was motivated by the desire to apply qualitative reasoning techniques to real-world situations. Qualitative knowledge refers to that aspect of knowledge which *critically* influences decisions. The situation determines which aspect of knowledge is critical, and the key to qualitative reasoning relies on the relevance of the knowledge being modelled. Using such relevant qualitative knowledge, complete with an appropriate reasoning system, enables a computer to conduct a behavioural analysis of real-world situations in a qualitative manner. A rich selection of qualitative representation and reasoning systems already exist, although there are relatively few real-world applications.

Humans (and other animals) tend to rely on visual stimuli when interacting or observing actions in the real-world. However, the information provided from existing tracking applications is, by nature, quantitative with the position and spatial extent of dynamic scene objects usually provided in screen coordinates. For the qualitative behavioural analysis we are interested in, the exact location is not required. Using the approximate zone or *region* rather than the exact location will collapse broadly similar behaviours into equivalence classes to provide a generic model. Unfortunately, it is not possible to arbitrarily segment a scene into regions — such regions should be conceptually relevant to the physical structure of the domain.

Although an appropriate spatial model has been located (Howarth & Buxton 1992*a*), such representations have, to date, been generated by hand. Our first intention is to demonstrate an effective learning strategy that can automatically generate a similar spatial representation from the observation of object movements over extended periods.

Following the success of our spatial representation learning strategy we turn our attention to qualitative visual surveillance. With a conceptually relevant representation of space, it becomes possible to determine abnormal behaviour patterns from the continued observation of objects travelling within the domain. The spatial model is obtained from the statistical evidence of observed behaviours in which the quantity of "normal" behaviour is significantly greater than that of abnormal behaviour. Thus, the locations where "abnormal" behaviour have occurred during the learning cycle should not adversely affect the spatial model construct. Should any unusual behaviour occur after the training period (for example, a motor-way crash) the default behaviour and movement of domain objects may change radically indicating an unusual situation.

However, visual surveillance is not just concerned with abnormal behaviour patterns. To conduct a full behavioural analysis, the system has to be capable of recognizing (and interpreting) interesting situations. Typically, systems designed to recognize sequences of situated actions (*events*) are provided with *a priori* system knowledge of event models that can be used to recognize instances of particular events. When analyzing a dynamic scene for objects involved in a particular event, an attentional control mechanism can assist in determining interacting objects which are usually found within the same vicinity. Rather than providing event models as *a priori* system knowledge we propose an event learning strategy that employs our own attention control mechanism to identify potentially interacting objects.

1.1 Approach Taken

Our method to automatically generate semantic regions relies on the analysis of objects moving within the domain. We employ an existing tracking application that provides the position and spatial extent (shape descriptions) of moving objects as well as associating each object with its own label (which is maintained throughout the period the object remains within the scene). The domains of interest are typically natural outdoor scenes (for example, see figure 1.1) where the movement of objects within the domain is strictly stylized (i.e. domains in which objects tend to comply with a number of default behaviours, like the movement of vehicles on a road which follows rules according to the Highway Code).

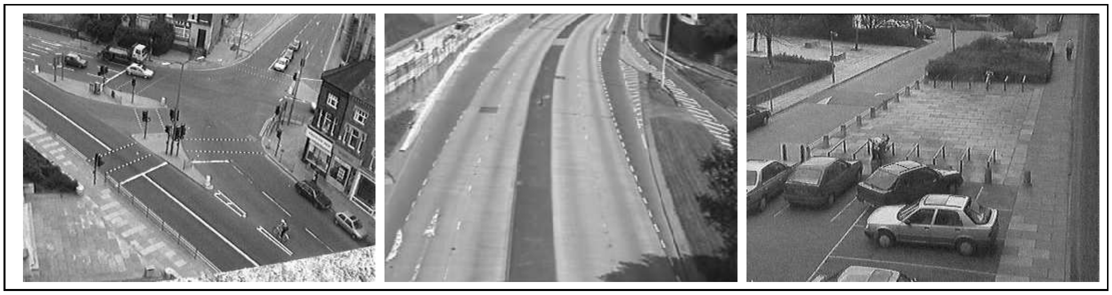


Figure 1.1: Example of test domains viewed from a static camera.

Dynamic scene data is used to construct a database of paths used by objects travelling through the scene. Statistical analysis indicates which entries are too infrequent to be included in the spatial model. Leaf regions for the spatial model are obtained from the combination of the remaining paths stored in the database. A previous (less successful) approach generated a mapping of the scene representing the frequency and distribution of all object movements over the training period. The intention was to employ traditional (image) segmentation techniques on the scene mapping to obtain the desired region model. Although the method did not provide sufficiently accurate results, it did indicate a number of shortcomings that assisted in the design of the second method.

Some form of attentional control mechanism is often employed in visual surveillance applications to identify interesting objects — it is not necessary (or practical) to examine every pair of objects in the scene. By using an attention control mechanism it becomes possible to focus on a more limited subset of those object pairs. This thesis is no exception, although our approach is somewhat different. The basis of our attention control mechanism relies on extending our spatial model to incorporate temporal information. When we construct the database of paths used by objects travelling through the scene we also incorporate point coordinates at regular time intervals that can be used later to form regions which sub-divide the composite regions within the spatial model into *equi-temporal regions (ETRs).* The spatial extent of an ETR is controlled by the velocity of objects as well as the distance from the camera (i.e. size due to camera perspective). However, the main feature is that it takes approximately the same time for an object to traverse each ETR in a composite ETR path. If the (approximate) time between two objects is known then "close" objects can easily be identified. Essentially, that is how the attention control mechanism functions.

To demonstrate the effectiveness of the spatio-temporal model, we present a qualitative event learning strategy (in contrast to the usual method of providing event models as *a priori* system information) that uses the contextually relevant features of the spatio-temporal model. Using the attention control mechanism "close" objects are identified and the qualitative relationships for relative position and relative direction of motion are maintained in object relationship history lists. When an object leaves the domain the associated history lists are verified and added to a database. On completion of the training period, the database can be statistically analyzed to determine which sequences of relationships occur sufficiently frequently to be considered as the basis for an event model.

1.2 Overview of the thesis

In this introduction we have provided a broad outline of the research conducted as part of this thesis. The next chapter (chapter 2) provides a review of the related bodies of work concerning qualitative reasoning methods, obtaining conceptual descriptions from the observation of moving objects over extended periods, as well as providing a brief overview of various machine learning paradigms. In the remaining chapters, we describe the original work of the thesis including relevant results from real image sequences. The work is organized as follows:

- **Chapter 3**

  We describe an existing region-based model of space that supports the behavioural analysis of objects moving through the domain. Two methods for automatically constructing a similar spatial representation are discussed — one being more successful than the other.

- **Chapter 4**

  An extension to our method for generating a semantic region-based model of space to include temporal sized sub-divisions (or regions) is demonstrated. Such temporal regions support an attentional control mechanism that allows objects within the same general vicinity to be identified.

- **Chapter 5**

  To demonstrate the effectiveness of our spatio-temporal model, details of a qualitative event learning system are provided and supported by experimental results.

- **Chapter 6**

  Finally, conclusions and aspects of possible future work are discussed.

Chapter 2

Overview of Related Work

2.1 Introduction

This chapter deals with the underlying foundations found in previous related work. It has been our intention to learn conceptual knowledge automatically from the input of video image sequences. In fact, we take this a step further and desire not just conceptual knowledge learning but to learn *qualitative* conceptual knowledge from dynamic scene data. Qualitative knowledge may be viewed as that aspect of knowledge which critically influences decisions. The particular aspect of knowledge which is critical depends on the situation, and the key to qualitative representation and reasoning relies on the relevance of the knowledge being modelled. Given the relevant knowledge and an appropriate reasoning system it becomes possible for a computer to predict, diagnose and explain physical behaviour of real-world situations in a qualitative manner, even when a quantitative description is unavailable or computationally intractable. A wide range of qualitative representation and reasoning systems now exist in this arena for both temporal and spatial aspects. We provide a review of the various systems in section 2.2. To show the type of conceptual knowledge we want to learn, we provide an overview of the existing vision systems in section 2.3 along with a brief overview of traditional machine learning techniques in section 2.4.

2.2 Aspects of Qualitative Reasoning

As discussed in Chapter 1, the work described in this thesis is motivated by the desire to apply qualitative reasoning techniques to real-world situations — in particular, we wish to apply such techniques to visual surveillance. Before we are in a position to do that, it is necessary to understand what qualitative representation and reasoning systems are currently available and the intended purpose of those system.

This section explores the rich set of existing qualitative representation and reasoning systems currently available for spatial and temporal reasoning. We start this review looking at temporal reasoning systems (section 2.2.1) before moving onto topological systems (section 2.2.2), Orientation or Direction systems (section 2.2.3), Size and Distance systems (section 2.2.4) and finally qualitative systems designed to deal with shape (section 2.2.5). Any similarities between the various systems are indicated throughout the text and we will finish the section with a summary.

2.2.1 Temporal

The representation and reasoning about temporal knowledge has been of great interest to researchers within Artificial Intelligence. Probably the most widely used and best known representation scheme is the algebra of temporal intervals proposed by Allen (1983). The simplicity and ease of implementation make this scheme particularly appealing.

Given any two complete intervals, Allen shows that there are only thirteen distinct relationships which precisely characterize the relative endpoints of the two intervals (see table 2.1). Disjunctions of these simple relations allow for some vagueness in modelling temporal event structures (e.g. G overlaps or starts M).

The reasoning mechanism is provided through table look-up with a composition table[1] which shows the possible relations between two intervals (X and Z) when the relationship between X and Z is known for a third interval, Y. For example, given

[1] Allen originally used the term *transitivity* table, but since more than one relation is involved the table represents relation composition rather than transitivity.

Relation	Symbol	Example
X *before* Y	$<$	XXX
Y *after* X	$>$	YYY
X *meets* Y	m	XXX
Y *met-by* X	mi	YYY
X *overlaps* Y	o	XXX
Y *overlapped-by* X	oi	YYY
X *starts* Y	s	XXX
Y *started-by* X	si	YYYYY
X *during* Y	d	XXX
Y *contains* X	di	YYYYY
X *finishes* Y	f	XXX
Y *finished-by* X	fi	YYYYY
X *equals* Y	$=$	XXX
		YYY

Table 2.1: The thirteen possible simple interval relationships.

that $X < Y$ and $Y\ d\ Z$ the composition table shows that $X\ \{<\ \ o\ m\ d\ s\}\ Z$. Using Allen's constraint propagation algorithm such inferences can be propagated through an entire temporal event network. However, Allen shows that this algorithm is incomplete and suggests that to ensure total consistency the computational complexity of such an algorithm would be exponential. Further work by Vilain, Kautz & van Beek (1990) shows this to be accurate and discusses an alternative algebra based on continuous end-point uncertainty. The restricted algebra is that subset of the interval algebra which can be completely encoded as disjunctions of continuous time point relations between the end-points of intervals. (i.e. disallows such disjunctions as {*before after*}). Both Nebel & Bürckert (1994) and Ligozat (1994) also analyze the maximal tractable subclasses of Allen's interval relations.

Kumar & Mukerjee (1987) re-interpret Allen's interval algebra using a state-based approach where the interpretation of the relations is viewed as propositions that hold at certain instants. This approach permits incomplete temporal intervals to be modelled (see table 2.2) and using state transition rules it becomes possible to determine the actual relationship as one of the two events terminate (i.e. on-line interpretation). In chapter 5, we show how this state-based approach can help identify learned events.

Relation	**Symbol**	**Example**
X *starts before* Y	sb	XXXXX???
Y *starts after* X	sb	YYY??
X *starts with* Y	sb	XXX??? YYY???

Table 2.2: Extra temporal relationships introduced by Kumar and Mukerjee that allow modelling of incomplete intervals. A fourth relation ϕ (null) expresses the relationship between two events whose proposition happens to be false at that instant. Question marks (?) in the example represent either the relevant symbol (X or Y) or blank.

Another approach that allows reasoning with incomplete knowledge or uncertainty is presented by Freksa (1992*a*). Although based around Allen's interval-based approach, Freksa splits an interval into 'beginnings' and 'endings' otherwise known as semi-intervals. New relationships to support semi-intervals are shown in table 2.3.

An important part of the theory is the idea of *conceptual neighbours*, *conceptual neighbourhoods* and *coarse knowledge* :

- two temporal relationships are conceptual neighbours if they can be directly transformed into one another by continuously deforming the intervals (in a topological sense).
- a conceptual neighbourhood is a set of temporal relationships where all the elements are path-connected through conceptual neighbour relations.
- when the associated disjunction of incomplete or uncertain knowledge about temporal relations forms a conceptual neighbourhood, it is classed as coarse knowledge.

An abstract composition table, based on conceptual neighbourhood relations rather than the base relations allows the simultaneous composition of several relations as well as

Relation	Symbol	Example

X *older than* Y
Y *younger than* X
	ol
	yo
		`XXX????`
		`    YY`

X *head to head with*	hh	`XXX???` `YYYY`

X *survives* Y
Y *survived-by* X
	sv
	sb
		`????XXX`
		` YY`

X *tail to tail with* Y	tt	`??XXXX`
		` YYYYY`

X *precedes* Y
Y *succeeds* X
	pr
	sd
		`XXX?`
		`    YYY`

X *contemporary of* Y	ct	` ?XXX???`
		`???YYY?`

X *born before death of* Y
Y *died after birth of* X
	bd
	db
		` XXX?????`
		`?????YYY`

Table 2.3: Freksa's semi-interval relationships. Question marks (?) in the example stand for either the relevant symbol (X or Y) or a blank.

a coarse reasoning strategy suitable for reasoning with incomplete or vague knowledge. This coarse reasoning strategy does not necessarily lead to coarser results being obtained, in fact the entries in the abstract table match those in the full table — they are just in different positions. Fine reasoning is also possible (although, computationally more expensive) by finding the conjunction of inferred results based on the boolean combination of neighbourhood relations that yields the desired base relation. Composition tables at various granularities can be generated but in general, more efficient processing is obtained when knowledge can be shifted to a coarser level.

A generalization of interval algebra to an n-interval algebra is demonstrated in Ligozat (1990) where the special case of n=2 coincides with Allen's interval algebra. This generalization is expressed in a relational algebra A_n where the atoms have a natural topological structure represented by polytype $H_{n,n}$. This generalized algebra can also represent (p,q)-relations (Ligozat 1991) (i.e. polytype $H_{p,q}$) where $H_{1,2}$ is set of point-interval relations (i.e. $<$, s, d, e and $>$).

Mukerjee & Schnorrenberg (1991) look at reasoning in multiple scales (i.e. various levels of detail/granularity). Depending on the scale, there is some threshold beyond which the distance between two objects disappears and those objects are perceived as being in contact. This threshold is known as tolerance space and is a scalar parameterization based on the observer, intent and the environment. Although combining information at very disparate scales will not yield meaningful information, over comparable tolerance spaces there is a possibility of reinforcing and exchanging information. Mukerjee and Schnorrenberg look at this combination of tolerance spaces for point-point relations, point-interval relations and interval-interval relations.

Within this thesis, we do not utilize any of these qualitative temporal logics directly. Rather we incorporate temporal knowledge directly into our spatial model in the form of equi-temporal regions (see chapter 4) where Mukerjee & Schnorrenberg's (1991) notion of tolerance space is used when originally forming the equi-temporal regions. Further temporal information is modelled indirectly when one frame progresses to the next.

2.2.2 Topological

Although essentially topological, the interval algebra introduced by Allen (1983) only considers the temporal (1-dimensional) domain. This section details the research into topological relationships in the spatial domain — for visual surveillance, we are interested in the interaction between two (or more) physical objects which have a multi-dimensional spatial extent (rather than just 1-dimensional).

An extension of the interval logic to multi-dimensional cases is explored by Mukerjee & Joe (1990). Relations along each of the axes in an orthogonal domain are represented in a multi-dimensional vector. However, each object typically has its own "natural" orthogonal system so that no one representation can model all of them. As such, the relative position of two moving objects is modelled based on the 'lines of travel' (based on the current trajectory) taken by the objects and their intersection. As the lines of travel are different for two objects the relation is non-commutative and does not have a well defined inverse i.e. given *pos(A/B)* it is not possible to determine *pos(B/A)*. However, considerable reasoning is possible when combined with the relative direction of the objects (discussed further in the next section). We use a similar approach in chapter 5 when classifying the relative position from one object to another.

The qualitative spatial calculi developed by Randell & Cohn (1989) is an adaptation of the calculus of individuals developed by Clarke (1981, 1985). Clarke's original theory is based around a single primitive dyadic relation, $\mathtt{C}_{x,y}$ meaning 'x is connected to y'. A mereological definition of the base relations is given along with quasi-boolean and quasi-topological[2] function definitions.

Randell & Cohn's (1989) adaptation is an improvement in three ways:

- Clarke makes use of 2nd and 3rd order variables in his definition. Randell & Cohn maintain a 1st order formulation expressed in a many sorted logic know as LLAMA (Cohn 1987) allowing an easier reasoning mechanism.
- the partial (or quasi) functions are made explicit in the many sorted logic with the addition of a `null` object making the functions complete.
- The inclusion of a new primitive, `conv(x)`, meaning the convex hull of x[3], which allows further base relations and distinctions to be made.

A refinement of the primitive definition for `C(x,y)` from 'regions x and y share a common point' to the weaker 'topological closures of regions x and y share a common point' overcomes various conceptual, pragmatic and computational problems (Randell, Cui & Cohn 1992). In total, the theory has eight jointly exhaustive and pairwise disjoint basic relations obtainable just from the `C(x,y)` primitive (see figure 2.1). The number increases to 23[4] when considering the convex-hull primitive. In fact, it is possible to increase the number of qualitative relationships extensively by considering small refinements to the logic (e.g. Cohn, Randell & Cui (1995) demonstrates over a hundred jointly

[2] The term *quasi* is used due to the unavailability of a `NULL` object.

[3] The `conv(x)` primitive can be thought of as a 'cling-film' operator that gives the convex hull of a (concave) object.

[4] Originally there where thought to be only 22 relationships until an additional relationship was shown to be possible.

exhaustive and pairwise disjoint relations).

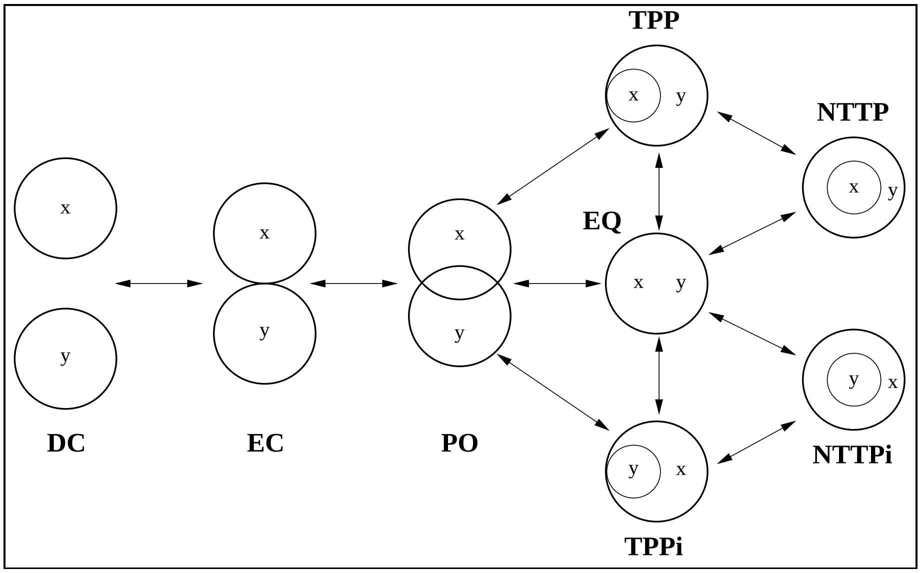


Figure 2.1: A pictorial representation of the eight base relations and their direct topological transitions (i.e. continuity network).

Continuity networks are used to represent the legal transitions from one relation to another (figure 2.1) and composition tables provide table look-up for the combination of two different spatial relations. These continuity networks are not dissimilar to Freksa's (1992*a*) definition of conceptual neighbourhoods although the differences are discussed in (Cohn, Gotts, Randell, Cui, Bennett & Gooday 1995) with regard to the generation of compact or abstract composition tables. A reformulation of the spatial calculi to an intuitionistic propositional logic representation is demonstrated in Bennett (1994) which can then be used in the automatic generation of composition tables.

The expressive power of the formalism is demonstrated using the continuity network and composition tables as the basis for the qualitative simulation of a force pump (Randell, Cohn & Cui 1992) as well as a biological example in an envisionment-based simulation of phagocytosis and exocytosis — the process used by unicellular organisms for garnering food and expelling waste material (Cui, Cohn & Randell 1992). We find these methods useful when verifying that each change in an object's history is legal (in chapter 5 section 5.4).

Another formalism based on Clarke (1981, 1985) is given by Vieu (1993) in the application of geographic space. This reformulation maintains a 1st-order logic and provides a redefinition of points to overcome Clarke's flawed definition (Vieu 1991). Distance and orientation are also introduced as inequalities.

The formalism described by Egenhofer & Franzosa (1991) is based around the practical needs for geographical information systems (GIS). Topological relations are described by the four intersections of the boundaries and interiors of two point-sets. This supplies a total of 16 mutually exclusive relations of which 8 are illegal when restricting the allowable relations to those which are homeomorphic to polygonal areas in a plane. Perhaps surprising considering the staring points, the remaining 8 relations are essentially identical to the base relations in Randell, Cui & Cohn (1992)[5]. An alternative formalism based on a 9-intersection model (boundary, interior and exterior intersections) provides a richer set of relations (Egenhofer & Herring 1991). This 9-intersection model is an improvement of the 4-intersection model as it considers relationships with the embedding space as well as the relations between feature parts.

Egenhofer & Al-Taha (1992) provide what they call the 'closest topological distance' graph. This is based on the topological distance between two pairs of regions represented by the 9-intersection model where the topological distance is the sum of all the differences in the intersection model. The most likely change for a topological relationship is that with the smallest non-zero topological distance value (a zero distance indicates no topological significance). The resulting graph is almost identical to Randell, Cui & Cohn's (1992) continuity network (figure 2.1).

Most topological approaches embody area/area relations and disregard other dimensions. In the context of GIS, Clementini, Di Felice & van Oosterom (1993) explore a dimension extended approach to Egenhofer & Franzosa's (1991) geometric point-set approach (i.e. examine relations between areas, lines and points). A total of 52 valid relations are shown to be possible — too many to be practical for a GIS query language. Instead a reduced set of mutually exclusive calculus-based relations is introduced (touch,

[5] Dornheim (1995) discusses the differences.

in, cross, overlap and disjoint). Clementini et al. (1993) demonstrate that *all* the relations in the dimension extended approach can be modelled using disjunctions of this reduced set of relations.

A way in which relations between regions with holes can be modelled is demonstrated in Egenhofer, Clementini & Di Felice (1994). If the holes are considered as separate regions then the problem of modelling relations between regions with embedded holes becomes one of expressing relations between simple regions. Many of the relations can be automatically inferred and Egenhofer et al. (1994) supply an algorithm to produce a minimized set of relations which does not include inferable relations.

When considering spatial regions in everyday contexts it is often found that they do not have precise boundaries: for example urban areas or the natural habitat of some creature. Such spatial regions tend to fall within two broad categories:

- Objects with sharp boundaries where the position and shape are unknown or cannot be measured exactly. This situation is known as "positional uncertainty".
- Situations where there is no well defined boundary for an object.

Both Clementini & Di Felice (1996) and Cohn & Gotts (1996) have examined this problem independently and proposed very similar models. Clementini & Di Felice extend Egenhofer & Herring's (1991) 9-intersection model describing the indeterminacy of an object's boundary as a two-dimensional zone surrounding the object separating the space that surely belongs to the object and the space that is surely outside. The model geometrically defines a region with a broad boundary by considering two "simple" regions with sharp boundaries representing the region enclosed by the inner boundary and the region enclosed by the outer boundary. The broad boundary is also a region, although with a hole, comprising the area between the inner boundary and the outer boundary. Clementini & Di Felice demonstrates 44 possible relations from which they construct a conceptual neighbourhood. The conceptual neighbourhood can then be clustered into similar relations which are a superset of those relations for simple regions.

Similarly, Cohn & Gotts (1996), extend the framework of 'RCC theory' (Randell, Cui & Cohn 1992) to cope with regions with indeterminate boundaries otherwise know as "vague" or "non-crisp" regions. As with Clementini & Di Felice (1996) the model defines a vague region as two subregions using an "egg-yolk" representation. The inner subregion is the "yolk" while the outer subregion is the "white". Together, both subregions are the "egg" — thus the "egg-yolk" representation. Randell, Cui & Cohn also define a "crisping" relation, `CR(X,Y)`, which refines the vagueness of region `Y` to (a less vague) region `X`. A complete "crisping" translates a region with an indeterminate boundary into a region with a sharp boundary. The only acceptable regions obtained from a complete "crisping" must lie between the inner and outer limits defined by the "egg-yolk". By considering all possible (logical) configurations 46[6] possible relations are obtained (when using RCC-5). These are clustered according to the possible relations obtained by a complete "crisping" of the two vague regions. The obtained clusters are similar but different to those shown by Clementini & Di Felice (1996).

Although the spatial model we currently generate contains regions with well defined boundaries, this is not necessary, and we suggest that as further work, the spatial model could be extended to consider spatial regions with indeterminate boundaries.

Throughout this section we have discussed a number of qualitative topological formalisms. Although there are differences in the way these formalisms were constructed the relationships identified are very similar. The spatial model we generate is essentially topological and can be described using any of these formalisms. However, in itself, topology is not sufficient for effective spatial reasoning in visual surveillance — there is no concept of direction or orientation which is required to sufficiently describe the relationship between two moving objects. In the next section we examine some of the qualitative formalisms that deal specifically with orientation and direction and often expand on the purely topological approaches..

[6] There are two more than Clementini & Di Felice (1996) which is unable to recognize the difference between two sets of particular configurations.

2.2.3 Orientation/Direction

As discussed in the previous section, the concept of direction or orientation is essential when describing the relationship of two objects in 2D or 3D space. When describing directions in space, concepts such as "right" and "left", "up" and "down" as well as "in front" and behind" are often used. These are all qualitative concepts that form the basis for qualitative vectors (Nielsen 1988) which have been successfully applied in a number of areas including the qualitative simulation of a clock mechanism (Forbus, Nielsen & Faltings 1991).

The points in a qualitative vector are described by the symbols {+,0,-} with respect to their orientation on a Cartesian coordinate system. In the 2-dimensional case this represents any of the four quadrants, an axis or the centre. Vector arithmetic is shown to be possible with only the addition and multiplication of signs necessary, although ambiguities will arise when adding opposing signs unless more information is know (c.f. table 2.4).

	+	0	-
+	+	+	?
0	+	0	-
-	?	-	-

Table 2.4: Addition of signs in qualitative quantity space {+,0,-}. Entries marked with a '?' represent ambiguities.

Extensions to qualitative vectors have been made by Weinberg, Uckun, Biswas & Manganaris (1992) and Kim (1992). Weinberg et al. (1992) look at the qualitative analysis of dynamics and extend qualitative vectors to an algebra allowing greater vector manipulation. Inequalities are used to describe qualitative magnitudes while angles are given by {*aligned, acute, perpendicular, obtuse and opposite*}. Improved vector addition is obtained by the comparison of the magnitude and angle of two vectors. A number of lemmas are provided to formalize the reasoning mechanism.

Qualitative kinematics of linkages is the focus of Kim's (1992) extension. In this theory, direction is represented by *sense* and *inclination* where sense is a qualitative vector. However, sense is not always sufficient to distinguish different kinematic linkage states and inclination must be used. Inclination is the level of incline from the x-axis and is represented by inequalities between different link angles.

Another similar approach is considered by Mukerjee & Joe (1990) although no connection is actually made. An intrinsic frame of reference based on the "front" of an object is used to determine the relative direction to another object and the quadrant in which that object lies. In 2D-space this gives eight qualitative angular relations with 26 in 3D-space. When combined with the relative position, discussed in the previous section, a collision parallelogram can be constructed which defines the area common to the 'lines of travel' of two objects. This allows the relationship between the two objects to be identified.

The intrinsic frame of reference (FofR) used by Mukerjee & Joe (1990) is one of three possibilities; *intrinsic*, *extrinsic*[7] and *deictic*. An intrinsic frame of reference exploits some inherent property of the reference object (e.g. 'front'), while an extrinsic frame of reference imposes an external immutable orientation (e.g. gravity). Orientation from a deictic frame of reference is with respect to some point of view (e.g. an observer).

Combining topological information with orientation is the focus in Hernández (1991, 1994). Spatial projection (3D to 2D) obtains typical topological relationships, disjoint, tangency, overlap and inclusion[8], which are incorporated with orientation relations based on 45 degree zones — front, left-front, left, left-back, back, right-back, right, right-front. Spatial knowledge is expressed as projection/orientation pairs with respect to some frame of reference (e.g. <A, [disjoint,back], B, {intrinsic}>).

Abstract maps, which exploit the structure of space, are available to model changes in the point of view as well as the more typical composition table allowing the simultaneous composition of relations or coarse reasoning. Constraint propagation algorithms (adapted from the temporal domain) allows the addition of new relations and their effect on an entire network to be generated. Also presented is an approach to deleting relations

[7] Nielsen's approach is extrinsic.

[8] Note: inclusion covers equality, inclusion at the border and the inverse relations.

and taking back the consequences of propagation by using a dependency network along with a reason maintenance system (Hernández 1993*a*, Hernández 1993*b*).

A qualitative model which defines directional orientation information as available through perceptual processes is described in (Freksa 1992*b*, Freksa & Zimmermann 1992). When considering the direction from a vector *ab* and its inverse vector *ba* to a point *c* it is possible to define fifteen possible locations (as shown in figure 2.2). Reasoning is possible through composition tables and a number of operations:

- Inversion: if `c:ab` is know, it is possible to precisely deduce `c:ba`.
- Homing: given `c:ab` then find `a:bc` (obtains imprecise result in the form of a conceptual neighbourhood).
- Shortcut: given `c:ab` find `b:ac` (obtains similar results to homing operation — entries in the table are just in a different order).

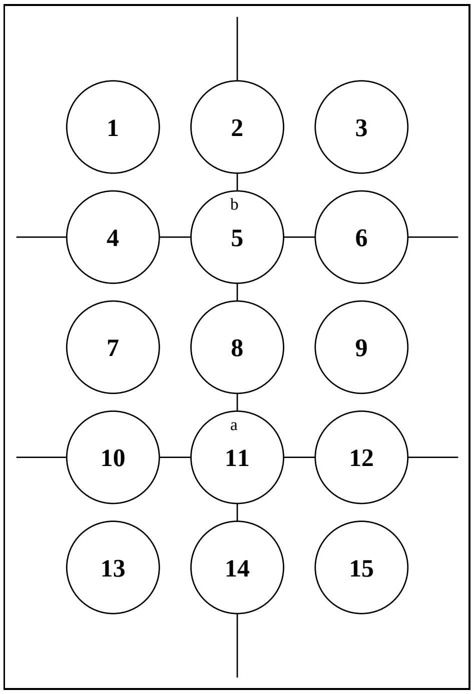


Figure 2.2: The fifteen different positions which can be determined using Freksa's model.

It is suggested that through algebraic combination of these operations along with composition it is possible to build and deduce a relation for every possible combination of points with respect to any vector.

An extension to this work in (Zimmermann & Freksa 1993) shows that improved inference results can be obtained if path knowledge is employed. Path knowledge is composed from the set of relations that define the path (assuming it is straight) between `b` and `c` when given the relationship `ab:c` (example). Individual composition of the path relations can refine the overall result improving inference results.

A generalization of Freksa's (1992*b*) qualitative model is demonstrated by Ligozat (1993) in terms of qualitative triangulation. Triangulation is the process of locating a third point by computing the angles and distance of the lines between two other points and the third point. With qualitative triangulation, qualitative knowledge of the angle values is used and propagated as new values are considered. Freksa's (1992*b*) qualitative orientation model is obtained when the scale of angles is restricted to 90 degree increments.

The last approach in this section considers the orientation of points in a plane. The orientation or ordering of points on a line (1D-space) is well known (<, = and > or [-,0,+]), what is less well known is that this is also possible for points on a plane (2D-space):

$$\texttt{(a,b,c)} = \begin{cases} +\text{ve} & \text{if counter clockwise order.} \\ -\text{ve} & \text{if clockwise order.} \\ 0 & \text{if collinear.} \end{cases}$$

This observation is utilized for qualitative navigation (Schlieder 1993).

Schlieder (1995) demonstrates that 1D-ordering information can exactly describe Allen's interval relations and extends this ordering idea into 2D-space using two pairs of connected points. As long as the points are not collinear then there are 14 relations (as shown in figure 2.3[9]). If collinear points are allowed, there are the 13-Allen relations

[9] Adapted from Schlieder (1995).

along with a further 36 relations when three of the points are collinear.

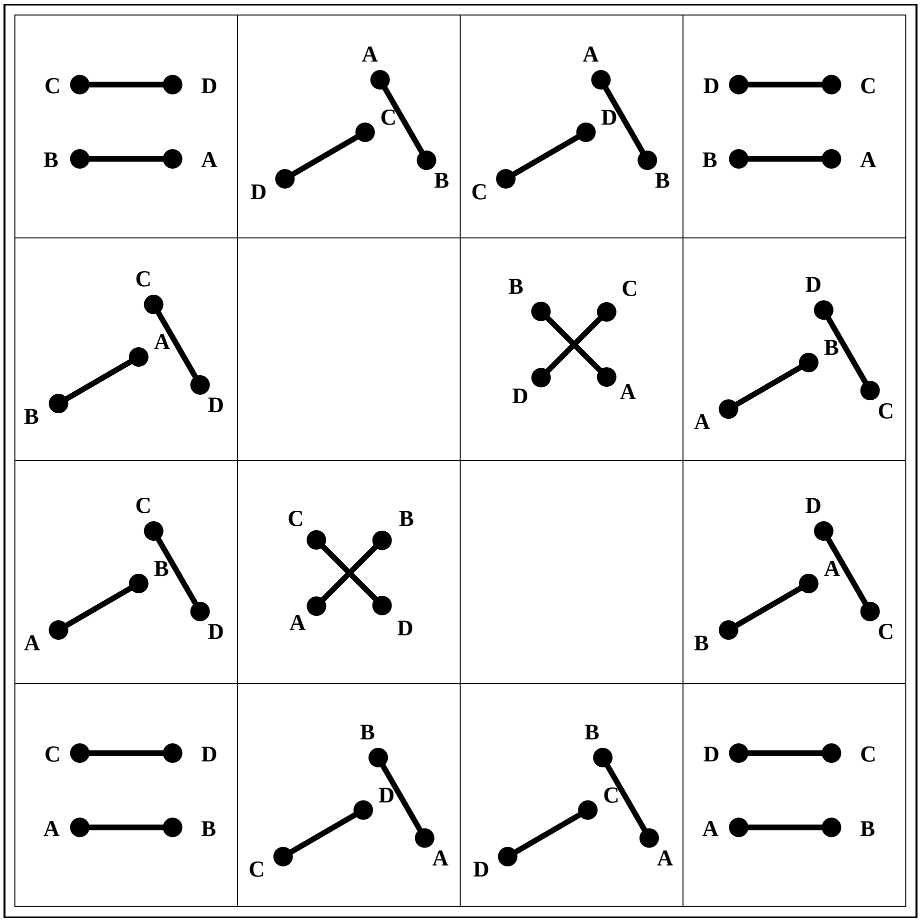


Figure 2.3: The 14 line segment relations determined by Schlieder.

Within this section we have examined a number of different approaches to modelling orientation and direction qualitatively. From these different approaches, the one which most closely matches our requirements for visual surveillance and the interaction of moving objects is that described by Mukerjee & Joe (1990). In this approach, an intrinsic frame of reference is used based on the "front" of a vehicle. In chapter 5 our approach to obtain the relative position of one object with respect to another is similar, although the "lines of travel" we use is based on the composite regions we generate as part of the spatial model in chapter 3.

Again in chapter 5 we classify the direction of motion using a deictic point of view based on the camera position. This is used to convert the quantitative vector supplied by the tracking algorithm into a qualitative (45 degree) zone as described by Hernández (1991).

The observation made by Schlieder (1993) about point ordering is also found useful when determining which equi-temporal region an object is contained in (as described in chapter 5.

The other approaches described within this thesis are not specifically utilized within this thesis. However, certain insights into relative position and direction have assisted and inspired the work described throughout the course of this thesis.

2.2.4 Size and Distance

Size and distance are related in so far as we tend to use linear scale systems to measure each of these aspects. Distance is typically thought of as a one dimensional concept whereas size is multi-dimensional (area or volume). The domain may also influence distance values (isotropic and anisotropic surfaces) but qualitative reasoning systems are typically concerned with linear quantity systems such that the qualitative algebras developed will apply equally to both size and distance representations.

Probably the earliest measuring scheme introduced to qualitative reasoning are the order of magnitude calculi (Raiman 1986, Mavrovouniotis & Stephanopoulos 1988) which allow a quantity to be described as being *much larger or smaller* than another. This means that many smaller values are required to surpass a “much larger” value.

A more recent representation known as the Δ-Calculus (delta-calculus) is described in Zimmermann (1995). This formalism considers the cognitive capabilities of humans reasoning about point-like measures (e.g. durations or object dimensions):

- Human observation tends to regard positive measures, so negative values are unsupported in the calculus.
- Direct multiplication of two measures is considered cognitively implausible (consider multiplying the size of a chair by the width of a table,) and as such is only supported

through the repeated summing of a measure.

The Δ-calculus introduces a triadic relation for difference measurements, $x(>, d)y$ where x is larger than y by some d. Measures (e.g. x,y and d) are maintained as relational knowledge with adaptive granularity, for example $x(>, d)d$ would make x twice as large as d meaning that y is three times as large as d and one and a half times as large as x.

Zimmermann (1993) combines the Δ-calculus with Freksa's (1992*b*) orientation model — considering the distance between the points in vector ab to a third point c it is possible to obtain (limited) distance information. Finer levels of distinction can be made if the perpendicular and vertical distance from point c are also considered.

A technique to determine the relative size of two objects was proposed by Mukerjee & Joe (1990). When the starting point of two objects is the same, the observed relationship between those two objects is determined by their relative size. For example, if the relationship is equality the objects must be the same size, whereas if the one object *starts* the other it must be smaller. From this observation, Mukerjee & Joe define a flush-translation operator, ϕ, which is used to translate two size and shape invariant regions, A and B. By observing the topological relationships between the flush regions ϕA and B, it is then possible to determine the relative size $\{<, =, >\}$.

A relatively new and sophisticated formalism for modelling distance qualitatively is proposed by Hernández, Clementini & Di Felice (1995). The distance relationship between two objects is expressed with respect to some frame of reference (analogous to that used in orientation systems); *intrinsic* distances are determined by inherent characteristics of the object (e.g. topology, size or shape), *extrinsic* distance is based on some external factor (e.g. object arrangement, travel time) and deictic distance relies on an external point of view (e.g. camera position).

Different levels of distinctions can be made for *distance ranges*, for example close and far, close, medium and far or very close, close, commensurate, far and very far. A distance system allows the choice of a distance range and requires a set of *structural relations* which provide additional information about how the distance ranges correspond to one another (e.g. monotonically increasing ranges or order of magnitude). Composition of relations is based on the structural relations and examples are provided for distances with the same orientation and different orientations. This method is currently being extended to include other orientations.

Within this thesis, we do not directly use qualitative size although it may be used to extend the scope of the work described here to consider the behaviour of two (or more) objects of dissimilar size. Due to time constraints, this idea was not pursued to any extent.

However, we do consider distance but as yet, we only look at "close" objects. Our use of "close" is not based particularly on spatial proximity and is discussed in more detail in chapter 5.

2.2.5 Shape[10]

A perceptual approach to the organization and representation of natural occurring forms examines the inherent regularities in organic and inorganic bodies (Pentland 1986). The complexity of shape description arises from the limited vocabulary in combing the finite number of basic forms in a myriad possible combinations. Pentland (1986) proposes a method allowing the representation of objects using the boolean combination of a few basic forms[11]. These basic forms are represented by a parameterized family of shapes known as *superquadrics*. Although a superquadric provides a quantitative description of that part, a qualitative description can be used for the boolean combination of different parts.

A further refinement exploits the general characteristics of natural occurring fractal forms (such as clouds or a mountain) where the ratio of a feature in one scale to the same feature in the next larger scale is constant. Such fractal surfaces can be constructed using superquadric parts at recursively smaller scales.

[10] Throughout the course of the research presented in this thesis we have not been concerned with qualitative shape description. This section is included only as a matter of completeness although future work may want to consider the shape of paths/regions as they may evoke different behaviour patterns.

[11] This method can be seen as equivalent to a 'naive' verbal description given by people.

Another hybrid qualitative and quantitative shape model uses an axial model (Mukerjee 1994) for shape descriptions. Qualitative shape models are ambiguous (by qualitative definition) and represent a class of conceptual objects — Agrawal, Mukerjee & Deb (1995) propose a method using a real-coded genetic algorithm to implement the visualization and optimization of such inexact shapes.

Jungert (1993) presents a formalism for the qualitative matching of object shapes. An object shape description is represented as characteristic points including the angle (*obtuse*, *acute* and *right-angled*), entry and exit directions. Sequences of points allow concave and convex areas to be identified. For shape matching, the relative angles are used and a simplified sequence obtained.

An exploration of the basic connection primitive (i.e. `C(x,y)`), used in the topological spatial calculi (RCC) developed by Randell, Cui & Cohn (1992), is conducted by Gotts (1994) to determine what level of topological complexity can be identified — in particular, to decide if the topology of a region is that of a solid torus (or a 'doughnut').

Cohn (1995) proposes a shape extension to the RCC theory which further exploits the convex-hull operator. This proposal allows a wider selection of shapes to be distinguished than just connection (Gotts 1994). The maximal connected (i.e. one piece) parts of the inside of a region and the relationships between them can be identified. Further distinctions can also be made to identify adjacent holes (concave areas) and holes on the 'same side'. Finer grained shape distinctions can be obtained by recursively applying the technique to each maximal inside of the original shape. More recent work by Davis, Cohn & Gotts (to appear) shows that it is possible to distinguish any shapes which are not affine related.

Qualitative shape representation based on ordering information is proposed by Schlieder (1996). A sequence of triangle orientations for the vertex points are used in the representation. The more triangle orientations included, the more refined the shape becomes. A complete set of qualitative relations for quadrilateral shapes is demonstrated along with a formalism to obtain the relations and its conceptual neighbourhood. The conceptual neighbourhood relations are based on the Hamming distance (i.e. the number of different components) between two relations (similar to the closest topological distance used by Egenhofer & Al-Taha (1992)). This formalism could easily be adapted to more complex shapes (i.e. those with more vertex points). However, using boundaries to describe shape can cause problems for practical reasoning — for example, when two objects are tocuhing which boundary points belong to which object? Fleck (1996) discusses these problems in more detail and suggests an alternative approach where boundary points are deleted from the representation of space allowing the attention to focus on region borders; thin strips of a region adjacent to the boundaries.

Within the spatial model we construct, we follow Fleck's (1996) approach when considering region occupancy in chapter 5.

2.2.6 Summary

Throughout this section we have provided a review of the rich set of qualitative representation and reasoning systems currently available for reasoning about space and time. Although it is not always possible to obtain accurate quantitative knowledge about a particular situation, it is typically possible to collapse the (potentially) inaccurate quantitative knowledge into a broader (qualitative) subset which contains the critical aspect of knowledge necessary to allow a qualitative reasoning system to predict, diagnose or explain the physical behaviour(s) being observed.

Our intended purpose is to utilize appropriate techniques for the purpose of visual surveillance. In particular, we want to be able to recognize particular behaviours or *events* observed in the domain. Rather than providing the descriptions of these events as *a priori* system knowledge part of our research has been to learn such models automatically through the extended observation of a particular scene. Such conceptual models can be sufficiently (and probably better) described using a qualitative representation.

Throughout this section we have indicated the approaches which most closely resemble our requirements (in particular in the section on Qualitative Orientation and Direction — section 2.2.3).

2.3 Conceptual Descriptions from Image Sequences

In providing conceptual descriptions of observed behaviours in real-world image sequences, it is necessary to perceive and understand the actions and interactions of objects moving in the scene. Computer vision, a large and diverse field of Artificial Intelligence, provides the basis for the artificial perception of situated actions. Essentially vision (both biological and machine) can be split into three stages; (1) Low-level (or early), (2) Intermediate-level and (3) High-level vision.

- Low-level vision is the most understood. Visual receptors provide a 2D array of intensity values (i.e. an image) representing the real-world view. Low-level processing is achieved using visual primitives to obtain image features such as edges. A large amount of image processing literature already exists and we will spend no more time covering these concepts. For more details, see any of the following books; Castleman (1979), Hall (1979), Gonzalez & Wintz (1987), Boyle & Thomas (1988), Schalkoff (1989) or Sonka, Hlavac & Boyle (1993).

- Intermediate-level vision typically concerns the recognition of objects. For single images this is usually object identification through model matching techniques whereas tracking individual objects is the focus for image sequences. For more information, see Ullman (1996).

- High-level vision is the least understood stage and, at present, contains the least amount of active research. Emphasis is placed on the conceptual understanding of information obtained from the intermediate-level visual processing such as the recognition or interpretation of situated actions or sequences of situated actions (events). For a more detailed review of high-level vision see Howarth (1995).

  By allowing the feedback of information based on the results from high-level visual processing to the intermediate and low-level visual stages, it is possible to control the processing that should be performed at those levels (Bajcsy 1988, Ballard 1991). Typically such systems will have a gaze control mechanism that can actively position the camera in response to physical stimuli allowing simpler execution of visual behaviours such as physical search and intelligent data acquisition.

2.3.1 Object Tracking

Although we are most concerned with high-level visual processing, we make use of resulting information obtained from an object tracking application. As such, we will provide a brief overview of the current state of visual tracking technology.

It has become recognized that to track objects effectively in a cluttered scene some sort of *a priori* information is necessary in order to find the object being tracked (although exceptions exist). Prior information usually takes the form of object shape models which may be derived statistically from training data using "Principal Component Analysis" (PCA) as described in Jolliffe (1986).

The type of shape model typically depends on the object to be tracked. Sullivan (1994) describes a model-based vehicle tracking system which was originally developed for the recognition and pose recovery of a vehicle in a single frame. Knowledge of the camera position with respect to the ground plane reduces the search space (for subsequent object positions) by constraining the possible degrees of freedom from six to three (full 3D movement and orientation to 2D movement and orientation in a single plane). The tracking procedure can be seen as an application of Lowe's (1991) refinement technique — an iterative procedure which begins with an initial rough estimate of the position and orientation of the object and at each iteration of the refinement, suggested movements are calculated from image features.

Sullivan (1994) relies on CAD-like geometrical models of objects to be recognized and the scene in which they appear. A "pose hypothesis" is generated through a process of Canny edge detection on the image, which is then reduced to a set of straight line segments. Strong lines of a significant length are compared against each vehicle model to find those lines which are consistent. After generating an orientation histogram and determining the possible model origin an "iconic evaluation" is performed to measure the quality of the object and pose hypothesis.

In a more recent paper (Ferryman, Worrall, Sullivan & Baker 1995), the geometrical object shape models have been generalized to a generic deformable model — composed initially from 29 parameters. Unfortunately, considering the three spatial degrees of freedom, these 29 parameters lead to a configuration space which, for all practical purposes, is too large to search naively when attempting to locate an object. However, to represent a vehicle strong structural constraints can be applied and obtained through principal component analysis. The 6 main PCA parameter prove sufficient to distinguish the three sub-classes of car (hatchback, saloon and estate) which is a searchable configuration space.

When tracking the motion of a non-rigid object, such as a walking person or a hand, an alternative shape model is more appropriate. The Point Distribution Model (PDM) introduced by Cootes, Taylor, Cooper & Graham (1992) and Cootes & Taylor (1992) is one such example. Typically, a PDM is a statistical model of a set of (2D or 3D) "landmark" points where each point corresponds to a particular feature on the object. The landmark points for the PDM are based on a statistical analysis of the point coordinates over a training set. In a related approach, Baumberg & Hogg (1995) describe a method which tackles the problem of modelling continuous deformable contours using a spline shape representation which provides a more efficient method for calculating a statistical shape model for continuous curves rather than using a dense set of sampled boundary points.

For tracking objects in the scene, Cootes & Taylor (1992) describe their "Active Shape Model" for locally optimizing the shape parameters of the object model to fit the features in the image. The actual method is similar to that used by Sullivan (1994) and regarded as a 2D application of Lowe's (1991) refinement technique where at each iteration of the refinement process, suggested movements for each landmark point are calculated from image features.

Other approaches such as the "snake" (or active contour model) of Kass, Witkin & Terzopoulos (1987), "Kalman Snake" (Terzopoulos & Szeliski 1992) and "Active Splines" (Blake, Curwen & Zisserman 1993) are 2D, contour based approaches where object shape is constrained to be continuous and to deform smoothly. A "snake" is an energy-minimizing spline (like an elastic membrane) that is attracted to image features such as edges. The "Kalman Snake" employs a Kalman filter to provide a mechanism for tracking a "snake" over successive image frames which allows model parameters to be derived from a statistical sensor model and varied over time. An "Active Spline" evolved from the principles of a snake and provides a framework for efficiently tracking B-spline contours using a Kalman filter mechanism. Through the implicit continuity and elasticity of a B-spline, a simple stochastic model can be applied without having to explicitly "regularize" the energy-minimizing function.

To date, less sophisticated tracking applications have found a home in commercial surveillance systems. In such systems, a simple background subtraction image processing technique is applied to recover moving objects in a scene. Connected components of flagged pixels usually correspond to moving objects although when several objects in an image overlap, or are too close to be distinguished, a single region will be obtained which represents several scene objects. This technique is also highly susceptible to changing lighting conditions, for example a cloud passing in front of the sun, although with more gradual changes an adaptive background can be applied. Baumberg (1995) uses this technique as a first step in his model generation process. This adaptive background technique is the method used throughout this thesis (for more details see chapters 3 and 5). It may be possible to improve the results detailed within this thesis by using a more sophisticated object tracker — as described in this section.

2.3.2 Interpretation of Image Sequences

This section deals with high-level vision systems that are capable of recognizing and able to interpret dynamic processes and situations within the real-world. A large proportion of this work combines computer vision systems with a natural language interface providing a means of conveying the system's understanding.

Perhaps the earliest work in this area can be attributed to Badler's (1975) pioneering work which proposed a model for organizing the visual world into conceptual structures based on the description of visually perceived motion concepts such as 'bounce' or 'swing'. Such conceptual structures are built from a hierarchy of motion concepts which are closely related to those concepts used to describe object movements in natural language. Using these concepts it becomes possible to look beyond movement or changes between two consecutive images and to describe change over a number of consecutive images (i.e. sequence spanning). Consider the notion of 'swing'; between two adjacent frames, it is only possible to determine that an object is rotating in a particular direction. If this sequence is followed over a number of frames, the overall motion can be described using a single motion concept. At the time Badler's research was conducted, obtaining sufficiently descriptive information automatically from visual input was not feasible so "ideal encodings"[12] of each image in a sequence were used.

This work was further developed by Tsotsos, Mylopoulos, Covvey & Zucker (1980) and Tsotsos (1981) to generate descriptions of the shapes and motions exhibited by a left ventricular wall — in particular noting any abnormalities or unusual occurrences. Unlike the previous work by Badler (1975) this research looked at real X-ray cinecardioangiograms[13] at up to sixty frames a second. A hypothesis rating scheme is used within the recognition scheme to select the most appropriate motion description.

The approach described by Badler and extended by Tsotsos derives the verbalizations bottom-up (i.e. the motion conceptualization is generated from an image sequence with a simple translation of the concept into words). An alternative top-down approach, outlined by Marburger, Neumann & Novak (1981) and known as NAOS, processes verbalizations in order to determine whether or not they correctly describe an image sequence. Using this approach, the system is capable of answering "yes" or "no" questions about moving objects in a real-world scene. An independent scene analysis system provides referential knowledge in the form of symbolic frame descriptions including object names, type and visual properties. Each object located in a frame is identified and labelled. When the same object appears in subsequent frames it is identified and labelled accordingly.

[12] These "ideal encodings" take the form of shape descriptions for the background and scene objects.

[13] The application looked at left ventricles that had received corrective surgery and during surgery nine tiny markers were implanted which allowed relatively simple cineradiography.

More formally, such a symbolic representation has become known as a Geometric Scene Description (GSD) (Neumann 1989). A GSD is an ideal representation of output from an intermediate-level vision process and should represent the original image sequence completely without loss of information — in principle, the data provided in a GSD is sufficient to reconstruct the raw images:

- the data for each frame includes
  - a time stamp
  - a list of visible objects
  - the camera viewpoint
  - camera illumination data
- the data for each object includes
  - an identity stamp
  - 3D-position and orientation in world coordinates
  - 3D-shape and surface characteristics (e.g. colour)
  - class membership and possible identity with respect to *a priori* knowledge (provides for example object name).

Such a representation is extremely idealistic and we are still far from a universally applicable AI system capable of completely analyzing any arbitrary sequence of images and providing a complete GSD. Instead, the components of the GSD are appropriately tailored to suit each system.

Generic event models (Neumann & Novak 1983) assist in the recognition of interesting temporal developments (i.e. events) in the observed scene. Event models, useful for both top-down (question answering) or bottom-up (scene description) approaches characterize a spatio-temporal representation for that event. The representation for each event model contains a declarative description of classes of actions organized around verbs of locomotion (for example see figure 2.4[14]) where the components are directly related to

[14] Example adapted from Neumann & Novak (1983).

the deep-case structure of a corresponding natural language description. These event models may be viewed as a template which must be matched against pertinent scene data (found in the GSD) in order to recognize instances of that event which can then be expressed in natural language.

```
(EVENT-MODEL OVERTAKE
  (PARAMETERS OBJECT1 OBJECT2 TIME1 TIME2)
    ((MOVE OBJECT1)@(TIME1 TIME2)
     (MOVE OBJECT2)@(TIME1 TIME2)
     (BEHIND OBJECT1 OBJECT2)@TIME1
     (BEHIND OBJECT2 OBJECT1)@TIME2
     (WITHIN (TIME3 TIME4)(TIME1 TIME2))
     (BESIDE OBJECT1 OBJECT2)@(TIME3 TIME4)
     (APPROACH OBJECT1 OBJECT2)@(TIME1 TIME3)
     (RECEDE OBJECT1 OBJECT2)@(TIME4 TIME2)))
```

Figure 2.4: Simplified event model for an "overtake" situation.

Another integrated vision and natural language processing system is LandScan (LANguage Driven SCene ANalysis) described by Bajcsy, Joshi, Krotkov & Zwarico (1985). This preparatory investigation outlines a system capable of dynamically updating and maintaining a model of an urban world over a number of aerial image views[15]. Processing is both data-driven (bottom-up) or query-driven (top-down):

- For data-driven processing, stereo aerial images are used to reconstruct polyhedral surfaces in a scene. Surface attributes and relations are computed using a geometric modelling system capable of determining a number of attribute values — including compactness, centroid, normal, area and type (e.g. building, sidewalk, or street) and topological relations (such as above, adjacent, contiguous and contains).

- A natural language front end allows query-driven processing to construct a logical representation of the scene and assists vision processing by restricting the scene analysis, through user interaction, to areas of current interest. The reasoning system analyzes the query, determines a strategy for obtaining an answer and provides

[15] The system outlined only considers single or stereo images, *not* image sequences.

feedback to the vision system. Should the query fail and no answer be found, the system will indicate whether the query was conceptually ill-formed, or whether insufficient information was available to answer the query.

Similar to Naos, the CityTour system described by André, Bosch, Herzog & Rist (1986) is also a (German) question-answering system. The system simulates a fictitious sight-seeing tour through the discourse world; an "interesting" part of a particular city containing both static and dynamic objects with the "sight seeing bus" being a special dynamic object. Static objects are represented as a closed polygon complete with a centroid and a prominent front edge along with a delineative rectangle oriented on the prominent front edge. Dynamic object movement is represented as a trajectory containing time stamps for each position. By examining the object trajectory along with a static object it is possible to define algorithms to recognize dynamic relations ('pass' and 'along' are the examples given in the paper). Unlike Naos, in CityTour the conversational partner is considered part of the scene (i.e. on the bus) and as such, the answer may take into account the position and orientation of the bus (i.e. allows a deictic point of view as well as an extrinsic viewpoint).

So far, all these earlier systems concentrate on an *a posteriori* analysis of dynamic scene data. The entire image sequence is considered before relevant events can be recognized. This means that the systems are only capable of providing a retrospective description of the analyzed scene. The system developed in the Vitra (VIsual TRAnslator) project is capable of recognizing events simultaneously as they occur in the image sequence using an incremental recognition strategy.

Initially, the domain of discourse considered in the Vitra project was a game of football (André, Herzog & Rist 1988) (or more specifically, short sequences of images obtained from a static camera watching a football match). The incremental recognition of events within the football game enables the system to provide a running commentary of the actions within the domain including perceived intentions (Retz-Schmidt 1988). The listener is assumed to have prototypical knowledge of the static background (in this case the football pitch). This world model can be seen as the stationary part of a Geometric

Scene Description and 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.

Events are described conceptually using events models, as with Neumann & Novak (1983) such event models represent *a priori* knowledge about typical occurrences in the domain and in particular the changes that people usually talk about. The core of an event model is described using a course diagram which is represented using a labelled directed graph, for example see figure 2.5[16]. Such course diagrams specify the sub-concepts and the situational context which characterize the instances of a particular event model. An incremental event recognition mechanism successively receives geometric data for the objects moving in the scene and attempts to match that information by traversing a course diagram. Propositional information, concerning events occurring at the moment, is generated and used to initiate the utterance for that event.

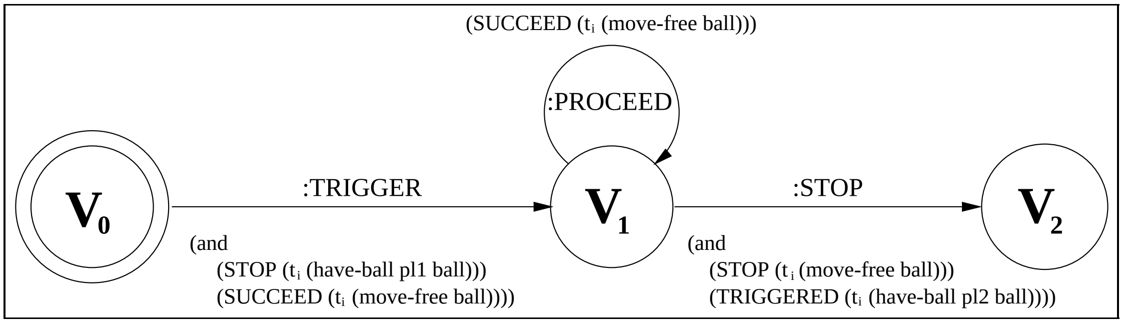

(SUCCEED (t$_i$ (move-free ball)))
:PROCEED
V$_0$
:TRIGGER
V$_1$
:STOP
V$_2$
(and
(STOP (t$_i$ (have-ball pl1 ball)))
(SUCCEED (t$_i$ (move-free ball))))
(and
(STOP (t$_i$ (move-free ball)))
(TRIGGERED (t$_i$ (have-ball pl2 ball))))

Figure 2.5: An example of a course diagram representing a "ball-transfer" event.

Spatial relationship's between various objects are represented by relational tuples (André, Herzog & Rist 1989) of the form:

```
(rel-name, subject, ref-obj₁, ref-obj₂, ..., ref-objₙ, <orientation>)
```

where `rel-name` is the spatial relationship between the object to be located, `subject`, (according to the `orientation`) with relation to one or more reference objects, `ref-obj`$_{1...n}$.

[16] Adapted from Herzog, Sung, André, Enkelmann, Nagel, Rist, Wahlster & Zimmermann (1989).

The spatial relationship is applicable if it can be used to characterize an object configuration. To determine the extent to which a spatial relationship is applicable an area of applicability is designated for each relation complete with a measure of the degree of applicability. More formally, Gapp (1994) provides a computational model of functions which define the degree of applicability for a number of basic spatial relations with respect to geometrical object properties.

For a more complete overview of the entire VITRA project see Herzog & Wazinski (1994).

Unlike the VITRA project where the primary concern has been to produce a natural language dialog (or running commentary) of situations occurring in a scene, the VIEWS (Visual Inspection and Evaluation of Wide-area Scenes) project (Corrall & Hill 1992) concentrates on Visual Surveillance in order to identify incorrect or illegal behaviour (i.e. *incidents*). This does not imply that a natural language engine could not be connected to provide a commentary, just that this has not been the aim of the project.

Similar to previous approaches, VIEWS is heavily knowledge based and relies on: a known representation of the scene; a (complete) set of object models to be identified; the camera configuration and a database of specific events and behaviours to be recognized. The location of individually labelled objects is provided frame by frame and can be thought of as part of a Geometric Scene Description. In this instance a 3D model based tracking method is used (Worrall, Marslin, Sullivan & Baker 1991) which makes use of the list of object models. An analogical representation of space (Howarth & Buxton 1992*a*) provides the static background for the GSD and allows situated actions and events to be identified.

Events and behaviours are scripted and formally represented as grammars. It follows that the recognition of events and behaviours is obtained by matching scene observations against these scripts (i.e. *parsing*). A behavioural parser based on island-parsing is used. Such a parser produces “islands” of recognized instances and needs to join these “islands” to infer which script is occurring. This parsing method allows intermediate states to be reported and is capable of tolerating diverse “noise” while still producing a correct interpretation — including errors such as insertions (unwanted events), deletions (missing events) and substitutions (transformed events).

Howarth & Buxton (1992*a*, 1992*b*) introduced their *analogical* representation of space as part of the VIEWS project. Their representation of space is a ground plane projection of the scene using a *hierarchical* structure based on *regions,* where a region is a spatial primitive defined as a (closed) two-dimensional area of space where the spatial extent of a region is controlled by the continuity of a particular spatial property.

For their purposes, the spatial representation is an extended form of the topological representation developed by Fleck (1988*a*, 1988*b*) — each region provides an encapsulation of space composed of cells which have topological properties (Munkres 1984). In particular, a regular cell complex is employed where the cells are made up of three cell-dimensions describing: vertices (0D); edges (1D) and faces (2D). The boundary cells which delimit a region provide a "skin" enclosing the contents. Although cells are not used directly, they do provide a topological foundation for the spatial structure that directly supports the topological reasoning required in their system. In itself, Fleck's *cellular topology* is purely qualitative, however, Howarth & Buxton also desire quantitative reasoning capability so they "fix" the topology by providing a coordinate system, through the addition of a Euclidean metric, on top of the basic cellular construct.

There are two kinds of region which they store in a spatial layout database:

- *Leaf regions* are the finest granularity of region and the most primitive database element. 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.

Howarth (1994) shows how such representations of space are produced manually for each new domain: A time consuming and painstaking process which provides the inspiration for our research into automatically generating such spatial structures. A knowledge acquisition program know as "MAP-EDITOR" assists the model generation process and produces a "map file" containing the geometric data in for the spatial model. Entries exist for points and lines, which provide polyhedra for leaf region descriptions. Leaf regions are used to define composite regions which can have associated attributes attached. The basic format used for the "map-file" is shown in table 2.5[17].

Map file format

`%P Pnnn float float float`	3D point defined by three floating point numbers
`%L Lnnn Pnnn Pnnn`	two points define a line
`%R Rnnn Lnnn ...Lnnn`	three or more lines define a leaf region
`%R Rnnn Rnnn ...Rnnn`	one or more regions define a composite region
`%A Rnnn attribute-index value`	assign value to given attribute of composite region `Rnnn`

Map file example

```
%P P169 14467.40768 -25836.72342 0.00000
%P P170 14629.23743 -26832.32174 0.00000
%L L140 P169 P170
%R R93 L140 L281 L342 L141
%R R222 R93 R76 R100 R27 R96 R92 R73 R63 R103
%A R222 Long-Name "Roundabout South Cycle-way"
%A R222 USED-BY CYCLE
```

Table 2.5: Basic format of Howarth and Buxton's spatial layout map-file.

Following Mohnhaupt & Neumann (1990), this decomposition is known as analogical because the representation explicitly matches the intrinsic structure of the scene. This means that the spatial model can directly be used as a support to any spatial reasoning involving objects within the scene. By using the analogical representation of space certain events can easily be determined. For example if a vehicle remains stationary in a region specified as a "give-way" zone then it can be postulated that the vehicle is giving way to another. Further, as the "give-way-to" zone is also labelled, the potential location of the other vehicle(s) is also known.

Howarth and Buxton further develop this representation to include a temporal aspect known as "conduits" which represent the space swept out through time by an

[17] Adapted from Howarth (1994).

object's path. This 2D+t structure is constructed by combining the consecutive locations occupied by an object throughout the scene with respect to time. Using this conduit it becomes possible to approximate more accurately the time in which a region was entered and exited as well as to allow reasoning about missing updates.

A number of requirements are highlighted to enable adequate reasoning about objects and interacting objects moving in the static scene. As well as converting the ground-plane coordinate data (in the map-file) into regions, the connectivity between those regions must be described. Essentially objects are treated in the same way as the static model — in each frame, the spatial extent of each object is obtained so that it can be positioned within a pose-box with labelled edges (front, left, right and rear). This pose-box allows the accurate identification of (partially) occupied leaf regions. To derive spatial behaviour, the speed and orientation of an object is required as well as inter-object orientation and distance when considering multiple objects.

Originally the project used a passive system which collected information about all objects in a scene. This approach has become known as HIVIS-MONITOR (Howarth 1994). The detection of single object events such as start, stop, turn left, turn right, speed up and slow down relies on the observation of changing object properties such as speed, orientation and region occupancy. For two or more objects it is necessary to determine if the objects are "near" any other by checking if they are in the same region (leaf or composite). In their terms, an event represents a state-change of some type and multiple events compose episodes. Typical episodes are: region-crossing; following; overtaking; give-way and waiting. Episodes are described using scripts. For example a valid turn right event[18] can be described as:

$$
\begin{array}{l}
\forall\, \mathrm{t}_1 \prec \mathrm{t}_2 \\
\mathrm{TRUE}(\mathrm{t}_1, \mathrm{t}_2, \mathrm{TURN\text{-}RIGHT}(\mathrm{x})) \;\Rightarrow \\
\qquad \exists\, \mathrm{t}_1 \prec \mathrm{t}_3 \prec \mathrm{t}_4 \prec \mathrm{t}_2 \\
\qquad \mathrm{TRUE}(\mathrm{t}_1, \mathrm{t}_2, \mathrm{IN\text{-}TURN\text{-}RIGHT\text{-}REGION}(\mathrm{x})) \wedge \\
\qquad \mathrm{TRUE}(\mathrm{t}_3, \mathrm{t}_4, (\mathrm{ORIENTATION\text{-}CHANGE}(\mathrm{x}, \theta) \;\wedge\; (\theta < -10)))
\end{array}
$$

A continuously evolving database contains entries for the history of each object and

[18] Adapted from Howarth (1994).

the interactions between them. An ongoing interpretation procedure follows the scripts to construct episodes which provide the desired behavioural descriptions.

An alternative approach (Hivis-Watcher) relies on a dynamic form of Bayesian network (Howarth & Buxton 1993) to provide a task-based control system identifying relevant objects in the scene which potentially fulfill the given surveillance task (Howarth 1994, Buxton & Howarth 1995). Rather than collecting information on all scene objects, only data potentially relevant to the task is processed.

To accomplish this, a dynamic Bayesian network (DBN) is used to combine the relative nearness, or proximity, measurements between two scene objects over time. The evolving network structure reflects the changing proximity relationships between objects in the scene where the relevance of that relationship towards the surveillance task is determined by a static Bayesian belief network (BBN) called TASKNET. If the temporally evolving relationship is deemed interesting and requires further attention, an "agent" is allocated to each of the objects in the relationship. The pair of agents is overseen by TASKNET which builds a coherent interpretation of the evolving relationship and is capable of terminating the attention should an uninteresting situation arise. The TASKNET receives data from its agents indicating the nearest object and the bearing to that object with respect to its frame of reference (i.e. a deictic reference such as "behind-me"). Using that information, the evolving network has a simple structure containing nodes representing the composite relationship obtained from the two deictic orientations (e.g. back-to-back or trans-overtaking-back) and the likely episode they represent (e.g. overtaking, following or queueing).

Howarth and Buxton claim that this Bayesian network approach can improve the interpretation process by incorporating what one is looking for (top-down expectations) with what could be appearing (bottom-up inference) to overcome the problems of uncertainty and incompleteness in the evaluation of behaviour. This dynamic Bayesian network has also been used successfully at other levels, for example tracking (Buxton & Gong 1995).

2.3.3 Summary

This section has looked at variety of systems capable of providing conceptual descriptions from image sequences. For our purposes, the most appropriate systems deal with high-level systems capable of recognizing and interpreting dynamic processes and situations within the real world. At a computational level, it is apparent that some form of symbolic representation of the scene (or Geometric Scene Description) assists the reasoning process. The analogical model of space introduced by (Howarth & Buxton 1992*a*) provides an ideal basis for such a symbolic representation. However, to date, this analogical representation has been provided by hand. We found this undesirable and the first part of our research has been involved in automatically learning a similar representation of space from the extended observation of image sequences (chapter 2). The resulting symbolic representation can then be used for further research into visual surveillance.

For event recognition purposes, typically event models are provided which act as a template to match against a sequence of image frames to recognize an instance of that event. As previously indicated, it is our intent to not only recognize sequences of actions, but to learn what those sequences of actions are. This means that as part of our research it is necessary to automatically generate an event model similar to that generated by Neumann & Novak (1983) (figure 2.4). From the "overtake" event model, it is clear that the relative position between two objects is required. Also, the relative direction of motion will be necessary for other events where the objects are not travelling in the same direction (for example, giving way). From section 2.2, the most appropriate qualitative reasoning systems are those described by Mukerjee & Joe (1990), Hernández (1991, 1994), Freksa (1992*b*) and Schlieder (1993) who each provide alternative methods for dealing with orientation and direction. The method we choose to use is most similar to Mukerjee & Joe (1990) for relative position and Hernández for relative direction of motion.

In the next section we provide a brief review of traditional machine learning methods and discuss our requirements indicating which learning method is the most appropriate to those requirements.

2.4 Machine Learning Techniques

Machine learning is the specific subfield of Artificial Intelligence that studies the automated acquisition of domain specific knowledge. Traditionally, the study of machine learning was reserved for the development of knowledge based (expert) systems. However, it has become of much wider relevance throughout the entire field of AI — learning can be important in any domain requiring intelligence. Although the study of machine learning is important in the process of automating knowledge acquisition, it is also relevant to the more philosophical question of understanding the nature and general principles of human learning.

This section will deal with some of the major paradigms that have emerged over the period of research of machine learning.

2.4.1 Neural Networks

Neural networks are one of the earliest approaches studied in machine learning (Nilsson 1965). They derive their name from the basic representation of knowledge and the computational style which is inspired from studies of biological nervous systems (i.e. the manner in which nerve cells (neurons) transmit impulses in the human brain). In other words, this approach attempts to create learning machines that operate in a similar way to the human brain by constructing them with components that behave like biological neurons.

Typically input nodes in the network are connected to a set of binary sensors which indicate if a particular feature is present or absent. Present features activate initial nodes and the weight of the links from the active initial nodes determine which subsequent nodes will be activated. This activation process iterates until the final node level is reached which produces the output.

Learning within a neural network consists of the incremental modification of link relations between input and output nodes which improves the mimicry of the desired relation. Pattern classification is a typical goal and learning is mostly supervised (although unsupervised learning is now receiving much attention) by providing a set of labelled training sets.

The simplest and most understood form of neural net is a (single layer) perceptron (a term first used by Rosenblatt (1958), who also first suggested using software to model the network rather than hardware). A layer of input nodes is connected directly to a single output node (see figure 2.6). If the sum of all link weights from the active input nodes is greater than some threshold then the output node is activated. The network learns when a classification error is made. If the output node is not active when it should be, the incoming link weights are lower than the threshold so all link weights are increased by a small constant. When the output node is active when it shouldn't be, the incoming link weight values are too high so they are decremented by a small constant. An alternative modification method (Widrow & Hoff 1960) uses a "least mean squares" (or LMS) function to modify each link weight differently to reduce the mean-square error between the desired output and the generated output.

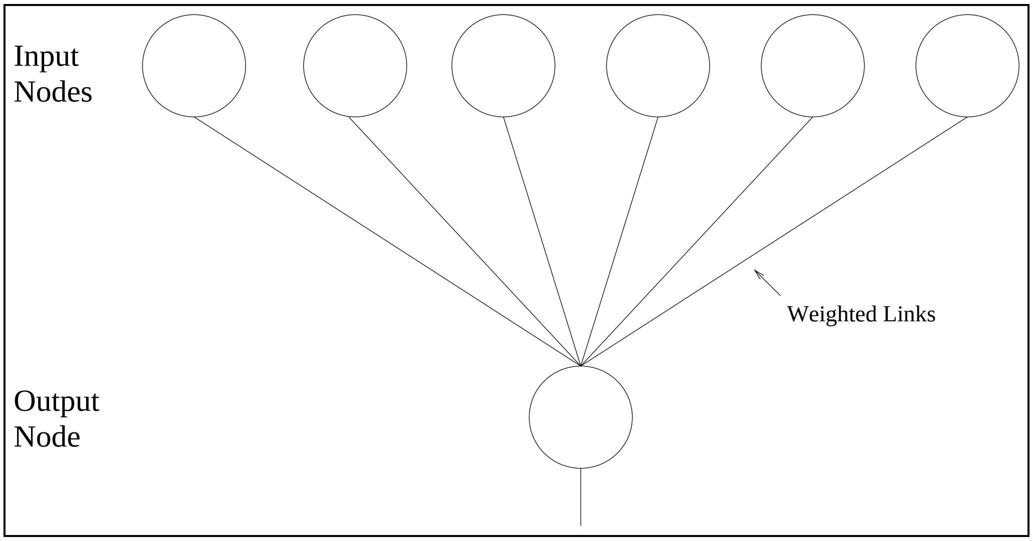


Figure 2.6: Single layer perceptron.

More complicated neural networks contain multiple layers. Intermediate (or hidden) nodes are (indirectly) connected to the input and output nodes of the network. Learning link weights for such networks is not trivial. Typically, a back propagation method is employed which applies the LMS procedure recursively through the network. Nodes are activated through the network in the usual direction, then based on the difference between the observed outputs and the desired outputs, back propagation computes the desired activation level on hidden nodes one level back using LMS. Back propagation now treats the level of hidden nodes as the output nodes and applies LMS recursively until it reaches the input nodes.

Multi-layered perceptrons are the most frequently used neural network. They perform a functional approximator, which provides a mapping between input and output nodes, allowing a wide range of applications including image interpretation (Hopgood, Woodcock, Hallam & Picton 1993) and path planning in robotics (Meng & Picton 1992).

The interest in unsupervised learning within neural networks has increased considerably within the last few years offering the possibility of exploring the structure of data without direct classification. A number of iterative clustering algorithms have been developed (known collectively as *Vector Quantizers*) for this purpose — for example: K-means clustering (Krishnaiah & Kanal 1982); the Gaussian Mixture model, or adaptive K-means, and Kohonen networks (Kohonen 1984).

Unfortunately, neural networks do not tend to produce a symbolic representation of the learned knowledge which is desirable for the spatial model. This means that learning through a neural network is not really practical for the work we describe in this thesis.

2.4.2 Learning from Observation

Another family of machine learning is collectively known as *empirical* learning (or learning from observation). The goal of empirical learning techniques is to provide a general description which characterizes a collection of observations. Making use of the descriptive generalization, the system is then capable of making predictions on novel cases. Typically, the training cases and the acquired knowledge employ a relational or structural representation (e.g. propositional clauses).

The most common *supervised* empirical learning techniques (meaning that training cases have been classified prior to training) are (production) rule learning (for example Michalski & Chilauski (1980)) and decision tree construction (for example Brieman, Friedman, Olshen & Stone (1984)). Probably the most widespread *unsupervised* empirical learning method is conceptual clustering (for example Michalski & Stepp (1980)).

Production rules represent the domain expertise as a set of conditions and actions. The rule conditions test the properties of a case and the rule actions specify the classifications. It is possible to learn the rules by starting with the most specific description and then remove or relax the conditions using a generalization operator. Alternatively, it is also possible to start with the most general case and, using a specialization operator, add or constrain the rule conditions. These approaches generally rely on the fact that pre-classified training instances are (usually) partially ordered according to generality.

The candidate-elimination algorithm (Mitchell 1977) employs both methods to conduct a bidirectional exhaustive search to identify the conditions for classification rules. Unfortunately, the algorithm assumes that a single, conjunctive rule can describe each class and that the training set is free from noise. To compensate, another methodology (for example Clark & Niblett (1989)) applies heuristic search to limit the computational expense. The heuristic search looks for individual rules which can discriminate between true and false instances of a class. During search, the candidate rules are minimally specialized (or generalized) in all permutations and then evaluated for predictive accuracy on the training instances. The most accurate permutations are further specialized (or generalized) and reevaluated. Search terminates when the new rules are no more accurate (statistically) than the previous set.

In *decision trees,* the set of training instances are presented to the system at the same time. Then the learned knowledge is represented in a tree-structure where:

- Each non-terminal node of the tree specifies some attribute to test.
- Each branch leaving a node specifies an alternative value, and
- Each terminal node represents a specific class.

The classification procedure (for a new case) iterates through the tree, testing each non-terminal node attribute and following the relevant branches until reaching a terminal node which provides a classification for that case.

The most common learning technique employs a classification rule to partition a collection of instances according to a selected domain attribute. In a recursive procedure, each partition is then processed by the same classification rule with a different domain attribute. An evaluation function selects the most discriminating attribute of the instances "contained" in a partition — these attributes form the non-terminal nodes. A sub-tree is complete when all instances in a partition have the same classification.

For *conceptual clustering*, the learning mechanism is supplied with an unlabelled set of instances and is expected to form "useful" concept descriptions. The learner determines how to cluster the instances and builds the description for those clusters, typically, in the form of a hierarchy or taxonomy of the concepts. Superficially, the structure is similar to that of decision trees, but each node in a concept hierarchy has an associated concept description that is used during classification. Also, different search and evaluation functions are employed.

As with neural networks, learning from observation does not typically provide a representation of the learned knowledge which is useful in visual surveillance — thus it is not pursued further in the work described in this thesis.

2.4.3 Explanation-based Learning

When learning from explanations, the emphasis is more on the compilation of existing domain knowledge into a more efficient form rather than creating new, or extending existing, knowledge. Unlike most other machine learning paradigms, explanation based learning is analytical, as opposed to inductive, using the domain knowledge to guide the deductive processes that compile the knowledge into a more useful form. Typically, knowledge provided by explanation based learning methods provides an efficiency benefit that favours problem-solving tasks.

One analytical approach compiles explanations into rules. The domain knowledge is specified as a set of inference rules or goal decompositions. Given a top-level goal, the system performs an AND-OR search to obtain a set of primitive actions, states or beliefs that achieve the goal. The knowledge search results in a proof tree, or explanation, for the achieved goal[19]. Explanation based learning makes use of the generated proof-tree to create a summary of the search that can simplify future search methods for a similar goal (for example, DeJong & Mooney (1986) and Mitchell, Keller & Kedar-Cabelli (1986)). By focusing on a relevant problem feature, the learner can summarize the problem-solution pair as a (new) general rule. As a result, similar problems can now be solved in fewer steps.

A similar approach uses an alternative search method rather than AND-OR search. State-space search applies a sequence of operations to the problem states in order to achieve the desired goal. Typically, the states and goal embody specific configurations while the operators specify preconditions and the postconditions after performing the action. Once the problem-solving system has discovered a set of operators leading from the initial state to the desired goal state the entire solution path can be composed into a single rule or *macro-operator* (Iba 1989). The conditions of the macro-operator include all the initial problem state aspects and the postcondition include all those actions not undone by rules in the solution path. As a result, the problem-solver can take larger steps which effectively shortens the length of the solution path. Typical examples include the eight puzzle and blocks world but learning macro-operators can also be applied to more complex tasks such as planning (Minton 1985).

Explicit control rules (or meta rules) can be employed in a means-end planning system (Minton, Carbonell, Knoblock, Kuokka, Etzioni & Gill 1989) to guide the selection of states to expand, operators or inference rules to apply, and the variable bindings for those operators. Should no explicit control rule be available, the problem solver defaults to a depth-first search and then attempts to explain the success or failure using (axiomatized) general knowledge of problem solving. The AND-OR explanation can then be compiled into a (new) control rule for future use.

[19] This sort of reasoning is supported directly by, for example, Prolog.

Although explanation based learning does not provide knowledge in the format we desire for visual surveillance, the search methods described here may be useful. We utilize a depth first search with the intention that if it provided too slow we could improve the search method to improve the performance. However, experimental results showed no performance issues using only depth first search, so other search methods were not pursued. This may change if the requirements change in future research.

2.4.4 Analogical learning

One of the most recent approaches to problem-solving and learning methods relies on the analogy of new experiences to the SPECIFIC knowledge of previously experienced problem situations. Mounting evidence suggests that humans (partly) rely on previous experience to guide problem solutions. *Case-based* reasoning exploits this idea using AI systems designed to classify new cases and formulate solutions based on the evidence of specific cases already held in memory. A good introductory text to case-based reasoning can be found in Aamodt & Plaza (1994).

In case-based reasoning, a *case* usually refers to a problem situation. A case-base (knowledge base) maintains previously experienced problems along with the corresponding solution in such a way that it can be reused in the solving of future problems. When a new situation is experienced, the problem solver attempts to match the new problem against the solved problems. Should a matching case be discovered, the previous solution is applied to the new situation — if the solution fails, the reason for that failure is identified and stored for future reference. Similarly, case-based reasoning can be used for classification problems where the stored solution predicts the desired classification.

Aamodt & Plaza (1994) have described the case-based reasoning method as a cycle described by four processes (and illustrated in figure 2.7[20]):

1. RETRIEVE the most similar case(s).
2. REUSE case solution to solve problem.

[20] Adapted from Aamodt & Plaza (1994)

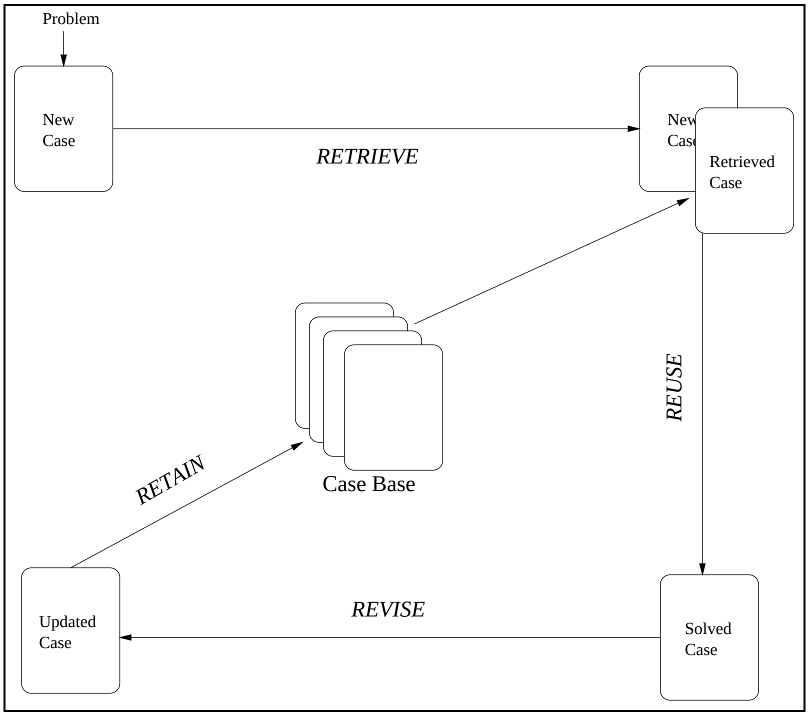


Figure 2.7: The Case-Based Reasoning Cycle.

3. Revise the proposed solution (if necessary).

4. Retain the new solution as part of a new case.

A new problem is matched against cases in the case base and one or more similar cases are *retrieved.* A solution suggested by the matching case is then *reused* and tested for success. Unless the retrieved case is a close match the solution will probably have to be *revised* producing a new case that can then be *retained.*

There is a number of different retrieval algorithms which have been used to identify the most similar cases to the current problem or situation, including nearest neighbour and analogical matching.

- In the *nearest-neighbour* technique, past instances are stored verbatim, and the best match (for the new case) is retrieved. Typically, the similarity assessment is based on matching a weighted sum of features (for example, the algorithm Kolodner

(1993) uses in the Cognitive Systems ReMind software). Missing information can be supplied directly from the best match.

Variants are possible — for example, the number of retrievals can be expanded to increase predictive accuracy by using a weighted average of the retrieved cases (Stanfill 1987).

- The nearest-neighbour technique presents relatively few problems for feature-based or attribute based representations. However, serious complexities to the match process can be introduced in domains requiring a structural or relational knowledge representation. In such domains, the two cases are unlikely to match exactly and some form of partial matching based on semantical similarities is required (in other words, *analogical matching*). Typical examples include PROTOS (Bareiss 1988), in which each case feature has a degree of importance assigned for the solution of the case, and CREEK (Aamodt 1991), which has a similar mechanism although values for the predictive strength of a feature and that features criticality (i.e. the potential influence the lack of the feature has on a case solution) are stored.

Analogical learning most closely matches our requirements in producing a symbolic representation of the learned knowledge although we have adapted the technique to allow an iterative learning strategy (as discussed in the summary that follows).

2.4.5 Summary

Although we have outlined a number of different machine learning techniques there is a number of requirements necessary in our research. In particular, the learning method must be capable of forming conceptual structures from the unsupervised observation of real world situations. Also, to reduce run-time storage requirements and to allow real-time learning an iterative method is desirable. These requirements immediately restrict our learning methods by removing “explanation-based” approaches (which cannot match our real-time requirements) and “neural networks” (which during the learning cycle tend to take extended periods of time and do not produce a symbolic representation). From our requirements, an iterative “conceptual clustering” approach or a case-based learning methodology would appear to be ideal learning techniques.

We select a case-based approach with elements of conceptual clustering incorporated into the strategy. Usually, in case-based learning, the abstraction of prior experience occurs in a *lazy* fashion, by which we mean that experiences are not aggressively compiled in anticipation of future use and instead, the majority of processing is saved until actual use occurs. Unfortunately, this can lead to large bodies of information being constructed. Rather than including *all* new cases in the database, we attempt to search the database for an existing equivalent entry. If successful, we merge the new case with the existing case in such a way as to maintain as much information (from the separate cases) as possible. Although equivalent entries can be overlooked, a verification step at the end of the learning period can discover any remaining equivalent entries. This method is an adaptation of current case-based learning strategies and is introduced here to meet our requirements for real-time learning as the information is made available.

Chapter 3

Generation of Semantic Regions

3.1 Introduction

As discussed in the previous chapter, event recognition provides a significant challenge for high-level vision systems and explains the impetus behind the work described in this thesis. 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 (André et al. 1988, Retz-Schmidt 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. Figure 3.1 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 figure 3.1*b* is an area of behavioural significance representing the location where vehicles must await oncoming traffic before turning right.

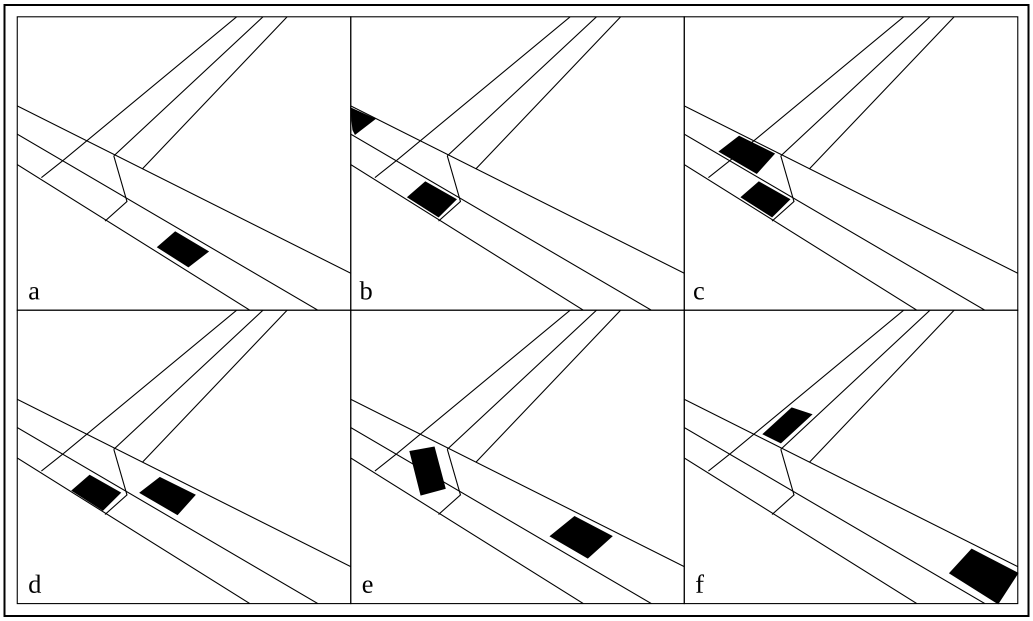


Figure 3.1: A simplified spatial model of a road junction showing a sequence of object locations. A vehicle approaches a junction (*a*), reaches it (*b*) and then awaits oncoming traffic (*c & d*) before turning right into the new road (*e & f*).

As discussed in the previous chapter (section 2.3), Howarth & Buxton (1992*a*, 1992*b*) introduced an analogical representation of space for spatial event detection in the domain of traffic surveillance. This representation is both flexible and multi-purpose and maintains the underlying structure of the domain in a usable form. A ground plane projection of the scene is defined using a *hierarchical* structure based on (two-dimensional) *regions.*

Overall, the spatial model they introduced is a (ground-plane) segmentation of a scene composed of two kinds of regions: leaf regions and composite regions (for more information refer back to chapter 2 section 2.3.2).

Howarth (1994) produced such representations of space manually for each new domain: a time consuming and painstaking process. In this chapter, we demonstrate a method to generate a similar (2D) spatial structure automatically for strongly stylized domains through the monitoring of object movement over extended periods. Following Howarth & Buxton, we will continue using the names “leaf” and “composite” to describe regions — these names adequately indicate the hierarchical region structure. However, we decided against producing a ground-plane projection of the spatial model (although it is possible, see the discussion in section 3.3.5). A number of factors contributed to this decision:

- To project a 2D representation of space onto the ground-plane relies on an interpretation system that can accurately determine the depth of all points in the image plane.
- Alternatively, a 3D model of space could be constructed relying on potentially hazardous 3D data obtained from the tracking process. Typically, such 3D positional information is not sufficiently accurate — meaning that assumptions have to be made for the (3D) spatial model which adds uncertainty to the reasoning process.
- Finally, we decided it would be useful to discover the extent to which automated visual surveillance can be conducted just in the image plane.

In this chapter, we discuss:

- Our initial approach to leaf region generation (section 3.2.1). This is based on simple segmentation techniques and looked promising but was eventaully dropped due to problems obtaining a satisfactory segmentation matching our requirements.

- Our second region generation method (section 3.3) based on the observation of object movements over extended periods which has proved much more successful.

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. Our approach, based on the extended analysis of moving objects, is less limited being able to successfully follow objects with more complex behaviour patterns like a vehicle turning a corner.

Johnson & Hogg (1995) demonstrate a related approach in which the distribution of (partial) trajectories in a scene is modelled automatically by observing long image sequences with the image data applied through a neural network. However, for our requirements, this method is limited by not yielding the symbolic structures we desire.

3.2 Initial Approach

3.2.1 Outline

Various image segmentation techniques already exist and, initially, it appeared possible for a *leaf* region segmentation of the scene to be obtained using such techniques. Typically, the intention of early image processing is to divide an image into a number of parts (*regions*) bearing a strong correlation to physical objects or their parts. As such, image segmentation tends to be one of the most important steps in the analysis of an image. With regions identified, subsequent intermediate and high-level vision processes can be used to identify objects in the image. Of course, there is a lot more to the analysis than indicated, but object identification is typically the highest level of processing attained.

This means that simple segmentation techniques (which concentrate on single frame or static images) alone will not be sufficient to generate the desired leaf region segmentation — some of the regions we want to identify exhibit a “semantic” nature with no visual distinction. In other words, at certain areas within a domain, objects may be observed displaying a particular behaviour but the area itself has no physical features which can discern it from the adjacent areas of space (for example, the area on a road where a vehicle would await oncoming traffic before turning right — as shown in figure 3.1). Such behaviours can only be observed over time (unless *a priori* system knowledge is provided about general vehicle behaviours) and it is highly unlikely that such areas can be located using a single image of the scene. However, we are not restricted to a single image. Rather, we have access to an entire sequence of images where typical behaviour patterns can be observed.

In this approach, the intention is to analyse the movement of objects throughout a scene observed by a static camera. Analysed data, corresponding to the location of moving objects in the scene, is used to generate a mapping of the scene representing the frequency and distribution of all object movements over a training period. The resulting map shows changes in intensity gradient similar to a grey scale image and, as such, simple image segmentation techniques may be applied in order to (hopefully) obtain a leaf region segmentation for the scene. Although sufficiently accurate results were not obtained, it is still worth covering the process. Figure 3.2 shows a diagram outlining this initial approach.

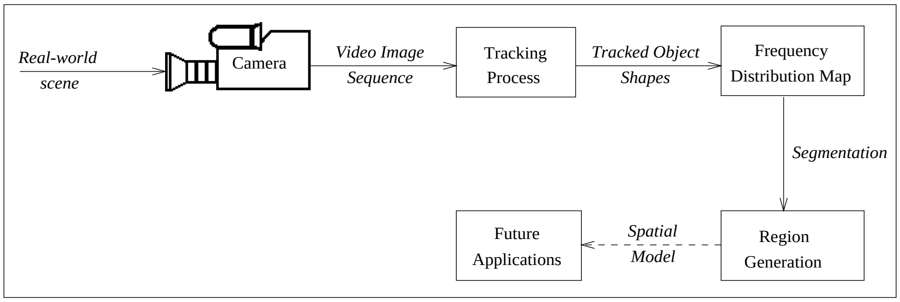


Figure 3.2: Overview of the initial approach.

The three main stages are:

- A *tracking* process obtains shape descriptions of moving objects (section 3.2.2).
- *Frequency distribution map* generation builds an image map of the scene showing the frequency and distribution of all objects moving throughout the scene (section 3.2.3).
- Traditional image *segmentation* techniques are applied to the frequency distribution map to generate leaf regions (section 3.2.4).

3.2.2 Tracking[1]

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 (figure 3.3*a*); an extremely busy dual carriageway (figure 3.3*b*) as well as a predominantly pedestrian scene (figure 3.3*c*).

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. Table 3.1 provides an example of the object descriptions provided by the tracking program and figure 3.4 gives a diagrammatic representation of the shape descriptions for a number of frames.

Although this method does not handle occlusion and is not particularly robust[2], it provides sufficient information for our purposes and it proves significantly faster than the active shape model described in Baumberg & Hogg (1994*a*).

[1] The same tracking process is used in both the initial (unsuccessful) approach and the improved (successful) approach.

[2] Slight camera movement or rapid changes in contrast can mask moving objects and incorrectly identify "noise" as an object until the camera stabilizes.

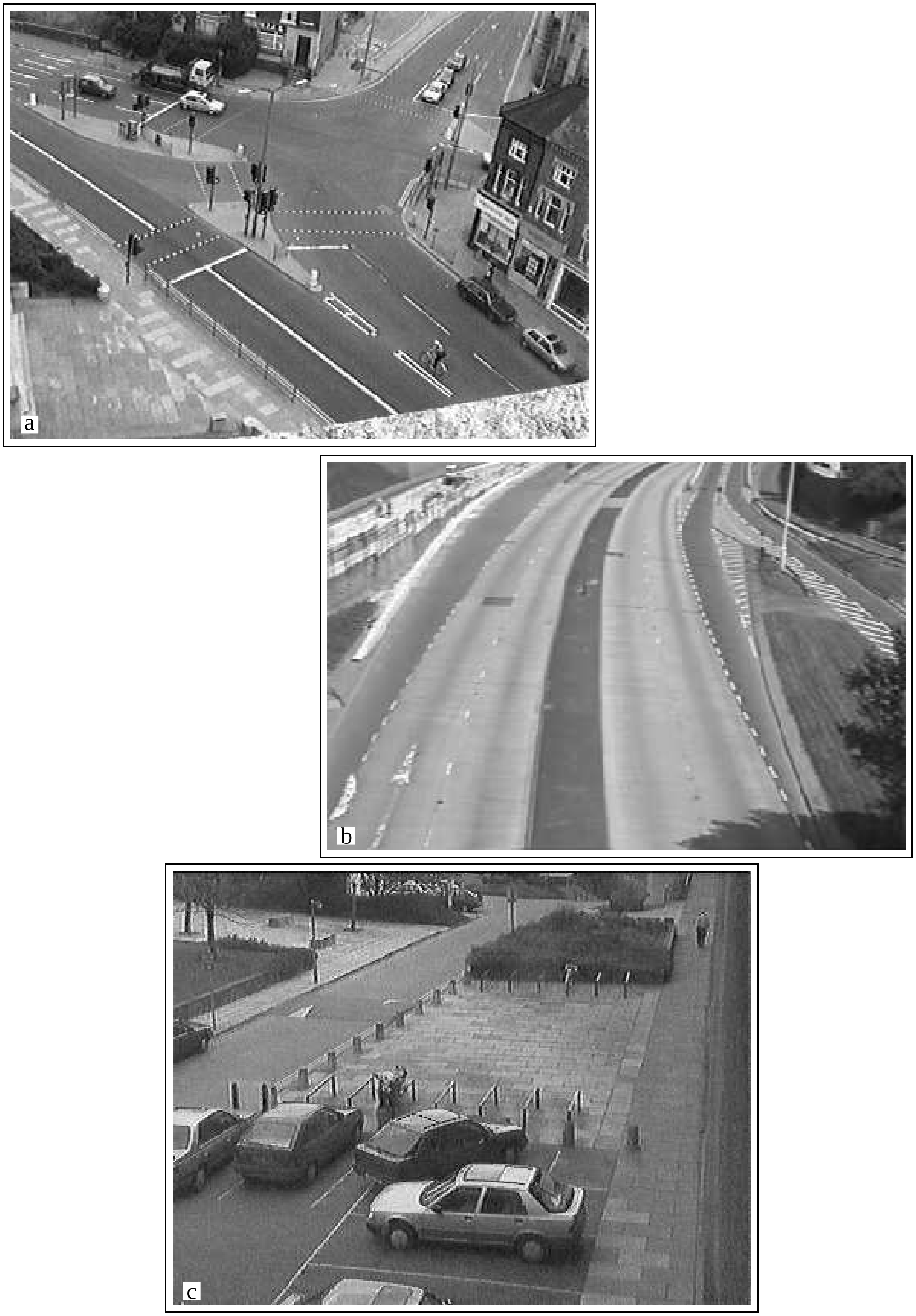


Figure 3.3: Example of test domains viewed from a static camera.

```
list_length = 14
label 1
origin (63.6117,105.466)
width 40
height 40
direction (1.26633,8.59552)
control_points (16.0495,16.1407) (16.0392,4.17215)
(11.6806,-3.36599) (6.75029,- 15.0023) (-1.38,-16.1585)
(-10.5585,-13.7835) (-20.858,-7.83551) (-10.2149,2.96858)
(-8.07889,11.7172) (0.613198,16.7709)
label 2
origin (187.019,121.255)
width 40
height 40
direction (0.822992,-4.49786)
control_points (-8.28647,13.1029) (-1.62511,18.7274)
(11.4624,14.098) (12.799,3.  75626) (14.7826,-6.10941)
(11.8223,-18.3557) (3.075,-12.3313) (-10.4461,-14.9221)
(-16.5301,-4.50971) (-15.7635,6.29561)
label 3
origin (222.969,150.985)
width 32
height 32
direction (-0.382882,-1.71961)
control_points (-5.38405,10.012) (-0.372716,12.7569)
(8.25357,8.55943) (7.84044, 1.58996) (12.7036,-4.8138)
(6.73559,-9.92285) (-0.5392,-8.6016) (-7.30112,-10.0489)
(-13.3605,-1.13137) (-7.05462,2.06805)
label 4
origin (101.171,174.59)
width 40
height 40
direction (1.54284,-1.14579)
control_points (13.3009,18.0155) (14.2404,9.90429)
(12.8495,-6.59882) (-1.12397, -6.7817) (2.99911,-16.3373)
(-1.74721,-11.8686) (-9.90921,-6.21254) (-19.8662,-0.174143)
(-1.72763,5.55178) (-6.53926,18.0309)
```

Table 3.1: Output from tracking application.

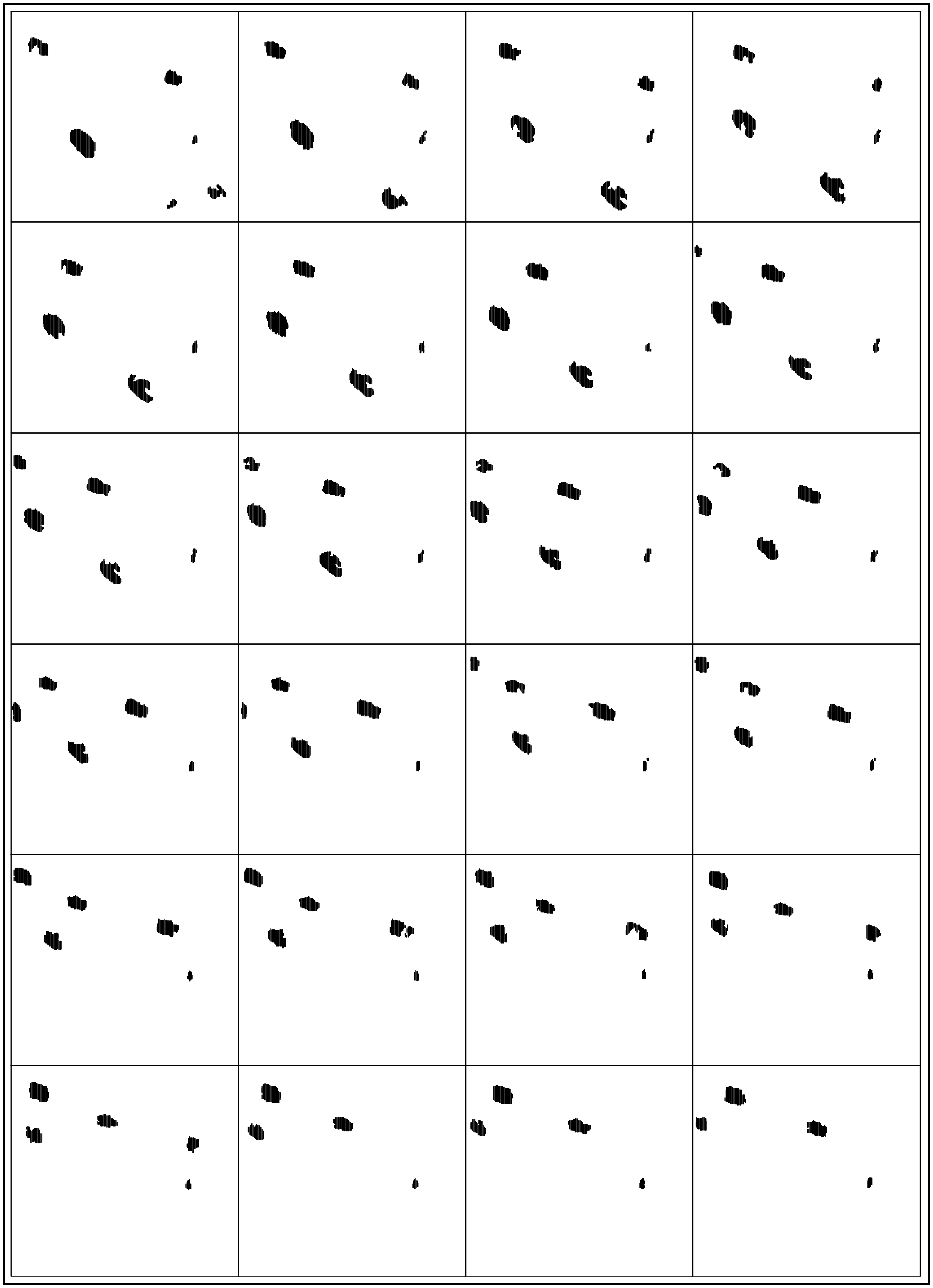


Figure 3.4: Object silhouettes for a short sequence of frames.

3.2.3 Frequency Distribution Map

In this original approach, our intention was to use the shape descriptions from moving objects, obtained from the tracking process, to build a "mapping" of the scene describing the frequency and distribution of all objects travelling throughout the scene (this became known as the "frequency distribution map" or FDM). Each point in the FDM corresponds to a pixel in the scene and indicates the total number of objects that have passed through that particular pixel.

The shape descriptions for each moving object are supplied on a frame-by-frame basis. In each new frame, all points within the FDM which match the pixels occupying the silhouette of each objects' shape are incremented. However, in adjacent frames the pixels occupying the silhouette of a moving object are likely to overlap. When this occurs, the object will have an undesirable impact on some points within the FDM. The resulting value of a point in the FDM is supposed to represent the total number of objects that have passed over the corresponding pixel in the scene. If the silhouette of an object overlaps a previous location then the shared pixels will contribute at least twice to the FDM value, *not* just once. To compensate, all pixels occupying the silhouette of an object's shape in both the current frame and the previous frame are discarded. The remaining pixels are then combined with the FDM:

i.e.

$$d(x, y) = d(x, y) + (\neg f_{i-1}(x, y) \& f_i(x, y))$$

where $d(x, y)$ is the FDM value at image position (x, y) and $f_i(x, y)$ indicates whether the pixel (x, y) is covered by the silhouette of an objects in frame i.

Although the remaining pixels (corresponding to a particular object) in the current frame may correspond to locations held in other earlier frames (than the previous) the number of such "overlapping" pixels should be minimal. This means that only a slight variation may occur between adjacent points in the FDM which should have no significant effect on subsequent processing.

In an attempt to filter out inadvertently tracked “noise”, the pixels occupying the silhouette of an object’s shape are not combined with the FDM until the second frame that the object appears in. Usually random “noise” will appear only in a single frame; by waiting for the second appearance, an object is more likely to be genuine rather than just “noise”.

When an object is perceived as having been stationary for two or more frames we do not discard the overlapping pixels. In this situation, the location occupied by the stationary object may be important in discerning significant behavioural regions (for example, that location may be a give-way zone). As such, we want the pixels composing the location occupied by the stationary object to have a greater impact on the corresponding points in the FDM. Should the overlapping pixels be discarded that impact would be lost.

Typically the length of the image sequence will be about 10–15 minutes, although it could be significantly longer. On completion, there is a strong correlation between the properties of the FDM and a grey scale image. The value contained at each point in both a grey scale image and the FDM represents the intensity of some property. In a grey scale image this property is light and in the FDM that property is object passage. In fact, figure 3.5 reconsiders the intensity values within an FDM as light intensity values to provide a visual representation of that FDM. Traditional image segmentation techniques operate on a function relying on image intensity values and as such these same techniques can be applied to the FDM.

3.2.4 Segmentation

The number of image segmentation techniques that currently exist is already quite large and these are adequately detailed in a number of sources (c.f. Castleman 1979, Hall 1979, Gonzalez & Wintz 1987, Boyle & Thomas 1988, Schalkoff 1989, Sonka et al. 1993). Rather than devising any new segmentation techniques, traditional methods such as

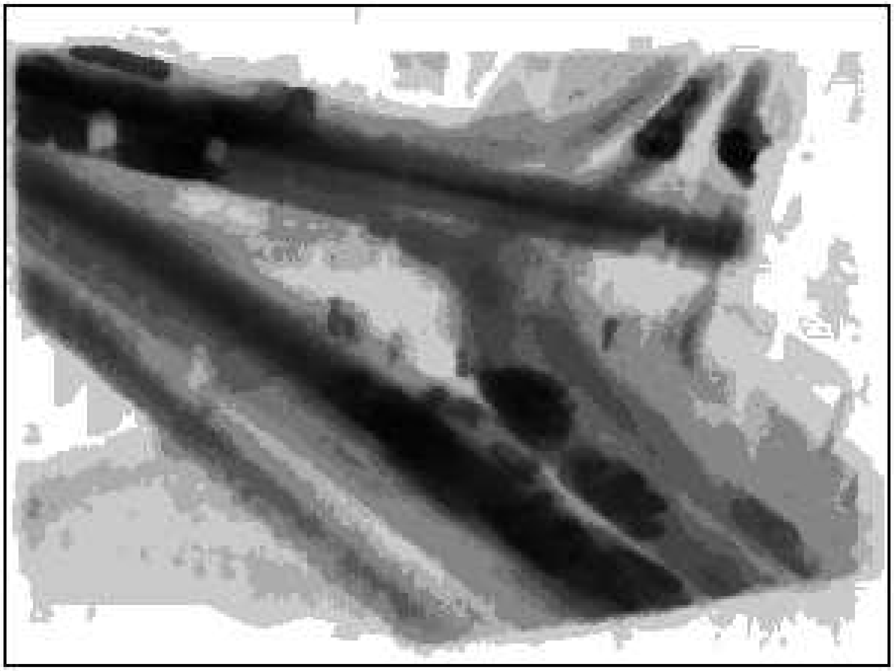


Figure 3.5: Frequency distribution map displayed as a grey scale image.

edge detection, crack-edge relaxation, region identification and region growing have been applied to the frequency distribution map with varying degrees of success (see figure 3.6).

Although, strictly speaking, edge detection and crack-edge relaxation are not segmentation techniques they can both be used to detect borders and subsequent processing may provide the desired segmentation. For instance, one possibility with a sparse edge representation would be to apply a general Hough transform to generate continuous lines for a complete border map. Regions could then be identified from closed areas.

Unfortunately, the results obtained from segmentation did not match our expectations. In gradient edge detection obtaining a suitable threshold value was not possible and the resulting edge image was either too sparse (insufficient edges) or too full (too many edges — figure 3.6(a)). Substantial improvements to the edge image could be obtained by the application of image preprocessing techniques (such as histogram equalization, smoothing and median filtering) to the frequency distribution map (figure 3.6(b)) although the results are still insufficient to construct a leaf region map from. Crack-edge relaxation (figure 3.6(c) produces similar results to the preprocessed image with gradient

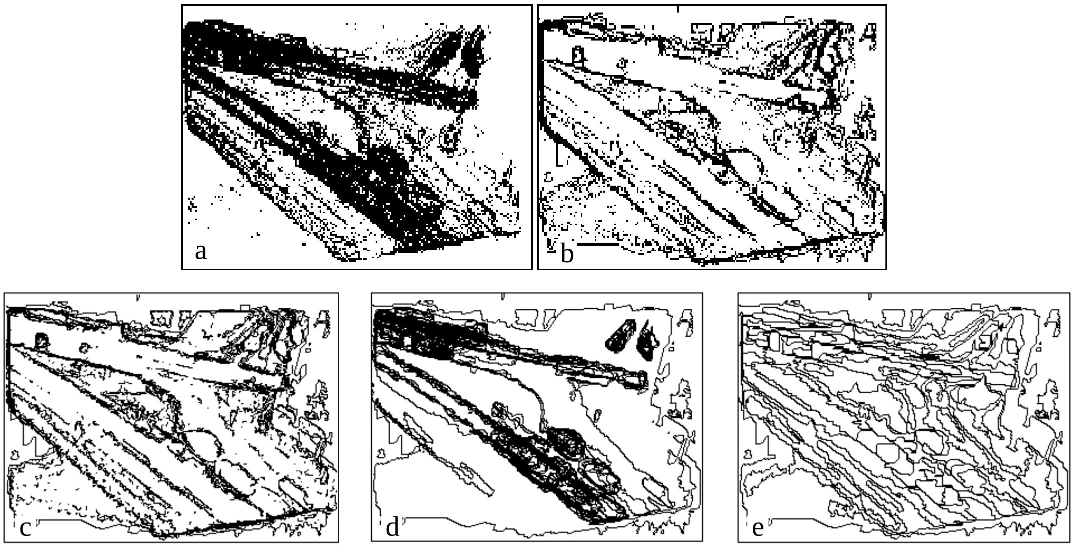


Figure 3.6: Borders obtained by segmenting the frequency distribution map using a number of techniques; (a) (Sobel's) gradient edge detection, (b) FDM image preprocessed before (Sobel's) edge detection, (c) Crack edge relaxation, (d) Region growing using phagocyte heuristic, (e) Regions grown until larger than a specific size.

edge detection — in other words the results still do not match requirements.

Region identification generates regions directly from the frequency distribution map by grouping together adjacent map position that have the same (or similar) intensity values. However, the variation of adjacent values can be considerable and many small regions can be constructed. Region growing methods attempt to recursively merge adjacent regions according to some criteria. The *phagocyte heuristic* merges regions based on the number of weak-edges and the shortest perimeter length between two adjacent regions. Unlike the previous methods, this method does result in actual regions. However, selecting suitable thresholds is still very difficult and the resulting segmentation from all attempted thresholds did not provide the desired results. Across the image, essential boundaries are dissolved while other unnecessary boundaries remain (figure 3.6(d)). In an attempt to improve the results, a further region growing method was applied which merges regions based on size and shortest perimeter length (figure 3.6(e)). Again satisfactory results were not obtained.

3.2.5 Discussion

Although the frequency distribution map showed initial promise and by normal human vision it is possible to discern potential regions, subsequent image segmentation was unable to produce satisfactory results. Early results were quite encouraging but we were unable to improve these sufficiently to produce the desired leaf region segmentation. A number of factors contribute to this lack of success:

- The desired effect of the FDM was based on the observation that abnormal or unusual object behaviours occur significantly less frequently than "normal" behaviours. Over a typical training period, the amount of abnormal behaviour occurring will be relatively low. So, the areas where such behaviours occur should be overwhelmed by the information obtained from "normal" behaviour patterns (i.e. there should be a minimal amount of variation between points in the FDM where unusual behaviour has occurred and points in the surrounding area). As a result, areas where abnormal behaviour has occurred should be indistinguishable and not be identified in the segmentation process.

  Unfortunately, some routes through the domain which correspond to acceptable behaviour patterns are also used relatively infrequently. Consequently certain desirable regions cannot be found — for example the intersecting area where pedestrians cross a road may not be identified due to the small number of pedestrians compared to the large number of vehicles.

- The outline of an object provided by the tracking process is heavily affected by shadows and reflections caused by lighting conditions. From frame-to-frame, the silhouette of a tracked object's shape may change substantially and the overlap between non-adjacent frames will be greater than expected. Our earlier assumption that 'the number of such "overlapping" pixels should be minimal' is actually inaccurate. This allows an undesirable amount of "noise" to be introduced to the FDM disrupting the segmentation process.

- Typical problems affecting any image segmentation task and the subsequent identification of relevant regions also occur. For example: finding the most appropriate threshold values or locating too many borders in one area with too few in another area. It is possible that improved results could be obtained using a hybrid segmentation method (reference).

Even after addressing these considerations, it is not certain whether a satisfactory leaf region segmentation could be obtained. Also, once a desirable leaf region segmentation is discovered there still remains the problem of composite region generation.

3.3 Improved Method

3.3.1 Outline

As with the initial approach, 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 figure 3.7.

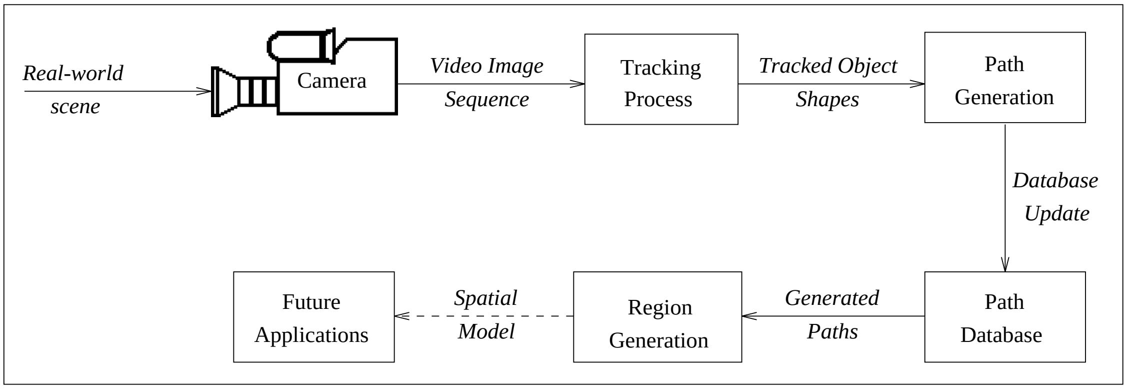


Figure 3.7: Overview of the improved method.

There are three main stages:

- As with the initial approach, a *tracking* process obtains shape descriptions of moving objects (section 3.2.2).
- *Path generation* builds a model corresponding to the course taken by moving objects and subsequently updates the database of paths (section 3.3.2).
- *Region generation* accesses the database of paths so that leaf and composite regions can be constructed for the spatial model within the domain (section 3.3.3).

3.3.2 Path Generation

A *path* is defined as the course that an object takes through the domain. More specifically, the spatial extent of an object's path is determined by the combination of all pixels occupied by that 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, considering the observation made in our original approach, object outlines are subject to various forms of noise. In particular, light reflections can alter the object silhouette dramatically (figure 3.8*a*). When combined, such object locations may produce a jagged path (figure 3.8*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 (figure 3.8*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 simple to convert them to a polygonal representation (Sonka et al. 1993, Chapter 6.2.5, pp. 212–214).

The convex hulls combine to give a significantly smoother path (figure 3.8*d*,) that is more likely to be correctly matched during database update.

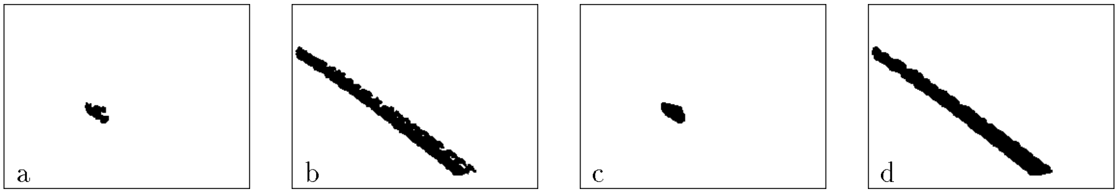


Figure 3.8: Example showing advantage when using convex hull of object outline. (a) Object outline, (b) Path generated using object outline, (c) Convex hull of object outline, (d) Path generated using convex hull of object outline.

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:

- 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[3]. Of course, this value is scene specific and will be discussed more in section 3.3.4.

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 analogous to arithmetic *addition* — allowing the value of each pixel for a database path to indicate the number of equivalent paths sharing that pixel (as with the frequency distribution map described in section 3.2.3).

At the end of the training period, each path held in the database will contain frequency distribution information for that path, figure 3.9*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, figure 3.9*b*.

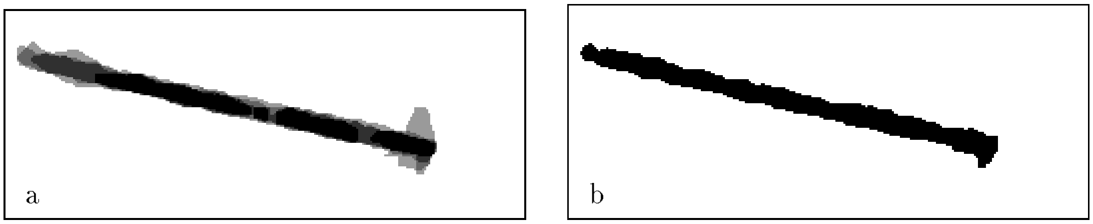


Figure 3.9: Obtaining the most "common" path; (a) Original path displayed with a grey scale representation of the frequency distribution and (b) Most common path obtained by thresholding the distribution.

3.3.3 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

[3] But see the discussion in section 3.3.5.

resumed) and uses that information to build the regions. A new region 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. The spatial model may change frequently and the latest underlying region map may differ substantially to that in the previous state. Without an accurate mapping between the adjacent states, object behaviours may prove difficult to interpret.

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 statistically "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 continuous region generation.

Alternatively, this operation could be performed during the database update process. The resulting database entry, after a new path is merged into the database, could then be reprocessed to check for any further equivalences. However, this operation may prove to be the bottleneck for real-time processing. It is possible that several database merges may be necessary before previously unmatched paths become equivalent. This means that several database update checks may be required. However, if the test is left until the start of the region generation stage, then any equivalent paths can be found in a single "verification" pass. In fact, experimental results have shown that fewer database checks and updates are made when using a single path verification process rather than continuous update.

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 here 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 binary 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 section 3.1, 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. The most appropriate adjacent region selected for the merge is obtained by considering the smoothness of the resulting merged regions. 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 (see figure 3.10). Figure 3.11 displays the leaf regions obtained for the test domains.

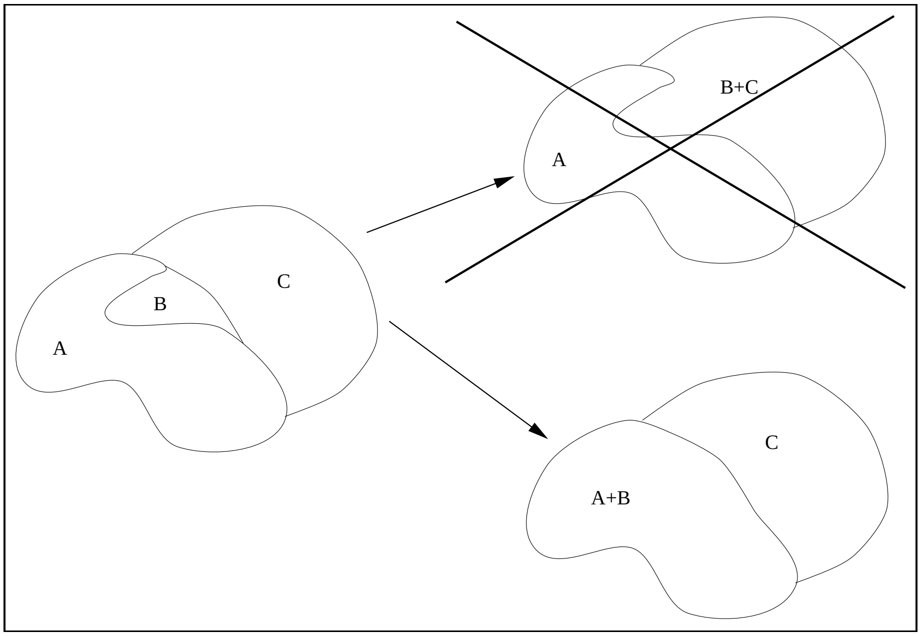


Figure 3.10: Merge operation for "useless" small regions.

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 figure 3.11 along with the identified leaf regions.

When complete, the spatial model is in raster format. Although this may be suitable for some applications, for storage, a vector representation is much more efficient. As such, a raster-vector conversion is applied to the raster data and then output to a "map-file" (as used by Howarth (1994) and shown in chapter 2 section 2.3.2). The

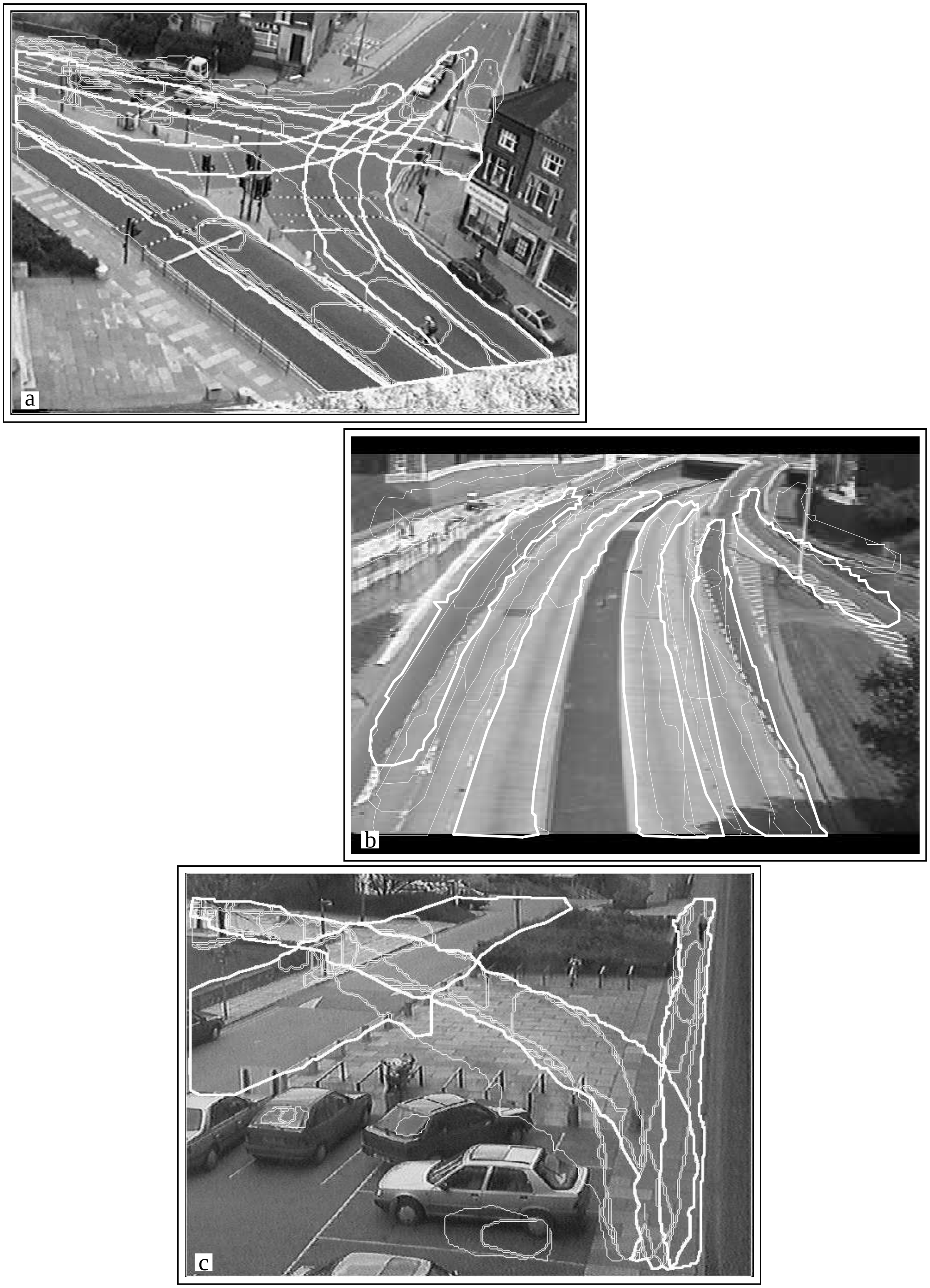


Figure 3.11: Test domains; (a) Road junction, (b) dual carriage-way and (c) pedestrian scene displaying identified leaf regions along with a selection of composite regions.

obtained spatial model is then composed of composite regions, leaf regions, line segments and points.

3.3.4 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. For the dual carriage-way, again about 10 minutes of video footage is used, this time with up to 20 objects in 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 dual carriage-way approximately 150 leaf regions are obtained from 21 recognized paths and in the pedestrian scene about 120 leaf regions are generated from 23 recognized paths.

These results rely on three threshold parameters we were unable to eliminate from the system. Thresholds remain necessary for the overlap value in the path equivalence test, the actual threshold operation used to obtain binary path representations and the size of leaf regions that are to be merged into an adjacent region. As previously indicated, the overlap value for path equivalence and the path threshold operation are linked — one being the dual of the other. Experimental results indicated that an overlap value of 80% was suitable for each test domain. 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 required 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 so it is quite conservative.

3.3.5 Discussion

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[4]. However, the extent of such stylized domains is sufficiently widespread 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[5]. Howarth & Buxton (1992*a*) 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 suf-

[4] Although, as an anonymous referee pointed out, even with the movement of fish in a pond there may be sufficient stylized behaviour to build a model. For example fish circle the perimeter of a pond and often return to a particular location to eat or to a shaded area in which to rest.

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

ficiently broad representation of object behaviour occurs throughout the training period the derived spatial model should be accurate without being prone to any misconceptions.

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.

Really, only one problem occurs with the generation of the spatial model. Occasionally, when the path database update checks for path equivalence it is not possible to set an overlap percentage that denies all inaccurate matches. In particular, in the test domain where we generate a representation of space for the dual carriage-way we have to deal with acceleration and deceleration traffic merge lanes. In this situation the amount of natural path overlap is extremely high between the merge lane and the inner carriage-way. As a result these lanes are represented as a single (Y-shaped) composite region with a concavity in the spatial model (see figure 3.12). Although it is not possible to increase the percentage overlap because desirable equivalences would not be identified, one possible solution to this minor problem is discussed in the next chapter.

3.3.6 Further Work

As well as the remaining work discussed in this thesis there are a number of possible extensions to the spatial model and applications in which such a representation of spaces may prove beneficial:

- 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. Although the static spatial model allows the easy recognition of "non-standard" events, problems may occur

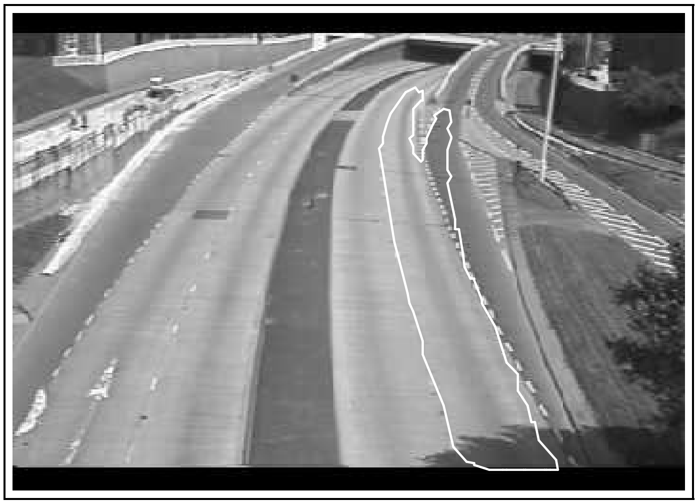


Figure 3.12: Lanes merged through undesirable equivalence determination.

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. It should be possible to enhance the method described here to provide an adaptive model of space.

- 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.
- Currently, the regions identified in the generation process are arbitrarily labelled. Although this is sufficient for visual surveillance, should the model be desired for a natural language interface other properties would also be required — in particular, a natural language description or name for a path represented by a particular composite region would be desirable. User input (or *a priori* scene knowledge) would be necessary for this type of property acquisition.
- At present, the boundaries of regions generated in the spatial model are defined precisely by thresholding the frequency map contained in the path database to obtain the most "common" path. However, the overall spatial extent of objects

using that path can extend beyond the "common" area and into the area we discard. It is possible that we could enhance the spatial model to include indeterminate boundaries similar to the "egg-yolk" representation described by Cohn & Gotts (1996). The "yolk" would represent the "common" area obtained by the usual threshold operation while the complete "egg" is comprised of the complete spatial extent of the database path without the threshold operation. This would provide a more complete representation of space that more accurately describes the passage of (various sized) objects through the domain.

- 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.

3.4 Summary

Within this chapter we have examined an existing representation of space that appears ideal for qualitative reasoning purposes. To date, such spatial models have been generated by hand. We have proposed two possible methods for automatically learning a similar representation of space. Although sufficiently accurate results where not obtained from the first method, it did provide certain insights to the problem that helped form the second approach. Results have been provided for both approaches and a number of possible improvements and further work have also been outlined.

Following the completion of the work described in this chapter we started to think about visual surveillance and, in particular, event recognition. Since we are intending to use qualitative modelling techniques, a method for identifying "close" objects was required without resorting to exact measurements. We address this problem in the next chapter by extending the spatial representation to a spatio-temporal model.

Chapter 4

A Temporal Extension

4.1 Introduction

For event recognition tasks, there is a necessity for an attention control mechanism which can assist in identifying objects of potential interest. Usually, this means any object whose behaviour pattern appears unusual or different to the expected ("normal") behaviour. Typically such situations arise between two (or more) interacting objects. As such, a mechanism which can identify "close" objects is highly desirable. By "close", we refer to the distance between two objects which may affect typical object behaviour (usually just lane or path following). In terms of moving objects, "close" is a function not only of distance but also of speed. Since we intend to use a qualitative methodology, we would like an approach that can automatically classify "close" objects without having to work out the exact speeds and distances of all objects in a dynamic scene.

Object speed can be determined using the formula $v = d/t$. This means that time plays a very important role in determining how close two moving objects are. In fact, time and distance are interchangeable concepts used in natural language. For example, consider the question "How far away are you?". The reply could be in terms of distance "About a mile" or time "About two minutes" In traffic domains, our primary concern, we are told in the Highway Code (HMSO 1996, rule 57) that a reasonable distance to be maintained between two vehicles moving in the same direction, under ideal weather conditions, is approximately two seconds (under adverse conditions this time gap should be at least doubled). When a vehicle enters this space, the behaviour of the two objects becomes more interesting. As such, we identify "close" vehicles by checking the temporal distance between them.

It should be noted that our concept of "close" does not refer directly to spatial proximity. Consider two vehicles moving at 30mph where the second vehicle is following the first and the distance between them is about a car length. At this speed, the distance maintained between the two vehicles is probably safe. However, consider the same situation on the motorway at 70mph, again the second vehicle is about a car length away from the first. In this situation the distance maintained between the two vehicles is significantly more dangerous than at 30mph. Our qualitative use of "close" must be able to identify potentially interacting vehicles which depends not only on distance but also on the speed at which the vehicles are travelling (i.e. the time taken for one vehicle to travel the distance between itself and the other).

In section 2.3 we examined the analogical representation of space introduced by Howarth & Buxton (1992*a*, 1992*b*) and in the last chapter we demonstrated a method of automatically generating a similar analogical representation. If the composite regions, in the spatial model, contain sub-divisions (or regions) of a specific time length (say two seconds), then it becomes possible to classify "close" objects as those occupying the same or adjacent sub-divisions. These are known as *equi-temporal regions* or *ETRs*.

In this chapter, we propose a method which extends our existing approach for the automatic generation of semantic regions to include *equi-temporal regions* which subdivide each composite region.

4.2 Outline

A (slightly modified) tracking process accepts live video images from a static camera. Shape descriptions corresponding to all moving objects within the scene are produced on a frame-by-frame basis. Real-time analysis of the dynamic scene data is performed to build a database of paths used by objects. Further information pertaining to time is also stored in a second (temporal) database. At the end of the training period, data stored in the two databases is processed to generate the (leaf, composite and *equi-temporal*) regions required for the spatio-temporal model. A diagram outlining this system is shown in figure 4.1.

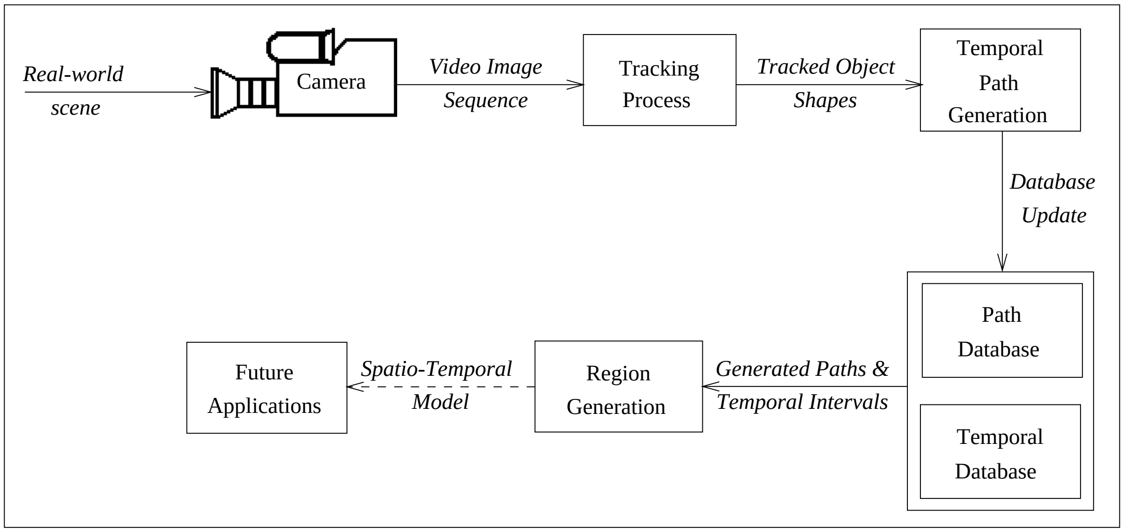


Figure 4.1: Overview of the temporally extended method.

There are three main stages:

- A (slightly modified) *tracking* process obtains shape descriptions of moving objects (section 4.3).
- *Temporal path generation* builds a model corresponding to the course taken by moving objects, complete with a sequence of temporal intervals where each interval has the same passage duration for that object. Subsequently, the database of paths and the database of temporal interval sequences are updated with information contained in the model (section 4.4).
- *Region generation* accesses the database of paths and the database of temporal interval sequences so that leaf, composite and equi-temporal regions can be constructed for the spatio-temporal model of the domain (section 4.5).

Only the extensions to our previous approach will be discussed in this chapter.

4.3 Changes to the Tracking Process

Although essentially the same tracking process is still used, one particular modification is required. In this chapter, we are describing a method to generate equi-temporal regions; as such the timing of each frame is essential.

Due to improved technology, it is now possible to specify exactly how many frames are to be processed each second (up to 30). As a result, the exact duration between one frame and another can be calculated, allowing the precise number of frames in any period of time to be ascertained. This is equivalent to providing a time-stamp for each frame which would have been equally acceptable. The actual frame count selected is 25 frames/second as this is currently the standard (UK) frame rate for full-motion video.

Unfortunately, the process still does not handle occlusion so we restrict the test domains to limit the amount of occlusion occurring in the domain.

4.4 Temporal Path Generation

4.4.1 Analysis

Originally, it was thought that equi-temporal region generation may be accomplished with a relatively simple approach. When the path is generated for each object the number of frames that each object spends travelling through the domain is also recorded (in a log). In the path equivalence test, the log for the number of frames can also be tested against the database entry to determine temporal equivalence as well as (path) area equivalence. Since the temporal equivalence check is between two integer counts there would be no need for a complicated algorithm. At the end of the training period, the average number of frames spent by all objects travelling each path can be used to generate constantly spaced temporal sub-divisions (or regions) for the corresponding composite region. This idea has at least two problems that have to be addressed:

- The most obvious problem is that constantly spaced sub-divisions provide no benefit to the spatial model and are certainly not temporally sized. Although not immediately apparent, constant spacing can only be applied using 2D (screen) coordinate points which, due to perspective, would not provide either constant distance or constant time for the sub-divisions. Even using a ground plane projection would not alleviate the problem due to changes in ground height and actual variations in speed when manoeuvering around corners and bends in a path.
- A less apparent problem may occur when the percentage overlap test between two paths show the paths to be equivalent but the second test for equivalent temporal interval sequences fails. Here, the two paths are really equivalent — only the velocity of one object is greater than the other object. In itself, this is not a problem — the temporal sized regions should be capable of handling a range of velocities and multiple identified temporal sub-divisions would provide that support. However, the spatial extent of a composite region relies on the combination of all equivalent paths (as in the path threshold operation discussed in section 3.3.3). If not all equivalent paths are combined then the spatial extent obtained for the composite region may not be as accurate. Also, the entire method relies on statistical frequency and if a number of temporally different paths emerge, the discrete "equivalent" paths may not prove to be sufficiently frequent to be accepted as composite regions.

From this potential approach and the analysis of the perceived problems it becomes apparent that for equi-temporal sub-divisions it is necessary to take into account camera perspective as well as velocity variations due to (complex) object manoeuvres.

Considering the second problem, discussed above, it is evident that the spatio-temporal model to be constructed should consist of single (separate) composite regions representing each (statistically frequent) object path (as generated for just the spatial model in the previous chapter). However, each composite region is further refined by one or more equi-temporal region sequences identifying the different speeds taken by objects travelling the path (represented by the composite region).

An alternative strategy would construct separate spatio-temporal paths (consisting of a composite region with a single temporal interval sequence), where the spatial extents may be identical. Conceptually, the complete spatio-temporal models are similar. However, as discussed above, for a path to be considered a composite region it requires sufficient (statistical) evidence which would be reduced by matching both spatially and temporally equivalent entries. As such, the first proposal is the method followed.

The complete spatio-temporal representation has a more hierarchical structure than the alternative strategy would produce. However, when using the complete model, it is possible to construct spatio-temporal paths on the fly (i.e. for more efficient storage, the hierarchical model would only need to store the spatial extent once, but when necessary the disparate temporal sequences could be applied to the associated composite region to construct a number of separate spatio-temporal paths with the same spatial extent).

4.4.2 Process Description

Currently, the spatial extent of an object's path is determined by the combination of all pixels occupying the silhouette of an object's convex hull along its course through the domain. This is still accurate. However, for the temporal sized sub-divisions and to account for camera perspective and speed variations over the length of the path, it becomes necessary to maintain a list of point coordinates indicating the location of the object at regular intervals of time. These temporal point coordinates will allow the equi-temporal regions to be subsequently constructed.

As previously mentioned, in this domain two seconds is a reasonable value to identify "close" objects. Therefore, the location of an object needs to be recorded at two second (or 50 frame) intervals. The centroid of an object's outline, which is readily available from the tracking process, is an appropriate selection for the temporal point coordinate — it will not cause bias in subsequent processes when locating objects in the relevant ETR. Figure 4.2 shows two example paths complete with temporal points located at two second intervals.

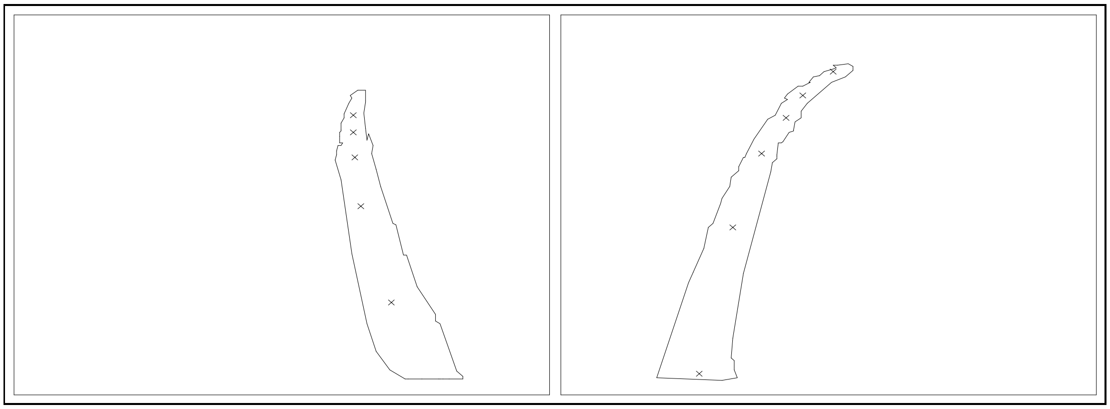


Figure 4.2: The path of an object complete with temporal point intervals.

As in the non-temporal approach, on completion an object's path is merged into the database of existing paths after searching the database for any equivalent entries. The path equivalence test is still the same, based on the percentage overlap of constituent pixels of the new path and the existing database path. If an equivalent path is not discovered in the database then:

- The temporal interval sequence, associated with the new path, is added to a second (temporal) database containing all the alternative temporal interval sequences.
- A link between the (new) temporal database entry and its path is created.
- Then, the new path is added to the path database.

Otherwise, an equivalent path has been discovered and should be revised to incorporate the information contained in the new path. As before, in the non-temporal approach, the new path is combined with the database path using a function analogous to addition. This provides a frequency value for each constituent pixel (of the database path) indicating the number of contributing equivalent paths (see section 3.3.2). Subsequently, the path threshold operation can be applied to generate the composite region.

Now, however, the temporal interval sequence for the new path also requires merging into the database of temporal interval sequences. This time, a temporal equivalence test is performed on the existing temporal interval sequences contained in the database. Not all database entries should be checked — only those associated with the equivalent (updated) database path. For this purpose, each database path entry contains a list of links (relations) to associated temporal interval sequences contained in the temporal database (as shown in figure 4.3). Should no equivalent temporal interval sequences be discovered, the new temporal interval sequence is added to the temporal database along with an associated link to the database path entry.

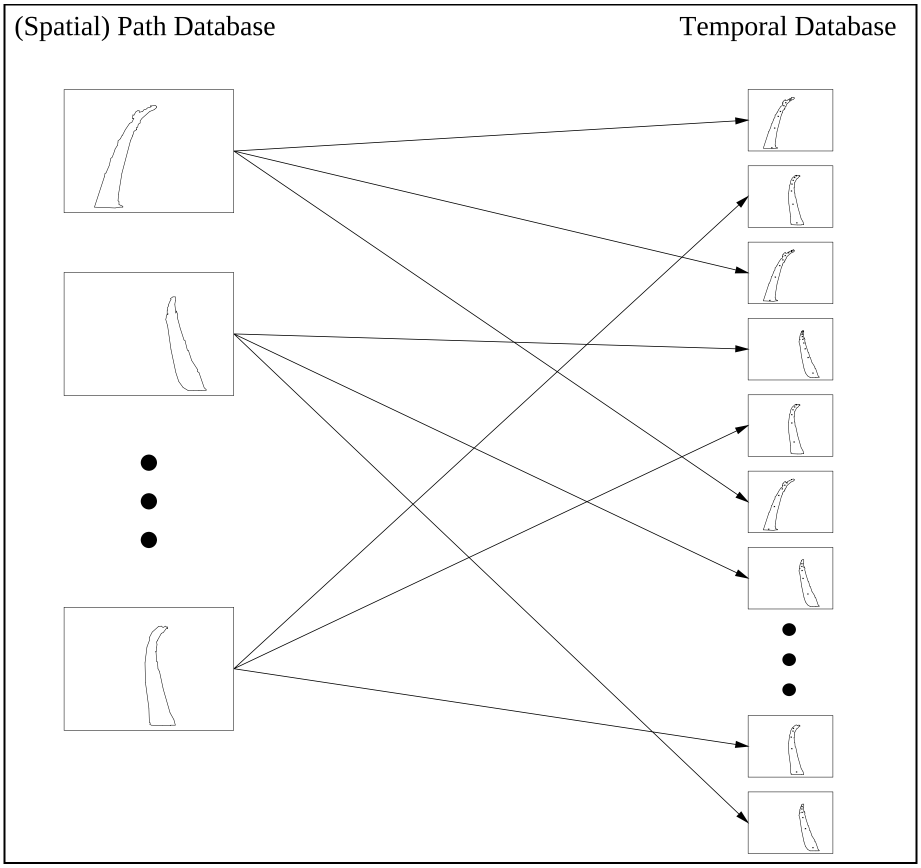


Figure 4.3: Structure of path and temporal database.

Temporal equivalence requires a different type of test to that of path equivalence. Unlike the generated object paths, there are no constituent pixels to coincide so, it is not possible to check a percentage overlap value. Instead, it is necessary to match points in both temporal interval sequences. Objects entering the domain should essentially appear in (approximately) the same location for a particular path. As such, to check whether two temporal interval sequences are equivalent all that should be necessary is to check that the number of intervals correspond and that the length of the corresponding intervals is approximately the same in each sequence.

Unfortunately, the tracking process does not always detect the initial appearance of an object. For example, a small vehicle, entering in the distance, combined with light reflections may not have enough presence to be detected immediately. This means that in the temporal equivalence test a starting point needs to be identified before matching the lengths of the remaining temporal intervals.

It is most unlikely that the starting points and subsequent interval distances will exactly coincide, although that would make the process simpler. Instead, these matches must be approximately the same. More formally, a threshold or *tolerance space* (Mukerjee & Schnorrenberg 1991) is required to provide reasonable matches. The value for the tolerance space changes with each interval to be matched and is calculated from the mean duration of the corresponding temporal intervals to be matched in the two interval sequences.

When checking for a starting position in each sequence, the test is for two corresponding point locations *not* interval lengths. However, a tolerance space is still appropriate and is calculated from the mean length of the temporal intervals on either side of the focus points in the two sequences. If the focus point is the initial point in the interval sequence there is no prior interval so only the next interval length is considered (in that sequence).

The actual value obtained for the current tolerance space is a 20% threshold value calculated from the mean temporal interval lengths. Figure 4.4 shows an example to demonstrate this calculation. From the diagram:

$$t = \frac{\ell_1 + \ell_2 + \ell_3 + \ell_4}{4} \times 20\%$$

As before, this threshold value appears reasonable. If the threshold value was higher more matches would be found and less matches if lower. So, if corresponding starting points can be determined and the remaining temporal intervals are approximately the same lengths, then, the two temporal interval sequences are seen to be equivalent.

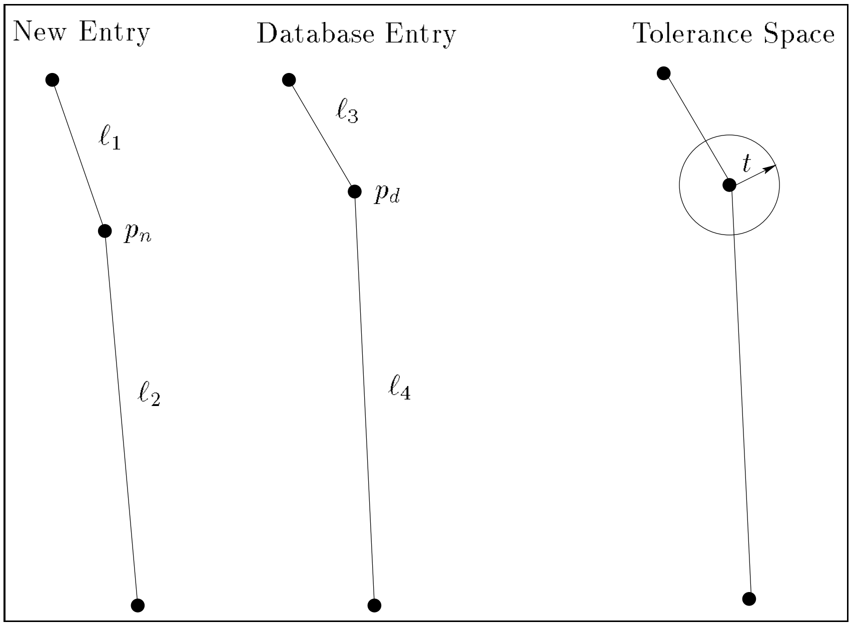


Figure 4.4: Calculation of *tolerance space* for temporal intervals.

When two temporal interval sequences are found to be equivalent, the temporal database entry should be updated. Beginning with the initial points matched in both interval sequences, the mean position for the two points is calculated and the temporal database entry updated. The mean position takes into account *all* temporal points that have contributed to its location not just the two current points — otherwise, each new temporal interval sequence would have a greater effect on the final location of each point in the database entry. As such, the number of contributors for each point in the temporal interval sequence is also required in the temporal database entries.

The calculation is then:

$$(x_i, y_i) = \frac{(x_i, y_i) \times N_i + (v_j, w_j)}{(N_i + 1)}$$

where (x_i, y_i) is the i^{th} temporal point in the database entry which matches the j^{th}

temporal point, (v_j, w_j), in the equivalent temporal interval sequence and N_i is the total number of contributors for the i^{th} temporal point in the database entry.

This algorithm is quite long and described through the text in this section. For clarification purposes, a sketch algorithm of the process is provided in figure 4.5.

```
for each frame
    receive object descriptions
    for each object in frame
        generate convex hull of object shape
        update object path matrix
        every 2 seconds
            record temporal point coordinate
    for each completed object path
        search path database for an equivalent entry
        if equivalent entry found
            merge new path with database path
            search temporal database entries for equivalent entry
            if temporal equivalent entry found
                update temporal database entry
            else
                add new temporal database entry
        else
            add new path database entry
            add new temporal database entry
```

Figure 4.5: Sketch algorithm of path and temporal database generation.

4.5 Equi-Temporal Region Generation

Leaf regions and composite regions are constructed as before in the non-temporal approach (section 3.3.3). Path verification reassesses the entries in the path database to ensure that any "new" equivalences are merged together. The information stored in the database is then analysed to determine which paths occur sufficiently frequently to contribute to the spatial model (i.e. those paths which are recognized as composite regions). Subsequent thresholding and combination of these paths results in the leaf regions and composite regions for the spatial model.

The temporal database entries associated with each of these paths are then processed to generate sets of equi-temporal regions for the relevant composite region. Similarly to the path database, each set of temporal database entries belonging to a particular path is verified to ensure that no equivalences have been created through the update process. Should any "new" equivalences be discovered they are merged together as described in the previous section. However, the calculation for the mean location of the points in the temporal interval sequences has to be generalized to take into account the number of contributors to the point location in both sequences.

i.e.

$$(x_i, y_i) = \frac{(x_i, y_i) \times N_i + (v_j, w_j) \times N_j}{(N_i + N_j)}$$

where (x_i, y_i) is the i^{th} temporal point in one database entry which matches the j^{th} temporal point , (v_j, w_j), in the equivalent temporal database entry. N_i is the total number of contributors for the i^{th} temporal point in the database entry and N_j is the total number of contributors for the j^{th} temporal point in the equivalent database entry.

The temporal verification stage also ensures that the temporal points within the interval sequence are all positioned within the boundary of the generated composite region. It is possible that the threshold operation applied to a path (to obtain the composite region) will leave some of these points outside the resulting area. Should this occur the entire interval sequence is discarded. Typically, this only occurs if the interval sequence has a low statistical frequency — otherwise, the mean location of each point obtained from the combination of more frequent equivalent temporal interval sequences is likely to place those points within the boundary of the resulting composite region. As the next step removes infrequently occurring temporal interval sequences from the database no significant information is discarded.

Each composite region will now have left at least one temporal interval sequence (should there be none then the composite region itself is invalid and should be discarded). Should the composite region have more than one associated interval sequence this would represent objects travelling at different speeds along the path thus containing relevant information. For example, push bikes typically travel slower than motor bikes but are likely to travel along similar paths, or at different times of the day when the traffic is heavier or lighter, the typical travelling speed changes.

The spatial extent of an equi-temporal region is bounded by the line segments obtained from the composite region border and the points to either side of a temporal interval. The line segments (from the composite region boundary) provide the (intrinsic) left and right edges for the ETR, whereas the (intrinsic) front and rear edges are obtained by generating lines passing through the points at either side of the temporal interval.

Although the initial temporal interval has a start point, it is not used to bound the first equi-temporal region (in the composite region) because it occurs at the entry location for new objects — any object entering a composite region should enter into the first temporal region whether before the first point or not. Typically, this will only occur if a small object is detected earlier than normal — which is unlikely. As such, the spatial extent of the first equi-temporal region is bounded to the left, right and rear by the line segments for the composite region boundary and to front by the *second* point in the interval sequence (i.e. the point at the end of the first temporal interval).

Although the left and right edges (obtained from the line segments for the composite region boundaries) are already known for the temporal region, the front and rear edges have to be constructed, This is achieved by considering each temporal point, (x_i, y_i), in turn along with the previous point, (x_{i-1}, y_{i-1}), and next point, (x_{i+1}, y_{i+1}).

The gradient, m_1, of the line joining the previous and next temporal points can be calculated with ease:

$$m_1 = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}$$

When multiplying the gradients of any two perpendicular lines we know that:

$$m_1 \times m_2 = -1$$

Therefore, the gradient of any line perpendicular to the line joining the previous and next temporal points is:

$$m_2 = \frac{x_{i-1} - x_{i+1}}{y_{i+1} - y_{i-1}}$$

In turn, it is now possible to define the equation for a perpendicular line that passes through the current temporal point:

$$y - y_i = \left(\frac{x_{i-1} - x_{i+1}}{y_{i+1} - y_{i-1}} \right) (x - x_i)$$

Using the equation of the perpendicular line it then becomes possible to find the location of the points which intersect the composite region boundary providing the "corner" points for the temporal region (see figure 4.6).

There is a special case for the last point in the temporal interval sequence. Unlike the first point in the temporal interval sequence, objects are still travelling along the path after the last point — they just leave the domain in less than the 2 seconds required for a complete interval. This means that there is a final equi-temporal region at the end of a composite region which occurs after the last temporal interval. In this situation, there are no further temporal points to obtain the line gradient from. Instead, the current (last) and previous temporal points are used in the gradient calculation rather than the next and previous temporal points.

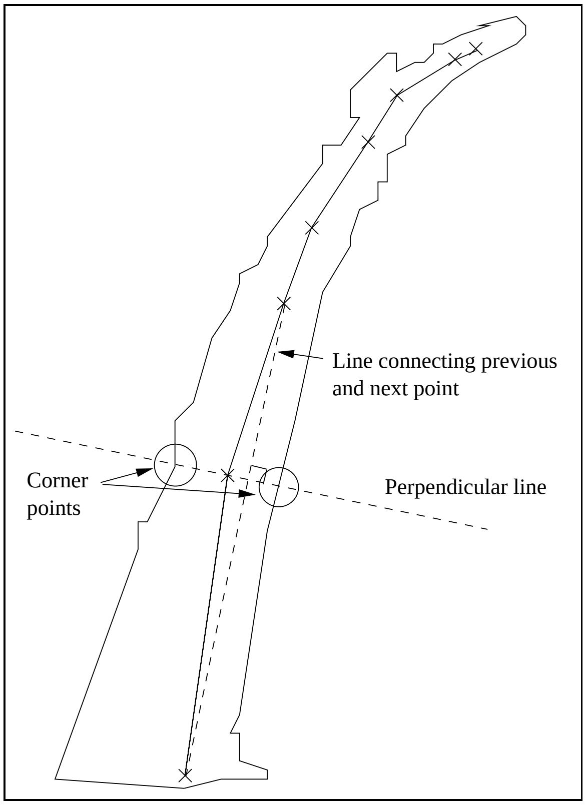


Figure 4.6: Obtaining edge points for equi-temporal regions.

i.e.

$$m_1 = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

where i is the last point in the temporal interval sequence. The remaining calculations are then followed as before.

One complication may occur as a result of two paths being inadvertently merged as a result of unusually high overlap, as in the case of a traffic merge lane and an inner carriage-way. As explained in the previous chapter (section 3.3.5), such a merge results in a (Y-shaped) region with a large concavity. When this occurs, the perpendicular line, passing through a temporal point located on either side of the concavity, will make four boundary intersections, not just two. As we desire all regions to be single piece, the nearest boundary intersections on either side of the temporal point are selected as the "corner" points for the temporal region (figure 4.7). This also provides another benefit that will allow these inadvertently merged regions to be separated. As this idea has not yet been implemented it will be discussed later in the section on further work (section 4.8).

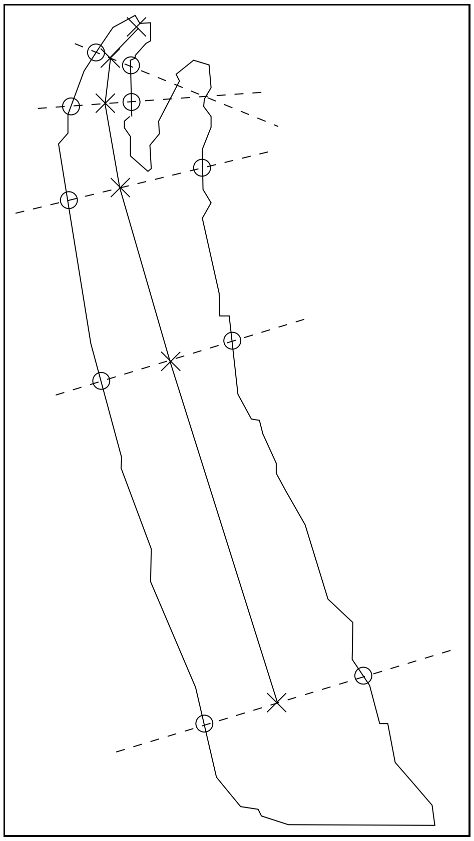


Figure 4.7: ETRs for a Y-shaped composite region.

The spatio-temporal model is complete when each temporal database entry associated with a composite region has been processed. Again, as the algorithm is detailed throughout the section, a sketch algorithm of the region generation process is provided in figure 4.8.

```
at end of training period
    verify path database entries
    for each verified and statistically frequent path database entry
        verify associated temporal database entries
        if path still valid
            threshold database path matrix
            update region map with threshold data
    find and merge small regions with relevant adjacent region
    for each verified and statistically frequent path database entry
        identify leaf regions each composite region
        for each associated temporal database entry
            find corner points for equi-temporal regions
```

Figure 4.8: Sketch algorithm of region generation process.

4.6 Experimental Results

The system maintains real-time performance during the database update stages and is only marginally slower when generating regions (which can still be generated at any time). Results are highly successful, providing a number of alternative sets of equi-temporal regions for the majority of the composite regions. Although some composite regions only show a single set of equi-temporal regions it is still acceptable — typically, objects travel at the same speed along that path. A selection of equi-temporal region sets, contained within their composite regions, are displayed for the dual carriage-way in figure 4.9.

4.7 Discussion

In this chapter, we have presented a temporal extension to our original spatial model and demonstrated a method of automatically generating this new spatio-temporal model still based on the movement of objects throughout the domain. Once again, we have utilized an existing tracking application which provides (2D) shape descriptions for tracked objects from a real image sequence. However, this time the image sequence is processed

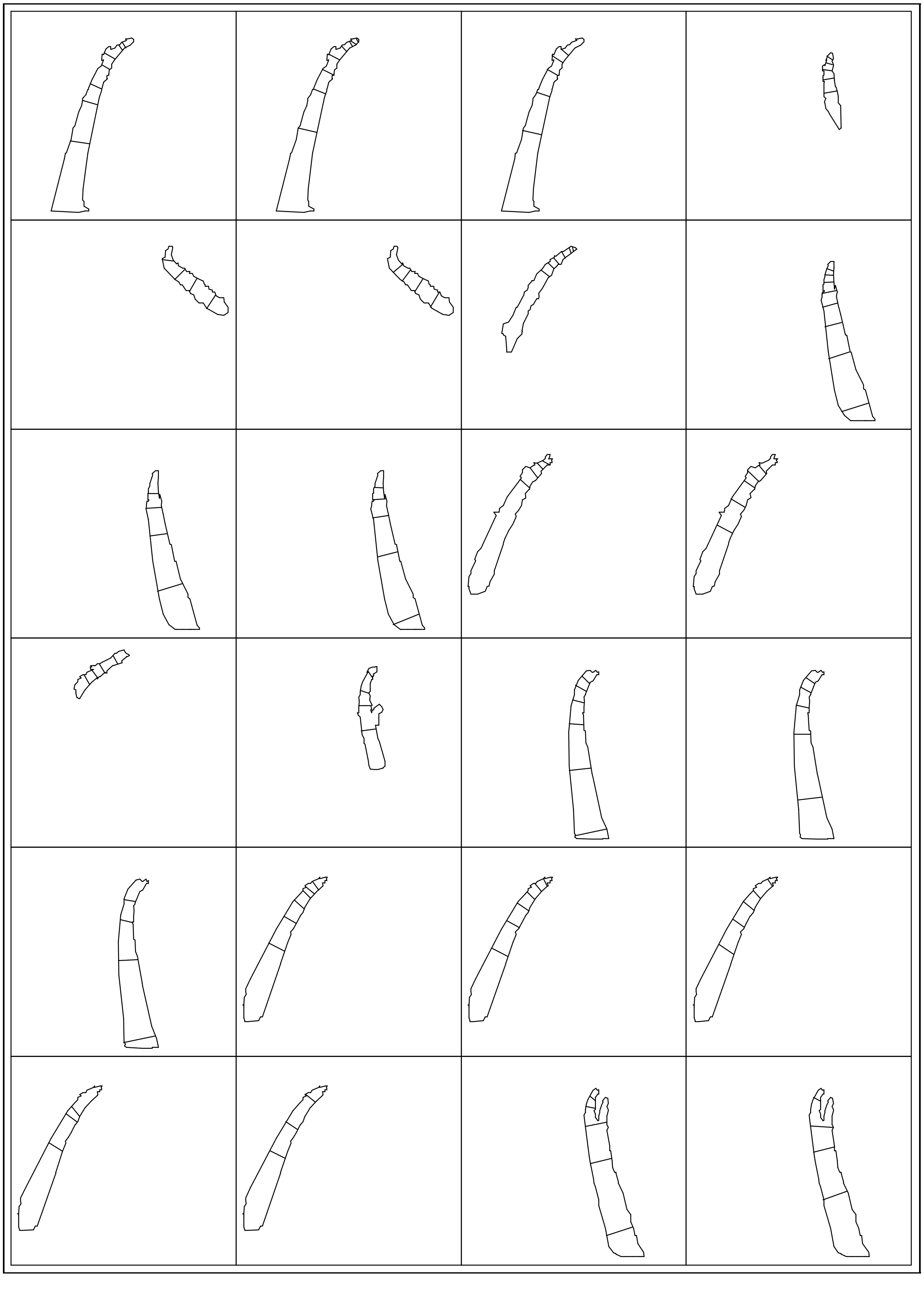


Figure 4.9: Resulting equi-temporal regions.

at a *fixed* frame rate allowing the precise timing of object movements throughout the domain.

The method copes well with equi-temporal region size variations due to camera perspective as well as handling situations where variable object speeds may occur as a result of situated obstacles, such as sharp corners where a vehicle will have to slow down considerably before manoeuvering around that corner.

If we had desired a three dimensional spatial model plus time (3D+t) then it would still be possible to use an alternative tracking process as long as that process allowed the precise timing of object movements — either by a known fixed frame rate or by providing a time-stamp with each frame.

The complete spatio-temporal model appears unique throughout the literature. It allow us to create an attention control mechanism which can identify "close" objects within each frame of an image sequence. As discussed in the introduction to this chapter, for our purposes "close" does not refer to spatial proximity but to temporal proximity. When two objects are considered "close" the time taken for one vehicle to reach the position the other occupies is less than two seconds. In the domain considered during this chapter and the next chapter, temporal proximity provides a useful mechanism for attention control. Consider the sequence of actions resulting in one vehicle overtaking the other. At some stage, the overtaking vehicle is behind the other, at the end of the sequence it is ahead and at some stage during the manoeuvre it is alongside (right) of the other. To recognize this sequence of actions, it is not necessary (or desirable) to examine the relationship between every pair of vehicles. Only those which are considered "close". The attention control mechanism described in the next chapter provides a mechanism which allows us to learn this type of sequence.

However, although temporal proximity provides a mechanism which can learn this type of interaction it is not sufficiently general to capture other sequences of actions — for example, a vehicle giving way at a junction is not moving but is obviously giving way to another vehicle. In this situation identifying "close" objects from the waiting vehicle is not possible as our definition of "close" depends on movement.

4.8 Further Work

One possible improvement that can be made is based on the observation that in some composite regions, the appearance of the last temporal region is not ideal (for example, see figure 4.10). The (perpendicular) line splitting the two regions may be skewed as a result of the line gradient calculation. In all other cases, the gradient of the (perpendicular) line passing through a particular point in the temporal interval sequence is calculated from the *mean* gradient of lines to either side of that point (i.e. the gradient of lines from the current point to the previous and next points in the sequence). Unfortunately, it is not possible to calculate a mean gradient for the last temporal point because there are no further points in the temporal sequence. As such, the line equation is calculated directly from the gradient between the previous point and the current point, assuming that it would provide satisfactory results. For the largest proportion of composite regions, this is actually true. However, in a minority of situations (in the test domains) unsatisfactory results are obtained indicating that the gradient calculated for the line equation could be improved. One possibility would be to base the line equation on a mean gradient using a "virtual" next point which is position midway between the points of the end line-segment of the composite region boundary.

A potential side-effect of the equi-temporal region generation is to provide a method that will allow a composite region that represents two distinct paths to be separated. Such composite regions occur when two adjacent paths have an unusually high percentage overlap. As in the dual carriage-way where the traffic merge lane and the inner carriage-way are combined. These lanes result in a single Y-shape composite region representing both lanes. It is not possible to prevent this merge by reducing the percentage overlap value because other desirable path equivalences will also be lost.

However, constructing the equi-temporal regions may provide a solution to this problem. When constructing the region boundary, an equation representing the perpendicular line passing through a single temporal point is calculated and applied to locate the point intersections with the composite region boundary. Usually, there should only be two such boundary intersections — one on either side of the temporal point. However, in

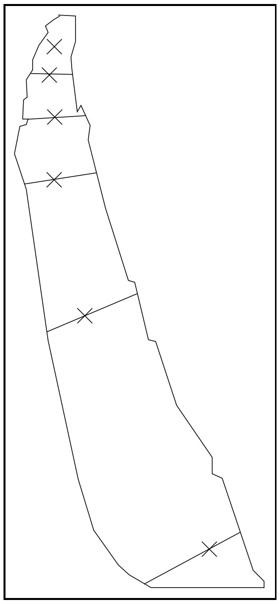


Figure 4.10: Undesirable shape obtained for the last equi-temporal region in some situations.

the undesirable Y-shaped composite region there is a large concavity where the perpendicular line will make four intersections. These four intersections will identify undesirable composite regions. As before, the nearest point on either side of the temporal point is selected for the actual intersection point but the point on the side of the concavity is tagged so that it can be easily identified. The first temporal point after the concavity will result in only two boundary intersections. At this time, it is possible to create a "new" composite region for the path described by this set of temporal intervals. The last tagged boundary point can be connected directly to the new boundary point found in the last intersection and then used to describe the boundary for the "new" composite region (figure 4.11). Similarly, another "new" composite region can be generated when the temporal interval sequence uses the second branch of the original undesirable composite region.

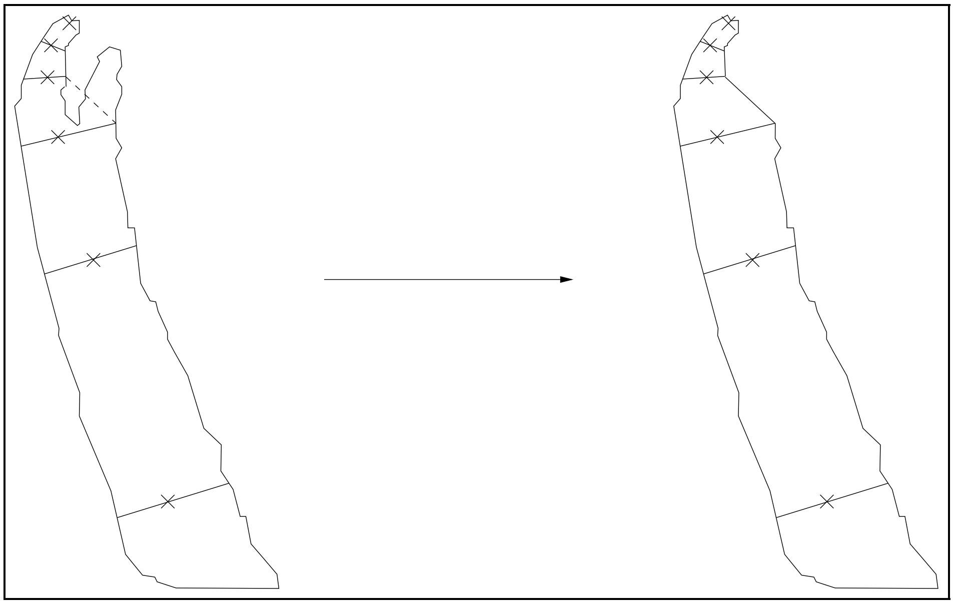


Figure 4.11: Separating undesirable composite regions.

4.9 Summary

Throughout this chapter we have described a temporal extension to the spatial model as well as providing the modifications and extensions to the learning process. Sets of equi-temporal regions sub-divide each composite region adding a further hierarchical component to the model. A single composite region (representing a particular path in the scene) may have more than one set of equi-temporal regions. The typical speed used by objects travelling throughout the scene may vary. For example, take a typical saloon vehicle and a bus manoeuvering around a corner (represented by a single composite region) the speed of the bus may be significantly less than that of the saloon. Similarly, traffic may travel faster out of rush hour. A single set of equi-temporal regions may not be sufficient to successfully handle these different travel speeds — meaning that two (or more) sets of equi-temporal regions are required.

The temporal aspect discussed within this chapter is context specific and obtains its accuracy from statistical evidence provided by real data. As such, precise timing is essential to generate an accurate model. This timing is provided by the (external) tracking program complete with improved hardware technology which allows us to process a specific number of frames each second. Alternatively, the tracking program could have attached a time stamp to the data received each frame which would have been just as adequate.

Through the (quantitative) frame information, the composite region occupied by each object can be identified along with the correct temporal sub-division. This identification procedure provides us with a method for locating “close” objects without resorting to exact measurements (which are further complicated by camera perspective). Being able to locate “close” objects is desirable as it limits the amount of processing by focusing the attention to potentially interacting objects. We refer to this “close” object location mechanism as “attention control” and provide a more complete description in the next chapter. We also explore an application of the spatio-temporal model combined with the attention control mechanism that allows us to learn qualitative event models (in contrast to providing such models as *a priori* system information).

Chapter 5

Event Learning

5.1 Introduction

The driving force behind the development of an automated technique to 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 (c.f. chapter 2) should be able to provide a more manageable way of handling this data if a formal framework for the given situation exists. By qualitative reasoning we refer to a methodology that only requires a minimum amount of critical information necessary to perform a specific task — as such qualitative information tends to be task oriented.

The spatial model described in chapter 3, being topologically based, is able to provide a formal framework suitable for a number of qualitative reasoning tasks, for example simulation, prediction and event recognition. In this chapter we concentrate on *event learning*. Unlike previous approaches where generic event models are provided as part of the *a priori* system information we propose a method that allows the automatic generation of contextually relevant qualitative event models. In this instance, by using qualitative reasoning our intention is that the derived event models will contain only the critical information (using a qualitative logic) necessary to recognize future instances of the events that have been modelled. We will demonstrate a case-based learning method that is capable of analyzing objects' locations, movements and the relationships to other objects in order to generate the desired event models automatically.

As already indicated, the event learning process relies on the representation of space we addressed in chapter 3. However, the spatial model alone was determined to be insufficient in locating objects of potential interest — in particular, it provides no mechanism for recognizing "close" or interacting objects. As such, an attentional control mechanism was deemed appropriate and can be achieved using the temporal extension (described in chapter 4) to our original spatial model.

5.2 Outline

Again, a tracking process accepts live video images from a static camera providing shape descriptions corresponding to each moving object within the scene on a frame-by-frame basis. An attention control mechanism locates each object in the correct composite region and the equi-temporal region being occupied. "Close" objects of potential interest are identified and a qualitative history of object relationships updated. The database (case-base) is updated from the associated object histories (cases) of each object leaving the domain. At the end of the training period event models can be obtained from the statistical analysis of object histories contained in the case-base. A diagram outlining this approach is shown in figure 5.1.

There are five main stages:

- The same *tracking* process previously used obtains shape descriptions of moving objects (sections 3.2.2 and 4.3).
- A *Classification* stage allows the identification of qualitative position and direction from the quantitative information provided by the tracking application (section 5.3).

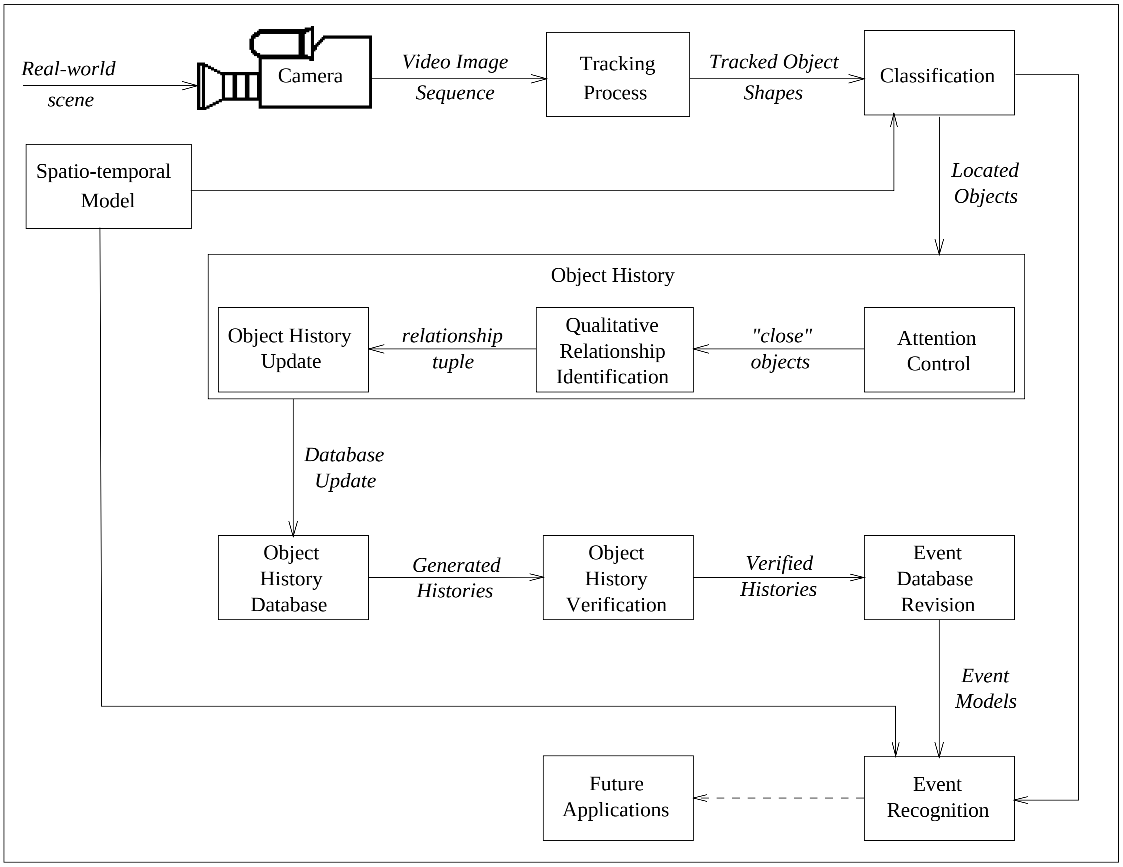


Figure 5.1: Overview of the temporally extended method.

- *Object history* generation uses an attentional control mechanism to identify "close" objects and the qualitative relationships to those objects are added to the object history (section 5.4).

- *Object history verification* analyzes each object history (case) to ensure that all relationship transitions are valid (section 5.5).

- In the *Event database revision*, each valid object history is added to the case-base. On completion of the training period, statistical analysis can determine event models from the object histories contained in the case-base (section 5.6).

5.3 Classification

The first step in generating a history for each object is correctly identifying the position of each object within the spatio-temporal model and classifying the direction each object is travelling in. For each object location, the composite region being occupied has to be established along with the correct equi-temporal region within that composite region. Essentially, this classification of position can be seen as data reduction in terms of converting the unnecessary quantitative location into a more (for our purposes) desirable qualitative location.

To this end, the database containing the spatio-temporal model is processed to produce a (two dimensional) leaf region map where each position indicates the leaf region occupying that pixel in the scene. Region borders lie between pixels and as such will cause no classification problems. Shape descriptions for each tracked object can be processed to provide a silhouette "mask" which can be located on the leaf region map. Any corresponding points will indicate the set of leaf regions overlapped by the object. To reduce potential errors (see below), the number of points overlapping each leaf region is also counted and if less than a predetermined threshold that leaf region will be ignored. In this case, the predetermined threshold is 10% of the object size. Each object and leaf region has to have a minimum size of 10 pixels (from the tracking application parameters and the removal of small regions). Therefore, 10% of the minimum size is a single pixel — the smallest discernible unit.

The database for the spatial model can then be queried to determine which composite regions contain that particular subset of leaf regions. It is possible that more than one composite region will be identified. In such cases the principle of momentum is applied to make the same composite region categorization as in the previous frame.

The potential errors, mentioned above, that may be created by not pruning out leaf regions with minimal occupancy include misidentifying the correct composite region. By removing those leaf regions, the "core" leaf regions being overlapped still remain and prove sufficient for the identification of the correct composite region.

An alternative method for identifying the set of leaf regions is more complex. Rather than re-building the leaf region map, each leaf region in the database can be processed against the (polygonal) object outline to determine:

- *Contains* — All line segments for the leaf region outline completely surround the line segments for the object outline — in this case only one leaf region is identified.
- *Overlap* — Line segments for the outline of a leaf region intersect the line segments for the object outline — at least two leaf regions will be identified.
- *Contained-by* — Line segments for the object outline completely surround the leaf region outline — again, at least two leaf regions will be identified.
- *Discrete* — No intersections and no occupancy between the object and leaf region being processed.

Other relationships can also be identified (see chapter 2, e.g. proper part of, tangential proper part of, equal...) but are not important for this categorization — all that is needed here is the set of (partly) occupied leaf regions. The problem of error reduction would involve a polygonal area calculation of the intersecting lines. This method for region identification was not pursued due to no perceived benefits. The potential benefits obtained by using vector data do not apply in this situation as that vector data was initially constructed from raster data. As such, sufficient accuracy and (improved) timing of composite region classification can be obtained by re-building the leaf region map.

For equi-temporal region classification a different method is used. In the equi-temporal region generation process, the centroid point of an object's outline helps determine the end points for a temporal interval. As such, if an equi-temporal region contains the centroid point of an object, then it is potentially occupied by the object. In mathematics, the equation of a line can be used to determine which side of that line a particular point occurs on.

i.e.

$$ax + by + c = 0$$

When the coordinates for a point not on the line are substituted into the equation the result will be +ve or −ve, indicating that the point is to the left or the right of that line. (To use Schlieder's (1993) terminology, the point order is either clockwise or anti-clockwise.) The composite region being occupied by an object is already known, meaning that only the front and rear line segments need to be checked.

Each set of equi-temporal regions is processed to determine which equi-temporal region in each set contains the centroid point. The most appropriate equi-temporal region is determined by matching the temporal interval distance (i.e. the distance between the front and rear line segments) to the potential distance moved by the object over 2 seconds (or 50 frames — the original time period when constructing the equi-temporal regions). The potential distance to be moved by an object over 50 frames can be calculated after two frames (by multiplication). Although this distance is likely to be wrong due to camera perspective and actual speed variations it will be minimal over 2 seconds. Also, successively improved distances can be calculated as the object proceeds. The object is classified as belonging to the equi-temporal region where the temporal interval distance is closest to the calculated (potential) distance moved by an object over 2 seconds.

A deictic frame of reference (as discussed in chapter 2 based on the position of the camera is used to classify the direction being taken by a moving object. This allows information provided directly from the tracking application to be used in the classification procedure. With each object description (after the first) a quantitative direction vector is provided indicating the direction just taken by that object. For a qualitative classification based on 45 degree zones (Hernández 1994) all that is required is to convert the vector into an angle and find the relevant zone. Internally a scale of 1–8 is used (see figure 5.2) to represent the qualitative direction. As such, the easiest classification method uses two mathematical functions:

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \text{ to obtain the angle in range } [-\pi, \pi]$$

and

$$dir = \frac{((\theta + \pi + \frac{\pi}{8})\,\%2\pi)}{\frac{\pi}{4}} \text{ to obtain the classification.}$$

$(\theta + \pi)$ adjusts the range to $[0, 2\pi]$ which when divided will provide the direction in the range of $[1, 8]$ as desired. However, the directions are offset (see figure 5.2 so the angle needs further adjusting by $\frac{\pi}{8}$ so that the correct classification is obtained.

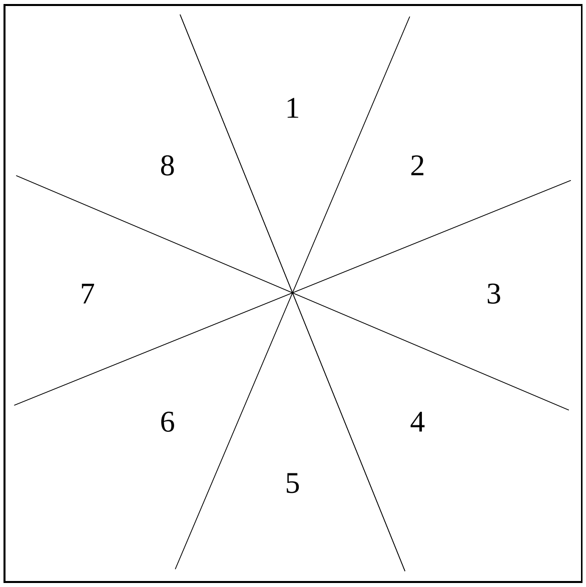


Figure 5.2: Direction classification.

5.4 Object Histories

Once the position (with respect to the equi-temporal region within a composite region of the spatial model) and (qualitative) direction of each object in the current frame has been classified, the *history* for each object can be updated. By history we refer to the sequence of relationships between each object and any other (potentially) interacting objects on each object's course through the domain. Such relationships are modelled qualitatively such that only critical changes are recorded. (Potentially) interacting objects refer to any objects within a "close" vicinity to the reference object such that the typical behaviour pattern exhibited by that object may be affected. As discussed in chapter 4, by "close" we refer not to spatial proximity but to temporal proximity whereby the speed of an object determines which objects are deemed "close".

There are two relationships modelled between "close" objects which are recorded in each history item:

- The relative position of the "close" object with respect to the reference object.

  There are eight possible classifications; ahead, ahead-left, adjacent-left, behind-left, behind, behind-right, adjacent-right and ahead-right (as illustrated in figure 5.3). The relationship model used is similar to the orientation model proposed by Mukerjee & Joe (1990) such that objects in the "lines of travel" are either ahead or behind the reference object. However, the "lines of travel" do not rely on the current trajectory of the reference object, but, rather on the composite region currently occupied by that object, which may include curves rather than just straight lanes.

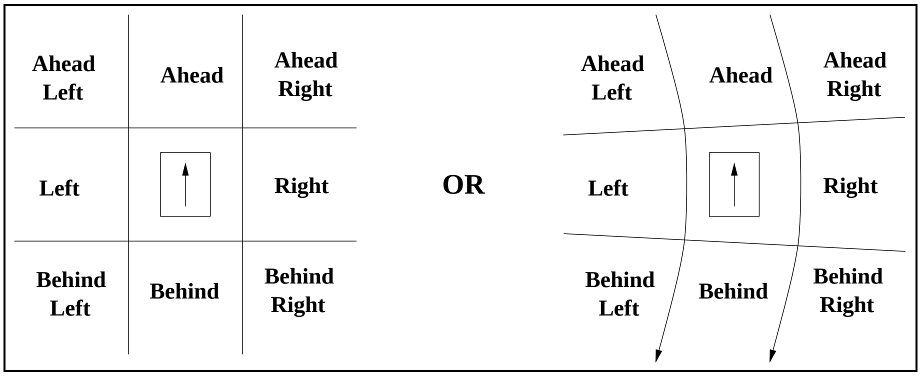


Figure 5.3: Position of "close" objects relative to the reference object.

- The relative direction of movement between the two objects.

  Unlike the relative position, the number of "interesting" relative directions is, at present, limited to four; same, opposing, (perpendicular) towards and (perpendicular) away (as illustrated in figure 5.4).

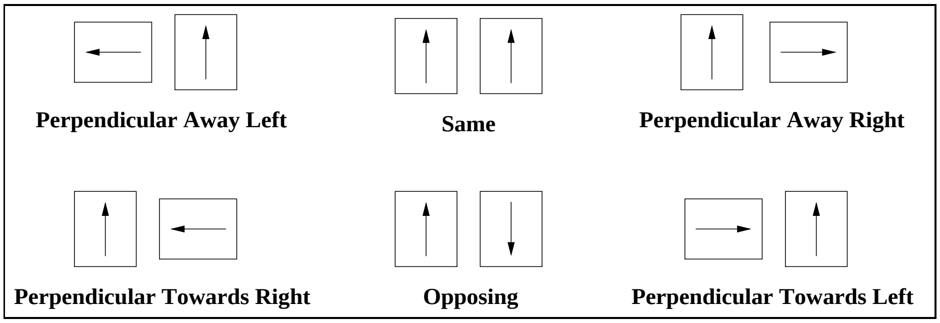


Figure 5.4: Relative direction of motion between two objects.

Before classifying the relationships between the reference object and any other objects it is necessary to identify any objects within a "close" vicinity (i.e. focus the system's attention). This is achieved through an *attention control* mechanism that utilizes the equi-temporal region occupied by an object to build a *temporal extent* within which all objects of potential interest can be identified.

The temporal extent, created by the attention control mechanism, incorporates the equi-temporal region occupied by the reference object and the equi-temporal regions immediately in front and behind the occupied one. Since the two objects may be towards the edge of their respective ETRs, this brings any object sharing the temporal extent within four seconds distance. To identify "close" objects in adjacent paths, the temporal extent is also broadened to encompass those paths (figure 5.5 shows an example of a temporal extent).

On completion, the attention control mechanism is capable of identifying any objects contained within the bounds of the temporal extent. Again, the centroid point of an object is the determining point of occupancy.

Following the identification of "close" objects it is necessary to classify the relative qualitative relationships (position and direction of motion) from the reference object to each identified "close" object. This is accomplished by splitting the temporal extent region (generated by the attention control mechanism) into nine sub-regions[1]. The sub-region that the identified "close" object (partially) occupies determines the relative

[1] Only 8 positions are relevant, the 9th position is the central location occupied by the reference object.

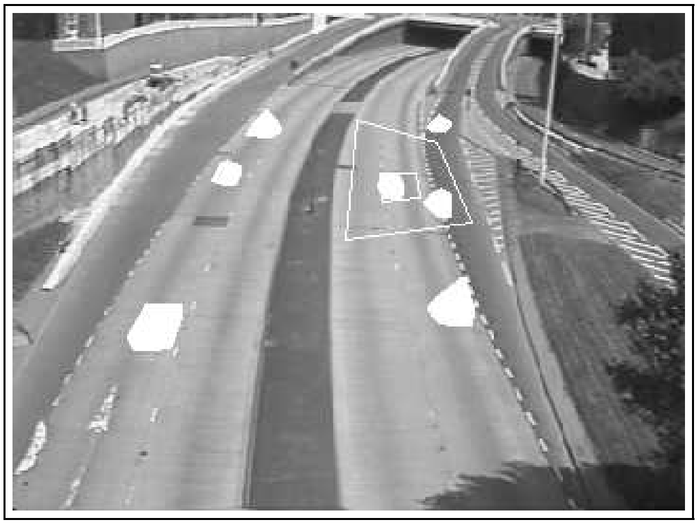


Figure 5.5: Example of a temporal extent generated by the attention control mechanism.

qualitative position (figure 5.3).

Determining the spatial extent of the sub-regions is not particularly difficult. The composite region already splits the temporal extent into three sub-regions. Occupation of these sub-regions would identify objects travelling in the same path, a left adjacent path or a right adjacent path. To obtain the finer grained distinctions required to determine objects travelling ahead, alongside or behind we use the bounding box for the reference object. The front and rear bounds can then be extended across the temporal extent to generate the remaining sub-regions.

Unfortunately, the bounding box is not provided with an object's shape description. However, the end bounds of the equi-temporal region have already been calculated and a line of the same gradient can be used to acquire the end-bounds of the bounding box. Using the equi-temporal region end-bound gradients for the object's bounding box will not necessarily provide a parallelogram — the usual shape used for a bounding box. However, a parallelogram would not take into account camera perspective which would distort the bounding box — the end-bounds obtained through this method are based on empirical evidence which *does* reflect camera perspective. As such, the acquired end-bounds are more appropriate to the situation. Figure 5.6 shows an actual example for the position classification.

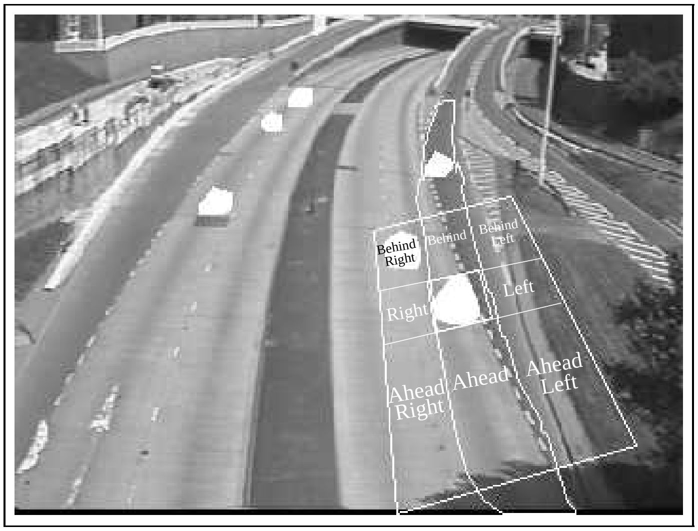


Figure 5.6: Classification of relative qualitative position using sub-regions obtained from temporal extent.

Calculating the relative direction of motion is far more simple. The actual directions of motion for the two interacting objects have already been converted to qualitative values. These two values now need to be compared to determine the relative direction. The absolute difference, obtained by counting the (minimum) number of direction segments, between the two directions is a number between 0 and 4 where the relative direction of motion is:

$$absdiff(dir_{object1}, dir_{object2}) = \begin{cases} 0-1 & \text{travelling in same direction} \\ 2 & \text{travelling adjacent} \\ 3-4 & \text{travelling in opposing directions} \end{cases}$$

The "travelling adjacent" test requires a little more work to determine if the other object is moving towards or away from the reference object. Rather than the absolute difference, the actual difference of an object moving on a perpendicular trajectory will be positive or negative indicating "perpendicular left" or "perpendicular right". Subsequent combination with the relative position will indicate the direction of movement (towards or away). Table 5.1 shows the combination operation to obtain the relative direction of motion for perpendicular moving objects.

	rel. pos. right	rel. pos. left
perp. right	perp. away (right)	perp. towards (right)
perp. left	perp. towards (left)	perp. away (left)

Table 5.1: Combination operation to discern relative direction of motion for objects travelling on perpendicular trajectories.

The relevant object history maintained for the reference object is then updated if it already exists or created if not. The qualitative relationship tuple is compared with the last history item. If the history item matches the relationship tuple then an associated count is incremented to indicate the total number of matches for the current relationship (for statistical analysis purposes). Otherwise the new relationship pair is appended to the history list.

Each reference object may interact with several other objects and a separate object history is maintained for each. The same procedure has to be followed for each reference object — both finding "close" objects and obtaining the relative qualitative relationships. Since each object can be travelling at different velocities, the associated temporal extents will be different. This means that the "close" operation is not necessarily commutative and although one object may be deemed "close" to another, the reverse is not always true. Similarly, even if the objects are deemed "close" in both situations, the resulting relationships may not correspond. Each object has its own frame of reference which may be different to that of the "close" object. Relative position depends on the reference object's frame of reference and if it is different to the other object's then the identified relationships may not correspond.

5.5 Object History Verification

When an object leaves the domain its associated object history lists can be merged with the expanding database. However, to ensure that the object history is valid and free from extraneous relationships caused by tracking "noise", the object history is analysed in a verification procedure that relies on the statistical data provided when generating the history.

First of all, the verification procedure checks whether or not the history has sufficient statistical strength to be considered. If the history sequence refers to a relatively short interaction between two objects, then that interaction could be between elements of tracked “noise” and not actual objects. Since it is not possible (at present) to determine the object types such short interactions are discarded. Over the entire training period, sufficient object histories will be processed such that discarding potentially “risky” histories will not adversely effect the learning process.

Next, the history sequence is analysed to locate (potentially) irrelevant history items. (Potentially) irrelevant refers to those history items that only occur in a sequence of one or two frames and more specifically lie between two matching items that appear for significantly more frames. For example, in the sequence:

- ...
- `((behind, same) 23)`
- `((behind-left, same) 1)`
- `((behind, same) 74)`
- ...

the relationship tuple `(behind-left, same)` only occurs in a single frame and splits a significantly longer sequence of `(behind,same)`. It is important to remember that a single frame takes $(1/25)^{\text{th}}$ of a second which is essentially negligible and this pruning operation only strengthens the re-combined relationship `(behind,same)`.

If the aberrant relationship tuple occurs between two that are not the same the removal process is more complicated. The transition from one relationship tuple to another has to respect the underlying assumption that motion is continuous. This is achieved by checking a *continuity network* (introduced in chapter 2 and also known as a *conceptual neighbourhood* (Cohn 1996)), for the relationship tuples on either side of the aberrant entry. Figures 5.7 and 5.7[2] provide continuity networks for the two relationship types (relative position and relative direction of motion).

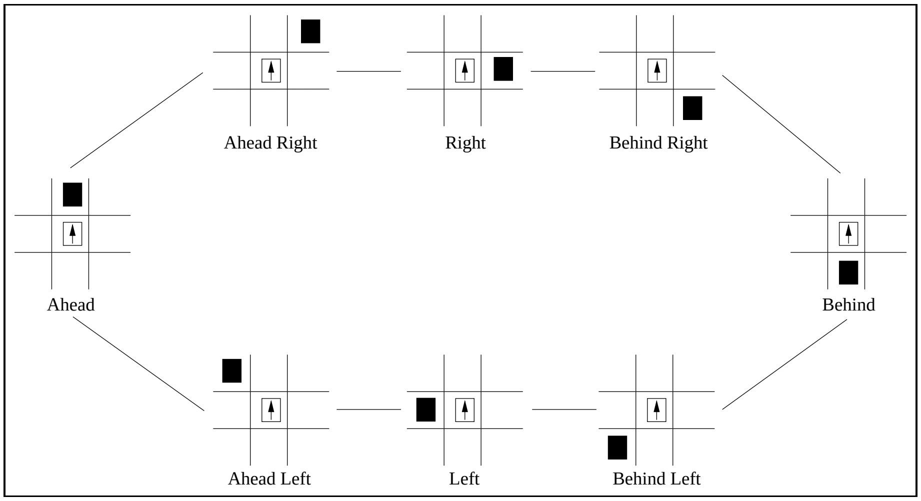


Figure 5.7: Continuity network for qualitative relative position.

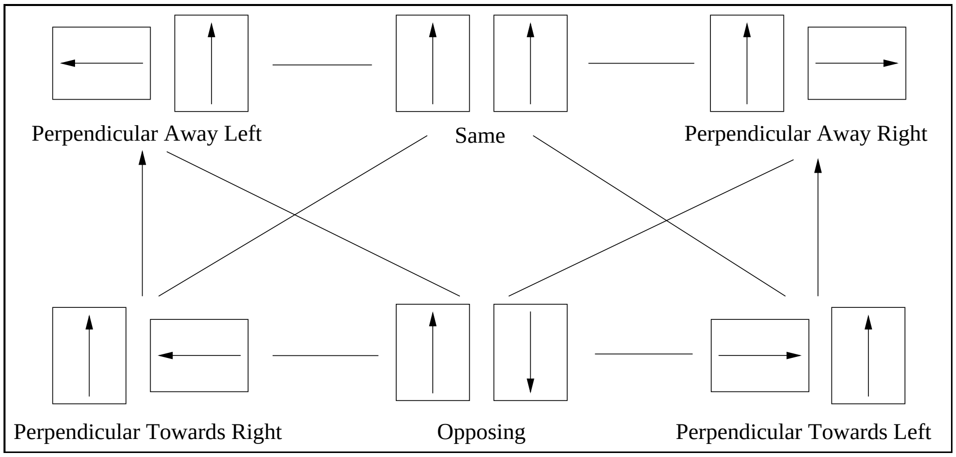


Figure 5.8: Continuity network for qualitative relative direction of motion.

Similarly, the final verification step involves checking that all adjacent items in the history respect the continuity network. In this example:

[2] This continuity network for the relative direction of motion assumes that both objects are moving and not stationary.

```
...
((infront, same) 34)
((right, same) 13)
((infront-right, same) 23)
...
```

there is no direct transition between the positional relationships `infront` and `right`. The continuity network (figure 5.8) shows that the only possible transitions from `infront` are to `infront-left` or `infront-right`. As with Hernández's (1994) topological/orientation model (chapter 2, section 2.2.3), simultaneous changes in both relationships (relative direction and relative direction of motion) may occur (i.e. `(right, opposing)` may change directly to `(behind-right, perp-away-right)`).

For single discrepancies, it may be possible to "fix" the history by inserting a missing relationship tuple or removing an extraneous one (though in general this is not uniquely possible). However, if the number of discrepancies is large, it is easier to discard the entire sequence rather than trying to "fix" the history and then including the conglomerate sequence in the database.

Figure 5.9 provides several examples of actual history sequences and the resulting history sequence after verification. In the first example (a), "Infront Right in Same Direction" only occurs in a single frame and the relationship on either side is identical. As such, that relationship is discarded and the remaining two are merged. Similarly in example (b). The last example (c) is somewhat more complex and results in a history consisting of three items although perhaps just "Behind in the Same Direction" may have been more appropriate. Had the minimum number of frames been set higher that would have been the result.

5.6 Event Database Revision

Following the verification stage, an object history list will represent a sequence of relationship tuples between two interacting objects depicting a single *event* or a composite *event episode.* An event will usually be represented by a single relationship tuple indi-

52: Infront in Same Direction 1: Infront Right in Same Direction 61: Infront in Same Direction	⟹	113: Infront in Same Direction

(a)

16: Behind Right in Same Direction 9: Right in Same Direction 18: Infront Right in Same Direction 1: Infront in Same Direction 17: Infront Right in Same Direction	⟹	16: Behind Right in Same Direction 9: Right in Same Direction 35: Infront Right in Same Direction

(b)

2: Behind in Same Direction 1: Behind in Opposing Direction 15: Behind in Same Direction 1: Behind Right in Same Direction 1: Behind in Same Direction 3: Behind Right in Same Direction 3: Behind in Same Direction 2: Behind Right in Same Direction 2: Behind Left in Same Direction 1: Behind Right in Same Direction 2: Behind Left in Same Direction 1: Behind Right in Same Direction 16: Behind in Same Direction 2: Behind Left in Same Direction 1: Behind in Same Direction 2: Behind Left in Same Direction 2: Behind in Same Direction 1: Behind Left in Same Direction 12: Behind in Same Direction	⟹	16: Behind in Same Direction 3: Behind Right in Same Direction 34: Behind in Same Direction

(c)

Figure 5.9: Example relationship history sequence along with the results from verification. The number refers to the number of adjacent frames in which the relationship tuple was observed.

cating simple behaviour patterns such as *following, being followed, travelling alongside left...*[3] Although a single event may occur through multiple relationships (for example *pulled out behind* would require `(behind, same)` and `(behind-right, same)`) such events usually follow a more simple relationship. In the example of *pulled out behind* the reference object would have been *followed* before the other object pulled out. Thus, the object history represents two events or a (composite) event episode.

This observation is important when updating the database. Not only is it necessary to search for equivalent database entries, it is also necessary to search the database for entries that match a (continuous) subset of the new entry. Such subsets represent simpler event patterns which compose the new event episode. Finally, if an equivalent entry has not been found, the database also needs searching for entries that the new history is a subset of. In this situation, the new object history represents an event pattern that currently hasn't been discovered. However, the new event may already be modelled within one or more composite event sequences.

Using a qualitative representation scheme for the relationships eases the database search. An entry is only equivalent (to the new object history) if the relationship tuple sequences are identical (i.e. each relationship tuple must appear in the matching sequence in the same order). The equivalence test does not include the item count — that was only necessary for the verification procedure. If an exact database entry is discovered a "hit" count is incremented otherwise a new entry is inserted into the database. The "hit" count indicates the number of times that particular sequence of relationship tuples has occurred in the training period and provides statistical information that will allow event models to be constructed.

This "hit" count is the reason why it is necessary to search for matching subsets. The first subset search finds the less complex event sequences that compose the new sequence (i.e. discovers the matching events in an event episode). Although these less complex event sequences have not occurred on their own, they are part of a more complex behaviour pattern that requires these less complex sequences. As such, the "hit" count

[3] The system does not generate these English names.

on those matching subsets is also incremented.

The final subset search, looking for database entries that the new sequence forms a subset of, is only necessary if an equivalent entry is not discovered. This test searches the database for more complex sequences (event episodes) that the new entry contributes towards. Rather than updating the "hit" count on the existing database entries, the "hit" count associated with the new entry is incremented. The search is not necessary if an equivalent entry was initially discovered because this process would have been performed when the entry initially appeared and subsequently updated with the previous search mechanism.

At the end of a training period any sufficiently frequent database entry represents the sequence of relationships in an event model.

5.7 Experimental Results

Over a 15 minute training period observing object interactions on a dual carriageway over 60 distinct relationship sequences were captured in the case-base. Subsequent analysis determined that of those, 25 prove sufficiently frequent to represent events. By far the most observed behaviour was "following" where the only relationships contained in the sequence show the focus object "behind" the interacting object and travelling in the same direction. Unfortunately, the most complex "overtake" sequence, where an object starts behind a second and pulls out and all the way around to finish in front of the other vehicle, was not discovered although the less complex version, where the objects start and finish in adjacent lanes, is modelled as well as other subsets like pulling out behind and pulling in front. It would appear that in this particular domain, the observed area is not large enough to obtain all the necessary information to form the more complex behaviour patterns that we would like to discern.

To demonstrate the effectiveness of the event models discovered in the learning process a demonstration program has been set up that allows the user to specify a particular event to watch for. The event models can be loaded from an "event-info" file along with the spatio-temporal map-file. These are interpreted into the desired format allowing the user to cycle through a list of event models and to decide which event sequence the program should watch out for. Currently a diagrammatic presentation has not been provided. Instead, the user is shown a linguistic description of the composing transitions of the event. For example, an overtake event episode would be described as:

- travelling *behind-right* in the *same* direction.
- travelling *right* in the *same* direction.
- travelling *infront right* in the *same* direction.

To allow the simultaneous interpretation of the observed actions a state transition network (similar to that used by André et al. (1988) as described in chapter 2, section 2.3) is automatically built from the event sequence. Figure 5.10 displays the discovered overtake sequence as a state transition network.

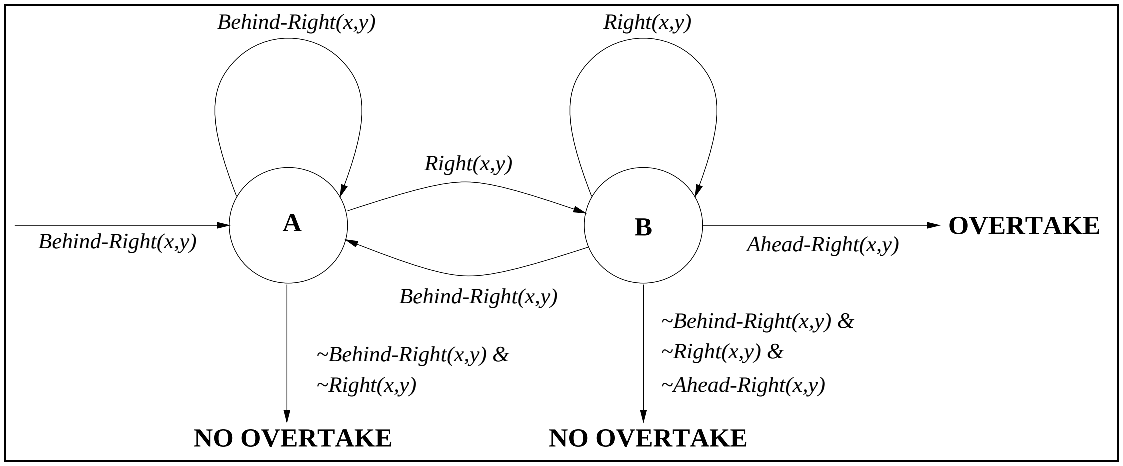


Figure 5.10: Overtake state transition network.

As before, the attention control mechanism isolates objects in the same vicinity which are then categorized with the correct relationship tuple. The relationship tuple is then checked against the starting state in the state transition network. If they match, the event episode has potentially been initiated. To show a potential event episode the relevant objects are coloured on the display. A green object indicates a target object in a relationship and a blue object represents the reference object potentially involved in the event episode. If the last state in the transition network is reached, the reference object shifts to red to indicate that the event episode has been recognized. Figure 5.11 shows a sequence of frames showing the recognition of the overtake sequence of relationships shown above.

Appendix A provides a complete list of the behaviours learnt by the system.

5.8 Discussion

In this chapter, we have demonstrated how, using our spatio-temporal model of space, it is possible to learn event models that are context specific to the domain. From the observation and analysis of object movements and interactions it is possible to generate the relationship history of two interacting objects. One such history constitutes a *case* which can then be added to the expanding database. Further statistical analysis of the database can be conducted to determine which relationship history lists occur sufficiently frequently enough to form event models. We also demonstrate a procedure that is capable of processing object movements and interactions in order to recognize instances of that event.

Using the event recognition procedure, it would be possible to classify *all* instances of occurring events (rather than just a selected one) but an effective means of displaying or conveying all that information is not always possible with such a busy scene. Different colours could be used to describe different events but those colours would have to be selected and the user would have to keep track of what each colour represents to effectively process that information. Also, there is the issue of what colour to use when an object is involved in more than one event. Alternatively, multiple windows could show several different event types being recognized simultaneously.

As mentioned above in the experimental results, the system has not been capable of learning the behaviour patterns associated with more complex event patterns as with the most complicated overtake manoeuvre. This does not occur due to inefficiency in the

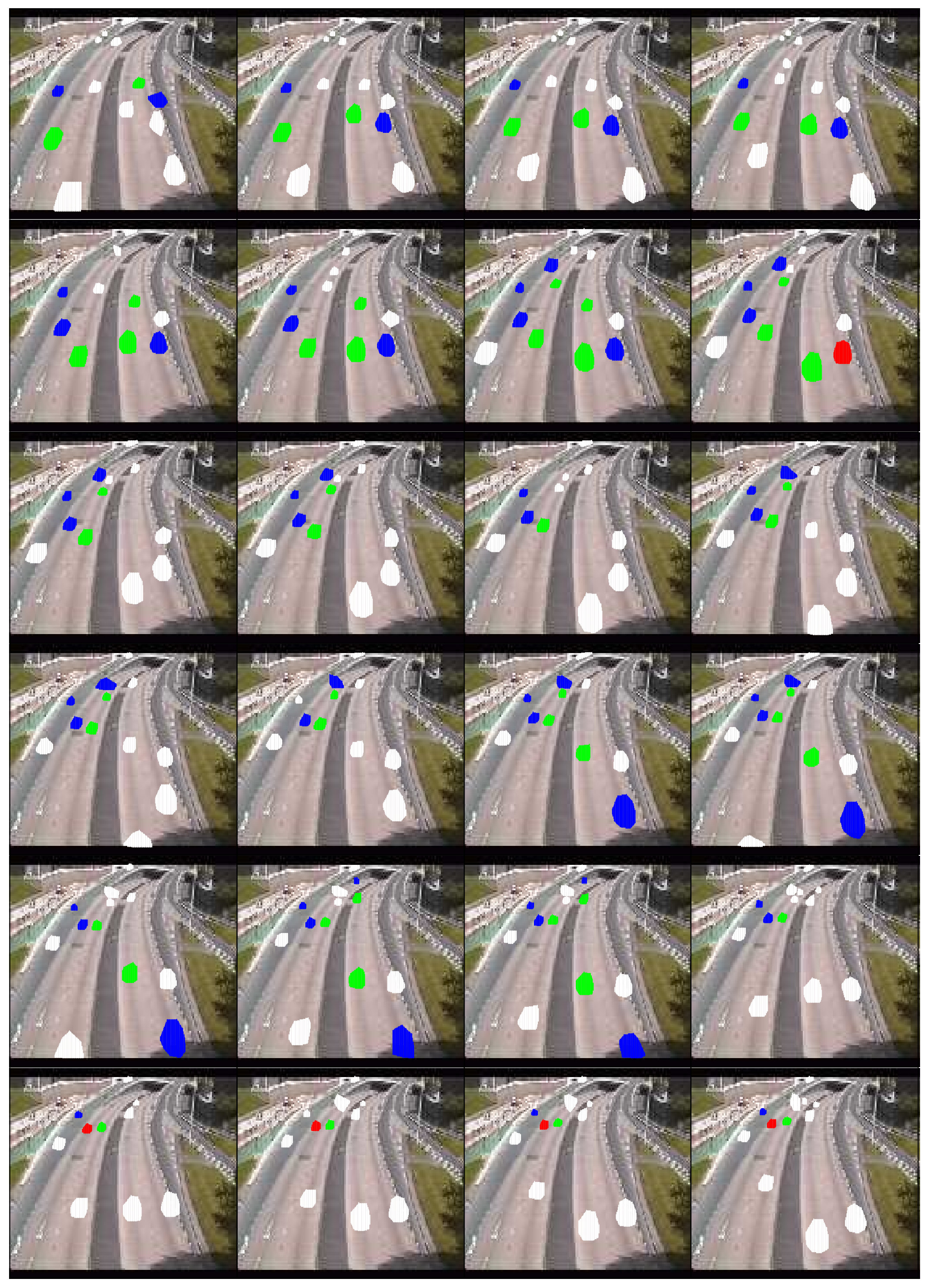


Figure 5.11: Sequence of frames showing the recognition of the overtake manoeuvre.

algorithm but more because the domain is not sufficiently large enough to observe these more complex behaviour patterns. For a more detailed behavioural analysis, the static camera would have to capable of observing a larger area. Alternatively a larger area may be observed with a number of static cameras or with a single moving camera (although the underlying spatial model would have to support this larger area).

5.9 Further Work

- At this time, the events that the system learns are only identified by an interpretation mechanism that provides a pseudo-language description of the relationship changes. Perhaps a better interface would be a diagrammatic representation of these relationship changes or alternatively a natural language description of the sequence of relationships, for example (in the process of) overtaking. Such descriptions can be provided, *a posteriori*, through an interface in which a user is shown the sequence of relationships (either verbally or diagrammatically) and is asked to provide a description of the sequence.

- Currently, the system only models qualitative relationships for relative position and relative direction of motion. As such, the system is not capable of learning any events associated with the velocity of an object, such as accelerating and decelerating. If the relative velocity was also modelled such events could be obtained. Through the analysis of the temporal extents being occupied by each object the relative velocity can be identified as faster, slower or the same which can then provide a further dimension in the relationship tuples.

- So far, the system has only examined relationships between two objects and, as such, only learns events involving two interacting objects. More complex behaviour involving several interacting objects (for example queueing) is currently not modelled as a single event between multiple vehicles. Instead, several events between two vehicles are modelled which does not sufficiently represent the more complex behaviour been observed. The system could benefit by being enhanced to model relationships between three or more interacting objects.

- The attention control mechanism described within this chapter only identifies potentially interacting objects through proximity based on the equi-temporal region occupied by the object under attention. From this mechanism, the event learning strategy relies on the assumption that events occur between “close” objects. Within this domain, this assumption is sufficient. However, in other situations this is not the case. For example, when one vehicle is being chased by a police car the two objects may not be “close” but they are still interacting. Another example occurs between two object travelling at vastly different speeds like a pedestrian crossing a road. Using our example, a pedestrian crossing a road would not be in “close” proximity to moving vehicles, but that pedestrian has examined the environment and decided that an accident will not be caused by crossing the road at that time. How the attention control mechanism can be enhanced to include such situations requires more work.

5.10 Summary

Within this chapter, we have shown how our spatio-temporal representation of space can assist in the identification of interacting objects through an attention control mechanism. From the composite and equi-temporal region occupied by a particular object, the attention control mechanism constructs a “temporal extent” within which all potentially interacting (or “close”) objects can be identified.

We demonstrate how relative direction of motion and relative position can be modelled qualitatively and how we can automatically generate event models through the analysis of objects’ movements and interactions. Throughout the period an object travels through the domain, object relationship histories are created between that object and all potentially interacting objects. Each object history represents a new case (which potentially represents an event model). A history item is added to the database, and at the end of the training period the database is (statistically) analysed to identify actual event models.

Finally, we provide a demonstration which allows the simultaneous recognition of (learned) events by following a state transition graph of each event. A sequence of frames is shown which shows the recognition of an overtake event sequence.

The next (final) chapter provides a summary of all the research discussed in this thesis as well as looking at any future work that may follow this research.

Chapter 6

Conclusions

6.1 Summary of Work

Throughout the course of this thesis we have examined methods of learning for spatial, spatio-temporal and event models which can assist processing tasks in visual surveillance applications. The domains of interest are typically natural outdoor scenes where the movement of objects within the domain are strictly stylized (i.e. domains in which objects tend to comply with a number of default behaviours, like the movement of vehicles on a road which follow rules according to the highway code). Such scenes are observed by a static camera over an extended period to provide training data for the learning processes.

In chapter 3, we demonstrate how a (hierarchical) region based model of space, corresponding to the underlying structure of a domain, can be automatically constructed from the extended observation of objects moving within the domain. Region types include leaf regions, which define the underlying structure of space, and composite regions, which are constructed from concatenation of adjacent leaf regions and describe areas of behavioural significance (such as a road lane or a give-way zone).

We discuss our original approach and the reasons for adopting an alternative method which has proved significantly more effective. Object paths are constructed from the area covered by an object travelling through the domain. These paths are then merged into a database before statistical analysis indicates which entries are (statistically) too infrequent to be included in the spatial model. Regions for the spatial representation are obtained from the combination of the remaining paths stored in the database. Although the spatial model we generate is similar to (and based on) an existing model (Howarth & Buxton 1992*a*), we demonstrate a novel method for automatically learning regions for the spatial representation in contrast to having to provide them by hand.

A temporal extension, to the original spatial model, is outlined in chapter 4 adding a further hierarchical layer. Here, the composite regions are divided into equi-temporal regions where the spatial extent of each sub-division is controlled by the speed objects typically move at. Each sub-division represents the distance moved by an object in a fixed time (we select a two second interval, reasons for which can be found in section 4.1). Dependant on a number of factors (for example: traffic light condition; vehicle load and time of day), objects may travel at different speeds within the same composite region. Should this occur, several ETR sets will be generated corresponding to the different ranges of travel speed. This temporal extension *appears* to be unique within the literature.

Using our spatio-temporal model, it becomes possible to determine a qualitative location for objects within the domain, in terms of spatial location (i.e. leaf and composite region placement) and velocity (i.e. depending on which ETR). This then allows the application of qualitative reasoning methods to real-world situations. Although this is not necessary, qualitative reasoning methods can often simplify complicated situations by considering only the critical information necessary to determine the situation.

To demonstrate how effective the spatio-temporal model, combined with qualitative logics, can be in real world situations we present an effective (qualitative) event learning strategy in chapter 5. In previous approaches which are capable of recognizing situated actions or events in the real world, *a priori* system knowledge is provided in the form of event models. We demonstrate an approach that, through the analysis of interacting objects, is capable of learning sequences of qualitative relationships that define particular events. The spatio-temporal model is used to obtain the qualitative position of each object (in terms of spatial region and equi-temporal region). From the occupied ETR, an attention control mechanism builds a “temporal extent” around that object, within which all potentially interacting objects can be identified. Using the results, an application is available that can watch a domain to recognize instances of a particular event simultaneously.

6.2 Discussion

The learning strategy is similar to case-based learning, although usually the abstraction of prior experience is delayed until that information is actually required. Instead, we combine an iterative conceptual clustering method that allows similar entries to be merged on entry. This strategy maintains a reasonable size for the database and improves the processing efficiency at the termination of the training period.

Training data, in each situation, is provided by an existing tracking application which provides shape descriptions (in the form of a cubic B-spline) for each object moving in a frame. For the duration of the training period, data is provided on a frame-by-frame basis with matching objects in adjacent frames given the same label. Unfortunately, the tracking application available for our use is incapable of handling occlusion (i.e. situations when, due to camera perspective, two objects overlap). In such situation, one of the object labels will be lost. Also, the tracker is not model based, meaning that it is unable to recognize the difference between actual objects moving in the scene and scene variations due to camera vibration or “noise”. As such, the learning applications have to be capable of handling incomplete, inaccurate or “noisy” data. However, results in each area may be improved by utilizing a more sophisticated tracking application

The spatial (and spatio-temporal) model generation process is data driven. As such, an alternative tracking application could be used to provide different results. For example, the tracking application applied throughout this thesis provides two-dimensional shape descriptions. If we were to utilize a three-dimension model based tracker the same region generation methodology would be capable of generating a three dimensional spatial representation[1] (plus a further temporal dimension for the spatio-temporal model). Similarly, the event learning strategy could also be extended to incorporate three-dimensional spatial relationships extending the current scope.

6.3 Future Work

We have already included a section on "Further Work" at the end of each chapter. In this section we will summarize the more important of those aspects as well as looking at the broader area.

- Currently, once generated, the representation of space becomes a static entity. However, the real-world is a changing place and typical behaviours may change over time. For example the typical velocity of vehicles on a road may change depending on the time of day (for example during rush hour traffic tends to be significantly more busy with vehicles travelling slower). Also, new obstacles may be placed within the domain (for example road works) and change the usual object movement patterns. In such situations, the existing representation becomes inadequate. One possibility that can form part of future work would be to extend this learning strategy to be adaptive and to learn new patterns and adjust the existing one. Rather than just learning for a specified training period, the method would have to extend to a continuous learning strategy.

- When generating the spatial model, it would be possible to provide an indication of where objects initially appear on the screen. This information could be combined with a tracking application to reduce the search space for new objects. Also, the expected location of an object in the next frame can be minimized from the spatial model which indicates the typical behaviour exhibited by objects within the domain.

- At present, the event learning strategy models qualitative relationships for relative position and relative direction of movement. Although this can model events in-

[1] Of course, the current application programs would have to be extended to cope with the extra information; the underlying method would remain the same.

directly related to the relative velocity of two vehicles (for example in an overtake manoeuvre) it is not capable of learning events directly related to the velocity (for example pull away from or approach). The event learning strategy could be extended to also model the relative velocity. Through the analysis of the identified ETR, it is relatively simple to determine which object is travelling faster than the other, (approximately) the same speed or slower. This information could enhance the range of event types that the system learns.

- The demonstrated event learning strategy only models relationships between two (interacting) objects. As such, events occurring between three or more vehicles (for example queuing) are only modelled indirectly (from object B following object A, object C following object B and perhaps object C following object A). Such sequences may be important in determining illegal manoeuvres like queue jumping.

- From just a single static camera, the application domain is fairly limited. This could be extended by combining several cameras with (slightly) overlapping views to follow the object movements throughout the entire observed area. If the connection between camera positions is unknown, the system could build the spatial model for each of the views and then combine them into a single area by finding the overlapping spatial model features. This would allow an integrated wide area surveillance system to be constructed as well as improving the event learning strategy that is currently constrained due to the size of observed area.

- From the spatial model construction, it is possible to identify areas with minimal occupation. This could be useful in a shopping centre (or other public area) when considering refurbishment. The spatial model obtained would show typical behaviour patterns that would allow the designers to place new features with the minimal amount of disruption.

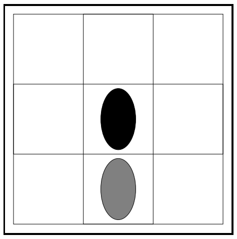
- The ideas presented in this thesis have application in most areas of visual surveillance. For example as a security system in a parking lot, the event system combined with the equi-temporal regions can easily identify unusual behaviour (e.g. a person not following the usual pedestrian paths or spending too long next to a vehicle).

- Qualitative reasoning methods to predict, diagnose and explain physical behaviour in real-world situation in a qualitative manner may be further investigated.

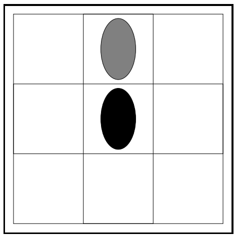
Appendix A

Behavioural Analysis

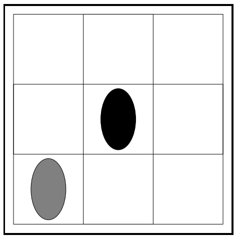
In chapter 5 we discussed our strategy to learn contextually relevent qualitative event models automatically from the extended observation and analysis of object interatactions in a scene showing a dual carriageway. A case-based learning strategy is presented along with (limited) experimental results. In this appendix, we present the complete set of behaviours learned by our system over the 15 minute training period discussed in chapter 5, section 5.7. These are ordered in terms of strength (based on frequency of occurence).



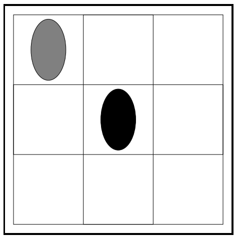
- Travelling *Infront* in the *Same* Direction.



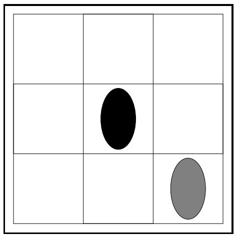
- Travelling *Behind* in the *Same* Direction.



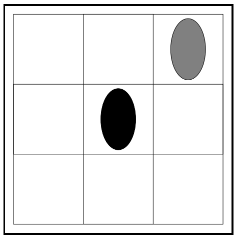
- Travelling *Infront-Right* in the *Same* Direction.



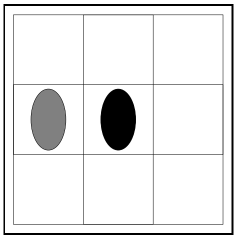
- Travelling *Behind-Right* in the *Same* Direction.



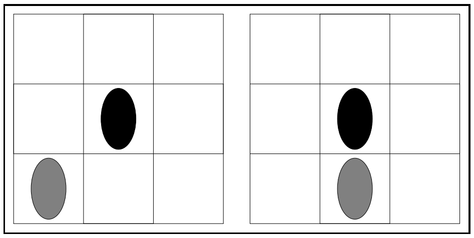
- Travelling *Infront-Left* in the *Same* Direction.



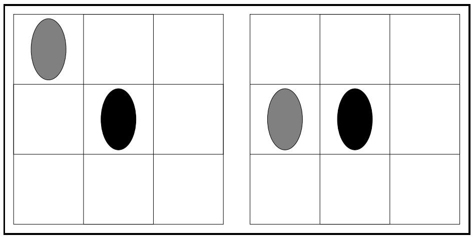
- Travelling *Behind-Left* in the *Same* Direction.



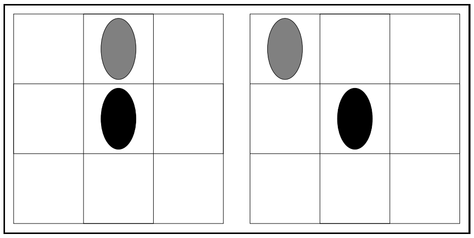
- Travelling *Right* in the *Same* Direction.



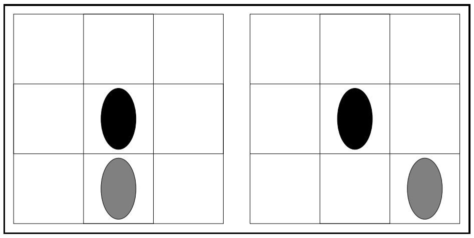
- Travelling *Infront-Right* in the *Same* Direction.
- Travelling *Infront* in the *Same* Direction.



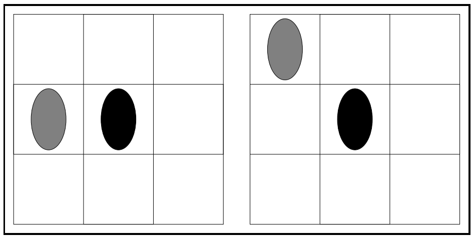
- Travelling *Behind-Right* in the *Same* Direction.
- Travelling *Right* in the *Same* Direction.



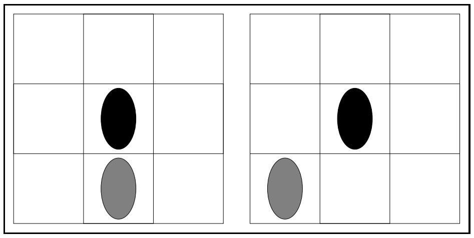
- Travelling *Behind* in the *Same* Direction.
- Travelling *Behind-Right* in the *Same* Direction.



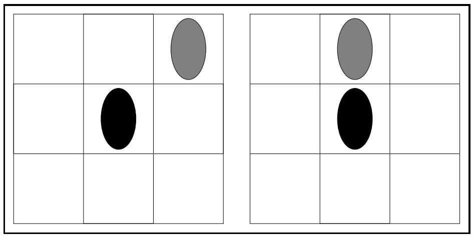
- Travelling *Infront* in the *Same* Direction.
- Travelling *Infront-Left* in the *Same* Direction.



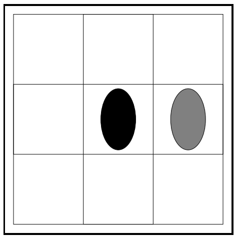
- Travelling *Right* in the *Same* Direction.
- Travelling *Behind-Right* in the *Same* Direction.



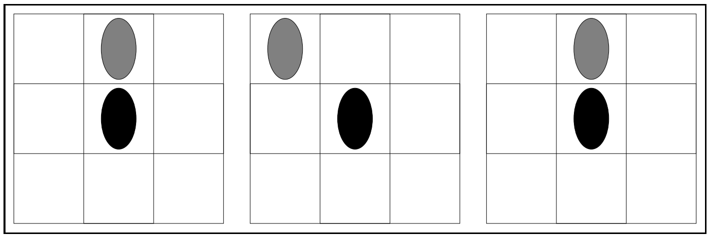
- Travelling *Infront* in the *Same* Direction.
- Travelling *Infront-Right* in the *Same* Direction.



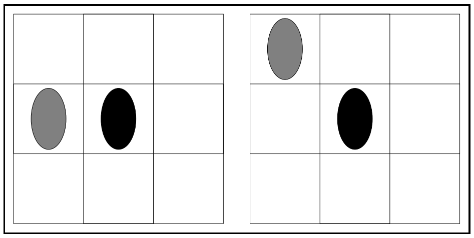
- Travelling *Behind-Left* in the *Same* Direction.
- Travelling *Behind* in the *Same* Direction.



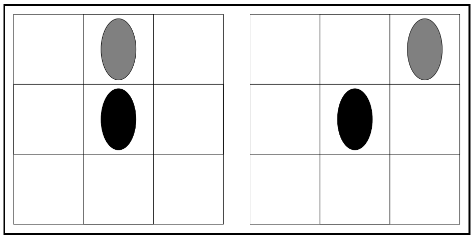
- Travelling *Left* in the *Same* Direction.



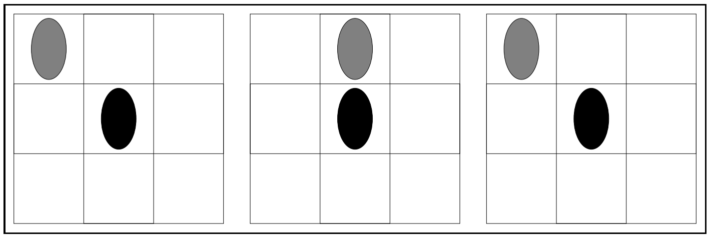
- Travelling *Behind* in the *Same* Direction.
- Travelling *Behind-Right* in the *Same* Direction.
- Travelling *Behind* in the *Same* Direction.



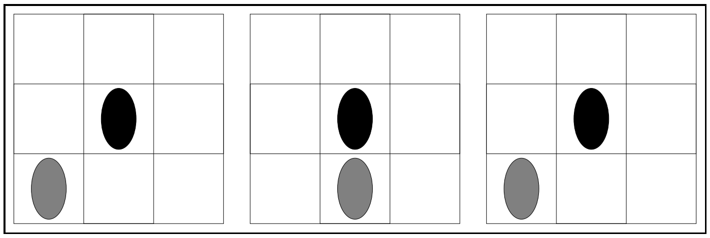
- Travelling *Right* in the *Same* Direction.
- Travelling *Behind-Right* in the *Same* Direction.



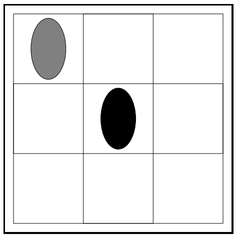
- Travelling *Behind* in the *Same* Direction.
- Travelling *Behind-Left* in the *Same* Direction.



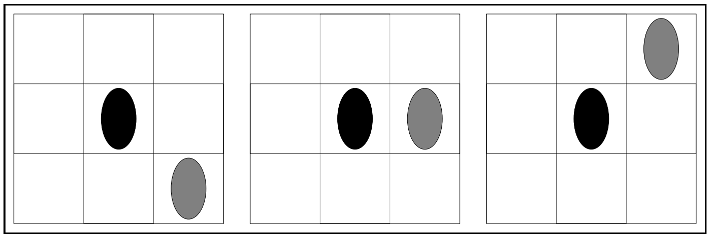
- Travelling *Behind-Right* in the *Same* Direction.
- Travelling *Behind* in the *Same* Direction.
- Travelling *Behind-Right* in the *Same* Direction.



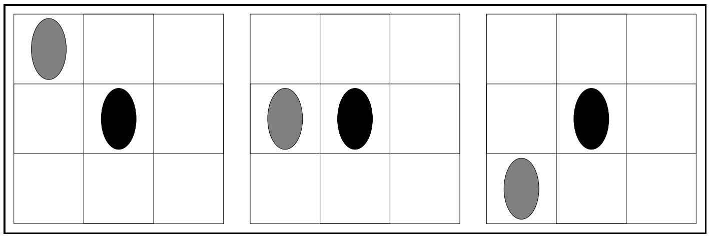
- Travelling *Infront-Right* in the *Same* Direction.
- Travelling *Infront* in the *Same* Direction.
- Travelling *Infront-Right* in the *Same* Direction.



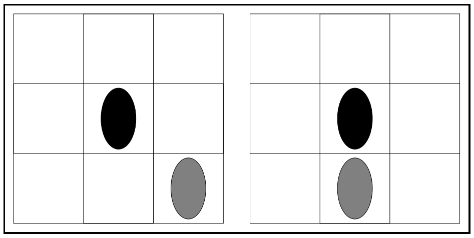
- Travelling *Behind-Right* in the *Same* Direction.



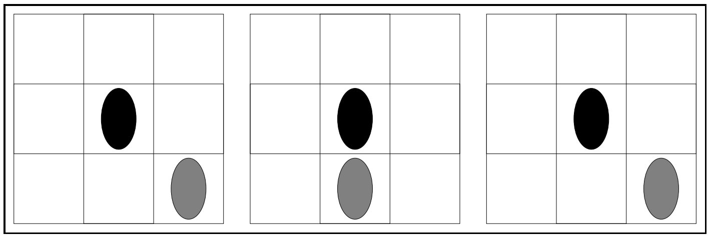
- Travelling *Infront-Left* in the *Same* Direction.
- Travelling *Left* in the *Same* Direction.
- Travelling *Behind-Left* in the *Same* Direction.



- Travelling *Behind-Right* in the *Same* Direction.
- Travelling *Right* in the *Same* Direction.
- Travelling *Infront-Right* in the *Same* Direction.



- Travelling *Infront-Left* in the *Same* Direction.
- Travelling *Infront* in the *Same* Direction.



- Travelling *Infront-Left* in the *Same* Direction.
- Travelling *Infront* in the *Same* Direction.
- Travelling *Infront-Left* in the *Same* Direction.

References

Aamodt, A. (1991), A knowledge-intensive approach to problem solving and sustained learning, PhD thesis, University of Trondheim, Norwegian Institute of Technology.

Aamodt, A. & Plaza, E. (1994), ‘Case-based reasoning: Foundational issues, methodological variations, and system approaches’, *Artificial Intelligence Communications* **7**(1), 39–59.

Agrawal, R. B., Mukerjee, A. & Deb, K. (1995), Modelling of inexact 2-d shapes using real-coded genetic algorithms, *in* ‘Proceedings of the Symposium on Genetic Algorithms’, Dehradum, India, pp. 41–49.

Allen, J. F. (1983), ‘Maintaining knowledge about temporal intervals’, *Communications of the ACM* **26**(11), 832–843.

André, E., Bosch, G., Herzog, G. & Rist, T. (1986), Characterizing trajectories of moving objects using natural language path descriptions, *in* ‘Proceedings of the 7th ECAI’, Vol. 2, Brighton, UK, pp. 1–8.

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.

André, E., Herzog, G. & Rist, T. (1989), Natural language access to visual data: Dealing with space and movement, *in* ‘1st Workshop on Logical Semantics of Time, Space and Movement in Natural Language’, Toulouse, France.

Badler, N. I. (1975), Temporal Scene Analysis: Conceptual Descriptions of Object Movements, Tech. report no. 80, University of Toronto, Toronto, Ontario, Canada.

Bajcsy, R. (1988), 'Active perception', *Proceedings of the IEEE* **76**(8), 996–1005.

Bajcsy, R., Joshi, A., Krotkov, E. & Zwarico, A. (1985), Landscan: A natural language and computer vision system for analyzing aerial images, *in* 'Proceedings of the 9th International Joint Conference on Artificial Intelligence', Los Angeles, CA, pp. 919–921.

Ballard, D. H. (1991), 'Animate vision', *Artificial Intelligence* **48**, 57–86.

Bareiss, R. (1988), PROTOS; a unified approach to concept representation and learning, PhD thesis, University of Texas at Austin, Department of Computer School.

Baumberg, A. & Hogg, D. (1995), An adaptive eigenshape model, *in* Pycock (1995), pp. 87–96.

Baumberg, A. M. (1995), Learning Deformable Models for Tracking Human Motion, PhD thesis, University of Leeds.

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.

Bennett, B. (1994), Spatial reasoning with propositional logics, *in* J. Doyle, E. Sandewall & P. Torasso, eds, 'Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR94)', Morgan Kaufmann, San Francisco, CA.

Blake, A., Curwen, R. & Zisserman, A. (1993), 'A framework for spatio-temporal control in the tracking of visual contours', *International Journal of Computer Vision* .

Boyle, R. D. & Thomas, R. C. (1988), *Computer Vision: A First Course*, Blackwell Scientific Publications.

Brieman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Wadsworth and Brooks, Monteret, California.

Buxton, H. & Gong, S. (1995), 'Visual surveillance in a dynamic and uncertain world', *Artificial Intelligence* **78**, 431–459.

Buxton, H. & Howarth, R. (1995), Spatial and temporal reasoning in the generation of dynamic scene descriptions, *in* R. V. Rodríguez, ed., 'Proceedings on Spatial and Temporal Reasoning', IJCAI-95 Workshop, Montréal, Canada, pp. 107–115.

Castleman, K. R. (1979), *Digital Image Processing*, Prentice Hall.

Clark, P. & Niblett, T. (1989), 'The cn2 induction algorithm', *Machine Learning* **3**, 261–284.

Clarke, B. L. (1981), 'A calculus of individuals based on 'connection'', *Notre Dame Journal of Formal Logic* **23**(3), 204–218.

Clarke, B. L. (1985), 'Individuals and points', *Notre Dame Journal of Formal Logic* **26**(1), 61–75.

Clementini, E. & Di Felice, P. (1996), An algebraic model for spatial objects with undetermined boundaries, *in* P. Burrough & A. M. Frank, eds, 'Proceedings, GISDATA Specialist Meeting on Geographical Entities with Undetermined Boundaries,', Taylor & Francis, pp. 155–169.

Clementini, E., Di Felice, P. & van Oosterom, P. (1993), A small set of formal topological relationships suitable for end-user interaction, *in* D. Abel & B. C. Ooi, eds, 'Third International Symposium on Large Spatial Databases', Lecture Notes in Computer Science No. 692, SSD '93, Springer-Verlag, pp. 277–295.

Cohn, A. G. (1987), 'A more expressive formulation of many sorted logic', *Journal of Automated Reasoning* **3**, 113–200.

Cohn, A. G. (1995), A hierarchcial representation of qualitative shape based on connection and convexity, *in* A. Frank, ed., 'Proc COSIT95', LNCS, Springer Verlag, pp. 311–326.

Cohn, A. G. (1996), Calculi for qualitative spatial reasoning, *in* J. Pfalzgraf, J. Calmet & J. A. Campbell, eds, 'Artificial Intelligence and Symbolic Mathematical Computation', Vol. 1138 of *LNCS*, Springer-Verlag, pp. 124–143.

Cohn, A. G. & Gotts, N. M. (1996), The 'egg-yolk' representation of regions with indeterminate boundaries, *in* P. Burrough & A. M. Frank, eds, 'Proceedings, GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries', Francis Taylor, pp. 171–187.

Cohn, A. G., Gotts, N. M., Randell, D. A., Cui, Z., Bennett, B. & Gooday, J. M. (1995), Exploiting temporal continuity in temporal calculi, *in* R. G. Golledge & M. J. Egenhofer, eds, 'Spatial and Temporal Reasoning in Geographical Information Systems', Elsevier. Extended versions of presented papers. To appear.

Cohn, A. G., Randell, D. A. & Cui, Z. (1995), 'Taxonomies of logically defined qualitative spatial relations', *International Journal of Human-Computer Studies, special issue on Formal Ontology in Comceptual Analysis and Knowledge Representation* **43**(5–6), 831–846. Originally scheduled to appear in a book from a workshop held in Padova in 1993, but for contractual reasons the book never appeared, so this special journal issue of the workshop was produced.

Cootes, T. F. & Taylor, C. J. (1992), Active shape models — 'smart snakes', *in* Hogg & Boyle (1992), pp. 276–285.

Cootes, T. F., Taylor, C. J., Cooper, D. H. & Graham, J. (1992), Training models of shape from sets of examples, *in* Hogg & Boyle (1992), pp. 9–18.

Corrall, D. R. & Hill, A. G. (1992), 'Visual surveillance', *GEC Review* **8**(1), 15–27.

Cui, Z., Cohn, A. G. & Randell, D. A. (1992), Qualitative simulation based on a logic of space and time, *in* 'QR-92', Heriot-Watt University, Scotland.

Davis, E., Cohn, A. G. & Gotts, N. (to appear), Constraint networks of topological relations and convexity. In preparation.

DeJong, G. & Mooney, R. J. (1986), 'Explanation-based learning: An alternative view', *Machine Learning* **1**, 145–176.

Dornheim, C. (1995), Vergleichende analyse topologischer ansaetze des qualitativen raeuml ichen schliessens, Studienarbeit, fachereich informatik, Universitaet Hamburg.

Egenhofer, M. & Franzosa, R. (1991), 'Point-set topological spatial relations', *International Journal of Geographical Information Systems* **5**(2), 161–174.

Egenhofer, M. & Herring, J. (1991), Categorizing binary topological relationships between regions, lines and points in geographic databases, Technical report, Department of Surveying Engineering, University of Maine.

Egenhofer, M. J. & Al-Taha, K. K. (1992), Reasoning about gradual changes of topological relationships, *in* A. U. Frank, I. Campari & U. Formentini, eds, 'Theories and Methods of Spatio-temporal Reasoning in Geographic Space', Vol. 639 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 196–219.

Egenhofer, M. J., Clementini, E. & Di Felice, P. (1994), 'Toplogical relations between regions with holes', *Int. Journal of Geographical Information Systems* **8**(2), 129–144.

Ferryman, J. M., Worrall, A. D., Sullivan, G. D. & Baker, K. D. (1995), A generic deformable model for vehicle recognition, *in* Pycock (1995), pp. 127–136.

Fleck, M. M. (1988*a*), Boundaries and Topological Algorithms, PhD thesis, MIT Artificial Intelligence Laboratory, MIT, MA, USA.

Fleck, M. M. (1988*b*), 'Representing space for practical reasoning', *Image and Vision Computing* **6**(2), 75–86.

Fleck, M. M. (1996), 'The topology of boundaries', *Artificial Intelligence* **80**, 1–27.

Forbus, K. D., Nielsen, P. & Faltings, B. (1991), 'Qualitative spatial reasoning: The clock project', *Artificial Intelligence* **51**, 417–471.

Frank, A. U. & Campari, L., eds (1993), *Spatial Information Theory: A Theoretical Basis for GIS*, Lecture Notes in Computer Science No. 716, COSIT'93, Springer-Verlag, Marciana Marina, Italy.

Frank, A. U. & Kuhn, W., eds (1995), *Spatial Information Theory: A Theoretical Basis for GIS*, Lecture Notes in Computer Science No. 988, COSIT'95, Springer-Verlag, Semmering, Austria.

Freksa, C. (1992*a*), 'Temporal reasoning based on semi-intervals', *Artificial Intelligence* **54**, 199–227.

Freksa, C. (1992*b*), Using orientation information for qualitative spatial reasoning, *in* A. U. Frank, I. Campari & U. Formentini, eds, 'Proc. Int. Conf. on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space', Springer-verlag, Berlin.

Freksa, C. & Zimmermann, K. (1992), On the utilization of spatial structures for cognitively plausible and efficient reasoning, *in* 'Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics'.

Gapp, K.-P. (1994), Basic meanings of spatial relations: computation and evaluation of 3d space, *in* '12th AAAI', pp. 1393–1398.

Gonzalez, R. C. & Wintz, P. (1987), *Digital Image Processing (Second Edition)*, Addison-Wesley.

Gotts, N. M. (1994), How far can we c? defining a 'doughnut' using connection alone, *in* E. S. J Doyle & P. Torasso, eds, 'Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR94)', Morgan Kaufmann, San Francisco.

Hall, E. L. (1979), *Computer Image Processing and Recognition*, Academic Press.

Hernández, D. (1991), Relative representation of spatial knowledge, *in* D. M. Mark & A. U. Frank, eds, 'Cognitive and Linguistic Aspects of Geographic Space', Nato Advanced Studies Institute, Kluwer, Dordrecht, pp. 373–385.

Hernández, D. (1993*a*), Maintaining qualitative spatial knowledge, *in* Frank & Campari (1993), pp. 36–53.

Hernández, D. (1993*b*), Reasoning with qualitative representations: Exploiting the structure of space, *in* 'Proceedings of the III IMACS International Workshop on Qualitative Reasoning and Decision Technology', QUARDET '93, CIMNE, Barcelona, pp. 493–502.

Hernández, D. (1994), *Qualitative Representation of Spatial Knowledge*, Vol. 804 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag.

Hernández, D., Clementini, E. & Di Felice, P. (1995), Qualitative distances, *in* Frank & Kuhn (1995), pp. 45–58.

Herzog, G., Sung, C.-K., André, E., Enkelmann, W., Nagel, H.-H., Rist, T., Wahlster, W. & Zimmermann, G. (1989), Incremental natural language description of dynamic imagery, *in* C. Freksa & W. Brauer, eds, 'Wissensbasierte Systeme. 3. Int. GI-Kongreß', Springer, Berlin, Heidelberg, pp. 153–162.

Herzog, G. & Wazinski, P. (1994), 'VIsual TRAnslator: Linking perceptions and natural language descriptions', *Artificial Intelligence Review* **8**, 175–187.

HMSO (1996), *The Highway Code*, Her Majesty's Stationary Office, HMSO Publications Centre, PO Box 276, London, SW8 5DT.

Hogg, D. & Boyle, R., eds (1992), *Proceedings of the British Machine Vision Conference*, Springer-Verlag, University of Leeds, Leeds.

Hopgood, A. A., Woodcock, N., Hallam, N. J. & Picton, P. D. (1993), 'Interpretating ultrasonic images using rules, algorithms and neural networks', *The European Journal of Non-Destructive Testing* **2**(4), 135–149.

Howarth, R. (1995), 'Interpreting a dynamic and uncertain world: High-level vision', *Artificial Intelligence Review* **9**, 37–63.

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*a*), 'An analogical representation of space and time', *Image and Vision Computing* **10**(7), 467–478.

Howarth, R. J. & Buxton, H. (1992*b*), Analogical representation of spatial events for understanding traffic behaviour, *in* B. Neumann, ed., 'Proceedings of the 10th European Conference on Artificial Intelligence', John Wiley & Sons. Ltd, pp. 785–789.

Howarth, R. J. & Buxton, H. (1993), Selective attention in dynamic vision, *in* 'Proceedings of the Thirteenth IJCAI Conference', pp. 1579–1584.

Iba, G. A. (1989), 'A heuristic approach to the discovery of macro-operators', *Machine Learning* **3**, 285–317.

Johnson, N. & Hogg, D. (1995), Learning the distribution of object trajectories for event recognition, *in* Pycock (1995), pp. 583–592.

Jolliffe, I. T. (1986), *Principal Component Analysis*, Springer-Verlag, New York.

Jungert, E. (1993), Symbolic spatial reasoning on object shapes for qualitative matching, *in* Frank & Campari (1993), pp. 444–462.

Kass, M., Witkin, A. & Terzopoulos, D. (1987), Sakes: Active contour models, *in* 'First International Conference on Computer Vision', pp. 259–268.

Kim, H.-K. (1992), Qualitative kinematics of linkages, *in* B. Faltings & P. Struss, eds, 'Recent Advances in Qualitative Physics', MIT Press, Cambridge, MA, pp. 137–151.

Kohonen, T. (1984), *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.

Kolodner, J. L. (1993), *Case Based Reasoning*, Morgan Kaufmann.

Krishnaiah, P. & Kanal, L., eds (1982), *Classification, Pattern Recognition, and Reduction of Dimensionality*, Vol. 2 of *Handbook of Statistics*, Amsterdam, North Holland.

Kumar, K. & Mukerjee, A. (1987), Temporal event conceptualisation, *in* 'Proceeding of the Tenth IJCAI Conference'.

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

Ligozat, G. (1991), On generalized interval calculii, *in* 'Proceedings AAAI-91', Anaheim, Calafornia, pp. 234–240.

Ligozat, G. (1994), Tractable relations in temporal reasoning: Pre-convex relations, *in* R. Rodríguez, ed., 'Proceedings of ECAI-94 Workshop on Spatial and Temporal Reasoning'.

Ligozat, G. F. (1990), Weak representations of interval calculi, *in* '8th AAAI', pp. 715–720.

Ligozat, G. F. (1993), Qualitative triangulation for spatial reasoning, *in* A. U. Frank & I. Campari, eds, 'Spatial Information Theory: A Theoretical Basis for GIS (Proceedings of COSIT'93)', Springer-Verlag, Berlin, pp. 54–68.

Lowe, D. G. (1991), 'Fitting paramterized three dimensional models to images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 441–450.

Marburger, H., Neumann, B. & Novak, H.-J. (1981), Natural language dialogue about moving objects in an automatically analyzed traffic scene, *in* 'Proceedings of the Seventh IJCAI Conference', Vancouver, pp. 49–51.

Mavrovouniotis, M. & Stephanopoulos, G. (1988), 'Formal order-of-magnitude reasoning in process engineering', *Computers and Chemical Engineering* **12**, 867–881.

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

Meng, H. & Picton, P. D. (1992), Planning collision-free paths in time-varying environments, *in* 'Proceedings 1st International Conference on Intelligent Systems Engineering', Edinburgh, UK, pp. 310–315.

Michalski, R. S. & Chilauski, R. L. (1980), 'Knowledge aquisition by encoding expert rules versus computer induction from examples: a case study involving soybean pathology', *International Journal on Man-Machine Studies* **12**, 63–87.

Michalski, R. S. & Stepp, R. (1980), Learning from observation: Conceptual clustering, *in* R. S. Michalski, J. G. Carbonell & T. M. Mitchell, eds, 'Machine Learning: An

Artificial Intelligence Approach', Vol. 1, Morgan-Kaufmann, San Mateo, California, pp. 331–363.

Minton, S. N. (1985), Selectively generalizing plans for problem solving, *in* 'Proceedings of the Ninth International Joint Conference on Artificial Intelligence', Morgan-Kaufmann, San Mateo, California, pp. 363–391.

Minton, T. M., Carbonell, J. G., Knoblock, C. A., Kuokka, D. R., Etzioni, O. & Gill, Y. (1989), 'Explanation-based learning: A problem solving perspective', *Artificial Intelligence* **40**, 63–118.

Mitchell, T. M. (1977), Version spaces: A candidate elimination approach to rule learning, *in* 'Proceedings of the Fifth International Joint Conference on Artificial Intelligence', San-Mateo, California, pp. 305–310.

Mitchell, T. M., Keller, R. M. & Kedar-Cabelli, S. T. (1986), 'Explanation-based generalization: A unifying view', *Machine Learning* **1**, 47–80.

Mohnhaupt, M. & Neumann, B. (1990), Understanding object motion: Recognition, learning and spatiotemporal reasoning, Technical Report FBI-HH-B-145/90, University of Hamburg.

Mukerjee, A. (1994), Metric-less modeling of one, two and three-dimensional metric spaces, *in* 'Working Notes of the AAAI Workshop on Spatial and Temporal Reasoning', AAAI-94, Seattle, pp. 39–45.

Mukerjee, A. & Joe, G. (1990), A qualitative model for space, *in* 'Proceedings of the Eighth National Conference on Artificial Intelligence', Vol. 2, AAAI Press/MIT Press, Menlo Park, pp. 721–727.

Mukerjee, A. & Schnorrenberg, T. (1991), Hybrid systems: Reasoning across scales in space and time, *in* 'AAAI Symposium on Principles of Hybrid Reasoning', pp. 15–17.

Munkres, J. R. (1984), *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park, CA.

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

Nebel, B. & Bürckert, H.-J. (1994), Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra, *in* 'Proceedings of the Twelfth National Conference on Artificial Intelligence, (AAAI-94)'.

Neumann, B. (1989), Natural language description of time-varying scenes, *in* D. L. Waltz, ed., 'Semantic Structure: Advances in Natural Language Processing', Lawrence Erlbaum Associates, pp. 167–206. Also technical report *FBI-HH-B-105/84* Fachbereich Informatik der Universtät Hamburg, FRG, 1984.

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 Eighth IJCAI Conference', pp. 724–726.

Nielsen, P. (1988), A qualitative approach to mechanical constraint, *in* '7th AAAI', pp. 270–274.

Nilsson, N. (1965), *Learning Machines*, Mcgraw-Hill, New York.

Pentland, A. P. (1986), 'Perceptual organization and the representation of natural form', *Artificial Intelligence* **28**, 293–311.

Pycock, D., ed. (1995), *Proceedings of the 6th British Machine Vision Conference*, BMVA, University of Birmingham, Birmingham.

Raiman, O. (1986), Order of magnitude reasoning, *in* 'AAAI-86: Proceedings of the National Conference on AI', pp. 100–104.

Randell, D. A., Cohn, A. G. & Cui, Z. (1992), Naive topology: Modelling the force pump, *in* P. Struss & B. Faltings, eds, 'Advances in Qualitative Physics', MIT Press, pp. 177–192.

Randell, D. A., Cui, Z. & Cohn, A. G. (1992), A spatial logic based on regions and connection, *in* 'Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning', Morgan Kaufmann, San Mateo, pp. 165–176.

Randell, D. & Cohn, A. (1989), Modelling topological and metrical properties of physical processes, *in* R. Brachman, H. Levesque & R. Reiter, eds, 'Proceedings 1st International Conference on the Principles of Knowledge Representation and Reasoning', Morgan Kaufmann, Los Altos, pp. 55–66.

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.

Rosenblatt, F. (1958), 'The perceptron: a probalistic model of information storage and organization in the brain', *Psychological Review* **65**, 386–408.

Schalkoff, R. J. (1989), *Digital Image Processing and Computer Vision*, John Wiley & Sons Inc.

Schlieder, C. (1993), Representing visible locations for qualitative navigation, *in* N. P. Carreté & M. G. Singh, eds, 'Qualitative Reasoning and Decision Technologies', CIMNE, Barcelona, pp. 523–532.

Schlieder, C. (1995), Reasoning about ordering, *in* Frank & Kuhn (1995), pp. 341–349.

Schlieder, C. (1996), Qualitative shape representation, *in* P. A. Burrough & A. U. Frank, eds, 'Geographic Objects with Indeterminate Boundaries', GISDATA, Taylor & Francis, pp. 123–140.

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

Stanfill, C. W. (1987), Memory-based reasoning applied to english pronunciation, *in* 'Proceedings of the Sizth National Conference on Artificial Intelligence', AAAI Press, Seattle, pp. 577–581.

Sullivan, G. (1994), Model-based vision for traffic scenes using the ground-plane constraint, *in* C. M. Brown & D. Terzopoulos, eds, 'Real-time Computer Vision', Publications of the Newton Institute, Cambridge University Press, Cambridge, pp. 93–116.

Terzopoulos, D. & Szeliski, R. (1992), Tracking with kalman snakes, *in* A. Blake & A. Yuille, eds, 'Active Vision', MIT Press, chapter 1, pp. 3–20.

Tsotsos, J. K. (1981), Temporal event recognition: An application to left ventricular performance assessment, *in* 'Proceedings of the International Joint Conference on Artificial Intelligence 1981', Vancouver, Canada, pp. 900–907.

Tsotsos, J. K., Mylopoulos, J., Covvey, H. D. & Zucker, S. W. (1980), 'A framework for visual motion understanding', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**(6), 563–573.

Ullman, S. (1996), *High Level Vision*, MIT Press.

Vieu, L. (1991), Sémantique des relations spatiales et inférences spatio-temporelles, PhD thesis, Université Paul Sabatier, Toulouse.

Vieu, L. (1993), A logical framework for reasoning about space, *in* A. U. Frank & I. Campari, eds, 'Spatial Information Theory: a Theoretical Basis for GIS', Vol. 716 of *Lecture notes in computer science*, Springer-Verlag, pp. 25–35. Proceedings of COSIT'93, Elba, Italy, September 1993.

Vilain, M., Kautz, H. & van Beek, P. (1990), Constraint propagation algorithms for temporal reasoning: A revised report, *in* M. Kaufmann, ed., 'Readings in Qualitative Reasoning about Physical Systems', San Mateo, CA, pp. 373–381. Revised version of paper that appeared in *Proceedings of AAAI-86*, 377–382.

Weinberg, J. B., Uckun, S., Biswas, G. & Manganaris, S. (1992), Qualitative vector algebra, *in* B. Faltings & P. Struss, eds, 'Recent Advances in Qualitative Physics', MIT Press, Cambridge, MA, pp. 193–208.

Widrow, B. & Hoff, M. E. (1960), 'Adaptive switching circuits', *IRE WESCON* **4**, 96–104.

Worrall, A. D., Marslin, R. F., Sullivan, G. D. & Baker, K. D. (1991), Model-based tracking, *in* P. Mowforth, ed., 'Proceedings of the British Machine Vision Conference', Springer-Verlag, University of Glasgow, Glasgow, pp. 310–318.

Zimmermann, K. (1993), Enhancing qualitative spatial reasoning — combining orientation and distance, *in* Frank & Campari (1993), pp. 69–76.

Zimmermann, K. (1995), Measuring without measures: The delta-calculus, *in* Frank & Kuhn (1995), pp. 59–68.

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.