

# Building a Model of a Road Junction Using Moving Vehicle Information

Xu Li-Qun  
David Young

School of Cognitive and Computing Sciences, University of Sussex  
Brighton BN1 9QH, UK

David Hogg

School of Computer Studies, University of Leeds  
Leeds LS2 9JT, UK

## Abstract

We describe a program to construct a model of a road junction using data from a single camera. The model specifies the ground plane orientation in camera coordinates and the positions of traffic lanes, and is obtained entirely from observations of vehicle movements, with no static image analysis. At present, the model is restricted to representing straight lane segments. We describe our methods for segmentation, object tracking, ground plane estimation and lane identification. Throughout, we emphasise techniques which are computationally cheap and can be used with fairly low resolution data and a low frame rate. We nevertheless obtain a reliable model by using the statistics of large numbers of vehicle movements.

## 1 Introduction

In the course of a project aimed at producing reliable information about vehicle movements through a road junction, using computer vision, we have tackled the problem of building a model of the junction. The model, which specifies the position of the ground plane in camera coordinates, and the positions of vehicle lanes in the ground plane, is produced using measurements of moving vehicles in a single camera's view. In this paper, we report our overall strategy and some of the technical problems and our choice of solutions to them, and we propose how the work could usefully be developed. There has been much other work on the analysis of road traffic scenes, e.g. [1-7]; our aim here is to demonstrate how some computationally straightforward techniques, applied to visual data with coarse spatial and temporal sampling, can give us a reliable site model on which to base more detailed analysis of individual vehicle movements.

Specifically, we examine: segmentation of moving vehicles from the static background; tracking of multiple vehicles from frame to frame; estimation of ground plane parameters; and estimation of the positions of lane boundaries in the ground plane.

The approach taken was motivated by two complementary factors. First, automatic model generation means that the need for a site survey, or any manual intervention, is avoided, allowing a complete system to be set up quickly, and recalibrated quickly following any movement of the cameras. Second, producing a good model does not necessarily require tracking every vehicle accurately; the model can be based on the statistics of many vehicle movements. Once a model has been generated, it will clearly assist in making more reliable measurements of individual vehicles' movements. For present pur-

poses, then, we require reasonably cheap computational methods, but we can afford to use methods that make occasional mistakes.

The traffic scene domain, and the practicalities of data collection, introduced some specific challenges. These included: a large inter-frame interval (determined by the low digitisation rate of our equipment, but not unrealistic for a practical application); variations in illumination (real, and also due to camera exposure compensation); static or slowly moving vehicles, producing difficulties with segmentation based on dynamic information; overlapping and very close images of vehicles, in queues and when travelling close together in parallel lanes; unconventional and unexpected behaviour by drivers; occlusion of vehicles by street furniture. Despite these problems, we chose to base our approach on dynamic information, since the difficulties and ambiguities of static scene analysis are likely to be harder to overcome. Since the purpose of the model is to provide a predictive framework within which individual vehicles can be tracked accurately, it is sensible to base the model on exactly this kind of information.

To test our ideas, we collected data from an intersection on a four-lane road, with two lanes in each direction, the two directions partially separated by islands. This was crossed by an ordinary two-lane road; the junction was controlled by traffic lights. We made a video recording using a conventional camcorder with automatic electronic exposure control from a foot bridge about 25 m from the junction. Traffic was moderately heavy, and contained a wide spread of vehicle sizes from motorcycles to large articulated trucks, travelling at speeds up to about 80 km/h. When the lights were against the traffic on the main road, queues formed, and there were frequently vehicles waiting to make right turns. Fig. 1 shows the view obtained by the camera.

We digitised about 30 minutes of the recording at a rate of about 3.3 frames/s, at a spatial resolution of  $180 \times 143$  pixels, so the data rate was only about 85 kilobytes/s. Efficient algorithms which can produce useful results from these data therefore stand a good chance of being applicable in practice. The algorithms described below were implemented in POP-11 and C using the POPLOG system.

## 2 Segmentation

The segmentation scheme described here is based on the combination of motion and contrast cues. First, difference information from successive frames of an image sequence is the key to detecting regions in which motion is taking place [9, 10]. Second, edges derived from individual frames indicate the surface boundaries of possible objects. Combining the filtering effect of the former and the localisation of the latter, the individual moving objects (vehicles or pedestrians), irrespective of their sizes, are isolated. The inspiration for this scheme can be traced to [11], [12] and [13], though the exact operation adopted differs from all these.

Let  $f_c$  and  $f_p$  denote the current and previous images respectively. The operation  $s$



Figure 1. Typical views of the road junction.

corresponds to taking the sum of the absolute values of the convolutions with the two  $3 \times 3$  Sobel edge templates;  $b_T$  is a threshold operation with threshold  $T$ ; and  $\wedge$  means the operation of coincidence or logical AND. The operation we used is then described as

$$B = b_{T1}(s(f_c)) \wedge b_{T2}(s(|f_p - f_c|)) \quad (1)$$

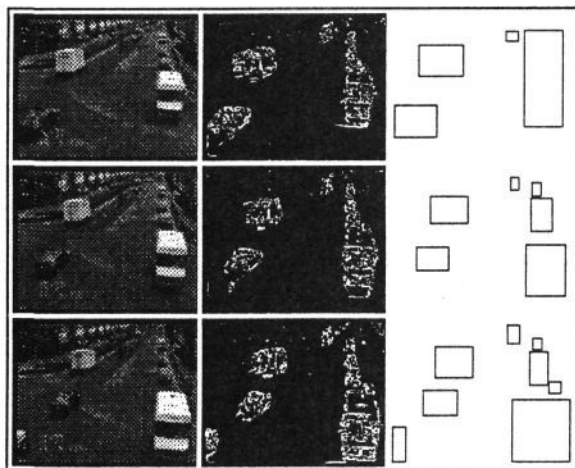
The thresholds  $T1$  and  $T2$  are in fact identical, and are chosen to be the mean of the local minima in the smoothed histogram of the values in  $s(|f_p - f_c|)$ .

Fig. 2b shows a typical binary image of the traffic scene after this processing. Moving objects are clearly picked out, although some failures still occur. In general, the most significant problems are the fragmentation of large slow-moving vehicles and the grouping together of close vehicles.

As a kind of domain-independent low-level processing, this approach can consistently remove all the static parts of the background scene, including of course temporarily stopped vehicles. This makes the method inadequate for atomic event recognition, but entirely suitable for the present purpose of collecting sufficient statistical information about vehicle movements to generate a site model, without the cost of maintaining a reference image.

We adopt a simple approach to segmenting the moving region into individual vehicles. Each vehicle's image is represented as a rectangle with its sides parallel to the image axes. These are extracted by recursively grouping set pixels in the binary image, using 8-connectivity, and recording the bounding rectangle for each connected blob. Rectangles which are close together relative to their sizes are merged. More sophisticated techniques such as chain codes [14] or medial axis transforms [15] would allow more exact representation of blob shape, at greater computational expense.

Fig. 2c shows the results of this processing. Generally speaking, the fit between the vehicle images and the rectangles is adequate, despite the coarse representation.



**Figure 2.** (a) Three images from the traffic sequence. (b) Results of applying the combined motion and texture detection operator. (c) Segmentation into rectangles.

### 3 Object tracking

The establishment of feature correspondence between two related images has been a challenging area in computer vision research. Examples can be found in stereopsis algorithms [16, 17] and in motion understanding systems [7, 18-22]. Furthermore, Jenkin & Tsotos [23], performing stereo matching simultaneously with temporal feature tracking, presented a successful method of attacking the dynamic stereo problem. Important theoretical and practical developments included Ullman's minimal mapping approach [24], based on the 2-D distance between tokens, relaxation-based algorithms [25, 26], and Scott and Longuet-Higgins' eigendecomposition method [27-29]. An important constraint in tracking moving objects, when more than two frames are available, is that of motion smoothness, and this was notably exploited in [23] and [30]. The relationship to human visual perception has been elucidated by many experiments [31-36].

Early work in the road traffic domain was well summarised by Dreschler & Nagel [37]. These authors tried firstly to detect and track the *prominent points* of a single moving car, then to recover 3-D positions of these points in object coordinates, and finally to approximate a convex hull description of the car by using a minimisation approach. The assumptions introduced include a known camera tilt angle and planar road surface. Recent work can be found in [38] and [39], where a model-based paradigm is adopted. Here, tracking is carried out in 3-D world coordinates rather than 2-D image coordinates; this has clear advantages for any predictive scheme, but requires prior knowledge of object and world models.

Since our aim is to construct a site description, we carry out tracking in 2-D, and retaining our criteria of simplicity and efficiency we adopt a method in which path coherence and motion smoothness are embedded in the formulation of a similarity matrix for inter-frame matches. The most important constraint on our choice is our large inter-frame interval of about 0.3 s, which results in large image motions between frames. In addition, many vehicles (up to about 15) are often visible and moving in assorted directions in the same frame, and frequently occlude or pass close to one another. These factors mean that strong assumptions, and probably a model-based approach, are necessary to get highly reliable results.

For each image frame, the processing in section 2 results in a list of *rectangles*. Each of these is supposed to represent a different vehicle or pedestrian, and we are required to track these to produce a list of *trajectories*, each consisting of a list of linked rectangles extending over a number of frames. Each trajectory is taken to correspond to one *object*. We make the assumption that it is adequate to take each frame sequentially and to match the rectangles in it to the current set of active trajectories; this avoids the need for any backtracking through previous frames. The first requirement is then a *similarity measure*: a function from trajectory-rectangle pairs to positive real numbers; given this, we also need an algorithm for deciding unique matches given a matrix of the similarity measures for the rectangles in the current frame and the currently active trajectories.

Our similarity measure involves the sizes, aspect ratios and image velocities of the rectangles. For each active trajectory, a predicted rectangle, or *template*, is created, whose height, width and position are calculated using polynomial extrapolation of the same parameters of the rectangles forming the trajectory [40]. (The "position" of a rectangle here is that of the centre of the base of the rectangle.) Taking a rectangle in the current frame, we then calculate the relative difference in width, height and position between this rectangle and the template (e.g. in the case of width we find  $(2|w_t - w_r|) / (w_t + w_r)$ , where  $w_t$  is the width of the template and  $w_r$  the width of the

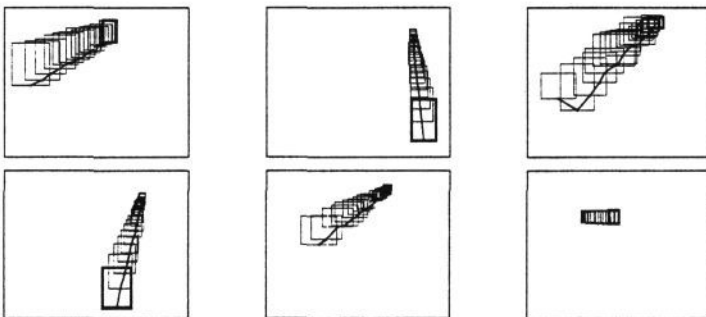
current rectangle). If all three of these measures are less than some thresholds, a potential match is recorded in the similarity matrix, with the distance between the positions of the template and rectangle stored as a measure of dissimilarity. Otherwise, we rule out a match between this template and this rectangle. We thus end up with a sparse {no of active trajectories}  $\times$  {no of rectangles} matrix for the current frame.

The important problem of discontinuous trajectories is tackled in building the matrix. Our low-level processing often causes objects to fail to be represented in a given frame; for example, two close vehicles may be merged into a single object, or one vehicle crossing the path of another may occlude it. In principle, an arbitrarily long gap can be tackled by keeping a trajectory active even when no rectangle has been appended to it in the current frame, and extrapolating over more than one frame interval in generating subsequent templates. In practice, we have found it sufficient to deal with gaps of one frame and no more. Thus a trajectory becomes inactive (i.e. is simply stored) when no rectangle has been appended to it for two frames. In matching a rectangle to a trajectory which has been extrapolated for two frame intervals, the thresholds mentioned in the last paragraph are increased.

Ideally, the sparse (dis)similarity matrix would have no more than one entry in each row or column, indicating unique matches. In practice, this will not be the case. Further processing is needed to establish the final matches. Our algorithm is straightforward: for each row of the matrix (i.e. for each trajectory) we find the minimum value, which corresponds to the closest rectangle, and remove all the other entries. Then for each column (i.e. for each rectangle), we find the minimum value amongst the remaining entries, and we note the corresponding match. This is a conservative approach, in the sense that matches made are likely to be good ones, but some matches can be missed.

Finally, if a frame cannot be matched to a trajectory, it is regarded as the start of a new trajectory.

This approach has proved quite robust to noise in our data, producing trajectories reliable enough for subsequent work. Fig. 3 shows some of the trajectories obtained. Errors arise mainly from occlusion between vehicles, resulting in merging or splitting mistakes in the segmentation. We have applied the method of Scott & Longuet-Higgins [28] to our similarity matrix, but the results, assessed in terms of mean trajectory smoothness, were not significantly better. The high noise levels and large inter-frame interval for our rectangle data mean that improved computational methods have strongly diminishing returns. Our simple method captures the essentials for adequate, if not optimal, tracking: trajectory prediction and a robust similarity measure.



**Figure 3.** Some trajectories found by the method described. The top right shows a trajectory affected by the proximity of other vehicles. The bottom right shows a pedestrian crossing the road.

## 4 Ground plane estimation

For simplicity, we treat the ground surface as a plane. This is reasonable in many urban environments, but we expect to need to use a more accurate model in the future. The problem is to find the orientation of this plane in camera coordinates. Although several methods are possible, for example using the image size of vehicles, one which we have found successful and which we describe here is based on finding vanishing points for parallel vehicle trajectories. Although this requires the assumption that some parts of the vehicle trajectories lie in families of parallel straight lines, this will be true for many sites. The technique turns out to be robust against noise, largely because it is based on non-local measurements: the parameters of each trajectory are derived from a significant number of images and from a large region of the image. Furthermore, a large number of trajectories can be analysed in order to derive the ground plane parameters, using a form of the Hough Transform. Other advantages are the possibility of ready integration with information from the analysis of static features such as the road boundaries and lane markings, and the fact that the segmentation into groups of parallel trajectories can be used immediately in finding traffic lane positions.

The basis of the method is the fact that under perspective projection, parallel lines in a 3-D scene form a fan of lines in a 2-D image, all intersecting at a common vanishing point. If the 3-D lines lie in a plane, two vanishing points are sufficient to determine the rotational part of the transformation from the camera coordinate system to a coordinate system with two axes lying in this plane. In our case, given straight segments extracted from traffic trajectories, the problem of finding two vanishing points, and hence the road surface parameters, reduces to a problem already well studied in, e.g., [41-44]. We extend this work by applying it to trajectories rather than to static features, although we note also the effective application of vanishing points in autonomous vehicle road following by Liou & Jain [45].

The first stage of the process is to segment the trajectories into straight line segments, using recursive subdivision with a test for goodness of fit based on maximal normal distance of the trajectory segment from the straight line joining its ends [46]. Segments shorter than a threshold are discarded.

Each straight segment is then represented by the unit normal  $\mathbf{n}$  of an *interpretation plane* which contains the segment and the focal point of the camera,  $O$ . If this normal is mapped onto the Gaussian sphere centred on  $O$ , the great circle normal to  $\mathbf{n}$  contains all the directions which could be parallel to 3-D trajectory segment, given the evidence of the line segment in the 2-D image. The intersection of more than one such great circle gives the vanishing point for a group of parallel segments, and their common direction in 3-space. For further details, see the references cited above, [47] and [48].

In our implementation, we parameterise the forward Gaussian hemisphere using azimuth  $\alpha$  and elevation  $\beta$ . These are related to image coordinates  $(x,y)$  and camera focal length  $f$  by

$$x = f \tan \alpha \quad y = f (\tan \beta / \cos \alpha) \quad (2)$$

A hierarchical Hough transform [43], which utilises an irregular division of the Gaussian sphere, was used to find the vanishing points. In this algorithm, the sphere is divided into four patches in terms of  $\alpha$  and  $\beta$ , and the line segments vote for the patch through which their great circles pass. Those patches with votes exceeding a user-defined threshold are further subdivided and the process proceeds recursively, using a breadth-first strategy on the quadtree, until the patch size is as small as the expected scatter in the

data. (We used a  $1.5$  degree  $\times$   $1.5$  degree limiting region.) In order to reduce the effects of noisy short segments, we weighted each line segment by a linear combination of its duration in time and its length in the image.

The procedure outputs a small patch of the Gaussian sphere and a list of the segments voting for it. After the dominant vanishing point has been found, the contributing line segments are removed from the whole data set, and the process repeated to find subsequent vanishing points.

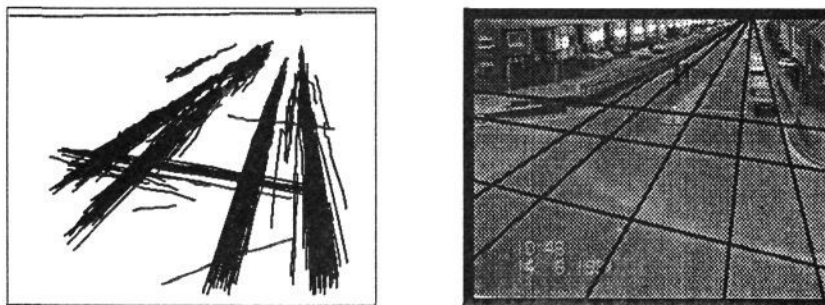
Results obtained from the traffic sequence are shown in Fig. 4. The first vanishing point along the main road is shown at the top right of Fig. 4a by a small blob, whilst the second vanishing point along the crossing road is outside the frame to the left, at the intersection of the roughly horizontal lines in Fig. 4b. The line joining the two vanishing points (the estimated horizon) is shown along the top of Fig. 4a. Although our data are dominated by vehicles moving along the dual carriageway, and trajectories in the direction across the line of sight are under-represented, the method nonetheless finds two groups of parallel trajectories. It can be seen that both the vanishing points found lie close to the apparent position of the horizon (which is not actually visible in the image), indicating qualitative success for the method.

Given two vanishing points in the image, it is straightforward to obtain the ground plane parameters, up to a scale factor. The two lines in 3-space through  $O$  and each of the vanishing points are both parallel to the ground plane. The normal to the ground plane in camera coordinates is therefore given by the cross product of these two vectors, and the rotation matrix follows easily.

## 5 Identification of traffic lanes

Since our goal is to provide a basis for event recognition in traffic scenes, it is valuable to have a description in terms of traffic lanes, which provide primary expectations for traffic behaviour, and summarise large numbers of vehicle movements. We begin with the assumption that the lanes are straight, which gives strong computational advantages, though clearly will not be adequate for all sites. We choose to carry out the lane analysis in image coordinates, since the vanishing point method for the ground plane leads naturally to an approach to lane segmentation, the direction histogram method.

Given a vanishing point, the line of each associated trajectory segment can be described in terms of a single parameter - its orientation in the image - since it must



**Figure 4.** (a) Trajectory line segments and estimated horizon position.. (b) The lane structure superimposed on an image from the original sequence.

(approximately) pass through the vanishing point. We measure this orientation as the angle between the horizontal and the mid-point of the trajectory segment. The set of trajectories can then be described by the histogram of their orientations. In practice, we weight each contribution to the histogram by the duration in time of the trajectory segment.

The histogram clearly contains information about the lane structure for the vehicles whose paths contributed to the particular vanishing point, for instance those moving along the dual carriageway road. In order to find the lane boundaries, we smoothed the histogram with a 3-point average, and found local maxima. These were taken to correspond to lane centres; we simply took bisectors of these directions as the lane boundaries. This is repeated for each vanishing point. Once found, the lane boundaries can be projected onto the image plane, or onto the ground plane.

Results are shown in Fig. 4b, by projection of the lane boundaries onto an image. It can be seen that the actual lanes are fitted quite well in this case, despite the fact that the main road description uses over 400 segments and the cross road only about 30.

## 6 Discussion

We have demonstrated that by making suitable assumptions and adopting appropriate algorithms, a site model can be generated from coarsely sampled image data, without great computational cost. For a computer vision system designed to interpret a traffic scene, this is a valuable ability. Our method relies on utilising the dynamic information in the scene, and on measuring reasonably large numbers of vehicle trajectories so that statistical techniques can overcome high noise levels. We have obtained promising results from challenging data.

Several improvements are possible to this initial system. The representation of vehicle images using an “upright” rectangle is probably over-simple, and some more degrees of freedom would be useful. Certainly one extra parameter specifying the orientation of the major axis of the blob would be worthwhile. The tracking algorithm could possibly benefit from more sophisticated prediction: Kalman filter techniques rather than the polynomial extrapolation would undoubtedly give improvements at some computational cost. The vanishing point method is sensitive to the thresholds used, and needs to be set automatically. Finally, we need to be able to deal properly with vehicles turning at the junction; one way to do this is to regard their movements as transitions from one lane to another; alternatively we could introduce curved lane segments into our model, extending the Hough Transform method to deal with circular trajectories.

The main aim, however, is to provide a relatively simple model to bootstrap more sophisticated processing. Once an approximate model has been obtained, image positions can be projected into 3-D, and various possibilities immediately arise, for example Kalman filtering on 3-D trajectories, which will allow much better tracking. The model can then be refined in parallel with the acquisition of information about individual traffic events.

## Acknowledgments

We are grateful to NTT-Data of Japan for supporting this work.

## References

- [1] D.C. Hogg, G.D. Sullivan, K.D. Baker & D.H. Mott, Recognition of vehicles in traffic scenes using geometric models. In: IEE First Conf. Road Traffic Data Collection. London, 1984, pp 115-119.
- [2] A. Houghton, G.S. Hobson, L. Seed & R.C. Tozer, Automatic monitoring of vehicles at road junctions. *Traffic Engineering and Control* 1987; 28: 541-543.
- [3] K.W. Dickinson, Traffic data capture and analysis using video image processing. PhD Thesis, Sheffield University, 1986.
- [4] K.W. Dickinson & C.L. Wan, Road traffic monitoring using the TRIP II system. In: IEE Second Conf. Road Traffic Monitoring. London, 1989.
- [5] A.T. Ali & E.L. Dagless, Computer vision for automatic road traffic analysis. In: Proc. Int. Conf. 'Automation Robotics and Computer Vision'. Singapore, 1990.
- [6] L. Du, G.D. Sullivan & K.D. Baker, 3-D grouping by viewpoint consistency ascent. In: Proc. British Machine Vision Conf. Glasgow, 1991, pp 45-53.
- [7] T.N. Tan, G.D. Sullivan & K.D. Baker, Structure from constrained motion using point correspondences. In: Proc. British Machine Vision Conf. Glasgow, 1991, pp 301-309.
- [8] R. Marslin, G.D. Sullivan & K.D. Baker, Kalman filters in constrained model based tracking. In: Proc. British Machine Vision Conf. Glasgow, 1991, pp 371-374.
- [9] D.C. Hogg, Model-based vision: a program to see a walking person. *Image and Vision Computing* 1983; 1: 5-20.
- [10] R.C. Gonzalez & P.A. Wintz, *Digital Image Processing*. Addison-Wesley, Reading Mass., 1987.
- [11] R. Jain, W.N. Martin & J.K. Aggarwal, Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing* 1979; 11: 13-34.
- [12] W.B. Thompson, Combining motion and contrast for segmentation. *IEEE Trans. PAMI* 1980; 2: 543-549.
- [13] S.M. Haynes & R. Jain, Detection of moving edges. *Computer Vision, Graphics and Image Processing* 1983; 21: 345-367.
- [14] H. Freeman, Computer processing of line-drawing images. *Computing Surveys* 1974; 6: 57-97.
- [15] D.H. Ballard & C.M. Brown, *Computer Vision*. Prentice Hall, Englewood Cliffs, 1982.
- [16] W.E.L. Grimson, Computational experiments with a feature based stereo algorithm. *IEEE Trans. PAMI* 1985; 7: 17-34.
- [17] N. Ayache & F. Lustman, Fast and reliable trinocular stereo vision. In: Proc. First Int. Conf. Computer Vision. London, 1987, pp 422-427.
- [18] J.K. Aggarwal, L.S. Davis & W.N. Martin, Correspondence processes in dynamic scene analysis. *Proc. IEEE* 1981; 69: 562-572.
- [19] O.D. Faugeras, F. Lustman & G. Toscani, Motion and structure from motion from point and line matches. In: Proc. First Int. Conf. Computer Vision. London, 1987, pp 25-34.
- [20] J.K. Aggarwal & N. Nandhakumar, On the computation of motion from sequences of images - a review. *Proc. IEEE* 1988; 76: 917-935.
- [21] H. Shariat & K.E. Price, Motion estimation with more than two frames. *IEEE Trans. PAMI* 1990; 12: 417-434.
- [22] M.E. Spetsakis & J. Aloimonos, A multi-frame approach to visual motion perception. *Int. J. Comp. Vision* 1991; 6: 245-255.
- [23] M. Jenkin & J.K. Tsotsos, Applying temporal constraints to the dynamic stereo problem. *Computer Vision, Graphics and Image Processing* 1986; 33: 16-32.
- [24] S. Ullman, *The Interpretation of Visual Motion*. MIT Press, Cambridge MA, 1979.
- [25] S.T. Barnard & W.B. Thompson, Disparity analysis of images. *IEEE Trans. PAMI* 1980; 2:

333-340.

- [26] K.E. Price, Relaxation matching techniques - a comparison. *IEEE Trans. PAMI* 1985; 7: 617-623.
- [27] G.L. Scott & H.C. Longuet-Higgins, Feature grouping by "relocalisation" of eigenvectors of the proximity matrix. In: *Proc. British Machine Vision Conf. Oxford, 1990*, pp 103-108.
- [28] G.L. Scott & H.C. Longuet-Higgins, An algorithm for associating the features of two images. *Proc. Royal Soc. London B* 1991; 244: 21-26.
- [29] L.S. Shapiro & J.M. Brady, A modal approach to feature-based correspondence. *Proc. British Machine Vision Conf. Glasgow, 1991*, pp 78-85.
- [30] I.K. Sethi & R. Jain, Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. PAMI* 1987; 9: 56-73.
- [31] J.T. Todd, Visual information about rigid and non-rigid motion: a geometric analysis. *J. Experimental Psychology: Human Perception & Performance* 1982; 8: 238-252.
- [32] V.S. Ramachandran & S.M. Anstis, Extrapolation of motion path in human visual perception. *Vision Research* 1984; 23: 83-85.
- [33] M.R.W. Dawson, The cooperative application of multiple natural constraints to the motion correspondence problem. In: *Proc. Seventh Biennial Conf. Canadian Soc. for Comp. Studies of Intelligence. Edmonton, 1988*, pp 140-147.
- [34] M.R.W. Dawson, Apparent motion and element connectedness. *Spatial Vision* 1989; 4: 241-251.
- [35] M.R.W. Dawson, The how and why of what went where in apparent motion: modeling solution to motion correspondence problem. *Psychological Review* 1991; 98: 569-603.
- [36] M.R.W. Dawson & R.D. Wright, The consistency of element transformations affects the visibility but not the direction of illusory motion. *Spatial Vision* 1989; 4: 17-29.
- [37] L. Dreschler & H.-H. Nagel, Volumetric model and 3D-trajectory of a moving car derived from monocular TV-frame sequences of a street scene. *Computer Graphics and Image Processing* 1982; 20: 199-228.
- [38] A.D. Worrall, R.F. Marslin, G.D. Sullivan & K.D. Baker, Model-based tracking. In: *Proc. British Machine Vision Conf. Glasgow, 1991*, pp 310-318.
- [39] D. Koller, K. Daniilidis, T. Thórhallsson & H.-H. Nagel, Model-based object tracking in traffic scenes. In: *Proc. Second European Conf. on Computer Vision. Santa Margherita Ligure, 1992*, pp 437-452.
- [40] J. Stoer & R. Bulirsch, *Introduction to Numerical Analysis*. Springer-Verlag, New York, 1980.
- [41] M.J. Magee & J.K. Aggarwal, Determining vanishing points from perspective images. *Computer Vision, Graphics and Image Processing* 1984; 26: 256-267.
- [42] H. Li M.A. Lavin & R.J.L. Master, Fast Hough transform: a hierarchical approach. *Computer Vision, Graphics and Image Processing* 1986; 36: 139-161.
- [43] L. Quan & R. Mohr, Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letters* 1989; 9: 279-286.
- [44] B. Brillault, Parallel and perpendicular line grouping in a 3-D scene from a single view. In: *Proc. British Machine Vision Conf. Oxford, 1990*, pp 257-292.
- [45] S.P. Liou & R.C. Jain, Road following using vanishing points. *Computer Vision, Graphics and Image Processing* 1987; 39: 116-130.
- [46] R.O. Duda & P.E. Hart, *Pattern Recognition and Scene Analysis*. Wiley, New York, 1973.
- [47] S.T. Barnard, Interpreting perspective images. *Artificial Intelligence* 1983; 21: 435-462.
- [48] T. Kanade, Geometrical aspects of interpreting images as a three-dimensional scene. *Proc. IEEE* 1983; 71: 409-460.