CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

A Machine Vision Based Surveillance System for California Roads

J. Malik,
S. Russell

California PATH Research Report
UCB-ITS-PRR-95-6

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Report for MOU 83

March 1995

ISSN 1055-1425

A Machine Vision Based Surveillance System for California Roads
PATH project MOU-83 Final Report

Principal Investigators: J. Malik and S. Russell
Other authors: J. Weber, T. Huang and D. Koller
Computer Science Division
University of California

Acknowledgments

We gratefully acknowledge the help of C. McCarley and his group at Cal Poly, San Luis Obispo, for providing us with video tapes of various traffic scenes.

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

Abstract

Automatic symbolic traffic scene analysis is essential to many areas of IVHS (Intelligent Vehicle Highway Systems). Traffic scene information can be used to optimize traffic flow during busy periods, identify stalled vehicles and accidents, and aid the decision-making of an autonomous vehicle controller. Improvements in technologies for machine vision-based surveillance and high-level symbolic reasoning have enabled us to develop a system for detailed, reliable traffic scene analysis. The machine vision component of our system employs a correlation-based tracker and a physical motion model using a Kalman filter to extract vehicle trajectories over a sequence of traffic scene images. The symbolic reasoning component uses a dynamic belief network to make inferences about traffic events such as vehicle lane changes and stalls. In this paper, we discuss the key tasks of the vision and reasoning components as well as their integration into a working prototype. Preliminary results of an implementation on special purpose hardware using C-40 Digital Signal Processors show that near real-time performance can be achieved without further improvements.

keywords: Video Image Processing, Traffic Surveillance, PATH, Incident Detection, Computer Vision, Automatic Vehicle Classification, Artificial Intelligence, Advanced Traffic Management Systems.

Executive Summary

In this paper we have described the successful combination of a low-level, vision-based surveillance system ([KWM94]) with a high-level, symbolic reasoner based on dynamic belief networks ([HOR93]). This prototype system provides robust, high-level information about traffic scenes, such as lane changes, stalled vehicles, and overall vehicle counts. We believe that the required accuracy can in the long run only be obtained using high-level reasoning under uncertainty.

A vision based system such as the one described here would provide one of a series of data collection modules for future ITS systems. Other data collection modules would include existing and future loop detectors, as well as other technologies such as possibly probe vehicles and light trip wire systems. Data from these modules would be collected and analyzed at a central Traffic Management Center (TMC). Some of the major uses of such a data collection system would be

- Fast incident detection without human monitoring of multiple video signals.
- Estimation of travel times between various points. This could be used in conjunction with variable message signs for flow control.
- Lane usage and heavy vehicle counts.
- Detailed traffic condition information for public use.

A vision based module provides unique and more detailed data than other methodologies. In addition, the artificial intelligence component reduces the data bandwidth necessary between the system and the TMC. The system need only send visual information when adverse conditions are detected, as opposed to continuously. If multiple vision systems are to be used this bandwidth reduction will be required.

This report also outlines the results of testing the system under a variety of conditions. Testing of the vision system on a range of lighting conditions has brought out one of the shortcomings of the system. This is the effect that long shadows have on the tracking. Long shadows are often detected as moving objects by this method. As a result, separate vehicles may be tracked as a single, large vehicle if the shadow of one vehicle extends to the other vehicle. Other vision-based surveillance systems have come upon this same problem and simple heuristics have been developed to deal with it [Kil92]. Future work will be looking into possible solutions.

Contents

1 Introduction

An important task for progress in IVHS (Intelligent Vehicle Highway Systems) is the development of methods for automatic traffic scene analysis. All three major applications of IVHS – ATIS (Advanced Traveler Information Systems), ATMS (Advanced Traffic Management Systems), and AVCS (Automated Vehicle Control Systems) – could benefit from accurate, high-level descriptions of traffic situations. For example, an ATIS and an ATMS could use information about traffic congestion and stalls to warn drivers or to direct vehicles to alternate routes. An ATMS also could analyze local traffic at intersections to identify those with higher risk of accidents. Finally, an AVCS would need information about the actions of neighboring vehicles and the condition of traffic lanes ahead to control an automated car moving along a freeway [NS91].

In this paper, we describe a prototype system in which we have successfully combined a robust, vision-based traffic surveillance system [KWM94] with a dynamic belief network dedicated to analyzing traffic scenes. Unlike conventional loop detectors, which are buried underneath highways to count vehicles, video monitoring systems are less disruptive and less costly to install. They also have greater range and allow for more detailed descriptions of traffic situations. Dynamic belief networks provide a flexible, theoretically sound framework for traffic scene analysis because they can easily model uncertainty and because they can provide high-level, symbolic descriptions by integrating low-level information from a variety of sources. They also provide a natural framework for expressing knowledge about typical traffic behavior, allowing more accurate analyses from a given sensor stream.

Symbolic traffic scene analysis using vision-based surveillance systems has been previously investigated by several research groups [SBSZ87, KHH91, HKN91, HOR93]. The challenges of this approach include identifying vehicles despite imprecise video data and changing lighting conditions, tracking individual vehicles despite their overlapping with each other, and efficiently providing high-level descriptions based on evidence accumulated over time. We have achieved improvements in performance, reliability, and accuracy by applying a new approach for detecting and tracking vehicles, by explicitly reasoning about vehicle occlusions [KWM94], and by devising techniques for fast belief network update, localized reasoning, and flexible node semantics.

2 Low-Level Machine Vision-Based Surveillance

Our traffic surveillance system is based on the block diagram shown in figure 1. This section focuses on the tasks of feature extraction and tracking, and the next section focuses on the tasks of symbolic reasoning and incident detection.

As figure 1 indicates, traffic scene analysis generally proceeds from low-level processing of road traffic images to high-level descriptions of the traffic situation (which can in turn be used to direct and disambiguate low-level processing). Given a sequence of traffic images, a vision-based surveillance system must identify the vehicles in the scene and track them as they progress along the image sequence. This requires not only estimation of the moving vehicle shapes and positions, but also association of these estimates from one image to the next.

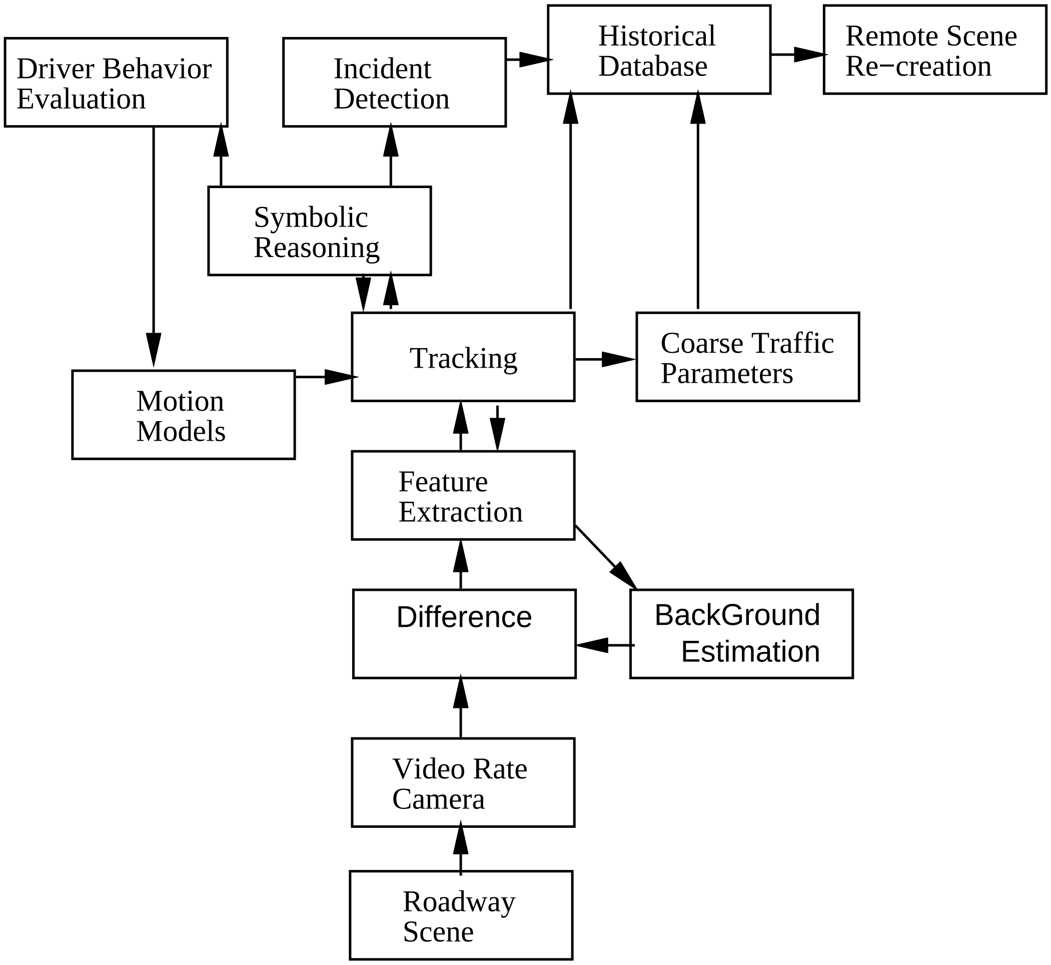


Figure 1: Block diagram of the complete traffic surveillance system. Arrows denote the flow of information.

Two primary factors that complicate this task are noisy sensors, which yield imprecise measurements, and vehicle occlusions, which make it more difficult to identify and disambiguate vehicles.

To address the problem of noisy measurements, we employ vehicle motion models that are updated in a Kalman filter[Gel74] formalism, thus yielding most likely estimates based on accumulated observations. The tracking is performed in a *world* coordinate system. This is accomplished by projecting the points on the image onto the road plane. Since the road can be assumed flat for the range of the image, this transformation only requires a simple linear transformation in homogeneous coordinates. The advantage of tracking in world coordinates is that physical constraints of a vehicles motion model can be used to guide tracking. For example, the knowledge that vehicles have finite acceleration will limit the range of motion a vehicle can have in the image from frame to frame.

Also, by exploiting a priori knowledge of the scene geometry, we can compute a depth ordering of the image regions associated with the moving objects. The knowledge of the vehicles position in world coordinates tells us which vehicles can occlude each other. This allows us to reason explicitly about occlusion and improve the identification of overlapping vehicles.

2.1 Motion Segmentation

A surveillance system initiates vehicle identification and tracking by determining what parts of each image belong to moving objects and what parts belong to the background. This is accomplished by examining the difference in pixel intensities between each new frame and an estimate of the stationary background. Reliable background estimation, which is critical for accurate identification of moving "blobs", is made more difficult as lighting conditions change. We perform this *initialization* step by using a modified version of the moving object segmentation method suggested by [KvB90] and implemented by [Kil92]. Our method employs a Kalman filter-based adaptive background model. This allows the background estimate to evolve as the weather and time of day affect lighting conditions. The background is updated at each frame using the following update equation:

$$B_{t+1} = B_t + (\alpha_1(1 - M_t) + \alpha_2 M_t)D_t \tag{1}$$

B_t is the background model at time t, D_t is the difference between the present frame and the background model, and M_t is a binary mask of hypothesized moving objects in the current frame. The gains α_1 and α_2 are based on estimates of the rate of change of the background. For a complete description, we refer the reader to [KWM93].

A block diagram for the low-level vision and tracking components is shown in figure 2. The track initiation and occlusion reasoning are performed in image coordinates while the tracking is performed in real world coordinates. The two coordinate systems are related by a simple transformation.

2.2 Vehicle Identification and Shape Estimation

After identifying moving blobs, the vision system attempts to disambiguate individual vehicles and estimate their shapes. This helps with associating data over a sequence of images and with obtaining accurate vehicle trajectories. Our system performs these tasks by developing a correlation mask over time. This mask conforms to the estimated appearance of the vehicle in the image. The shape of the mask could also be used to identify the size of the vehicle and estimate if it is a truck or passenger vehicle.

2.3 Motion Estimation

The final task of the video system is to track identified vehicles from one frame to the next. To accomplish this, we track the state of the vehicle using coordinates based on the road plane. This state is tracked using a standard linear Kalman Filter. The state of a vehicle,

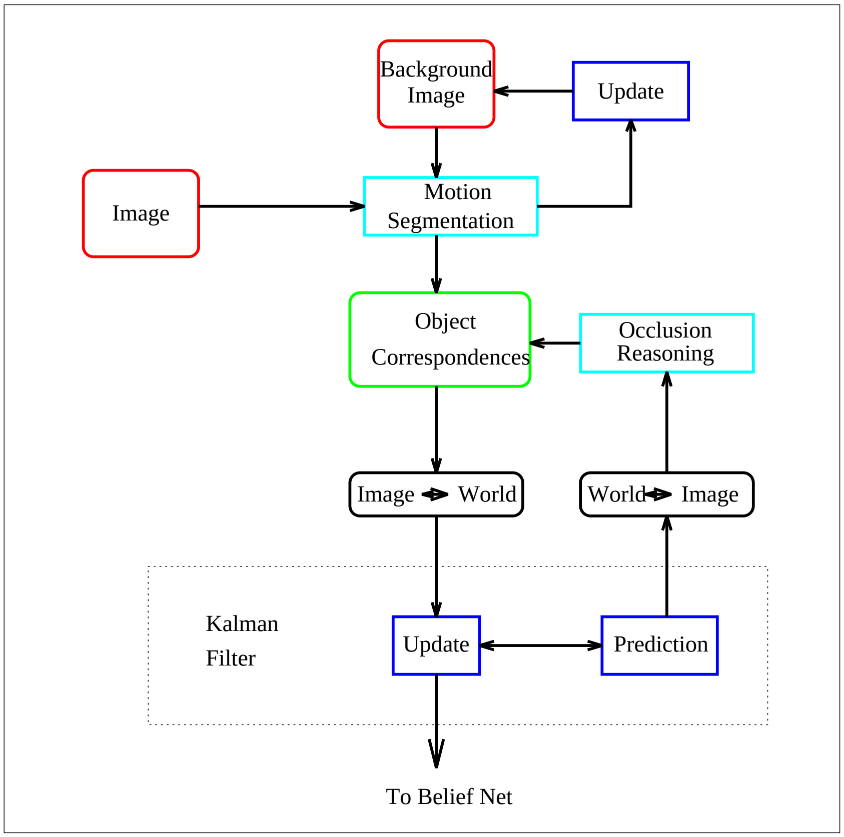


Figure 2: Block diagram of the vision and tracking components of the traffic surveillance system. Arrows denote the flow of information.

$\vec{X}_t$, consists of its position and velocity, $\vec{X}_t = (x, y, \dot{x}, \dot{y})$. The motion model for this state is

$$\vec{X}_{t+1} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \vec{X}_t + \vec{n}_t \tag{2}$$

Accelerations of the vehicle are contained in the vector, $\vec{n}_t$. Under this model, the acceleration vector can change with every frame.

At each time frame we measure the position of the center of the vehicle in the image. This position is translated into world coordinates and used as our state measurement, $\vec{Z}_t$. The relationship between the measurement and the state is simply

$$\vec{Z}_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \vec{X}_t + \vec{v}_t \tag{3}$$

where $\vec{v}_t$ is the measurement noise. The measurement noise is found by taking the known measurement variance in image coordinates and transforming it into world coordinates. In this way we can use the fact that as a vehicle becomes more distant, its apparent size becomes smaller and the uncertainty in its position increases. This fact is often not used in systems which track purely in image coordinates.

2.4 Occlusion Reasoning

Because vehicles often overlap with each other in the road images, the extracted contours of vehicles will become distorted for some frames. This can cause artificial shifts in vehicle trajectories, since tracks are obtained by connecting centers of contours along the image sequence. To avoid these artificial shifts and to obtain reasonable tracks, we employ an explicit occlusion reasoning algorithm, which compensates for overlapping vehicles.

The occlusion reasoning algorithm works because the traffic scene geometry is known and because motion is assumed to be constrained to the ground plane [KWM93]. This knowledge makes it possible to determine a depth ordering among the objects in the scene, and this depth ordering defines the order in which objects are able to occlude each other.

Figure 3 shows two object contours overlapping. The estimated contour of the occluded object, object 2, contains an segment that is occluded by object 1. Since we have an estimate of the objects shape as well as its position and velocity, we can continue to estimate (albeit with growing uncertainty) the occluded part of the contour.

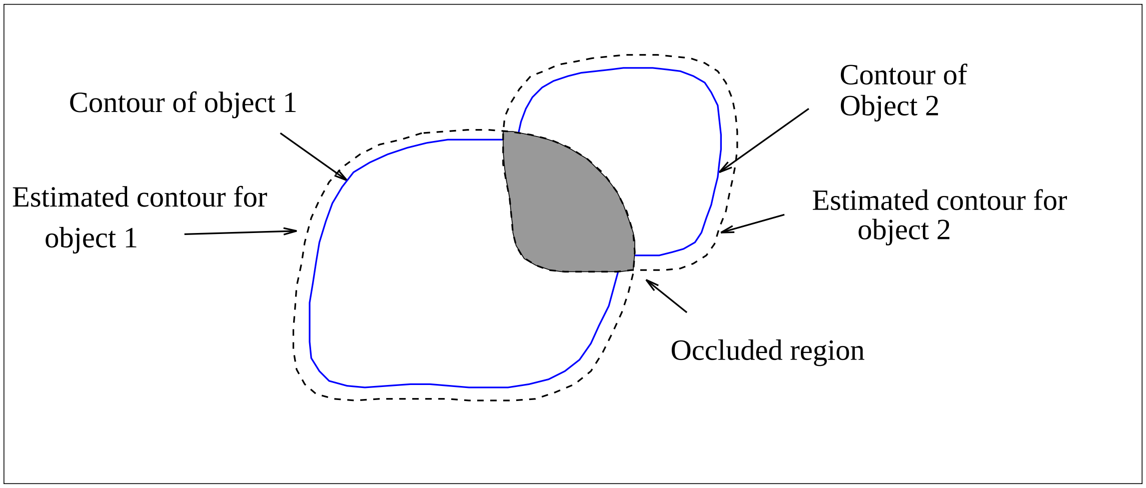

Contour of object 1
Contour of Object 2
Estimated contour for object 1
Estimated contour for object 2
Occluded region

Figure 3: Vehicles can occlude each other in the image. Since the positions of each vehicle relative to the camera are known, the occluded vehicle can be identified. The tracking algorithm maintains an estimate of the occluded vehicle's shape while it is occluded.

3 High-Level Reasoning Using Belief Networks

We now address the task of using vehicle track information (e.g., their positions and velocities) to arrive at high-level symbolic descriptions of vehicles and the traffic scene. To accomplish this, our symbolic reasoner uses multiple, per-vehicle dynamic belief networks with fast roll-up.

3.1 Concepts

Belief networks are directed acyclic graphs in which nodes represent random variables (usually discrete) and arcs represent causal connections among the variables [Pea88]. Associated with each node is a probability table that provides conditional probabilities of the node's possible states given each possible state of its parents. When values are observed for a subset of the nodes, posterior probability distributions can be computed for any of the remaining nodes. This updating takes place using a compiled form of the belief network that is more suitable to propagating the influence of evidence to other nodes.

Belief networks offer a mathematically sound basis for making inferences under uncertainty. The conditional probability tables provide a natural way to represent uncertain events, and the semantics of the updated probabilities are well-defined. Knowledge of causal relationships among variables is expressed by the presence or absence of arcs between them. Furthermore, the conditional independence relationships implied by the topology of the network allow exponentially fewer probabilities to be specified than the full joint probability distribution for all the variables in the network.

Dynamic belief networks allow for reasoning in domains where variables take on different values over time. Typically, observations are taken at regular "time slices", and a given network structure is replicated for each slice. Nodes can be connected not only to other nodes within the same time slice but also to nodes in the previous or subsequent slice. As new slices are added to the network, older slices are removed. Before a slice is removed, its influence is "rolled-up" into the next slice by recomputing probability tables for certain nodes in that slice. Thus, evidence accumulated over time is always integrated into the current belief network model [Nic92, Kja93].

3.2 Traffic network structure

The symbolic reasoning component for our system is built on the HUGIN inference engine for belief networks [AOJJ89]. Figure 4 shows an example belief network fragment for a single vehicle. Figure 5 shows the fragment projected over one time slice. For each vehicle in a traffic scene, there is a separate belief network corresponding to it.

Some of the nodes in figure 4, such as Xpos_sens.t and Xdot_sens.t, correspond to discretized sensor values that are set in each new slice when the slice is added to the network. For instance, the Xpos_sens.t node represents a vehicle's left-right position among the lanes of a highway and can take on one of ten states indicating the vehicle's distance from the right edge of the lanes. Other nodes, such as Stalled.t and Lane_Change.t, correspond to high-level events. For example, the Lane_Change.t node can take on one of three different states indicating if a vehicle is going straight, changing lanes to the left, or changing lanes to the right. The posterior probability distributions for these high-level events are affected by the sensor values in the current slice as well as the posterior probabilities of nodes in the previous slice. These distributions are then used to provide symbolic descriptions of the traffic scene.

Figure 5 shows how nodes are replicated from time slice 0 to time slice 1, as well as how some variables in time slice 1 depend on variables in the previous time slice. For example, Ypos.t1 (representing a vehicle's forward position on the highway) depends on Ypos.t0 (its previous position) and Ydot.t0 (its previous velocity).

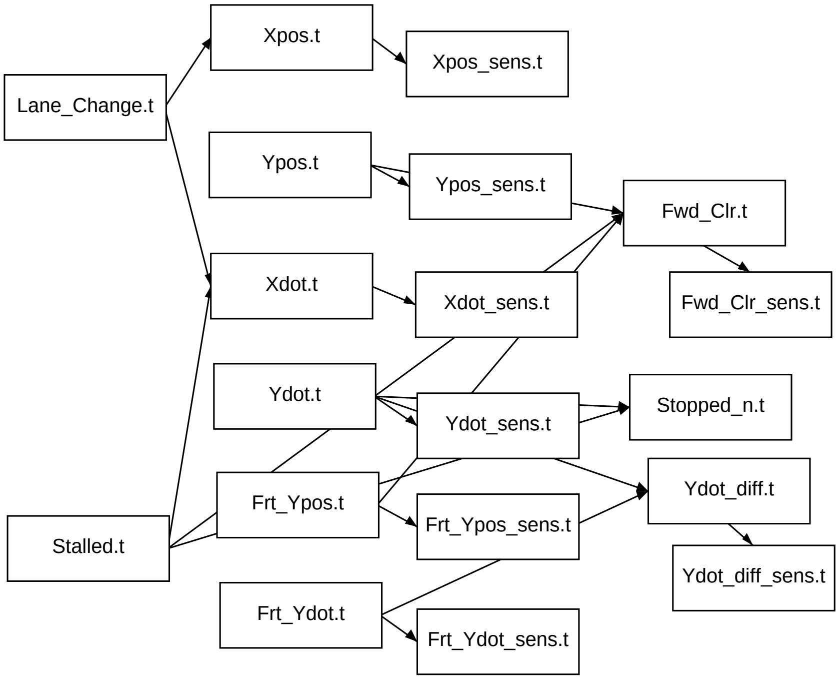


Figure 4: Belief network fragment for a single vehicle.

The probabilities associated with each node provide a natural framework to encode knowledge about traffic behavior and rules. For example, the probability table for Ydot.t1 in figure 5 contains probabilities for each of Ydot.t1's possible states (e.g., 21-30 km/hr, 31-40 km/hr, etc.) given the states of Ydot.t0, Fwd_Clr.t0 (the space in front of a vehicle), and Ydot_diff.t0 (the difference in speed between a given vehicle and the vehicle in front of it). A driver is likely to slow down if there isn't much distance between his vehicle and the vehicle in front and if his vehicle is going faster than the vehicle in front. Thus, the appropriate entries in the probability table will indicate a high probability that the vehicle's speed at time t1 will be lower than its speed at time t0. Similarly, the other entries in the table encode probability distributions for the new velocity given the combinations of parent states. Additional traffic knowledge that is or will be encoded includes knowledge about lane-changing and braking behavior, the effect of road geometry and weather on driving behavior, and the significance of brake, hazard, and signal lights.

3.3 Network Issues

Handling multiple vehicles. As mentioned earlier, in each time slice the structure in figure 4 is replicated for each tracked vehicle in the traffic scene. Clearly, the positions and velocities of different vehicles will affect each other. Thus, determining globally consistent probability distributions for each vehicle involves a large network consisting of changing interconnections between vehicle subnetworks.

Our current approach is to assign each vehicle its own dynamic belief network. We incorporate the influence of nearby vehicles on the current vehicle by assigning some nodes

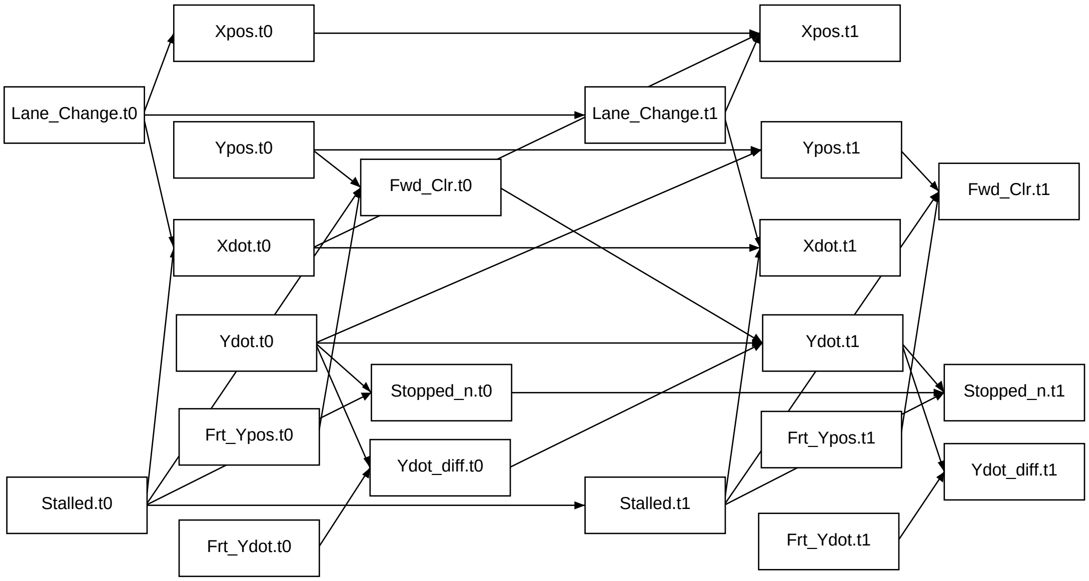


Figure 5: Belief network fragment for a single vehicle projected over one time slice. Sensor nodes have been omitted for simplicity.

to those vehicles. For example, Front_Ypos.t and Front_Ydot.t in figure 4 refer to "the vehicle in front of the current vehicle". Since the actual vehicle in front may change, these "indexical" [Pea88] nodes do not correspond to a specific vehicle. Instead, a preprocessing step uses sensor data to determine which vehicles are currently in front of each other and then sets those node states accordingly. Using multiple, per-vehicle belief networks with indexical nodes has yielded a reasonably inexpensive approach to achieving locally consistent high-level descriptions for each vehicle while considering the affect of nearby vehicles.

Nodes with variable semantics. When a vehicle first stops on the highway, it could be for any number of reasons. The probability that the vehicle is stalled may be small at first, but it increases over time if the vehicle continues to remain stopped while no vehicles are stopped in front of it. To allow flexible representation of how a vehicle being stalled relates to it being stopped for some time, we made it possible for nodes to have variable semantics, i.e. a node refers to a different event in different time slices. This is accomplished by modifying the node's conditional probability table from one time slice to another. For example, the Stopped_n.t node has some value n associated with it, and the node refers to the event that the vehicle has been stopped for n time slices. The probability table for the node is modified according to the value of n associated with it. This can be computed with a simple function to simulate a counter (e.g., we can give vehicles positive probability of being stalled only if they've been stopped for over 50 time slices) or with any arbitrarily complex function.

Rolling the network forward. Because the HUGIN system is geared toward standard rather than dynamic belief networks, we developed the facilities necessary for rolling the network forward. Essentially, this involves adding the capability to add new time slices to the network and to incorporate information from old slices to the rest of the network so that the old slices can be deleted.

We developed two approaches to this problem. In the first approach, we generated and compiled a new network for each time slice, and we used a new network every time a slice was added. This approach seemed adequate and offered the opportunity to dynamically alter the actual network structure (which would be necessary for a global network of all the vehicles), but it suffered from the poor performance noted earlier.

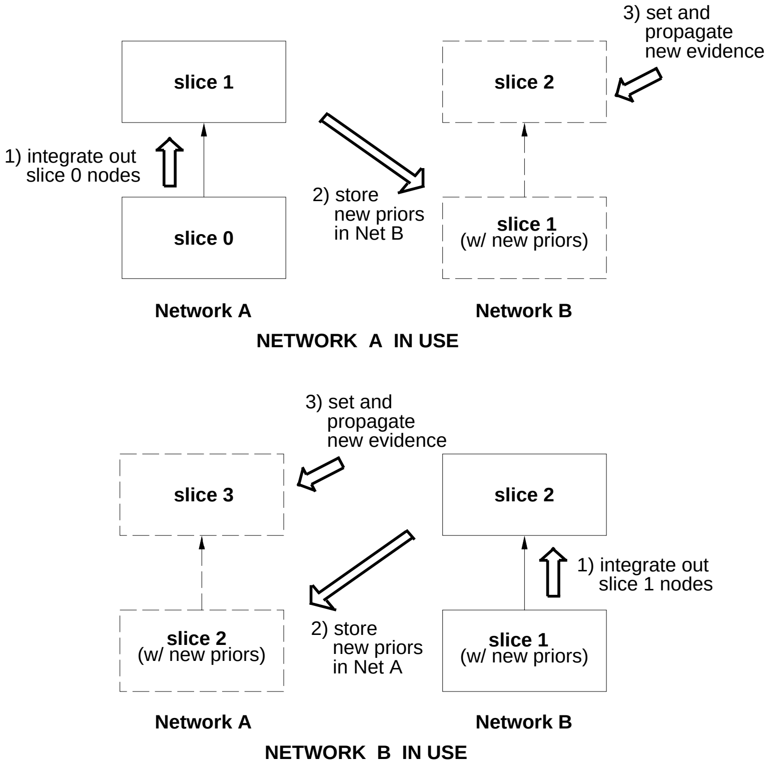

3) set and propagate new evidence
slice 1
slice 2
1) integrate out slice 0 nodes
2) store new priors in Net B
slice 0
slice 1 (w/ new priors)
Network A
Network B
NETWORK A IN USE
3) set and propagate new evidence
slice 3
slice 2
1) integrate out slice 1 nodes
slice 2 (w/ new priors)
2) store new priors in Net A
slice 1 (w/ new priors)
Network A
Network B
NETWORK B IN USE

Figure 6: Steps for rolling the dynamic network forward.

We currently use our second approach, which employs two precompiled networks, each with two slices. As shown in figure 6, the system alternates between the two networks. To introduce sensor information from a new time slice, the system incorporates the evidence from the oldest slice into the rest of the model through a series of straightforward matrix multiplications. The resulting probability tables are stored in the first slice of the other network. The new sensor information is then added to the second slice of this network, and their influence is propagated to obtain new posterior probabilities. This other network is then used until the next time slice, when the roll-up procedure is repeated back to the first network. This approach does not allow dynamic alteration of the belief network structure, but it greatly improves performance by eliminating the need for network re-compilation after every time slice.

The dHUGIN package [Kja93] provides extensions to HUGIN for dynamic belief networks, but it does not provide the flexibility of our first approach for changing the network structure from one time slice to the next, and it does not provide the performance speedup of our second approach.

By explicitly modeling uncertainty, the interaction of low-level information from a variety of sources, and the effect of evidence accumulated over time, dynamic belief networks provide a flexible, theoretically sound framework for temporal high-level reasoning about traffic scenes. We have incorporated enhancements that improve the performance of belief network evaluation, that reduce the complexity of evaluation to be linear in the number of vehicles tracked, and that provide greater robustness by varying the semantics of network nodes from one time slice to another. These enhancements are described in more detail in [HKM+94].

4 Results with Real-World Traffic Scenes

We have tested our system on both real-world image sequences and syntheticly generated sequences. The synthetic sequences were used for simulating stalls and accidents (which can not be planned for the filming schedule). The real-world sequences included sequences from 2, 4 and 6 lane highways. The time of day varied from morning, mid-day and late afternoon. One sequence was taken during a light rain.

In figure 7 we present the results of one 270-frame sequence of a divided four-lane freeway. The image at the top of figure 7 shows frame #40 of the sequence overlaid with contour estimates of the vehicles. The image at the bottom shows only the vehicle contour estimates and their tracks (starting from frame #0).

Figure 8 and 9 show some of the information being tracked by the inference network for a different sequence on a divided 3 lane highway. The top image of figure 8 shows a single frame of the tracking sequence. Displayed on the top of the image is the speed, estimated lane-change direction and engine status (green, yellow or red) for the vehicle with the white square around it. The engine status would go to red if the network believed the vehicle had stalled and was affecting traffic flow. The lower image of figure 8 shows vehicle tracks overlayed from 10 seconds of video. On the top of the image are displayed the average speed and lane densities for each lane. Figure 9 shows the positions of the tracked vehicles in world image coordinates as if viewed from directly above the freeway. Many vehicles were tracked for more than 100 meters. Also notice the slight curvature of the road which is evident in both figure 9 and figure 8.

5 Towards Real-time Performance

The current implementation of the tracking and belief net systems performs its analysis in batch mode. Video sequences are digitized and stored on disk. The algorithm retrieves the frames at rates less than realtime. Running on a Sun SparcStation 10, the performance of the vision component reaches about 7 Hz for simultaneous tracking of about 8-10 vehicles (without I/O).

We have examined the use of multiple, high-speed processors to implement the system in real-time, thus allowing deployment of the system in a real world setting. The vision and tracking components were implemented on high-speed special purpose hardware using

Texas Instruments TMS320C40 digital signal processors. Our current system contains 7 processors with special hardware for video digitization and graphical display. The system currently tracks vehicles at a sustained rate of 6 Hz. This rate is too slow to track the vehicles accurately. But since the algorithm has been made fully parallel, the addition of more processors will speed the algorithm. Also, we expect processor speeds to double within the time frame of this project being developed into a commercial product. With these considerations a realtime implementation is possible.

The performance of the belief network varies greatly with the network design, but generally requires about 100 milliseconds per vehicle per frame. We expect to improve the performance by an order of magnitude with various optimizations. Presently the network has not been ported to the parallel network. This is not necessary since the heterogeneous system allows for the network to run on the host computer while communicating to the tracker code on the parallel array.

6 Conclusions / Future Work

In this paper we have described the successful combination of a low-level, vision-based surveillance system ([KWM94]) with a high-level, symbolic reasoner based on dynamic belief networks ([HOR93]). This prototype system provides robust, high-level information about traffic scenes, such as lane changes, stalled vehicles, and overall vehicle counts. We believe that the required accuracy can in the long run only be obtained using high-level reasoning under uncertainty.

The symbolic reasoner is already capable of using other vehicle features, such as vehicle type, turn signals and brake lights, to improve its analytical performance. The vision system needs to be upgraded to detect these features, as well as to handle vehicle shadows. Furthermore, the inferences of the symbolic reasoner can be fed back to the tracker's Kalman filter to further increase its reliability. For example, if a vehicle is signaling left, its expected motion update should be biased toward leftward acceleration rather than a random perturbation. This allows for reduced variance, and hence greater reliability in tracking. In the extreme case, if the low-level tracker loses a vehicle (for example, in heavy rain), the high-level system can automatically "track" its most likely position by a combination of extended projection and inference from the behavior of other vehicles.

Another benefit is the robust data fusion provided by Bayesian inference. This is especially important at dusk or dawn, when the surveillance system will see both vehicle outlines and vehicle tail lights. Finally, by including a simple sensor failure model, the network can detect and diagnose sensor failure, while continuing to track vehicles using remaining sensor inputs [Nic92].

Testing on various lighting conditions has brought out one of the shortcomings of our system. This is the effect that long shadows have on the tracking. Our initialization procedure begins with the difference between the present frame and an image of the road surface without vehicles. Long shadows are often detected as moving objects by this method. As a result, separate vehicles may be tracked as a single, large vehicle if the shadow of one vehicle extends to the other vehicle. Other vision-based surveillance systems have come upon this same problem and simple heuristics have been developed to deal with it [Kil92]. We will be looking into both histogramming methods and the use of color information to help alleviate this problem.

Besides continuing to refine the network design and to optimize its performance, we are investigating methods for enabling the symbolic reasoner to handle mixed networks with both continuous and discrete variables [LW89]. This offers the opportunity for greater performance over purely discrete networks, and it seems reasonable, since sensor variables such as vehicle positions and velocities are adequately modeled as Gaussians. The symbolic reasoner can also be enhanced to provide other types of descriptions, such as driver behaviors. Machine learning techniques applied to a library of image sequences can be used to generate detailed probabilistic models of driver behavior, which are useful both in our own work and in analytical and simulation studies of highway designs.

We are continuing to increase the speed of the implementation on the parallel hardware. The clock speed of the C40 chip has increased 50% in the last 12 months and we plan on upgrading to the new clock rate. Also, the introduction of more processors will increase the processing rate.

To better assess the system's usefulness and accuracy, we plan to measure its performance on a more extensive collection of video sequences. This includes night and bad weather scenarios.

The most important idea we plan on pursuing is to track vehicles as they pass through multiple camera views. Since the tracking is performed in a coordinate system on the road surface, it is possible to "hand off" vehicles from camera to camera provided the distance between their views is not too large. The estimate of position and velocity in one camera can be used to estimate the vehicle's appearance in the next camera. Filming is presently under way to provide a test set of multiple view sequences.

References

[AOJJ89] S. Andersen, K. Olesen, F. V. Jensen, and F. Jensen. Hugin* – a shell for building bayesian belief universes for exp ert systems. In *Proc. of the 10th Int'l Joint Conference on Artificial Intel ligence*, 1989.

[Gel74] Arthur Gelb, editor. *Applied Optimal Estimation.* The MIT Press, Cambridge, MA., 1974.

[HKM+94] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J.Weber. Automatic symbolic traffic scene analysis using belief networks. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 966–972, Seattle, WA, July 31–Aug. 4, 1994.

[HKN91] N. Heinze, W. Krüger, and H.-H. Nagel. Berechnung von bewegungsverben zur beschreibung von aus bildfolgen gewonnenen trajektorien in straßenverkehrsszenen. *Informatik – Forschung und Entwicklung*, 6:51–61, 1991.

[HOR93] T. Huang, G. Ogasawara, and S. Russell. Symbolic traffic scene analysis using dynamic belief networks. In *AAAI Workshop on AI in IVHS*, Washington D.C., 1993.

[KHH91] D. Koller, N. Heinze, and H.-H. Nagel. Algorithmic characterization of vehicle trajectories from image sequences by motion verbs. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 90–95, Lahaina, Maui, Hawaii, June 3-6, 1991.

[Kil92] Michael Kilger. A shadow handler in a video-based real-time traffic monitoring system. In *IEEE Workshop on Applications of Computer Vision*, pages 1060–1066, Palm Springs, CA, 1992.

[Kja93] Uffe Kjaerulff. User's guide to dhugin. Technical report, Institute of Electronic Systems, Aalborg University, 1993.

[KvB90] Klaus-Peter Karmann and Achim von Brandt. Moving object recognition using an adaptive background memory. In V Cappellini, editor, *Time-Varying Image Processing and Moving Object Recognition, 2*. Elsevier, Amsterdam, The Netherlands, 1990.

[KWM93] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. Technical Report UCB/CSD 93/780, Computer Science Division (EECS), University of California, Berkeley, 1993.

[KWM94] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proc. Third European Conference on Computer Vision*, pages 189–196, Stockholm, Sweden, May 2-6, 1994, J.-O. Eklundh (ed.),Lecture Notes in Computer Science **800-801**, Springer-Verlag, Berlin, Heidelberg, New York, 1994.

[LW89] S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17:31–57, 1989.

[Nic92] A. Nicholson. Monitoring discrete environments using dynamic belief networks. Phd thesis, Oxford University, 1992.

[NS91] A. Niehaus and R.F. Stengel. Rule-based guidance for vehicle highway driving in the presence o f uncertainty. In *Proceedings of the 1991 American Control Conference*, 1991.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[SBSZ87] J. R. J. Schirra, G. Bosch, C. K. Sung, and G. Zimmermann. From image sequences to natural language: A first step towards automatic perception and description of motion. *Applied Artificial Intelligence*, 1:287–307, 1987.

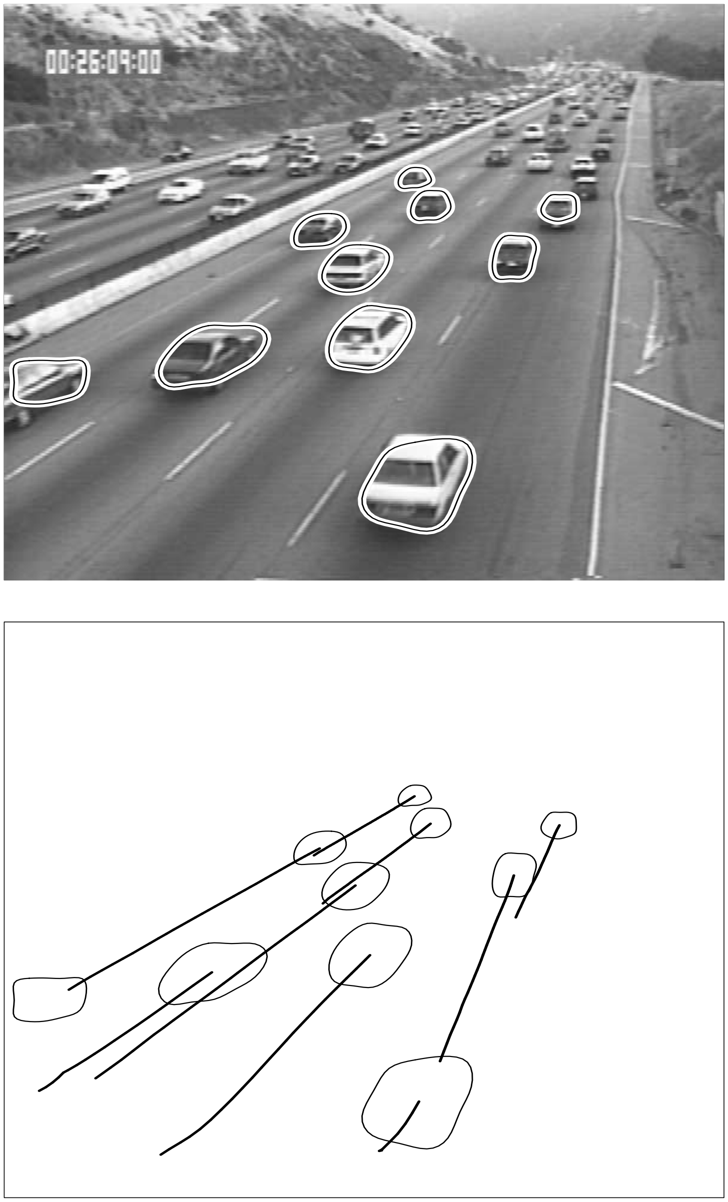
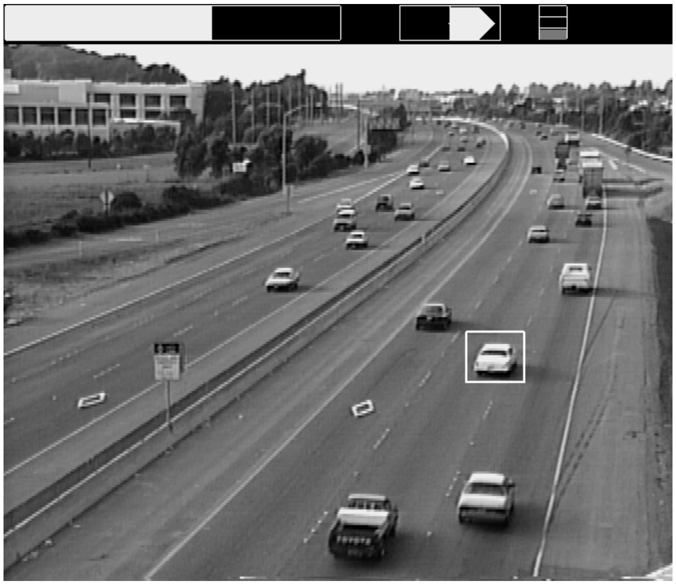


Figure 7: The upper image shows frame #40 of the image sequence with overlaid contour estimates of the cars. The bottom image shows the contour estimates with their tracks (starting from frame #0).



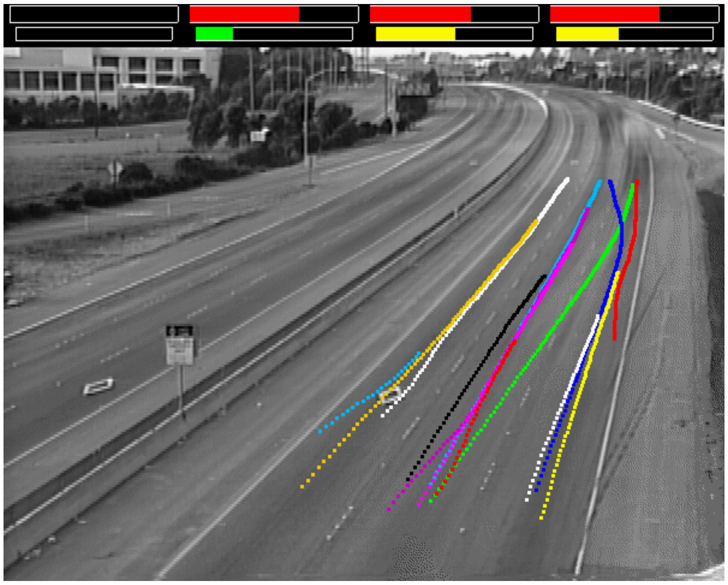


Figure 8: The information tracked by the inference network for a single vehicle is shown in the upper image. The network uses the estimated position and speed of each vehicle. Displayed on the top of the image are the speed, estimated lane-change direction and engine status (green, yellow or red, presently green) for the vehicle with the box around it. The lower image shows tracks overlayed from 10 seconds of video. On the top of the image are displayed the average speed and lane densities for each lane.

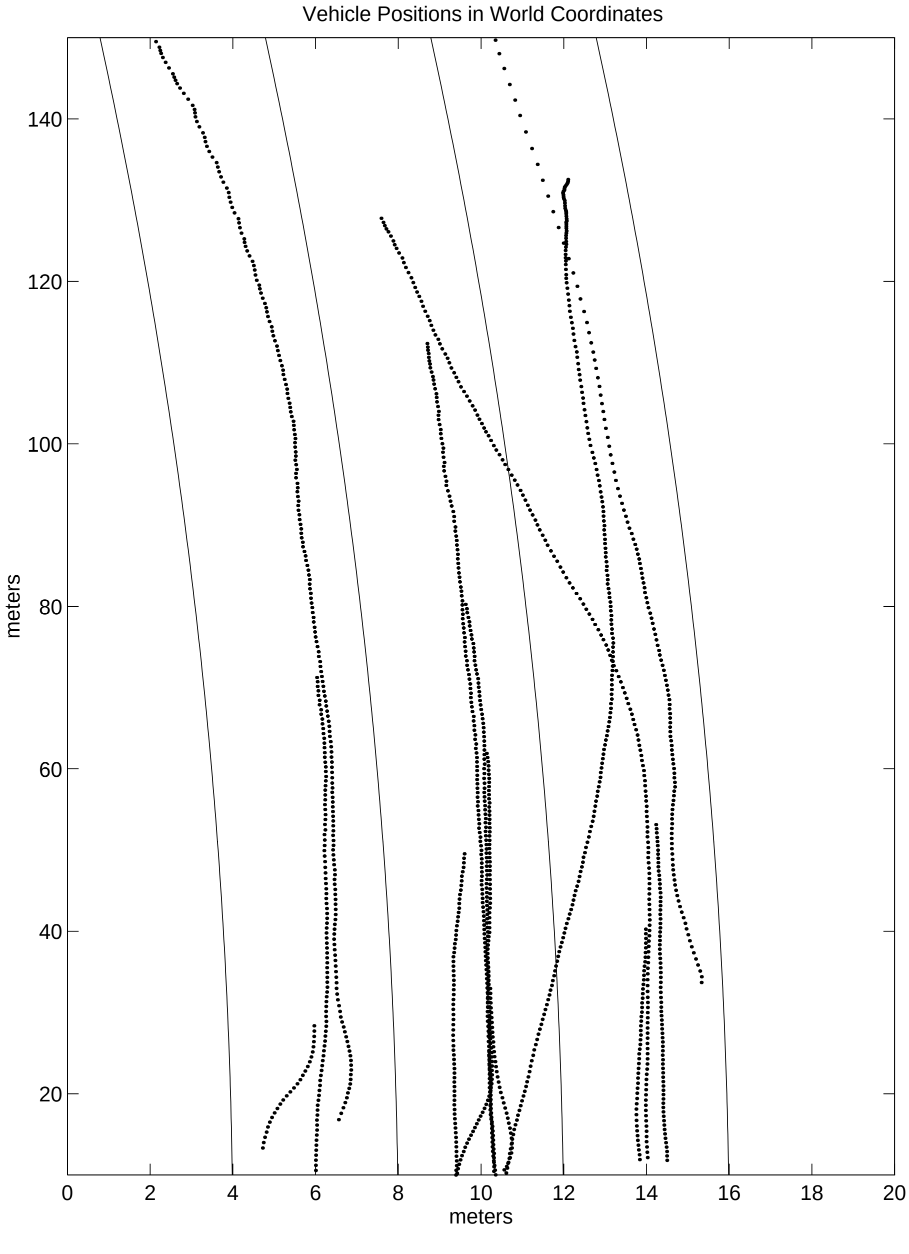


Figure 9: Vehicle positions plotted in world coordinates as if viewed from straight above the freeway. These results are from the same sequence as Figure 7. The freeway curves slightly to the left. Two lane changes were detected by the belief net for this sequence.