Target Tracking and Image Interpretation in Natural Open World Scenes

By

Martin K. Teal

A thesis submitted in partial fulfilment of
the requirements governing the award of
the degree of Doctor of Philosophy at
the City University.

City University,
School of Engineering,
Information Engineering Centre,
London.
June 1997.

I hereby grant the powers of discretion to the City University Librarian to allow the thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

____

Martin Kenneth Teal.
(author)

Acknowledgements.

Foremost, I would like to thank my supervisors, firstly Dr T. J. Ellis who introduced me to the world of image processing and has been my main source of continuous support and encouragement throughout this research. I would also like to thank Dr D. Ait-Boudaoud for his help whilst here at Bournemouth University.

I would also like to thank Mr D. G. W. Virgin for his support during the early part of this research at British Aerospace (DYNAMICS) PLC. Several people have contributed to making the daunting task of undertaking this research a more pleasurable experience. I would therefore like to thank Mr D. Knight and Mr J. Roach of Bournemouth University for their support.

I would like to give a very special thanks to my wife Pauline, my daughter Stephanie and finally to my mother all of who have been very understanding and supportive through the most difficult periods of this research.

Contents.

List Of Figures.

List Of Tables.

Abstract.

This thesis is concerned with tracking man made objects moving in natural open world scenes and based on the tracking data, construct a structural representation of that scene, frame by frame. The system developed uses a static camera and a statistical frame differencing technique for detecting motion in an image that has a relatively static background. Objects with a measured temporal consistency are tracked across successive image frames. Based on the tracking data, regions in the scene are associated with particular types of dynamic event. For example regions containing movement (could be roads) and regions where objects seem to disappear or partially disappear (could be hedges).

Because of the sensitivity of the motion estimator to changes in scene illumination and environmental conditions, a tile-based method is used to detect scene motion based on the estimations of statistical variations within the tiles. An updating process is used to ensure that a reliable estimate of the background reference image is maintained by the system. Motion cues are matched against tracked objects from a previous frame using an estimate of the temporal continuity of an object. A spatial-temporal reasoning process is used to infer the structure in the image. This inference mechanism is implemented using a semantic network.

The system has been tested on several open world sequences and in each case has demonstrated that it can identify and track vehicles moving in the scene. Based on the motion of these vehicles regions in the image were identified and scene maps constructed for each scene. The map identified regions where vehicles can be expected to be observed moving and regions where they could become occluded.

A CD-ROM is included with this thesis that contains the results obtained by the system for the two image sequences used in chapter seven. These results incorporate some of the enhancements outlined in chapter 8, section 8.3. A windows movie player is included on the CD-ROM and appendix d provides information on the contents of the CD-ROM together with installation and operating instructions.

Chapter 1 Introduction.

1.1 Background.

There are many civilian and military applications where it is important to identify and track man made objects moving in natural open world scenes. In a military situation for example the need is to identify and track potential threats with the possibility of an engagement with that threat. One of the problems faced in this type of situation is that the object to be engaged can become occluded just prior to the engagement, this means that time is lost whilst trying to re-acquire the target, with potentially very serious consequences. The military environment has a variety of systems in use that acquire, identify and track targets ready for an engagement. The target being engaged will quite naturally be trying to manoeuvre (man-made non predictable manoeuvres) and employ countermeasures such as smoke or flares etc, to avoid being engaged or break the engagement.

The current systems in use do have serious problems if the target is lost during the engagement as corrective actions are taken only after the target is lost. The problem of a target moving behind another object in the scene, such as a building (the target is still in the field of view, but not visible to the camera) is not addressed at all by current weapon systems and the consequences of the loss of a target while in an engagement situation are self evident. Target identification, tracking and engagement are all concerned with the so called 'sharp end of the stick', there are other military applications where being able to track objects even if they have become occluded would be required.

Battlefield management and surveillance is a prime example of where we would also need to be able to identify and track objects, particularly if the objects being tracked have become occluded by other objects in the scene. The ability of a system to still be able to track that object after it had become occluded would prove extremely useful in initial and mid-course defence encounter scenarios. Though military applications continue, the current political situation has reduced the need to develop military systems and this shift has fuelled the development of civilian applications.

Civilian applications have become more relevant in the 1980's-90's and the trend suggests that applications such as crowd monitoring and security will be major areas for machine vision as we head into the next century. With security and surveillance systems we would be more concerned with being able to track man-made objects moving around perimeter fences, particularly if the object being tracked moved behind another object in the scene, such as a hedge, becoming occluded from the system but still in its field of view. The tracking system must still be able to track the object in the scene, not from a point of view of engagement, but more concerned with the fact that the object is still in the scene and although it is not visible, generate some form of condition (automatic alarm) to the operator.

The understanding of crowd behaviour in semi-confined spaces is a very important part of the design of new pedestrian facilities (Davis, [49]). Closed circuit television systems support the data collection and monitoring of crowds, with the potential for expansion to security applications where such systems could track individuals moving in a crowd providing extra support to security staff in the prevention and solving of crime.

With the growth of crime and the increasing problem of congestion in our pedestrian areas, the need to maintain higher and tighter security has led to a major growth area for the application of machine vision. There would therefore be a clear need to develop machine vision systems that are capable of spotting crowd problems or detecting and tracking man made objects moving by security fences and automatically triggering alarm conditions.

The above applications are all concerned with identifying and tracking objects moving in open world scenes. Systems have been developed for the identification and tracking of man made objects moving in open world scenes. Feature based geometric model matching, (Tan [1], Worral [2]) has been shown to be successful in traffic management situations where vehicles have been identified and tracked in a road traffic scene, primarily traffic roundabouts and in airport scenes where aircraft service and support vehicles have been identified and tracked as they moved in and around parked aircraft on a runway.

The vehicles in the traffic roundabout scene do however occupy a significant proportion of the image and even in the airport situation, the vehicles were less than 200 meters from the camera. These model based methods tend to be less successful when the object to be identified and tracked is further away from the camera and occupying a smaller proportion of the image pixels. In this research the objects to be identified and tracked are expected to be large distances from the camera and consequently will only occupy a small number of image pixels. In this case, it has been found that the matching of crude object descriptors is more robust, (Rosin [3], Teal [4]).

If the objects to be tracked are expected to be further away from the camera, then more of the image will be occupied by the background objects in the scene. We cannot consider that these objects will be static: trees, bushes, even the grass can have apparent motion when the wind blows. The wide angled view of the scene compounds the apparent motion problem further as illumination changes can also be perceived as motion.

To the problems of wind and illumination changes can be added further weather conditions such as rain or snow. If we consider an image processing system that is analysing the motion of vehicles moving within an open world scene, then it can be clearly seen that these environmental conditions make it a complex image to analyse.

If we consider tracking, then the system must resolve the ‘correspondence problem’. This is where to track an object moving in a scene the tracking system must first recognise the object to be tracked in say frame$_n$ by extracting a set of object descriptors (features) that describe the object. At a later time say frame$_{n+1}$ it must extract those object features again and resolve that the two sets of object features are in fact from the same object viewed at two different times. The environmental problems outlined above complicate this process as they can generated a large amount of apparent motion (clutter) which the system must reject.

The task of tracking however is further complicated by the fact that the objects to be tracked will have changing pose with respect to the camera and the wide range of distances between object and camera will give many possible interpretations to the identity of the object. The objects as they are tracked may become partially or fully occluded by other tracked objects or by static objects in the scene. These factors are generally beyond control but complicate both the recognition and frame to frame matching processes. The tracking of objects that are no longer visible in the scene (they have become fully occluded) but are still in the field of view of the camera has received little attention by current machine vision systems.

Clearly to overcome the problems outlined a new approach is required. The system must learn over many frames areas where certain motion events occur and based on this learning process build some form of symbolic map representation of that scene that associates areas of the image with motion events and other areas of the image with occlusion. The constructed map could then be used to focus the image processing system to regions of the image that are expected to contain object motion. The consequence of this is that any motion detected in those areas of the map would increase our belief that this is a region associated with object motion but also that the current detected motion is an object of interest. The areas of the image identified with potential occluding objects could be used to predict potential object occlusion during tracking.

If the tracked object fails to be re-located in the image (it does not re-appear), the fact that the tracked object had become occluded in an identified region of the map could be used as the basis for continued tracking of that object.

1.2 Aims and objectives.

1.2.1 Aims.

The first aim of this research was to investigate the problems associated with tracking man made objects moving in open world scenes, where the tracked objects are expected to be a large distance from the camera.

In a security application we would want to be able to continue tracking the object even if it became occluded (went behind a large bush for instance). The importance of the system still being able to detect and track that object has already been outlined. For the machine vision system to be able to continue tracking an object when it has become occluded, it would require the system to have an interpretation of structural features within the scene.

This led to the second aim of this research, to analyse and interpret structural features in a scene based on the motion of objects moving within the scene. From this detected motion, to build a symbolic representation (a map) on a frame by frame basis. These two aims led to three objectives.

1.2.2 Objectives.

1: To develop a new algorithm for tracking man made objects moving in natural open world scenes, where the object to be tracked is expected to be a large distance from the camera.

2: To develop a spatial-temporal reasoning algorithm that builds a structural representation (a map) of a scene on a frame by frame basis, based on the object motion within the scene. The map giving a level of confidence in the structural interpretation of that scene.

3: To integrate these two algorithms into a complete system.

1.3 Organisation of the thesis.

This thesis has been organised into 8 chapters. Each chapter is largely self contained with an introduction and overview, analysis, results and finally a summary and discussion. The first chapter introduces some of the reasons why we would want to track man made objects in natural images, together with the problems faced by systems developed to accomplish this task. The second chapter reviews the literature on identification and tracking of man-made objects, it also examines some of the issues concerned with the analysis and interpretation of structure in natural open world scenes.

The third chapter gives an overview of the image processing system that has been developed by this research. It shows how the main processing task of the system has been functionally decomposed into three image processing functions. The fourth chapter looks at problems concerned with the acquisition of images and the generation of motion cues representing objects moving within a scene.

Chapter five addresses the problem of tracking and develops a new algorithm for tracking man made objects moving in a scene where the tracked objects are expected to be a large distance from the camera. The sixth chapter looks at building a symbolic map using spatial-temporal data of objects moving in the scene. It develops a new algorithm that interprets structural features within the scene. It builds a map of that scene on a frame by frame basis, identifying regions in the image where tracked objects can be expected to be observed moving and regions where objects can become occluded.

The seventh chapter addresses the integration of the tracking and spatial-temporal reasoning processes into a complete system. It shows that the system is capable of identifying and tracking man-made objects moving within open world scenes, constructing a map of that scene based on the tracked object motion. It discusses the results obtained by the application of the algorithms to real world scenes. The final chapter rounds up the thesis with conclusions and outlines future work.

1.4 Discussion and Summary.

In this chapter we have identified the reasons for undertaking this research and highlighted areas and applications of machine vision where the identification and tracking of man made objects moving in natural open world scenes would be necessary. Perhaps more importantly this chapter has identified that any system engaged in this task faces considerable problems.

Occlusion, illumination and environmental effects all complicate the frame to frame matching process needed to describe the spatial-temporal behaviour of the objects moving within a image sequence.

Chapter 2 Identification, Tracking and Structure (A Review).

2.1 Introduction.

In computer vision we are concerned with the transformation of information. Generally that information is an image or sequence of images consisting of thousands of picture elements that are to be transformed into a concise symbolic description of the image which can be used by a viewer or computer system and is not cluttered with irrelevant information. This process can be divided into a series of levels (low, intermediate and high) giving a range of image representations. Low-level vision involves operations directly on the picture elements, intermediate to high-level vision uses the information obtained from the lower-level image processing to build more useful descriptions of the image and the world that is viewed.

The low-level processing is essentially general purpose, applying image processing algorithms direct to the pixels and usually does not make use of domain specific knowledge. High-level vision however is concerned with finding a consistent interpretation of the features found by the lower-level processing and is therefore based on recognition, i.e. the matching of internal representations of the world with the image data obtained from the sensors. Depending on how the vision systems internal representation of the outside world is organised and how the recognition mechanism is implemented, the vision system can be classified into one of two main categories, namely

1: model-based.
2: knowledge based.

In a model-based vision system the internal representation of data is based on geometric models and the mechanism for recognition consists of matching these models with two dimensional or three dimensional descriptions obtained from the image. Model-based vision systems must have a robust feature extraction mechanism to obtain reliable image data for the generation of the object descriptions that will be used in the matching and recognition processes.

In knowledge-based systems the representation of an object is symbolic, and includes information about the objects and their relationships to one another. The recognition process does not require an explicit model of the object, this process is realised by the inference of the known data and the known facts about the domain. Feature extraction is performed on the image to construct a symbolic description of the objects that could be in the scene. The knowledge about objects that could be in the scene and their relationships to one another is generated before the recognition process begins. The extracted image features are used in combination with acquired knowledge to deduce the location of relevant objects; however, our knowledge about the relationships between the image features (evidence) and the objects in the world (hypothesis) is often uncertain, leading to the misclassification of objects found in the image.

Many ad-hoc systems have been developed that use techniques from both model-based and knowledge based systems to solve the recognition problem, but generally they tend to be domain specific.

The majority of computer vision literature concerns itself with the interpretation and analysis of constrained images, the so called widget, where the environment is well controlled, for example an industrial inspection environment. In contrast to this there are many applications where we are concerned with the interpretation of the more difficult open world scene such as that depicted by figure 2.1 on the next page. In this image for example we may be concerned with identifying and tracking the van as it moves in the image. Alternatively, we may be concerned with interpreting structural features in the scene, such as the roads or buildings.

The interpretation of these images introduces many more complexities into the analysis, namely:

(i) uncontrolled and variable light conditions.
(ii) different object scales.
(iii) orientation of the object with respect to the camera.
(iv) partial or full occlusion of the object.

These additional features further complicate the recognition and tracking

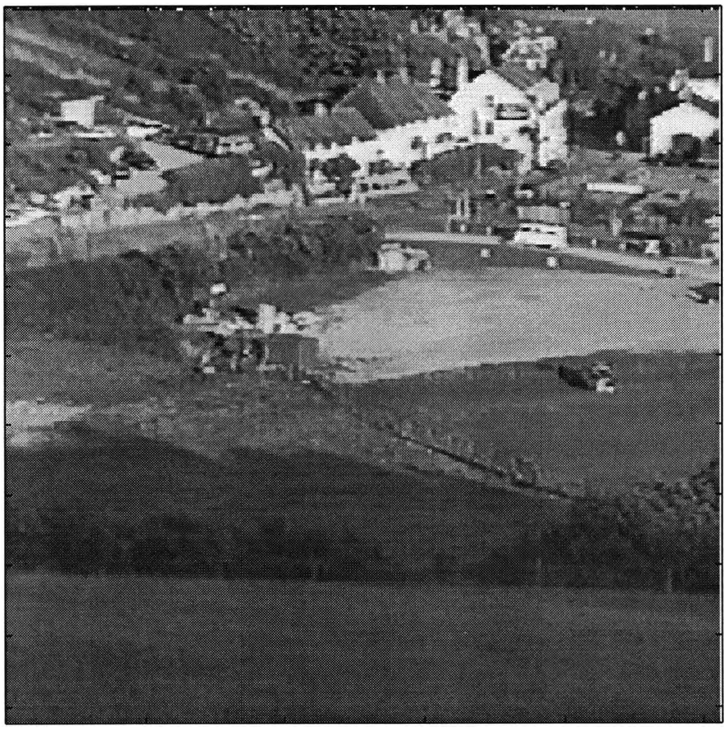


Figure 2.1 Natural Open World Scene.

2.2 Model Based Identification.

Object identification in natural scenes relies on the segmentation of an image into a number of self contained regions where each region represents a particular object in the known world. There has been a considerable amount of research effort put into the identification of vehicles moving in open world scenes, not only for military applications but also civilian applications, particularly traffic monitoring and control. A major objective of the MMI 007 Alvey consortium was the integration of data driven image processing methods and goal driven methods to develop a generalised vision system architecture capable of identifying vehicles in natural open world scenes.

The data driven approach uses the measurement of local attributes in the image to identify features which characterise objects that may be in that scene. Conversely, the goal driven approach uses prior expectations, often in the form of explicit models to guide the analysis of the image. 'The knowledge based approach' (Baker & Sullivan, [5]) presented the philosophy of the MMI 007 exemplar which identified and discussed the major issues faced.

One of the important points highlighted was the fact that the vision system was not expected to operate on a single hypothesis generation and evaluation cycle, but rather expected a reasoning strategy to generate many sub hypotheses and evaluate them using evidence drawn from different sources of information, combined with knowledge of the structure of objects in the scene. This would of course require a large number of intermediate hypotheses to be generated as the understanding of the scene evolves. However a first stage hypotheses could be inferred from the results of an initial low-level segmentation process together with the reasoning processes that labelled the segmented areas.

This initial hypothesis generation would provide a bounded set of 2D image data, though only imprecise evidence for the existence of an object, it would strongly constrain the search space for more computationally expensive verification algorithms. Having generated these hypotheses, it is necessary to perform a quantitative evaluation of the presence of the object in the 2D image data. There are two major factors affecting this task, namely the number of different viewpoints of each object and the possible number of different objects. Together they form a vast search space which must be made manageable so as to limit the computational costs of performing the actual search. Several methods can be employed to limit the search space, but if a general scene understanding system is to be developed, then clearly these restrictions would have to be removed.

Limiting the search space to say for example a single object and constraining the position of occurrence in the image to certain defined regions assists in reducing the total search space. These constraints however do not find the exact location and orientation of the object with respect to the viewer. Establishing the correspondence between 2D image features and 3D object components is a major problem in model based vision systems. The solution of the non-linear spatial correspondence problem being one of the major impediments to the application of model-based methods for vision systems.

Psychological evidence has suggested that the human vision system relies on the perceptual organisation of objects for the interpretation of a scene, (e.g. Kubovy and Poerantz, [6]). Image features can be used as cues in an automatic recognition process (Lowe, [7],[8]), where the discovery of a specific feature group can be used to search for related features according to the structure contained in a model. This would progressively constrain the search each time new evidence contributes to the scene interpretation. Lowe's work however used images which contained multiple instances of well defined groups of edge segments, which have a low probability of accidental occurrence.

The open world scene does not tend to produce this form of simple edge grouping, with the edge data in these scenes being viewpoint determined. It is more likely that Lowe's work would need to consider object specific features in addition to or in support of the extracted edge data. The use of knowledge based systems to perform the identification of vehicles in a natural scene requires a reasoning strategy to guide the identification and verification process. 'The development of reasoning strategies', (Baker & Sullivan, [9]) looked at the concept of reasoning strategies that would be the pre-requisite to specifying the control mechanisms needed to guide the automatic recognition of man made vehicles.

The reasoning strategy used a common technique in vision systems in that the recognition process begins with the detection of a cue, (some feature or cluster of features that are thought to have some perceptual significance). This can be especially significant for cues that trigger a series of reasoning processes each of which is seeking more evidence to support or disprove the evolving hypothesis. The combination of cues and reasoning processes is usually known as the reasoning strategy and this work highlights the fact that any reasoning strategy developed is likely to have a limited application and that a general vision system reasoning strategy would probably require many different strategies.

This is partly due to the fact that in an open country scene parallel lines and coloured image segments would provide strong cues for the identification of a vehicle, but in an urban scene these cues would produce much poorer performance. The main problems arise from the fact that it is fundamentally difficult to devise robust algorithms for the initial segmentation and feature extraction in an image of a natural open world scene. For a vision system, the identification of characteristics in the image data which could be used to initialise the high-level structures and thus initiate a reasoning process is a particularly difficult problem to solve. As part of the MMI-007 consortium, Godden et al [10] looked at the problem of image segmentation and attribute generation, (a major problem area highlighted by Baker & Sullivan [9]).

The nature of the images used were found not to lend themselves easily to derivation of precise 3D structural information by either optical flow or stereo techniques and they used static segmentation techniques to capture information present within the image. The scheme used a bottom up strategy to provide image description and an initial set of hypotheses to bootstrap top down processes. Surface homogeneity, texture homogeneity, colour, boundary smoothness and continuity were used to provide segmentation and region information. The algorithms demonstrated that a vehicle could be segmented in an image and that attributes generated for that segmented region used to identify a vehicle. The types of algorithms developed were region and edge based, running very loosely coupled. It was highlighted that better performance could be achieved if the results of these two techniques were more closely coupled.

Sullivan [11] highlighted that part of the motivation for using a knowledge based approach is due to the fact that with natural scenes it is very difficult to derive a 3D description of the scene as large areas of these images tended to consist of groups of objects such as trees, bushes, roads and buildings which generally have no easy 3D description. The paper linked together the work carried out by Godden, [10] who used attributes of segmented regions to classify major areas of the image, which provided initial cues to the presence of a vehicle, with the work carried out by Brisden [12]. This used detailed knowledge of the 3D geometry of a vehicle expressed as an explicit model to make the decision as to the existence, position and type of vehicle in the image.

This model defined the exact relationship between the object features of the vehicle, specifying the features that are present in a vehicle in the image under any viewing condition. This allows a given instance of the model to be evaluated with great precision if the view point is know. The hypothesis generation uses low-level scene description to identify potential regions and provide loose bounds on the position of the vehicle in the image together with its distance and orientation with respect to the camera. These bounds establish constraints on the eventual matching between the object model and the image, but there is still a great deal of uncertainty.

Two more methods are now used in support of one another to reduce the level of uncertainty in the model matching process. The first method uses geometric reasoning, which is based on assumed correspondences between object features and key features found in the image. The second method is to limit the search applied to small subspaces of the view transform. These methods help to reduce the computational burden of testing and model matching, but it can still be prohibitively high and relies on partial hypothesis to concentrate the search.

However a further constraint can be introduced from prior knowledge of the camera position with respect to the ground plane and the knowledge that the vehicle will be in an upright position. These additional constraints leave only one degree of freedom. The approach taken adopted a hypothesis and test strategy, where different types of knowledge constrain the hypothesis generation stage. Scene knowledge and groupings of features initialised the detailed analysis of candidate areas using specific knowledge of the geometry of the car. Attention is drawn to a plausible hypothesis which is progressively refined to the point where a specific evaluation could be carried out. This means that there would be potentially many erroneous matches between irrelevant features and the model due to low-level features being mislocated or perhaps view dependent boundaries. These features must be rejected by the initial stage of the scene analysis. The paper highlighted the fact that the identification strategy could extract a vehicle from an image since the strategy bypassed the hardest problem for a vision system; that of accessing the appropriate object specific knowledge from the very large host of possible interpretations of the image.

The geometric modelling so far discussed has concentrated on how a 3D model of the object is matched to extracted features in the image. An alternative to this method is to extract features from the image space and transform them into a parameter space and identify features in the parameter space.

'Vehicle detection in open world scenes using a Hough transform technique' , (Radford [13]), looks for arc/circle features in parameter space which could correspond to features in the image space such as wheels and wheel-arches. Simple spatial operators are used to extract edge data in the image and identify straight line segments with the edge map. A Sobel operator is first applied to the image and the resulting edge image thinned to single lines by a process of non-maximal gradient suppression. Noise in the edge image is further reduced by suppressing edgels that turn more than 90 degrees with respect to the neighbours of the edge pixel. A hysteresis technique similar to Cannys (Canny [58]) is applied to the edge gradient and the edge strength is thresholded between an upper and lower bound via an 8 neighbourhood connectivity method. A fractal line discriminator is applied to the thresholded edge image and a roughness factor 'R' calculated by applying a mask to the image in a raster scan order and calculating the change in edgel direction $\Delta\theta$ between each pixel in the mask and its four neighbours. Radford argued that natural features in an image would give rise to rough edges, but man made features are more likely to have straighter edges. The roughness measure R is calculate from

$$R = \frac{\sum \Delta\theta}{n} \qquad (2.1)$$

where

n is the total number of edgels in the connected line.

$\Delta\theta$ is the change in edge direction between the centre pixel and its four neighbours as defined by the mask M.

R is thresholded to retain lines which are either above the threshold and therefore would give rise to natural edges (trees, bushes, etc) or below the threshold which would give rise to man made edges. The resulting straight line segments only are transformed into a 3D parameter space (a, b, r) using a standard Hough Transform defined by:

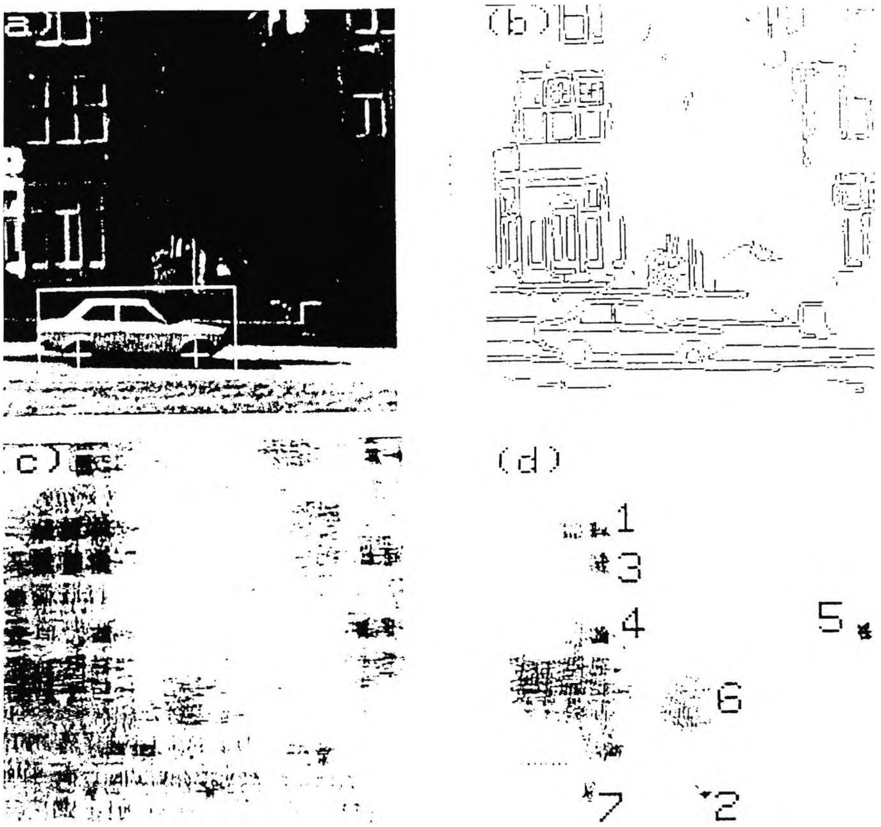
$$(x - a)^2+(y - b)^2= r^2 \qquad (2.2)$$

One of the main virtues of the Hough Transform is its ability to accumulate partial local evidence in an image for a shape into its parameter space, giving strong support for the existence of that shape in the image. Once the Hough Transform is complete, the algorithm looks for the centres of wheels or wheel-arches in the parameter space by finding maximum values and performing statistical analysis on that portion of the accumulator space, based on the assumption that the vehicle is approximately horizontal and a pair of centres are to be found.

The algorithm found the vehicle in the image, as long as the vehicle was viewed side on. It was stated that this form of identification was likely to be used more as a region cueing aid for a more complete object recognition algorithm than as a stand alone recognition system. Figure 2.2(a) on the next page shows the highlighted wheel centres and a cue window for the car and figure 2.2(b) shows the edge input data to the Hough transform after the fractal line finder was applied with threshold set to $0<=R<=0.5$. Figure 2.2(c) shows the circle Hough transform space for $r<=15$. Finally figure 2.2(d) shows the extracted peak regions from the Hough space labelled in the order they were extracted.

Techniques for model based vision usually rely on a hypothesise, test and refine cycle to recover an accurate estimate of an objects pose, typically using Lowe's method [14] or an exhaustive tree search. However Lowe's method has been found to frequently fail (L. Du, [15]) and an exhaustive search is computationally expensive. An alternative approach has been developed by L. Du [15], where detection of an object specific cue feature begins a search for additional evidence to support that cue feature. This method provided a better estimate of the pose of the object as it is now based on the correspondence of an extended feature set. This grouping method is referred to as the viewpoint consistency constraint (VCC).

Measurement of the viewpoint consistency is based on the match between model features and image features determined by three criteria. First the difference in orientation between the model features and the image features (measured in degrees), second the perpendicular distance between the image and model features (measured in pixels) and third, the length of the image feature to the model feature.



Figures 2.2(a) to 2.2(d) Results obtained using Radfords open world vehicle detection algorithm, (Radford [13]).

Two criteria are used to determine the viewpoint consistency constraint, the first is a graded measure of the result of matching each of the above three features with each of the features being weighted to ensure that they are roughly equal. Secondly a binary acceptance measure is used given a pre-set threshold to accept or reject the match.

The starting point for determining the set of features for measurement of viewpoint consistency is derived from an initial pose estimate (Lowe, [14]) and to allow for possible inaccuracies with this initial pose estimate a relaxed value is used for the pre-set threshold. This gives a set of n data and m model features which require an exhaustive search of all possible pairings of these features. The evaluation of consistency is the main computational burden for determining the 3D grouping between image features and model features, with the computational cost being determined by the estimate of the number of VCC evaluations. While there maybe a perfect match, a strategy is needed to find desirable matches and pursue these.

This 3D grouping problem has been tackled by a number of methods. Lowe's incremental model matching method (Lowe, [14]) works iteratively and has the advantage that no backtracking is needed so that it is computationally inexpensive. When it is applied to more complex images, such as natural open world scenes where there is a large amount of clutter in the scene, the method fails due to mismatching features in the image with features in the model, particularly initial features.

Mismatches cannot be corrected nor can further mismatches be prevented. This failure is typical for systems that are attempting to match 3D models to 2D edge images. Alternative methods to this, are to use shape constraints to specify thresholds between feature pairs, but even these shape-only constraints cannot discriminate against clutter.

The VCC can be used to prune an interpretation tree however, providing an acceptance criteria that justifies the pruning. Interpretation trees are combinatorial, but the pruning operation of the VCC removes subtrees and hence reduces the computational complexity. L. Du [15] experimented with different criteria, by changing the pre-set threshold for the VCC, but it was found difficult to successfully solve the grouping problem with open world images. A new algorithm was developed for solving the 3D grouping problem using a state space representation approach.

The 3D grouping process became a sequence of state transitions with the performance determined by the initial state, transition steps and a termination condition. The developed algorithm was called the Viewpoint Consistency Ascent (VCA), this new algorithm demonstrated an improved reliability and a worse case computational complexity of $O(n^2)$, it also demonstrated that in cluttered images new methods would have to be developed to impose the VCC constraint for model matching far more stringently. Figure 2.3 on the next page shows the results obtained from using both Lowe's algorithm and the VCA to recover the pose of the vehicle.

2.3 Model Based Tracking.

The model based techniques outlined in the section 2.2 have shown techniques that can be used for the recognition and the pose recovery of a vehicle in a single image. This knowledge can be used to track an object (typically a vehicle) moving through a sequence of images. Work has been carried out under ESPRIT P2152 VIEWS project, where model based techniques are used to classify and track moving objects in uncontrolled and cluttered scenes.

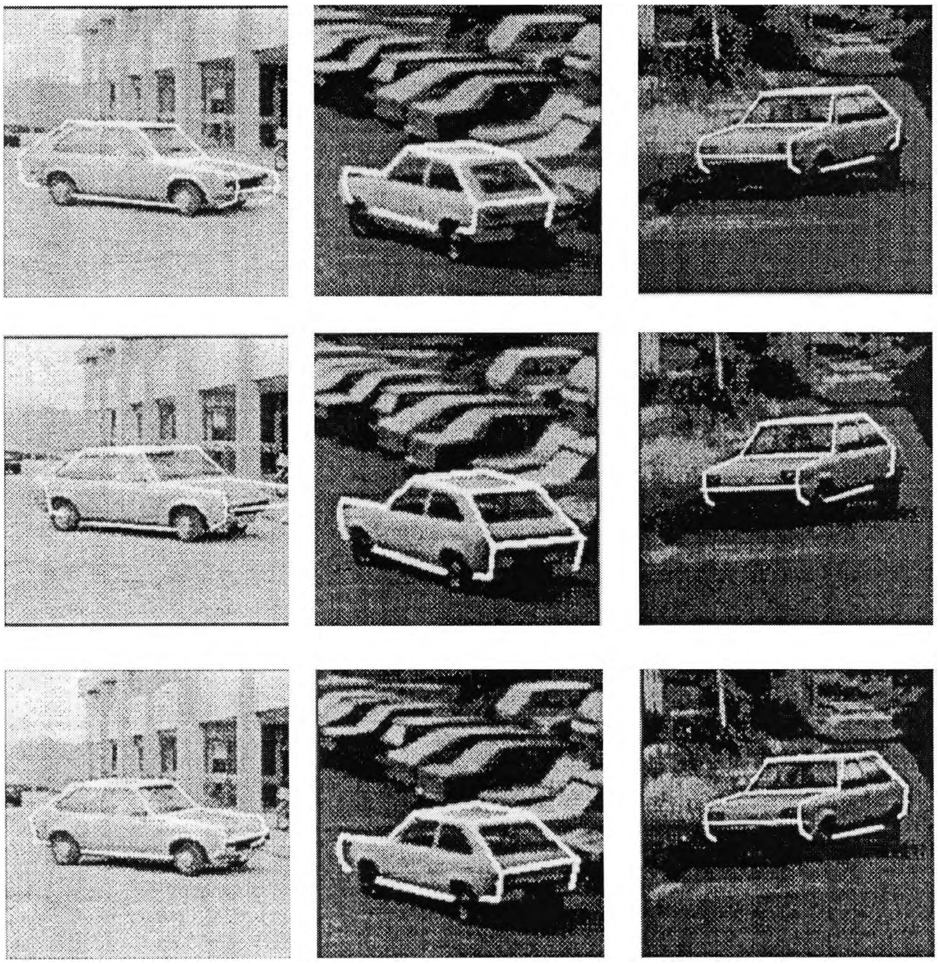


Figure 2.3 The top three images are the initial pose of the vehicle, the middle three images are the results using Lowe's algorithm and the lower three images are the results obtained using the VCA, (L. Du 15).

Worrall [2] uses models consisting of three dimensional geometric representations of known vehicles together with a constructed camera and scene model to track vehicles moving in a road traffic scene. Having models for both the camera and scene, and given the initial position and orientation of the vehicle moving in the scene, a 3D model can be projected onto a set of extracted 2D image features. A 'goodness of fit' score can be obtained by comparing the modelled features with the extracted image features.

This results in a search in both position space and orientation space which is used to maximise an evaluation score between the extracted image features and the 3D model features. When the maximum score is found, the three dimensional position and orientation of the object is known and this information used to predict the position and orientation of the vehicle in the next frame, enabling the tracking of the vehicle through the sequence of frames. The evaluation process essentially defines a scalar function of six dimensions. In world co-ordinates this defines three Cartesian co-ordinates and three angles.

To cut down on the computational requirements, the vehicle can be specified as travelling on the ground plane. This assumption limits the search space from six dimensions to three. A search space of three dimensions can still be computationally high to search and performing three one dimensional searches has been found to be quicker.

The system demonstrated that it could recover the position and orientation of the vehicle to a fair degree of accuracy, however the tracking tended to go awry when strong edges in the image were detected that were not part of the object and the recovered position tended to oscillate when the vehicle was viewed head on.

The paper demonstrated that model-based vision techniques developed for the recovery of the pose of a vehicle in a single image could be used to recover the position and orientation of a vehicle in a sequence of images and track that vehicle across the sequence. The constraint of the vehicle only being allowed to travel on the ground plane, as outlined in the previous paragraph reduces the problem of localisation and recognition of the vehicle from a six degree of freedom problem to a three degree of freedom problem. Figure 2.4 on the next page shows the superposition of the 3D model onto the original 2D image from selected frames between 140 and frame 280.

Using this constraint, Tan [1] developed a generalised Hough Transform algorithm based on grouping evidence from line features. This is used to identify approximate poses of a vehicle. Simple geometric reasoning about the peaks in the expected orientation of the object give rise to mutually exclusive object hypotheses of the object pose and accepting the hypotheses with the greatest evaluation score, recovers the correct pose of the vehicle. The algorithm is fast and robust coping with identifying the vehicle in an outdoor scene even if the vehicle became partially occluded.

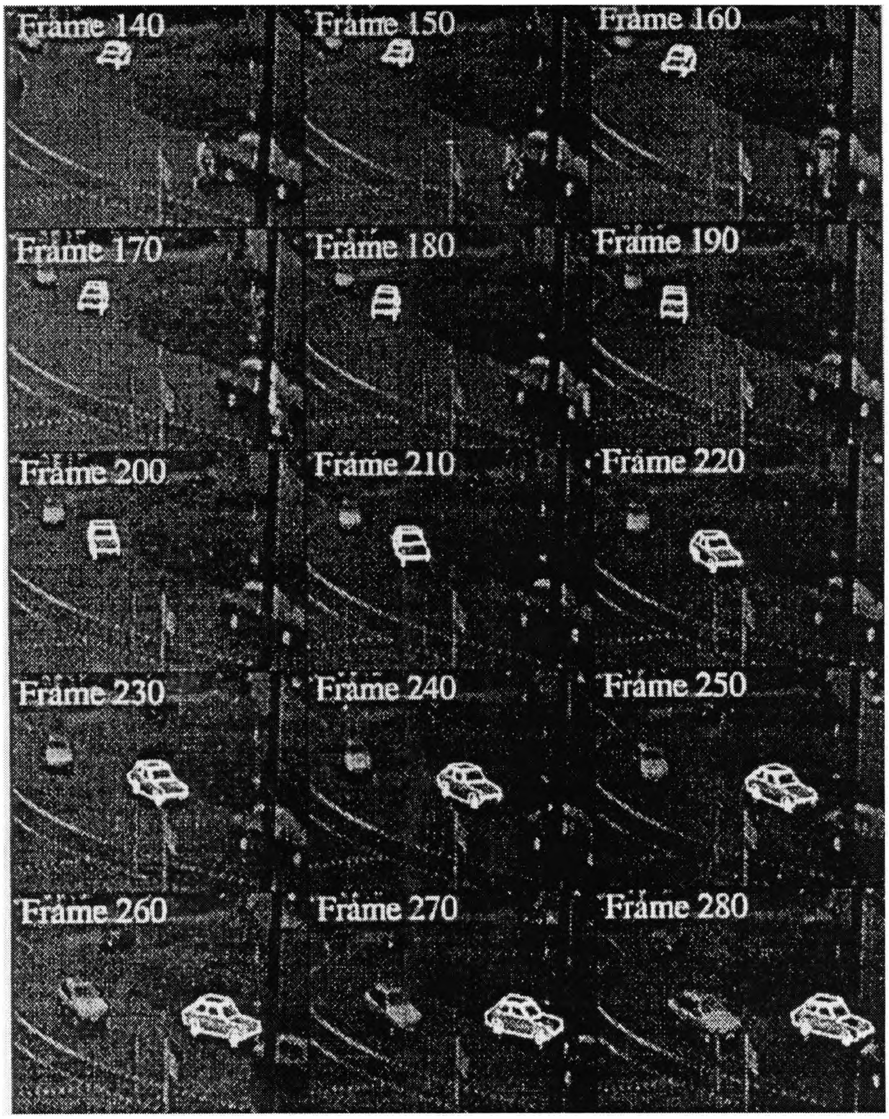


Figure 2.4 Tracking a single car in an image sequence from frame 140 to frame 280, (Worral [2]).

The system could also reject image clutter generated by this form of scene, still being able to extract the vehicle from the scene in area's of high clutter. Model based recognition schemes usually require explicit feature extraction and matching, which has a high computational cost. The developed algorithm eliminates the need for explicit feature extraction and matching and hence the computational costs are considerably lower requiring neither explicit line segments nor symbolic image features. This is made possible by using two assumptions, (i) the ground plane constraint, which reduces the number of degrees of freedom from six to three and (ii) the weak perspective assumption, where if the angle subtended by the object is small and the object is viewed on the axis, then the projection is scaled orthographic.

The algorithm determines the object orientation by matching image line directions with directions of model lines. These are estimated from the peaks in a 1-D histogram of the gradient directions of the original intensity image, this eliminates the need for explicit line extraction. The algorithm computes the location of the object on the ground plane by analysing the projections of the intensity gradients along the directions in the image as determined by the orientation previously found.

The algorithm was extensively tested with outdoor traffic scenes and could successfully recover the vehicle from the scene. This algorithm considerably reduces the computational costs for the model matching, with the constraints specified and the algorithm has been developed specifically for real-time applications.

Koller et al [17] developed a system that would automatically track vehicles in image sequences of road traffic scenes. The system exploited *a priori* knowledge about the shape and motion of the vehicles, with a vehicle model being used for interframe matching and a recursive estimator based on a motion model being used for the motion estimation. Motion was initially detected by segmentation of optical-flow vectors which are assumed to represent a moving object in the image. Based on the assumption that detected motion is on the road and that the motion is forward yields an estimate of the position and principle orientation axis of the model as this is assumed to be parallel to the motion direction. Straight line edge segments are extracted from the image and matched to line segments in the model using the Mahalanobis distance (Deriche et al [31]).

In order to avoid incorrect matches between model segments and image segments that arise from shadows of vehicles, an illumination model provides *a priori* knowledge of the geometrical relationship between vehicles and the projection of shadows from the vehicles onto the road. A motion model describes the dynamic behaviour of the vehicle in the absence of any knowledge of the drivers intention. If the steering angle of the vehicle remains constant, then the motion of the vehicle is described by a constant angular component and a constant magnitude component. To allow for the unknown driver intention, a process noise component is added to the motion parameters. These motion parameters are estimated using a recursive maximum *a posterior* estimator. Figure 2.5 below shows the results of the tracking system when the vehicle being tracked is well defined against the road (the white vehicle produces well defined edges together with strong shadow edges).

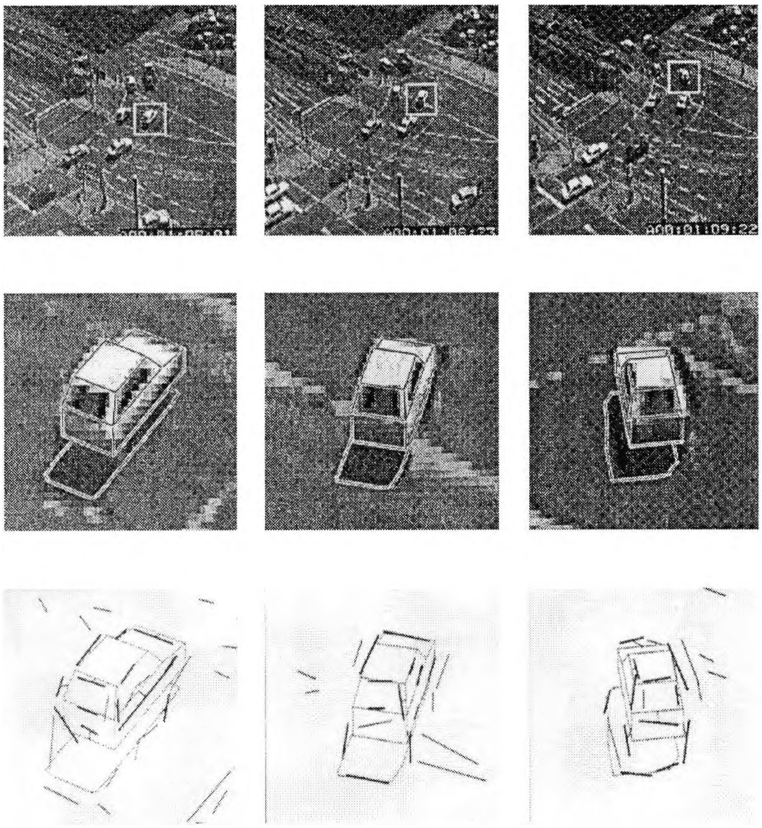


Figure 2.5 The 3rd, 25th and 49th frame of a road traffic sequence. The top row shows the original image sequence, the middle row shows the corresponding enlarged portions of the image where the detected vehicle is. The lower row shows the extracted line segments and model segments in the same enlarged section of the image as the middle row, (Koller [17]).

Figure 2.6 on the next page shows the results for tracking a vehicle that does not produce well defined edges. In the first case the inclusion of the illumination model allows for correct matching of edges in the image with the model of the vehicle, in the second case the stronger edge lines of the shadow enable the matching process to match the shadow edges as there are poor edges extracted from the vehicle.

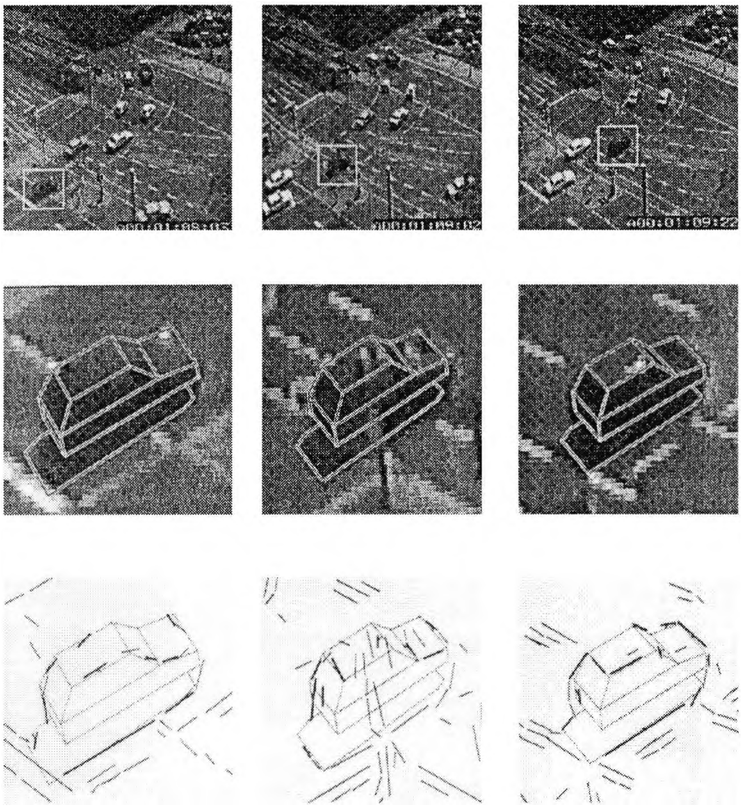


Figure 2.6 Shows the original image sequence, an enlarged portion of the image where the detected vehicle is and the extracted line segments and model segments in the same enlarged section of the image as the middle row. This example demonstrated the necessity of using shadow edges in the matching process as the dark coloured car produces very poor edges against the road, (Koller [17]).

An alternative method for model based tracking to that of specific geometric model matching previously described, is to use a deformable model. Shen [16] developed a method for recovering the shape of a 3-D object from a moving sequence of images using a deformable model. The deformable model is initially set to be spherical, but is allowed to be progressively deformed under the action of simulated forces derived from the image and applied to the model. This drives the model to fit the profile of the object extracted from the image. The model is constrained to be symmetric to a plane parallel to the direction of motion, the resulting deformed model represents the 3-D shape in the image, effectively recognising and tracking the object. The system was tested on several sequences of images depicting a traffic scene in a car park, demonstrating that from the initial spherical state the model progressively deformed into the shape of the vehicle under the forces derived from the image and approached the shape of the image object, in this case a vehicle.

2.4 Tracking Objects in Cluttered Scenes.

The previous section has concentrated on tracking rigid objects that can be well described by a geometric model. However the vehicles to be tracked may be several hundred meters from the camera and consequently will only occupy a small number of pixels. The object will therefore not provide sufficient extracted image features for matching with a model. In fact at these ranges the vehicle may even appear to be an articulating non-rigid objects as it moves and changes orientation in the scene. These features tend to suggest that geometric modelling would become less robust as the distance increases and that tracking based on some form of crude object descriptor may be more robust in such circumstances.

Rosin and Ellis [3] developed a knowledge based vision system for the interpretation of alarm events resulting from a perimeter intrusion detection. Unlike the previous systems which are tracking rigid geometric objects, Rosin and Ellis are concerned with tracking articulating non-rigid objects which are not easily modelled. The actual vision system has the task of interpreting alarm events, discriminating between humans (an alarm event) and noise (weather-related or animals, false alarms). Problems that occur when recognising complex objects in outdoor scenes, include occlusion, shadows, illumination changes, but more importantly, the large range over which objects have to be detected, recognised and tracked, results in poor spatial resolution.

Motion is detected in the image by using a frame differencing technique. Optical flow or feature correspondence techniques are inappropriate due to the time interval between successive frames (up to 1 second), but more importantly the low spatial resolution of the object to be tracked means that little image data will be available for feature extraction. Following the frame differencing, a thresholding operation forms regions in the image that have shown motion. These regions are analysed by a boundary based feature extraction algorithm which calculates size and position. Temporal filtering allows noise regions to be removed from the image sequence.

Scene models which consist of a map of the area covered by the camera, are constructed for each fixed camera, (a map contains labelled areas such as the ground, fence etc). There is a camera calibration model for each map. The map allows sequences to be ignored that are outside the trigger zones, the labelled map aids model matching by restricting the interpretation based on the location of the object in the image, and the camera model enables range measurements of objects to be made. Target models are described by two components; the first component describes individual instances of the object, the second describes the dynamic behaviour of the object over time. Model matching is performed in a top down manner, with matches between individual models and models extracted from the image sequence. At each level in the model matching, the match with the highest probability is chosen. The system reliably detected and classified humans in a number of test image sequences, however it was less robust in its classification of false alarms.

Figure 2.7 below shows the binary detected objects for two birds overlaid on the original image (top). The middle and bottom images in figure 2.7 show the results for a human running across the image. The images show the minimum bounding rectangles detected in the sequence and the final extracted sequence for the human. The system correctly classifies this form of image sequence as a human.

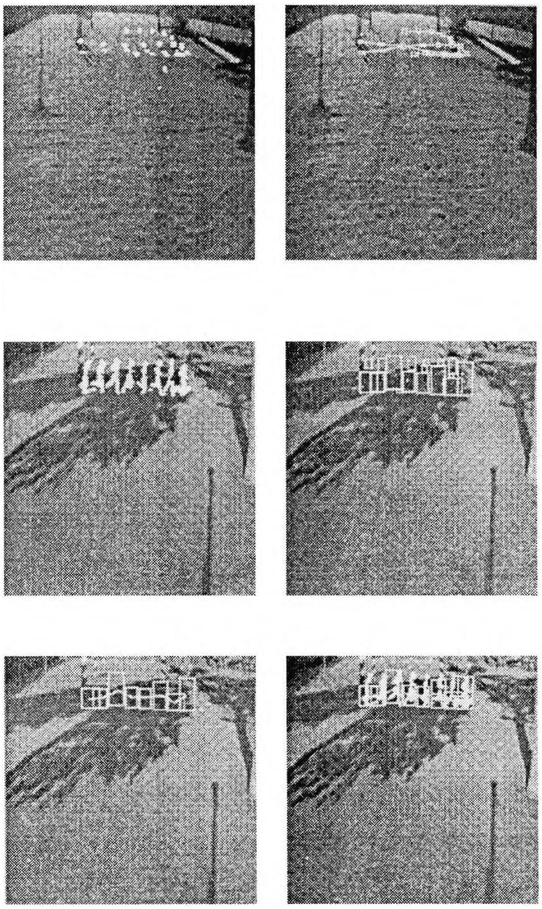


Figure 2.7 Binary objects detected and tracked, birds (top) and a human (centre) and (bottom), (Rosin and Ellis [3]).

Although not aimed at any specific application or defining any particular object, Picton [19] developed a system that could segment and track a moving object in a natural scene without any specific prior knowledge. The system used only general object knowledge, hence making it as flexible as possible. Picton proposed a system which is only interested in pixels that are moving and that certain predictions can be made about the scene based on the information obtained from the changing pixel values only.

He loosely based his system on studies in biological vision carried out by Schneider [20], who demonstrated that the vision in hamsters can be divided into two separate vision systems, namely: one for following a moving object with the eyes and the second for identifying the object. The first observation would tend to suggest that detecting object motion exists in a much older part of the brain (in an evolutionary sense) than the second. Picton tended to postulate that there is some form of evolutionary advantage in being able to track objects without necessarily identifying them. Differences between consecutive frames of image data is used to generate motion cues about object movement in the image.

However rather than using just difference pixel information, the difference edge (DE) was used. This was calculated using

$$DE = (|F(i,j,n)| - |F(i,j,n-1)|).E(i,j,n) \qquad (2.3)$$

where

$|F(i,j,n)|$ is the grey level intensity at pixel co-ordinates i,j in frame n.

$|F(i,j,n-1)|$ is the grey level intensity at pixel co-ordinates i,j in frame n-1.

$E(i,j,n)$ is the edge strength at pixel co-ordinates i,j in frame n.

If DE is greater than some pre-set threshold, the pixel would be classified as a moving edge. The main points put forward for using 2.3, were that a pixel could only generate a large value if it had both a large difference value and a large edge value. There would also be a requirement for one and not two thresholds values thus reducing one of the areas where vision systems can be made to work in one instance and not another. This point would tend to move the vision system away from a general form to a specific form. The edge strength was determined using a Sobel operator. The Sobel operator has been shown to produce accurate values for the direction of the edge (Kittler [21]). The moving edges found in the current frame are stored and compared with the moving edges found in the previous frame. The values that corresponded to the difference in co-ordinates of the moving edges are used to increment a Hough accumulator cell whose parameters where expressed in terms of the difference in pixel co-ordinates. The system worked well but was limited to tracking only one object and it had problems tracking that object if the object rotated or moved away from the camera.

All tracking systems so far discussed have used monochrome images. Brock-Gunn [18], developed a system for tracking people in crowded scenes by using colour templates. The system examines the colours of objects that are moving in the scene and translates these colours into a template space. The templates are then matched with a pre-stored data base.

A simple frame differencing technique is used to generate object cues. The difference image is thresholded so that objects below a certain size are removed and for each of the remaining objects a four dimensional template (three colour and one spatial) is calculated. The templates are tracked by using a comparison technique, where each template is compared to one in the data base. The use of a four-dimensional template will mean a task of matching over 65000 pairs of values, which would take a considerable amount of computational time. To overcome this problem a hierarchical approach is used to perform the template matching where a pyramid of templates are calculated for each object. The pyramid gets progressively coarser, going from four-dimensions with 16 bins (65536) to four-dimensions with 2 bins (16). The generation of the new templates at each resolution effectively involves averaging the finer resolution bins, requiring a minimum of processing time to calculate the lower resolution templates.

The matching starts at the coarser resolution, where the matching process still contains sufficient information to dismiss incorrect matches and only the object templates that match go on for further processing at the finer resolution. The advantages of this hierarchical approach are self evident, with the computation saving in the order of twenty for a database size of thirty, this rapidly increases as the size of the database is increased. Experimental results of the system demonstrated that two people can be tracked even when one person occluded the other, the system successfully re-acquired the occluded person as that person became visible to the camera. The hierarchical approach worked well for dissimilar looking objects, however for objects that are similar in appearance (same colour structure and size), then these were found to require the higher resolution templates to resolve the matching.

The general nature of the system demonstrated that it was capable of tracking objects of similar size and shape even if those objects have irregular motion and go through occlusion. If a large database is necessary then the hierarchical approach to the template matching can be computationally very efficient, further enhancing the system performance.

Gong [22] addressed the issues of focused computation in computer vision using a scheme to link scene-oriented contextual knowledge and computational constraints to perform visual motion segmentation and tracking in a road traffic scene. The approach claimed that perception is really an opinion on the state of affairs in the world rather than a passive response to sensory stimuli. It emphasised the importance of focused vision using explicit contextual knowledge to constrain the computational requirements of the system in a framework that dynamically determines the way the visual modules function.

The studies showed that it is possible to link contextual knowledge of visual behaviour to an appropriately linked network of chosen parameter sets. Thus the issue of mapping knowledge to computational constraints resides in how explicit contextual knowledge can be represented as distributed implicit parameter sets and what computational mechanisms are used to manipulate these parameters sets. Using a Bayesian network for the knowledge representation (belief network) it demonstrated that improved consistency in segmentation and tracking in the VIEWS system can be obtained for a small computational overhead introduced by mapping explicit knowledge to computational constraints.

2.5 Structure.

It has already been highlighted how complex the task is to identify and track man made objects moving in a natural open world scene. This task has been accomplished to varying degrees of success with several different image processing systems and techniques. One of the objectives of this research was to construct some form of representation of the structure within the scene, a symbolic map that represents that structure prior to or during the tracking. With this map it may be possible to predict object occlusion and identify the regions in the image that are associated with object motion.

Xu Li-Qun [23], presented research on building a model of a road junction using moving vehicle information. The constructed model, specified the ground plane orientation in camera co-ordinates and the position of traffic lanes. The model was constructed based on the movement of vehicles in the scene and no static analysis was performed at all. The developed system used the differencing between two consecutive frames of image data based on the absolute difference between the convolution of two Sobel templates to determine motion cues. Threshold values used were determined by use of the mean of the local minima in the smoothed histogram of the absolute difference between image frames. The difference regions generated were grouped using an 8-neighbourhood connectivity method to form a bounding rectangle. The establishment of feature correspondence between the two related images uses a method where the path coherence and motion smoothness form a similarity matrix. In the similarity matrix are measurements of height, position and width, which are used to form a template. This template is matched with calculated features in the next frame. Problems of discontinuous objects found in each frame are dealt with by the assumption that this is a new object. This simple object tracking method was found to be adequate if not optimal for tracking.

It was found that applying more optimal methods for solving the feature correspondence between frames (such as [24]) were not significantly better but increased the computational cost for solving the correspondence process. The ground plane estimation was based on the fact that under perspective projection, parallel lines in a 3 dimensional scene form a fan of lines in a 2 dimensional scene. These lines all intersect at a common point, this point is called the vanishing point. The vanishing points are determined by segmenting the trajectories into straight line segments and using a parameterised Gaussian hemisphere together with a hierarchical Hough transform to find the vanishing points. Given two vanishing points, it is possible to obtain the ground plane parameters up to a certain scale factor.

Each line segment should pass through a vanishing point and the orientation between the trajectory line and the horizon is measured. These measurements are weighted by the duration in time of the trajectory and then inserted into a histogram. Smoothing the histogram and finding local maxima yields the lane centres and taking a bisection of the centre in the direction of the segment gives the lane boundaries. The system demonstrated that by making certain suitable assumptions a model of a traffic scene could be constructed. Figure 2.8 on the next page shows the tracking trajectories found by the system and the estimated horizon together with the lane structure determined by the system superimposed on the original image.

Stat [25] describes results of ongoing work in context based vision which is trying to recognise objects using both 2 dimensional and 3 dimensional information. The system analyses complex outdoor scenes and using simple procedures, processes colour and stereo monochrome images to build a scene description of the image. One of the major problems in analysis of natural scenes is that natural objects do not have uniform shapes and as the local surface properties of these objects is variable, it is difficult to uniquely determine their identity. Explicit contextual knowledge is used to control the decision making process involved in identifying the natural objects in the scene. The developed system, 'condor', integrates information obtained from both 2-D and 3-D images and uses a consensus of many simple procedures to achieve reliable results. It exploits contextual information to aid its recognition process and augment its own database of contextual information with the results of its own recognition process.

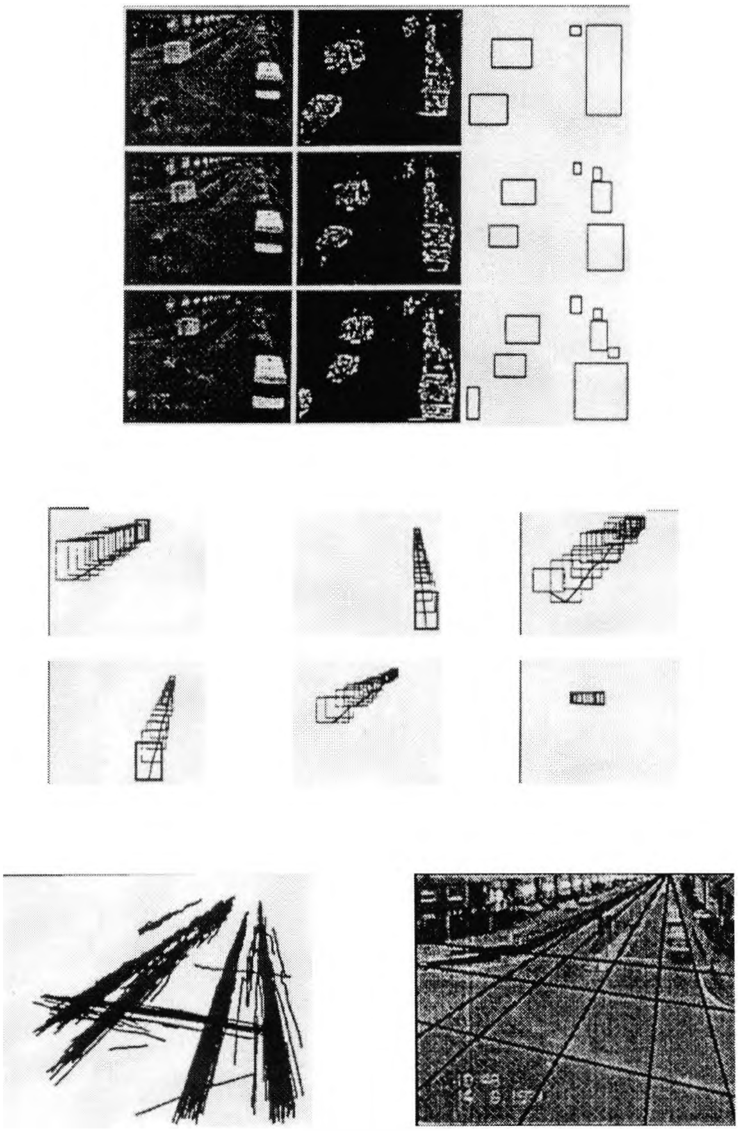


Figure 2.8 Trajectories found by the tracking system (top & middle rows), the lane structure superimposed on the original image (bottom left) and the trajectory line segments together with estimated horizon position (bottom right), (Xu Li-Qun [23]).

This technique improves the system performance incrementally over time. The 'condor' system eliminates the traditional dependence on stored geometric models and the usual image segmenting algorithms providing a basis for semantic interpretation of the image. The system only analysed static images and does not take into account any motion in the image.

Toal et al [26] developed a knowledge based system for spatio-temporal reasoning within a traffic surveillance system. The system integrated a perception component which detects and recognises the vehicle trajectories and a situation assessment component which understands a situation as it develops over time. The system required additional modules for the real-time control and scene acquisition together with behavioural knowledge of the objects in the scene.

The main exemplar the system was tested on, concentrated on a roundabout where a lorry occluded a saloon car, with the saloon car later re-emerging from behind the lorry. Perceptual processing involves the detection, tracking and classification of the visible moving objects. This information is given by updates at different levels of analysis and speed of processing. These estimates are always with respect to the ground plane and are passed to the situation assessment module in a basic form of <label, position, time>. The first level processing deals with events such as starting/stopping and entering/exiting, the next level deals with consistency checking and maintains space time histories of vehicles moving in the scene.

The paper highlighted that a vision tracking system must be able to overcome problems of occlusion and any situation assessment process must be able to maintain a long term memory capability. This enabled the situation processor to keep high level explicit representations of total occlusions that occur in the dynamic scene, coupling the behavioural knowledge to aid the relabelling of objects that have become occluded and then re-emerge in the scene.

The multi-purpose representation maintains the structure of the world. Key static knowledge requirements are conversion of the ground plane geometry into meaningful regions and a description of the connectivity of those regions. The tracking system also requires a means to represent dynamic information such as the identifying areas occupied by a vehicle, the velocity and orientation of that vehicle and any inter-vehicle orientations and distance. The perception component of the system is based on the VIEWS configuration as detailed by (Sullivan et al, [27]).

In the case of tracking being lost by total occlusion, mechanisms for re-acquiring the object based solely on velocity have been found to be poor performers in the long term due to the unpredictable vehicle manoeuvres that an occluded vehicle might make. The situation processor maintains an occluded by relationship that defines in the image a potential emergence area. This area aids in relabelling the vehicle when it reappears in the image. During a typical occlusion a number of situations may develop, such as vehicles may emerge from occlusion or vehicles may join the occlusion. The occluding vehicle may itself become occluded, the occluded vehicle may exit the scene and the occluded vehicle could become occluded behind vehicles that are already occluded. To overcome these problems behavioural knowledge is used to disambiguate the labelling problems as vehicles emerge. The exemplar demonstrated that the system could deal with total occlusion and develop spatio-temporal histories for use in a behavioural evaluation.

2.6 Discussion And Summary.

Section 2.2 of this review looked at the problems and complexities in using model based techniques to identify man made vehicles in natural open world scenes. The work carried out demonstrated that it was not only possible to identify a man made object in a scene but it was also possible to recover the pose of the object and fit a three-dimensional model of the object to the image data. The techniques developed for the recognition process were further expanded in section 2.3 to show that model based techniques could be used to identify, segment and track a vehicle moving in a road traffic scene with successful results (Worrall et al [2]).

However, one of the major points with these techniques is the fact that the vehicles generally occupy a significant proportion of the image (perhaps 15% or more of the image). With my research the vehicles are expected to be a large distance from the camera and hence will only occupy a small area of the image and under these conditions model based techniques are expected to be less successful.

Koller's tracking system (Koller et al [17]) did track vehicles at street intersections where the vehicle size in the image varied from 30 by 60 pixels to 20 by 40 pixels across the sequence, (more comparable with vehicle sizes I shall be tracking). Their work emphasised the complexity in tracking objects that only occupy a small part of the image. However the vehicles being tracked were moving across a road intersection. This type of scene reduces the edge clutter in the image because the roads, being man made, tend to be smooth and so produce fewer edge pixels. If the vehicles to be tracked only occupy a small proportion of the image and are moving in a open world scene which will have many natural features, then the amount of background edge clutter will be significantly larger. This background edge clutter can have edge pixel boundaries with a stronger edge response than the actual vehicle to be tracked. These clutter edges make the matching of a 3-D model of the vehicle to the extracted edge information far more prone.

When the object to be tracked only occupies a small area of the image, it has been found that matching crude object descriptors is more robust, (Rosin [3], Teal [4]). Section 2.4 highlighted the fact that when the objects to be identified and tracked were at a larger distance from the camera then model based methods tended to fail and instead of trying to match complex models to the image data, the matching of simple object descriptors provide a more robust method for tracking.

The work by Xu Li-Qun [23], showed that it was possible to track vehicles moving in a traffic scene using simple techniques and still achieve accurate and robust results. From these results it was possible to build a representation of a traffic scene based only on the motion of the objects moving in the scene. Again however the vehicles were close to the camera, the camera being mounted approximately 25 meters above the road junction. This shows that complex techniques are not required and that tracking can be achieved using knowledge of the objects motion characteristics and the context in which the tracking system is being used.

Static analysis of open world natural scenes has been widely researched with many schemes, systems and algorithms being developed to build up a representation of the structure within the scene ([25], [29], [30]). They make use of specific object models and tend to analyse open world scenes where objects to be identified in the scene and on which the interpretation of that scene is to be performed are close to the camera.

This research is concerned with the identification and tracking of man made objects moving in a natural open world scene where these objects are expected to be a large distance from the camera. If the tracked object becomes occluded, then the fact that the tracked object has become occluded behind a static object in the scene is of interest, not the fact that this static object is a hedge or a wall and as such these systems yield little usable information.

The man made objects of interest here are vehicles and as such will have a rigid not articulating geometry, but at a large distance from the camera coupled with the fact that they can be viewed from any angle, these objects only occupy a small proportion of the image pixels. With reference to figure 2.1, it is self evident that the lighting conditions can vary across the entire scene causing illumination changes and shadows that make edge based model matching analysis difficult as the scene will yield a considerable amount of edge clutter. This edge clutter has already been shown to make model based tracking particularly difficult. This of course does not include the added complexity of self occlusion, occlusion by another tracked object and occlusion by a static object in the field of view.

In addition to these problems must be added the large range over which the system has to identify and track objects. The tracked objects at large range may only occupy 30 by 40 pixels. Consequently very little information is available for a model based tracking method. To this fact must be added that model based methods require specific geometric models of the scene, knowledge of where the ground plane is and a camera model. With reference to figure 2.1, the identification of the roads, ground plane etc would be a considerable task in its self and would require camera models for a large number of areas of the image (Rosin, [3]).

The identification and tracking system being developed by this research is to have no initial specific geometric knowledge of the scene. Therefore the system being developed will have to be able to detect and track using a new form of identification and tracking process. Rosin [3] has shown that the matching of crude object descriptors can be more robust for tracking when the objects to be tracked have poor spatial resolution. Ballard has suggested that vision is best understood in terms of the context of visual behaviours in which the vision system is engaged as these behaviours often do not require elaborate representations of the three-dimensional world (Ballard, [28]).

As already outlined the system will have no specific geometric knowledge of the scene, it is to be a ‘plug and go’ system. The ideas of Toals grammar based system maintaining history files of objects being tracked (Toal, [26]), will be extended to deal with occlusion of objects being tracked. Using the history of tracked objects, regions in the image will be associated with dynamic events.

These dynamic events effectively map out areas in the image where objects can be expected to be observed moving. Breaks in the trajectories of tracked objects could define regions where objects may become occluded behind other objects that are in the field of view. These regions effectively form a structural representation of the scene (a map). This structural representation of the scene is generated on a frame by frame basis with continual updating every time new objects are tracked in the scene. If the motion is seen on a regular basis in certain regions of the image, this would further increase confidence, that the region is associated with vehicle motion (a road for example).

Chapter 3 System Overview.

3.1 Introduction.

When tackling complex problems a frequently used approach is to split the problem up into several smaller tasks (modules). Each of these modules is then solved independently of the others. When the smaller tasks have been implemented and tested, they can be gradually recombined (system integration). The integration process highlights if there are any problems with the modules. If a problem is to be tackled using this approach, then some form of structured analysis and design methodology is required to ensure that the developed system meet its functional requirements.

Throughout this research, the YOURDON structured analysis and design methodology has been used for the development of the machine vision system. Appendix A of this thesis gives an overview of the YOURDON structured analysis and design methodology.

Complex image processing systems require a visual control strategy, which will dictate the processing and flow of information through the system. Generally two possible control strategies can be employed; namely: Hierarchical control and Heterarchical control. Hierarchical control systems operate by controlling the flow of data through the system in either a bottom-up or top-down manner. A bottom-up system is where the control process is data driven, starting with raw image data it proceeds to segment structures, produce geometric relationships and finally produce decisions based on the processed image data.

A top-down system is where the control strategy is driven by an internal model where high level models in the knowledge data base generate expectations and/or predictions of the geometric, segmented or image structure that is in the input data which is then verified.

This form of strategy is commonly implemented in a hypothesis and test procedure where an initial hypothesis is made, the system attempts to verify this by calling high level modules which process the image to produce data that either supports or dis-proves the initial hypothesis. This control method has an advantage over a bottom-up control strategy in that the evolving hypothesis can be used to guide the processing, choosing one from a number of methods for extracting information from the image data, whereas in bottom-up, all the image processing methods are run simultaneously.

Heterarchical control uses a combination of both top-down and bottom-up control and can sometimes yield better results. For example an initial bottom-up control could be used to generate the most likely hypothesis that then invokes a top-down strategy. The image processing system being developed here is mainly data driven. It requires information derived directly from the input image data, and as such a bottom-up control strategy has been used.

3.2 Overview of the system.

When designing any system, the first major task is to identify the input and output information. The context diagram in figure 3.1 identifies all the external interfaces to the system (the terminators) together with the data that will flow to and from these interfaces. From this diagram and by analysing the aims and objectives of the image processing system that is to be developed by this research, a first level DFD for the target tracking and image interpretation system was designed and is shown in figure 3.2.

The first level DFD identifies the main processing tasks and how the processed data from these tasks will flow through the system. From the level 1 DFD the data processing tasks identified are:

(i) Process 3:- ‘Acquisition And Motion Detection’.

(ii) Process 4:- ‘Target Identification And Tracking’.

(iii) Process 5:- ‘Spatial-Temporal Reasoning’

and the control processing tasks identified are:

(v) Process 1:- ‘Goal Processor’.

(vi) Process 2:- ‘Graphics User Interface’.

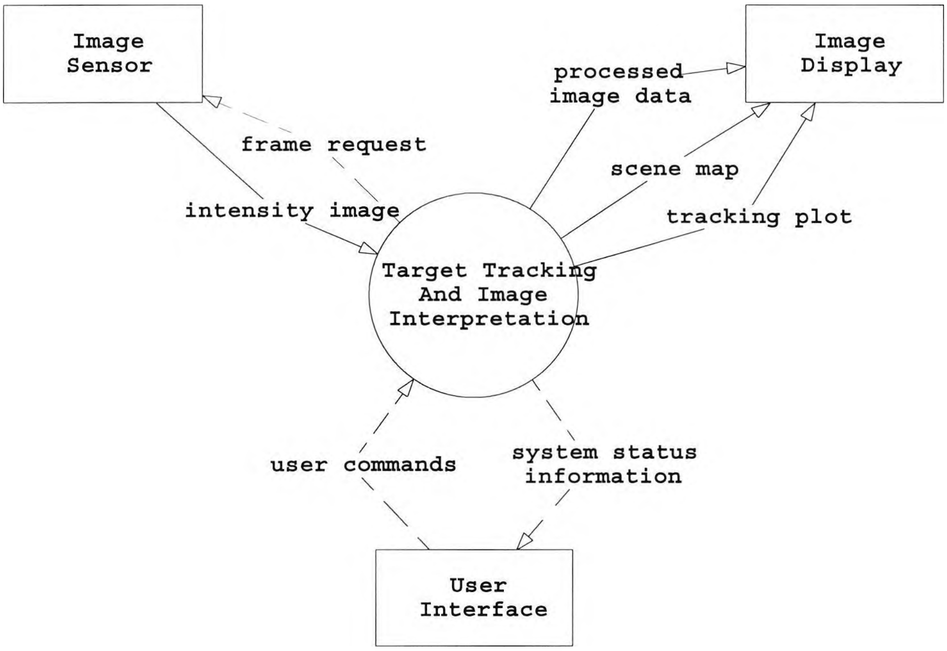


Figure 3.1 Context diagram for the target tracking and image interpretation system.

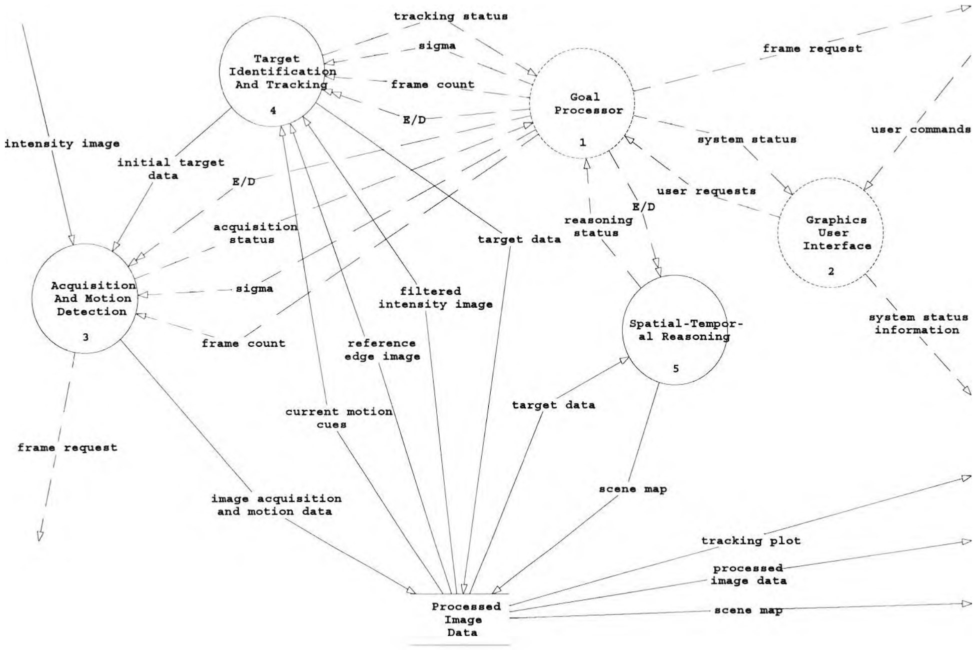


Figure 3.2 Level 1 DFD for the target tracking and image interpretation system.

3.2.1 Data Processes.

(i) Process 3 'Acquisition And Motion Detection'.

This processing task has four sub-processes to perform, namely

(a) acquire intensity images from the sensor.
(b) filter the images to reduce noise.
(c) produce reference image data for the motion detection.
(d) generate motion cues based on the differences in statistics between the current input frame of image data and the reference frame of image data.

This task performs processing in the following order 'a', 'b', 'c' and finally 'd'. The acquisition and motion detection process then returns a status signal indicating the results of the processing just executed. This task's processing actions can be overridden by the goal processor in that only certain processes may be called; thus the goal processor can alter the processing requirements of the acquisition and motion detection process.

(ii) Process 4 'Target Identification And Tracking'.

This processing task has two sub-processes to perform, namely

(a) initial object identification.
(b) tracking.

The process determines which motion cues are objects and which motion cues are targets (vehicles). It tracks both objects and targets, generating target and object data. Processes 4 performs its processing tasks 'a' first then 'b', a status signal is returned upon the completion of sub-process 'b' indicating the status of the processing just carried out by the target identification and tracking process.

(iii) Process 5 'Spatial-Temporal Reasoning'.

This processing task has two sub-processes to perform, namely

(a) spatial analysis
(b) spatial reasoning

This process constructs a symbolic map of the image based on the motion of targets moving within the image. It defines areas where targets are likely to be observed moving, and areas where targets could become occluded.

It takes data from the target identification and tracking process and analyses the target tracks across multiple frames. It attaches a statistical probability to each identified region. The probability value effectively gives an indication to the confidence with which the region has been identified in the map. As with processes 3 and 4, the spatial-temporal reasoning process performs first task ‘a’, then ‘b’ and a status signal is returned indicating the status of the processing just carried out by the spatial-temporal reasoning process.

3.2.2 Control Processes.

(v) Process 1 ‘Goal Processor’.

The goal processor controls the entire system; it is responsible for two main tasks, namely

(a) belief maintenance

(b) goal achievement

The goal processor first performs all initialisation required by the system. It carries out the task of belief maintenance, a passive background data driven activity that keeps beliefs consistent and updated. The goal processor is also responsible for the goal achievement of the system, an active knowledge driven foreground activity that consists of planning all future system activities. It also performs all the systems error detection and correction activities to ensure that the processing carried out by the system is consistent with the aim of the system.

(vi) Process 2 ‘Graphics User Interface’.

The graphic user interface provides an interactive capability between the user and the image processing system. It permits the user to display results, alter processing priorities, take alternative error corrective actions and override system decisions.

The three data transforms form the core of the image processing required by the system and all the image processing carried out by these data transforms, together with the results of intermediate processing carried out within each task, is written to the processed image data store. All data written to the processed image data store is accessible to the user via the graphics user interface.

3.3 Discussion and Summary.

The YOURDON structured analysis and design methodology has been extensively used in the analysis, design, and development of the target tracking and image interpretation system. Section 3.2 used YOURDON to analyse the main interfaces to the system (context diagram) and build a top level design for the target tracking and image interpretation task (level 1 DFD). The first level DFD shows the key data and control processes, together with how the processed data will flow around the system.

A brief explanation of each of the data and control processes has been given and appendices B and C show the complete behavioural and environmental models developed during this research for the system. The actual implementation has been undertaken using the C++ programming language, the developed code has been written using Borland's C++ compiler version 3.1 for the DOS environment.

Chapter 4 Acquisition and Motion Detection.

4.1 Introduction and Overview.

It has been demonstrated that motion can be extracted from monochrome images obtained from a static camera using a frame differencing technique to perform the motion detection (Rosin [3], Picton [19], Brock-Gunn [18], Karmann [86]). The level 1 DFD for the system (figure 3.2) has identified that the task of motion detection will be carried out by the 'acquisition and motion detection' process. After initialisation the first processing task the system must perform is to acquire images from the sensor and determine if there was any motion in those images. Analysis of image acquisition and motion detection process, identified that this task can be broken down into four smaller data processes. A second level data flow diagram describing these processing tasks is shown in figure 4.1 on the next page. The four process identified are:

(i) acquiring the images.
(ii) filtering those images (noise removal).
(iii) reference data generation (if required).
(iv) motion cue generation.

The level 2 DFD does not show the dynamic behaviour of the image acquisition and motion detection sub-processes, i.e. the order in which these processing tasks are going to be performed. To do this we require a state transition diagram. Figure 4.2 on the next page shows the state transition diagram for the 'acquisition and motion detection control' process. This diagram describes the dynamic behaviour of the image acquisition and motion detection process.

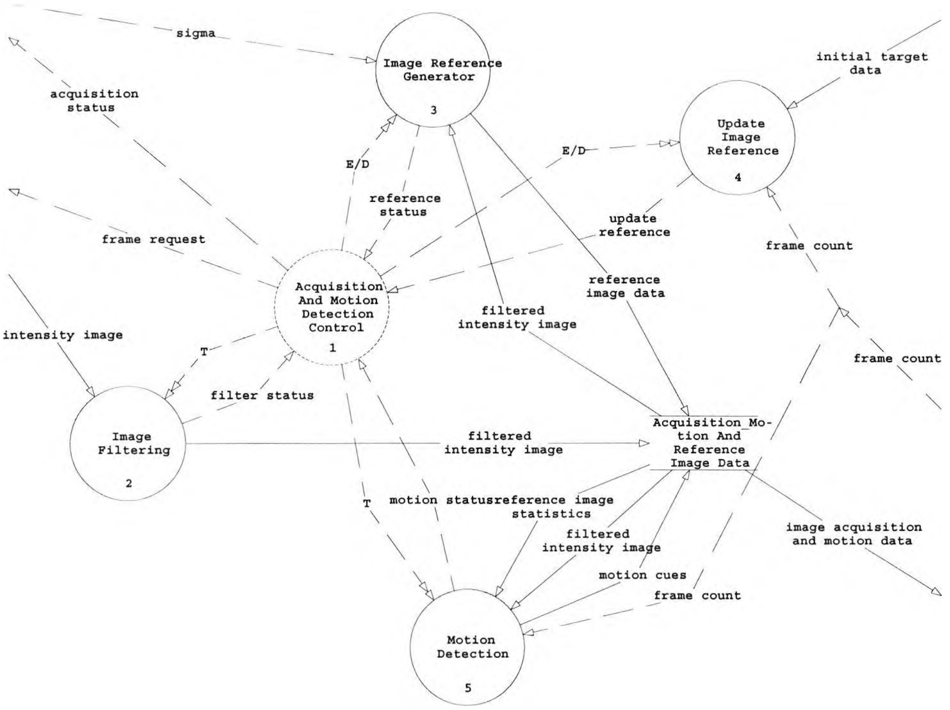


Figure 4.1 Acquisition and motion detection Level 2 DFD.

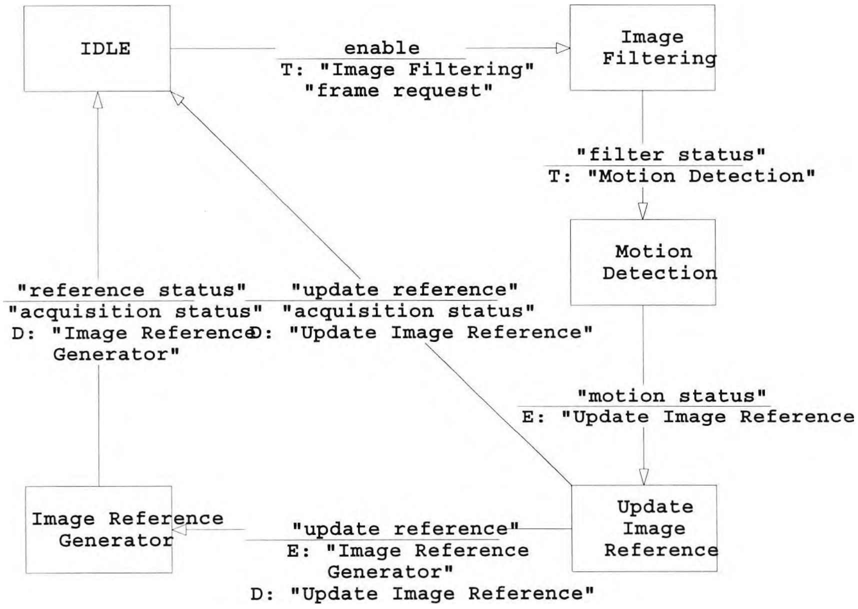


Figure 4.2 Acquisition and motion detection control State Transition Diagram.

Each of the four data transformation processes generates a control signal upon completion of its processing. These control signals are used as conditions to traverse the state transition diagram generating the trigger or enable signals that control the processing and flow of data through the acquisition and motion detection process. Upon receipt of an enable signal from the 'goal processor', the 'acquisition and motion detection control' generates a frame request to the image sensor which supplies a frame of image data to the system (the intensity image). A trigger is then sent to the 'image filtering process'.

The image filtering process applies a median filter to the intensity image producing the ‘filtered intensity image’. Upon completion of its processing the image filtering process passes a filter status control signal back to the acquisition and detection control process. The filter status signal triggers the 'motion detection process', which performs statistical analysis on fixed size areas of the filtered intensity image.

This process calculates the difference between the current frame of image statistics and a reference frame of image statistics. The results of this operation provide regions of interest in the image where object motion may have occurred (motion cues). Upon completion of its processing the motion detection process passes a motion status control signal back to the acquisition and detection control process.

The motion status signal enables the 'update image reference process' which uses the results generated by the 'initial target analysis process' (chapter 5) to determine if a new set of reference image data needs to be generated. After analysis of the perceived motion in the image is complete, the acquisition and motion detection control process receives an update reference control signal. If new reference image data is required then the 'image reference generator process' is enabled, which then generates new reference image data based on the current frame (intensity image).

Finally the acquisition and motion detection control process passes an acquisition status word to the 'goal processor', indicating the status of the processing carried out this frame. The filtered intensity image, reference image data and the motion cues are all written to the 'acquisition, motion and reference data store'. This store is accessible by the rest of the system.

4.2 Image Filtering.

The images used to test the system were obtained from a static camcorder set up to film sequences of open world scenes where objects (typically vehicles) were moving in that scene. The output from the camcorder was later digitised to disc as a 768 by 576 pixel intensity image with 8 bits per pixel using JPEG coding. From these 768 by 576 images, 512 by 512 pixel images were generated by taking the centre 512 pixels only. The camcorder images are initially recorded onto tape and later played back into a image digitisation system. This form of image generation results in noise being added to the image. Noise has been added to the image from:

the camera, (optics, digitisation, etc)
recording onto tape,
playing the recorded image back,
the digitising processes (JPEG etc).

The noise added from these sources will be accumulative and complex to analyse. However I have assumed that this accumulative effect may be approximated by a Gaussian distribution and that spatial filter operators will reduce the effects of the accumulated noise. The noise added to the image will have the effect of generating false motion cues and edge clutter. These effects need to be minimised by some form of filtering operation. Spatial filter operators can be used to suppress noise in an image due to the fact that noise generally has a higher spatial frequency spectrum than normal image components and may be effectively reduced (smoothed) by using a low pass filter.

If we consider a general 2D convolution operator:

$$G(i,j)= \sum_n \sum_m P(x,y)H(x-i,y-j) \tag{4.1}$$

where

$G(i,j)$ is the output image.
$P(x,y)$ is the input image.
H is a convolution mask.

For a low pass filter, typical masks available are simple average, centre emphasis, centre plus neighbourhood emphasis; these are only a few of the common low pass filter operators available to smooth out noise. Each mask contains a scaling factor to give the filter an overall gain factor of unity. The filter works by convolving the weighted masks with the image pixels, summing the results of the convolution process and multiplying the result by the scaling factor gives the new pixel value. A problem with using a simple averaging filter to overcome noise in an image, is that it tends to smooth out image features such as edges. Edges can be an important feature in motion analysis and need to be retained, therefore a simple convolution filter is required that suppresses the noise but retains edge information.

4.2.1 Median Filter.

Median filters are effective at removing noise from an image. The median filter retains image details such as edges. This is due to the fact that unlike general spatial low pass filters that convolve a weighted mask with the image pixels to obtain a new pixel value, the median filter works on the spatial areas of the image by sorting the image pixels into an ascending order by grey level value. The value in the middle of the sorted grey levels is used to replace the centre pixel in the local neighbourhood of the image.

The acquisition and motion detection control process triggers the image filtering process, which applies a 3 by 3 median filter to the image. Upon completion of the filtering operation the image filtering process returns the filter status signal to the acquisition and motion detection control process which indicates that the image filtering is complete. Figures 4.3(a) and 4.3(b) show an original image input (intensity image) as supplied by the image sensor to the system and the results of applying the median filter to the image (filtered intensity image).

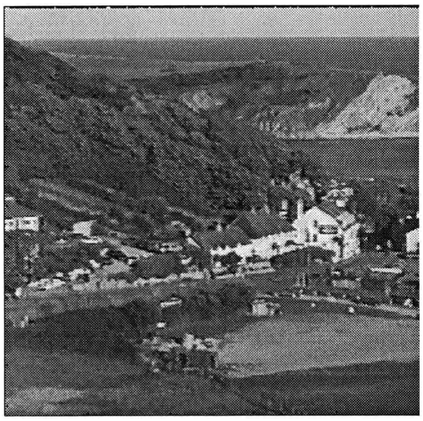


Figure 4.3(a) Original image image (intensity image).

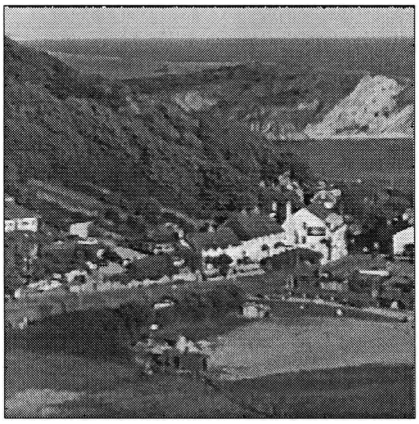


Figure 4.3(b) Median filtered (filtered intensity image).

4.3 Motion Detection.

With the open world scene that is of interest here (figure 2.1) there is a wide field of view. This creates problems with variations in intensity of individual pixels due to illumination change. In addition to this problem there can be apparent motion in the scene due to the grass, bushes or trees moving in the wind, shadows from clouds and movement by the camera. These problems all contribute to making areas in the image appear to be in motion, when to all intents and purposes they are either static background objects (grass, bushes, trees etc) or do not exist (cloud shadows, camera movement). To alleviate these problems a pyramid has been adopted. Levels in the pyramid are created by simply averaging square pixel regions of different sizes. This effectively smoothes out small pixel variations (false motion cues).

However this smoothing processes reduces the image resolution by a factor 2^n where n is the size of the square pixel region that will be averaged. This averaging process affects the distance at which objects of interest can be detected (resolution). Experiments were carried out to determine which level in a pyramid of averaged square pixel regions would give best performance at rejecting false motion cues whilst detecting actual motion cues. Square pixel areas of 2 by 2, 4 by 4 and 8 by 8 were used. Figure 4.4 below shows the actual number of motion cues generated for an input sequence of open world images.

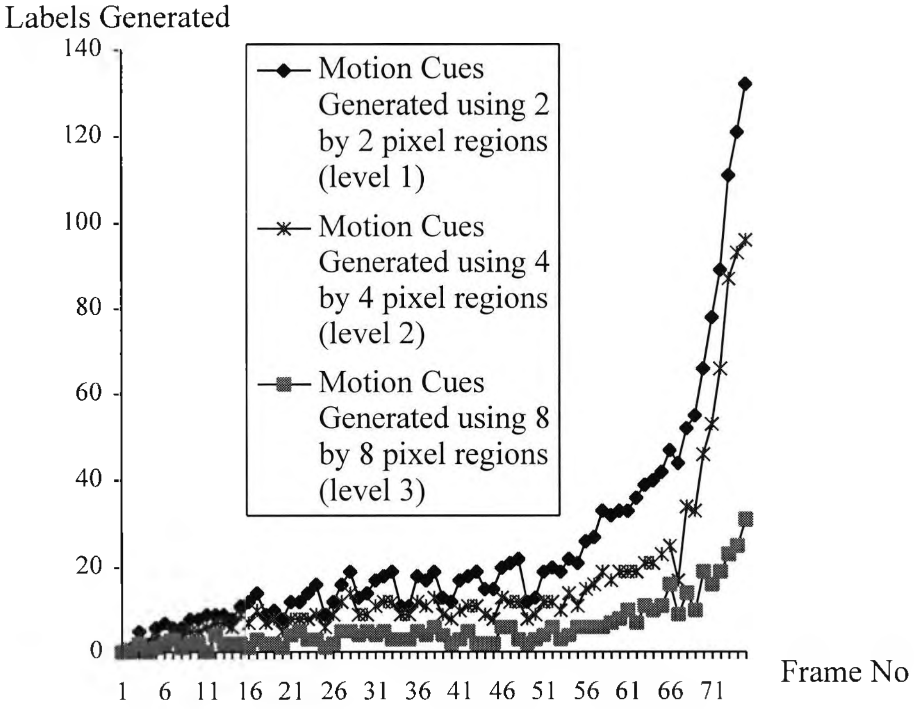


Figure 4.4 Plot showing the total number of motion cues found per frame using an open world image sequence for 2 by 2, 4 by 4 and 8 by 8 square pixel regions.

From figure 4.4 we can see that the number of false motion cues generated was reduced the larger the area of pixel averaging. From this we can conclude that an 8 by 8 pixel region should be used. However it was detected visually that motion cues were lost when vehicles travelled down the road into the cove. In fact at this level in the pyramid only vehicles on the entry\exit road were detected. This is due to the loss of resolution caused by the averaging process. The 2 by 2 pixel region generated motion cues for all vehicles observed moving in the scene. But the number of false motion cues generated by this level in the pyramid (grass and bushes moving in the foreground) would result in a large number of false motion cues having to be processed.

The 4 by 4 pixel region provided a compromise between the number of false motion cues generated and the range from the camera that vehicle motion could still be detected. By observation of the motion cue sequence, vehicles could still be detected moving within the image at this resolution, whilst the number of false motion cues was reduced by approximately 30% across the 75 frames of the input image sequence. From figure 4.4 level 2 in the pyramid gives the best compromise between false motion cue generation and target resolution, therefore only level 2 in the pyramid will be used.

Using a pyramid structure and a frame differencing technique for motion detection, identifies two distinct data process, namely, statistical analysis (generation of level two in the pyramid) and motion cue generation. The motion detection process was therefore broken down into a third level DFD, which is shown on the next page in figure 4.5.

When triggered by the ‘acquisition and motion detection control’, the 'motion detection control' triggers the 'image statistics process' this calculates the mean and standard deviation of fixed four by four pixel regions in the filtered intensity image. A 4 by 4 pixel region defines a single statistical image tile. Upon completion of the statistical analysis the image statistics process returns a statistical status control signal to the motion detection control process to indicate that the statistical processing is complete.

The motion detection control then enables the 'image motion cues process'. This process uses a standard t-test, [32] to identify significant changes in the grey level statistics between the reference frame of image statistics and the current frame of image statistics. Regions of the image where there is no significant difference (null hypothesis) shows that there was no motion in that region. However regions that show significant difference (alternative hypothesis) are areas where motion may have occurred. The hypothesis testing effectively forms a statistical difference map which defines regions in the image that may contain motion.

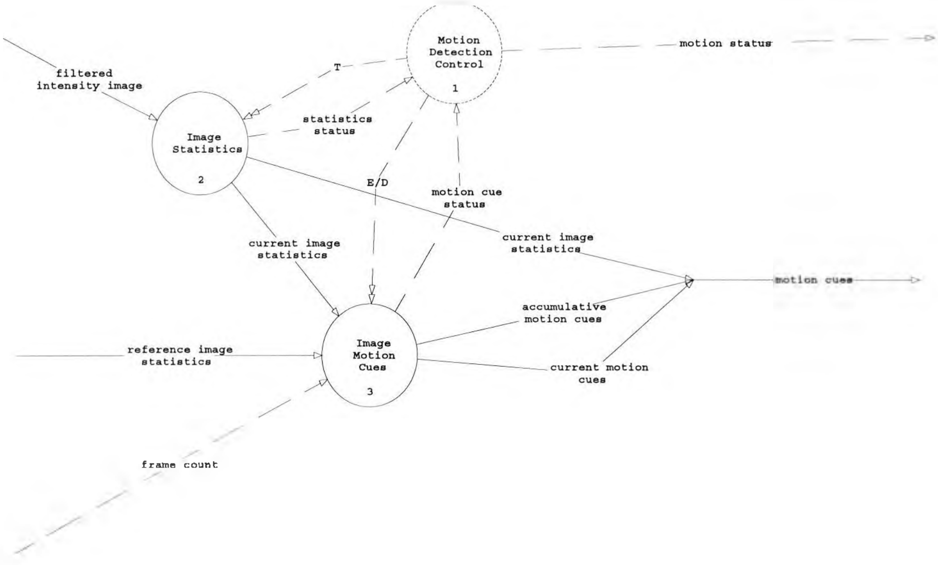


Figure 4.5 Level 3 DFD for the motion detection process.

4.3.1 Image Statistics.

The image statistics process calculates the mean and standard deviation for each image tile in the current input image frame based on groups of 16 pixels. These groups are arranged as 4 by 4 non-overlapping regions (image tile). For each region its mean μ_t and standard deviation σ_t are calculated using 4.2 and 4.3 respectively.

tile mean μ_t

$$\mu_t(x,y) = \frac{1}{16} \times \sum_{\Delta x=1}^{4} \sum_{\Delta y=1}^{4} P(x + \Delta x, y + \Delta y) \qquad (4.2)$$

tile standard deviation σ_t

$$\sigma_t(x,y) = \sqrt{1/16 \times \sum_{\Delta x=1}^{4} \sum_{\Delta y=1}^{4} (P(x + \Delta x, y + \Delta y) - \mu_t)^2} \qquad (4.3)$$

where

P(x,y) is pixel magnitude at position (x,y) in the image

The statistical analysis generates the second level in the pyramid. This is shown diagramatically in figure 4.6 on the next page.

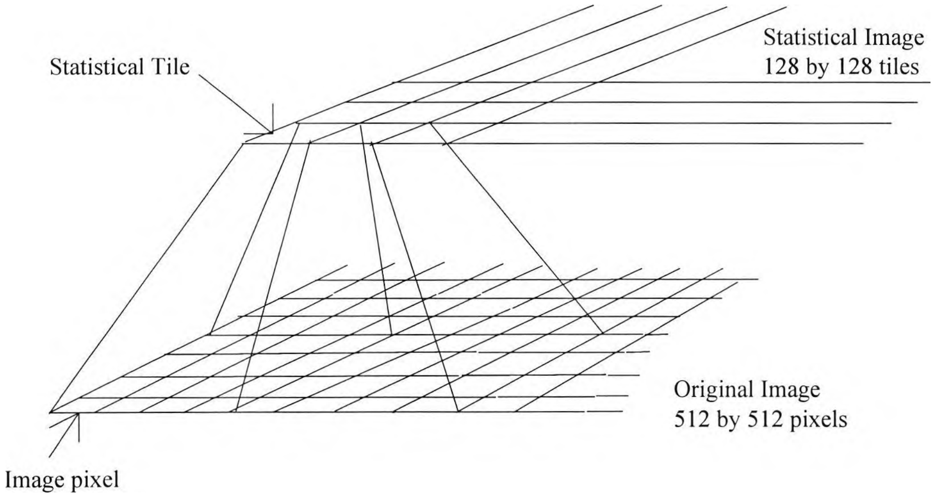


Figure 4.6 Statistical Image Tile Plane.

The statistical technique reduces the resolution of the image from 512 by 512 pixels to 128 by 128 tiles. This reduction in resolution (a factor of 16) can be tolerated as objects of interest moving within the scene including object orientation changes have been found to be still detectable at the required distances, (Teal et al, [4]). Objects of interest occupy several or more statistical tiles depending on their exact distance and orientation from the camera. The reduction in resolution becomes a problem for objects moving at distances further away from the camera, typically ranges above 500 meters.

4.3.2 Image Motion Cues.

One of the simplest techniques for detecting the changes between consecutive frames of image data is to use a difference image (Jain ,[33]). If $f(x, y, t_i)$ is a reference frame of image data taken at a time t_i and $f(x, y, t_j)$ is a frame of image data at some time t_j, where t_j is later than t_i then a difference image may be defined by

$$d_{ij}(x,y) = \begin{cases} 1 \text{ if } |f(x, y, t_i) - f(x, y, t_j)| > \theta, \\ 0 \text{ otherwise} \end{cases} \qquad (4.4)$$

where θ is a pre-set threshold.

The difference $d_{ij}(x,y)$ has a value 1 at spatial co-ordinates (x,y) only if the grey level difference between the two images is above the threshold. This form of motion detection relies on the illumination between images remaining relatively constant within the limits set by the threshold θ. In an open world natural scene there will be motion value entries in $d_{ij}(x,y)$ as a result of noise and illumination changes in the image. The statistical analysis however provides a set of statistics for the image tiles, where the mean represents spatially smoothed areas of the original image. This averaging process reduces noise due to variation in pixel intensities at the cost of resolution of objects to be identified and tracked. The differences between two frames of statistical tile data $d_{ij}(x,y)$ can be determined using a standard t-test, [32]. The t-test identifies tiles that have no differences (the NULL hypothesis) or tiles that have significant differences in their grey level statistics (the ALTERNATIVE hypothesis).

Consider:

$$\sigma^2 = ((n_1 - 1)*\sigma_{ref}^2 + (n_2 - 1)*\sigma_i^2)/(n_1 + n_2 - 2) \qquad (4.5)$$

where

σ^2 is the 'pooled estimate of variance'
$n_l = n_2 = 16$ pixels (size of 1 image tile)
σ_{ref}^2 is the variance of a reference image tile
σ_i^2 is the variance of a current image tile

and

$n_l + n_2 - 2 = 30$ degrees of freedom.

The variation in statistics is given by

$$\Delta = (\mu_{ti} - \mu_{tref})/\sigma\sqrt{(1/n_1+1/n_2)} \qquad (4.6)$$

where

μ_{ti} is the current tile mean.
μ_{tref} is the corresponding reference tile mean.

For a significance level of 5%, we can evaluate a test measure from the table of percentage points of the t-distribution:
then

$$d(x,y) = \begin{cases} 1 & \text{if } |\Delta| >= 2.042 \\ 0 & \text{otherwise} \end{cases} \qquad (4.7)$$

If the statistical difference image $d_{ij}(x,y)$ has tiles with a value 1 at spatial co-ordinates (x,y), they are considered to be due to the result of motion in the image. Tiles with a value 0 at spatial co-ordinates (x,y) are as a result of no detectable motion in the image. The resulting statistical difference image is now scanned in a raster scan order using an 8 region neighbourhood connectivity operator. Single instances of statistical tiles showing motion are removed from the statistical difference image. The results of the current image motion cue processing is added to a store of previous image motion cues found. This effectively builds an accumulated motion cue image (Jain, [35]) showing all areas of the input image sequence where motion has been detected. The current image statistics, the current motion cues and the accumulated motion cues form the motion cue data and are written into the acquisition and motion detection store. Upon completion the motion detection control generates a motion cue status signal to the acquisition and motion detection control indicating the completion of the motion detection process.

Figures 4.7(a-d) are four input frames from an image sequence show a car park scene (lower left hand side of the image). The car park has a single entry\exit road (left hand side of the image slightly above centre) which joins a main road (just above the centre of the image), the main road travels down into a cove (top left hand side of the image). A vehicle is seen leaving the car park and becoming occluded behind a hedge. The vehicle later re-emerges from behind the hedge, but it is still partially occluded.

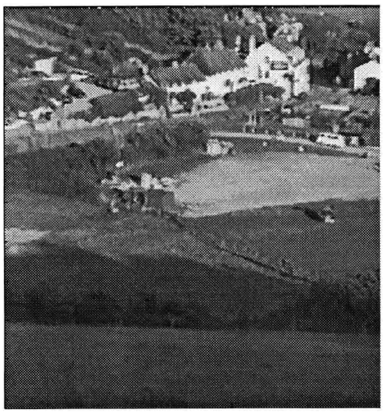


Figure 4.7(a) Input image frame 3.

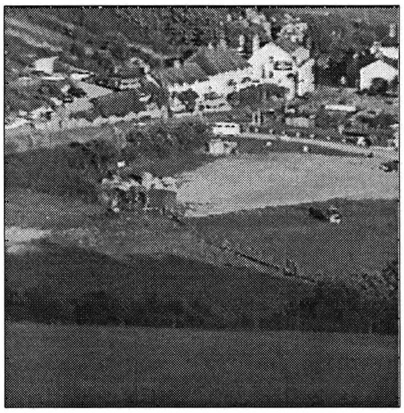


Figure 4.7(b) Input image frame 5.

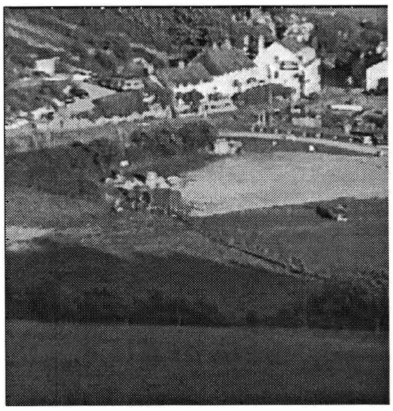


Figure 4.7(c) Input image frame 7.

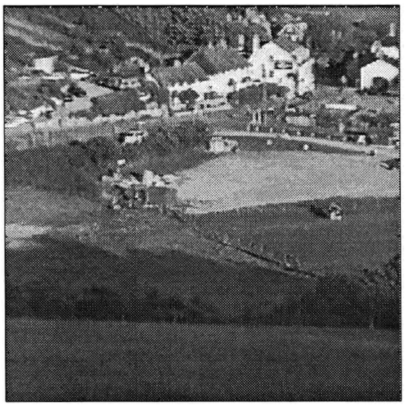


Figure 4.7(d) Input image frame 9.

Figures 4.8(a-d), show the motion cues generated by the motion detection process for the input image sequence shown in figures 4.7(a-d). Figure 4.9 shows the accumulated motion cues generated across this complete input image sequence, demonstrating that vehicles can still be tracked despite the statistical process reducing the image resolution by a factor of 16.

Figure 4.8(a) Motion cues frame 3.

Figure 4.8(b) Motion cues frame 5.

Figure 4.8(c) Motion cues frame 7.

Figure 4.8(d) Motion cues frame 9.

Figure 4.9 Accumulated motion cues.

4.4 Update Image Reference.

When using a static camera and a frame differencing technique to perform motion detection, there are two important requirements that such systems must meet. The first is that there must be no camera motion between consecutive frames of image data at any time, the second is the generation of the background image data that will be used as the reference against which current images will be compared. It can be difficult to ensure that the camera remains static at all times and undesired motion arising from the camera moving between consecutive image frames usually occurs due to a wind or ground disturbance. Disturbances from either source generate large numbers of differences between the current frame of image data and the reference frame of image data, leading to an excessive number of objects to analyse.

To limit the effect of these problems, the maximum number of objects found per frame by the 'target identification and tracking process' (chapter 5) is limited to 64, with any number of objects greater than this being removed. If the target identification and tracking has to limit the number of objects found in a frame, then that frame number and the object count for that frame are stored as an error vector in the ‘Initial Object Description Table’ (table 5.2) calculated by the ‘initial target identification process’. If the system has to continually limit the number of objects found per frame, then this could be an indication that the background reference image data is no longer a true representation of the background of the image. This could perhaps be due to fluctuation of the light intensity level in the scene. Alternatively the camera may be in motion due to one or more of the highlighted disturbances. The motion of the camera could have been caused either in this frame or when the reference image was generated. If the camera motion occurred in this frame, then the disturbance will probably be temporary, which can be detected and compensated for.

If excessive motion is occurring every frame or nearly every frame, then that motion is probably due to the camera being in motion when the reference image was generated, in which case the reference image needs to be updated. The system uses multi-level reference images for the motion detection and initial target identification processes. The filtered intensity image is used to generate reference statistics and reference edge data (section 4.5). The motion detection process uses differences between the reference image statistics and the current image statistics to determine if there was any perceived motion in the image. These motion cues are used to focus the initial target identification process which is performing an initial identification of the motion found in the image. The reference filtered intensity image is therefore essential to both the motion detection and identification processes.

The selection of a reference image for motion detection using a frame differencing technique has been widely researched with several algorithms for selecting or determining the background reference image being developed (Long et al, [37] Borofferio [38], Rosin et al [47]). These factors make the process of selecting the reference image difficult. Building up a stationary background image (Long et al, [37]) on a frame by frame basis for use as the reference image would probably be prone to error. This would be due to the wide variation of pixel intensities both spatially and temporally that will be encountered from the above sources.

The motion detection process calculates image statistics for four by four pixel regions (tiles) in the image. A simple neighbourhood operator raster scans the statistical difference image removing small instances of tile differences reducing the amount of apparent motion in the image caused by small local disturbances, pixel errors, etc, (the statistical operation effectively smoothes the image). Rather than building up a background image from pixels averaged over time that have not displayed any motion, a classification process has been developed. The classification process classifies the results of a statistical analysis of the perceived motion in the image to determine if a new reference image is required.

4.4.1 Update Reference Selection Criteria.

The reference selection criteria is just another term for thresholding, that is, some processed characteristic of the image is going to be compared to one or more pre-determined values and a decision taken based on the results of that comparison. Thresholding can be considered as a classification problem (Kittler et al, [36]), and the reference selection criteria is a classification problem where the analysis of the motion in the image needs to be classified and used to determine if a new reference image is needed. The update image reference process is broken down into two further data processes, this is shown on the next page in a level 3 DFD.

Upon receiving an enable signal from the 'acquisition and motion detection control' the 'update image reference control' triggers the 'motion analysis process'. An initial target analysis process (chapter 5, section 5.2). calculates a set of object descriptors and performs edge analysis on each motion cue found in each frame. Based on the results of this analysis, the motion cues are labelled as either targets (vehicles) or objects (anything not deemed to be a vehicle).

The motion analysis takes the data generated by the initial target analysis process and produces a set of motion statistics based on the areas and distribution of the image occupied by moving objects across a window of processed image frames. Upon completion of its processing the motion analysis generates a motion analysis status signal to the update image reference control process, indicating the status of the processing carried out by the motion analysis.

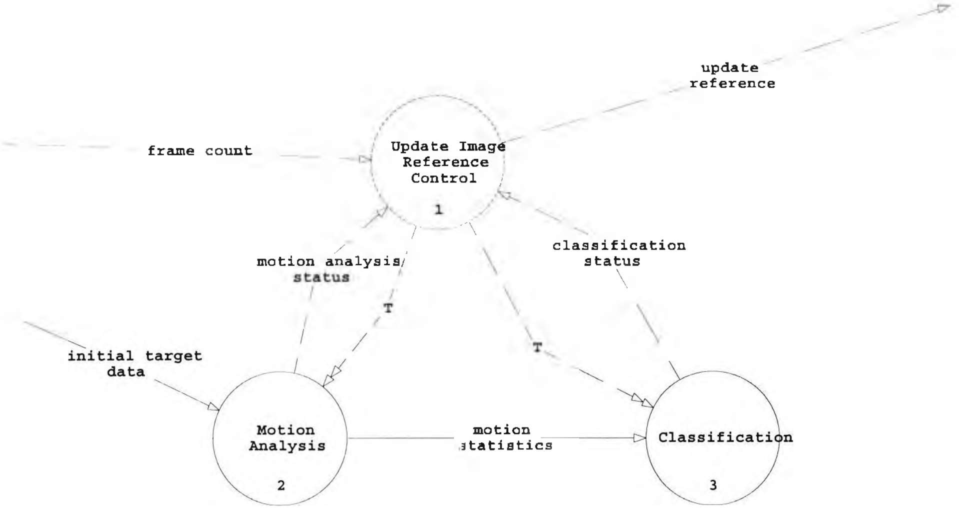


Figure 4.10 Level 3 DFD for the update reference process.

The motion statistics data is passed to the ‘classification process’ which classifies the data based on type and area of objects and targets found moving within the processed image window. It passes a classification status signal to the image reference control process indicating the status of the processing carried out by the classification process. This information is passed back to the acquisition and motion detection control process by the update reference signal.

This signal controls the path through the acquisition and motion detection control process (state transition diagram shown in figure 4.2). If new reference image data is required (update image reference is true), then the current filtered intensity image is taken as the new reference intensity image and new reference data generated from that image. If excessive motion still continues, then either the camera is unsteady or a large number of objects are moving in the image. If the excessive motion continues to occur despite corrective action being taken (three reference images generated one after the other), the ‘goal processor’ flags an error to the user via the ‘graphics user interface’ and awaits a user response.

4.4.1.1 Motion Analysis.

The ‘initial target analysis process’ (chapter 5) performs an initial identification of the motion cues in the image producing an initial object description of a rectangular window that encompasses the entire region of interest (the motion cue) every frame. These results are written into the initial object description table (chapter 5, table 5.2) on a frame by frame basis, which effectively forms a history of the detected motion in the image. Statistical analysis is performed across a sliding window of the detected motion as shown in figure 4.11. The following statistical parameters are calculated across the sliding window:

1:- The Window Mean u_{win}.
2:- The Window Median $\mathrm{MED}_{\mathrm{win}}$.
3:- The Window Mean Target Area u_{tawin}.
4:- The Window Object Motion Area u_{oawin}.

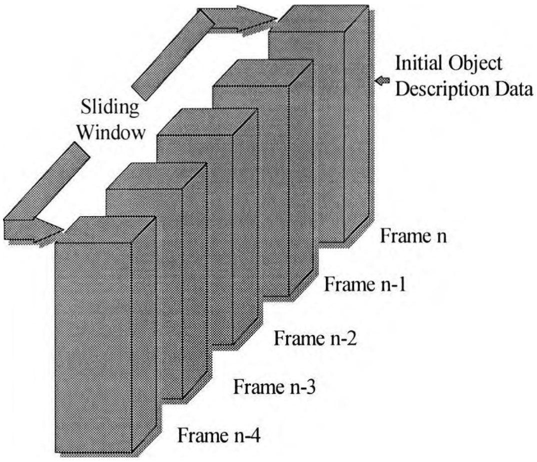


Figure 4 .11 Sliding window.

Having performed the statistical analysis across a sliding window of the perceived motion in the image, criteria have to be developed on which some form of classification of that perceived motion can be made. If we were analysing a scene that contained a motorway for instance, we could expect there to be a large number of vehicles moving. It could also be argued that the motorway would restrict the area in the image where these objects would be moving. In this case parameters could be determined that allowed large numbers of objects to be moving, but within restricted areas of the image. However the natural open world scene as depicted in figure 2.1, would require different parameters for the numbers of objects and their position in the image.

There are of course many more situations than these two extremes and ideally the system would learn what particular situation it is in by the distribution and occurrence of motion in the image. However this research is aimed at the open world scene, where the objects to be observed could be in any region of the image. It can be argued that it is unlikely that in these scene's there will be a large number of man made objects moving (greater than 64) as small country roads tend not to have large amounts of traffic on them. The man made objects to be tracked are expected to be several hundred meters from the camera and hence the areas of individual motion cues should be relatively small.

If then there are any large areas of the image displaying motion, then these motion cues could indicate that the background reference image is out-of-date, (the illumination level has changed for instance) and that a new reference image must be generated. These situations led to an experiment (section 4.3) where the number and size of motion cues were observed across the image sequence. From the experiment an initial estimate on the maximum number of objects (motion cues) that should be in the image at any one time was set to be 64. Further an estimate of the amount of an image area associated with that motion should be less than 20% of the total image area. These values were found to give acceptable system performance for the test sequence input to the system.

4.4.1.2 Classification.

The basic idea behind classification is to recognise objects based on a set of measured object features. Commonly used classification techniques such as the 'Bayesian Classifier' or 'Nearest Neighbour Classifiers' assume that 'N' features have been detected in the image and that these features have been normalised so that they can be represented in some form of parameter space. The nearest neighbour classifier determines the class of an object by computing its distance from points representing each class in the feature space and assign the nearest class. The distance measure is usually some form of Euclidean or weighted combination of features and this type of classifier can be effective for recognising objects when the distribution of the objects is straightforward. The Bayesian approach is based on the use of probabilistic knowledge about the features and frequency of the objects.

This approach means that knowledge of the conditional probability of an object belonging to a class j given a feature value of x is $p(x/w_j)$ and this is known *a prori*.

Based on this knowledge the *a posteriori* probability p(w_j/x) can be calculated for the unknown object belonging to class j.

Using Bayes rule, this probability is given by

$$P(w_j/x)=\frac{p(x/w_j)P(w_j)}{p(x)} \tag{4.8}$$

where

$$p(x)=\sum_{j=1}^{N} p(x/w_j)P(w_j)$$

The unknown object is assigned to the class with the highest *a posteriori* probability P(w_j/x). However in this application *a priori* knowledge about the feature probabilities and the class probabilities is not available and the distribution of object features is unlikely to be straightforward, making both of these classical classification process unsuitable.

There are two basic matching techniques that can be used to classify an object, the first of these is ‘feature matching’. Here the object class is represented by a set of features, which are compared to a set of model features using an absolute difference or Euclidean type technique. The result of this comparison is then weighted by the relative importance of the feature. These are summed over all features and the object is labelled with the model giving the highest sum. The second method for matching features is ‘symbolic matching’. Here the object to be classified is not only represented by its features but also by relationships among its features and the object is usually represented in a graphical form. Each feature in the object is a node in the graph and the object classification problem then becomes a graph matching problem.

The classifier developed here uses a modified form ‘feature matching’ to make the classification. The extracted motion statistics calculated in 4.4.1.1 are used to traverse through a set of conditions and the classifier makes it decision based on the path traversed. Figure 4.12 on the next page shows how the classification process is implemented using a flowchart.

The classification process is triggered by the update reference image control process after the completion of the motion analysis. This trigger is effectively a function call. The mean and median values across the sliding window were set to 64. The mean area threshold was set to 20 percent of the total image area. Figure 4.13 shows the rate of cue detection over the entire input frame sequence. At frame 75 the classification process updates the image reference with the current image frame (frame 75) and it is observed that the number of motion cues perceived in the image was reduced from 96 down to 2.

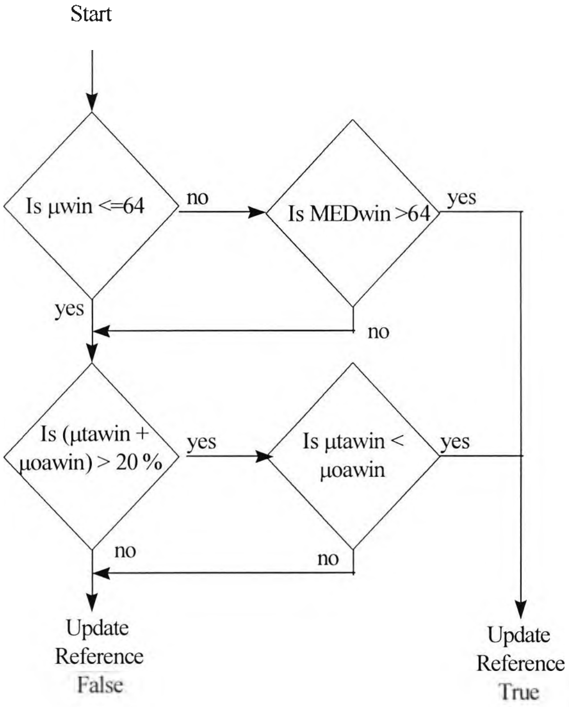


Figure 4.12 Flowchart representation of the classification process.

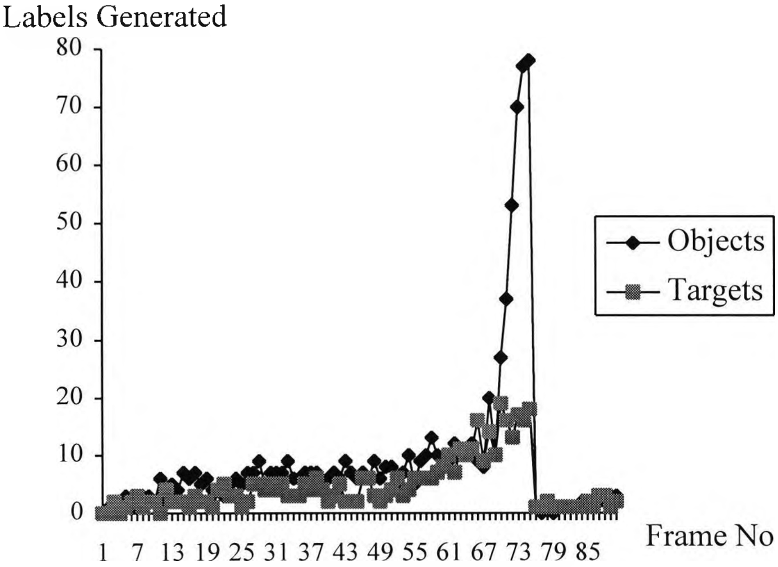


Figure 4.13 Plot showing the total number of object and target cues found per frame across the input image sequence.

4.5 Image Reference Generator.

There are two sets of reference data required by the system, the first is the reference statistics (mean and standard deviation of fixed four by four square regions) for the motion cue generation process and secondly a reference edge image is required by the initial target analysis process (chapter 5). The 'image reference generator process' generates new reference data for the system when enabled by the acquisition and motion detection control process.

A level 3 DFD for the image reference generator process is shown below in figure 4.14. Upon receipt of an enable signal, the 'reference generator control process' triggers the 'reference edge data process'. The reference edge data process applies a Marr-Hildreth edge operator to the current filtered intensity image which produces the reference edge data. On completion of its processing, the reference edge data process generates a reference edge status signal to the reference generator control which triggers the 'reference statistics process'. This applies the statistical operators described in section 4.3.1 to the current filtered intensity image to produce the reference image statistics.

Upon completion the reference statistics generates a reference statistics status signal to the reference generator control which then returns the reference status signal to the acquisition and motion detection control process. All reference image data is written to the 'acquisition, motion and reference image data store'.

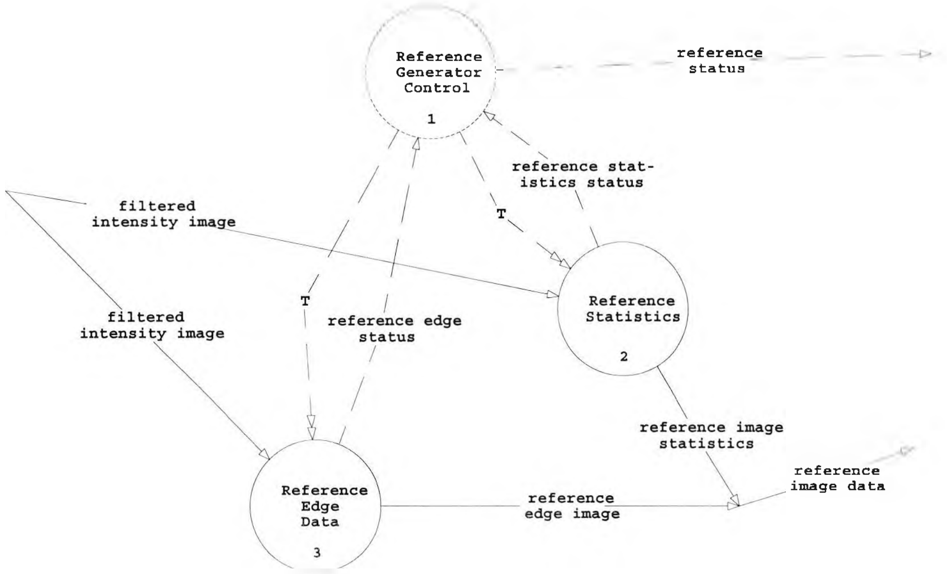


Figure 4.14 Image reference generator Level 3 DFD.

4.5.1 Reference Statistics.

The reference statistics are generated using the image tile technique described in section 4.3.1, using equations 4.3 and equation 4.4. The results of the statistical processing are written to the acquisition, motion detection and reference image data store.

4.5.2 Reference Edge Data.

The reference edge data process uses a Marr-Hildreth edge operator to perform the edge detection. The Marr-Hildreth is a common and well known edge operator, (Marr & Hildreth, [34]). The image is first smoothed by applying a Gaussian function, (the Gaussian is unique in that it has a minimal bandwidth frequency product), the Gaussian filtered image is then convolved with the Laplacian operator. These two operators are usually combined to form a single convolution mask which is convolved with the image. If there is an edge in the image or a sharp change in the intensity, this gives rise to a zero crossing in the convolved image. Detection of the zero crossings enables a edge map to be determined.

The Marr-Hildreth operator can be defined by

$$\nabla^2 G(x,y) \otimes P(x,y) \tag{4.9}$$

Where

P(x,y) is the original image.

$\otimes$ is the convolution operator.

$\nabla^2 G(x,y) = \frac{-1}{\pi\sigma^4}(1 - (x^2 + y^2/2\sigma^2)\exp^{-(x^2+y^2/2\sigma^2)})$

and

∇^2 is the Laplacian operator.

σ is the standard deviation of the Gaussian filter and is proportional to the size of the neighbourhood on which the filter operates.

x,y are image co-ordinates.

The Marr-Hildreth edge operator is applied to the entire image. The results of the edge detection process is written to the acquisition, motion detection and reference image data store.

4.6 Discussion and Summary.

The image acquisition and motion detection process provides all the low level image processing functions necessary to reduce the noise added to the image by the digitisation process and detect object motion in the image. Rather than just applying a simple spatial convolution operator to remove noise introduced into the image by the digitisation process, a median filter was used as this retains image details such as edges, which simple spatial convolution operators tend to smooth out.

The statistical analysis of the pixels within four by four pixel regions (tiles) yields the mean and standard deviation of the pixel values within those tiles. This statistical analysis effectively smoothes those areas and removes most of the false motion cues generated by pixel differences not due to object motion. The statistical differencing technique demonstrated that sufficient resolution remained to generate motion cues and identify regions of interest where objects may be moving in the image irrespective of the object orientation.

Figures 4.6(a) to 4.6(d) show a sequence of images where the vehicle goes through orientation changes and occlusion as it move across the image. Figures 4.7(a) to 4.7(d) show the results of the motion cue generation. Despite the orientation changes and occlusion, the vehicle is still detected across the sequence by the motion cue generator.

The reference images required for the frame differencing technique are computed by the reference generation process. This process generates not only the reference image statistics but also a reference edge image that is used later by the initial target identification process (chapter 5). The reference selection criteria for this process is not based on grey-level histograms (Otsu, [39]), but applies a set of rules to statistical parameters calculated across a window of the detected object motion.

The update criteria is based on the fact that with open world scenes, if an excessive number of objects is moving or if large areas of the image are showing motion, we need to determine if this motion has been generated by genuine objects moving in the scene, the camera moving or a change in the illumination condition across the scene. If there appears to be an excessive number of objects moving in the image, then the reference selection criteria identifies whether that motion is from a large number of objects moving (high mean and a high median value) or if the camera moved between frames (high mean, low median).

If the camera moved between frames, a large number of objects appear to be in motion in the image and this detected motion will increase the mean across the window. However this motion is unlikely to affect the median motion value. This fact can be used to classify the detected motion as camera motion (it appears only as a temporary disturbance). If the median and the mean have high values then this indicates that a large amount of object motion has been observed across a number of consecutive image frames.

This detected motion can either be due to a considerable amount of object motion (not very likely with an open world country scene), the camera is continuously in motion (strong winds) or the reference data is out of date. In this case the selected course of action is to update the reference. If this selected course of action does not reduce the amount of apparent motion in the image, then this condition can be recognised and an error flagged to the user.

If the image is showing a large area of motion or the total motion detected across the image in that frame is large, (> 20 %), then the update reference rules identify the contributing factors from that motion. For example, is it a vehicle close to the camera or is it a large cloud moving across the scene. If it is a cloud moving in front of the sun and causing a general illumination change across the whole scene then the large area showing motion in the image will be identified as an object and not a target, (illumination change should have little effect on the edges detected in the image, i.e. no structural change within that region). However if it was a vehicle close to the camera, then the structure of the vehicle (being man made and smooth with straight edges) reduces the edginess in the image for the region showing motion. These conditions are detected by the update reference criteria rules, and for the illumination change condition, new reference image data is generated from the current image.

A real world image sequence of over 90 frames was input to the system. The initial target analysis (chapter 5) is analysing the perceived motion in this sequence. At frame 75 based on this analysis the classification process determined that a new reference image was required. New reference data was generated based on the current image frame. Updating the reference at this point reduced the perceived motion in the image from 96 objects down to 2.

This simple classification process has demonstrated that good results can be achieved based on a few simple assumptions about the number and size of objects expected to be observed moving in the image. Frame differencing techniques in open world scenes are used extensively for detection of both man made objects (Malik, [48]) and people (Rosin, [47]). Though both applications were with open world scenes, Malik was detecting vehicle movement on motorways and Rosin, individuals who are 50 to 100 meters from the camera.

The acquisition and motion detection process developed here is for detecting man made objects moving in scenes which will have a large amount of background clutter. To this end the system was tested on real world image sequences, where the objects moving within the scene became both fully and partially occluded and made manoeuvres that change the orientation of the object with respect to the camera. These manoeuvres were man made and non-predictable. The motion detection process has demonstrated that it can generate motion cues for these objects. Figures 4.7(a-d) show a van moving across the scene. It becomes first partially and then fully occluded, making manoeuvres that change its orientation with respect to the camera. Despite this, the motion detection process continued to generate region of interest cues for this object.

The loss of spatial resolution as a result of applying the statistical analysis process is thus deemed not to impact on the performance of the system for vehicle detection and tracking in this application. The motion detection process in conjunction with the classification process provided a method for motion cue detection of man made objects moving in natural open world scenes.

Chapter 5 Target Identification and Tracking.

5.1 Introduction and Overview.

Tracking man made objects moving in an open world scene is a complex task. The motion of these objects makes their outline variable and hence these objects are difficult to model in conventional terms. The problem of identifying and tracking objects moving in natural open world scenes has received considerable attention in the literature, with model based vision techniques outlined in chapter 2 (Koller [17], Worral [2]) and more recently Ferryman [67], Worral [68].

In these papers they are concerned with trying to fit an *a proiri* geometric representation of the object (a parameterised model) to some form of extracted image feature, (typically edges) on a frame by frame basis and thus tracking the object across the image sequence. If however the object to be tracked occupies only a small proportion of the image, then the image data will not contain sufficient resolution to reliably extract the geometric features necessary for model based methods. One of the main objectives of our research is to develop an identification and tracking algorithm that is able to recognise and track man made objects (vehicles) where the tracked object is expected to be a large distance from the camera (typically 400 meters).

Kollnig et al, [40] estimated the pose of a vehicle by directly fitting image gradients to polyhedral models of the vehicle. The system could still track vehicles that were partially occluded by textured objects and a large distance from the camera, (typically a vehicle occupied 50 by 30 pixels).

However the scene depicted was a road traffic scene and although this type of scene generates clutter, we can expect there to be more background clutter generated by a natural open world scene (trees, bushes, grass etc). In this type of scene, the vehicle orientation and motion are not so well constrained and the tracked object may also be capable of making unpredictable manoeuvres which may partially occlude it from the camera. These additional features considerably complicate the model matching process.

Clearly we need to move away from the approach of trying to fit some form of geometric model to individual object data extracted from the image and develop a new identification and tracking algorithm that will address these issues. The matching of object descriptors (Rosin, [3], Teal, [4]) has been found to be more robust for tracking objects where the tracked object only occupies a small proportion of the image pixels and may be viewed at any orientation. A new algorithm has been developed that does not rely on fitting specific geometric models to extracted image data, but uses a description of the 'edginess' of the object to provide an initial indication of the presence of the vehicle in the image and then an *a priori* estimation of the motion of the object to track it on a frame by frame basis.

A simple form of motion detection is to subtract the current frame of image data from an estimate of the background (Kilger, [41], Dubusson, [42]). The acquisition and motion detection process (chapter 4) uses the differences between a reference frame of image statistics and a current frame of image statistics to provide an estimate of moving areas (motion cues) in the image. Although this form of motion detection is relatively simple, many potential moving objects generated by this method may not be due to motion of objects of interest (vehicles) and these unintentional signals have to be identified and removed from the tracking process.

The first stage in the identification process is to take the individual areas of interest generated by the motion detection process (motion cues) and segment them into regions. For each of these regions a set of object descriptors is calculated and translated back into image co-ordinates that define a Region Of Interest (ROI) in the original image. An edge operator is applied to the regions of interest defined in the original image. The generated edge data gives a measure of the 'edginess' for that region. The natural open world scene consists of objects that do not have uniform shape or intensity and consequently an edge operator generates large quantities of edgel data for these types of objects.

Man made objects on the other hand consist mainly of straight line edges which usually do not occur in nature (Radford [13]) and the surfaces of man made objects (vehicle) tend to be smooth and uniform in intensity. Consequently this form of object gives fewer edge pixels. Given the region of interest in the original image and having generated a reference edge image (chapter 4), then a measure of edginess in the reference image for that ROI can also be calculated. Edgel analysis of the corresponding area in the current frame enables a measure of edginess for the region to be calculated in that frame. Edge detection operators are relatively invariant to changes in intensity; hence if the motion cue was generated by a cloud or noise in the image then the measure of edginess for that region should remain relatively unchanged.

However, if the motion cue was due to a vehicle, then the structure within that region is different and a different measure for the edginess for that region would be detected. This difference between the two regions edginess is used to give a further cue as to the identification of that region and a ‘target label’ is generated, indicating an initial identification of that region of interest as a target. The target label together with the object descriptors for the region are entered into an initial object description table (IODT).

If there is no detectable difference in the edginess in the region then an ‘object label’ is generated for that region and the object label together with the object descriptors for the region are also entered into the initial object description table, (an object that fails this initial edge analysis may still be a target). This initial identification is only another cue in the identification of a region of interest. The variation of intensity, disturbances in the field of view such as trees or bushes moving in the wind or partial occlusion of the vehicle, all affect this identification process.

The initial identification is repeated on a frame by frame basis. Target and object labels are generated for each region found in the image and object descriptors calculated for both. Vehicles exhibit known motion characteristics, and constraints (estimates) on that motion can be applied to all the entries in the IODT. The use of motion constraints enables a frame to frame correspondence between target labels and object labels to be assessed (Roberts [56], Zhang [57]).

Essentially there are four possible outcomes for this analysis; firstly all target labels that satisfy the motion constraints across a number of frames (which is a further cue to the identification of a region), are identified as targets (vehicles are considered to be targets) and displayed. Secondly, target labels which have not satisfied the constraints are re-labelled as objects (objects are anything else not considered to be a target). Thirdly, object labels that have satisfied the motion constraints (it is unlikely that a natural disturbance would move in the characteristic manner of a vehicle) are identified as targets, re-labelled and displayed. Lastly, objects that have failed to meet the motion constraints are removed from the tracking process and no longer displayed. The target identification and tracking process can therefore be broken down into two distinct processes, namely, 'initial target analysis' and 'target tracking'. From the first level DFD (figure 3.2) process 4 can be sub-divided into another DFD to reflect the processing requirements of the target identification and tracking. The level 2 data flow diagram for the target identification and tracking process is shown below in figure 5.1.

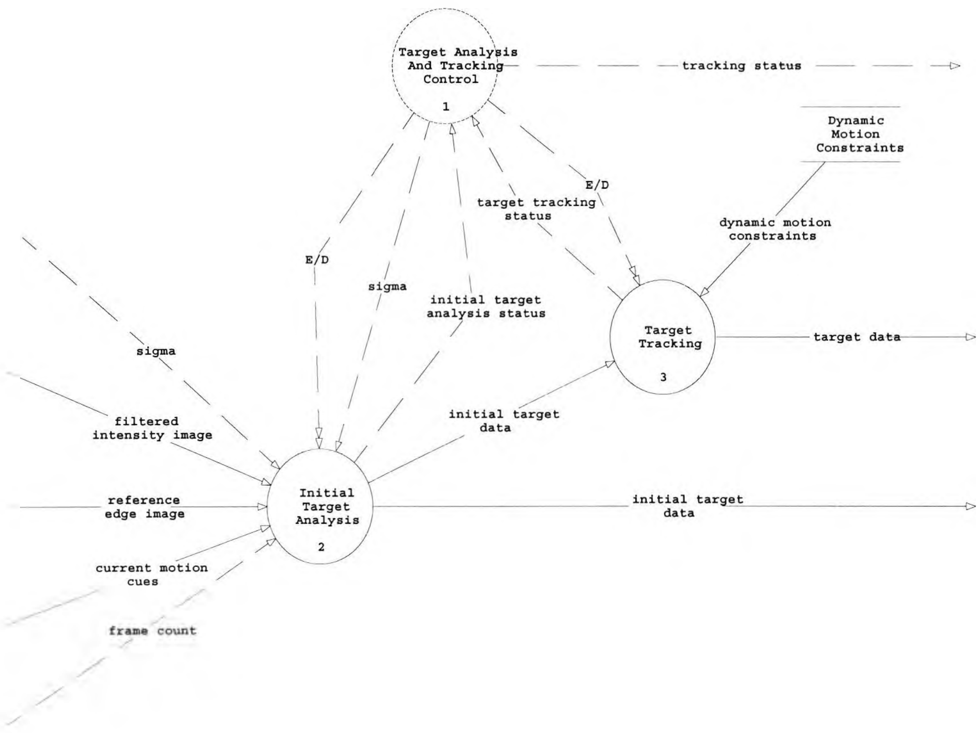


Figure 5.1 Level 2 DFD for the target identification and tracking process.

Upon receipt of an enable signal from the 'goal processor', the 'target analysis and tracking control process' enables the 'initial target analysis'. The initial target analysis uses the current motion cues generated by the image acquisition and motion detection process and segments the regions into objects and calculates a set of object descriptors for each region of interest. Next the initial target analysis performs an edge detection operation on the current filtered intensity image. Analysis of the difference between this edge image and a reference edge image (initially the first frame) generates target or object labels for those regions dependent on the results of the edge difference calculations. This information together with the object descriptors is written into the initial object description table.

Upon completion of its processing the initial target analysis generates a status signal to the target analysis and tracking control process. This signal then enables the ‘target tracking process’. The target tracking process determines the frame to frame correspondence between target and object labels using constraints on the permitted motion of these labels. The results of this processing identifies targets or objects that have moved in the scene to be identified as targets (target data). The target analysis and tracking control process now writes the target data to the ‘processed image data store’ and as such is available to all processing elements of the system.

Finally a tracking status signal is generated and sent to the goal processor indicating the status of the processing carried out this frame by the target identification and tracking process. Figure 5.2 below shows the state transition diagram for the ‘target analysis and tracking control process’. This diagram describes the dynamic behaviour of the target identification and tracking process. Both of the data transforms generate a status signal upon completion of their processing, these status signals are used to traverse the state transition diagram controlling the processing and flow of data through the initial identification and tracking process.

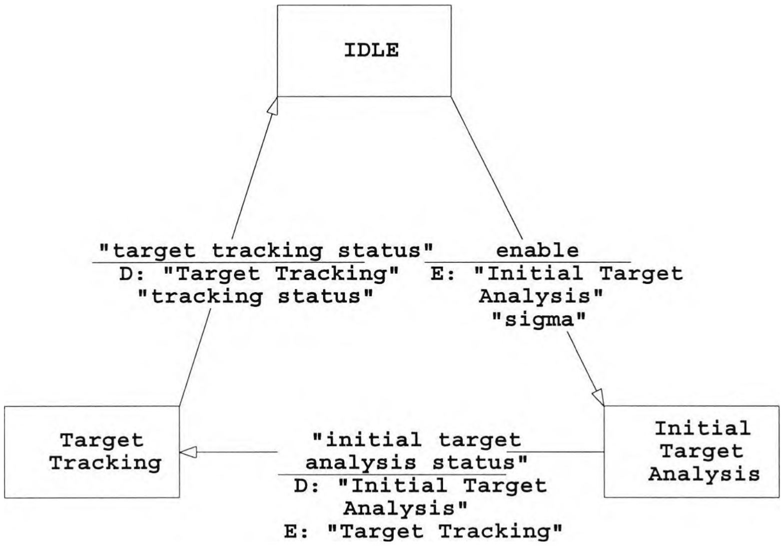


Figure 5.2 Target Identification and Tracking
State Transition Diagram.

5.2 Initial Target Analysis.

The ‘initial target analysis process’ provides region segmentation and analysis together with an initial target identification of regions of interest found by the image acquisition and motion detection process. This processing requirement is functionally decomposed into three further sub-processes, namely region segmentation and analysis, edge detection and initial target identification. These are shown below in figure 5.3.

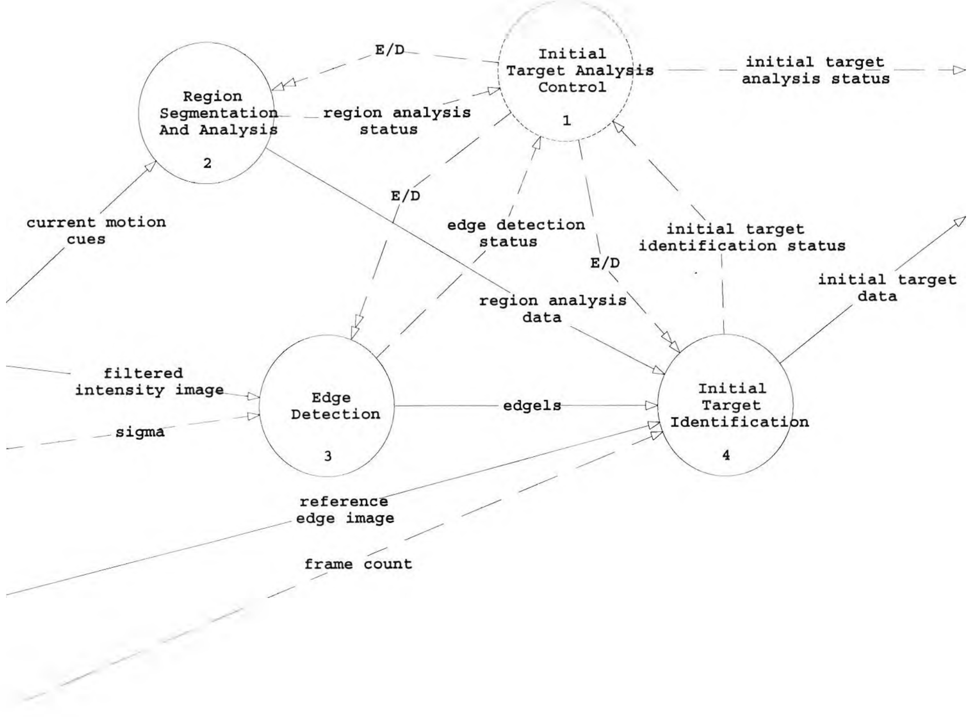


Figure 5.3 Level 3 DFD for the initial target analysis process.

When enabled by the ‘target analysis and tracking control process’, the ‘initial target analysis control’ enables the ‘region segmentation and analysis process’. This process segments (labels) each ROI found in the image and calculates a set of descriptors for each of them. Upon completion of its processing it generates a status signal back to the initial target analysis control indicating that its processing is complete. The initial target analysis control then enables the ‘edge detection process’, which performs edge detection on the current filtered intensity image for each ROI found in this frame. When the edge detection operation is complete a status signal is generated back to the control process to indicate that the edge detection process is complete.

Finally the ‘initial target identification process’ is enabled, which performs initial target analysis on each ROI in the current frame. Upon completion a status signal is returned to indicate completion of the initial target analysis. The results of region segmentation and analysis, edge detection and initial target analysis are combined to form the initial target data which is used by the target tracking process, but is also accessible by the rest of the system.

5.2.1 Region Segmentation And Analysis.

Region segmentation labels each region of interest found in the image with a unique integer identifier. Each of these identified regions has a set of object descriptors calculated for it. The region segmentation algorithm is based on an 8-neighbourhood connectivity mask using a two pass connected component analysis algorithm (Sonka et al [43]). Each region in the image displaying motion is labelled with a unique integer identifier. If we assume that a segmented image R consists of n disjoint regions R_i , then

$$\bigcup_{i=1,i\neq b}^{n} R_i = R_b , R^C \qquad (5.1)$$

where

R_b is considered to be background
R^c is the set complement

The segmentation and analysis employs a sequential approach to labelling the segmented image. An 8-neighbourhood connectivity mask is applied to the current motion cues by scanning through the motion cue data and labelling any non-zero value in the mask with an integer number and then increasing the integer value by one. This form of labelling suffers from 'label collision' [43], where regions within the mask have already been labelled and consequently have a non-zero value. To overcome this problem during the first pass of the region labelling, if a label collision occurs then this is detected and we store the two numbers as an equivalent label pair.

The label pairs are grouped to form an equivalent label pairs table. A second scan pass of the region data is then performed using the equivalent pairs table to re-label regions where a collision has occurred. The region labelling algorithm produces a segmented image with all regions in the image that may be a potential moving object labelled with a non-zero integer value. The region analysis algorithm calculates for each labelled region its :

(i) Area.
(ii) Centroid co-ordinates.
(iii) Minimum x,y and maximum x,y co-ordinates.

The area and centroid co-ordinates for an object are calculated using moments. All moment characteristics are dependent on the linear grey level transformations of regions. The moments of a digitised bounded image function of two variables can be defined by 5.2 [44].

$$Mpq = \sum_i \sum_j i^p j^q f(i,j) \qquad (5.2)$$

Where

f($_{i,j}$) = pixel magnitude at i,j
p and q define the order of the moment
i and j are pixel co-ordinates

However to describe region shape properties, the input image is put into a binary form where f($_{i,j}$) = 1 for a region pixel and f($_{i,j}$) = 0 for a none region pixel. This removes the dependence on the linear grey level transformation, thus we can re-write equation 5.2 as

$$Mpq = \sum_i \sum_j i^p j^q \qquad (5.3)$$

Using 5.3 we calculate the zeroth order moment as

$$M_{00} = \sum_{i=1}^{nop} \sum_{j=i}^{nop} i^p j^q \qquad (5.4)$$

where

nop is the number of tiles in a labelled object.

The zeroth order moment defines the area of the object and this moment can be used to normalise the higher order moments. The normalised first order moments M_{10} and M_{01} define the centroid of the object and can be calculated using 5.5 and 5.6.

$$\bar{i} = M_{10} / M_{00} \qquad (5.5)$$

$$\bar{j} = M_{01} / M_{00} \qquad (5.6)$$

Alternatively they may be calculated relative to the image origin (0,0) by using

$$M_{10} = \sum_{n=1}^{nob} x_n / M_{00} \qquad (5.7)$$

$$M_{01} = \sum_{n=1}^{nob} y_n / M_{00} \qquad (5.8)$$

where

nob is the number of boundary tiles.
x_n,y_n are the co-ordinates of each boundary point

The max and min x,y co-ordinates for each region in the segmented image is determined by scanning the segmented image in a raster scan order and noting the max and min x,y co-ordinates for each numbered region. This gives each region an x_max, x_min, y_max and y_min co-ordinate (tile co-ordinates). However the initial target analysis process examines the image edge structure. This requires the regions of interest to be image co-ordinates not tile co-ordinates (tile co-ordinate system has x,y values between 0 and 127, the image co-ordinate system has x,y values between 0 and 511). A simple translation from tile co-ordinates to image co-ordinates is achieved using equations 5.9, 5.10, 5.11 and 5.12 respectively.

$$\text{x\_max}' = (\text{x\_max}+1)*4+3 \tag{5.9}$$

$$\text{x\_min}' = (\text{x\_min}-1)*4 \tag{5.10}$$

$$\text{y\_max}' = (\text{y\_max}+1)*2048+1536 \tag{5.11}$$

$$\text{y\_min}' = (\text{y-min}-1)*2048 \tag{5.12}$$

The values for x_max', x_min', y_max' and y_min' provide a region of interest window in image co-ordinates that is one statistical tile larger (in both x and y directions) than the corresponding region found by the motion analysis. This takes into account the potential loss of object data in the original image caused by using fixed four by four pixel regions in the motion detection process. The object descriptors generated by the region segmentation and analysis together with the number of objects found, the object number and the region of interest co-ordinates are written into a region analysis table every frame. This process builds up a record (history) of the regional analysis results for any input sequence applied to the system. The table is shown below in table 5.1.

Frame No		No Objects
1		n
Object No	Object Descriptors	window co-ordinates
1	M_{001},M_{011},M_{101}	$\bar{x}\_\mathrm{max}_1$,$\bar{x}\_\mathrm{min}_1$,$\bar{y}\_\mathrm{max}_1$,$\bar{y}\_\mathrm{min}_1$
:	:	:
n	M_{00n},M_{01n},M_{10n}	$\bar{x}\_\mathrm{max}_n$,$\bar{x}\_\mathrm{min}_n$,$\bar{y}\_\mathrm{max}_n$,$\bar{y}\_\mathrm{min}_n$

:

:

Frame No		No Objects
N		n
Object No	Object Descriptors	window co-ordinates
1	M_{001},M_{011},M_{101}	$\bar{x}\_\mathrm{max}_1$,$\bar{x}\_\mathrm{min}_1$,$\bar{y}\_\mathrm{max}_1$,$\bar{y}\_\mathrm{min}_1$
:	:	:
n	M_{00n},M_{01n},M_{10n}	$\bar{x}\_\mathrm{max}_n$,$\bar{x}\_\mathrm{min}_n$,$\bar{y}\_\mathrm{max}_n$,$\bar{y}\_\mathrm{min}_n$

Table 5.1 Region Analysis Table.

5.2.2 Edge Detection.

The 'edge detection process' is performed using the Marr-Hildreth edge detector as described in chapter 4 section 4.5.2. The standard deviation of the filter is passed by the parameter sigma to the edge detection process. The entire image is processed by the edge detector operator and the resulting edge data is stored as the current edge image.

5.2.3 Initial Target Identification.

One of the problems already highlighted with using a frame differencing technique to perform motion detection of natural open world scenes, is that intensity differences can result from changes in illumination and scene conditions that are beyond our control (clouds, wind etc.). To further complicate the identification and tracking process, the object to be tracked is expected to be a large distance from the camera and may be viewed at any orientation. Other objects within the scene (grass, bushes, trees etc) produce a large amount of clutter making the identification of the region of interest more complex.

The initial target identification attempts to provide an indication that a region of interest is a potential target. Rather than attempting to fit some form of geometric model to edges extracted from the image (Kollnig et al [40], Worral et al [45]), a simpler edge measure is used. An edginess reference is calculated for the region of interest from the reference edge image and an edginess value is calculated for the same region in the current image. As already outlined, images which contain man made objects are assumed to consist mainly of straight line edges and their surfaces tend to be smooth and uniform in intensity, consequently giving fewer edge pixels (Radford [13]).

Therefore a difference between the two edge measurements could indicate the presence of a man made vehicle. This is a crude metric for the detection of a vehicle, but in cluttered scenes (bushes, trees, hedges etc) can give an initial indication to the presence of the vehicle. If the absolute size of this edge difference is more than a threshold Φ, then it can be postulated that the change in the number of edge pixels within the region is likely to be due to the presence of a man made object, rather than an illumination change for example as the edge detection process is fairly invariant to changes in illumination. The initial target identification can therefore be broken down into three further processes, namely, edgel analysis, edge difference analysis and initial object description.

5.2.3.1 Edgel Analysis.

The objects to be analysed will probably occupy small regions of the image and the edge information within these regions is likely to be very sparse (road feature) or conversely it could be very rich (trees, bushes, hedges etc). The edgel analysis has to calculate a reference edge pixel ratio and an object edge pixel ratio for each region found in the region analysis table (table 5.1).

These calculations are the same except for the fact that the reference edge pixel ratio calculation uses the reference edge image data and the object edge pixel ratio calculation uses the current edge pixel data. These calculations are performed using 5.13 and 5.14 respectively.

Reference Edge Pixel Ratio (REPR) for each region is given by

$$\mathrm{REPR} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} E_{ref}(i,j)}{(n*m)} \qquad (5.13)$$

and the Object Edge Pixel Ratio (OEPR) for each region is given by

$$\mathrm{OEPR} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} E_{o}(i,j)}{(n*m)} \qquad (5.14)$$

where

$\sum_{i=1}^{n}\sum_{j=1}^{m} E_{ref}(i,j)$ is the sum of edge pixels in the reference image ROI.

$\sum_{i=1}^{n}\sum_{j=1}^{m} E_{o}(i,j)$ is the sum of edge pixels in the current image region ROI.

where

m = x_max’- x_min’
n = y_max’- y_min’

5.2.3.2 Edge Difference Analysis.

The edge operation is fairly invariant to illumination changes, so differences in edge pixels within regions of interest should be due to structural changes within those regions. We calculate an edge difference ratio between the reference edge pixel ratio and the objects edge pixel ratio using

$$EDR = |OEPR-REPR|/REPR \tag{5.15}$$

and

```
IF EDR >= $\Phi$ THEN
        label := target;
ELSE
        label := object;
```

From the experiment carried out in chapter 4 (section 4.3) the edge difference ratio for all objects moving in the image were calculated. It was observed that for vehicles moving in the image the EDR averaged 10.77%. and that for false motion cues (motion cues that are not due to vehicle movement) average only 5.57 % difference. With the threshold Φ initially set to 10%, the initial target analysis correctly labelled 57.3% of the motion cues due generated by vehicle motion with a target label and 81.5% of the motion cues generated by non-vehicle motion were correctly labelled with an object label. The low value of correctly identified target labels was found to be due to the distance that vehicles were from the camera as they moved down into the cove. At these distances there is insufficient edge information for the initial target analysis to make decision. Figure 5.4 below shows a region of interest found from the input image sequence translated back into co-ordinates in the original image (chapter 4, figure 4.8(a) shows the actual motion cue generated). Figures 5.5(a), 5.5(b) and 5.5(c) show the region of interest enlarged, the edges found within that region and the edges found within the same region in the reference image respectively.

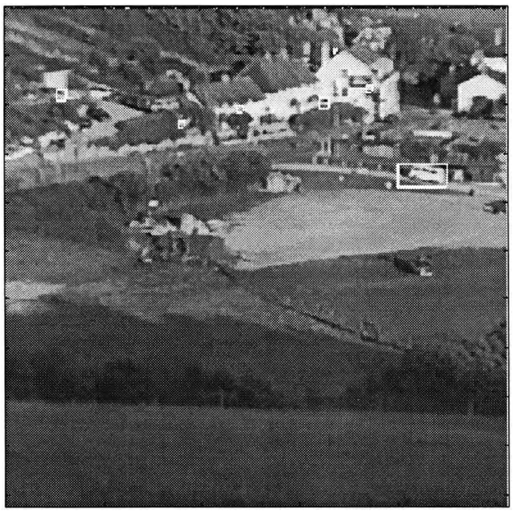


Figure 5.4 Region Of Interest (ROI) found in the current frame.

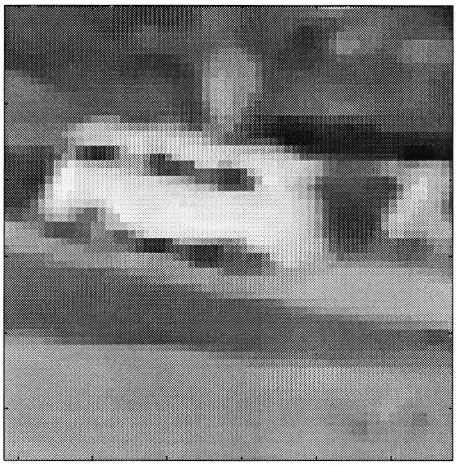


Figure 5.5(a) Enlarge region of interest.

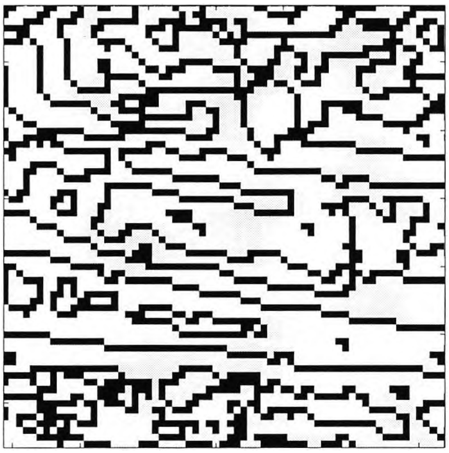


Figure 5.5(b) Edges in current image.

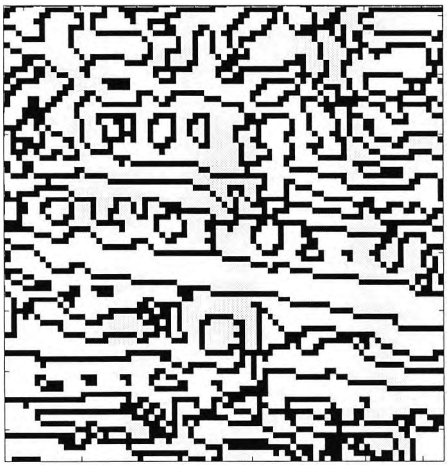


Figure 5.5(c) Edges in reference image.

5.2.3.3 Initial Object Description.

The final parameter calculated by the initial target analysis is a rectangular window that encompasses the entire region of interest (win_size). This can be easily calculated from the window co-ordinates. The results of the segmentation analysis, (the region analysis table) is combined with the initial target analysis results to form the initial target data. This data is written to the Initial Object Description Table (IODT), shown on the next page in table 5.2. This data is available for access by the rest of the system.

5.3 Target Tracking.

The initial identification process can only provide another cue to the identification of a region of interest in the image. This is due to the fact that orientation of a vehicle (viewpoint), partial occlusion of the vehicle and the distance of the vehicle from the camera all affect the edginess of a region and hence the initial identification process. The initial target identification process is repeated on a frame by frame basis, with target\object labels being generated for each region found in the image and each region has a set of object descriptors calculated for it.

Frame Number		Total Number Of Objects
1		n
Object No	Object Label	Object Parameters
1	Object	$\text{OEPR}_1,\text{REPR}_1,\text{M}_{001},\text{M}_{011},\text{M}_{101},\text{win\_size}_1,$ $\bar{x}\text{\_max}_1,\bar{x}\text{\_min}_1,\bar{y}\text{\_max}_1,\bar{y}\text{\_min}_1$
:	:	:
n	Target	$\text{OEPR}_\text{n},\text{REPR}_\text{n},\text{M}_{00\text{n}},\text{M}_{01\text{n}},\text{M}_{10\text{n}},\text{win\_size}_\text{n},$ $\bar{x}\text{\_max}_n,\bar{x}\text{\_min}_n,\bar{y}\text{\_max}_n,\bar{y}\text{\_min}_n$

:

:

Frame Number		Total Number Of Objects
N		n
Object No	Object Label	Object Parameters
1	Object	$\text{OEPR}_1,\text{REPR}_1,\text{M}_{001},\text{M}_{011},\text{M}_{101},\text{win\_size},$ $\bar{x}\text{\_max}_1,\bar{x}\text{\_min}_1,\bar{y}\text{\_max}_1,\bar{y}\text{\_min}_1$
:	:	:
n	Target	$\text{OEPR}_\text{n},\text{REPR}_\text{n},\text{M}_{00\text{n}},\text{M}_{01\text{n}},\text{M}_{10\text{n}},\text{win\_size},$ $\bar{x}\text{\_max}_n,\bar{x}\text{\_min}_n,\bar{y}\text{\_max}_n,\bar{y}\text{\_min}_n$

Table 5.2 Initial Object Description Table.

The structure of the scene will constrain the motion of a vehicle moving within that scene; trees, buildings, walls and fences would be areas in an scene where vehicles are not expected to move. However roads, tracks and possibly level grass fields are areas where vehicles would be expected to move. Vehicles moving in these areas exhibit known motion characteristics (acceleration, velocity, orientation change etc) which constrain the motion of the vehicle in a known way. These characteristics can be estimated *a priori* and used to help solve the frame to frame correspondence problem (Lacey [66]).

Solving the correspondence problem not only determines the trajectory of the vehicle in the image but also provides another cue to the identification of that region of interest. This is because the region appears to be moving in a known manner and as such is unlikely to be due to noise Target labels that satisfy the motion constraints are simply kept as targets. Object labels on the other hand that satisfy the motion constraints indicate that this region of interest is probably not an object but in fact is likely to be a target, (an object is unlikely to move with the known characteristics of a vehicle) and was probably mis-identified by the initial target analysis and as such it is re-labelled as a target. Target labels that fail to be matched are re-labelled as objects for processing in the next frame. Objects that fail to be matched are no longer processed. This method does address some of the issues of incorrect data (outliers, Torr [62]) by solving the correspondence problem between consecutive frames of image features. However if the system was to solve the correspondence problem over a larger number of frames, i.e. the labels must meet the motion constraints across a consecutive five frame window to be identified as a target, then this would give the system a larger temporal consistency for the motion tracks of objects moving in the image (Sobttka [85], Daum [46], Bordia, [63]) which may provide a more robust mechanism for tracking.

5.3.1 Frame To Frame Correspondence.

The continuous action of the motion cue generation (chapter 4) and initial target analysis essentially form a feature vector for each region of interest in the image, Consider:

$$\overline{\overline{O_j}} = [\, A^{1}{}_{j} \ldots A^{m}{}_{j} \,] \qquad (5.16)$$

where

j = 1, 2 .... n. (number of regions of interest this frame).

$A^{1}{}_{j}$ .... $A^{m}{}_{j}$ (measured features of a region).

Across an image sequence, an array of feature vectors would be formed:

$$\overline{O_{ji}} = [\, A^{1}_{ji} \ldots A^{m}_{ji} \,] \qquad (5.17)$$

where

j = 1, 2 .... k.
i = 1, 2 .... f. (image frames).

The frame to frame correspondence process has to match the features found in frame$_i$ with the features found in frame$_{i+1}$.

5.3.1.1 Dynamic Motion Constraints.

The objects of interest for tracking are man made (cars, vans etc) and as such their speed and turning ability (change of orientation) with respect to the camera can be estimated *a priori*. The objects to be tracked are assumed to be rigid, with any detected motion in the image being generated by the object moving and not due to any deformable surface in the object being tracked.

Though the performance characteristics of vehicles are different, a Ferrari F40 for example has 'slightly' higher performance than a Mini Metro, in general a fixed set of constraints can be placed on the velocity, acceleration and change in size that the object may undergo between consecutive frames. These constraints are based on the fact that the objects to be tracked are a large distance from the camera (>100 meters) and that these objects may be viewed at any orientation. Given this information and knowing the frame rate (frames per second input to the system), motion constraints can be estimated.

These estimates have been translated into a simple set of rules which limit the variation in object parameters between frames. Table 5.3 below shows the estimated variation for object parameters between consecutive frames.

Item	Parameter.	Variation.
1	OEPR	±10 %
2	M00	±6 image tiles
3	M01	±8 tile positions
4	M10	±8 tile positions

Table 5.3 Object Parameter Variation Table.

5.3.1.2 Tracking Implementation.

The tracking algorithm attempts to minimise a Euclidean distance measure between a subset of parameters from the object vectors, within the limits defined by the object parameter variation table.

A Euclidean function can be defined by

$$\Delta = d^2(\overline{O_j},\overline{O_i}) = \sum_{n=1}^{f} (\overline{O_{ni}} - \overline{O_{nj}})^2 \qquad (5.18)$$

Object correspondence is solved on a frame by frame basis. From table 5.3, the area of a region of interest is used as the initial matching feature. The largest area in frame 'i' is matched using the Euclidean distance measure to all areas in frame 'i+1' within a search perimeter space defined by items 3 and 4 in table 5.3. This matching process continues until all regions of interest are matched; any unmatched regions are assumed to be new tracks (Li-Qun, [23]). The correspondence matching process is shown in flow diagram form by figure 5.6.

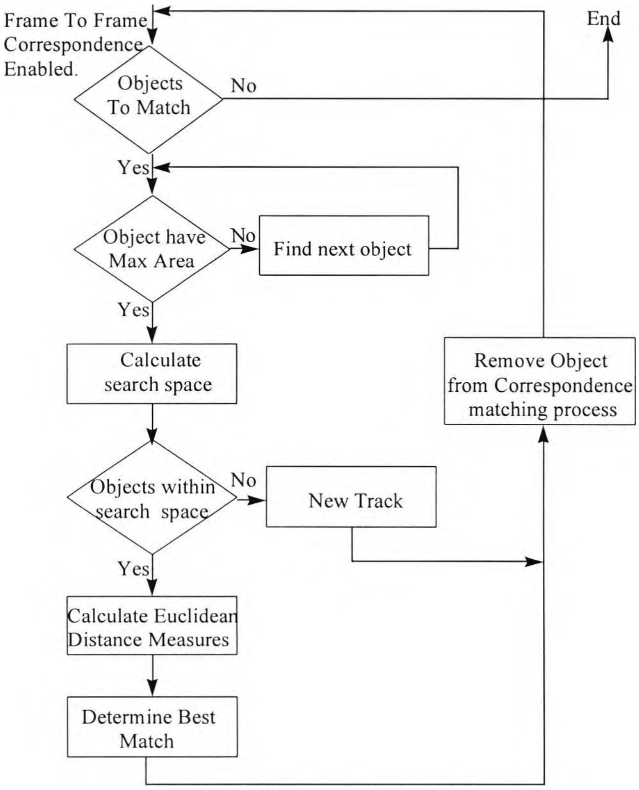

Frame To Frame Correspondence Enabled.
End
Objects To Match
No
Yes
Object have Max Area
No
Find next object
Yes
Calculate search space
Remove Object from Correspondence matching process
Objects within search space
No
New Track
Yes
Calculate Euclidean Distance Measures
Determine Best Match

Figure 5.6 Flow Diagram for the frame to frame correspondence matching process.

The target data generated by the tracking process consists of the active target data (table 5.4) and the in-active target data (table 5.5). The active target data is made up from the Object Edge Pixel Ratio (OEPR), the Euclidean distance measure and the zeroth and first order moments. The in-active target data comprises the Object Edge Pixel Ratio, zeroth and first order moments, and the frame number that the object was removed from the tracking process.

Frame Number	Target Parameters
3	Label$_k$, Label$_m$, Δ,OEPR, M00$_k$...M00$_m$,M10$_k$...M10$_m$,M01$_k$...M01$_m$
:	:
:	:
n	Label$_k$, Label, Δ,OEPR, M00$_k$...M00$_m$, M10$_k$...M10$_m$,M01$_k$...M01$_m$

Table 5.4 Active Target Data Table.

Label	Parameters	Frame Removed
Target	OEPR,M00,M10,M01	Frame No
Object	OEPR,M00,M10,M01	Frame No
:	:	
:	:	:
Target	OEPR,M00,M10,M01	Frame No

Table 5.5 In-active Track Data Table.

5.4 Results.

A static camcorder was set up and an open world image sequence filmed showing vehicles and people moving within that scene. From this sequence a 90 frame clip was digitised to disk at a rate of approximately two frames a second.

5.4.1 Single Target Identification and Tracking.

Figure 5.7 shows a six frame clip from that sequence where a single vehicle is moving. This vehicle is approximately 400 meters from the camera (focal length of the camera is f1.4) and figure 5.8 shows the resultant track of the target superimposed on the original image.

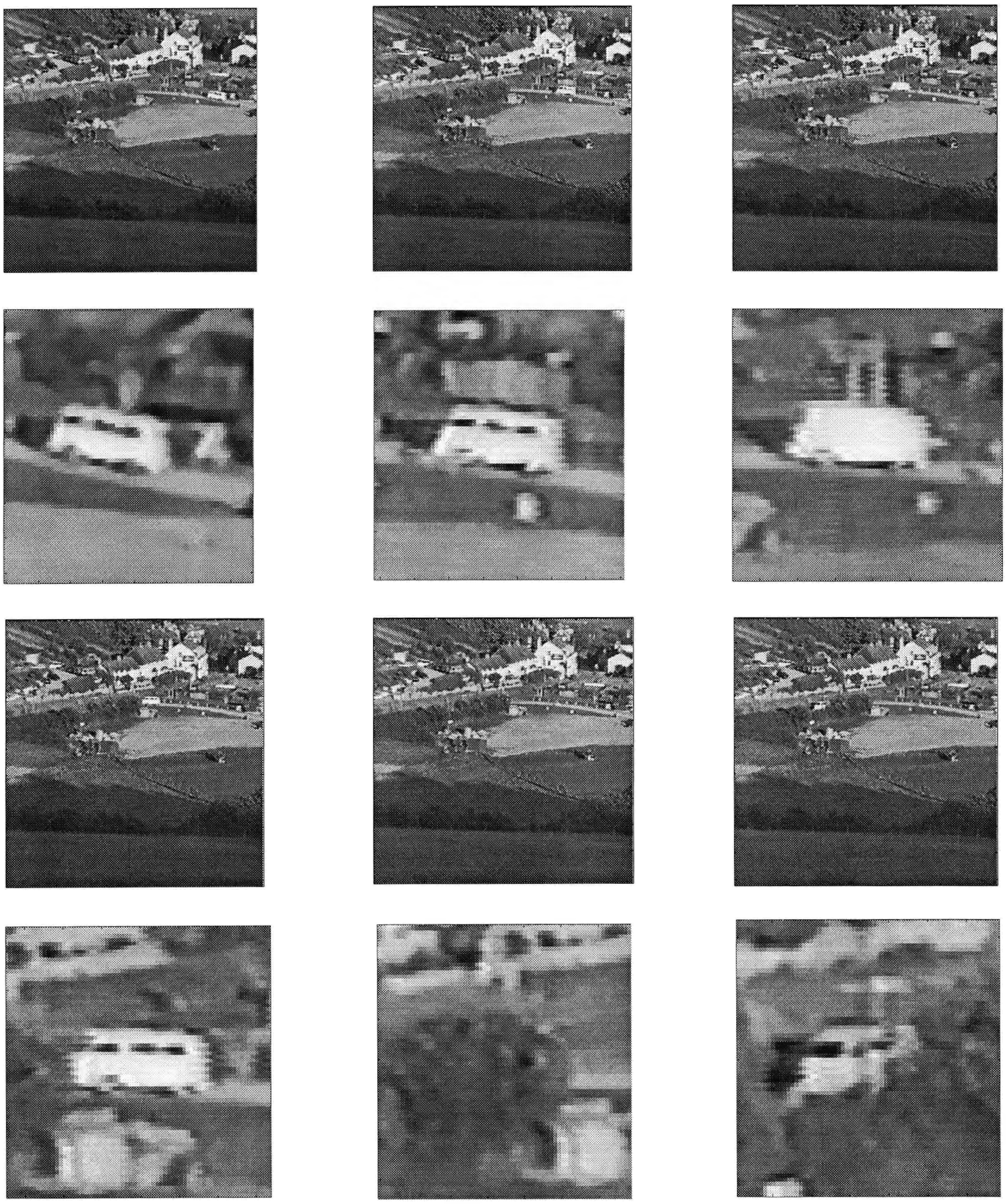


Figure 5.7 A six frame clip from an image sequence of over 90 frames. The top three images show the vehicle moving in a well defined manner left to right and up a slight gradient. The next three images show enlarged areas of the original image where the vehicle is moving. The following three images show the vehicle turning left and then becoming fully and then partially occluded behind a hedge. The last three images show the corresponding enlarged portions of the original image depicting this motion.

This sequence was used initially to test the tracker and debug the system as the vehicle movement is relatively well defined. Figure 5.9 shows a plot of the tracking error between the systems calculated centroid position for the target in the image and the actual position of the target in the image, determined by manually measuring the object motion using vehicle landmark features.

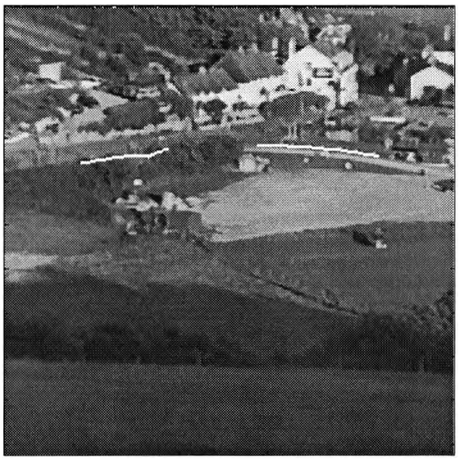


Figure 5.8 Resultant tracking sequence superimposed onto the original image.

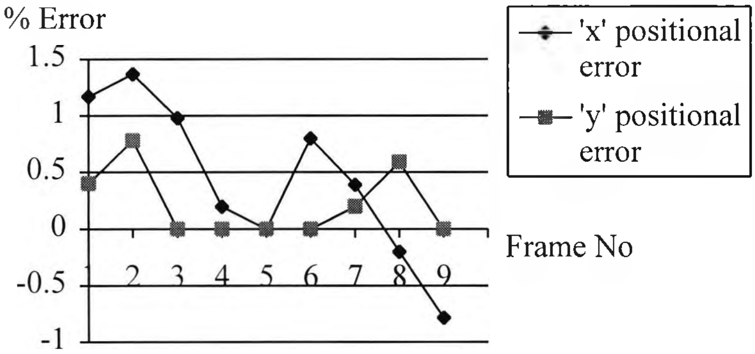


Figure 5.9 Plot showing the tracking error between the manually estimated centre of the tracked target and the trackers calculated centre of the target.

5.4.2 Multiple Target Identification and Tracking.

The tracking system will encounter multiple targets moving within the scene. Figure 5.10 shows six frames from an image sequence in which multiple moving targets that become occluded and occlude one another. Figure 5.11 shows the output from the tracker, as a result of these motion cues.

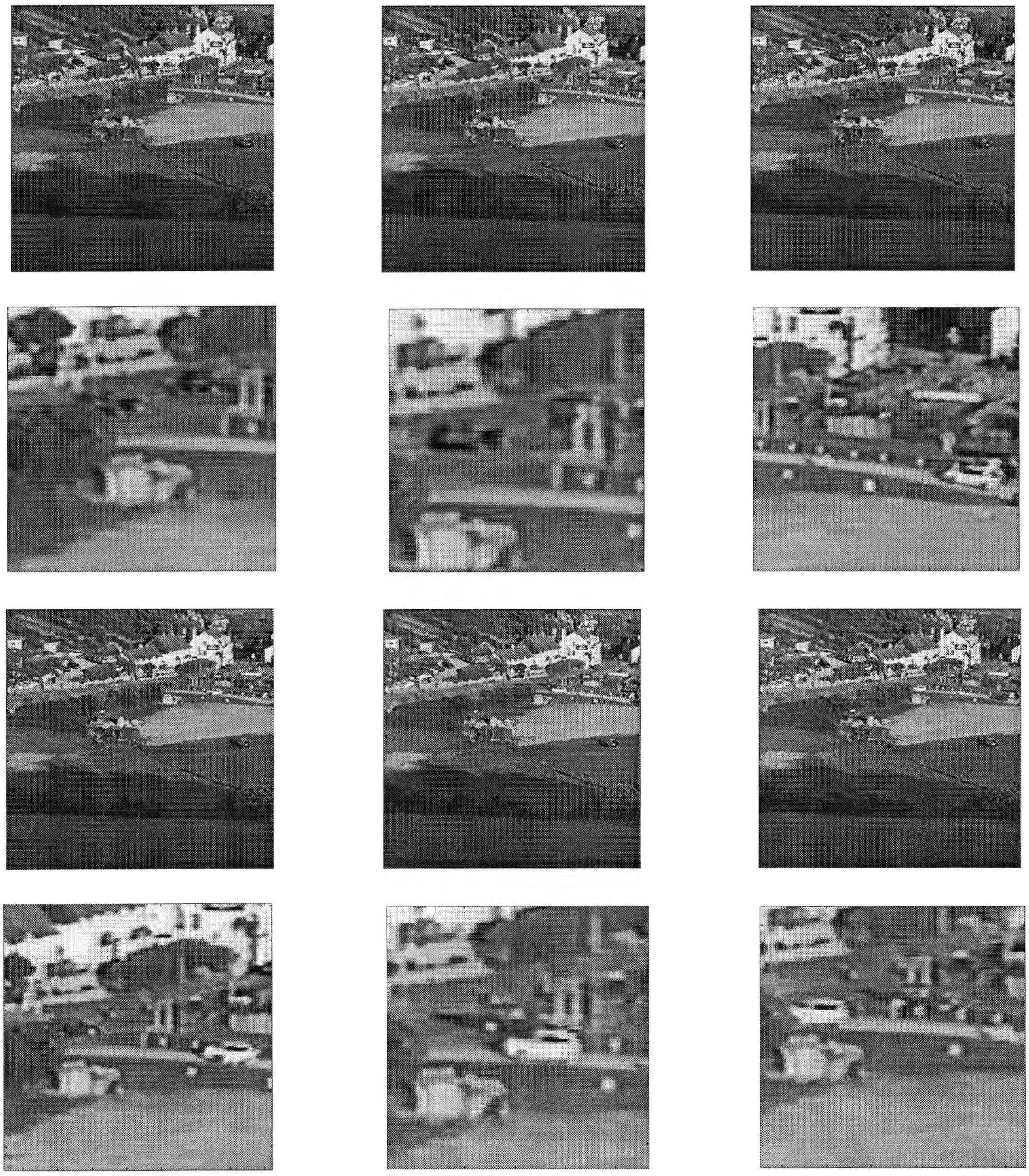


Figure 5.10 A six frame clip from an image sequence of over 90 frames. The top three images shows multiple vehicles moving within the scene, two vehicles are moving down the road into the cove and a third vehicle is starting to leave the car park moving up the slight gradient in a left to right direction. The next three images show enlarged areas of the original images where these vehicles are moving. The following three images are slightly later in the sequence and show a fourth vehicle turning right and entering the car park. The car leaving the car park eventually occludes the vehicle that entered the car park. The last three images show enlarged corresponding portions of the original image where these vehicles are moving.

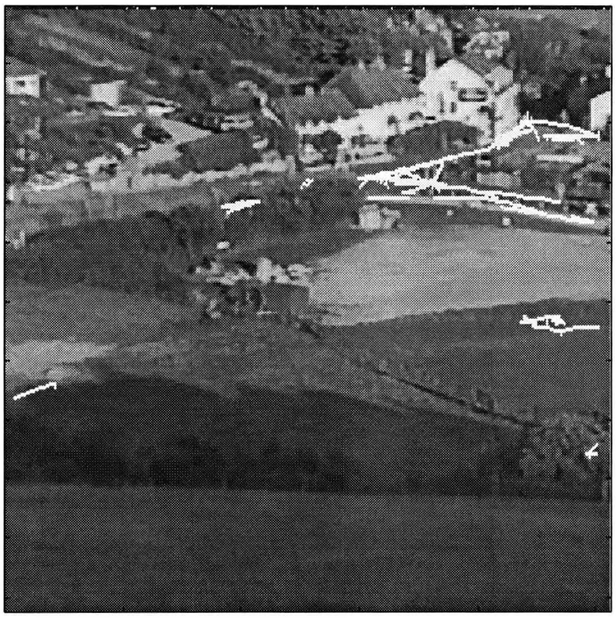


Figure 5.11 Tracking plots superimposed onto the original image.

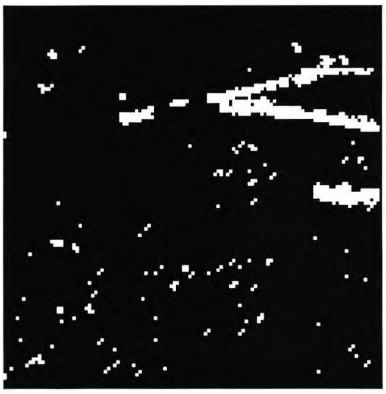


Figure 5.12(a) Motion cues generated frames 2 to 69.

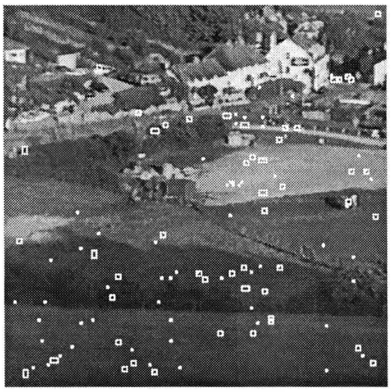


Figure 5.12(b) Extracted un-matched motion cues for frames 2 to 69.

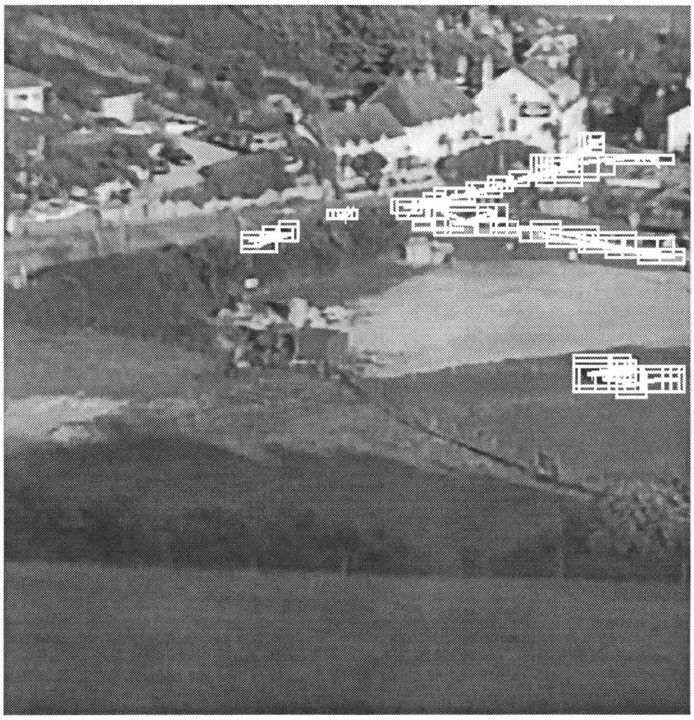


Figure 5.12(c) Identified and tracked vehicles for frames 2 to 69.

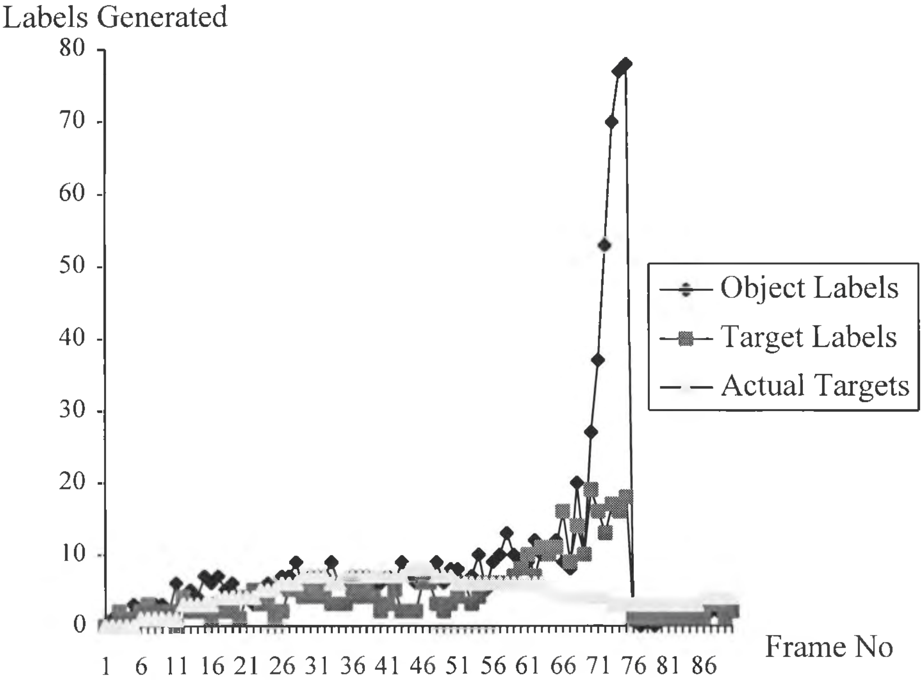


Figure 5.13 Plot showing Object and Target labels generated per frame with actual targets present each frame.

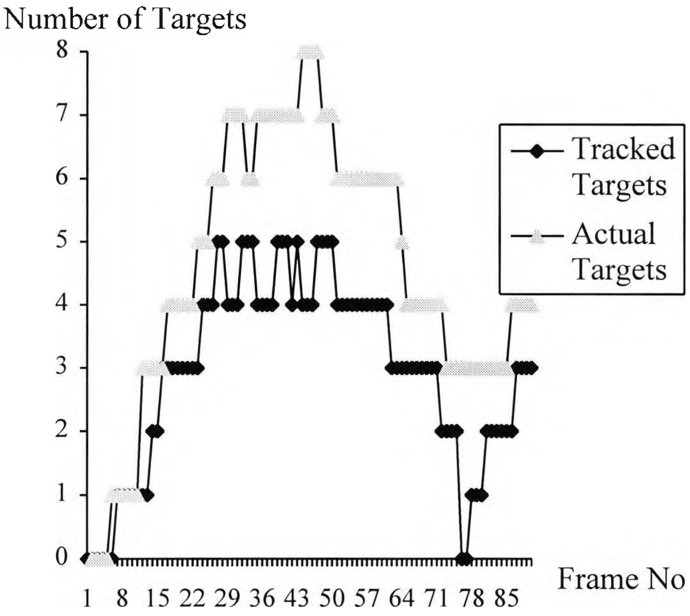


Figure 5.14 Plot showing actual targets per frame against actual tracked targets per frame.

5.5 Discussion and Summary.

The open world image sequence used to test the identification and tracking algorithm was filmed for approximately one and a half minutes using a static camcorder. The resulting image sequence shows multiple vehicles moving such that they become occluded behind static objects in the scene. Vehicles also occlude one another as they moved into or out of the car park. The initial target analysis uses a crude measure of the edginess of an object to perform the initial identification of that object. This simple measure gave good results for the correct identification of a region. Looking at figure 5.4, the van is approximately 400 meters from the camera and only occupies an area of 48 by 16 pixels.

Figure 5.5(a) shows the van enlarged with figure 5.5(b) showing the edge pixels found within that region of interest. Figure 5.5(c) shows the edge pixels for that region of interest found within the reference edge image. There is a clear change in the distribution and number of edge pixels within this region. This is expected as the structure for that region has changed. As just outlined the initial target identification process is not trying to match edge pixels within a region with some form of parametric model of a vehicle. Instead it uses a simple edge ratio calculation between the edges found for an object in the current frame and edges found in the corresponding area of a reference edge image. For the van, the reference edges found totalled 350 and the current edges found totalled 305, a decrease of nearly 13% in the ‘edginess’ between regions. This measure includes a contribution due to the shadow of the vehicle. The shadow of the vehicle does not significantly affect the identification and tracking process as the vehicles are a large distance from the camera. As the distance from the camera is large the contribution of the shadow to the motion cue is small. However for vehicles that are closer to the camera, if the shadow of the vehicle is found to significantly affect the initial identification process, then the shadow could be removed using a shadow detection algorithm such as that of Scanlan, [50].

Figure 5.13 shows a plot of the number of object and target labels generated by the initial target identification process, together with the actual number of targets in the scene. The misclassification of target and object labels is due to the fact that vehicle motion occurs at distances over 500 meters and at this distance the motion cues generated have insufficient edge information for the edginess comparison.

Due to the large distances and orientation changes that the target undergoes as it moves in the image, tracking mechanisms based on vertices or line segments (Crowley [59], Deriche [60]) will tend to fail. The approach taken here is therefore based on region tracking using the centre of gravity of the object being tracked to form the motion vector (Gordon [61]). Meyer [82], highlighted that region tracking generally reduced to tracking the centre of gravity of an object and that this form of tracking could not capture complex motion of objects in the image plane. However the objects to be tracked in the image do not have complex motion components and this is not deemed to be a problem.

The target tracker was initially tested using part of the image frame sequence where there was only a single vehicle moving in the image. Figure 5.7 shows six frames of a sequence where a van is moving in the image. Although its movement is relatively well defined, the vehicle undergoes first partial then full occlusion. The system tracked the van through the image sequence until it became fully occluded behind the hedge. It was also able to re-acquire the target after it re-appeared in the image even though the frame used for re-acquisition shows the van still partially occluded by the hedge.

Figure 5.8 shows the output from the tracker superimposed onto the original image for this sequence. The tracking system is implementing a multi-resolution approach (Caplier [69]) to tracking that is capable of tracking objects despite the fact that the tracked object is viewed at various orientations. Figure 5.9 shows a plot of the error in the trackers calculated position in the image for the centre of the van and a manually determined value for the centre of the van. If you take into account the fact that the reference centre for the van was determined manually and the region analysis process calculates the centroid positions based on tiles not pixels (possible error of +/- 0.8%). Then the error of 1.5% is probably worse case and as Bers [77] highlights it is very difficult to assess the performance of a tracking system.

Figure 5.14 shows a plot of the actual number of targets in each frame across the sequence together with the actually tracked targets. The disparity is due to the tracking algorithm not being able to resolve object motion unless there is at least 1 tile between object tracks. If objects occlude one another, then the motion cue generator will only produce one motion cue (a draw-back of a frame differencing technique, Rowe [75]) and the tracker associates the best matched track with the single target during the occlusion.

The problem of target tracking being lost during occlusion is due to the mechanistic nature of the algorithm. It has already been highlighted that this may be improved by considering the correspondence problem over a larger number of frames. However we must remember that tracking targets that are capable of making unpredictable manoeuvres in natural scenes is considerably difficult (Hutchins, [78]). The tracking system did reject most multiple false motion cues, however two tracks were produced by the system, but these tracks were however only identified as objects and were generated just prior to the reference data being updated. The identification and tracking system is intended to be used as a region cueing aid providing scene driven information to the 'spatial-temporal reasoning process' (chapter 6) which is constructing a map of the scene on a frame by frame basis based on the motion of targets within the scene.

What is significant is that only identified vehicles are tracked and that motion cues due to other sources (illumination changes, noise etc) must not be tracked. Looking at figure 5.12(a), it clearly shows that a large number of motion cues were generated by the input image sequence, but those motion cues actually tracked were in fact cues due to vehicle motion. Figure 5.12(b) shows the motion cues that were extracted and not tracked (false motion cues), while figure 5.12(c) shows the motion cues tracked by the system. These results show that the system can track actual vehicles using relatively simple techniques, producing accurate and reliable estimates of vehicle motion in the image (this is seen as an essential factor in target tracking in dense environments [51]).

Also the multi-resolution approach does not significantly alter tracking accuracy, b ut provides a more focused computation (Burt, [73]). The tracking system has demonstrated that it is capable of extracting and tracking man made objects in open world scenes, even though those tracked objects changed their orientation or became occluded. In the next chapter we shall see that the extracted motion data is sufficiently accurate for use by a spatial-temporal reasoning process.

Chapter 6 Spatial-Temporal Reasoning.

6.1 Introduction and Overview.

Scene understanding is generally dominated by two main themes, firstly measurement of local attributes in an image are used to identify features which characterise objects in a scene and secondly the use of prior expectation to guide interrogation of the image. In AI studies the two themes generally correspond to data driven methods and goal driven methods. The spatial reasoning process requires input from the bottom up processes in the form of factual descriptions of areas of the image and so encompasses aspects of a data driven approach, but it also incorporates knowledge of objects and their relationships in the scene and so incorporates aspects of goal-directed methods.

Systems developed by Godden et al [10] and Morton [64] inferred that a first-stage hypothesis can be generated from the results of low-level segmentation algorithms together with some form of contextual reasoning process to label those segmented areas where an object might exist. Further support for the presence of the object could then be obtained by statistical analysis on groups of those regions (Hutber et al [76]). However unlike these systems, the system described in this thesis bases the support for the existence of an object on the detection of image motion which, coupled with constraints placed on that motion, are used to determine its presence within the image.

The knowledge-based reasoning being carried out here is not trying to determine the presence of the object within the image, as this task has already been accomplished (chapters 4 & 5). The system is trying to determine the likely structure of areas within the image based on this motion. The knowledge-based reasoning is attempting to construct a map of the scene based on the movement of man made objects in the image.

The map gives a measure for regions in the scene where vehicles are likely to be observed moving and regions where they are likely to become occluded. The second objective of this research was to develop an algorithm capable of interpreting major structural features in the scene based on the motion of objects moving within the image, where these interpreted regions are related to spatial-events (Howarth [65]) that have been detected across a number of frames. Major structural features are defined as regions in the image where man made vehicles can be expected to be observed moving and regions where vehicles could become occluded from the camera but are in fact still in the field of view. In general objects only enter or leave the field of view at image boundaries (or at occlusions). The point of identifying potential occlusive regions in the image does not appear to have received much attention in the machine vision literature.

The previous chapters developed an identification and tracking algorithm that uses a multi-resolution approach (James, [81]) to track objects moving in an image. The output from this tracking stage is target data that represents the trajectory of the vehicles, a time index and information about their size (tables 5.4 and 5.5). The target data thus yields information on the spatial-temporal events of vehicles moving in the scene. It is now the task of the spatial-temporal reasoning process to use this tracking data in conjunction with a knowledge database, to build up the map that represents spatial areas within the image where vehicles can be expected to be observed moving and perhaps more importantly, identify areas in the image where the tracked objects could become occluded from the camera but are still in its field of view.

6.2 Knowledge Representation.

The internal knowledge base is divided into both 'analogical' and 'propositional' models as this reflects a similar theory concerning how the human vision system represents the world (Johnson-Laird, [89]). It is argued that a multi-representation strategy for machine vision would be more efficient than translating all the problems into one form of representation and solving the recognition problem using one specific representation.

The system developed here uses both analogical and propositional representations for the knowledge representation and a semantic network is a convenient way to represent both forms of knowledge. The semantic network supports analogical and propositional knowledge by representing analogical knowledge as objects and propositional knowledge as relationships between those objects in a graph structure of nodes and labelled arcs.

6.3 Implementation.

It has already been demonstrated [23, 55] that the motion of objects moving within an image can be used to construct some form of representation of that image. This interpretation process takes the form of identifying areas within the image where vehicles can be expected to be observed moving and areas where vehicles could become occluded. This interpretation process is split into two main tasks, firstly the data supplied by the tracker must be analysed. The analysis groups the trajectory data into connected sets of tiles which represent spatial areas in the image where vehicles have been observed moving (map segments). The second task takes the map segments from the spatial analysis process and applies a spatial reasoning process to the possible spatial and temporal interpretations between these map segments.

Figure 6.1 on the next page shows the level 2 DFD for the spatial reasoning process. When enabled by the ‘goal processor’ the ‘spatial-temporal control process’ triggers the ‘spatial analysis process’ which constructs the map segments from the data supplied by the tracker. Upon completion of its processing it generates a status signal indicating that its processing is complete and the spatial-temporal control then triggers the spatial-reasoning process that interprets (infers) the spatial structure of the scene based on a rule-production scheme of the most likely spatial relationships between the constructed image segments. When the spatial reasoning process has completed its processing, it generates a status signal back to the spatial-temporal control process, which then returns a ‘reasoning status’ signal to the goal processor indicating the status of the processing carried out by the spatial-temporal reasoning process.

6.3.1 Spatial Analysis.

The spatial analysis takes the target tracking data output from the tracker and constructs sets of map segments based on the target trajectories. The segments are mapped into the image in tile co-ordinates. Each segment has an edginess factor calculated for it, a time index based on the frame number and an observation factor is calculated from the number of instances targets have been observed in that segment. The map segments generated by the spatial analysis process are used by the spatial reasoning process to perform an interpretation of the possible structure of that region of the image. No assumptions are made about any *a priori* structure within the scene, so initially all regions in the map are labelled as being 'unknown', thus the system effectively starts with an empty map.

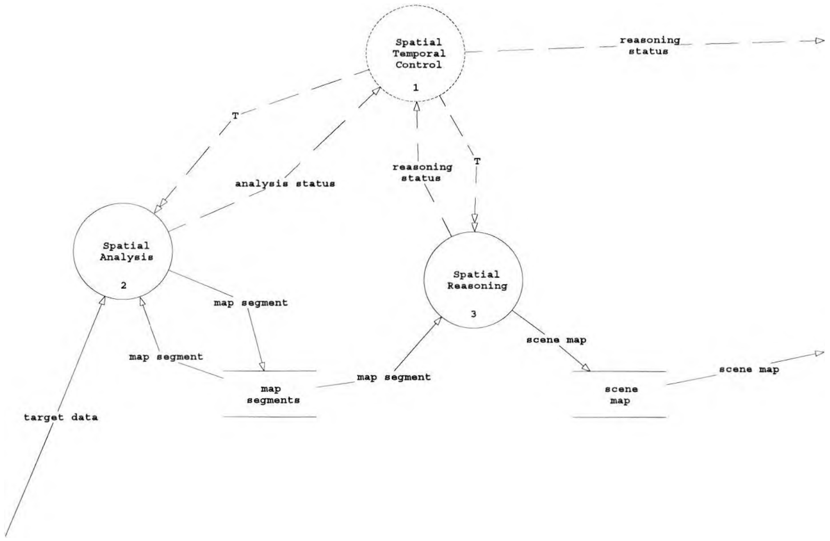


Figure 6.1 Level 2 DFD for the Spatial-Temporal Reasoning Process.

6.3.1.1 The Map Segment.

The map segment is the fundamental building block in the generation of the map. A map segment consists of a connected set of image tiles and each of these connected sets has a basic form < Label, Position, Time > (Toal et al, [26]), plus an additional quantity which is a measure of edginess in the areas of the map that have shown motion. The structure of a map segment is shown on the next page in figure 6.2.

6.3.1.2 Time Index.

The time index for the map segment is derived directly from the frame number.

6.3.1.3. Observation.

Map segments are stored in an array as they are created. The observation function searches the previously processed map segments (stored in the array) and increments an observation count for each segment if motion is detected in the same position in the current map segment.

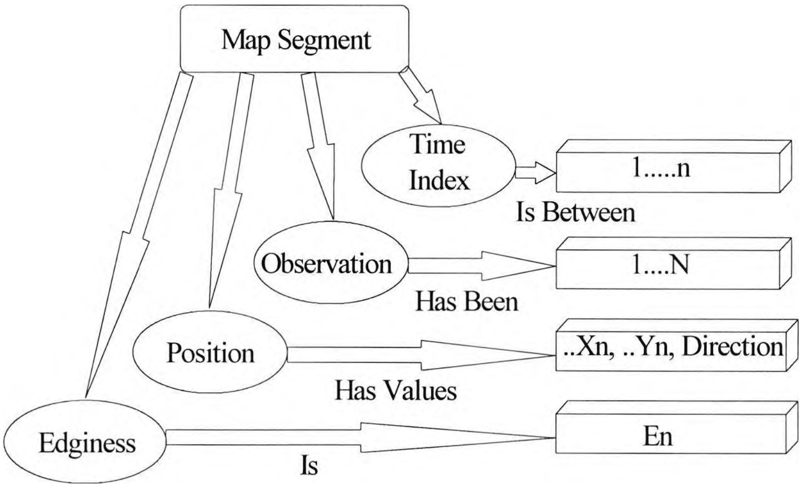


Figure 6.2 Map Segment Structure.

6.3.1.4 Position.

Position information is derived directly from the active target data table (chapter 5, table 5.4) which contains the area and centroid positions of the matched targets in the current frame in tile co-ordinates. From this information a straight line segment is derived from the pair of centroid co-ordinates using a modified form of Bresingham's line drawing algorithm (as described in [87]) to extract the tiles in a straight line segment.

The assumption that a vehicles movement can be approximated by a straight line segment is based on the large distance a vehicle is expected to be observed moving from the camera and that image data will be generated at sub-second intervals (typically 10 frames a second); in this case the distance a vehicle can move in the image between frames is small and this movement can be approximated by a straight line.

6.3.1.4.1 Directional Vector.

As part of the position data, a directional vector is added based on the targets motion in the x and y directions between consecutive frames of target data. The compass is shown in a diagrammatic form in figure 6.3.

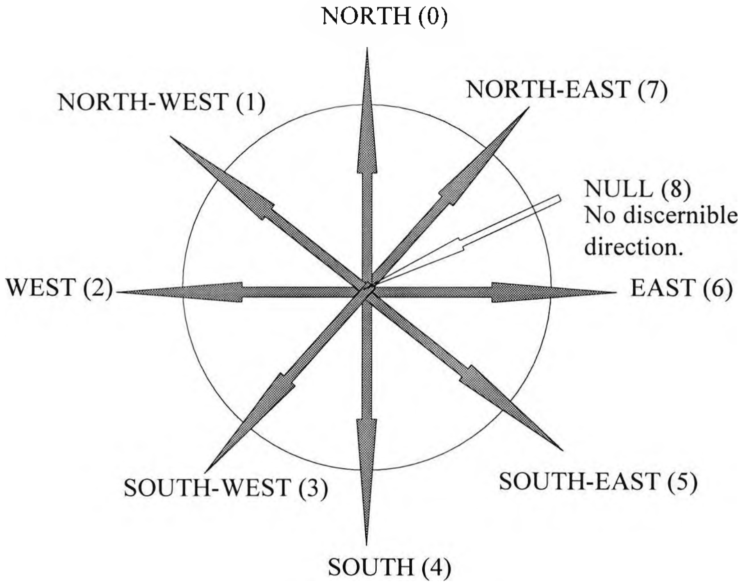


Figure 6.3 8-point directional compass.

6.3.1.5 Edginess.

The tiles calculated by the position function indicate an area in the image where motion has been detected. From this information the edge structure for that area can be determined. Unlike the initial target analysis, which is attempting to identify changes in edge structure between a reference frame of edge data and the edge data generated from the current frame, the edginess function is only interested in evaluating the background edginess for the region that has shown motion using the edginess of the region to give an indication as to its possible identity. The number of edgels within a single tile can be calculated using

$$\sum_{i=1}^{4}\sum_{j=1}^{4} Eref(i,j) \tag{6.1}$$

From 6.1 the number of edgels within a map segment can be calculated using

$$\frac{1}{No}*\sum_{1}^{No}\sum_{i=1}^{4}\sum_{j=1}^{4} Eref(i,j) \tag{6.2}$$

where

Eref(i,j) is the reference edge pixel image.

No is the number of tiles showing motion (position data).

6.4 Spatial Reasoning.

The spatial reasoning process builds the map of the scene based on the premise ‘that a scene can contain one or more of the following objects: Road, Ground, Static and Unknown’. The spatial reasoning process takes the map segments generated by the spatial analysis process and uses a rule based approach to infer the most likely interpretation for regions in the image that have exhibited target motion. The spatial reasoning is structured using the semantic network shown below in simplified form.

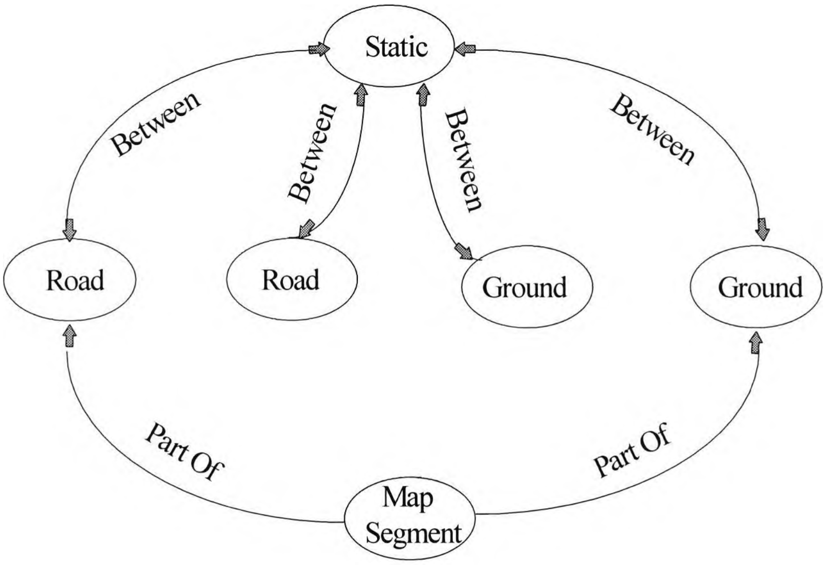


Figure 6.4 Semantic Network.

The network has two arcs, namely ‘part of’, and ‘between’ and four object nodes ‘road’, ‘ground’, ‘static’ , and ‘map segment’. However as the scene map starts off by being labelled as ‘unknown’, the system effectively starts with an empty map. The unknown node is not shown on the semantic net as the map can only be labelled as being unknown by the system on initialisation, any segments that are labelled as either road, ground or static cannot be re-labelled as unknown unless the system is re-initialised.

As all regions of the map have initially been labelled as being unknown, the network starts off with no *a priori* knowledge of any structure within the scene. ‘Part of’ takes a map segment and checks to see if it could already be part of either a road segment or a ground segment, invoking a set of spatial and structural operators to accomplish this task. If the identified segment is a repeat of an already identified map region, then ‘part of ‘ stores this in a current scene map (short term memory) and a node consistency rule adds it to the already identified map region, resolving any label conflicts. If the identified segment has not been observed before then ‘part of’ generates a new map region for that segment. The ‘between’ operation applies a set of geometric rules that use the premise, ‘roads or ground regions that are associated with motion can be linked using straight line segments (roads are considered to be straight within a defined search space). If a link can be established then this could potentially indicate the presence of an occluding object in that region of the image. If links are established between identified regions, those links are labelled as static, i.e. that area of the image contains an object that may occlude vehicles moving into that region of the image.

However if motion is observed in any of the spatial links established between identified regions using the above premise, then that region is re-labelled as either road or ground. The resulting scene map is an array of labelled nodes, essentially a 2-D array of structures. Each node in the array has a likelihood value and node activity value associated with it. The likelihood value gives the measure of the node being correctly labelled as a road and the activity value gives a measure of the amount of target motion that has been observed for that labelled node. Figure 6.5 below shows the expected layout of a scene map generated by the spatial reasoning process.

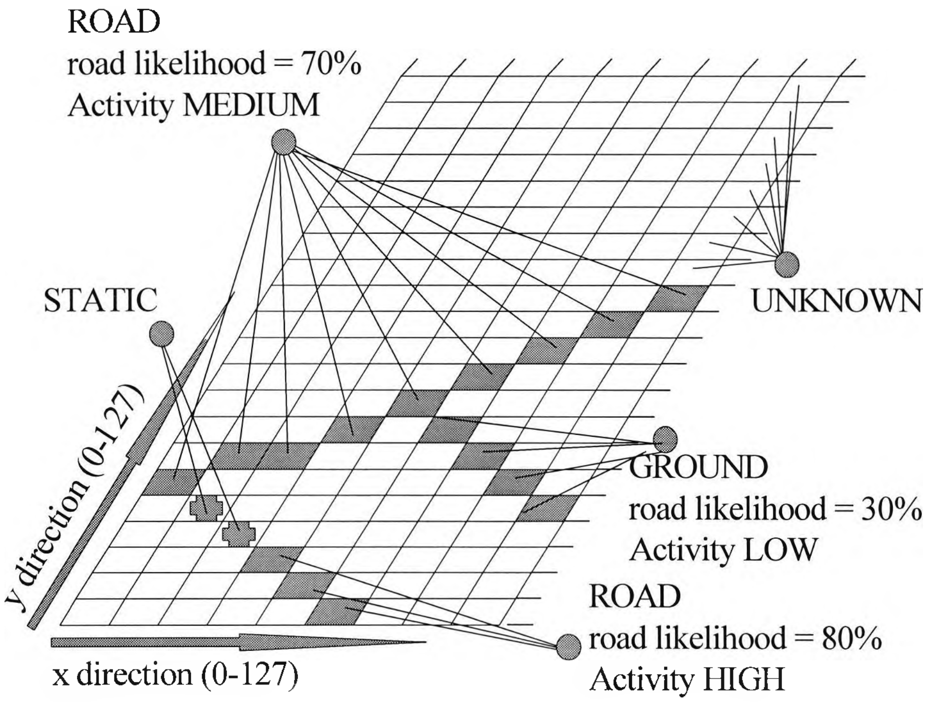


Figure 6.5 Structure of the Scene Map.

6.4.1 Network Arcs.

The network arcs are used to infer the likely identity of a map segment. Inference is a process of deducing facts from known facts, which is the cornerstone of rational thought. There are a number of mechanisms we can use to implement inference, namely , predicate logic, production systems, labelling and active knowledge (Ballard & Brown, [84]).

Taking each in turn, predicate logic is a method for expressing propositions and deriving consequences based on facts. Production systems are a general rewriting system, which consist of a set of rules and an executive program which applies those rules. Labelling schemes tend to involve methods of mathematical optimisation in some form of continuous space. Finally active knowledge is where each chunk of our knowledge is represented by a program, this effectively procedurises the implementation of propositions.

All of these methods for implementing inference have their strengths and weaknesses. However production-based knowledge systems generally tend to be more robust and easily modified up to a certain level of complexity, but above that level rule-based systems tend to become un-manageable. The system being developed here keeps the complexity of the inference mechanism to a minimum and as such the inference mechanism used to construct the map adopts a rule-based probabilistic reasoning approach (Sucar et al, [88]).

Structuring the inference mechanism in a semantic network offers the advantages of

Being easily modified.

Producing a modular design and implementation.

It is easily understandable.

Production systems support a general form of inference by using a matching technique to identify which inference to make. This action requires an explicit set of situation-action nodes which are evaluated against a database of situations. To form even a simple production system requires a database, a set of rules and an interpreter for those rules. These requirements are used to form the basis for traversing each of the arc's and consequently each arc has its own set of rules, its own data base and its own interpreter.

6.4.1.1 Part Of.

The 'part of' arc has the task of identifying whether a map segment in the current image frame is either road or ground (ground is considered to be anything else that can contain vehicle motion that is not a road) and then create either a new road segment or a new ground segment in the map, (the process of creation can also mean just adding the current map segment to an already identified map segment). The 'Part Of' arc uses two processes to determine the likelihood of a map segment being a road, namely: Has Structure and Has Displacement.

6.4.1.1.1 Has Structure.

The map segments calculated by the spatial analysis provide the target trajectories in the image in tile co-ordinates. These co-ordinates represent a straight line segment calculated from the centroid of the target between two frames. This trajectory data is not entirely representative of the area of the image containing target motion (the road for example), and so the straight line segment is expanded by a factor equal to half the area of the target (lower half of target area) which gives the area of the image that the target is actually moving in (the road is assumed to be under the vehicle).

Figures 6.6(a and c) show two enlarged sections of the image where vehicle motion has been detected and tracked by the tracker. In the first instance a vehicle has been detected and tracked on a road, in the other, the vehicle has been detected and tracked on grass. Figures 6.6(b and d) show the edgels generated by the edge detector for those two regions before the vehicles entered that area of the image. This edge data is extracted from the reference edge image (no further application of the edge detector being required), thus the tracking data has effectively focused the image processing on those particular sections of the edge image.

The target trajectory co-ordinate data for the road case, covered an area almost identical to that occupied by the road. The edge count for this extracted region was 21 edge pixels and the tile area was 18 tiles (18*16=288 pixels). Using equation 5.13 the edge pixel density (ratio of edges to area) can be calculated, this gives an edge density for the extracted map segment for the road region of 0.073. The grass region however yielded an edge count of 61 edge pixels for a corresponding tile area of 15 tiles (15*16=240 pixels), giving an edge pixel density for this grass region of 0.254.

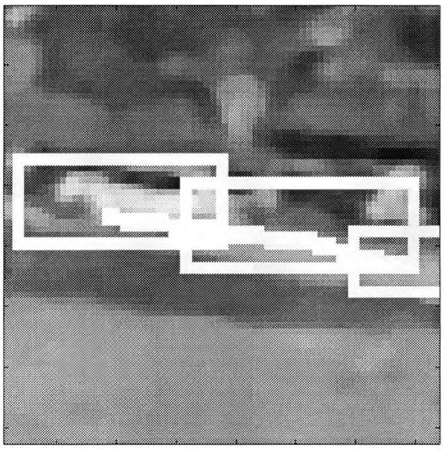


Figure 6.6(a) Identified and tracked vehicle on a road region.

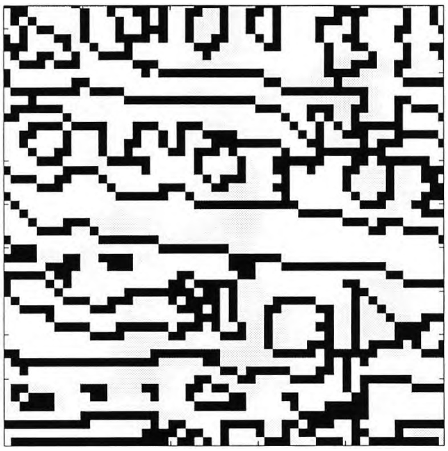


Figure 6.6(b) Extracted edges for the corresponding road region.

The structural data base table is based on the premise ‘that road surfaces being man made tend to be smooth and flat and as such will produce significantly fewer edges’ (as shown by figure 6.6(b)). However with none road surfaces the opposite of this premise will be true, as grass or dirt regions (dependent on their range) will be highly textured producing a rich number of edges, which is shown by figure 6.6(d). From the experiment carried out in chapter 4 (section 4.3) edge pixel densities were calculated for all regions that contained vehicle motion.

From this experiment a database was constructed that relates the edge pixel density to a percentage likelihood of a region being a road, this is shown in table 6.1. In these two cases shown (figures 6.6(a) - 6.6(d)) the ‘Has Structure’ rule returned that it was 85 % likely that the road segment was in fact a road, but that it was only 30 % likely that the grass region was a road.

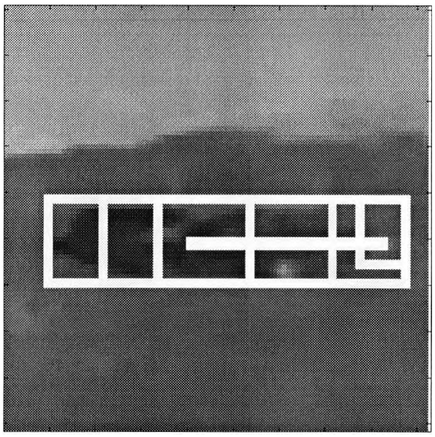


Figure 6.6(c) Identified and tracked vehicle on a grass region.

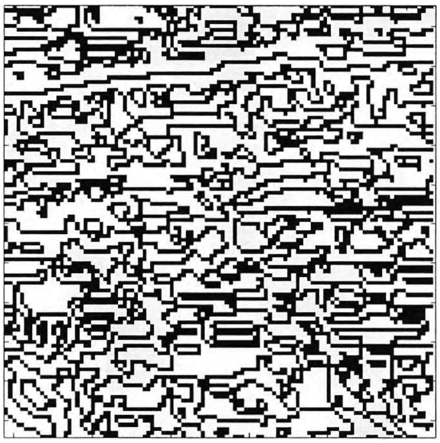


Figure 6.6(d) Extracted edges for the corresponding grass region.

Edge Density	Likelihood of Road (%)	Likelihood of Ground (%)
0.000 - 0.025	95	05
0.026 - 0.050	90	10
0.051 - 0.075	85	15
0.076 - 0.100	80	20
0.101 - 0.150	70	30
0.151 - 0.200	60	40
0.201 - 0.250	50	50
0.251 - 0.300	30	65
> 0.301	20	80

Table 6.1 Percentage likelihood of segment being road or ground, based on region edge structure.

6.4.1.1.2 Has Displacement.

We could postulate that vehicles will tend to have a higher velocity when moving on a road surface than when they are moving on a dirt track or a grass surface. It could be argued that when cars turn from one road junction to another or when they come to enter or leave a car park that their velocity would also be small. In general a vehicles velocity on a road will tend to be higher than when the vehicle is moving on a non road surface.

Here the velocity of a vehicle is calculated based on the distance the target moves between two frames (centroid to centroid). As the distance from the vehicle is unknown, it is necessary to normalise this calculation by the average area of the vehicle between the two frames. The area of the vehicle could be considered to provide a crude estimate of the range the vehicle is from the camera as the area changes approximately linearly with distance. Consider

$$\text{x\_diff} = \text{abs}(\text{M10}_k - \text{M10}_m) \tag{6.3}$$

$$\text{y\_diff} = \text{abs}(\text{M01}_k - \text{M01}_m) \tag{6.4}$$

the mean area of a target between frames can be calculated using

$$\text{mean area} = (\text{M00}_k + \text{M00}_m)/2 \tag{6.5}$$

then

$$\text{estimated displacement} = \sqrt{(\text{x\_diff}^2 + \text{y\_diff}^2)} \Big/ \text{mean area} \tag{6.6}$$

For the map segments depicted by figures 6.6(a) and 6.6(c) the x,y centroid positions for the current and previous frames were (119,44), (113,43) and (122,67), (117,67) respectively and the associated areas for each instance was (27,24) and (34,31) respectively. This gives an estimated displacement for the vehicle moving on the road of 0.225 and an estimated displacement of 0.164 for the vehicle moving on the grass. Like the data table for the edginess of a region being used as a measure of its structure, the values within table 6.2 for structure of a region based on motion were determined experimentally using velocity results for all the vehicles moving within the image sequence (determined by experiment).

The values estimated for the displacement are now used to look up the likelihood of that particular map segment being a road, based on that estimated displacement. From table 6.2 the system returned a 60 %

likelihood of the road segment being road and a 50 % likelihood of the grass being a road segment.

Estimated Displacement	Likelihood of Road (%)	Likelihood of Ground (%)
0.000 - 0.050	10	90
0.051 - 0.100	20	80
0.101 - 0.150	40	60
0.151 - 0.200	50	50
0.201 - 0.250	60	40
0.251 - 0.300	80	20
> 0.301	90	10

Table 6.2 Percentage likelihood of segment being road or ground, based on a targets velocity.

These results are not unexpected, as in these particular cases the car on the road is leaving the car park and going up a slight gradient and as such is not moving very quickly. With the grass region, the car manoeuvres as it comes to a halt for parking and its area reduces between the two frames giving a slightly higher apparent displacement. The system cannot make a more definitive determination of the likelihood of the map segment being a road based on the estimated displacement. However this problem can be alleviated by increasing the frame rate of the system from approximately a second to a tenth of a second, which would reduce the effect of self occlusive manoeuvres on the estimated displacement.

Overall the system has calculated a likelihood road value of (85% + 60%)/2 = 72.5% for the road region being a road and a likelihood road value of (30% + 50%)/2 = 40.0% for the grass region being a road. For this particularly difficult case the system demonstrated that it could discern between road and grass regions.

6.4.1.1.3 Label Current Map.

The function 'Label Current Map' is not an arc in the network, it is a control function within the overall semantic network. This function labels a current scene map (a short term memory representation of the scene, it exists for one frame only). The node kernel updates the scene map (long term memory) when it evaluates the node labels for consistency.

6.4.1.2 Between.

If target motion is detected and tracked in the image at some position (x_i,y_i) at time t_i and at a later time t_j target motion is again detected and tracked at a position (x_j,y_j), then these two target tracks could be related to the same target but the target was occluded by some object in the image for the time (t_i-t_j). A region in the image can therefore be defined by a straight line segment between co-ordinates $(x_i,y_i;x_j,y_j)$ which may indicate the presence of an occluding object in the image at those co-ordinates. This region is estimated by searching the scene map using the line segment end points and directional vector, linking any regions with a 'static label' if they are within the predefined search area. Using the directional vector to control the search direction results in 9 possible masks, figure 6.7 below shows the typical layout of the south-east (5) mask.

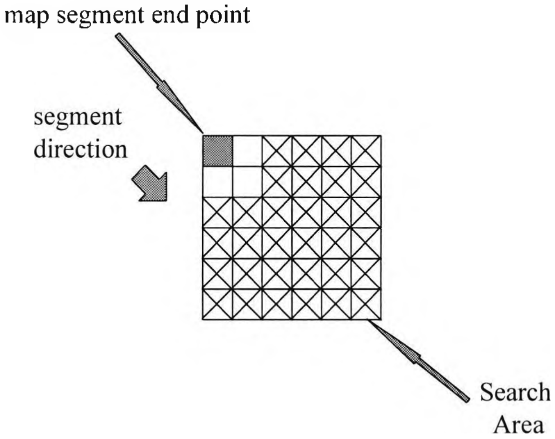


Figure 6.7 Typical layout of the search mask.

The 'Between' arc implementation has its rules, data-base and interpreter embedded within it. If at any time a new map segment is added that occupies the same co-ordinates as a static node, the static node is overwritten with the new map segment. The 'between' arc identified two main static object areas in the image. These areas corresponded to an actual hedge in the scene where vehicles did become occluded from the camera, but were still in the field of view.

6.4.2 Network Nodes.

The basic requirements of an image processing system engaged in image analysis and interpretation are, an ability to represent classes of objects or events that may be in the scene and some form of criteria for calling the knowledge representation scheme adequate. The 'epistemological adequacy' that is: "a representation is called epistemologically adequate for a person or machine if it can be used practically to express the facts that one actually has about aspects of the world" (McCarhty [80]). Brachman [83] pointed out that the epistemological adequate scheme must be neutral with respect to a conceptional level of knowledge base. That level should be built of concepts and relationships between those concepts that are relevant to the given task.

To this end the semantic network used here for the knowledge representation scheme has unlike Niemann [79] or Deruyver [74] only 4 nodes and two arcs to represent the domain knowledge. The system is engaged in a tracking and interpretation task, identifying regions of the image that are associated with motion and as such within the problem domain only a few objects are permitted. These few objects however can in fact encompass a far greater number of objects, for instance, the ground node could represent a dirt track or grass. Both of these objects occur in the natural world, but for my application they are just objects that can contain vehicle motion and as such I am not particularly interested in whether the region is a dirt track or grass.

The network clearly serves two of the three general requirements for an image sequence understanding system as defined by Nagel [72]; that is: the system should serve a clearly defined purpose, and the system should be able to recognise explicitly the limits of its capabilities. Nagel's third requirement called for an exhaustive internal representation for all its tasks and environmental conditions it is expected to handle. This would generally result in an extremely complex and difficult to manage database that will tend to defeat the objective of the system by making the computational burden of the recognition and interpretation processes excessive. Knowledge based systems are also faced with the problem that information about the problem domain, the knowledge database, tables 6.1 and 6.2 for example, only provide uncertain knowledge. To this must be coupled the fact that the symbolic data extracted from the image sensor will also be uncertain (illumination conditions, viewing angle, disturbances etc) which also leads systems to mis-classify objects with the consequence that different labels are generated for the same region in the image.

To resolve this problem the network nodes are constructed such that each node has a core process, 'a node kernel'. The node kernel is responsible for maintaining the label consistency of the node, it therefore resolves any label inconsistency. The node kernel also calculates an activity factor for each identified node which gives an indication of the level of motion activity for each identified region in the scene map.

6.4.2.1 Node Kernel.

To accomplish the tasks of maintaining consistent labelling of nodes and estimating the activity of a node, the node kernel uses two core processes, namely the 'Node Label Consistency' process and the 'Node Activity' process.

6.4.2.1.1 Node Label Consistency.

The system is learning the structure of the scene on a frame by frame basis, and has both a short term and long term memory, (figures 6.8(a) and 6.8(b)) and as such does not want any single (or few) mis-classifications to adversely effect the interpretation process. The node consistency process uses temporal filtering to resolve label mis-matches. Initially this can lead to the system mis-classifying segments due to the fact that only a few frames will have been processed. However as the system continues to learn, the average across a wider number of frames will tend to classify a segment as either road or ground correctly.

The label current map function labels the short-term memory with the current label for a map segment as shown in figure 6.8(a). The node kernel scans through the scene map (effectively long term memory, figure 6.8(b)) checking to see if the current map segment has been observed before.

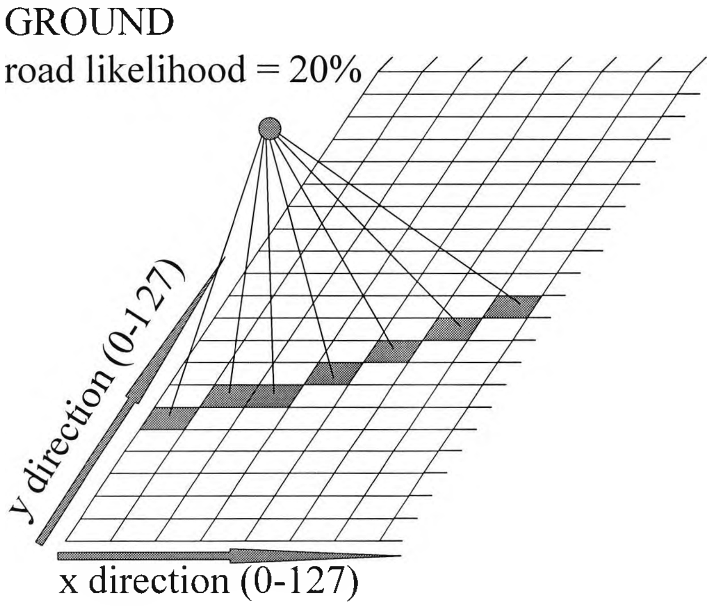


Figure 6.8(a) Section of short term memory.

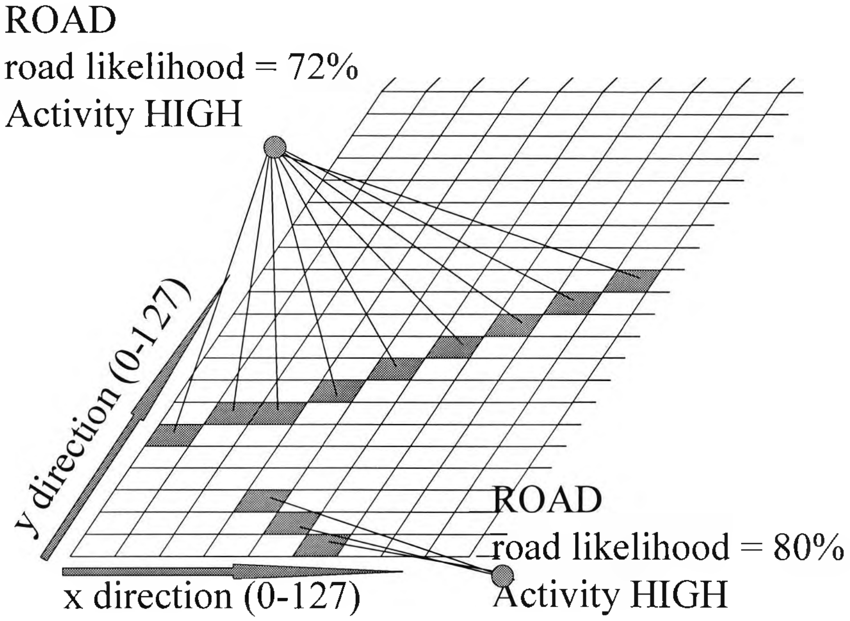


Figure 6.8(b) Corresponding section of Scene Map (Long term memory).

If a co-ordinate clash is detected then the labels for the current and previous map segments are checked for consistency. If the labels are the same then the node consistency process updates the likelihood value adding any extra co-ordinates to the previous node (if required). If the labels are inconsistent then the node consistency process resolves the label conflict by averaging the two likelihood values.

6.4.2.1.2 Node Activity.

The node activity process calculates the node activity based on the ratio of observed segment motion to the number of processed frames (time index). This ratio is used as a look up value into table 6.3, which adds the current motion activity for the labelled node.

If node activity becomes excessive, then this is likely to be due to a tracked object that has entered the field of view of the camera and then stopped at some point in the image, so that its motion is constant at one point. The node activity function now scans the entire current scene map (short term memory) and updates the scene map (long term memory) with the results of the current short term memory, it then clears the short term memory ready for the next frame.

node activity ratio	node activity
0.000 - 0.050	VERY LOW
0.051 - 0.100	LOW
0.101 - 0.150	MEDIUM
0.151 - 0.200	HIGH
0.201 - 0.250	VERY HIGH

Table 6.3 Node activity table.

6.5 Results.

The input image sequence was applied to the system and frame 1 was taken as the reference image (first frame into the system) with reference statistical and edge image data being generated from it. The following pages show frames 1,9,25,29,37 and 60 from the input image sequence, where frame 1 is the initial frame (reference), frame 9 shows a car entering a grass region of the car park and frame 25 shows the car in the grass region manoeuvring to park as another car starts to leave the car park on a road region (entry\exit road).

Frame 29 shows the car that is leaving the car park at the junction to the main road just as another car enters the car park from the road junction, which then becomes occluded behind the car leaving the car park. Frame 37 shows a stream of cars moving down the main road into the cove and the car on the grass finally parking. Finally frame 60 shows another car starting to enter the car park from the main road junction. Frames 1, 9, 25, 29, 37 and 60 are shown in figures 6.9(a, b, c, d, e and f).

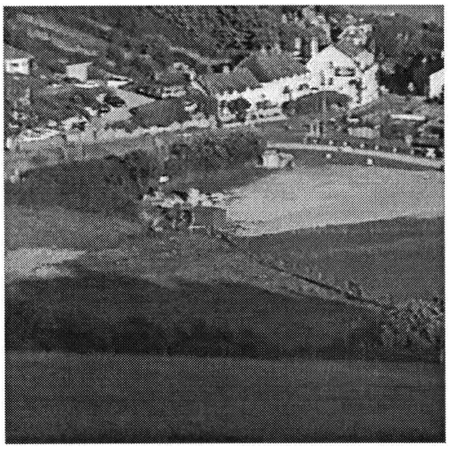


Figure 6.9(a) Input image sequence frame 1.

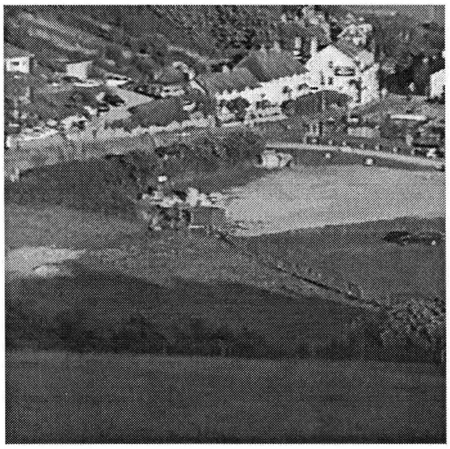


Figure 6.9(b) Input image sequence frame 9.

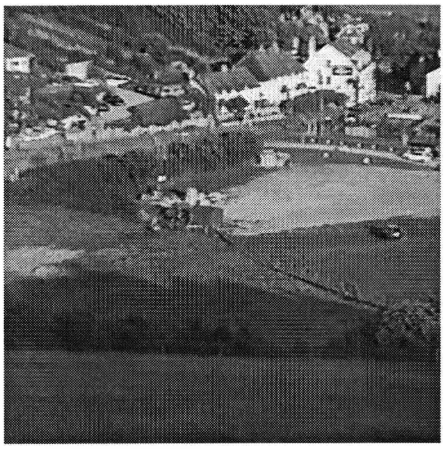


Figure 6.9(c) Input image sequence frame 25.

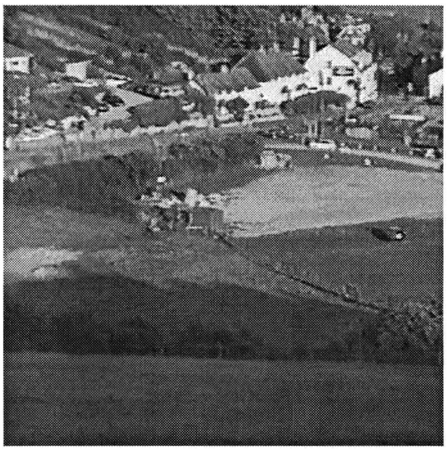


Figure 6.9(d) Input image sequence frame 29.

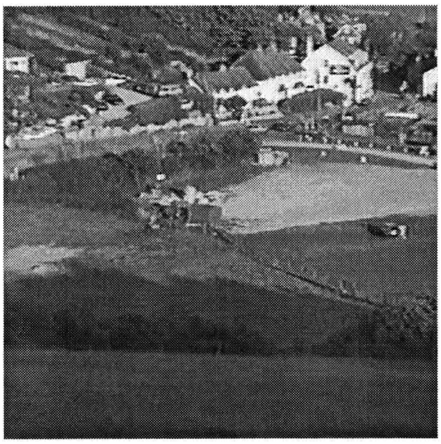


Figure 6.9(e) Input image sequence frame 37.

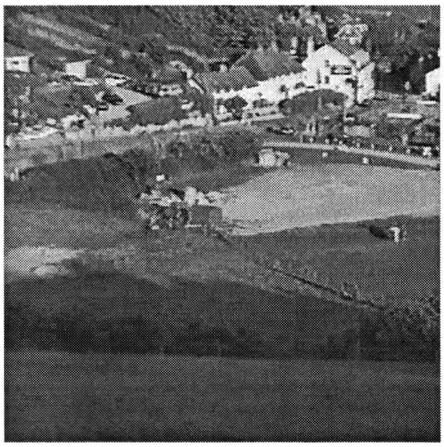


Figure 6.9(f) Input image sequence frame 60.

Figures 6.10(a to f) show the scene map being constructed at those specific instances, i.e. figure 6.10(a) shows the initial scene map, no identified regions, and figures 6.10 (b, c, d, e, and f) show the scene map as it is constructed by the system at frames 9, 25, 29, 37 and 60 respectively.

Figures 6.11(a to f) show the main regions of interest identified by the spatial temporal reasoning process together with the label that the system generated for that region and its corresponding node activity value. Figures 6.11(a, b, c, d, e, and f) show the results of the map building process at frame 60.

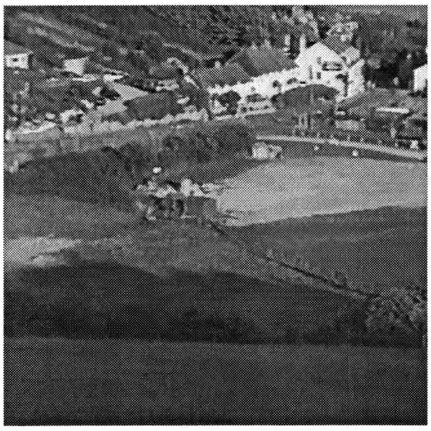


Figure 6.10(a) Scene map construction at frame 1.

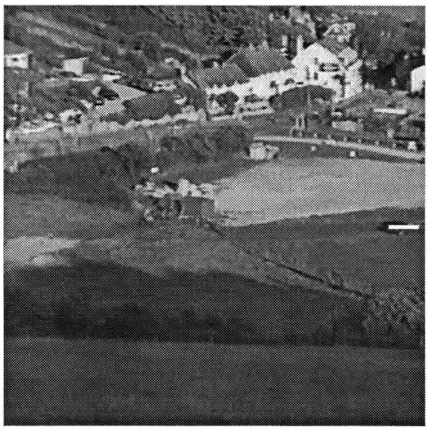


Figure 6.10(b) Scene map construction at frame 9.

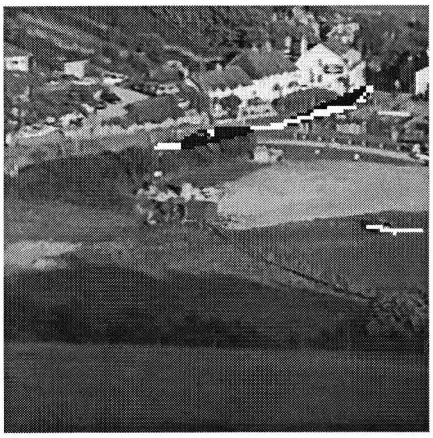


Figure 6.10(c) Scene map construction at frame 25.

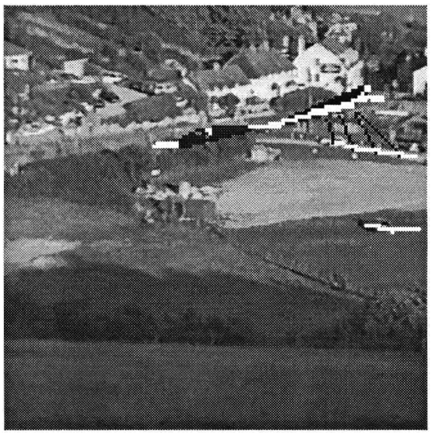


Figure 6.10(d) Scene map construction at frame 29.

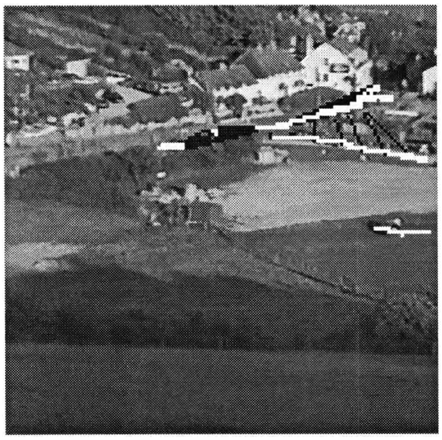


Figure 6.10(e) Scene map construction at frame 37.

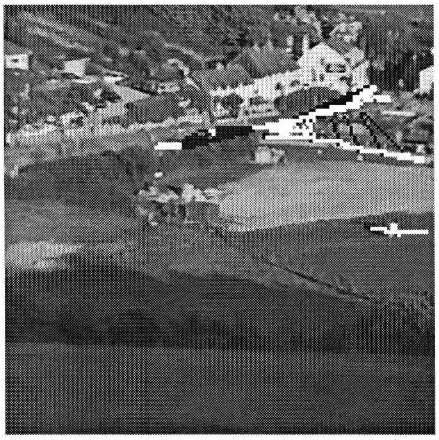


Figure 6.10(f) Scene map construction at frame 60.

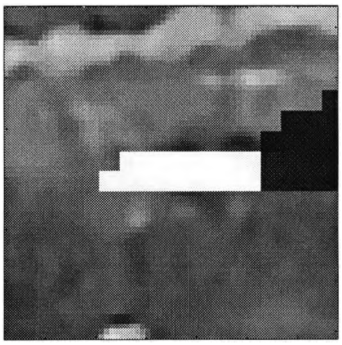


Figure 6.11(a) Ground region.
Region Identified as Ground.
Likelihood Road = 30%.
Node Activity = High.

Figure 6.11(b) Static region.
Regions identified as Static.
Likelihood Road = 0%
Node Activity = Very Low.

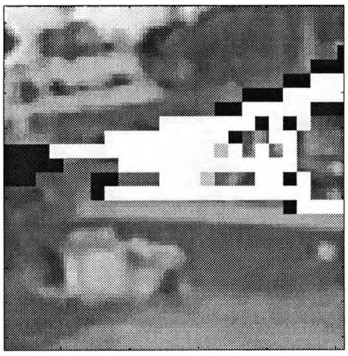


Figure 6.11(c) Road region.
Region Identified as Road.
Likelihood Road = 80%.
Node Activity = High.

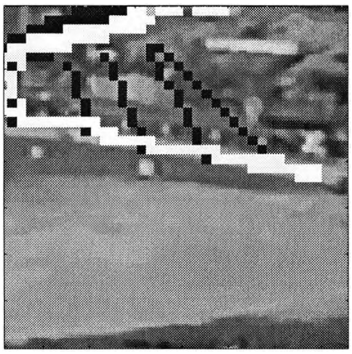


Figure 6.11(d) Road region.
Region identified as Road.
Likelihood Road = 75%
Node Activity = Medium.

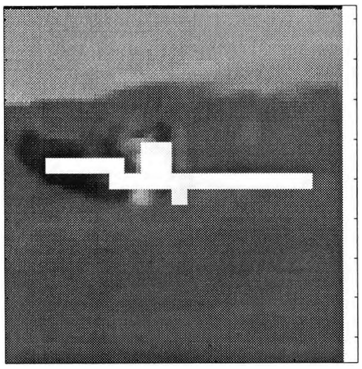


Figure 6.11(e) Ground region.
Region Identified as Ground.
Likelihood Road = 35%.
Node Activity = Low.

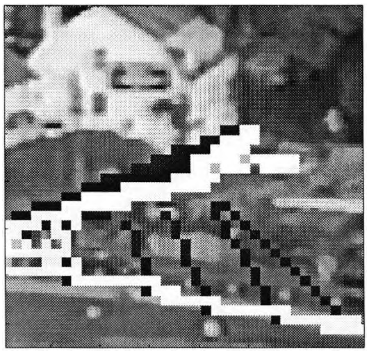


Figure 6.11(f) Ground region.
Region identified as Ground.
Likelihood Road = 30%
Node Activity = Very High.

6.6 Discussion and Summary.

In our daily lives we frequently reason about shapes and how these shapes are arranged as objects in a scene. We use practical reasoning through a variety of levels about how these objects can be manipulated (Fleck, [71]). Understanding of scene structure based on a sequence of images requires very careful selection and management of the information that they offer, as the interpretation of visual data is a classically under estimated problem (Buxton, [70]). The algorithm developed by this research for interpreting the scene structure uses multi-resolution image data and practical reasoning about how objects are arranged in the image to build a structural representation of the scene.

The multi-level representation of the image available to the spatial-temporal reasoning process has two levels. The 1st level consists of edge pixels derived directly from the image using an edge operator and has a resolution of 512 by 512 pixels. The second level consists of target motion data. This motion data has been derived from the image sequence by the target tracker and represents areas of the image that are of interest to the system. They are essentially high level data structures that represent target motion in the image. Stewart [91] highlights that for an image processing system engaged in traffic scene analysis, the system must be focused to areas of interest to reduce the amount of image data that must be processed.

The target data extracted from the image sequence by the tracker effectively focuses the image processing system directly to areas of interest in the scene, a key point for any real-time implementation. The input image sequence used to test the spatial-temporal reasoning process is shown in figures 6.9(a) to 6.9(f) inclusive. These show frames 1, 9, 25, 29, 37 and 60 from the complex outdoor scene used to test the rest of the system. Multiple vehicles can be observed moving into and out of a car park, with some of the vehicles undergoing both full and partial occlusion by both other vehicles and other objects in the scene.

Figures 6.10(a) to (f) show the construction of the scene map at those particular frames, superimposed onto the original image, together with a region label, percentage likelihood of that region being a road and an indication of the level of activity in that region. Figures 6.11(a) to (f) show six enlarged areas of the scene map corresponding to identified structural regions. Figure 6.11(a) shows an actual section of road, that has been identified as ground; this is a mis-classification of the region. However the actual road surface is occluded by the hedge and only partial observation of vehicles lead to this region being mis-identified.

The mis-classification is not deemed to be a problem because the node activity for this region is high. Two main static regions were identified (figure 6.11(b)) showing two distinct areas in the image where tracked vehicles became occluded. Figures 6.11(c) and 6.11(d) show the road junction with the car park entry\exit road has been correctly identified as road. The difference in node activity is due to the fact that a number of vehicles were tracked in the junction region but they did not enter the car park, but in fact went down the road into the cove.

Figure 6.11(e) shows a grass area of the car park where a vehicle entered the scene and parked. The region has been correctly identified and has a low node activity value. With figure 6.11(f) the road down into the cove was mis-classified as being ground; this is due to the road being well over 500 meters away from the camera and as can be seen in the image, it is not completely visible to the camera. Figure 6.11(f) also shows four possible static objects that have been generated using the 'between' arc.

Several vehicles were tracked moving both to and from the cove and as the system processed the tracks using the 'between' premise it identified effectively occluding objects between those tracks and the entry\exit road. In fact this area does not contain occluding objects in the sense that vehicles can disappear behind them and then later re-appear in the image. This region of the image contains buildings and other features that inhibit the motion of vehicles in that region. This indicates that static regions may not just define an area where objects can become occluded, my original aim, but may also represent a region where vehicles may never be expected to be observed moving.

The results obtained by the spatial-temporal reasoning process for this input image sequence correctly identified 67% of the main structural features the system was attempting to learn from the scene. The system successfully identified all areas of the image associated with target motion. It did not identify the large hedge in the image that vehicles were fully occluded behind, due to the fact that motion was never again observed in those regions. The system did however highlight that motion stopped short of the edge of the image, as such it would be possible to extrapolate in a straight line to the edge of the image in a direction based on the last observed target motion. This would identify that region of the image as containing an occluding object.

The mis-classified regions have already been highlighted and are not deemed to cause a problem due to their high level of node activity. The regions have in fact still been identified as being associated with target motion. The results obtained show that using relatively straight forward image processing techniques coupled with a simple reasoning strategy, a map of a complex scene can be constructed. This map identifies areas of the image where vehicle motion is likely to be detected and areas where vehicles can become occluded from the camera but are in fact still in the field of view. This last point is considered to be one of the main results of my research.

Unlike Bouthemy [93] and perhaps to a lesser extent Murray [92] the system developed here is heuristic in nature. This is due to the fact that the observed motion in the image is being derived from vehicles that can be viewed over a wide range of distances from the camera; the consequence of which is that their appearance in the image varies considerably. To this can be added the problem that these objects may be partially occluded in the image as they are being tracked.

The problem with a heuristic system and in general image processing systems that have any form of thresholding decision making processes, is that they can be adapted to work for a specific image sequence. In chapter 7 ‘goal achievement’ two distinctly different complex outdoor traffic scenes are applied to the system; the results of which clearly show that the system is robust and not reliant on a specific type of image scene.

Chapter 7 System Integration.

7.1 Introduction and Overview.

With high-level vision we are generally concerned with constructing some form of model of the world and using this model at a later time for recognition tasks. Chapter 2 gave a brief introduction to high-level vision and knowledge based systems, outlining that such systems use analogical, propositional and procedural models to represent knowledge about objects in the real world and the recognition task is generally realised using inference. Inference does not require the construction of explicit models of an object but instead uses the results obtained from evaluating the analogical and propositional models with extracted image features to deduce the presence or not of an object in the image.

To perform the identification tasks a knowledge based system usually divides this task into a number of main processes, namely:

(a) Feature extraction.
(b) Knowledge acquisition.
(c) Inference.
(d) Planning.
(e) Control.

(a) Feature extraction is concerned with extracting information from the image, (chapters 4 & 5) from which a symbolic description of the objects in the image can be constructed.

(b) Knowledge acquisition is the construction of a model that will represent our knowledge of the real world. This is generally done prior to the start of the recognition process and this previously acquired information is stored in a knowledge data base (experiments carried out in chapters 4, 5 and 6).

(c) Inference is the process by which the knowledge based system deduces (infers) from facts in the knowledge data base and information extracted from the image, the identity and location of objects in the real world (chapters 5 & 6).

(d) Planning is the problem solving and simulation activity that anticipates future world states, it determines how the visual environment is expected to change if certain actions are performed.

(e) Control or control strategies, control the way the high-level vision system performs its processing. These control strategies are important because computation in image processing systems is very expensive and any operations that are not really necessary, should not be performed and the control strategy should ensure this. However it should be noted that even the simplest biological vision systems exhibit sophisticated control of their image processing functions.

The processes just outlined for a knowledge based vision system can be organised into a structure as shown below in figure 7.1.

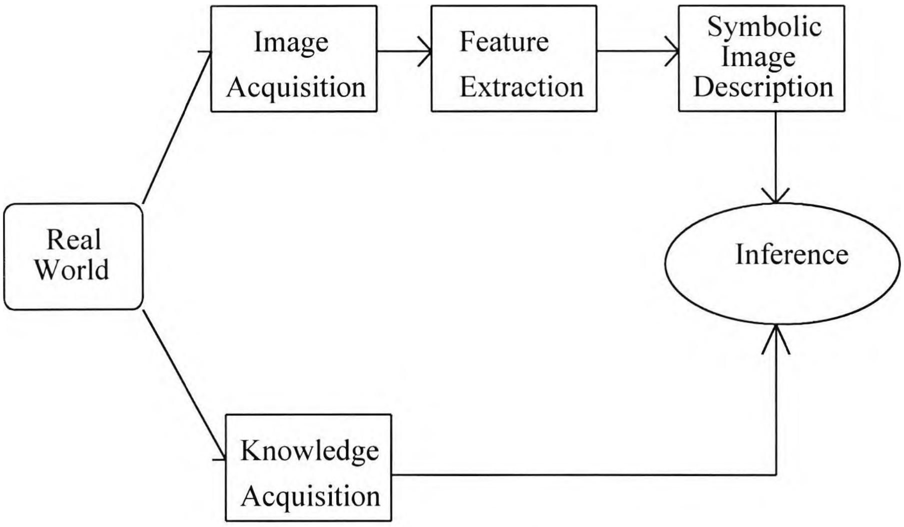


Figure 7.1 Knowledge based vision system.

A vision system using such a knowledge data base is therefore engaged in two distinct activities, namely

(i) belief maintenance.
(ii) goal achievement.

(i) Belief maintenance is a passive, data driven background activity that keeps beliefs consistent and updated through the use of inference.

(ii) Goal achievement is an active, knowledge driven foreground activity that consists of planning the future activities of the system via a planning mechanism.

However the knowledge that we acquire prior to the recognition process (evidence) and the information that we extract from the image (hypotheses) is often uncertain. This uncertainty leads to a probabilistic relationship between the observed image features and the knowledge data base. To this end high-level vision systems tend to use semantic networks, production systems and predicate logic, all of which have logic as their underpinning principle with the reasoning (inference) processes also based on logic.

Generally knowledge based systems when applied to real world images attempt to interpret the structure of the scene using some form of region analysis to segment the image into regions which produces a data set for those regions that contains information about their size, location, edges, colour etc (Draper, [52]). This data is used to generate hypothesis about objects that may be in the scene and control strategies build up and maintain beliefs about objects in the scene in a test and refine manner. Draper [52], highlighted results obtained for a house scene and a road scene, in these examples the major objects in the scenes were identified, but the scene analysis was based on the classification of large scale regions in the image and made no use of object motion. More recent work on the analysis of outdoor scenes has involved the use of neural networks to interpret the major structure features in a scene.

Campbell [53], developed a new segmentation quality metric to segment regions in outdoor scenes using a neural network. The neural network was trained to recognise a set of features within these regions that could be used to classify those regions into sky, roads, buildings, grass etc. The system automatically classified over 81 % of the area of the image into those known regions, but again used static information to interpret the scene. However we live in a dynamic not a static world and Giusto [54], used a knowledge based system to interpret 3-D time varying scenes from an input of 2-D images of simple solid shapes.

Giusto system used colour to recognise the solid shapes and interpretation was obtained by segmenting matching regions to solid models defined by a qualitative view. Analysis of the motion was used to propagate the most likely hypothesis and help solve ambiguities in the system. Johnson, [55] trained a neural network to learn the distribution of object trajectories from an image sequence of a pedestrian scene. The resulting model did not identify grass or sky, but built a representation of the scene where ‘meanings’ could be attached to areas of the image to flag event recognition and trajectory prediction. The results of this work showed that trajectories of objects moving in the image could be learned by the neural network. However the scene is very constrained, with objects that are to be tracked occupying a significant proportion of the image pixels. The system used the motion of objects within the scene to build up an interpretation of that scene, demonstrating that motion alone could be sufficient to interpret structural features within a scene. Each of these systems had their own desired goal, whether it was to identify a house and a road or track an object. There is a considerable amount of literature available on knowledge based systems for analysing static images. The goal is generally identifying objects within that image, the identification process usually relates to areas of the image identified.

7.2 Goal Processor.

As already outlined goal achievement is an active knowledge driven foreground activity which plans the future activities of the system. The goal processor has the task of controlling the overall image processing system such that the processing being carried out is consistent with the aims of the system. To meet these aims, three image processing tasks were identified as being required namely, ‘acquisition and motion detection’, target identification and tracking’ and ‘spatial-temporal reasoning’. These tasks were implemented such that they are totally self contained. That is having been enabled by the goal processor, they require no further attention from it, running until they have performed their processing on a single frame of image data. They then return a status signal when their task is complete.

Planning is therefore relatively straight forward in so far as the goal processor must ensure that each task is called in the correct order and that the status returned by the task indicates that the processing produced no errors. A display function takes the results of the image processing task and displays them in one or more image display windows, textual information is displayed in a message window indicating the status of the processing carried out. If an error occurs the system halts the processing sequence and awaits a user response. The previous three chapters have demonstrated the performance of each image processing task (chapter 4 :-acquisition, filtering and reference generation, chapter 5 :- target identification and tracking and in chapter 6 :- scene map construction). Although the scene used was a particularly difficult open world scene, difficult because the vehicles to be tracked were around 400 meters from the camera and going through occlusion, the system must now demonstrate that it is not scene dependent.

7.3 Scene Analysis.

To test that the image processing system was not scene dependant and that results were not obtained by careful choosing of threshold values, two further open world image sequences were applied to the system without any adjustment to threshold or database values. The scenes were filmed using a static camcorder and the recorded video sequence was later digitised to disc at approximately 10 frames a second for about 40 seconds, giving each sequence a length of 400 frames. The two sequences were chosen such that they are distinctly different from the image sequence used in the development of the image processing system.

7.3.1 Scene 1 Discussion and Results.

Scene one depicts a road traffic junction where the vehicles in the scene are approximately 100 meters from the camera. These vehicles were moving at relatively high speeds up and down the main road (50 mph+) and to add to this, two areas of the junction were partially obscured by trees. While the image sequence was being filmed the camera was placed such that it was undergoing motion due to wind disturbances which would result in random motion cues being generated. Figures 7.2(a) to 7.2(j) show the input sequence at various frames, where vehicles have been identified and tracked, and from the extracted track trajectories the construction of the scene map at those instances.

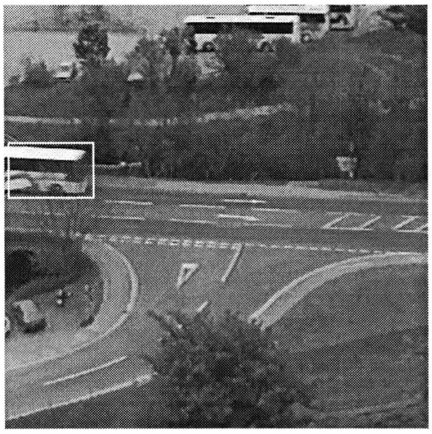


Figure 7.2(a) Frame 6 Tracking Data.

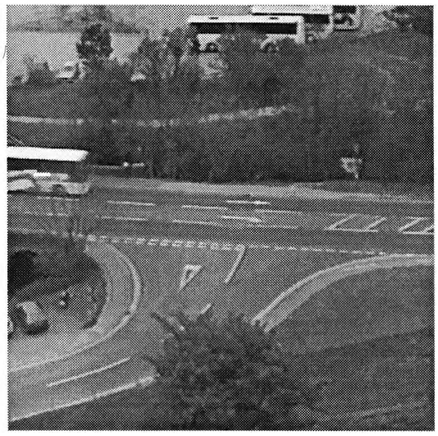


Figure 7.2(b) Frame 6 Scene Map.

Figures 7.2(a and b) show the entry of a coach into the scene and the start of the map construction. The coach was moving at speed (50 mph+) and at one instant a large segment of the image was in motion (nearly 800 image tiles) at one time. Figure 7.2(c) shows that the coach was successfully tracked as it crossed the image sequence. From the extracted trajectory data a road section was identified and inserted into the scene map, which is shown by figure 7.2(d).

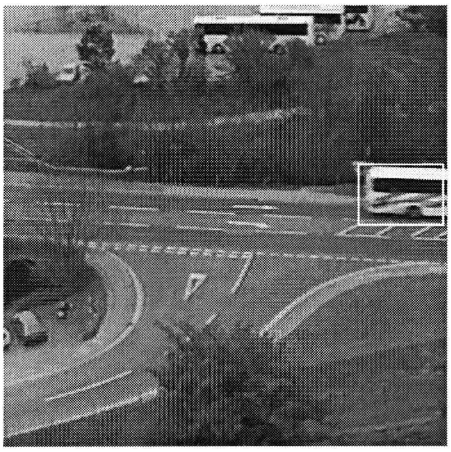


Figure 7.2(c) Frame 28
Tracking Data.

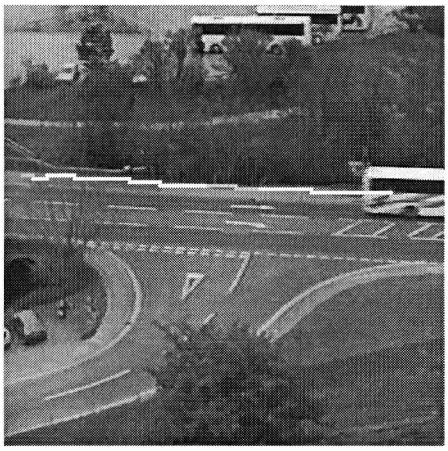


Figure 7.2(d) Frame 28
Scene Map.

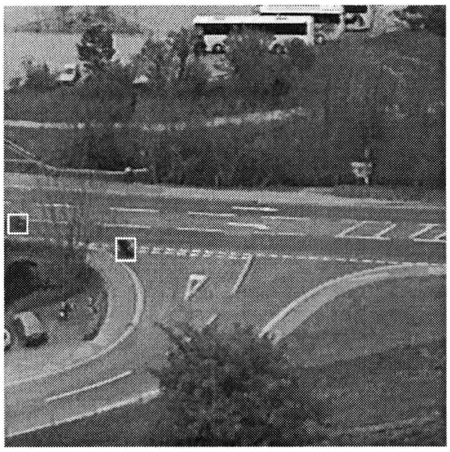


Figure 7.2(e) Frame 91
Tracking Data.

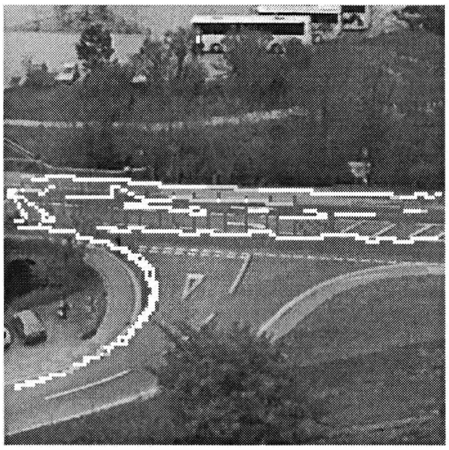


Figure 7.2(f) Frame 91
Scene Map.

Between frames 52 and 108 two motor-cyclists entered the scene from the minor road (lower left hand side of the image). The two motor-cyclists became partially occluded behind the branches of the tree as they entered the main road. Figures 7.2(e and f) show frame 91 from this part of the image sequence, figure 7.2(e) is the results obtained by the system tracker showing that both motor-cyclists have been identified as targets and tracked. Figure 7.2(f) shows that the system has successfully extracted the trajectory data for the two motor-cyclists and constructed the corresponding road segment. Figure 7.2(g) below shows part of the image sequence with four vehicles being identified and tracked. Two are moving left to right on the main road, one is moving right to left and the fourth is moving up the side road to join the main road. In the background the coaches in the car park are loading passengers and getting ready to depart the car park. The resulting tracks have been added to the scene map shown in figure 7.2(h).

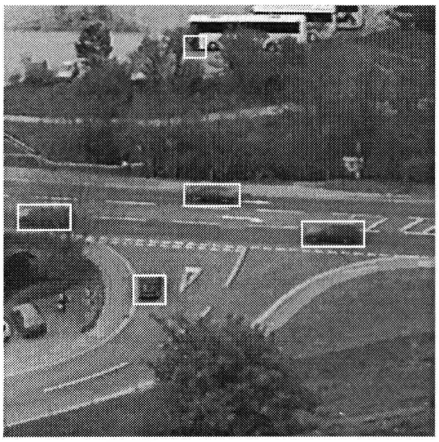


Figure 7.2(g) Frame 265
Tracking Data.

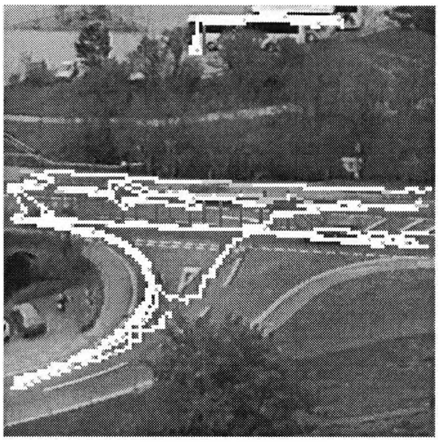


Figure 7.2(h) Frame 265
Scene Map.

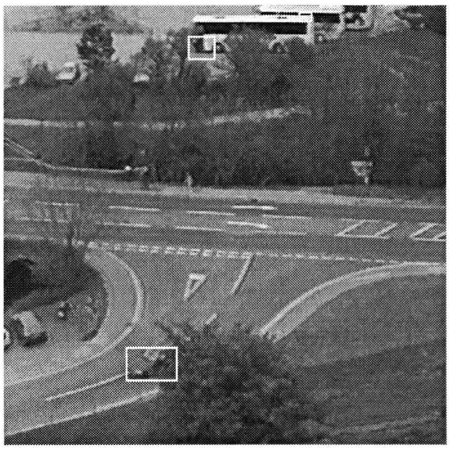


Figure 7.2(i) Frame 377
Tracking Data.

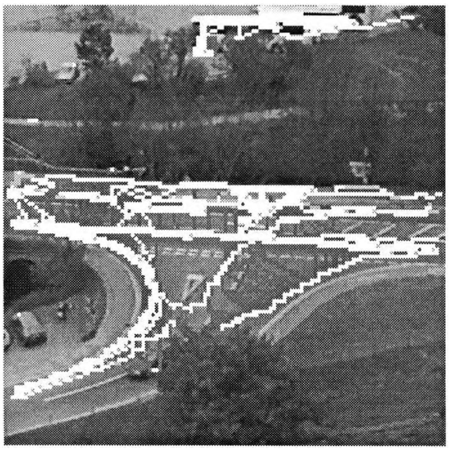


Figure 7.2(j) Frame 377
Scene Map.

Figure 7.2(i) shows frame 377, which is towards the end of the input image sequence. A vehicle has turned right from the main road into the side road and moved down the side road becoming fully occluded behind a tree next to the road. the vehicle became visible (partially) in frame 377, which shows the system re-acquiring the target. Figure 7.2(j) shows the extracted trajectory for this vehicle added to the scene map, an occluding path has been identified by the system in between the last known trajectory point and the cars new trajectory point. Further occluding points were identified by the system across the road, these points although not actual occlusions indicate that no target trajectories have been extracted for this specific region of the scene. The vehicle eventually exited the scene in the lower left hand corner of the image, which the system tracked until the vehicle actually left the image.

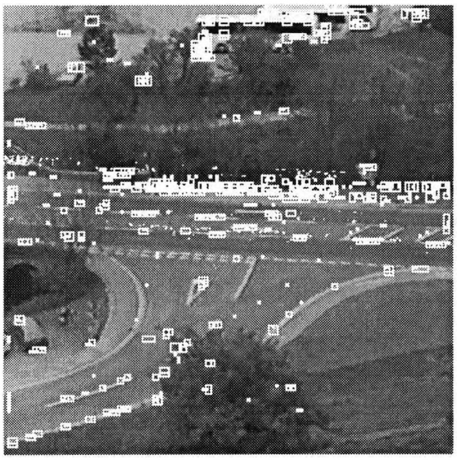


Figure 7.3(a) Un-matched motion cue windows generated between frames 3 and 400.

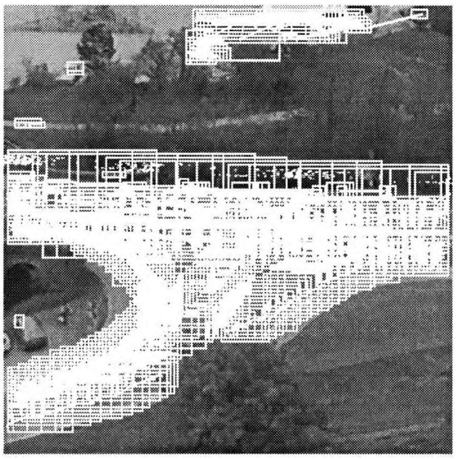


Figure 7.3(b) Tracking windows for matched motion cues between frames 3 to 400.

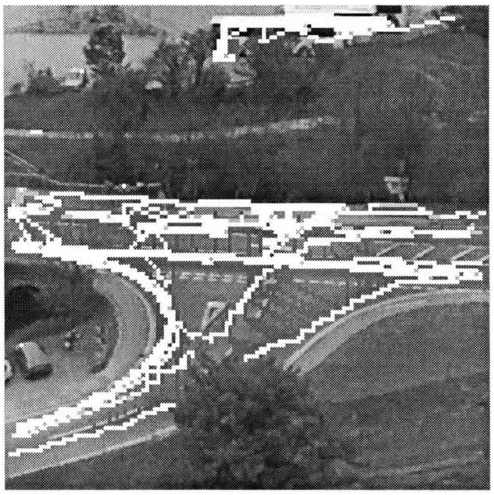


Figure 7.3(c) Constructed scene map for frames 3 to 400.

Figure 7.3(a) shows the un-matched motion cues found across the input image frame sequence. The cues along the sides of both the main and minor roads are due to the camera motion caused by the wind disturbances. This camera motion principally affects regions in the image that have vertical or horizontal edges. However the system has not tracked these points and no false map segments were generated for these regions. Cues generated due to trees and bushes moving in the wind have only been identified and tracked in the background of the image (vertical straight edges of the coaches). Figure 7.3(b) shows the accumulated tracking windows for identified and tracked targets across the image sequence.

Figure 7.3(c) shows the scene map constructed for this 400 frame image sequence, the system came back with an assessment that the scene map construction was incomplete. This is a result of the car that left the image at the bottom of the scene. This vehicle provided new target data as no vehicles had been observed in that region before the end of the frame sequence. Of the identified map segments, 82% were correctly identified as road and 18% were incorrectly identified as ground. Areas identified by the system as potential static occluding objects were on the road. However given more frames with more vehicle motion, then these areas would have been overwritten as road segments. More importantly the actual area of the image where vehicles could undergo total occlusion (e.g. the tree in the bottom centre of the image) was identified as a static object.

The map segments generated by this scene could now be used to directly focus the attention of an image processing system to those areas of the image expected to contain vehicle motion. This would focus the more computationally intensive image processing algorithms required for target identification and tracking to smaller regions of the image, reducing the processing time required to identify and track a vehicle. A second feature of the map is that having identified static objects in the field of view that can occlude vehicles, this feature could enable vehicles to be tracked that although not visible to the camera are still in the field of view.

7.3.2 Scene 2 Discussion and Results.

Scene two depicts a wide open scene with a main road winding uphill into a town. Vehicles moving on the road are travelling at approximately 40 mph and at the top of the hill, just before the road enters the town the vehicles are nearly 1000 meters from the camera. To add to the problem of target range there are a number of trees overhanging the road that either partially or fully occlude vehicles travelling up and down the road.

While filming this image sequence, the camera was placed such that it was not (a far as possible) undergoing motion due to wind disturbances, however being a windy day the trees and bushes in this scene are undergoing motion.

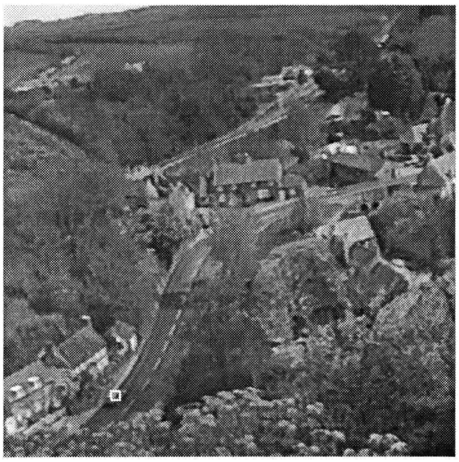


Figure 7.4(a) Frame 7
Tracking Data.

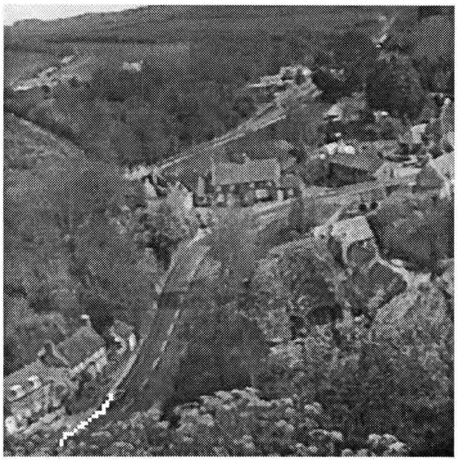


Figure 7.4(b) Frame 7
Scene Map.

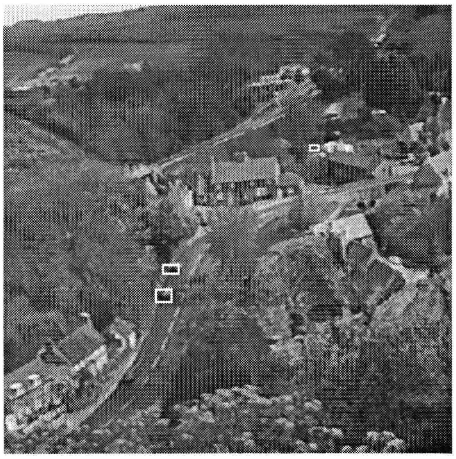


Figure 7.4(c) Frame 50
Tracking Data.

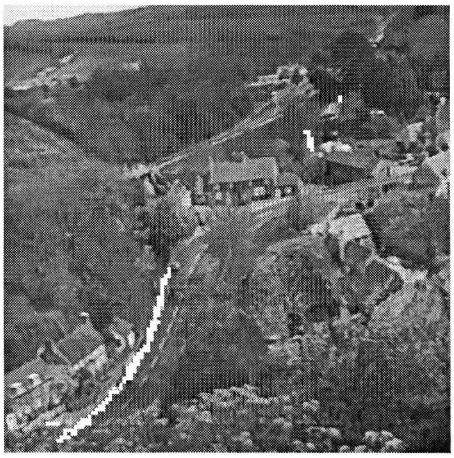


Figure 7.4(d) Frame 50
Scene Map.

Figures 7.4(a and c) above show the entry of a car into the scene, followed by a second car; both vehicles proceeded up the road towards the town. The system has successfully tracked both vehicles as they moved up the road and from the extracted target trajectory data for these vehicles has correctly constructed a road segment for that region of the image (figures 7.4 (b and d)).

However the number of motion cues generated by the scene had slowly risen (see figure 7.6) as the image sequence progressed, with the result that false motion cues were being identified (trees moving next to the house, off centre right in the image) and map segments plotted for them. At frame 123 a new reference set of images was generated by the system.

Figure 7.4(e) below shows frame 160, where there were multiple targets in the scene moving up and down the road. The number of false motion cues has reduced and of the six identified targets only one is a false motion cue (third one up from the bottom of the image). Figure 7.4(f) shows the scene map constructed at this point, and despite the update image references, there are now thirty-five incorrect map segments in the scene map, though only three have a medium node activity value, the remaining thirty-two have low node activity values. The identified road segments all have a high node activity value associated with them.

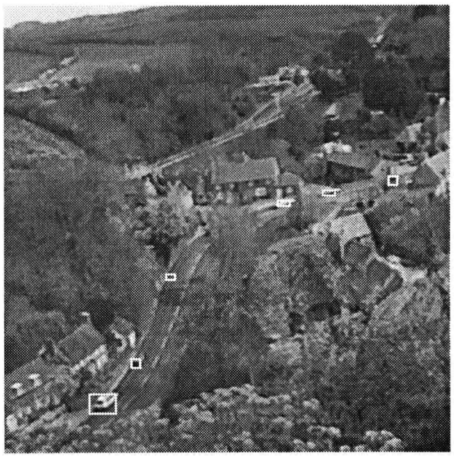


Figure 7.4(e) Frame 160
Tracking Data.

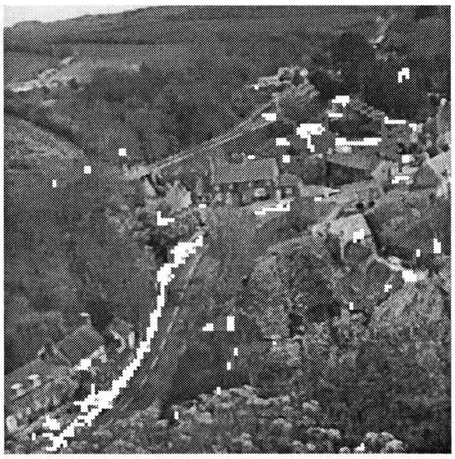


Figure 7.4(f) Frame 160
Scene Map.

The number of false motion cues continued to rise as the image sequence was processed, despite the reference update at frame 123, such that the system reference was again updated at frame 240. Figures 7.4(g and h) on the next page show frame 251, where the system has re-acquired two vehicles that had become occluded by the tree covering the road.

The target at the top of the image was moving towards the town, the other was moving down the road away from the town. The scene map at this time now had forty-seven incorrectly labelled map segments though only five have a medium node activity value, again all correctly identified road segments still have a high node activity value.

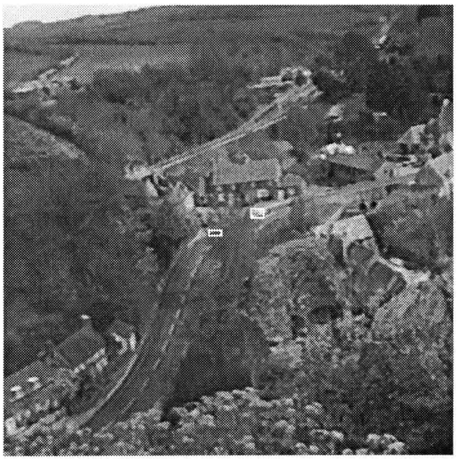


Figure 7.4(g) Frame 251
Tracking Data.

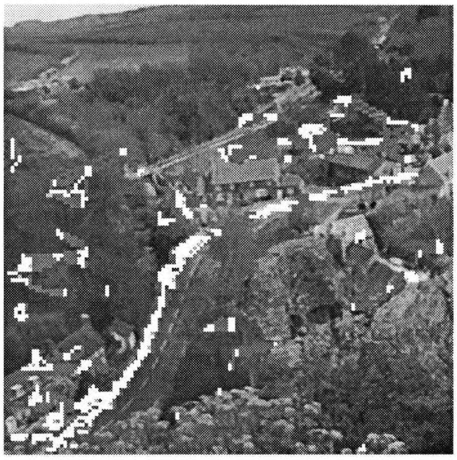


Figure 7.4(h) Frame 251
Scene Map.

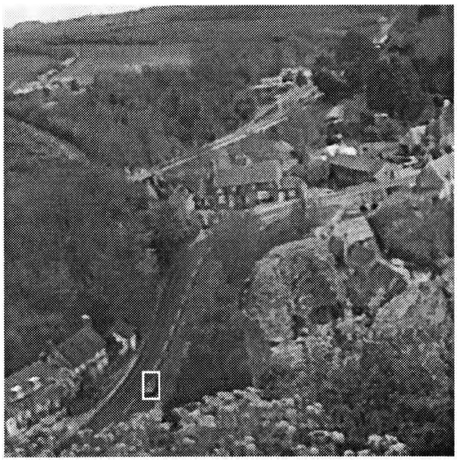


Figure 7.4(i) Frame 367
Tracking Data.

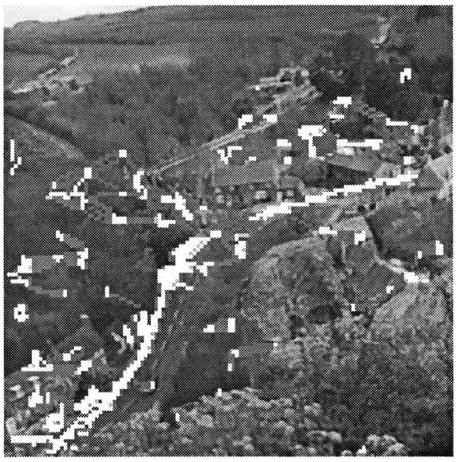


Figure 7.4(j) Frame 367
Scene Map.

The system had at frame 240 generated three sets of image reference data and at frame 345 generated a fourth set of reference data as the number of motion cues generated by the input image sequence had again risen sharply.

Figure 7.4(i) shows frame 367 of the image sequence where a vehicle moving down the road having been occluded by the trees on that side of the road had been re-acquired and tracked. The number of false map segments having risen to fifty one (figure 7.4(j)). The final scene map generated by the system is shown on the next page by figure 7.5.

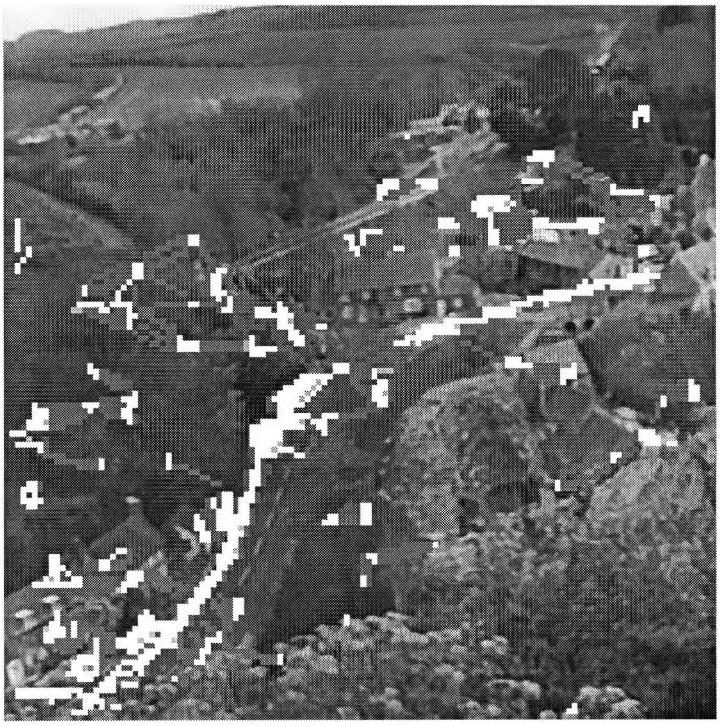


Figure 7.5 Final scene map constructed by the system for image scene 2.

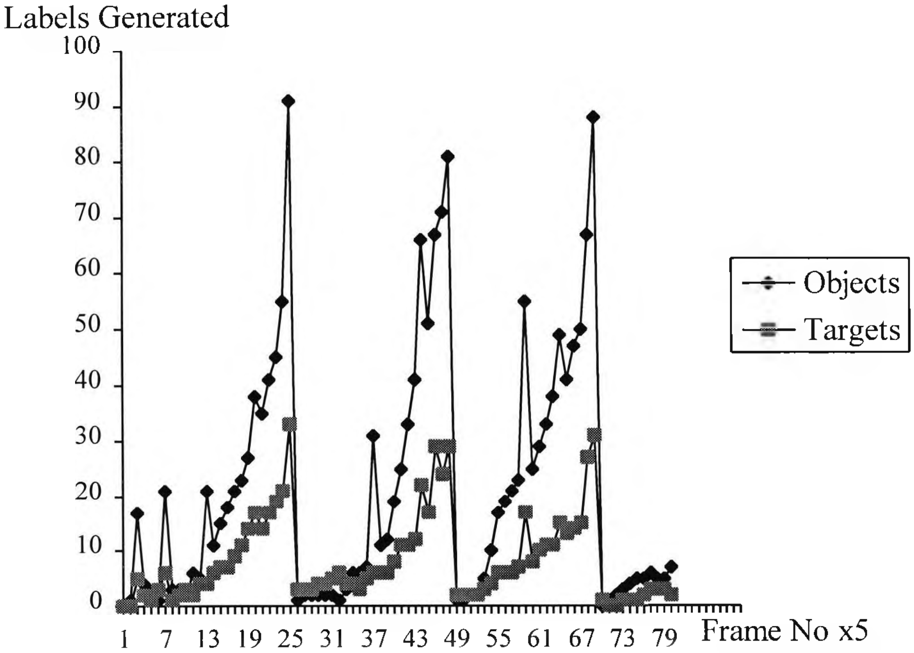


Figure 7.6 Plot showing Object and Target labels generated every fifth frame for image scene 2.

The system has clearly failed to build an accurate map of the scene based on the motion of the man made vehicles moving within that scene. However the map constructed has accurately identified the road regions and assigned a high node activity value to those regions.

It has also correctly identified the occluding trees in the centre of the image that occludes target motion both up and down the road. Of the map segments identified, 57% are correctly labelled, with 43% being incorrectly labelled. The incorrectly labelled regions are all due to false motion cues being generated by the input image sequence. From figure 7.6, it is apparent that the scene is generating a large number of false motion cues, which are only temporally reduced each time a new reference image is generated.

These false motion cues appear to be due to the fact that with this image sequence, approximately 70% of the scene is either trees or bushes which are moving in the wind. These are being interpreted by the system as object motion, the consequence of this, is that the system matches these false motion cues across a number of consecutive frames and extracts corresponding map segments for them.

Having extracted a number of false motion segments when the spatial-temporal reasoning process applies its between premise, static objects are also generated and inserted into the map. Further investigation of the input image sequence revealed that there were in fact bursts of image sequences which had very poor image definition, possibly due to digitisation and JPEG compression effects. The goal processor at the end of the 400th frame reported that the scene map was incomplete, this result is again due to new target data (and false target data) being extracted towards the end of the sequence.

7.4 Discussion and Summary.

It has already been identified (Howarth, [94]) that when interpreting a dynamic and uncertain world, it is important to have a high-level vision system guiding the reasoning processes of your image processing system. The image processing system developed by my research uses low-level image features, in this case motion cues derived from the motion of man made vehicles moving in the scene, to drive high-level vision processes that reason about the structural features within the scene based on this detected motion. The algorithms developed use a hierarchical approach to the target tracking and scene map construction, as it has been found (Jones, [95]) that such methods tend to provide a better temporal stability for moving regions in an image sequence. Chapters four, five and six of this thesis have seen development of an image processing system designed to extract motion data from consecutive frames of image data, identify and track the extracted motion and finally construct a structural representation of a scene based on that motion, (final aim of my research). The goal processor has the task of guiding and controlling the image processing system, checking for errors and reasoning about the state of the structural interpretation of the scene being produced by the system. However all the image processing development work used a single input image sequence.

This image sequence was complex in nature (multiple moving vehicles being either partially, fully occluded), the system was trained on this sequence. To ensure that the system is not scene dependant, then it must be able to construct a structural map of any open world form of traffic image sequence with no *a priori* knowledge of the structure of that scene. Therefore to fully test my system, two further and distinctly different complex open world traffic image sequences were applied.

In scene one the vehicles to be tracked were considerably closer to the camera (approximately 100 meters) and consequently larger areas of the image would be in motion. However the camera was undergoing horizontal motion due to wind disturbances. The system successfully identified and tracked vehicles moving in this scene, from which a structural representation of the scene was constructed. 82% of the extracted map segments were correctly labelled as road, with only 18% of the extracted map segments being due to false motion cues. These false motion cues were generated by undesirable camera motion (environmental disturbances).

Figure 7.3(c) however clearly shows that the system has constructed a map on a frame by frame basis, that identifies the major structural features within that scene. These structural features had either a medium or high node activity value. The false map segments however all had low node activity values.

Scene 2 did not give such good results. However in this scene vehicles were still being tracked despite the fact that some vehicles in the image, (middle left hand side) only occupied an area of 64 pixels (4 tiles). Only the extracted road segments had a high node activity value and only 11% of the false map segments had a medium node activity value, the rest were all low node activity. Considering the poor quality of the input image sequence, the hierarchical method used for motion detection and tracking shows that the key structural feature in the image (the main road) was identified.

The three open world scenes used to test the goal achievement processor were not only complex but together exercised the system across a range of distances that objects were expected to be identified and tracked. They incorporated false motion cues due to both environmental effects (intentional) and digitisation effects (unintentional) and on average depending on the number of false motion cues the system had to process took a little over 2 minutes running on a 100Mhz 486 PC.

Systems have been developed that use multi-resolution motion estimation techniques to segment regions in an image (James, [81]). However although they apply their algorithms to open world scenes with real world noise (Haralick, [103]), they tend to be very scene dependent. I have developed (Teal, [96]) an image processing system that unlike (Cornish, [97]) can construct a scene map of an open world scene without any *a priori* knowledge on the structure of that scene.

Chapter 8 Discussion and Conclusions.

8.1 Discussion.

This research has investigated the complex problem of tracking man made objects moving in open world scenes and based on this motion construction of a representation of the structural features within that scene on a frame by frame basis. The developed system has learned where vehicle motion can be expected without any *a priori* knowledge of the scene. This task has already been identified as an essential component for traffic control algorithms (Kan, [100]).

Two particular features were required of the system, the first was that the tracking should be able to track objects that were a large distance from the camera (over 400 meters) irrespective of the viewing angle. The second feature was that the scene map would identify regions in the image where vehicles could become occluded from the camera but still be in the field of view, (an important feature in a security application, Daniels, [90]).

Analysis of the problem broke the task down into three distinct areas, the first of which was the acquisition of images and the detection of object motion within those images. Due to the long range at which the system was expected to detect object motion, a hierarchical frame differencing technique was employed. Frame differencing has been found (Rosin, [3]) to be a robust method for motion detection when the objects to be detected are a large distance form the camera, however frame differencing techniques require a reference image frame.

Open world scenes tend to produce large numbers of motion cues due to illumination and natural disturbances occurring in the scene. The hierarchical method employed calculates statistical values of fixed four by four pixel regions of the image. This effectively spatially smoothes those areas, reducing the motion generated by small pixel variations. The motion detection was based on the statistical differences between a reference frame (initially the first frame) and the current frame in the image sequence. The decision to update the reference was based on the perceived motion in the image, i.e. the number of identified objects and targets moving. Statistical analysis of the perceived motion across a five frame window was used in conjunction with a decision strategy to update the reference if the perceived motion exceeded pre-set thresholds.

The method has been shown to produce accurate and robust results and can even be modified without degrading system performance to compensate for poor quality image sequences. Having developed a method that can detect object motion in the image, it was necessary to discern which motion is due to objects of interest (targets) and which motion is from other sources (trees, bushes moving etc). The second task was to develop an algorithm that could use some form of metric that would distinguish between the two types of motion.

Due to the large range that targets were expected to be from the camera, model matching methods such as Tan [1] or Koller [17] and more recently Ferryman, [67] would tend to fail as there is insufficient object information in the image to match against a model. It has been found that the matching of crude object descriptors (Teal, [4], Rosin, [3]) provide a more robust form of tracking.

Here a new algorithm has been developed that uses a measure of the change in ‘edginess’ of a region to perform an initial identification of the region. This is used with a correspondence process that matches these regions across a number of frames to identify a target. When the reference image statistics are generated, a reference edge image is also created from the same intensity image. The motion cues extracted are used to create windows into the reference edge image where the amount of edge structure within that region is determined, thus effectively focusing the attention of the image processing system on that region (Meier, [101]).

The window is then placed on the current image and the edge structure within that window is also determined. The initial target identification performed checks to see if there are any differences between the two edge structures. As edge operators are fairly invariant to changes in intensity and changes due to trees and bushes moving would be fairly constant (the bush is still a bush), if a vehicle has moved into that region then we can expect to see a change in the edge structure. Matching these features across frames within certain constraints supports the identification of a motion cue as a target.

The developed algorithm correctly identified and tracked targets to an accuracy of less than 1.5% between the trackers estimated centre of the object and a manually derived centre of the object. The final task developed here was to analyse the target motion and based on that motion construct a structural map of the scene identifying roads and potential objects that could occlude targets as they moved in the image.

A spatial-temporal reasoning algorithm has been developed that extracts the trajectory data based on the target motion and calculates structural features for each of these extracted trajectories. These features form a map segment which is passed to a high-level vision process that reasons (infers) a likely structural interpretation for that map segment.

A semantic network is used to structure the knowledge, and the inference process is implemented using a rule-based approach with the database, rules and the interpreter being embedded into the network arc's and nodes. The possible structural objects permitted in the scene was limited to objects that can contain target motion (roads, ground) and a static object, which is deemed to be anything that can occlude vehicle motion.

The spatial-temporal reasoning algorithm uses the edge structure found in the trajectory of the target together with consistent motion observed to determine if a map segment is either road or ground. It is argued that roads being man made tend to be smooth and as such do not give rise to large numbers of edge pixels unlike a grass surface. It may well be that analysis of straight line segments found within the map segment using a Hough Transform technique (Princen, [98]) may provide a more robust method for identifying the road.

However the algorithm developed here did correctly identify most road surfaces in the image and as already outlined the only time the algorithm actually gave poor results was when the input image sequence had poor quality and even then the major road feature in the scene was correctly identified. The image sequences applied to the system tested the image processing algorithms.

The three image sequences used can have their vehicle motion categorised as 'near', (e.g. scene 1, vehicles 100 meters from the camera); 'medium' (test sequence, vehicles 400 meters from the camera) and 'far' (e.g. scene 2, vehicles up to 1000 meters from the camera). The results obtained by the system showed that even over such a large range of vehicle distances, vehicles could be detected, tracked and a scene map constructed based on the tracking data. The map gives an indication of the likelihood of a segment being correctly labelled, as well as an indication as to the amount of activity that is associated with that region. At present there are three main area's in the system which could be improved upon. The first is the initial target identification. The initial identification uses a measure of the change in edginess of a motion cue region to determine if that cue could be a target. Statistical analysis of the edge magnitudes within the motion cue window in both the current and reference frames may provide a more robust mechanism than simple difference mechanism currently being used.

The second area is the tracking. The tracking algorithm needs to reject most of the false motion cues. This could be addressed by solving the frame to frame correspondence problem across a larger temporal window, i.e. the target must be identified and tracked for say five frames rather than two. Vehicles should exhibit a consistent uniform motion as they move in the image, (this is unlikely to be true for false motion cues).

Finally the analysis of the road and ground segments again uses a measure of edginess of a region to determine its most likely structure. However it was noted that with road traffic scene 1 (chapter 7, section 7.3.1) where the vehicles were closer to the camera, that the edginess for the road approaches values that were found for grass regions in the test sequence. This is due to the fact that the road now occupied a significant part of the image, and the edge detector found large numbers of edges for the road. However it was noted that the road now produced straight line segments which could be used by a Hough transform to determine the presence in the image of a road surface.

8.2 Conclusion.

The aim of my research was to be able to detect, identify and track vehicles moving in an open world scene and based on this motion identify structural features in that scene. This already complex problem was further complicated by the fact that the vehicles to be identified and tracked would likely be a large distance from the camera, and as such there would be very little image information available.

I have found that model based techniques currently being used to identify and track vehicles in open world scenes are robust and can process image sequences at or near real-time. However they would probably fail when the vehicle is a large distance from the camera. I have found that relatively straight forward image processing techniques together with a general knowledge about how vehicles move in their environment can be used to identify and track a vehicle.

I have used the identified motion to reason about the structural representation of the scene. Based on this reasoning process a map has been constructed which identifies areas of the image where vehicle motion can be expected to be observed and perhaps more importantly areas in the scene where vehicles can become occluded from the camera but are still in the filed of view.

The image test sequence used to develop the image processing algorithms was based on an open world scene. The vehicles moving in the scene occupied areas in the image of between 200 and 600 pixels. However I found that my developed algorithms would without modification track vehicles that occupied areas in excess of 16000 pixels down to vehicles that only occupied an area of 64 pixels and still construct a map of the scene based on that motion.

8.3 Future Work.

This research has concentrated on building the scene map. Future work is aimed in two directions. The first is to improve the system performance in target identification and tracking. These improvements have already been highlighted, however tracking may be further improved by using the extracted vehicle trajectory data to resolve loss of target tracking when vehicles occlude one another.

The second is to extend the map construction and use the map to predict when a vehicle could become occluded by static objects in the scene. The results of my work tend to indicate that unless the scene was viewed such that the occluding objects and vehicles were approximately the same distance from the camera (this is in fact pretty well much the case with the three image sequences used here) it would be possible for objects in the scene that are closer to the camera to be observed moving and create (correctly) an area in the map where vehicle motion can be expected. This detected motion could overwrite (incorrectly) a static object that does occlude vehicles moving in the scene that are at a greater distance from the camera.

This problem could be addressed by building not one scene map, but a number of scene maps. Each scene map would be based on the detected target motion area. This would mean that each map effectively represents the scene at a certain range from the camera (assuming that the targets are approximately the same size). The maps would form a 3-D scene description rather than a 2-D one. The maps would be stacked on top of one another (forming a ‘stack’) and as already outlined each map in the stack would be constructed from object motion of different sizes. The size of the object motion would provide a crude measure as to the range of the object from the camera. However rather than using the area of the target as a crude distance measure a second fixed camera could be incorporated. Stereo data could be used in a similar manner to Hanna [99], which would give more accurate target range data for building up a particular level in the map. In either case it is envisaged that extra processing will be needed to solve these problems and incorporate environmental conditions (Gaynor, [102]) that occur in the open world scenes.

References.

[1] Tan T. N, Sullivan G. D and Baker K. D, 'Fast Vehicle Localisation and Recognition Without Line Extraction and Matching', Proc British Machine Vision Conference 1994 Volume one pp 85-94.

[2] A. D. Worrall, R. F. Marslin, G. D. Sullivan, 'Model-based Tracking', K. D. Baker, British Machine Vision Conference 1991, pp 310-318.

[3] Rosin P. L and Ellis T, ‘Detecting and Classifying Intruders in Image Sequences', Proc British Machine Vision Conference 1991, pp 293-300.

[4] Teal M. K and Ellis T. J, 'Target Tracking In Open World Scenes Using Motion Cues and Target Dynamics', IEE 5th International Conference on Image Processing and its Applications , July 1995, pp 276-280.

[5] Baker K.D and Sullivan G. D, 'The Knowledge Based Approach', Proc Alvey Vision Conference 1987, pp 1-4.

[6] Kubovy M and Pomerantz J. R, 'Perceptual Organisation', Published by Lawrence Erilbaum Associates 1981.

[7] Lowe D. G, 'Perceptual Organisation and Visual Recognition', Published by Kluwer, Boston MA 1985.

[8] Lowe D. G, ‘Three-dimensional Object Recognition from single two-dimensional images', Artificial Intelligence 31, 1987, pp 233-235.

[9] Baker K. D and Sullivan G. D, 'The Development of Reasoning Strategies', Proc Alvey Vision Conference 1988, pp 25-30.

[10] Godden R. J, Fullwood J. A. and Hyde J, 'Image Segmentation and Attribute Generation', Proc Alvey Vision Conference 1987, pp 27-32.

[11] Sullivan G. D, 'Performance and Limitations', Proc Alvey Vision Conference 1988, pp 39-45.

[12] Brisdon K, 'Evaluation and Verification of Model Instances', Proc Alvey Vision Conference 1987, pp 33-37.

[13] Radford C. J, 'Vehicle Detection in Open World Scenes Using a Hough Transform Technique', IEE 3rd International Conference On Image Processing and applications 1990, pp 394-399.

[14] Lowe D, 'The Viewpoint Consistency Constraint', International Journal of Computer Vision, Vol 1 1987, pp 57-72.

[15] Du L, Sullivan G. D and Baker K. D, '3D Grouping by Viewpoint Consistency Ascent', Proc British Machine Vision Conference 1991, pp 45-53.

[16] Shen X and Hogg D, '3-D Shape Recovery Using A Deformable Model', Proc British Machine Vision Conference 1994, Volume 2, pp 387-396.

[17] Koller D, Daniilidis & Nagel H-H, 'Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes, International Journal of Computer Vision, 10:3, 1993, pp 257-281.

[18] S. Brock-Gunn and Tim Ellis, 'Using Colour Templates for Target Identification and Tracking', Proc British Machine Vision Conference 1992, pp 207-216.

[19] P.D. Picton, 'Tracking and Segmentation of moving objects in a scene', IEE 3rd International Conference On Image Processing And Applications 1990, pp 389-93.

[20] G.E. Schneider, 'Science, 163', 1969, pp 895-902.

[21] J. Kittler, 'Image and Vision Computing 1',1983, pp 37-42.

[22] Gong S, Buxton H, 'From contextual knowledge to computational constraints', Proc British Machine Vision Conference 1993, Volume 1, pp 229-238.

[23] Li-Qun X, Hogg D, 'Building a Model of a Road Junction Using Moving Vehicle Information', Proc British Machine Vision Conference 1992, pp 443-452.

[24] Scott G. L and Longuet-Higgins H. C, 'An algorithm for associating the features of two frames', Proc Royal Soc. London B 1991; 244: pp 21-26.

[25] Stat T. M, Fischler M. A, 'Context-Based Vision: Recognising Objects Using Information from Both 2-D and 3-D Imagery', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 13, No 10, October 1991, pp 1050-1065.

[26] Toal A. F, Buxton H, 'Spatio-temporal Reasoning within a Traffic Surveillance System, 2nd European Conference on Computer Vision 1992, pp 885-892.

[27] G. Sullivan, Z. Hussain, R. Godden, R. Marslin and A. Worrall, Technical report D102, 'Knowledge Based Image Processing', Esprit-II P2152, 'VIEWS', 1990.

[28] Ballard D, 'Reference frames for animate vision', In IJCAI, 1989.

[29] Ohta Yu-ichi , Kanade T. and Saki T, 'Colour Information for Region Segmentation', Computer Graphics and Image Processing 13, 1980, pp 222-241.

[30] Draper B. A, Collins R. T, Brolio J, Hanson A. R and Riseman E. M, 'The Schema System', International Journal of Computer Vision, 2, 1989, pp 209-250.

[31] Deriche R & Faugeras O. D. ' Tracking line segments', Image and vision computing, Vol 8, 1990 pp 261-270.

[32] Rees D. G, 'Essential Statistics, Second Edition', Chapman & Hall, ISBN 0-412-32030-4.

[33] Jain R, 'Dynamic Scene Analysis using pixel based processes', IEEE Computer, 1981, pp 12-18.

[34] Marr D and Hildreth E, 'Theory of Edge Detection', Proc R Soc Lond, B 207, 1980, pp 187-217.

[35] Jain R, Difference and Accumulative Difference Pictures In Dynamic Scene Analysis', Image and Vision Computing, 1984, Vol 2, pp 99-108.

[36] Kittler J & Illingworth J, 'Minimum Error Thresholding', Pattern Recognition, 1986, Vol 19, No 1, pp 41-47.

[37] Long W & Yang Y. H, 'Stationary Background Generation: AN Alternative To The Difference Of Two Images', Pattern Recognition, 1990, Vol 23, No 12, pp 1351-1359.

[38] Brofferio S, Carnimeo L, Comunale D, Mastronardi G, ‘ A Background Updating Algorithm For Moving Object Scenes’, Time-Varying Image Processing and Moving Object Recognition 2, 1990, pp 297-307.

[39] Otsu N, ‘ A Threshold Selection Method from Grey Level Histograms, IEEE Transactions on Systems, Man and Cybernetics, Vol, SMC-9, No 1, January 1979, pp 63- 66.

[40] Kollnig H, Nagal H-H, ‘3D Pose Estimation by Fitting Image Gradients Directly to Polyhedral Models’, Proc of IEEE, 0-8186-7042-8/95, pp 569-574.

[41] Kilger M, ‘ A Shadow Handler in a Video-Based Real-Time Traffic Monitoring System, Proc IEEE Workshop on Applications of Computer Vision, Palm Springs/CA, 1992, pp 11-18.

[42] Dubusson M. P, Jain A. K, ‘ Contour Extraction of Moving Objects in Complex Outdoor Scenes’, International Journal of Computer Vision 14:1, 1995, pp 83-105.

[43] Sonka M, Hlavac V and Boyle R, 'Image Processing Analysis and Machine Vision', Chapman and Hall, ISBN 0-412-45570-6.

[44] Pitas I, 'Digital Image Processing Algorithms', Prentice Hall, ISBN 0-13-145814-0.

[45] Worrall A. D, Sullivan G. D, Baker K. D, 'Advances in Model-based Traffic Vision', Proc British Machine Vision Conference, 1993 Volume 2, pp 559-568.

[46] Daum F. E, 'Parallel Prefix and Data Association', SPIE Vol 1096 Signal and Data Processing of Small Targets 1989, pp 174-186.

[47] Rosin P. L, Ellis T, ‘ Image difference threshold strategies and shadow detection’, Proc British Machine Vision Conference, 1995, Vol 1, pp 347-356.

[48] Malik J, Weber J, Luong Q-T & Koller D, ‘Smart Cars and Smart Roads’, Proc British Machine Vision Conference, 1995, Vol 2, pp 367-381.

[49] Davis A. C, Yin J. H, Velastin S. A, 'Crowd monitoring using image processing', IEE Electronics & Communication Journal February 1995, pp 37-47.

[50] Scanlan J. M., Chabries D. M and Christiansen R. W, ‘ A shadow Detection and Removal Algorithm for 2-D Images’, Proc IEEE, ICASSP, 1990, pp 2057-2060.

[51] 'Panel Discussion: Problems and Solutions in tracking in a dense environment', SPIE Vol 1096 Signal and Data Processing of Small Targets 1989, pp 140-143.

[52] Draper B. A., Collins R.T, Brolio J, Hanson A. R, ‘ The schema System’, International Journal of Computer Vision, 2, pp 209-250, 1989.

[53] Campbell N. W, Mackeown W. P. J, Thomas B. T, Troscianko T, ‘Automatic Interpretation of Outdoor Scenes’, Proc British Machine Vision Conference, Vol 1, 1995, pp 297-305.

[54] Giusto D. D, Milano A, Perotti F, Serpico S. B and Vernazza G, ‘ Integration of motion analysis and model-based interpretation for 3-D scene understanding’, Time-Varying Image Processing and Moving Object Recognition 2, 1990, pp 308-315.

[55] Johnson N, Hogg D, ‘ Learning the distribution of Object Trajectories for Event Recognition’, Proc British Machine Vision Conference, Vol 2, 1995, pp 583-592.

[56] Roberts J and Charnley, ’Attentive Visual Tracking’, Proc British Machine Vision Conference, Vol 2, 1993, pp 459-468.

[57] Zhang Xu, ‘Computation of Vehicle Trajectories Using a Neural Network’, Proc British Machine Vision Conference, Vol 2, 1993, pp 489-498.

[58] J. Canny, ‘A computational approach to edge detection’, IEEE transactions, PAMI, 1986 Vol 8, pp 679-698.

[59] Cowley J.L, Stelmaszyk, Discours C, 'Measuring Image Flow by Tracking Edge-Lines', Proc 2nd Int Conf Computer Vision, Dec 1988, pp 658-664.

[60] Deriche R, Faugeras O, ' Tracking Line Segments', Proc 1st European Conf on Computer Vision, April 1990, pp 259-268.

[61] Gordon G. L, ' On the tracking of featureless objects with occlusion', Proc Workshop on Visual motion, March 1989, pp 166-172.

[62] Torr P. H. S, Beardsley P. A, Murray D. W, ' Robust Vision', Proc British Machine Vision Conference, Vol 1, 1994, pp 145-154.

[63] Borida T. J Chella,' Estimating the kinematics and structure of a rigid object from a sequence of monocular images', IEEE Trans. PAMI-9, Vol 13, No 6: pp 497-513, June 1991.

[64] Morton S. K, 'Object Hypothesis by Evidential Reasoning', Proc Alvey Vision Conference 1987, pp 15-26.

[65] R. J. Howarth, H. Buxton, ' Analogical representation of spatial events for understanding traffic behaviour', 10th European Conference on Artificial Intelligence, 1992, pp 785-789.

[66] Lacey A.J, Thacker N. A and Seed N. L, ' Feature Tracking and Motion Classification Using A Switchable Model Kalman Filter', Proc British Machine Vision Conference, Vol 2, 1994, pp 599-608.

[67] Ferryman J. M, Worrall A. D, Sullivan G. D and Baker K. D, ' A generic deformable model for vehicle recognition', Proc British Machine Vision Conference, Vol 1, 1995, pp 127-136.

[68] Worrall A. D, Ferryman J. M, Sullivan G. D and Baker K. D, ' Pose and structure recovery using active models', Proc British Machine Vision Conference, Vol 1, 1995, pp 137-146.

[69] Caplier A, Luthon F, 'Spatio-Temporal Multi-resolution Associated to MRF Modelling For Motion Detection', IEE 5th International Conference on Image Processing and its Applications , July 1995, pp 158-162.

[70] Buxton H & Walker N, 'Query based visual analysis: spatio-temporal reasoning in computer vision', Image and vision Computing, VOL 6, No 4, November 1988, pp 247-254.

[71] Fleck M. M, 'Representing space for practical reasoning', Image and Vision Computing, VOL 6, No 2, May 1988, pp 75-86.

[72] H-H Nagel. 'From image sequences towards conceptual descriptions', Image and Vision Computing, VOL 6, No 2, May 1988, pp 59-74.

[73] Burt P. J 1984, 'The pyramid as a structure for efficient computation', Multi-resolution image processing analysis, Rosenfeld A, Ed Berlin : Springer-Verlag, pp 6-35.

[74] Deruyver A, Hode Y, ' True three dimensional image labelling: Semantic graph and arc consistency', IEE 5th International Conference on Image Processing and its Applications , July 1995, pp 105-108.

[75] Rowe S and Blake A, ' Statistical Background Modelling for tracking with a Virtual Camera', Proc British Machine Vision Conference, Vol 2, 1995, pp 423-432.

[76] Hutber D and Sims P.F., ' Use of Machine Learning to Generate Rules, Proc Alvery Vision Conference 1987, pp 5-13.

[77] Bers K, Koop B, 'Evaluation of tracker performance', Proc of SPIE, Optical Engineering, 1994, Vol 2235, pp 650-660.

[78] Hutchins R. G., 'Image Enhanced passive tracking of manoeuvring targets', Proc of SPIE, Optical Engineering, 1994, Vol 2235, pp 594-600.

[79] Nieman H, Sagerer G. F, Schroder S and Kummert F,' ERNEST: A Semantic Network System for Pattern Understanding', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 9, 1990, pp 883-905.

[80] McCarthy and Hayes, 'Some philosophical problems from the standpoint of artificial intelligence', B. Meltzer and D. Ritchie Eds, Machine Intelligence 4, Edinburgh University Press 1969, pp 463-502.

[81] James P. D and Spann M, ' Multiresolution Motion Estimation/ Segmentation incorporating Feature Correspondence and Optical Flow, Proc British Machine Vision Conference, Vol 2, 1995, pp 593-602.

[82] Meyer F and Bouthemy P, 'Region-Based Tracking in an Image', Sequence', Lecture Notes in Computer Science, Vol 588, pp 476-484.

[83] Brachman, ' On the epistemological status of semantic networks', N. V. Finder Ed, Associative Networks, New York Academic 1979.

[84] Ballard D. H, Brown C. M, 'Computer Vision', Prentice Hall, ISBN 0-13-165316-4.

[85] Sobttka K and Wetzel D, ' Attention Control Integrated in a System to Autonomous Driving and Collision Avoidance under Egomotion', IEE 5th International Conference on Image Processing and its Applications , July 1995, pp 796-800.

[86] Karmann K. P and Brandt A, 'Moving Object Recognition Using an Adaptive Background Memory', Time-Varying Image Processing and Moving Object Recognition 2, 1990, pp 289-296.

[87] Lampton C, Flights of Fantasy', Waite Group Press, 1993, ISBN 1-878739-18-2.

[88] Sucar L. E. and Gillies D. F, 'Probabilistic reasoning in high level vision', Image and Vision Computing Vol 12 No 1, January/February 1994, pp 140-143.

[89] Johnson-Laird, P. N, 'Mental models in cognitive science', Cognitive Science 4, January-March 1980, pp 75-115.

[90] Daniels D. J, 'Surface Penetrating Radar as an aid to search Operations', IEE European Convention on Security and Detection, May 1995, pp 293-300.

[91] Stewart B. D, Reading I. A. D, Thomson M. S, Wan C. L and Binnie T. D, ' Directing Attention for traffic scene Analysis', IEE 5th International Conference on Image Processing and its Applications , July 1995, pp 801-805.

[92] Murray D.W and Buxton B. F, 'Scene Segmentation from Visual Motion Using Global Optimisation', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No 2, 1987, pp 220- 228.

[93] Bouthemy P, Francois E, ' Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence, International Journal of Computer Vision, 10:2, 1993, pp 157-182.

[94] Howarth R, 'Interpreting a dynamic and uncertain world: high-level vision', Artificial Intelligence Review, Vol 9, Iss 1, 1995, pp 37-63.

[95] Jones G. A, 'Motion detection in security applications using tracking and hierarchy', Proceedings of the SPIE, Applications of Digital Image Processing XVII, Vol 2298, 1994, pp 479-488.

[96] Teal M. K, 'Spatial-Temporal Reasoning Based On Object Motion', Proc British Machine Vision Conference, Vol 2, 1996, pp 465-474.

[97] Cornish M. T, 'Automatically Locating an Area of Interest and Maintaining a Reference Image to Aid the Real-Time Tracking of Objects', Proc British Machine Vision Conference, Vol 2, 1996, pp 475-484.

[98] Princen J, Yuen H. K, Illingworth J, Kittler J, 'A comparison of Hough Transform methods', IEE 3rd International Conference On Image Processing & Applications 1990.

[99] Hanna K. J, Okamoto N. E, 'Combining stereo and motion analysis for direct estimation of scene structure', Proc Fourth International Conference on Computer Vision, 1993, pp 357-365.

[100] Kan W. Y, Krogmeier J. V, Doerschuk P C, ' Sensor signal processing for IVHS applications', 1995 International Conference on Acoustics, Speech and Signal Processing, Vol 4, pp 2683-2686.

[101] Meier W, Vom Stein H. D, 'Statistical analysis of infrared image sequences', Proceedings of the SPIE, ISPRS Commission III Symposium Spatial Information from Digital Photogrammetry and Computer Vision, 1994, Vol 2357, Iss pt2, pp 562-568.

[102] Gaynor W, Moore C, Coppola M, Bassingthwaite J, 'Target identification and sensor performance (TISP) imagery-based targeting trainer', Proceedings of the SPIE, Infrared Imaging Systems: Design, Analysis, Modelling and Testing V, 1994, Vol 2224, pp199-205.

[103] Haralick R. M, 'Performance Charateristerization in Computer Vision', CVGIP : Image Understanding, 60, 2, 1994, pp 245-249.

Bibliography.

Borland C++ DOS Reference.

Stevens R. T, 'The C Graphics Handbook', Academic Press, ISBN 0-12-668320-4.

McCord J. W, 'Borland C++ 3.1 Programmers Reference', 2nd Edition, QUE, Programming Series, ISBN 1-56529-082-8.

Peterson M, 'Borland C++ Developers Bible', The WAITE GROUP, ISBN 1-878739-16-6.

Swan T, 'Mastering Borland C++', SAMS, ISBN 0-672-30274-8.

Lafore R, 'Object-Oriented Programming in TURBO C++', WAITE GROUP Press, ISBN 1-878739-06-9.

Goldsmith S, 'A practical Guide to Real-Time Systems Development', Prentice Hall, ISBN 0-13-718503-0.

SELECT Case Tool Reference Book.

Gilhooly K. J, 'Human And Machine Problem Solving', Plenum Press, ISBN 0-306-42962-4.

Duda R. O, Hart P. E, 'Pattern Classification and Scene Analysis', Wiley-Interscience, ISBN 0-471-22361-1.

Schalkoff R. J, 'Digital Image Processing and Computer Vision', Wiley, ISBN 0-471-50536-6.

Gonzalez R. C and Wintz P, 'Second Edition, 'Digital Image Processing', Addison Wesley, ISBN 0-201-11026-1.

Jain R, Kasturi R, Schunck B. G, 'Machine Vision', McGraw Hill, ISBN 0-07-032018-7.

Haralick R. M, Shipiro L. G, 'Computer and Robot Vision Volume I', Addison Wesley, ISBN 0-201-10877-1.

Haralick R. M, Shipiro L. G, 'Computer and Robot Vision Volume II', Addison Wesley, ISBN 0-201-56943-4.

Yourdon E. (1985) Structured Walkthroughs, Prentice Hall, (Yourdon Press), Hemel Hempstead.

Yourdon E. (1985) Modern Structured Analysis, Prentice Hall, (Yourdon Press), Hemel Hempstead.

Appendix A YOURDON Structured Analysis and Design Methodology.

A1: YOURDON structured analysis and design methodology.

The YOURDON structured analysis and design methodology addresses both structured analysis and structured design, though each of these processes is dealt with independently. Structured analysis is addressed by the 'essential model' and structured design by the 'implementation model', each model attempts to describe several aspects of the system.

A2: Essential Model.

The essential model is built up from two other models, namely, the environmental model and the behavioural model.

1:- Environmental Model, this model is used to describe the external systems which will interface to the system being developed.

2:- Behavioural Model, this model is used to describe how the system will behave in response to events from its environment.

A3: Implementation Model.

The implementation model consists of three other models, namely, the processor model, the task model and the module model.

1:- Processor Model, this model allocates functions from the requirements to real processors within the system.
2:- Task Model, this model describes the allocation of the functions to the tasks on each processor.
3:- Module Model, this model is used to describe the internal structure of each task. It can also define the processing and control aspects of the module, which is known as the 'program implementation model'.

It is important to note that there is no rule which forces you to use both the essential model and the implementation model. Each individual system has its own level of complexities and restrictions, therefore each project will have to identify the optimum set of models to use. However several of the tools which are used to create both the essential model and implementation model make use of the same technique, and it may be that in small projects the essential model may well be developed into the implementation model. To implement the YOURDON structured analysis and design methodology a number of tools are used to create the models.

A4: Data Flow Diagrams.

As information moves through the system, it is modified by a series of transformations. A Data Flow Diagram (DFD) is a graphical technique that depicts information flows and transformations to those flows as data moves from input to output; DFD's can also be known as a data flow graphs, or bubble charts. Data Flow Diagrams may be used to represent systems or software at any level of abstraction. The highest level, level 0 is called a 'fundamental system model' or a 'context diagram'. DFD's are organised into levels 1, 2, 3 .... etc where these levels are derived from a data transformation of the previous level, i.e. level 1 is derived from level 0, (there will only be one level 1, but there may be several level 2 diagrams derived from level 1).

Each of these diagrams expands further the data and processing requirements of the transform from which they have been derived (function de-composition). The basic principles used to draw any data flow diagram rely on defining the scope of the system by naming the flows that enter and leave the system highlighting the interfaces between the system and the outside world. Terminators are used at the highest level to represent the outside world; they produce the flows that the system will process and accept flows that the system has produced in response to the input flows. However DFD's do not illuminate:

1. The organisation of data, though data is shown.
2. The dynamics of the system.
3. The processing carried by the data transform.

To address these problems several other tools are used, namely:

1: Data Dictionary.
2: State Transition Diagram
3: Process Specification.

A5: Data Dictionary.

A DFD shows the flow of data in the system and the transformations that can occur to that data, but a data item may be made up from a collection of individual data items which the DFD does not show. The data dictionary sometimes called a requirements dictionary, is an organised listing of all data elements that are pertinent to the system together with a textural description of structure and organisation of each data item.

A6: State Transition Diagram.

As already mentioned the DFD does not model the behaviour (dynamics) of a system; to do this we use a State Transition Diagram. STD's model the dynamics of a system as a network of states connected by transitions. The main task of an STD is to highlight the time-dependent behaviour of the system; accordingly this model is important for the development of real time systems. State Transition Diagrams are described using four types of component :

(i) States.
(ii) Transitions.
(iii) Conditions.
(iv) Actions.

(i) States are the foundation of the diagrams, but unfortunately the most difficult components to work out. Formally a state is an abstract concept which contains enough information to determine the future behaviour of the system.

(ii) Transitions represent movement (sequence) from one state in the system to another and have two consequences, namely:

(1) They cause the actions to happen.
(2) They may change the state of the machine.

In the implementation model, a transition should represent an uninterruptable sequence.

(iii) Conditions are events which happen to the system to cause a transition from one state to another. YOURDON uses different words to help distinguish the external events of the system with the internal control signals passing around the state transition diagram.

(iv) Actions are the results of a condition, that is when the system is subject to a condition, the system will "do" those actions.

Conditions may come from terminators on the context diagram, in this case they may not be word perfect copies as they are often grouped together into events and used to form an event list. Other conditions are those which are generated internally in the system and come from control processes or data transformations. Likewise actions go to other parts of the system (control processes or data transformations), or will appear in an outgoing event flow column of the event list, that is, they may go to :

(a) Terminators.
(b) Data transformations.
(c) Other control processes.
(d) Timers.

A7: Process Specification.

The purpose of a process specification is to define what processing must be done by a transform on the input data flows to generate the output data flows. Several tools could be used to perform the task of describing a process. The main method used for process specification is structured English, but any method can be used so long as it satisfies two crucial requirements, namely:

(i) The process specification must be expressed in a form that can be verified by the user and the systems analyst.
(ii) The process specification must be expressed in a form that can be effectively communicated to the various audiences involved.

Pseudo-Code satisfies these two conditions and is frequently used to implement the process specification. Tables A.1 and A.2 show the models used in YOURDON's structured analysis and design methodology and the individual tools used to construct the YOURDON models.

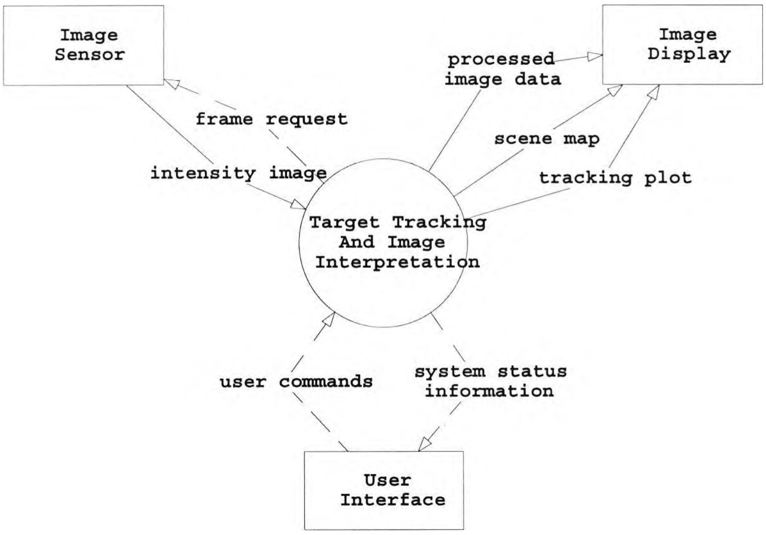
Stage	**Model**	**Sub-model**	**Comprising**
Analysis	Essential model	Environmental model	Data context diagram
			Control context diagram
			Requirements dictionary
			External timing specification
			Event list
		Behavioural model	Data flow diagrams
			Control flow diagrams
			Entity relationship diagrams
			Requirements dictionary
Design	Implementation model	Processor model	Context diagram
			Data flow diagrams
			Control flow diagrams
			Entity relationship diagrams
			Task resource specification
			Design dictionary
		Task model	Data flow diagrams
			Control flow diagrams
			Entity relationship diagrams
			Module resource specification
			Design dictionary
		Module model	Structure charts

Table A.1 The Models used in the YOURDON analysis and design methodology.

Document	**Description**
Dictionary	A textual description with a formal syntax, but using extensive comments.
Event list	A table relating events and their consequences.
Data flow diagram	A diagram showing the movement of information through a process. Probably the document most identified with YOURDON.
State transition diagram	A diagram showing, in an abstract way, how a process is controlled.
Entity relationship diagram	A diagram showing logical connections between data items.
Structure charts	A diagram showing how software units are composed.

Table A.2 Major tools used to build the YOURDON models.

Appendix B Environmental Model.



B1: Data Context Diagram.

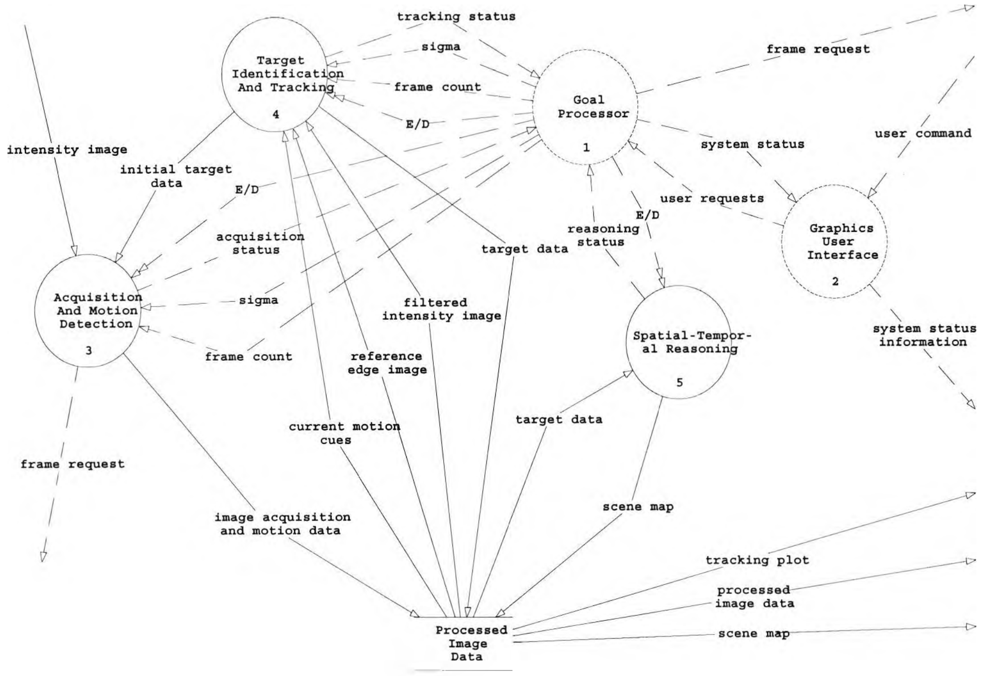
Event	Response	Classification
Quit	Terminates all processing and hand control back to the operating system	C
Step	Process a single frame of image data and then return control back to user	C
Run #1	Provide a user menu for inputting the number of image frames that are to be processed	D
Run #2	Process all frames of frames of image data and then hand control back to user	C
Display	Provide a user menu for inputting the processed image data for displaying, display image data and / or return control to the user.	C/D

B2: External Event List.

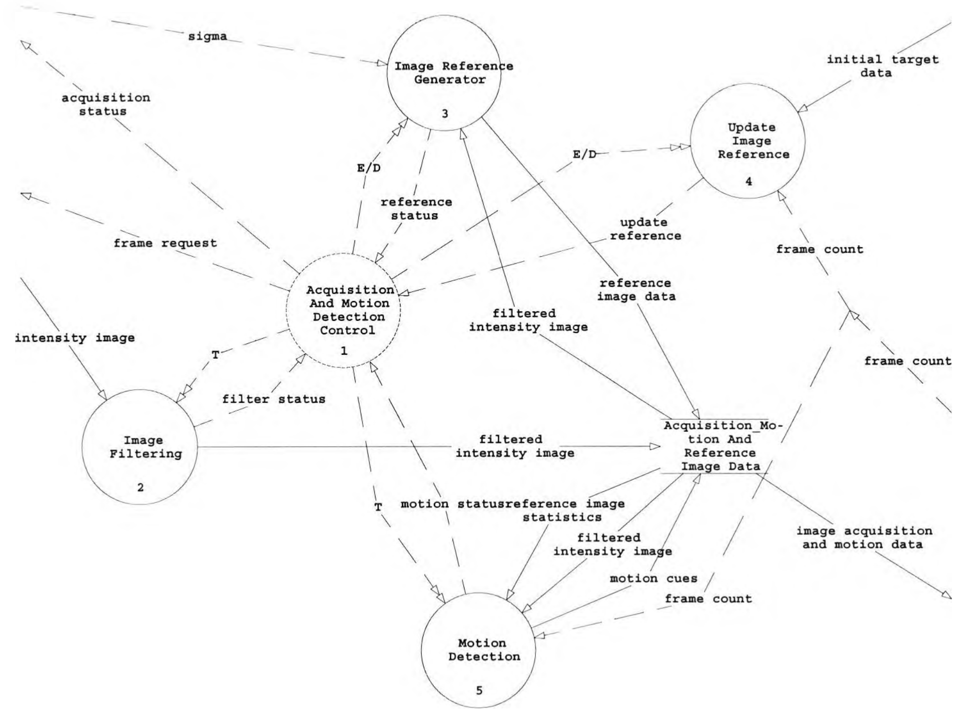
Name	frame request
Composition	frame request = {frame request} 100
Meaning	controls the rate at which frames of image data are input to the system
Name	intensity image
Composition	intensity image = {image pixels} 262144 {512 by 512}
Meaning	frame of image data made up of 512 rows by 512 columns of image pixels
Name	processed image data
Composition	processed image data = {processed image pixels} 262144 {512 by 512}+ {image tiles} 16384 {128 by 128)
Meaning	processed image data in pixel format OR processed image data in tile format
Name	scene map
Composition	scene map = {image tiles}16384 {128 by 128} + {scene pixels} 262144 {512 by 512}
Meaning	areas of the image that have been identified to exhibit motion or where objects may become occluded
Name	tracking plot
Composition	tracking = {window co-ordinates}256 max {range between 512 by 512} +
Meaning	provides information on the position in tile co-ordinates of objects that have become occluded
Name	processed image data
Composition	processed image data = edge image + filtered intensity image + motion cues + un-matched target tracks + map segments
Meaning	provides visual information produced by the image processing sub-functions.
Name	system status information
Composition	system status information = acquisition status + tracking status + reasoning status
Meaning	provides visual information on the system status.
Name	user commands
Composition	user commands = step + display + run + quit
Meaning	controls the target tracking and image interpretation that will be carried out by the system

B3: Requirements Dictionary.

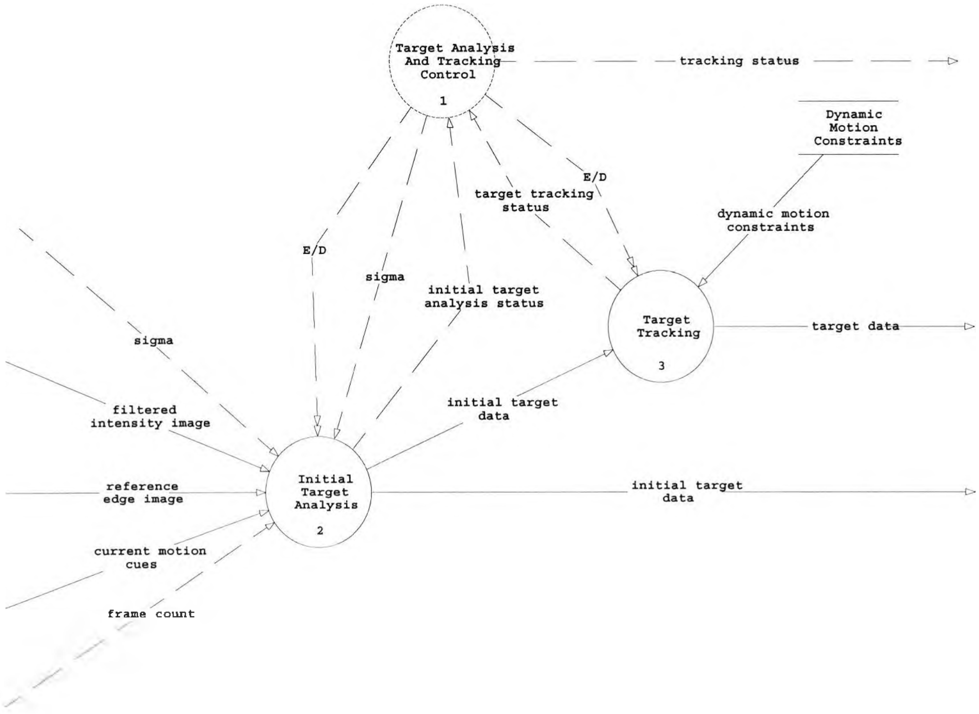
Appendix C Behavioural Model.



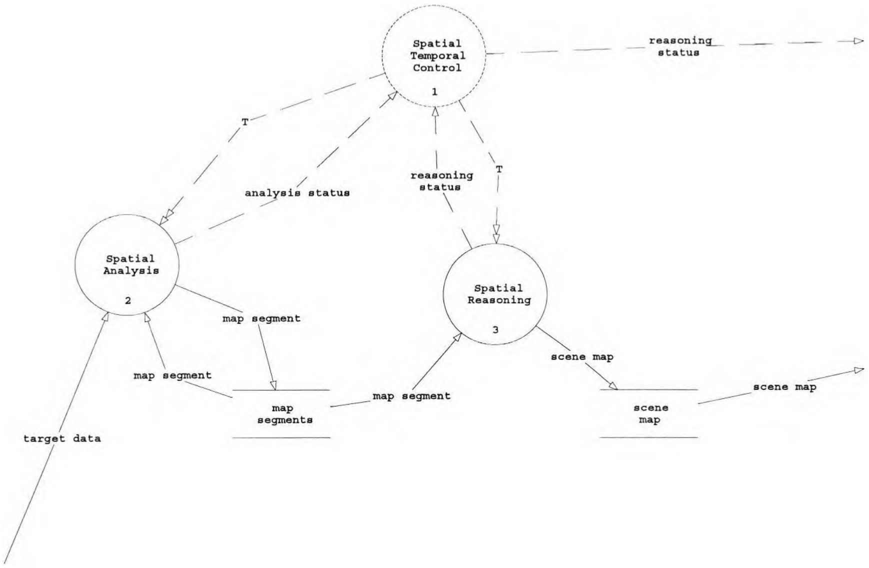
C1: Target tracking and image interpretation level 1 DFD.



C2: Acquisition and motion detection Level 2 DFD.



C3: Target identification and tracking level 2 DFD.

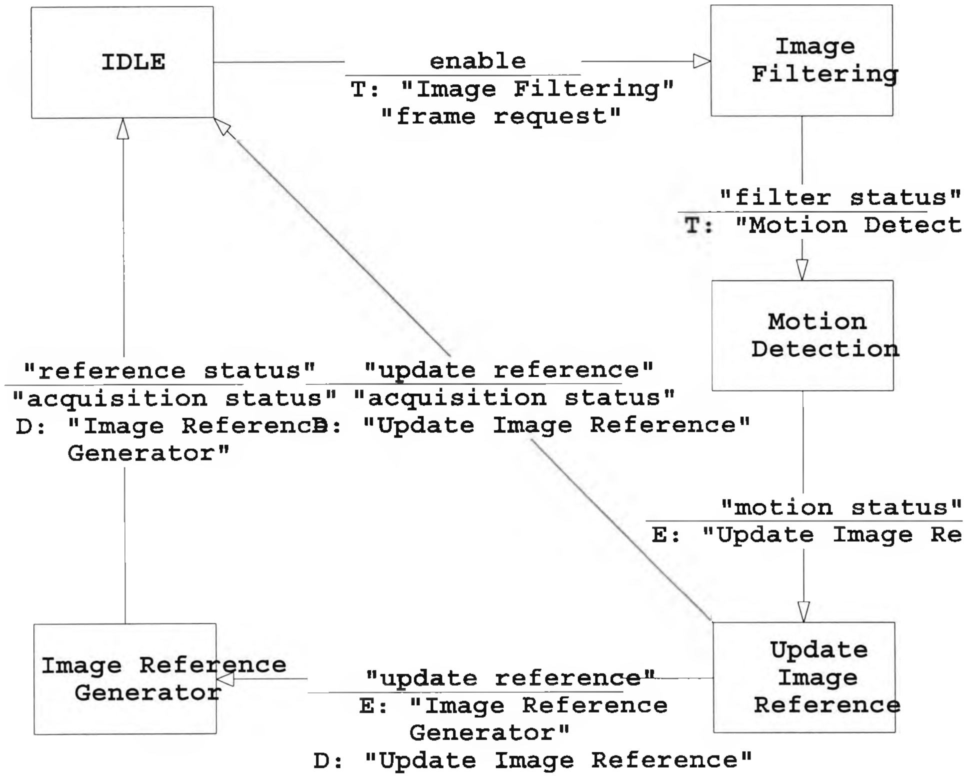


C4: Spatial-Temporal reasoning level 2 DFD.

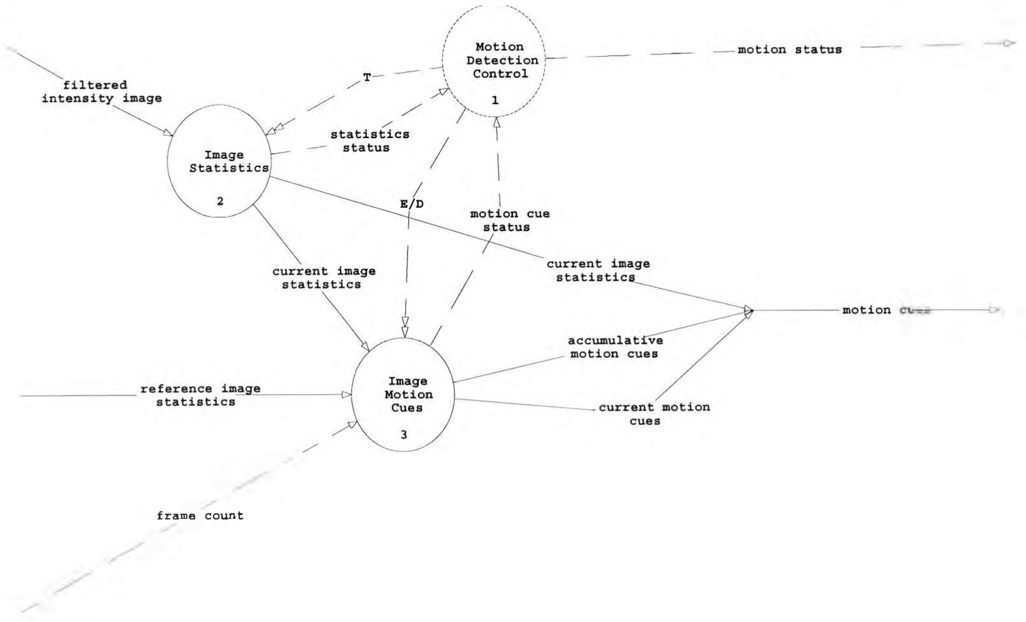
```
BEGIN
  Initialise System;
  WHILE user requests <> Quit DO
    BEGIN
    Evaluate User Request(user requests);
    WHILE frame to be processed  DO
      BEGIN
        Main Processing Task(frame count, sigma);
        System Assessment(system status,scene map,target data,
                                     frame count);
        Display(system assessment);
      END;
    ENDDO;
    END;
  ENDDO;
END.

Main Processing Task(frame count, sigma);
BEGIN
  frame request = TRUE;
  ENABLE Acquisition And Motion Detection(frame count, sigma);
  IF acquisition status = FALSE
    BEGIN
      system status = Acquisition Error;
      Display(system status);
    END;
  ELSE
    BEGIN
      Display(image acquisition and motion data);
      DISABLE Acquisition And Motion Detection(frame count,sigma);
      ENABLE Target Identification And Tracking(frame count,sigma);
      IF tracking status = FALSE
        BEGIN
          system status = Tracking Error;
          Display(system status);
        END;
      ELSE
        BEGIN
          Display(target data);
          DISABLE Target Identification And Tracking(frame count,
                                                     sigma);
          ENABLE Spatial-Temporal Reasoning();
          IF reasoning status = FALSE
            BEGIN
                system status = Reasoning Error;
                Display(system status);
            END;
          ELSE
            BEGIN
              Display(scene data);
              DISABLE Spatial-Temporal Reasoning();
            END;
          ENDIF;
        END;
      ENDIF;
    END;
  ENDIF;
frame request := FALSE;
END;

System Assessment(system status, scene map,  target data, frame count);
BEGIN
  IF system status = FALSE;
    BEGIN
      Display(user requests);
      Evaluate User Request(user requests);
    END;
  ELSE
```

```
      BEGIN
        IF scene map AND  new target data AND still frame count
          system assessment := still building map;
        ELSE IF scene map AND old target data AND still frame count
          system assessment := still building map;
        ELSE IF scene map AND old target data AND end frame count
          system assessment := map construction complete;
        ELSE IF scene map AND new target data AND end frame count
          system assessment := map construction incomplete;
        ELSE
          system assessment := still building map;
        ENDIF;
      END;
    ENDIF;
END;
```

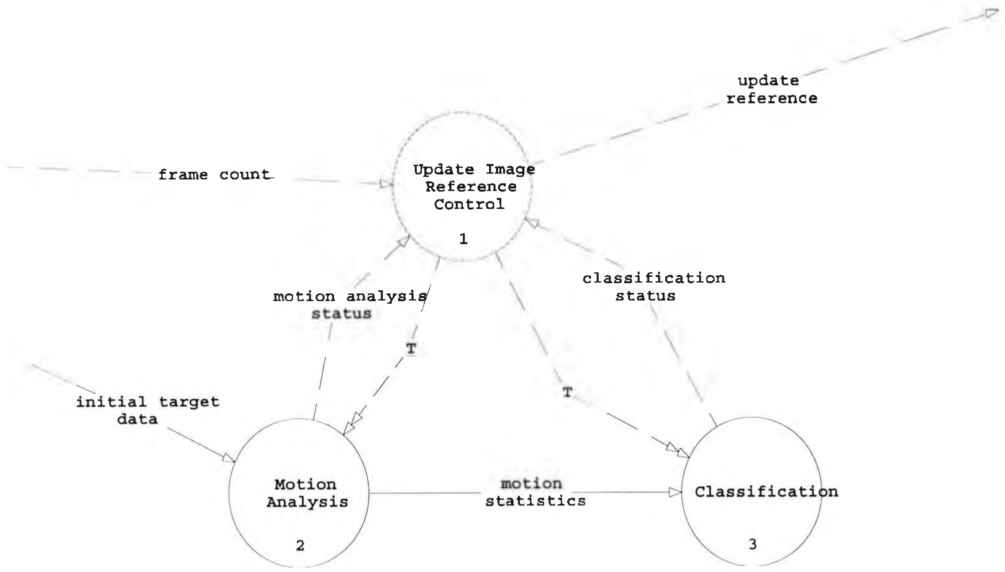
C5 : Goal achievement Control Specification.



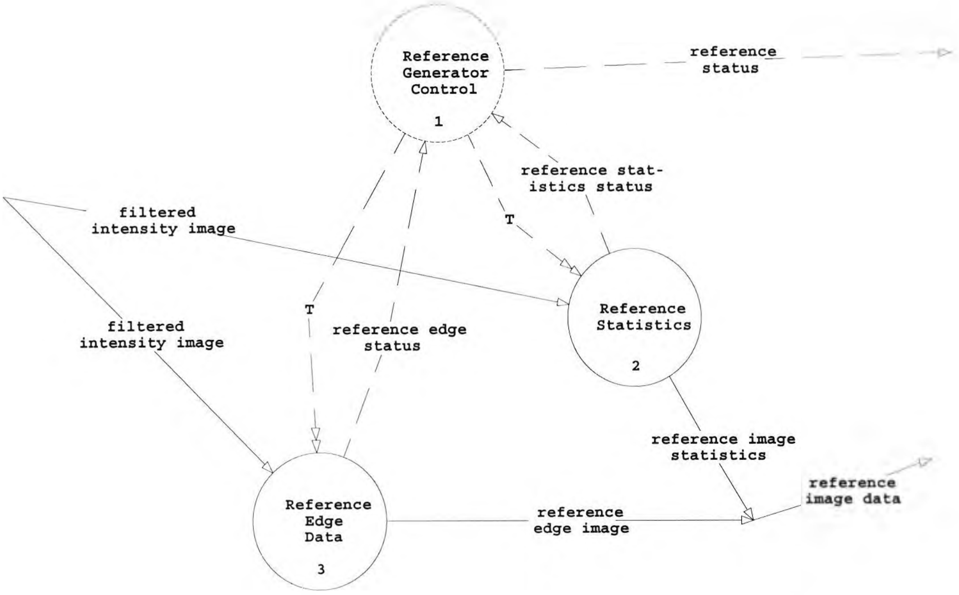
C6: Acquisition and motion detection STD.



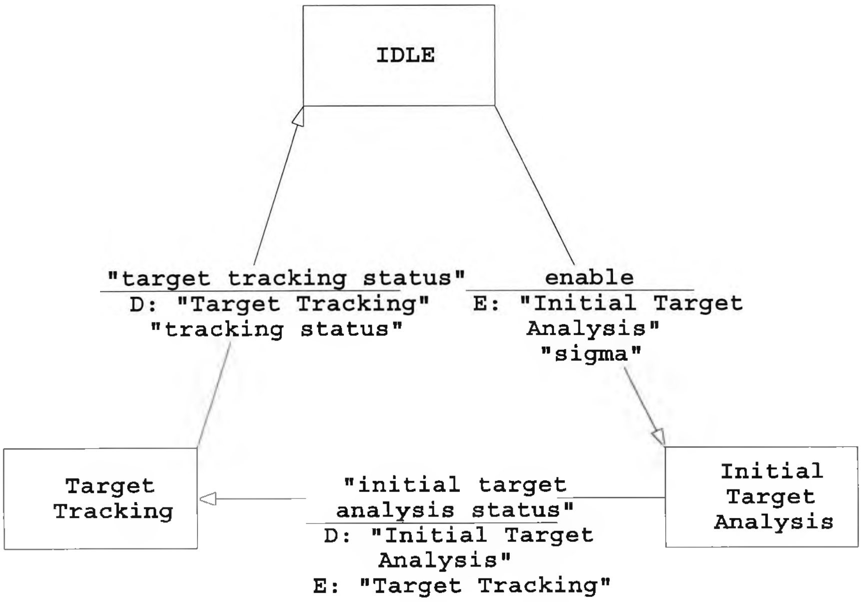
C7 : Motion detection level 3 DFD.



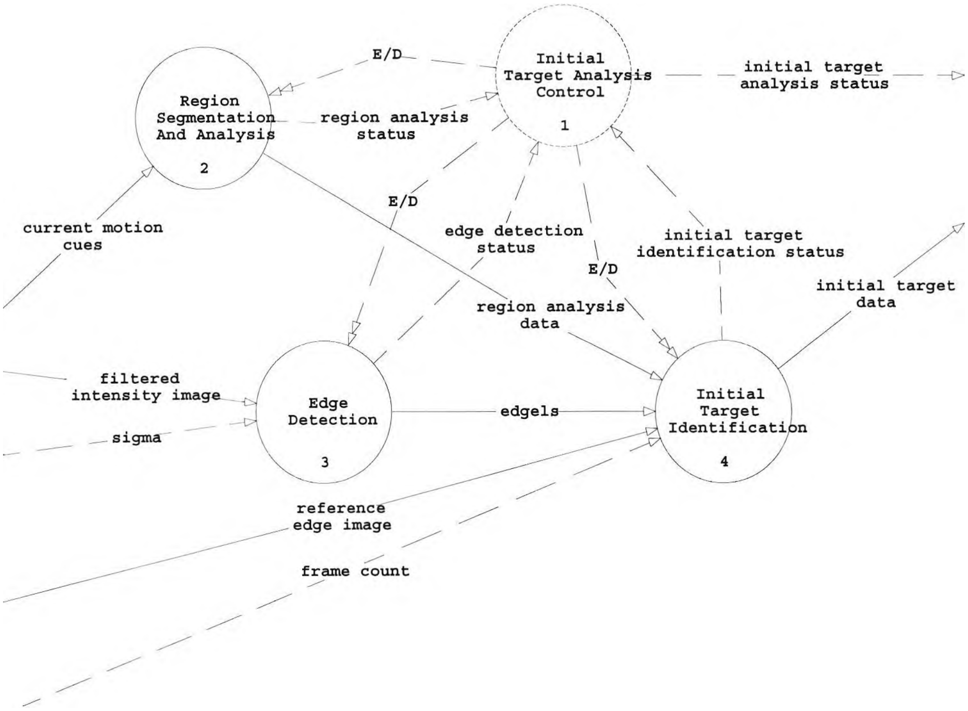
C8: Update reference level 3 DFD.



C9: Image reference generator level 3 DFD.



C10: Target Identification and Tracking STD.



C11: Initial target analysis level 3 DFD.

Appendix D CD-ROM.

D1: CD-ROM Directory and file structure.

Research\Movies.

RoadMovie1.fli
RoadMovie2.fli

Research\Waaplayer.

Aaplay.dll
Aaplay.h.
Aaplay.lib
Aapldev
Aavga.dll
Aawin.exe
Aawin.gid
Aawin.hlp
Aawin.ico
Martinsh.tif
Mciaap.drv
Mplayer.ini

Research\Thesis.

frontsht.doc
contents.doc
Chapter1.doc
Chapt2a.doc
Chapt2b.doc
Chapter3.doc
Chapter4.doc
Chapter5.doc
Chapter6.doc
Chapter7.doc

Chapter8.doc
refs.doc
biblog.doc
append_a.doc
append_b.doc
append_c.doc
append_d.doc

D2: Installation.

System Requirements.

PC based system with

Pentium processor (preferred).
16 Mbytes of RAM (minimum).
4 speed CD-ROM drive (minimum).
140 Mbytes free disk space (full installation).
Windows 95.
Word 6 or later (thesis only).

The software that accompanies this thesis has been run and tested using windows 95. To install the software simply copy all files in each directory on the CD-ROM to your hard disc (thesis is optional).

D3: Autodesk Animation Player for Windows Version 1.00.

This shareware animation player for windows is simple and easy to use. To run the player from windows 95 use the following procedure.

1: Choose the run option from windows 95 and select the Aawin.exe file.
2: Select **File** from the toolbar menu.
3: Select **Open Animation** from the file menu and load either roadmo~1.fli or roadmo~2.fli from your directory.
4: Select **Anim Settings** from the file menu.
   Set loops:Frames to 1.
   Set Pause at End to 20.
   Click OK.
5: Click >> to run.