Proceedings of the 2004 ICRA Conference, New Orleans, LA, April 26 - May 1, 2004.

A Hierarchical, Multi-Resolutional Moving Object Prediction Approach for Autonomous On-Road Driving

C. Schlenoff, R. Madhavan, and T. Barbera
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230, U.S.A.
Tel: (301) 975-3456 Fax: (301) 990-9688
Email: raj.madhavan@ieee.org, {craig.schlenoff, tony.barbera}@nist.gov

***Abstract*— In this paper[1], we present a hierarchical multi-resolutional approach for moving object prediction via estimation-theoretic and situation-based probabilistic techniques. The results of the prediction are made available to a planner to allow it to make accurate plans in the presence of a dynamic environment. We have applied this approach to an on-road driving control hierarchy being developed as part of the DARPA Mobile Autonomous Robotic Systems (MARS) effort. Experimental results are shown in two simulation environments.**

I. Introduction

Over the past ten years, the National Institute of Standards and Technology (NIST) has been involved in developing control architecture and supporting software algorithms to enable autonomous navigation. We have used the 4D/RCS reference model architecture to serve as the underlying foundation for this effort [1], [2]. 4D/RCS is a hierarchical, distributed, real-time control system architecture that provides clear interfaces and roles for a variety of functional elements.

Under 4D/RCS, the functional elements of an intelligent system can be broadly considered to include: behavior generation (task decomposition and control), sensory processing (filtering, detection, recognition, grouping), world modeling (store and retrieve knowledge and predict future states), and value judgment (compute cost, benefit, importance, and uncertainty). These are supported by a knowledge database, and a communication system that interconnects the functional elements and the knowledge database. This collection of modules and their interconnections make up a generic node in the 4D/RCS reference model architecture [3]. A generic node (see Figure 1) is defined as a part of the 4D/RCS system that processes sensory information, computes values, maintains a world model, generates predictions, formulates plans, and executes tasks. Each module in the node may have an operator interface.

NIST's work in autonomous vehicle navigation has most recently been demonstrated in the DEMO-III eXperimental Unmanned Vehicle (XUV) effort. This effort seeks to develop and demonstrate new and evolving autonomous vehicle technology, emphasizing perception, navigation, intelligent system architecture, and planning [10] and represents the state of the art in autonomous driving. The most recent DEMO-III experiments have demonstrated autonomous mobility in highly static, wooded environments, where the only moving object of interest is the autonomous vehicle itself. Objects in the environment included rocks, trees, tall grass, paths, etc. This is a much different environment than what would be encountered during on-road driving, in which most of the objects in the environment are highly dynamic (e.g., pedestrians, vehicles, animals, debris, etc.).

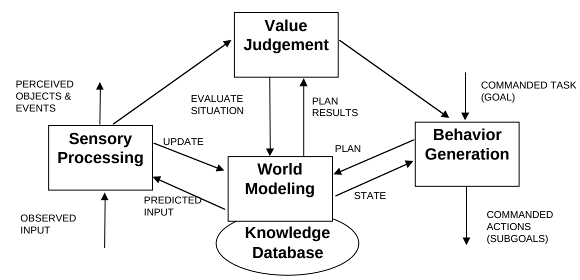


Fig. 1. A Real-time Control System (RCS) node.

Statistical methods for estimating obstacle locations using statistical features have been proposed by other researchers such as the Hidden Markov Models (HMMs) to predict obstacle motion [11], Poisson distribution to describe the probability of collision with obstacles [9], autoregressive models for one-step ahead prediction of moving obstacles [5] or probability of occupancy of cells in grid maps [8]. The principal disadvantages of these methods are that they are computationally intensive thus precluding real-time implementations and perhaps most importantly have only been implemented for 2D polygonal environments.

Nagel [6] has performed some research on moving object prediction during on-road driving based upon the concept of generally describable situations, fuzzy logic, and situation graph trees. However, based on the literature, Nagel has not tried to project out what the next actions of the moving object will be and has not assigned probabilities to those actions.

[1]Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Dickmanns [4] has performed some research on situation assessment and intention recognition for on-road driving.

The work described in this paper is different than other related efforts in that it introduces a novel way to perform moving object prediction based upon a multi-resolutional, hierarchical approach which incorporates multiple prediction algorithms into a single, unifying framework. The lower levels of the framework utilize estimation theoretic short-term predictions based upon an Extended Kalman Filter (EKF) while the upper levels utilize a probabilistic prediction approach based upon situation recognition with an underlying cost model.

The paper is organized as follows: In Section II, we describe our 4D/RCS-based architecture for on-road driving and describe the levels in the hierarchy that need to deal with moving objects in the environment. In Section III, we describe an estimation-theoretic approach that is employed in the lower levels of the control hierarchy. In Section IV, we describe the situation recognition and probabilistic approaches that are used in the higher level of the control hierarchy followed by the experimental results in Section V. Section VI provides the conclusions and future research efforts.

II. On-road Driving Hierarchy

The 4D/RCS design methodology attempts to define an information representational architecture for all of the task knowledge. This architecture is dependent on both the domain knowledge of the task and the requirements to decompose and structure the information in such a manner as to match the human techniques of information management by "chunking" and hierarchical decomposition. It does this by providing a layering of the task activities by levels of abstraction or detail and separating out at each level by subtask context, the task control knowledge that represents the next appropriate subtask action and its transitioning situational knowledge. It then attempts to describe what world model states have to exist for each transitioning situation to be true.

The overall approach is to analyze the driving tasks through a discussion of a large number of scenarios of particular on-road driving subtasks and to derive from these descriptions a task decomposition tree representation of all the task activities at various levels of abstraction and detail. From this task tree we can organize the activities into a more rigorous layering by the artifice of identifying an organizational structure of agent control modules that are responsible for executing the different levels of the task decisions. This use of separate executing agents organized into an execution hierarchy provides a mechanism to formalize the task decision tree by assigning certain decisions to particular agent control modules as seen in Figure 2. This creates well-defined sets of subtask commands from each agent control module to its next level subordinate agent control module, thus forcing us to group and label various sets of related activities of the driving task with a context identifier such as `GoOn` *roadname* `TurnLeftOn` *roadname*, `PassVehInFront`, `ChangeTo_LeftLane`, etc. Each of these identifiers is really a subtask goal command at different levels in the execution hierarchy.

The task decision rules appropriate to each of these subtask goal commands can be encoded within Finite State Machines (FSMs). These rules use context relative situations as their input conditions to cause the transitioning to the next output action to accomplish the goal command. As an example, the "PassVehInFront" FSM has as one of its input conditions the situational assessment that "conditions good to pass". This situation being true will cause this rule to fire and send the `ChangeTo_LeftLane` output command to its next lower subordinate control module in order to go around the vehicle in front. This situation is really a cost evaluated summation of the present situation identification and the predicted evolution of this into future situations describing the expected actions of all of the moving objects.

The evaluation of each FSM transitioning situation is framed by the architecture. As shown in Figure 2, the 4D/RCS partitions the task into various levels of resolution or abstraction. This means that the situations being analyzed at the higher levels are more abstract and look out much further in time to decide on more general responses. For example, the high level `RouteSegment Manager` might detect a slow moving truck in the right lane a quarter of a mile ahead and set the goal lane to the left lane of a four lane divided highway. The `DrivingBehavior` Control Module might detect a slower moving vehicle in front on a two lane undivided road and determine that no oncoming vehicles pose a collision risk, there are no pedestrians moving towards the road, the vehicle in front is expected to continue driving normally down the road, the vehicle behind is not expected to try to pass own vehicle - and from all of these situations and predicted situations evaluate the present situation as "conditions good to pass" thus evoking the `ChangeTo_LeftLane` command. The `ElementalManeuver` Control Module might make a short-term estimation-theoretic prediction on the next position and direction of a vehicle in close proximity to our goal path.

Thus, the 4D/RCS approach reduces the apparent complexity that has to be resolved by partitioning the problem up so that appropriate moving object prediction techniques can be brought to bear at the proper level of the task decision process. These moving object prediction techniques are elaborated on in the following sections.

III. Estimation-theoretic Short-term Prediction

Estimation-theoretic schemes using Kalman Filters (KFs) are well established recursive state estimation techniques where estimates the states of a system are computed using the process and observation models [7]. The recursive nature of the algorithm utilizes the system's CPU more uniformly to provide estimates without the latency resulting from batch processing techniques. The (linear) KF is simply a recursive estimation algorithm that provides minimum mean squared estimates (MMSE) of the states of a linear system utilizing knowledge about the process and measurement dynamics, process and measurement noise statistics subject to Gaussian assumptions and initial condition information. When these assumptions

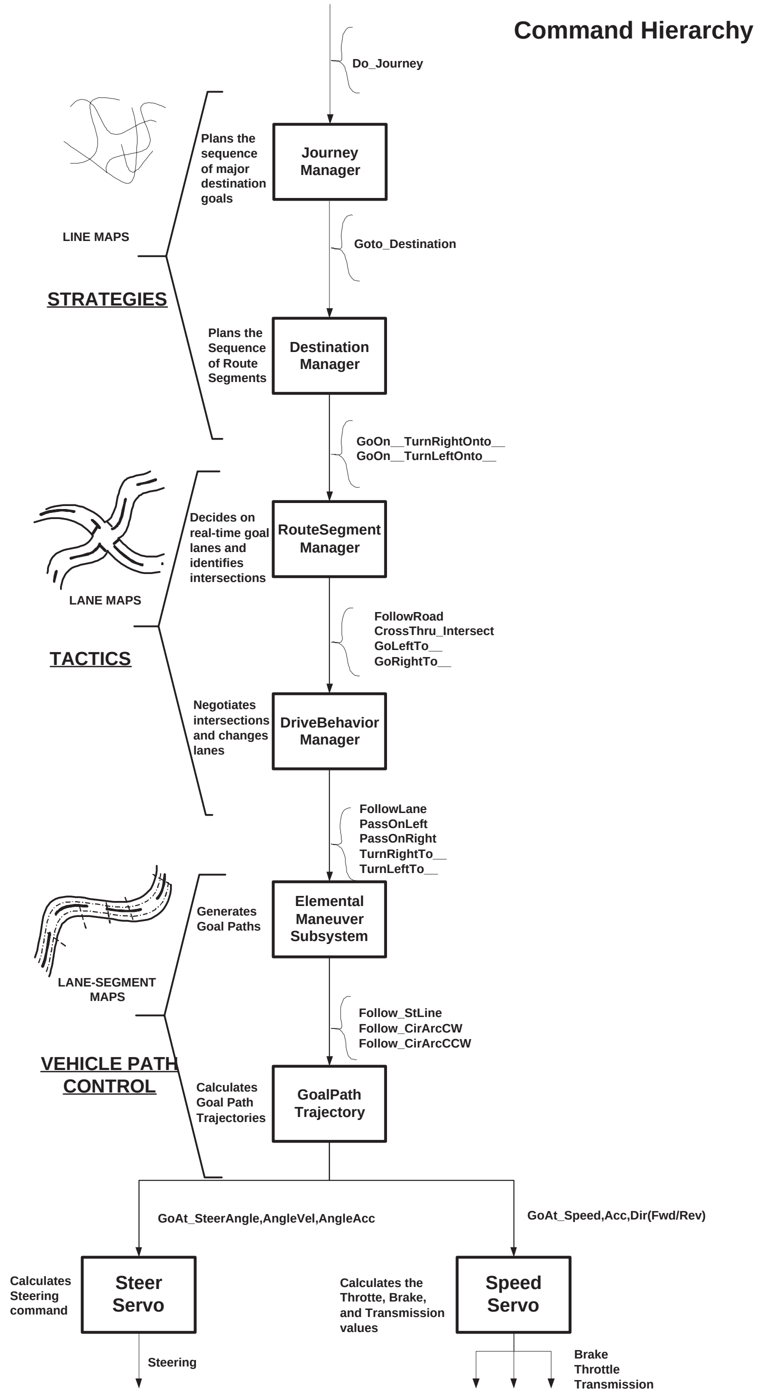


Fig. 2. 4D/RCS Multi-Resolutional Control Hierarchy for On-road Driving.

are satisfied, the estimates provided by the Kalman filter are *optimal*. The extension of the linear Kalman filtering ideas to a non-linear system is termed extended Kalman filtering.

The EKF is a linear estimator for a non-linear system obtained by *linearization* of the nonlinear state and observation equations. For any non-linear system, the EKF is the best linear unbiased estimator with respect to minimum mean squared error criteria. Within the on-road driving hierarchy, short-term prediction of objects moving at variable speeds and at given look-ahead time instants are predicted using the EKF. For completeness, we present an EKF estimation cycle for predicting the future position of a moving object. Consider a system that can be described by a non-linear discrete-time state transition equation of the form:

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, k\right) + \mathbf{w}_k \quad (1)$$

where $\mathbf{u}_k$ is a known control vector, $\mathbf{x}_k$ is the state at time instant k, $\mathbf{f}\left(\cdot, \cdot, k\right)$ is the non-linear function that maps the previous state and control inputs to the current state.

The observations of the state(s) of this system are made via a non-linear discrete-time observation equation of the form:

$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k, k\right) + \mathbf{v}_k \tag{2}$$

where $\mathbf{h}\left(\cdot, k\right)$ is the non-linear function that maps the current state to observations. The process noise, $\mathbf{w}_k$ and the measurement noise, $\mathbf{v}_k$, are assumed to be Gaussian, uncorrelated and zero-mean.

In an autonomous vehicle navigation context, the prediction stage uses a model of the motion of the vehicle (a process model having the form described in Equation (1)) to predict the vehicle position, $\hat{\mathbf{x}}_{(k|k-1)}$, at instant k given the information available until and including instant $(k-1)$. The state prediction function $\mathbf{f}(\cdot)$ is defined by Equation (1) assuming zero process and control noise. The prediction of state is therefore obtained by simply substituting the previous state and current control inputs into the state transition equation with no noise. The prediction of covariance is obtained as:

$$\mathbf{P}_{(k|k-1)} = \nabla\mathbf{f}_{\mathbf{x}_k}\mathbf{P}_{(k-1|k-1)}\nabla\mathbf{f}_{\mathbf{x}_k}^T + \nabla\mathbf{f}_{\mathbf{w}_k}\mathbf{Q}_k\nabla\mathbf{f}_{\mathbf{w}_k}^T$$

where $\nabla\mathbf{f}_{\mathbf{x}_k}$ and $\nabla\mathbf{f}_{\mathbf{w}_k}$ are the state and control Jacobians.

Once the state and covariance predictions are available, the next step is to compute a predicted observation and a corresponding innovation for updating the predicted state. The predicted observation $\hat{\mathbf{z}}_{(k|k-1)}$ is found by using the non-linear relation (a observation model having the form described in Equation (2)) and taking expectations conditioned on the first $(k-1)$ observations such that

$$\begin{aligned}\hat{\mathbf{z}}_{(k|k-1)} &\stackrel{\triangle}{=} \mathrm{E}[\mathbf{z}_k|\mathbf{Z}_{k-1}] \\ &= \mathbf{h}(\hat{\mathbf{x}}_{(k|k-1)})\end{aligned}$$

The innovation vector and innovation covariance are given by

$$\begin{aligned}\nu_k &= \mathbf{z}_k - \hat{\mathbf{z}}_{(k|k-1)} \\ \mathbf{S}_k &= \nabla\mathbf{h}_{\mathbf{x}_k}\mathbf{P}_{(k|k-1)}\nabla\mathbf{h}_{\mathbf{x}_k}^T + \mathbf{R}_k\end{aligned}$$

where $\nabla\mathbf{h}_{\mathbf{x}_k}$ is the observation Jacobian.

The state estimate and covariance updates are:

$$\begin{aligned}\hat{\mathbf{x}}_{(k|k)} &= \hat{\mathbf{x}}_{(k|k-1)} + \mathbf{W}_k \cdot \nu_k \\ \mathbf{P}_{(k|k)} &= \mathbf{P}_{(k|k-1)} - \mathbf{W}_k\mathbf{S}_k\mathbf{W}_k^T\end{aligned}$$

where the Kalman gain matrix is given by

$$\mathbf{W}_k = \mathbf{P}_{(k|k-1)}\nabla\mathbf{h}_{\mathbf{x}_k}^T\mathbf{S}_k^{-1}$$

It should be noted here that the estimation-theoretic short-term prediction approach does not incorporate a priori knowledge such as road networks and traffic signage and assumes uninfluenced constant trajectory.

IV. Situation Recognition- and Probabilistic Prediction-based Moving Object Prediction

At the higher levels of the control hierarchy, namely the `RouteSegment Manager` and the `DrivingBehavior Manager`, moving object prediction needs to occur at a much lower frequency and a greater level of inaccuracy is tolerable. At these levels, moving objects are identified as far as the sensors can detect, and a determination is made as to which objects should be classified as "objects of interest". In this context, an object of interest is an object that has a possibility of affecting our path in the time horizon in which we are planning.

Once objects of interest are identified, we use situation recognition and probabilistic prediction algorithms to predict we expect that object to be at various time steps into the future. In these algorithms, we are typically looking at planning horizons on the order of tens of seconds into the future with plan steps at about one second intervals. At this level, we are not looking to predict the exact location of the moving object. Instead, we are attempting to characterize the types of actions we expect the moving object to take and the approximate location the moving object would be in if it took that action. The approach to performing this prediction is exposed in Figure 3 and is described as follows:

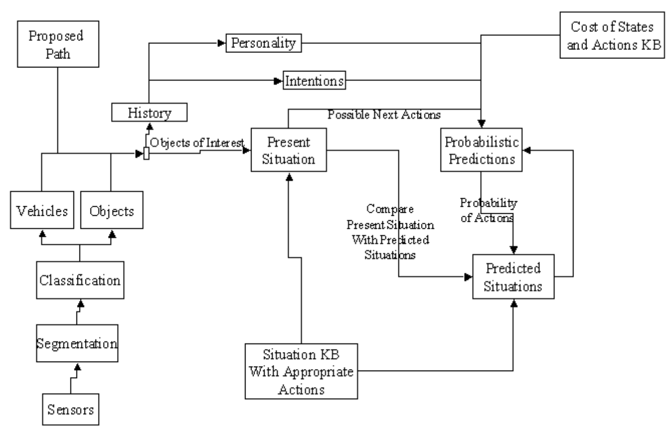


Fig. 3. Probabilistic prediction based on situation recognition.

1) Using a situation knowledge base, attempt to characterize the current situation(s) that the object of interest is in based upon the type of object using pattern matching.
2) Based upon the identified situation(s), identify the pertinent actions that could result from being in that situation. A separate knowledge base is used to perform this, which captures the association between discrete situations and pertinent actions. If the moving object can be characterized as being on more than one situation, the union of all actions associated with each situation is considered.
3) For each pertinent action, associate a probability that the action will occur based upon an underlying cost model. The cost model will contain costs that the moving object will incur based upon the resulting state that the object

will be in one it complete the action and costs that the moving object will incur according to the transition it makes in between states (the action itself). An example of a cost that the moving object may incur for occupying a state includes being too close to another object. A cost the vehicle may incur during the action is crossing over a double yellow line or exceeding the speed limit. The probabilities are roughly inversely proportional to the costs, such that the highest cost action would result in the lowest probability and vice versa.

4) Based upon the results of each pertinent action, go to step 1 and attempt to classify each resultant state which a situation within the knowledge base.
5) This process is repeated until we get to the planning horizon that is appropriate to the level of planning within the control hierarchy in which we are attempting to plan with.
6) As time progresses and data comes in as to which action the moving object took, the initial action probabilities are updated such that the action that was taken now has a 100% probability and all other branches of the tree which correspond to actions that were not taken are eliminated.

The costs mentioned in step 3 are not fixed but vary depending on the perceived characteristics of the moving objects. In addition to the steps that were described above, a separate process is tracking the history of the moving object and abstracting from that history the pertinent actions that could help to infer pertinent characteristics of that object, such as personality and intentions. For example, this process may extract out information about the number of time the moving object changes lanes, how often it exceeded the speed limit by a certain amount, how closely it followed other vehicles, and the standard deviation in its speed over a given time interval. Utilizing this information, the system would assign a personality level (roughly analogous to aggressive driving levels) and would try to infer the intention of the object based upon patterns of behavior. For example, if the vehicle was perceived to have switched to the right lane (on a four lane highway) three times in a short span of time, the vehicle may have an assigned intention of desiring to get off at the next exit ramp on the right hand side.

If the moving object is identified as being aggressive, it would have a lower cost for lane changes, which would result in a higher probability of them happening. If the moving object has an assigned intent of getting off at the next exit ramp, the cost for performing that action would be extremely low, again raising the probability that it would happen. Other factors, such as the type of car, also influence the probabilities. A perceived red sports car may have a higher probability of exceeding the speed limit than an old pickup truck.

V. Experimental Results

Both of these prediction methods have been implemented in two different simulation environments. The EKF approach has been implemented in the OneSAF testbed (www.onesaf.com). OneSAF is a composable, next generation computer generated forces (CGF) that represents a full range of operations, systems, and control process from individual combatant and platform to battalion level, with a variable level of fidelity. OneSAF is able to represent moving objects and provide the object's location and velocity at any point in time, through Application Programming Interface (API) calls. A user-interface was built on top of OneSAF to display the predicted locations of the moving objects.

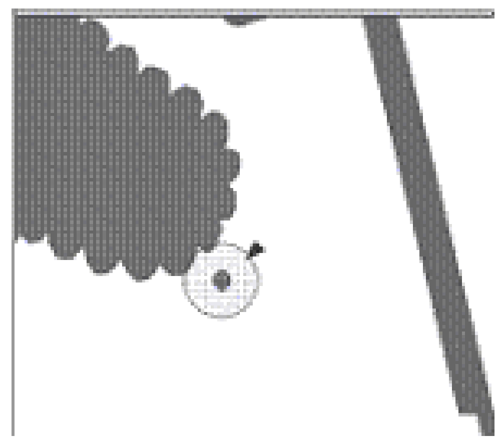


Fig. 4. Estimation-theoretic short-term prediction.

In Figure 4, the triangle ($\triangle$) represents the moving object whose future location is to be predicted. The large circle in front of the triangle is the area in which we are 50% confident that the object will be in two seconds and the small shaded circle is the area in which we are 99% confident that the object will be in two seconds. For our implementation, we found that the EKF provided reasonable predictions within a two second horizon. A horizon greater than two seconds introduced too much uncertainty to be useful for our autonomous driving scenarios.

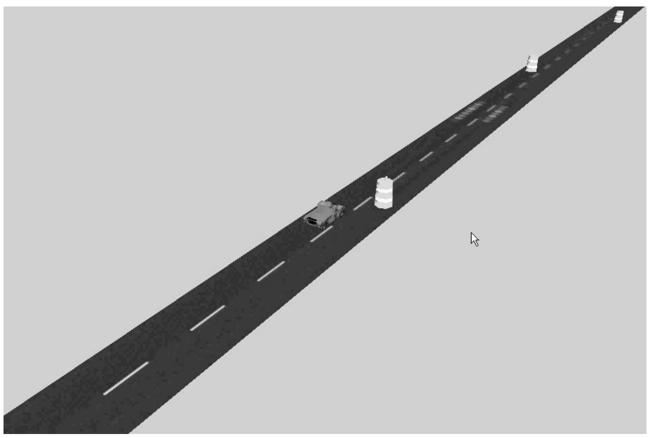
The situation-based probabilistic prediction approach has been implemented in the AutoSim simulation package developed by Advanced Technology Research Corporation. AutoSim is a high-fidelity simulation tool which models details about road networks, including individual lanes, lane markings, intersections, legal intersection traversibility, etc. Using this package, we have simulated typical traffic situations (e.g., multiple cars negotiating around obstacles in the roadway, bi-directional opposing traffic, etc. and have predicted the future location of individual vehicles on the roadway based upon the prediction of where other vehicles are expected to be.

At the point this paper was written, we have simulated a handful of driving situations and have used approximately a dozen costs to determine the probabilities of one action over another. Current costs are incurred based on:

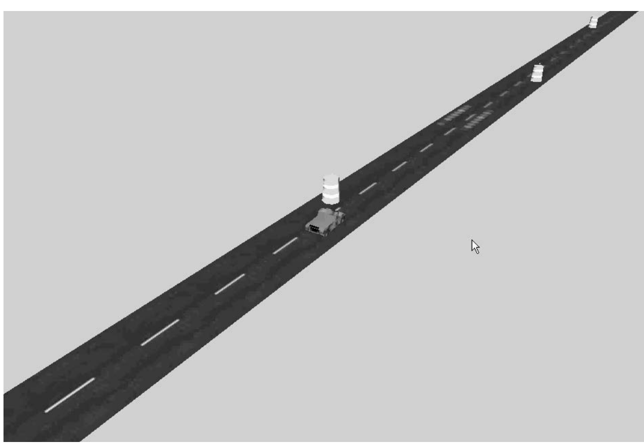
- proximity to other objects in the environment as a function of upon necessary stopping distance,
- exceeding or going below the speed limit by a given

threshold,
- changing lanes,
- not being in the right most lane,
- rapidly accelerating or decelerating, and
- changing lanes where double yellow lines in the road exist, among other costs.

It should be emphasized that costs are not static numbers. The cost that a vehicle incurs by taking an action is heavily a function of the perceived personality and intention of the moving object, as discussed in Section IV. Using these costs, we were able to predict up to ten seconds into the future at a rate of two predictions per second. A snapshot of the implementation is shown in Figure 5, where the different shades of dots correspond to the different probabilities for the vehicle to be in that location five seconds into the future.



(a)



(b)

Fig. 5. Situation-based probabilistic prediction. (a) and (b) show a vehicle performing a passing operation around stationary obstacles.

VI. Conclusions and Future Work

In this paper, we described a hierarchical, multi-resolutional approach for moving object prediction during autonomous on-road driving. The proposed approach currently employs two different prediction methodologies that lend themselves best to the constraints imposed by the planning horizon and replanning rates of the planners at different levels of the control hierarchy. An estimation-theoretic prediction approach is used at the lower levels of the hierarchy that require a fast replanning rate and where constraints on the environment do not greatly affect the predicted location of the moving object. A situation-based probabilistic prediction approach is used at the higher levels of the control hierarchy that require slower replanning rates and where constraints on the environment greatly affect the probabilities of where the moving object will be in the future.

Though both of the above approaches show great promise, there is still much work to be done. For the short-term EKF based approach, we need to build additional kinematic and dynamic models corresponding to different types of vehicles we perceive in the environment. Such models will allow for more accurate predictions that are specific to the types vehicles we encounter. For the situation-based probabilistic approach, we need to encode additional situations (and pertinent actions when encountering those situations), and a more elaborate cost model. We are also investigating methodologies to integrate these two approaches more tightly, such that the results of one prediction approach can help to validate, at some level, the results from the other prediction approach.

References

[1] J. Albus. Outline for a Theory of Intelligence. *IEEE Trans. on Systems, Man, and Cyberbetics*, 21(3):473–509, 1991.

[2] J. Albus et al. 4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems. Technical Report NISTIR 6910, NIST, 2002.

[3] J. Albus and A. Meystel. *Engineering of Mind*. John Wiley & Sons, Inc., 2001.

[4] E.D. Dickmanns. The Development of Machine Vision for Road Vehicles in the Last Decade. In *Proc. of the Intl. Symp. on Intell. Vehicles*, pages 644–651, 2002.

[5] A. Elnager and K. Gupta. Motion Prediction of Moving Objects Based on AutoRegressive Model. *IEEE Trans. on Systems, Man and Cybernetics—Part A: Systems and Humans*, 28(6):803–810, 1998.

[6] H. Haag and H.-H. Nagel. Incremental Recognition of Traffic Situations from Video Image Sequences. *Image and Vision Computing*, 18:137–153, 2000.

[7] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Trans. of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[8] H. Moravec. Sensor Fusion in Certainty Grids for Mobile Robots. *AI Magazine*, 9(2):61–74, 1988.

[9] R. Sharma. Locally Efficient Path Planning in an Uncertain, Dynamic Environment using a Probability Model. *IEEE Trans. on Robotics and Automation*, 8(1):105–110, 1992.

[10] C. Shoemaker and J. Bornstein. The Demo III UGV Program: A Testbed for Autonomous Navigation Research. In *Proc. of the IEEE ISIC/CIRA/ISAS Joint Conference*, pages 644–651, September 1998.

[11] Q. Zhu. Hidden Markov Model for Dynamic Obstacle Avoidance of Mobile Robot Navigation. *IEEE Trans. on Robotics and Automation*, 7(3):390–396, 1991.