Dynamic Image Sequence Analysis Using Fuzzy Measures

Zhi-Qiang Liu, *Senior Member, IEEE*, Leonard T. Bruton, *Fellow, IEEE*, James C. Bezdek, *Fellow, IEEE*, James M. Keller, *Fellow, IEEE*, Sandy Dance, *Member, IEEE*, Norman R. Bartley, *Member, IEEE*, and Cishen Zhang, *Member, IEEE*

***Abstract*—In this paper, we present an image understanding system using fuzzy sets and fuzzy measures. This system is based on a symbolic object-oriented image interpretation system. We apply a simple, powerful three-dimensional (3-D) recursive filter to tracking moving objects in a dynamic image sequence. This filter has a time-varying 3-D frequency-planar passband that is adapted in a feedback system to automatically track moving objects. However, as objects in the image sequence are not well-defined and are engaged in dynamic activities, their shapes and trajectories in most cases can be described only vaguely. In order to handle these uncertainties, we use fuzzy measures to capture subtle variations and manage the uncertainties involved. This enables us to develop an image understanding system that produces a very natural output. We demonstrate the effectiveness of our system with complex real traffic scenes.**

I. Introduction

In scene understanding and image analysis, we often face many uncertain factors. Traditionally uncertainty is handled by probabilistic methods such as Bayesian networks [28] and the Dempster–Shafer (D–S) paradigm [17], [30]. However, these methods are based on the occurrence of crisp events, and in many cases are unable to cope with vagueness and subtle changes. Furthermore, such methods are not suitable for natural linguistic descriptions.

In image analysis tasks, we are dealing with two-dimensional (2-D) data that has a complex connection between the pixel-level description and the object-level (and causal level) description, especially when we are concerned with temporal image sequences. Such descriptive complexities lead to uncertainty that should be handled with uncertainty measures by which alternative hypotheses can be judged.

Manuscript received June 1, 1999; revised March 27, 2001. This work was supported in part by University of Melbourne International Collaboration Grants, ONR Grants N00014-96-1-0439 and N00014-96-1-0642, NSERC Canada, and the Australian Research Council. This paper was recommended by Associate Editor W. Pedrycz.

Z.-Q. Liu is with the Department of Computer Science and Software Engineering, University of Melbourne, Victoria, Australia. He is also with the School of Creative Media, City University of Hong Kong, Hong Kong, China.

L. T. Bruton and N. R. Bartley are with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB, T2N 1N4, Canada.

J. C. Bezdek is with the Department of Computer Science, University of West Florida, Pensacola, FL 32514 USA.

J. M. Keller is with the Department of Computer Engineering and Computer Science, Columbia, MO 65211-2060 USA.

S. Dance is with the Australian Bureau of Meteorology Research Centre, Melbourne, Australia.

C. Zhang is with the Department of Electrical and Electronic Engineering, University of Melbourne, Victoria, Australia.

Publisher Item Identifier S 1083-4419(01)05981-7.

For instance, given an isolated low-level region (blob), it is desirable from the high-level perspective to hypothesize what object it could be. Each hypothesis requires a measure of fit. In a traffic scene, a blob could be a motorcycle or a pedestrian depending on (bottom-up) its shape, and (top-down) its behaviors. The image understanding system relies on mechanisms that trade off between bottom-up "goodness-of-fit" and top-down "expected behavior"—the latter being adherence to the causal rules applied to the system.

There has been little work on applying uncertainty measures to high-level vision tasks, particularly the analysis and interpretation of dynamic object interaction in image sequences [6]. Recently, Huang *et al.* [19] and Buxton and Gong [11] used a Bayesian belief network and inference engine [2] in sequences of highway traffic scenes to produce high-level concepts like "car changing lane" and "car stalled." In general, belief networks propagate values around the network as vectors between events [28]. Belief networks are regarded as bottom-up systems, which have a lot in common with the connectionist paradigm. One problem with Bayesian inference is that each node, consisting of a set of exhaustive and mutually exclusive states, must have a set of *a priori* conditional probabilities, which are almost impossible obtain in real vision applications. In addition, uncertainties in such cases are not usually statistical [32].

To handle uncertainty, Bezdek *et al.* [6] presented several examples of fuzzy models using fuzzy certainty factors to evaluate matching degrees for high-level concepts, for instance, *sky*, *field*, and *road*. It should also be noted that such techniques produce a 2-D segmentation of the scene rather than a high-level interpretation.

Despite the fact that interpretation and understanding of traffic sequence have important applications in *intelligent highways* systems, cooperative autonomous robots, and intelligent surveillance systems, very few systems have been developed. Koller *et al.* developed a system that generates verbs to characterize the motion of the object (i.e., vehicles) [23]. Gerber and Nagel proposed a simple traffic interpretation system that is based on detected vehicle trajectories [18]. Although their systems can apparently give some simple verbal descriptions, such as "drove forward" and "turned left," they lack the ability to understand the motion and interprerate the interaction between moving objects. In addition to detecting and tracking the motion, understanding and interpretation of vehicle motion and interaction require a *cognitive* structure to combine motion trajectories with rules and to generate symbolic (e.g., linguistic) interpretation.

In this paper, we are interested in automated interpretation of dynamic activities of objects in image sequences in uncertain environments. Fig. 1 shows the basic structure of the system which consists of two major subsystems: 1) object tracking and 2) image sequence interpretation. We apply the 3-D recursive filter to tracking moving objects, which produces a robust estimate of the instantaneous 3-D space–time velocity vector of the tracked object, which is then used as a training sample for a conventional multilayer perceptron (MLP) neural network [20]. MLP identifies and categorizes useful motion and positional characteristics of the objects in the image sequence.

In the image interpretation process, uncertainties present in the trajectories and moving objects (vehicles) are handled by fuzzy belief measures. We incorporate fuzzy membership functions and belief measures in our symbolic object-oriented picture interpretation (SOO-PIN) system [14]–[16], which is based on a network-of-frames approach. This results in fuzzy SOO-PIN, which uses fuzzy membership functions to convey bottom-up goodness-of-fit for object features. Based on object belief measures and causality, we develop a set of rules for top-down interpretation. Fuzzy logic is ideally suited to this problem as it deals well with uncertainty in the data, and vagueness in the concepts, and it maps onto the high-level output description language in a natural manner. Abnormal and uncertain events can be handled *gracefully*, resulting in robust system performance [5], [24]. To demonstrate the effectiveness of our system we describe in detail a specific application to traffic scene interpretation.

Section II presents the general architecture of the SOO-PIN system for traffic interpretation that uses fuzzy sets and belief measures for handling the uncertainties in objects and trajectories; and the vagueness in descriptions. Section III describes the motion tracking algorithm used in traffic scene interpretation. In Section IV, we discuss the type of uncertainties, "object-ness" in terms of fuzzy membership functions, and fuzzy measures for dealing with trajectory assignment. Section V shows some results from our traffic implementation, which utilizes all the techniques described earlier.

II. Networked Architecture for Dynamic Scene Understanding

Research on high-level interpretation of dynamic object interactions in image sequences has received little attention despite the fact that such a vision system has significant and diverse applications in many areas; for instance, vision guided autonomous vehicles, automated video editing, dynamic biomedical image analysis, and intelligent image information access in multimedia.

A recent high-level dynamic image understanding system on SOO-PIN is discussed in [14]–[16]. The SOO-PIN architecture is based on the idea that high-level interpretation is best performed using a network of independent "processes" (called *concept-frames*), each of which is concerned with a specific aspect of the interpretation, and all of which communicate via message passing. This architecture embodies both data-driven and knowledge-based control, where automatic (`check` and `create`) messages represent the data-driven mode, and on-demand (`inquiry`) messages represent the knowledge-based

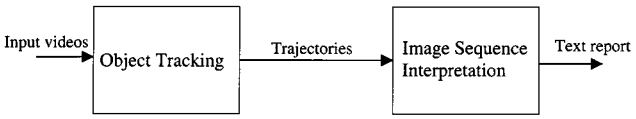

Fig. 1. Dynamic image sequence analysis system consists of two major subsystems. The input video is first processed to generate object trajectories, which is then followed by the image interpretation subsystem that uses a networked-frames architecture for motion understanding and use fuzzy measures for handling uncertainties in the trajectories. Finally, the system produces a text report.

mode (these messages are prompted from the exploration of higher level hypotheses). Using these two modes together avoids the large search space resulting from using either alone. For instance, knowledge-based control can entail exhaustive depth-first search as the system begins searching at the top nodes of the search tree and must check all nodes down to the data level, usually with failure-driven backtracking. On the other hand, data-driven search modes entail combining all data in all possible patterns in order to explore higher nodes in the search tree, and as Tsotsos [33] has pointed out, this method is NP-complete. Our conjecture that mixing top-down and bottom-up modes is more efficient than either alone, is backed up in the area of linguistic parsing by Allen [3], who gives mixed-mode chart parsing as an example. For image analysis we use fuzzy belief measures to handle uncertainties present in trajectories and moving objects (e.g., vehicles). Our system uses both fuzzy sets and fuzzy measures to model bottom-up goodness-of-fit for object features.

The traffic scene interpretation problem is rich enough to require an interesting network-of-frames, and yet constrained enough to be tractable. Real traffic scenes are interesting because they can be understood from single images, which require very complicated knowledge bases of possible dynamic behaviors of objects in the scene. However, image *sequences* and our networked architecture make it possible to obtain fruitful interpretations with a very simple knowledge base, namely, the traffic rules. The interactions of cars are a language in which the *intentionality* of the driver can be expressed, without much contamination from more direct driver-to-driver communication; that is, there is very little communication between drivers that is not expressible through the movements of their cars. The interactions, however, are completely visible to our system (an exception to this is the use of turn indicators). Consequently, it is possible from observing the cars to interpret the drivers' states of mind *vis a vis* their views on what the other cars are doing.

Fig. 2 shows a typical frame from an intersection sequence taken in downtown Melbourne, which has two sets of electric tram tracks in the middle (grey) area of the image. Bearing in mind that cars move forward on the left side of the road, try to determine the projected tracks of the eight vehicles in Fig. 2. You can do it pretty easily—this supports our contention that simple, intuitive rules can be used to analyze traffic scenarios.

In the traffic scenario we describe an application which collects temporal sequences of images of road intersections, interprets vehicle activities (e.g., vehicle A is turning right from the west), and produces legal analyses[1] of the scene such as

[1]According to Australian traffic rules, traffic travels on the left.

Fig. 2. Typical Australian traffic scene processed by the SOO-PIN system.

- whether the car is on the wrong side of road or intersection,
- when a car should give way to another. For instance
  - — give way to right at intersection;
  - — give way to oncoming when turning right;
  - — at T-intersections, cars in ending road give way to those on through road;
  - — at traffic lights or give way signs.
- traffic jams (i.e., give-way deadlocks).

The concepts above have been incorporated in the network-of-frames shown in Fig. 3, consisting of the following concept-frames.

- The primitive concept-frames `car`, `inXn`, `tInXn`, and `road` correspond to the objects either found by low-level processing (see Section IV) or given (e.g., intersection, T-intersection and road are constant). The `traj` and `collision` concept-frames calculate car velocities and determine whether they are likely to collide.
- Turn concept-frames, `carInXn`, `carTInXn`, and `carRoad` determine the containment of cars in road structures (roads and intersections), for instance, "car A is in intersection B."
- Give-way concept-frames check to see if any pairs of vehicles are in a give-way relationship. Output from the network is generated by the give-way concept-frames.

Some of the major uncertainties are:

1) the presence of a car indicated by segmented blob;
2) the concept of velocity, such as *slowing* down, *speeding*, *fast* approaching, etc.;
3) car's trajectories, i.e., whether it is turning left, right, or going straight ahead.

All these events are characterized by vague information and are difficult to handle with traditional approaches; this is why we incorporate fuzzy logic in our SOO-PIN system.

III. Tracking of Dynamic Objects

Tracking moving objects has been a research topic in computer vision for more than two decades. Many techniques have been developed [1], [4].[2] Bradshaw *et al.* proposed a system for computing 3-D motion trajectories. Their system uses an active camera platform and the Kalman filter for estimating trajectories [7]. However, this system is sensitive to camera calibration and requires a complicated camera setup for active vision. In [37], Zheng and Tsuiji presented an image projection technique with a scanning slit camera that generates dynamic, spatial-temporal images. The system relies on prior knowledge of object motion types, e.g., linear or pure rotation. Whereas their technique may be useful in dynamic image compression, it is not effective in our application.

Bruton and Bartley [9] have shown that 3-D recursive digital *linear-trajectory (LT)* filters may be used to selectively track the motion of objects in digital image sequences. Their method adaptively steers the 3-D frequency-planar passband of the filter to track the slowly time-varying, 3-D frequency-planar energy density function of the moving object in space–time. This tracking method is especially immune to additive noise and can successfully discriminate between objects that have motions that differ only slightly from that of the object being tracked. We use this algorithm to compute the motions and trajectories of cars in the traffic image sequences in our examples. Although indeed there are many motion tracking algorithms available, the 3-D motion tracking filter has the following advantages.

- *Simplicity*: The 3-D difference equation is just first-order in each of the spatial and temporal variables. Real-time (video-rate) implementations have been achieved in both dedicated hardware (as reported in [25]) and on high-speed general-purpose DSP processors, such as the Texas Instruments TMS320C80 Multimedia Video Processor.
- *Robustness*: Objects can be tracked and enhanced both in the presence of heavy noise contamination and in the presence of other, possibly occluding, objects.

To identify and categorize motion and positional characteristics of the objects in the image sequence, we use a conventional MLP neural network that is trained by the velocity vector generated by the tracking algorithm [20].

A. 3-D Linear Trajectory Signals

In this section, we briefly describe the fundamental class of 3-D linear trajectory (LT) signals and the simple 3-D recursive filter algorithms that are used to enhance LT signals.

Many important 3-D filtering problems associated with the enhancement of digitized image sequences involve 3-D signals that can be characterized in a continuous space–time domain $\boldsymbol{t} \equiv (x, y, t)^T \in \Re^3$. An LT signal $v_{\mathrm{LT}}(\boldsymbol{t})$ is defined in [10] as belonging to the class of 3-D signals for which there exists a direction $\boldsymbol{d}$ along which $v_{\mathrm{LT}}(\boldsymbol{t})$ is constant along all lines parallel to $\boldsymbol{d}$, as illustrated in Fig. 4(a). When presented as a temporal succession of frames in t, such signals have the property that they move with constant spatial velocity in the (x, y) plane [10].

The 3-D LT signal in Fig. 4(a) has a corresponding energy density spectrum that exists entirely on a signal plane passing through the origin in $\boldsymbol{\omega} \equiv (\omega_1, \omega_2, \omega_3)^T \in \Re^3$, as shown in

[2]Interested readers may find the survey paper [13] useful.

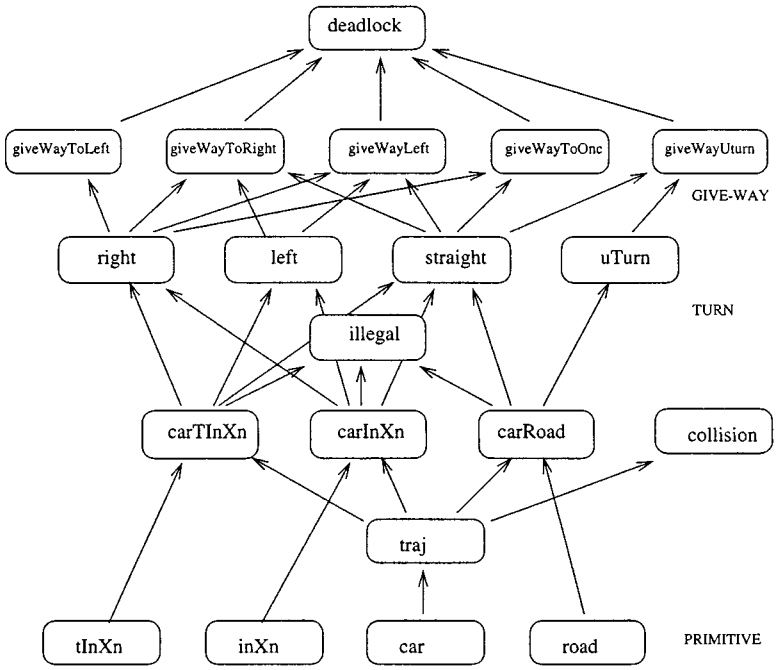

Fig. 3. Traffic scenario network-of-frames. The arrows refer to check or create messages; inquiry and update messages are not shown [16]. `inXn` refers to "intersection," `tInXn` to "T-intersection," `carInXn` to the concept of a car in an intersection, `carTInXn` refers to a car in a T-intersection, and `carRoad` to a car in a road.

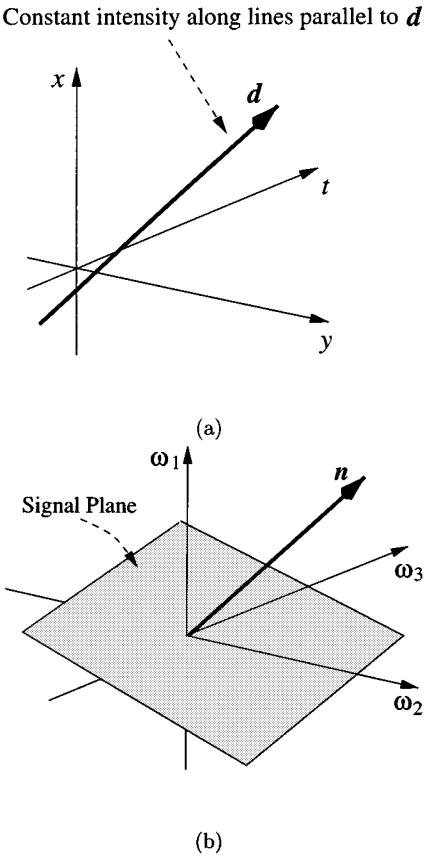

Fig. 4. Three-dimensional LT signals.

Fig. 4(b). The normal $\mathbf{n}$ to the *signal plane* has the same direction in $\boldsymbol{\omega}$ as the direction $\boldsymbol{d}$ of the LT signal in $\boldsymbol{t}$ [10].

We can generate corresponding discrete-domain LT temporal image sequences, $I_{\mathrm{LT}}(x, y, i) \triangleq v_{\mathrm{LT}}(x, y, i), x, y, i \in \mathcal{N}$, by sampling $I_{\mathrm{LT}}(\boldsymbol{t})$ on uniform time intervals. In [10] and [36], we developed a simple, robust 3-D recursive filter algorithm for the enhancement of 3-D digitized LT image sequences.

B. Adaptive 3-D Recursive Tracking Filter

In most temporal image sequences, objects do not proceed along linear trajectories, but instead exhibit temporally-smooth motion along curved trajectories in space–time. However, it is convenient to represent curved-trajectory signals in (x, y, t) space with *piecewise* linear trajectory signals in (x, y, i) space which depicts a time sequence of linear segments. The enhancement of piecewise linear trajectory signals is then carried out by using an *adaptive* 3-D LT filter whose resonant plane is oriented in every frame i to selectively enhance an LT signal having the direction of the segment $\boldsymbol{v}_i$.

Consider the output of the 3-D LT filter at frame i, and let the position of the object be centered at the pixel position $(\overline{x}_i, \overline{y}_i)^T$ and similarly at $(\overline{x}_{i-1}, \overline{y}_{i-1})^T$ in frame $i-1$. We define the instantaneous velocity vector as $\boldsymbol{v}_i$, where

$$\boldsymbol{v}_i = (\Delta\overline{x}_i\, \Delta\overline{y}_i)^T = (\overline{x}_i - \overline{x}_{i-1} \overline{y}_i - \overline{y}_{i-1})^T. \qquad (1)$$

We use the adaptive 3-D filter system shown in Fig. 5. Following [9], we determine the position of the object at the output of the 3-D LT filter in each frame by computing the intensity centroid within an object window, as shown in Fig. 6. For a window size of $X_W \times Y_W$, whose lower left corner is located at

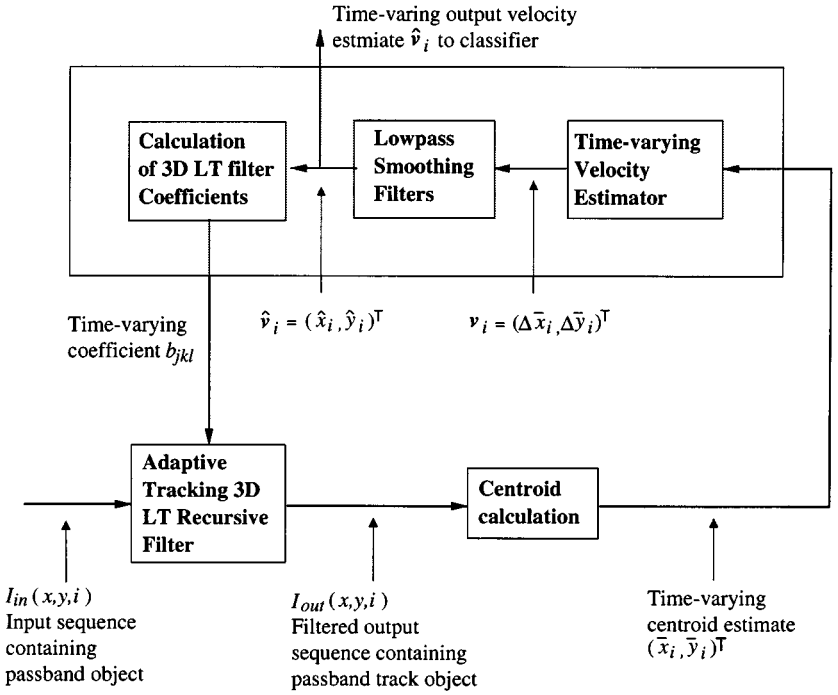

Fig. 5. Three-dimensional system for tracking a moving object and obtaining velocity estimates.

the pixel address (X_i, Y_i), the centroid of the object $(\overline{x}_i, \overline{y}_i)^T$ is given by

$$\overline{x}_i = \frac{1}{A_i} \sum_{x=0}^{X_W} \sum_{y=0}^{Y_W} I_{\text{out}}(x + X_i, y + Y_i, i)x \qquad (2a)$$

$$\overline{y}_i = \frac{1}{A_i} \sum_{x=0}^{X_W} \sum_{y=0}^{Y_W} I_{\text{out}}(x + X_i, y + Y_i, i)y \qquad (2b)$$

where $I_{\text{out}}(x + X_i, y + Y_i, i)$ is the output intensity value in image frame i, and

$$A_i = \sum_{x=0}^{X_W} \sum_{y=0}^{Y_W} I_{\text{out}}(x + X_i, y + Y_i, i) \qquad (3)$$

is the total output in frame i and used as the normalizing factor.

The object's centroid $(\overline{x}_i, \overline{y}_i)^T$ is then used as the input to the feedback portion of the system in Fig. 5, where the instantaneous time-varying velocity estimate $\boldsymbol{v}_i$ is determined. Then, $\boldsymbol{v}_i$ is filtered to get $\boldsymbol{v}_i$ using two second-order one-dimensional (1-D) lowpass recursive filters to smooth the frame-to-frame discontinuities that occur in the velocity estimates. Finally, based on the smoothed velocity estimate $\hat{\boldsymbol{v}}_i = (\hat{x}_i \hat{y}_i)^T$, we use the method in [10] to determine the time-varying coefficients $\{b_{jkl}, j, k, l = 0, 1\}$ of the 3-D LT filter for each frame i. We will use the smoothed velocity estimate $\hat{\boldsymbol{v}}_i$ to classify the trajectory of the object, as described in the following section.

C. Trajectory Classification of the Passband Tracked Object

To classify the trajectory of a passband object we use a conventional MLP network which is fed by a time-delay neural network (TDNN) [20]. The TDNN consists of a tapped delay line

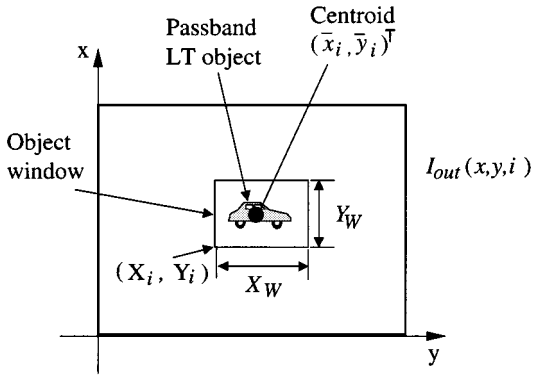

Fig. 6. Intensity centroid of the passband LT object in frame i.

of length M having unity tap weights and is used to collect a time series of M smoothed velocity estimates $\{\hat{\boldsymbol{v}}_i\}$. Velocity estimates are input to the TDNN when the 3-D tracking system has captured a particular object. After M estimates have been collected, the TDNN output vector, which represents the *static velocity profile* of the object, is used as the input to the MLP.

The MLP is a conventional three-layer network:

1) input layer;
2) hidden layer;
3) output layer.

There are $2M$ inputs for the M frames used in the estimation process. The hidden layer has N nodes. There are N outputs: Each of the N outputs corresponds to a particular velocity profile that the network has been trained to recognize. Training of the network is achieved by applying a matrix of N such velocity profiles in batch to the network and using the conventional backpropagation training algorithm [20].

D. Classification of Trajectories

The tracking and classification algorithms are used to classify the trajectories of vehicles passing through an intersection. The system runs on images recorded on videotape from fixed cameras mounted above city intersections in Melbourne (Fig. 2), and the images were digitized using an Abekas Digital Video system (in 720×576, 24-bit RGB format). A 10-s video sequence is obtained that contains vehicles moving on four distinct trajectories. The trajectories correspond to vehicles that pass directly through the intersection at constant speed and vehicles that begin left and right turns, pausing to yield to oncoming traffic. These four trajectories are shown in Fig. 7 and have been assigned the numeric classifications 0–3. A 100-input three-layer MLP is trained to classify the velocity profiles of each of these four trajectories.

The detection of vehicles for tracking and classification is automated by placing a number of fixed *sensor windows* on each lane of traffic where vehicles enter the intersection, as shown in Fig. 8. In each sensor window, we employ a fixed 3-D LT filter having a passband that is tuned to enhance vehicles moving in the direction and average speed of traffic for that lane.

The selectivity of the passband is less than that used in the tracking algorithm so that vehicles that are slightly slower or faster than the average speed are still transmitted by the filter. In Fig. 6, the total output intensity A_i in (3) for each sensor window is determined and monitored in each frame. Once A_i has exceeded a threshold in a particular window, the vehicle in the window becomes a candidate for tracking. As shown in Fig. 8, two such candidates have been identified, as indicated by the brighter window borders and cross-hairs over each vehicle's centroid.

The overall system of vehicle detection, tracking and classification is shown at work in Fig. 9. Each vehicle that is being tracked is indicated by its surrounding rectangular object window containing the LT-filtered output image and cross-hairs over the centroid. For vehicles that have been tracked for 50 frames, the 50 velocity estimates $\hat{\boldsymbol{v}}_i = (\hat{x}_i \hat{y}_i)^T$ are input to the MLP for classification. The resulting numeric classification is then superimposed on the roof of the vehicle, as shown for two vehicles in Fig. 9, which are classified as "0" and "1."

For training the MLP, we used 50 consecutive velocity estimates of the training vehicles. To generate the training samples, we manually picked one vehicle that appeared to best represent one of the four trajectories we wanted to classify. Training was performed on velocity estimates produced by the 3-D tracking filter for each of the training vehicles and was done once *off-line*. However, because there were not enough frames of the training vehicles in the original sequence, we had to interpolate (with an up-sampling ratio of 1 : 4) the frames of the training vehicles. The system achieved 100% correct trajectory classification for the given image sequence which included the 50 frames that were used for training. The detected cars and classified trajectories were then used in the subsequent high-level interpretation.

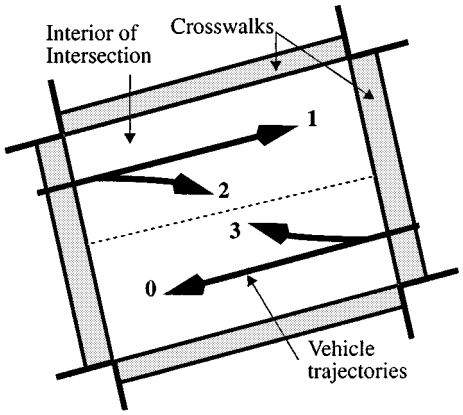

Fig. 7. Vehicle trajectory classifications.

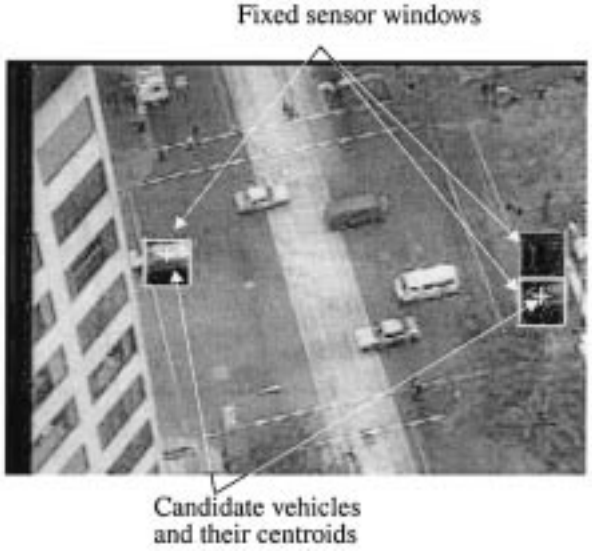

Fig. 8. Vehicle detection using sensor windows.

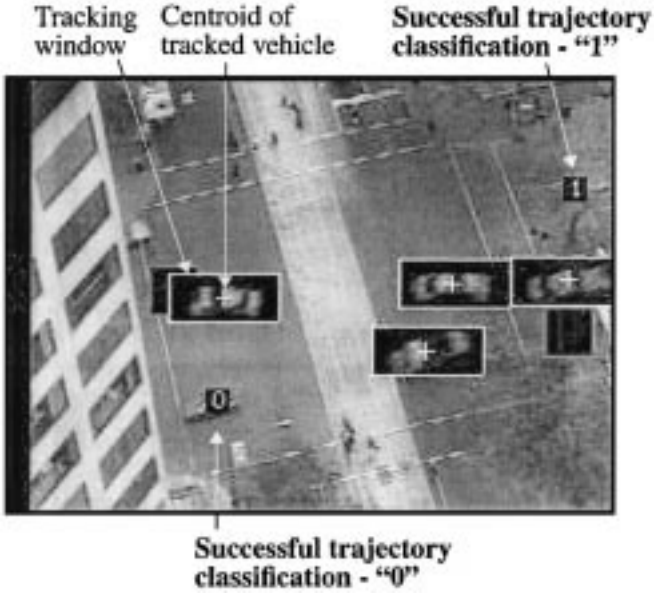

Fig. 9. Vehicle tracking and classification.

IV. Vehicle Attributes

As cars in the traffic scene are detected in sensor windows described above, they are represented as blobs with sizes defined as their *pixel counts*. Each blob is assigned major and minor

axes (orientations). Blob sizes, orientations, and trajectories, for each image and region, are stored in a database together with car identities and frame numbers. The intersection and road coordinates are also stored in this database and are accessed by the high-level network-of-frames, as shown in Fig. 10.

A. "Object-ness" for Car Description

Cars are found by isolating segments (blobs) which are output by a "change" detector. Consequently, a car's identity varies considerably. Such variations cannot be easily modeled statistically. A natural and effective way to handle the uncertainty about vehicle identity is to use fuzzy membership functions to define degrees of "car-ness" in terms of *rectangularity*, and area of the blob.

The membership function for rectangularity is based on determining the axis of symmetry of the blob. Specifically, we use the ratio r of the minimum to maximum eigenvalues of the matrix that is used to determine the principle axis of the segment [29]. A ratio of $r = 1$ indicates that the segment is squarish, and a ratio of $r = 0$ indicates a linear blob. The membership function for blob rectangularity $m_R(r)$ is shown in Fig. 11. Most cars are rectangles with an aspect ratio of about $2:1$, so this degree of rectangularity is given fuzzy membership of 1. It is possible for cars to have rectangularity of 1, so these are given a membership of 0.5.

Fig. 12 shows the membership function $m_A(a)$ that represents the extent to which a blob belongs to the set of cars. The input measurement is blob area $a = total\ pixel\ count$, and the value $m_A(a)$ determines the match between blob area and car area. The parameter $a = 0$ when the blob has no area, and $a = 1$ for a blob with an area between 2000 and 3000 pixels. The membership drops down gradually and is fixed at 0.6 $\forall\ a \geq 6000$ pixels, for larger objects.

The membership values of a potential car in these two fuzzy sets are combined with the min operator because the segment's "car-ness" depends upon the segment having both an appropriate area and rectangularity (i.e., conjunction). More formally, let B denote a detected blob with ratio r_B and area a_B. We define

$$\mu(B) = \min\{m_R(r_B),\, m_A(a_B)\}. \tag{4a}$$

The minimum is used here, but more generally, any T-norm [22] can be used to aggregate these two pieces of evidences

$$\mu_T(B) = T(m_R(r_B),\, m_A(a_B)). \tag{4b}$$

The confidence (membership) value $\mu(B)$ is then conveyed through the network as a parameter attached to each blob B.

One problem with this scheme is that the area a measured in pixels depends upon the distance between the road and the camera, and its focal length. Ideally, normalization of a should take place before finding $m_A(a)$. However, in the experiments reported here, these factors do not vary much, so we have chosen $m_A(a)$ reasonably flat. This observation also applies to rectangularity when the view is oblique, in which case an affine transformation is required to restore the apparent view to the vertical. However, in our system the view has to be close to the vertical to avoid vehicle occlusion, so problems with rectangularity are also avoided. A more sophisticated vision system based on, for instance, model-based object recognition, would have to overcome the need for vertical views and account for area normalization.

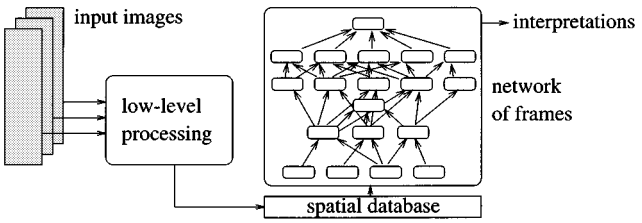

Fig. 10. Overall processing structure, with three images on the left processed by the low-level system. The network-of-frames generates the interpretations.

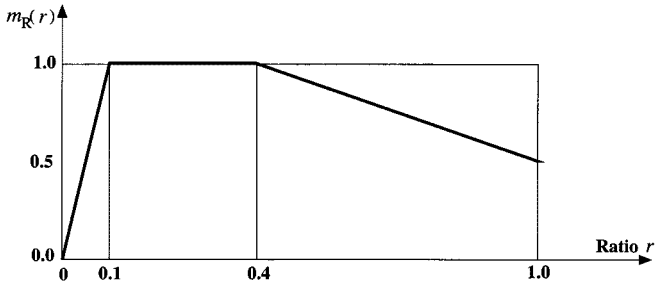

Fig. 11. Fuzzy set associated with car rectangularity.

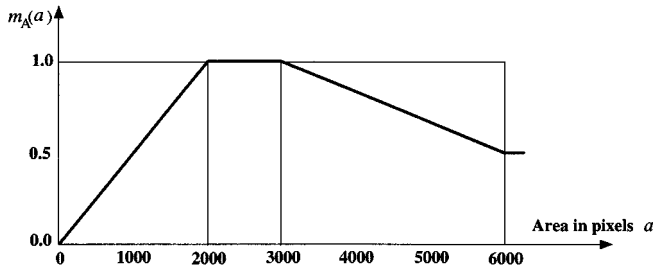

Fig. 12. Fuzzy set associated with car area, measured in pixels.

B. Computing Car Velocities

In image sequences with many dynamic objects, measuring velocity generally involves the correspondence problem, that is, finding segments in two or more images in the sequence that are the *same* object. Although our tracking algorithm and MLP are able to track and classify trajectories with very high accuracy for images with multiple moving objects we still have a problem of consistent trajectory assignment—that is, we must make sure that Car A's trajectory will not be assigned to Car B throughout the image sequence. Fortunately, we can impose some constraints on vehicle movements in the matching process. Matching has to take uncertainties into consideration. The system must choose the *best* match from the possible matches. Fuzzy sets provide an ideal mechanism for dealing with such uncertainties because the dynamical constraints (in our case involving the inertia of cars) imposed on possible matches are not inherently statistical, but rather, express concepts like "degree of fit to possible trajectory." The system has to deal with ambiguities resulting from

- objects moving in close proximity to each other;

- objects moving in parallel;
- objects moving along crossing trajectories.

The system described below is able to disambiguate these situations through dynamical constraints, each constraint providing a source of evidence for that trajectory.

Trajectory matching is based on the computed centroids and orientations of the blobs (cars) in every three frames. The middle frame is called the *master frame*, and each car found in this frame is compared with cars in the preceding and succeeding frames. A pairing is made if the computed velocity required to move the car from a frame to its successor is within specified bounds, and if the velocity is consistent with the orientations of the two blobs as determined from their rectangularity. The set of possible pairings from the previous to middle frame is compared with that from the middle to successor frame, and the pair of pairs with the highest confidence value is chosen as the most likely trajectory of the car.

What confidence value can we use for the match between two pairings of cars in three frames? Frames are about one-third of a second apart, for the pair of frame at times t and $t+1$ we have

- the apparent velocity $V_{t,\,t+1}$;
- the rotation of the principle axis $\phi_{t,\,t+1}$.

From the velocity and rotation data from each potential trajectory *triple* we construct six measures of how *good* or *believable* the trajectory is. For instance, given the momentum and time constraints, it is unlikely that in the car can dramatically change its heading in the third frame (Fig. 16 graphically illustrates these two constraints). Therefore, the problem of matching trajectories to cars becomes that of assigning confidence to potential trajectories of the cars. The six measures we use are then treated as information sources [32], and are combined with their estimated *importances* using (11).

C. Fuzzy Integrals

Image understanding often entails the need to combine information from several sources. This can be done in a number of ways; for instance, using a Bayesian approach, D-S, or fuzzy logic [6]. One technique that has enjoyed success in other vision applications is the use of fuzzy measures. In this approach, the information sources are given grades of compatibility, and their evidence is weighted and combined accordingly. In this paper, we follow the method proposed by Tahani and Keller [32].

In 1977, Sugeno [31] introduced a fuzzy integral and fuzzy measure, and proposed a set of fuzzy measure axioms. In particular, Sugeno's measure obeys the following properties. Let X be any set, let $\wp(X)$ be the power set of X, and let g: $\wp(X) \rightarrow [0,\,1]$. We call g a fuzzy measure if $\forall\ A,\ B,\ A_i \in \wp(X)$

$$g(\phi) = 0; \quad g(X) = 1 \tag{5a}$$

$$g(\phi) \geq g(A), \quad \text{if } B \supset A \tag{5b}$$

$$\text{if } \{A_i\}_{i=1}^{\infty} \text{ is monotonic, then } \lim_{i\rightarrow\infty}\{g(A_i)\} = g\left(\bigcup_{i=1}^{\infty} A_i\right). \tag{5c}$$

When g satisfies (6), it is called a λ-fuzzy measure, and we write g_λ instead of g

$$g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B). \tag{6}$$

See [22] and [34] for more detailed discussions.

In practice, we must determine a good value for λ in (6) given the value of the measure for the singletons in X, i.e., a value which is useful for this dynamic image understanding application. Let $X = \{x_1, x_2, \ldots, x_n\}$, and let $g^i = g_\lambda(\{x_i\})$. The values $\{g^i\colon i = 1, 2, \ldots, n\}$ are called the fuzzy densities associated with X, and are interpreted as the importance of the individual (singleton) information sources. Now, suppose $A \subset X$, say, $A = \{x_{i_1}, \ldots, x_{i_m}\}$. A is viewed as a set of information sources, and the g_λ measure of A $g_\lambda(A)$ is regarded as the importance of that subset of sources for answering some question. The measure of A is [27]

$$g_\lambda(A) = \sum_{j=1}^{m} g^{i_j} + \lambda \sum_{j=1}^{m-1} \sum_{k=j+1}^{m} g^{i_j} g^{i_k} + \cdots + \lambda^{m-1} g^{i_1} g^{i_2} \cdots g^{i_m}. \tag{7}$$

For $\lambda \neq 0$ and $A = X$, this can be rewritten as

$$g_\lambda(X) = \frac{1}{\lambda}\left[\prod_{i=1}^{n}(1 + \lambda g^i) - 1\right]. \tag{8}$$

The value of λ can be found from this expression since, from the definition of fuzzy measure, $g_\lambda(X) = 1$

$$1 + \lambda = \prod_{i=1}^{n}(1 + \lambda g^i). \tag{9}$$

In [31], it is shown that there is a unique solution for this expression for $\lambda > -1$ and $\lambda \neq 0$.

When g is a g_λ-measure, the values of $g_\lambda(A_i)$ can be determined recursively as [32]

$$g_\lambda(A_1) = g_\lambda(\{x_1\}) = g^1 \tag{10}$$

$$g_\lambda(A_i) = g^i + g_\lambda(A_{i-1}) + \lambda g^i g_\lambda(A_{i-1}) \quad 1 < i \leq n \quad \text{and} \quad A_i = x_1, \ldots, x_i. \tag{11}$$

There are a number of interpretations of the meaning of fuzzy measures. We regard g_λ as the belief assignable to an object regarding its membership in a class given the densities from each of the sources. The fuzzy density (the importance of the sources) can be derived from training data, or from subjective evaluations by experts [32], [21].

To generate the six measures we need, we first calculate the match between frame trajectory pairs for the following four rates:

- average speed over three sequential image frames (V_{avg});
- speed difference (ΔV);
- difference between velocity rotation rate and principle axis (orientation) rotation rate (ΔVO);
- rotation rate difference (ΔR).

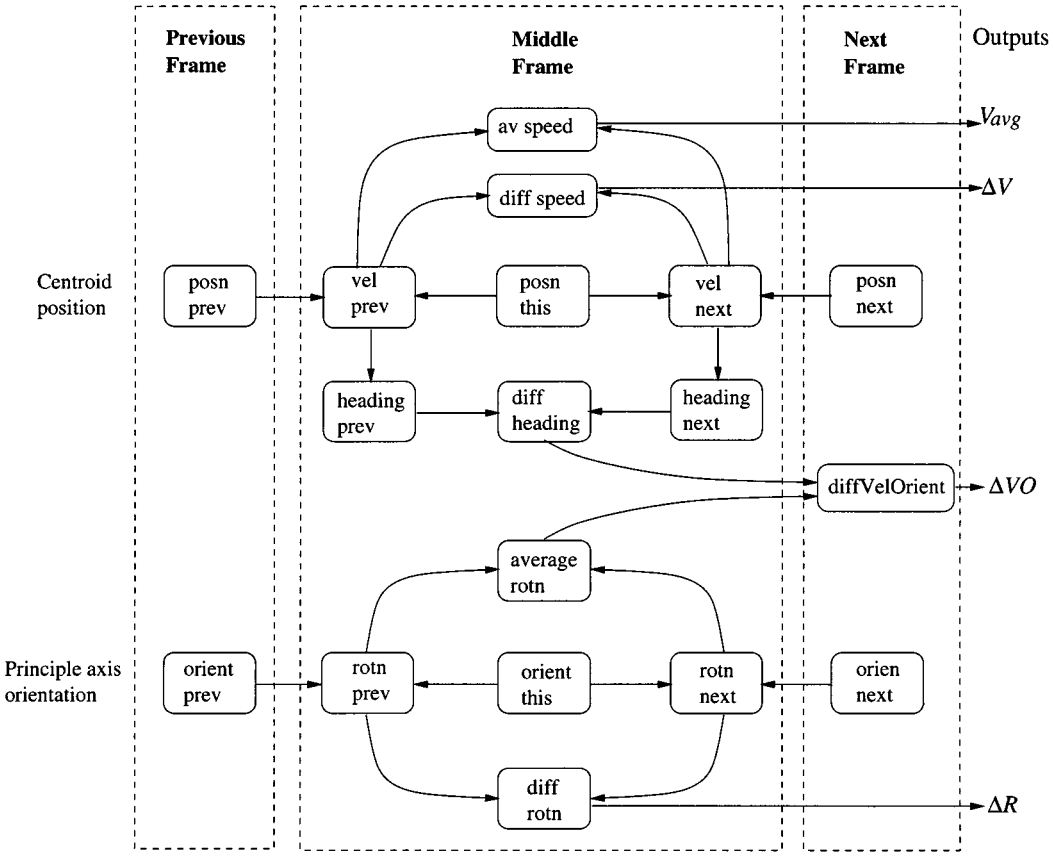

Fig. 13. Data flow for calculating input values for fuzzy processing.

The data flow diagram for these calculations is shown in Fig. 13.

The centroid position and orientation of the cars in the three frames are used to produce `avSpeed` (V_{avg},), `diffSpeed` (ΔV), `diffVelOr` (ΔVO), and `diffRotn` (ΔR). The last three measures are inverted by subtracting them from 1. This renders value 1 when there is certain evidence for the trajectory. The fuzzy densities $\{g^i\}$ are obtained by trial and error. The set of densities that gave the best matching capability are

$$\begin{aligned} x_1 &= \text{LowDiffOK} & g^1 &= 0.4 \\ x_2 &= \text{LowSpeedOK} & g^2 &= 0.25 \\ x_3 &= \text{NormSpeedOk} & g^3 &= 0.25 \\ x_4 &= \text{BadRotn} & g^4 &= 0.1 \\ x_5 &= \text{OutOB1} & g^5 &= 0.3 \\ x_6 &= \text{OutOB2} & g^6 &= 0.2 \end{aligned} \tag{12}$$

$\{x_1, \ldots, x_6\}$ are the frames of discernment for sets of membership functions that describe linguistic values for each of the four liguistic variables (see Figs. 14 and 15). When a 4-tuple is evaluated, the resulting membership values are combined as shown in (13a)–(13f) to produce values for the six measures. The left-hand sides of (13a)–(13f) use a total of 13 values extracted from the four membership functions represented by Figs. 14 and 15 to make an evaluation

$$\text{lowDiffSpeed} \wedge \text{lowDiffRotn} \wedge \text{lowDiffVelOr} \Rightarrow \text{LowDiffOK} \tag{13a}$$

$$\text{lowAvSpeed} \wedge \text{lowDiffSpeed} \wedge \text{lowDiffRotn} \Rightarrow \text{LowSpeedOK} \tag{13b}$$

$$\text{okDiffSpeed} \wedge \text{okDiffRotn} \wedge \text{okDiffVelOr} \Rightarrow \text{NormSpeedOK} \tag{13c}$$

$$\text{lowAvSpeed} \wedge \text{medDiffRotn} \Rightarrow \text{BadRotn} \tag{13d}$$

$$\text{highDiffSpeed} \vee \text{highAvSpeed} \vee \text{highDiffRotn} \Rightarrow \text{OutOB1} \tag{13e}$$

$$\text{medAvSpeed} \wedge \text{highDiffVelOr} \Rightarrow \text{OutOB2}\,. \tag{13f}$$

Using (9), we obtain a g_λ-measure with $\lambda = -0.6763$. The pair with the highest fuzzy measure of match is chosen as the correct match, and is used to create a trajectory instance.

D. Velocity Examples

In order to demonstrate the effectiveness of the g_λ fuzzy measure, we generated the scenarios in Fig. 16, of which three are **abnormal**.[3]

[3]Abnormal scenarios do **not** occur in actual traffic image sequences because constraints make this physically impossible.

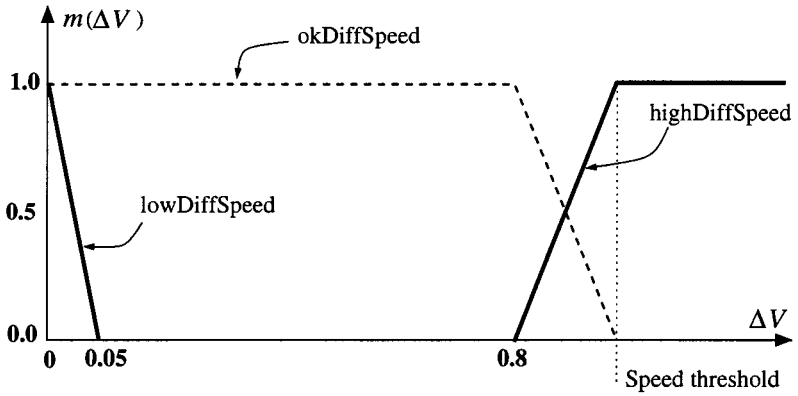

Fig. 14. Fuzzy sets in the speed difference (ΔV) frame of discernment. The sets are `lowDiffSpeed`, `okDiffSpeed`, and `highDiffSpeed`. The fuzzy sets in the frames of discernment `diffRotn` (ΔR) and `diffVelOr` (ΔVO) are the same.

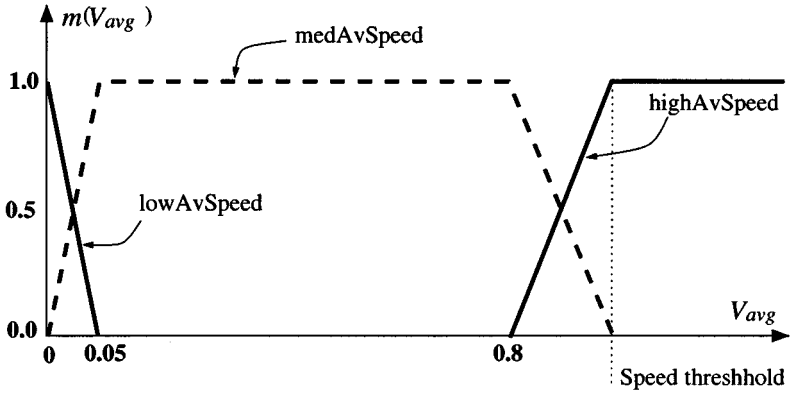

Fig. 15. Fuzzy sets in the average speed (AV) frame of discernment. The sets are `lowAvSpeed`, `medAvSpeed`, and `highAvSpeed`.

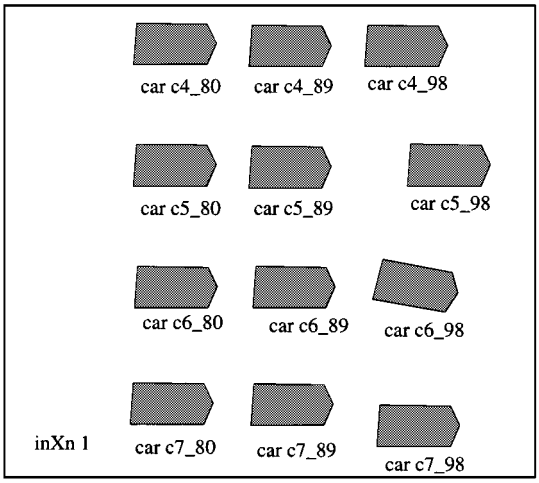

Fig. 16. Diagram of four car trajectories.

In Fig. 16, various bad matches are demonstrated. The top car is normal, the rest have a problem designed to provoke a low confidence in the matching. Each car is shown in three successive frames, where the frame number is encoded in the car labels as $car\ c\ \langle car\ number\rangle\_\langle frame\ number\rangle$. Car c5 has an unusually high acceleration, car c6 changes its rotation rate too fast, and car c7 slides sideways. For comparison, the trajectory of car c4 is normal. The output from a run for each of the cars is presented in Tables I–IV. Notice that many of the membership functions used on the left sides of (13a)–(13f) take zero values for this input 4-tuple. The last column in each table shows the evidence for each of the six hypotheses. The total confidence in the match from (11) is displayed in the bottom right cell of each table.

Let us have a close look at Tables I–IV to see how the confidence values are computed. In the tables we list all the fuzzy membership functions and their values that have been used in (13a)–(13f) for computing the six measures. The total confidence in match is calculated using (11). Table I is associated with car c4 in Fig. 16 that has a normal movement. As expected, car c4 activated no low confidence values in the fuzzy functions and produced the high fuzzy integral value of 0.81. In Table II, car c5 has a high ΔV of 1.59, which activates the highDiffSpeed fuzzy set. This in turn activates the evidence source outOB giving a low fuzzy integral value of 0.28.

Fig. 16 shows that car c6 has a highly abnormal movement; it had a spin in frame c6_98, which gives a high ΔR and a high ΔVO, resulting in two outOBs and thus a low value 0.29. The frame c7_98 in Fig. 16 shows that car c7 skidded, which produced a high ΔVO that in turn activated outOB1 resulting in a low fuzzy integral value of 0.38.

E. Fuzzy Linguistic Hedges

When the car's activities have been identified, we can follow the process in the SOO-PIN [16] to derive the relationships between cars. The fuzzy mechanisms and logical combination

TABLE I
RESULTS FOR CAR c4_98 WHEN INPUTS ARE $V_{avg} = 0.63$, $\Delta V = 0.036$, $\Delta R = 0.0$, AND $\Delta VO = 0.07$

Eqn	Left Side						Right Side
(13*a*)	lowDiffSpeed = 1	∧	lowDiffRotn = 1	∧	lowDiffVelOr = 0.81		LowDiffOK = 0.81
(13*b*)	lowAvSpeed = 0	∧	lowDiffSpeed = 1	∧	lowDiffRotn = 1		LowSpeedOK = 0.00
(13*c*)	okDiffSpeed = 1	∧	okDiffRotn = 1	∧	okDiffVelOr = 1		NormSpeedOK = 1.00
(13*d*)			lowAvSpeed = 0	∧	medDiffRotn = 0		BadRotn = 1.00
(13*e*)	highDiffSpeed = 0	∨	highAvSpeed = 0	∨	highDiffRotn = 0		OutOB1 = 1.00
(13*f*)			medAvSpeed = 1	∧	highDiffVelOr = 0		OutOB2 = 1.00
(11)	Total Confidence Value:						0.99

TABLE II
RESULTS FOR CAR c5_98 WHEN INPUTS ARE $V_{avg} = 0.74$, $\Delta V = 1.59$, $\Delta R = 0.0$, AND $\Delta VO = 0.12$

Eqn	Left Side						Right Side
(13*a*)	lowDiffSpeed = 0	∧	lowDiffRotn = 1	∧	lowDiffVelOr = 0		LowDiffOK = 0.00
(13*b*)	lowAvSpeed = 0	∧	lowDiffSpeed = 0	∧	lowDiffRotn = 1		LowSpeedOK = 0.00
(13*c*)	okDiffSpeed = 0	∧	okDiffRotn = 1	∧	okDiffVelOr = 1		NormSpeedOK = 0.00
(13*d*)			lowAvSpeed = 0	∧	medDiffRotn = 0		BadRotn = 1.00
(13*e*)	highDiffSpeed = 1	∨	highAvSpeed = 0	∨	highDiffRotn = 0		OutOB1 = 0
(13*f*)			medAvSpeed = 1	∧	highDiffVelOr = 0		OutOB2 = 1.00
(11)	Total Confidence Value:						0.28

TABLE III
RESULTS FOR CAR c6_98 WHEN INPUTS ARE $V_{avg} = 0.65$, $\Delta V = 0.16$, $\Delta R = 0.86$, AND $\Delta VO = 0.24$

Eqn	Left Side						Right Side
(13*a*)	lowDiffSpeed = 0.61	∧	lowDiffRotn = 0	∧	lowDiffVelOr = 0		LowDiffOK = 0.00
(13*b*)	lowAvSpeed = 0	∧	lowDiffSpeed = 0.61	∧	lowDiffRotn = 0		LowSpeedOK = 0.00
(13*c*)	okDiffSpeed = 1	∧	okDiffRotn = 0.28	∧	okDiffVelOr = 0.54		NormSpeedOK = 0.28
(13*d*)			lowAvSpeed = 0	∧	medDiffRotn = 0.28		BadRotn = 1.00
(13*e*)	highDiffSpeed = 0	∨	highAvSpeed = 0	∨	highDiffRotn = 0.71		OutOB1 = 0.29
(13*f*)			medAvSpeed = 1	∧	highDiffVelOr = 0.45		OutOB2 = 0.54
(11)	Total Confidence Value:						0.29

rules used by SOO-PIN attach a fuzzy integral value to each possible interpretation of car activity in the monitored scene. The belief measures are used to modify the linguistic output of the system as describe in Section IV-C and thereafter.

Table V shows the correspondence between the values of any belief measure and the linguistic hedges used to modify SOO-PIN output statement.

For instance, in the following output from the system:

∗ *Give Way to left-turner*: *car c*4_98 *turning right from west has probably given way to car c*2_98 *from east*.

the hedge "probably" is inserted before the verb "given way" because the activity interpretation had a total belief value of 0.67.

TABLE IV
RESULTS FOR CAR c7_98 WHEN INPUTS ARE $V_{avg} = 0.67$, $\Delta V = 0.21$, $\Delta R = 0.0$, AND $\Delta VO = 0.41$

Eqn	Left Side					Right Side
(13a)	lowDiffSpeed = 0	$\wedge$	lowDiffRotn = 1	$\wedge$	lowDiffVelOr = 0	LowDiffOK = 0.00
(13b)	lowAvSpeed = 0	$\wedge$	lowDiffSpeed = 0	$\wedge$	lowDiffRotn = 1	LowSpeedOK = 0.00
(13c)	okDiffSpeed = 1	$\wedge$	okDiffRotn = 1	$\wedge$	okDiffVelOr = 0	NormSpeedOK = 0.00
(13d)			lowAvSpeed = 0	$\wedge$	medDiffRotn = 0	BadRotn = 1.00
(13e)	highDiffSpeed = 0	$\vee$	highAvSpeed = 0	$\vee$	highDiffRotn = 0	OutOB1 = 1.00
(13f)			medAvSpeed = 1	$\wedge$	highDiffVelOr = 1	OutOB2 = 0.00
(11)	Total Confidence Value:					0.38

TABLE V
CORRESPONDENCE BETWEEN TOTAL BELIEF MEASURES AND LINGUISTIC HEDGES

Total Belief measure	Output Hedge
0.0–0.05	No output
0.05–0.2	conceivably
0.2–0.5	possibly
0.5–0.9	probably
0.9–1.0	No hedge

V. RESULTS

Fig. 17 shows a typical input image together with an intermediate result showing positions of cars found and their velocities (indicated by arrows whose length is proportional to car speed), if any, within the intersection boundary. The intersection boundary is input manually. If camera position is fixed, this has to be done once at the system initialization. North is up in both views. The following shows the SOO-PIN generated interpretation of the activity of one car, c5_116.

$*$ *Give Way to oncoming*: *car* c5_116 *turning right from west has probably given way to car* c4_116 *from east*. (14a)

$*$ *Give Way to oncoming*: *car* c5_116 *turning right from west has possibly given way to car* c3_116 *from east*. (14b)

$*$ *Give Way to oncoming*: *car* c5_116 *turning right from west has probably given way to car* c2_116 *from east*. (14c)

$*$ *Give Way to left-turner*: *car* c5_116 *turning right from west has possibly given way to car* c1_116 *from east*. (14d)

$*$ *Give Way to left-turner*: *car* c5_116 *turning right from west has possibly given way to car* c6_116 *from east*. (14e)

$*$ *Give Way to left-turner*: *car* c5_116 *turning right from west has probably given way to car* c7_116 *from west*. (14f)

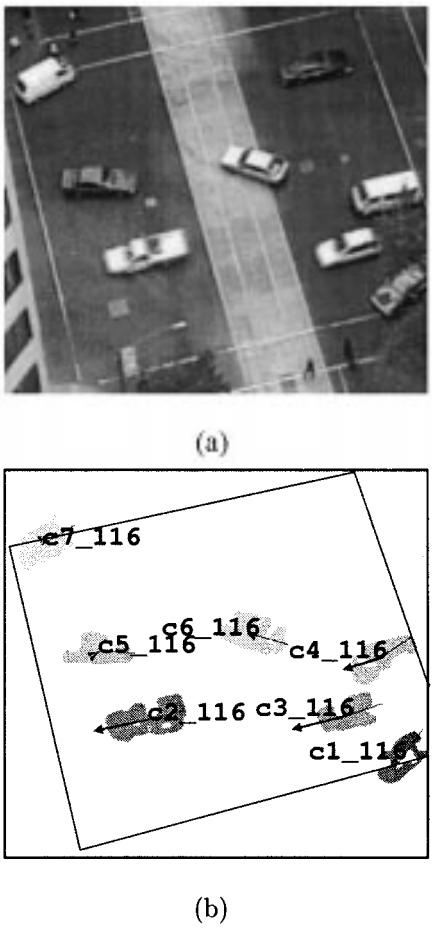

Fig. 17. (a) Traffic scene, middle image of the triple. (b) Cars found by system with labels attached, and velocity given by arrow lengths.

The fuzzy SOO-PIN correctly found each interaction between c5_116 and the other cars. However, SOO-PIN

completely ignored the black car at top right in Fig. 17(a), which was not detected because its intensity merged it with the background. The cars $c3\_116$ and $c4\_116$ are visible for only two frames, so the algorithm does not interpret other car activities for them, but their activities are interpreted for the other cars [as in (14a) and (14b)]. In this example SOO-PIN produced interpretations similar to (14a)–(14f) for all other four cars: c1_116, c2_116, c6_116, and c7_116, which resulted in a total of 30 interpretation statements. Table VI shows the fuzzy belief measures computed by (11) and were used in interpretation (14a)–(14f).

Cars $c1\_116$ and $c5\_116$ have very high ΔVO values because they are stopped and thus have arbitrary headings. Trajectory beliefs are such as those in Table VI then fed into the higher level interpreter (figure hierarchy) and are incorporated in the final belief values generated by SOO-PIN.

The next example (Fig. 18) depicts another intersection, with a lower camera resulting in an oblique view. Conventions are the same as in Fig. 17: north is up, the intersection boundary is manually inserted, and arrow lengths are proportional to car speeds. This camera angle stresses the system, reducing the rectangularity of some of the cars and thus reducing the output fuzzy beliefs. Note, however, that all seven cars, including partially occluded car c3, are correctly detected during the segmentation procedure. Sample interpretations are given as follows:

$*$ *Give Way to left-turner*: *car* $c4\_150$ *turning right from west has possibly given way to car* $c1\_150$ *from east*. (15a)

$*$ *Give Way to left-turner*: *car* $c5\_150$ *turning right from west has possibly given way to car* $c1\_150$ *from east*. (15b)

In this case, every car activity was correctly identified, and interactions in the interpretation shown in (15a) and (15b) are correct. You can see from the lengths of the arrows how c2 and c7 are speeding through the intersection, and the turners are creeping forward with short but correctly oriented velocity arrows. The linguistic hedge "possibly" has been generated in interpretations (15a) and (15b), reflecting the lower values of the belief measure for this situation.

Finally, Fig. 19 shows a case where an illegal turn occurred. conventions are as in previous figures. This is a T-intersection ($tInXn1$), with the straight-through traffic traveling east–west. The intersecting roadway enters the east–west thoughway from the south, and the intersection is barely visible in Fig. 19. A sample interpretation follows:

$*$ *Illegal*: *car* $c5\_316$ *from east is possibly doing a right turn on the wrong side of* $tInXn1$. (16)

The car, c5_316, is doing an illegal U-turn across tram tracks in the intersection from the east and back to the east, currently facing north west. The system detects that c5_316 is at an unusual angle in this T-intersection and deduces that it is doing a right-turn illegally, i.e., into a nonroad [remember this is in Australia, where cars are moving toward us in Fig. 19 in the bottom of the image from right to left (east to west)]. Since the other cars did not have three frames for processing, the fuzzy SOO-PIN can produce only the single interpretation output with the linguistic hedge "possibly." Another running sequence can be viewed on the website: http://www.cs.mu.oz.au/~zliu on the UNIX system and can be played with the UNIX command mpeg_play.

TABLE VI
PARAMETER VALUES AND BELIEF MEASURES FOR GENERATING THE STATEMENTS IN (14a)–(14f)

Car	V_{avg}	ΔV	ΔR	ΔVO	Belief
c1_116	0.0031	0.034	0.14	13.02	0.64
c2_116	0.29	0.28	0.38	0.35	0.37
c5_116	0.00	0.007	0.24	11.60	0.63
c6_116	0.036	0.09	0.50	0.29	0.38
c7_116	0.00	0.015	0.28	1.64	0.57

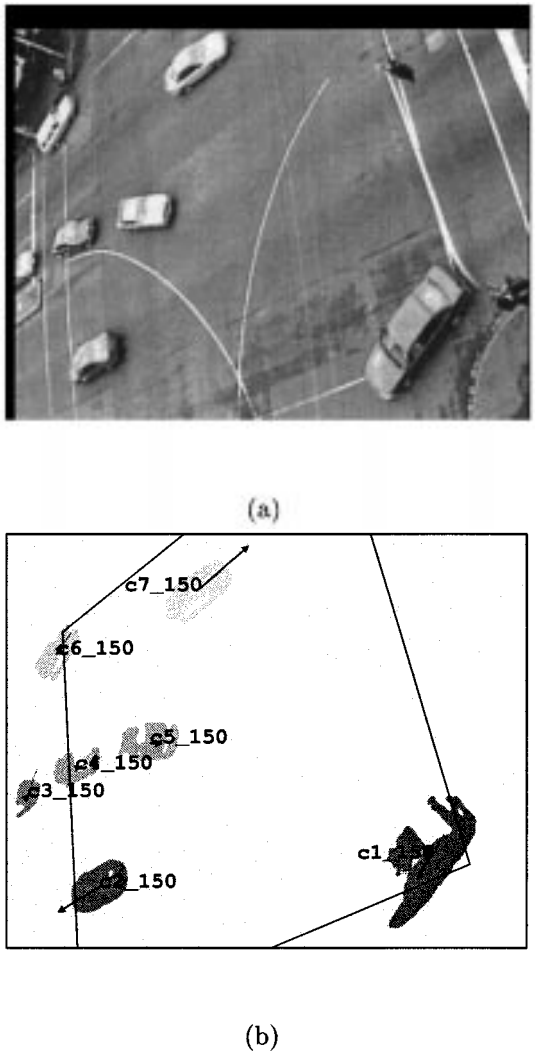

Fig. 18. Traffic scene. (a) Middle image of the triple. (b) Processed image with car labels. Velocities are shown as arrow lengths. The intersection boundary is input manually; north is up.

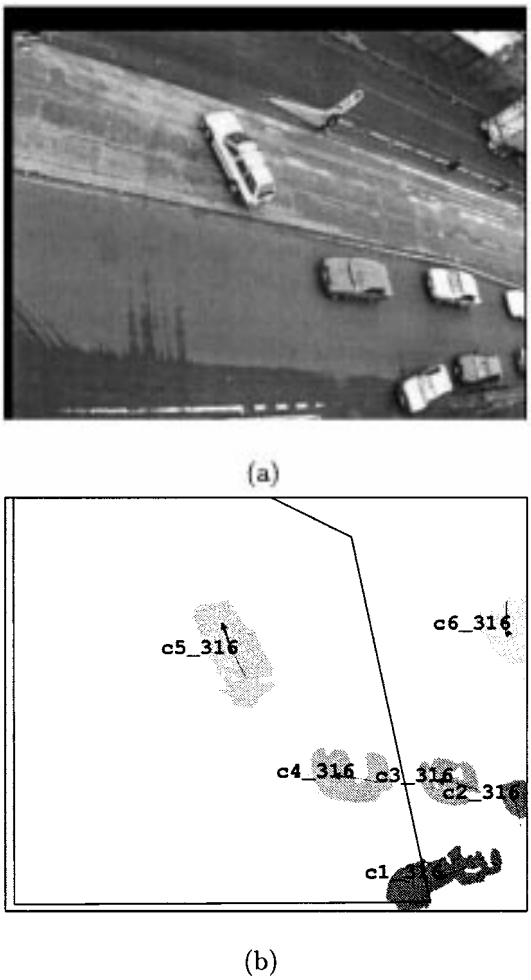

Fig. 19. Traffic scene. T-intersection with car doing illegal U-turn. Oblique view looking east. (a) The second (middle) image of the triple. There is a intersection close to the bottom right-hand corner (on the south boundary) of the image, where a light colored car is turning left (due south). (b) Processed image with car labels, where the boundary was manually defined for the camera at this intersection.

To judge the effectiveness of the fuzzy sets and measures, the system described was run on 27 images (nine triples of frames, three sets of triples from three different intersections), when using our original version (the *crispt* version) of SOO-PIN [16] that does not use the fuzzy neasures, we obtained 19 correct interpretations and five erroneous interpretations (i.e., 79% correct), as judged by visual inspection. When the interpretations were weighted by fuzzy beliefs, the accuracy increased to 89%. This indicates that several of the wrong interpretations in the crisp system have lower-than-average belief values, in other words, that our fuzzy belief calculations improved system performance by more than 12% for this small set of test cases. All of the wrong interpretations in both versions of SOO-PIN involved cars that were segmented into two regions due to a section of the car merging with the background, or being obscured by foreground. On an old Sun SparcStation II, the high-level (Parlog++) processing took 97.4 s of CPU time to process the nine scene triples, i.e., just over 10 s per scene. This was achieved with no attempt to optimize the system for speed. In all our tests, the images were digitized using an Abekas Digital Video system and were in 720×576, 24-bit RGB format.

VI. Conclusion

We have presented a new approach to interpretation of dynamic object interactions in temoral image sequences using fuzzy sets and measures. To track and classify moving objects we used a multidimensional filter-based tracking algorithm and a simple MLP. Uncertainties in the assignment of trajectories and the description of objects can be handled effectively by fuzzy logic and fuzzy measures.

The effectiveness of our system was demonstrated in the domain of complex traffic scene analysis. We discussed the uncertainties present in the description of *car-ness*, vehicle speed, vehicle trajectories, and other activities in traffic scenes. We used different fuzzy membership functions to represent and manipulate these uncertainties.

Our experiments show that the incorporation of fuzzy mechanisms enables the system to handle diverse uncertain situations and to produce natural and consistent results.

At present, the use of fuzzy measures in SOO-PIN is limited to performing the correspondence for car velocity determination, identifying segments as cars, and modifying the car activity calculations with belief measures. We hope to extend the use of fuzzy measures throughout the SOO-PIN network. We believe this will usefully refine interpretative power in scene understanding. Recently, we have developed a new low-level processing technique using motion cue and fuzzy measures for detecting cars that have lower contrast [26]. We believe using fuzzy measure at low-level processing will further improve the system performance.

Moreover, we are currently extending our system to analyze more complicated and subtle cases such as autonomous robots, software agents, traffic interpretation, air-combat training, and medical information analysis.

References

[1] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer. A*, vol. 2, no. 2, pp. 284–299, 1985.

[2] S. K. Andersen *et al.*, "HUGIN—A shell for building Bayesian belief universes for expert systems," in *Proc. IJCAI*, Detroit, MI, 1989, pp. 1080–1085.

[3] J. Allen, *Natural Language Understanding*. Redwood City, CA: Benjamin Cummings, 1987.

[4] J. L. Barron, D. J. Fleet, and D. J. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 21, no. 1, pp. 43–77, 1994.

[5] J. C. Bezdek and S. K. Pal, "Fuzzy models for pattern recognition background, significance, and key points," in *Fuzzy Models for Pattern Recognition*, J. C. Bezdek and S. K. Pal, Eds. Piscataway, NJ: IEEE, 1992, pp. 1–27.

[6] J. C. Bezdek *et al.*, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Norwell, MA: Kluwer, 1999.

[7] K. Bradshaw, I. D. Reid, and D. W. Murray, "The active recovery of 3-D motion trajectories and their use in prediction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 219–233, Mar. 1997.

[8] L. Bretzner and T. Lindeberg, "Feature tracking with automatic selection of spatial scales," *Comput. Vis. Image Understand.*, vol. 71, no. 3, pp. 385–392, Sept. 1998.

[9] L. T. Bruton and N. R. Bartly, "The enhancement and tracking of moving objects in digital images using adaptive three-dimensional recursive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 604–612, June 1986.

[10] ——, "Three-dimensional image processing using the concept of network resonance," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 664–672, July 1985.

[11] H. Buxton and S. G. Gong, "Visual surveillance in a dynamic and uncertain world," *Artif. Intell.*, vol. 78, pp. 431–459, 1995.
[12] M. Cayrol, H. Farreny, and H. Prade, "Fuzzy pattern matching," *Kybernetes*, vol. 11, pp. 103–116, 1982.
[13] C. Cedres and M. Shah, "Motion-based recognition: A survey," *Image Vis. Comput.*, vol. 13, no. 2, pp. 129–155, Mar. 1995.
[14] S. Dance and T. Caelli, "A symbolic object-oriented picture interpretation network: SOOPIN," in *Advances in Structural and Syntactic Pattern Recognition, Proceedings of the International Workshop*, H. Bunke, Ed. Singapore: World Scientific, 1993, Machine Perception and Artificial Intelligence, pp. 530–541.
[15] ——, "On the symbolic interpretation of traffic scenes," in *Proc. ACCV*, Osaka, Japan, 1993, pp. 798–801.
[16] S. Dance, T. Caelli, and Z.-Q. Liu, "Picture interpretation: A symbolic approach," in *Machine Perception and Artificial Intelligence*. Singapore: World Scientific, 1995.
[17] A. P. Dempster, "A generalization of Bayesian inference," *J. R. Stat. Soc.*, vol. 30, pp. 205–247, 1968.
[18] R. Gerber and H.-H. Nagel, "Knowledge representation for the generation of quantified natural language description of vehicle traffic in image sequences," in *Proc. IEEE ICIP*, 1996, pp. 805–808.
[19] T. T. Huang *et al.*, "Automatic symbolic traffic scene analysis using belief networks," in *Proc. AAAI*, Seattle, WA, 1994, pp. 966–972.
[20] D. R. Hush and W. G. Horne, "Progress in supervised neural networks," *IEEE Signal Processing Mag.*, vol. 10, pp. 8–39, Jan. 1993.
[21] J. Keller *et al.*, "Advances in fuzzy integration for pattern recognition," *Fuzzy Sets Syst., Special Issue Pattern Recognit.*, vol. 65, pp. 273–283, 1994.
[22] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Upper Saddle River, NJ: Prentice-Hall, 1988.
[23] D. Koller, N. Heinze, and H.-H. Nagel, "Algorithmic characterization of vehicle trajectories from image sequencies by motion verbs," in *Proc. IEEE ICCVPR*, 1991, pp. 90–95.
[24] B. Kosko, *Neural Networks and Fuzzy Systems*. Upper Saddle River, NJ: Prentice-Hall, 1992.
[25] C. J. Kulach, L. T. Bruton, and N. R. Bartly, "A real-time video implementation of a three-dimensional first-order recursive discrete-time filter," in *Proc. IEEE ISCS*, Atlanta, GA, May 1996, pp. II-699–II-702.
[26] X. B. Li, Z. Q. Liu, and K. M. Leung, "Detection of vehicles from traffic scenes using fuzzy measures," *Pattern Recognit.*, 2001, in press.
[27] K. Leszczynski, P. Penczek, and W. Grochulski, "Sugeno's fuzzy measure and fuzzy integral," *Fuzzy Sets Syst.*, vol. 15, pp. 147–158, 1985.
[28] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufman, 1988.
[29] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed. New York: Academic, 1982.
[30] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press, 1976.
[31] M. Sugeno, *Fuzzy Measures and Fuzzy Integrals—A Survey*. Amsterdam, The Netherlands: North-Holland, 1977, ch. 6, pp. 89–102.
[32] H. Tahani and J. M. Keller, "Information fusion in computer vision using the fuzzy integral," in *Fuzzy Measure Theory*. New York: Plenum, 1992, pp. 319–341.
[33] J. K. Tsotsos, "The complexity of perceptual search tasks," in *Proc. Int. Joint Conf. Artif. Intell.*, N. S. Sridharan, Ed. San Mateo, CA: Morgan Kaufman, 1989, pp. 1571–1577.
[34] S. T. Wierzchon, "Applications of fuzzy decision-making theory to coping with ill-defined problems," *Fuzzy Sets Syst.*, vol. 7, pp. 1–18, 1992.
[35] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning: Part I," *Inform. Sci.*, vol. 8, pp. 199–249, 1975.
[36] Y. Zhang and L. T. Bruton, "Applications of 3-D LCR networks in the design of 3-D recursive filters for processing image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. CSVT-4, pp. 369–382, Aug. 1994.
[37] J. Zheng and S. Tsuiji, "Generating dynamic projection images for scene representation and understanding," *Comput. Vis. Image Understand.*, vol. 72, no. 3, pp. 237–256, Dec. 1998.

Zhi-Qiang Liu (S'82–M'86–SM'91) received the M.A.Sc. degree in aerospace engineering from the Institute for Aerospace Studies, University of Toronto, Toronto, ON, Canada, and the Ph.D. degree in electrical engineering from the University of Alberta, Edmonton, AB, Canada, in 1983 and 1986, respectively.

He is currently a Professor with the School of Creative Media, City University of Hong Kong, Hong Kong, China, and the Department of Computer Science and Software Engineering, University of Melbourne, Victoria, Australia. He has taught computer architecture, computer networks, artificial intelligence, C programming language, machine learning, pattern recognition, and computer graphics. His research and development interests include machine intelligence, fuzzy-neural systems, visual communications, biometric systems and applications, and web/data-mining.

Leonard T. Bruton (M'71–SM'80–F'81) received the B.Sc. degree in electrical engineering from the University of London, London, U.K., in 1964, the M.Eng. degree in electrical engineering from Carleton University, Ottawa, ON, Canada, in 1967, and the Ph.D. degree in electrical engineering from the University of Newcastle Upon Tyne, Newcastle Upon Tyne, U.K, in 1970.

Currently, he is a Professor of Electrical and Computer Engineering at the University of Calgary, AB, Canada. He has worked in the Canadian telecommunications industry and as a teacher, researcher, and administrator in Canadian universities. His research interests include digital and analog filters, multimedia signal processing, multidimensional signal processing, and audio signal processing.

Dr. Bruton is a Fellow of the Royal Society of Canada. He has received numerous international and national awards for his teaching, research, and leadership contributions, including the IEEE Circuits and Systems Society Golden Jubilee Medal, the IEEE Outstanding Engineer Award for Region 7, and the Manning Award for Innovation.

James C. Bezdek (M'80– SM'90–F'92) received the B.S.C.E. degree from the University of Nevada, Reno, in 1969 and the Ph.D. degree in applied math from Cornell University, Ithaca, NY, in 1973.

He is currently with the Department of Computer Science, University of West Florida, Pensacola. His interests include woodworking, fuzzy mathematics, cigars, optimization, motorcycles, pattern recognition, gardening, image processing, fishing, computational neural networks, blues music, and medical applications.

James M. Keller (M'79–SM'92–F'00) received the Ph.D. degree in mathematics from the University of Missouri, Columbia, in 1978.

He has had faculty appointments in the Bioengineering/Advanced Automation Program, the Electrical and Computer Engineering Department, and the Computer Engineering and Computer Science Department at the University of Missouri, Columbia, where he currently holds the rank of Professor. He is also the R. L. Tatum Research Professor in the College of Engineering. His research interests include computer vision, pattern recognition, fuzzy set theory and fuzzy logic, fractal geometry, and neural networks. He has coauthored over 175 technical publications. He is an Associate Editor of the *International Journal of Approximate Reasoning*, and is on the editorial board of *Pattern Analysis and Applications*, *Fuzzy Sets and Systems*, and the *Journal of Intelligent and Fuzzy Systems*.

Dr. Keller has presented live and video tutorials on fuzzy logic in computer vision for the IEEE, is a national lecturer for the Association for Computing Machinery (ACM), is an IEEE Neural Networks Council Distinguished Lecturer, and is a past President of the North American Fuzzy Information Processing Society (NAFIPS). He is the Editor-in-Chief of the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He was the Conference Chair of the 1991 NAFIPS workshop, Program Co-Chair of the 1996 NAFIPS meeting, Program Co-Chair of the 1997 IEEE International Conference on Neural Networks, and the Program Chair of the 1998 IEEE International Conference on Fuzzy Systems.

Sandy Dance (M'95) was born in 1948 in Melbourne, Australia. He received the B.Sc. and M.Sc. degrees in mathematics from Monash University, Clayton, Australia, in 1969 and 1972, respectively, and the Ph.D. degree from the University of Melbourne, Victoria, Australia, in 1995.

For a number of years, he worked in the commercial computer industry before returning to study for his Ph.D. degree. After two years as a Post-Doctoral Fellow, he moved on to the Australian Bureau of Meteorology Research Centre, Melbourne, where he has been working with weather radar. His interests lie in the areas of high-level interpretation of images, artificial intelligence, and agent networks.

Norman R. Bartley (S'75–M'78) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Calgary, AB, Canada, in 1976 and 1978, respectively.

From 1979 to 1998, he was with the University of Calgary as a Research Associate in the Department of Electrical and Computer Engineering. He is now an Instructor in the same department. His main fields of interest include multidimensional systems, image processing, and digital signal processing systems and architectures.

Cishen Zhang (S'87–M'89) received the B.Eng. degree from Tsinghua University, Beijing, China, in 1982 and the Ph.D. degree in electrical engineering from Newcastle University, Callaghan, Australia, in 1990.

From 1971 to 1978, he was an Electrician with Changxindian (February Seven) Locomotive Manufactory, Beijing, China. From 1983 to 1985, he carried out research work on control systems at Delft University of Technology, Delft, The Netherlands. Since 1990, he has been with the Department of Electrical and Electronic Engineering, University of Melbourne, Victoria, Australia, and is currently an Associate Professor and Reader and the Deputy Head of the department. His research interests include time-varying systems and adaptive systems for signal processing and control.