Towards Autonomous On-Road Driving via Multi-resolutional and Hierarchical Moving Object Prediction

Jerome Ajot[a], Craig Schlenoff[b], Raj Madhavan[c]
National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD, USA 20899

ABSTRACT

In this paper, we present the PRIDE framework (Prediction In Dynamic Environments), which is a hierarchical multi-resolutional approach for moving object prediction that incorporates multiple prediction algorithms into a single, unifying framework. PRIDE is based upon the 4D/RCS (Real-time Control System) and provides information to planners at the level of granularity that is appropriate for their planning horizon.

The lower levels of the framework utilize estimation theoretic short-term predictions based upon an extended Kalman filter that provide predictions and associated uncertainty measures. The upper levels utilize a probabilistic prediction approach based upon situation recognition with an underlying cost model that provide predictions that incorporate environmental information and constraints. These predictions are made at lower frequencies and at a level of resolution more in line with the needs of higher-level planners.

PRIDE is run in the systems' world model independently of the planner and the control system. The results of the prediction are made available to a planner to allow it to make accurate plans in dynamic environments. 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.

Keywords: Autonomous vehicle, On-road driving, PRIDE, moving object prediction, hierarchical.

1. INTRODUCTION

The field of autonomous systems is continuing to gain traction both with researchers and practitioners. Funding for research in this area has continued to grow over the past few years, and recent high profile funding opportunities have started to push theoretical research efforts into practical use. Autonomous systems in this context refer to embodied intelligent systems that can operate fairly independently from human supervision.

Many believe that the DEMO III Experimental Unmanned Vehicle (XUV) effort represents the state of the art in autonomous driving [11]. This effort seeks to develop and demonstrate new and evolving autonomous vehicle technology, emphasizing perception, navigation, intelligent system architecture, and planning. It should be noted that the DEMO-III XUV has only been tested in highly static environments. It has not been tested in on-road driving situations, which include pedestrians and oncoming traffic.

There have been experiments performed with autonomous vehicles during on-road navigation. Perhaps the most successful has been that of Prof. Dr. Ernst Dickmanns [4] as part of the European Prometheus project in which the autonomous vehicle performed a trip from Munich to Odense (over 1,600 kilometers) at a maximum velocity of 180 km/h. Although the vehicle was able to identify and track other moving vehicles in the environment, it could only make basic predictions of where those vehicles were expected to be at points in the future, considering the vehicle's current velocity and acceleration.

[a] jerome.ajot@nist.gov; phone 1 301 975 3434; www.isd.mel.nist.gov
[b] craig.schlenoff@nist.gov; phone 1 301 975 3456; www.isd.mel.nist.gov
[c] raj.madhavan@nist.gov; phone 1 301 975 2865; www.isd.mel.nist.gov

Mobile Robots XVII, edited by Douglas W. Gage, Proceedings of SPIE Vol. 5609
(SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.580172

What is missing from all of these experiments is a level of situation awareness of how other vehicles in the environment are expected to behave considering the situation in which they find themselves. When humans drive, we often have expectations of how each object in the environment is expected to move based upon the situation they find themselves in. When a vehicle is approaching an object that is stopped in the road, we expect it to slow down behind the object or try to pass it. When we see a vehicle with its blinker on, we expect it to turn or change lanes. When we see a vehicle traveling behind another vehicle at a constant speed, we expect it to continue traveling at that speed. The decisions that we make in our vehicle are largely based upon these assumptions about the behavior of other vehicles.

To date, the authors are not aware of any autonomous vehicle efforts that account for this information when performing path planning. To address this need, we have developed a framework, called PRIDE (PRediction in Dynamic Environments) that provides an autonomous vehicle's planning system with information that it needs to perform path planning in the presence of moving objects [9,10]. In this paper, we describe the "high-level" cost-based probabilistic prediction algorithms in detail.

In Section 2, we survey some related work in moving object prediction and traffic simulation. In Section 3, we give an overview of the PRIDE Framework. In Section 4, we describe the details of the cost-based probabilistic moving object prediction algorithms. Section 5 explains how the moving object prediction output can but used, Section 6 describes some preliminary results, and Section 7 concludes the paper.

2. RELATED WORK

Most of the work in the literature dealing with drivers' actions and predicted behavior has been performed by psychologists in an attempt to explain drivers' behaviors and to identify the reason for certain disfunctions.

There have been a few efforts that have tried to simulate traffic patterns. One of more prominent ones in the literature is ARCHISM [3,5], but even this effort is based upon driving psychology studies. These traffic simulations use laws that can be applied for a specific environment or a specific situation. Some of those postulates can be expanded to generic situations.

Additional work preformed at the Sharif University of Technology of Tehran [6] is based on two assumptions: the maximum speed on a road segment and the potential danger that could be incurred by traversing this road segment. These assumptions and concepts allow the simulation program to create a rank of actions the vehicle can execute based on the danger that could be incurred.

3. THE PRIDE FRAMEWORK

We are using the 4D/RCS (Real-Time Control System) reference model architecture [1] as the basis in which to apply the approaches that are being developed in this effort. 4D/RCS was chosen due to its explicit and well-defined world modeling capabilities and interfaces, as well as its multi-resolution, hierarchical planning approach. Specifically, 4D/RCS allows for planning at multiple levels of abstraction, using different planning approaches as well as utilizing inherently different world model representation requirements. By applying this architecture, we can ensure that the approaches being developed for predicting the future location of moving objects can accommodate different types of planners that have different representational requirements.

The RCS architecture supports multiple behavior generation (BG) systems working cooperatively to compute a final plan for the autonomous system. The spatial and temporal resolution of the individual BG systems along with the amount of time allowed for each BG system to compute a solution are specified by the level of the architecture where it resides. In addition to multiple BG systems, multiple world models are supported with each world model's content being tailored to the systems that it supports (in this case the BG system). As such, it is necessary for the future location of moving objects to be determined differently (at different scales and resolutions) at the different levels of the architecture.

To support this requirement, NIST has developed the PRIDE (PRediction In Dynamic Environments) framework. The underlying concept is based upon a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. This framework supports the prediction of the future location of moving objects at various levels of resolution, thus providing prediction information at the frequency and level of abstraction necessary for planners at different levels within the hierarchy. To date, two prediction approaches have been applied to this framework.

At the lower levels, we utilize estimation theoretic short-term predictions via an extended Kalman filter-based algorithm using sensor data to predict the future location of moving objects with an associated confidence measure. This will not be further described in this paper but more information can be found at [8].
At the higher levels of the framework, 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 a moving object prediction approach based on situation recognition and probabilistic prediction algorithms to predict where object will be at various time steps into the future. Situation recognition is performed using spatio-temporal reasoning and pattern matching with an *a priori* database of situations that are expected to be seen in the environment. These algorithms will be the focus of the remainder of this paper.

Active research is exploring the integration of these two prediction approaches in a way that the predictions from one can help to enforce or not enforce the predictions of the other.

4. COST-BASED PROBABLISTIC PREDICTION ALGORITHMS

The algorithms described in this section are used to predict the future location of moving objects in the environment at larger time planning horizons on the order of tens of seconds into the future with plan steps at about one second intervals. During the explanation of the algorithm, the following scenario will be used (Figure 1). This scenario is composed of three vehicles, two of which (A and B) are in lane L1 and moving to the right, and the third (C) is in lane L2 and moving to the left. In this scenario, D is a static object and is located in L1.

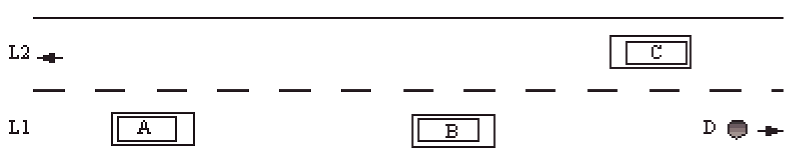

Fig. 1. The Situation

4.1. Implementation details

In this section, we will describe, in detail, the moving object prediction (MOP) algorithms. Figure 2 graphically shows the overall process flow. Figure 3 describes the process in pseudocode.

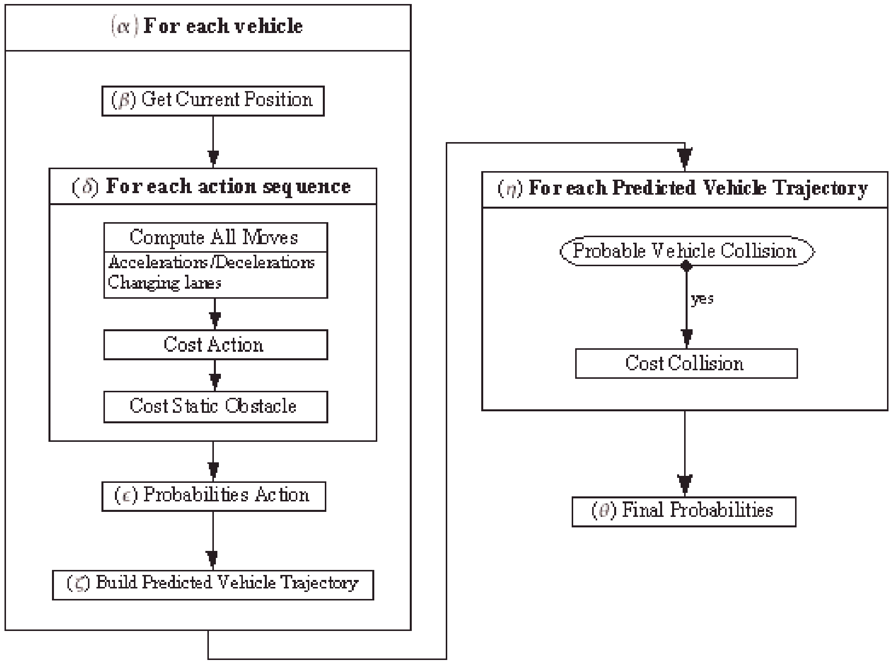

Fig. 2. The Moving Object Prediction Process

```
Begin
  Loop
    For each vehicle (α)
      Get Current Position (β)
      For each action sequences (δ)
        Compute All Moves
        Calculate Cost Action
        Calculate Cost Static Obstacle
      End for
      Calculate Probabilities Action (ε)
      Build Predicted Vehicle Trajectory (ξ)
    End for
    For each Predicted Vehicle Trajectory (η)
      If Probable Vehicle Collision
        Then Calculate Cost Collision
      End if
    End for
    Calculate Final Probabilities (θ)
  End Loop
End.
```

Fig. 3. The Moving Object Prediction (MOP) Pseudocode

The steps within the algorithm are:

1. For each vehicle on the road (α), the algorithm gets the current position and velocity of the vehicle by querying external programs/sensors (β).

2. For each set of possible future actions (δ) (explained in Section 4.2.), the algorithm creates a set of next possible positions and assigns an overall cost to each action based upon the cost incurred by performing the action and the cost incurred based upon the vehicle's proximity to static objects. (explained in Section 4.3.).
3. Based upon the costs determined in Step 2, the algorithm computes the probability for each action the vehicle may perform (ε). At this step in the scenario (Figure 2) the possible actions/probabilities for the three vehicles are shown in Figures 4-6:

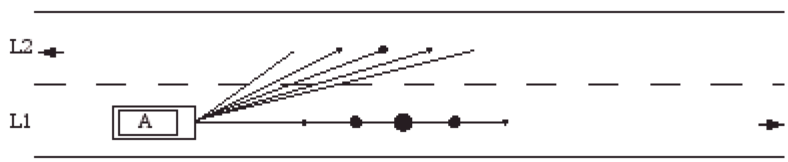

Fig. 4. A-Vehicle Actions-Probabilities

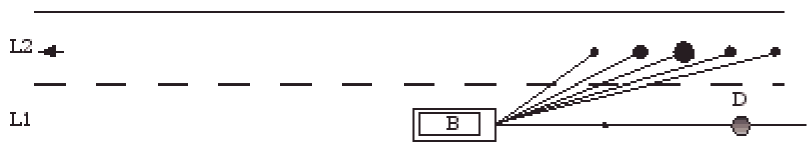

Fig. 5. B-Vehicle Actions-Probabilities

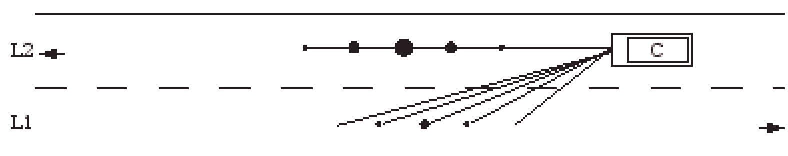

Fig. 6. C-Vehicle Actions-Probabilities

where the size of each dot represents the relative probability with respect to the others.

4. Predicted Vehicle Trajectories (PVT) (ξ) are built for each vehicle which will be used to evaluate the possibility of collision with other vehicles in the environment. PVTs are a vector that indicates the possible paths that a vehicle will take within a predetermined number of time steps into the future. The Predicted Vehicle Trajectory notion is explained in more detail in Section 4.4.
5. For each pair of PVTs (η), the algorithm checks if a possible collision will occur (where PVTs intersect) and assigns a cost if collision is expected.

   In the scenario, for the vehicles A and C, Figure 7 shows two PVTs that cross, indicating that a collision is possible.

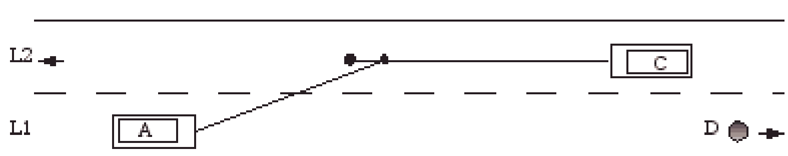

Fig. 7. Possible Collision between A and C

6. In this step, the probabilities of the individual actions (θ) are recalculated, incorporating the risk of collision with other moving objects, as shown in Figures 8-10.

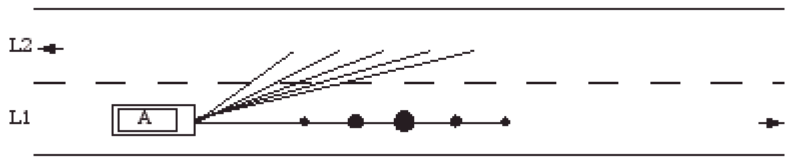

Fig. 8. A-Vehicle Final Probability

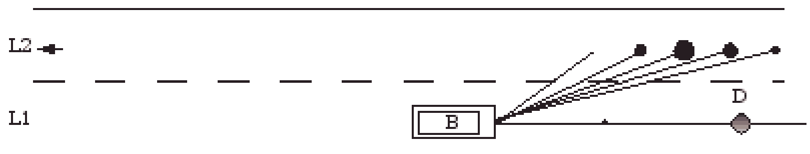

Fig. 9. B-Vehicle Final Probability

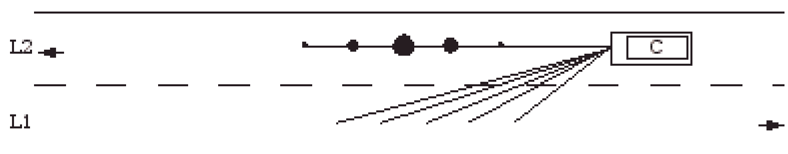

Fig. 10. C-Vehicle Final Probability

At the end of the main loop, the future positions with the highest probabilities for each vehicle represent the most likely location of where the vehicles will be in the future.

4.2. Actions

For the purpose of this work, actions are a discretized set of basic behaviors that a driver may perform during on-road driving. To represent the process of predicting several time steps into the future, a series of continuous actions (action sequences) are created *a priori* that are consistent with a set of preset rules. These rules will be explained in Section 4.2.2.

4.2.1. Elementary actions

On the straight road, a vehicle can execute two types of actions. The first type of action pertains to its acceleration profile. The possible values of this type are: Quick Acceleration (QA), Slow Acceleration (SA), Keep the same Speed (KS), Slow Deceleration (SD), and Quick Deceleration (QD). The second type of action pertains to the changing lane process. The vehicle can stay in the same lane or change lanes to the right or the left. So there are: Change to the Left lane (CL), stay in the Same Lane (SL), and Change to the Right lane (CR). Note that at this time we had only dealt with continuous road segments. At a later date we will be addressing intersections.

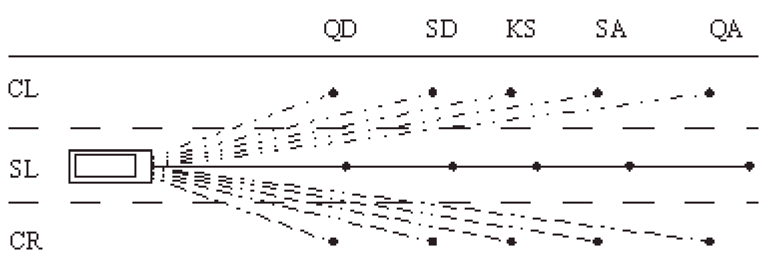

Fig. 11. Vehicle Actions

Thus a vehicle can have up to fifteen possible actions (Figure 11). Every possible elementary action is assumed to be able to be accomplished in one time step. Some actions may not be possible due to the vehicle's current velocity (for example, a vehicle moving very slowly cannot change lanes in one second during a deceleration) or location (if a vehicle is in the rightmost lane, it can not change lanes to the right). In this case, those actions are not considered.

4.2.2. Action Sequences

One action is assumed to be done in one time step, thus to predict *n* time steps in the future, *n* actions will be completed. This set of *n* actions is referred to as an action sequence. These action sequences are only made up of only the acceleration profiles (QA, SA, KS, SD, and QD (Section 4.2.1.)). Assuming that every time step is set to one second (note that this is not a fixed value, it can be set by the user), the action sequence has to be realistic. As such, there are some action sequences that are improbable, and thus eliminated from consideration.

For now, a single rule is applied to all of the possible action sequences to generate the most realistic ones. To evaluate these rules, we associate a value to each "acceleration profile": 2 for QA, 1 for SA, 0 for KS, -1 for SD, and -2 for QD. The rule states that a vehicle can only switch from an action to another action if their values differ by one. Other rules will be added as this work progresses. Example of action sequences and their associated validity is shown in Figure 12.

SD	SD	SD	SD	QD	Valid	
QD	QD	QA	QA	QA	Invalid	QD to QA illegal
QA	QA	SA	SA	KS	Valid	

Fig. 12. Example of valid and invalid Set of action

4.3. Cost Model and Probability

The Moving Object Prediction (MOP) algorithms can be separated in two parts, the first one is the creation of a set a predicted positions independent of other moving objects and the second is the evaluation of the interaction between each vehicle on the road. Every evaluation is based upon an underlying cost model, which in turn is converted to probabilities.

The Cost Model (CM) simulates the danger that a driver would incur by performing an action or occupying a state. These costs can be separated into two different categories.

1. Cost representing the vehicle's actions. This cost represents the penalties to perform an action as a function of the amount of attention needed. For example, the changing lane action needs more concentration than going straight in the same lane, thus the cost for changing lanes is greater. In the same vain, a slow deceleration needs less attention than a fast deceleration, thus the slow deceleration has a lower cost.
2. Cost representing possible collisions on the road. This includes collisions with static and moving objects. Examples of static objects on the road are roadblock and barrier. Examples of dynamic objects on the road are other vehicles. The costs associated with static or moving objects is proportional to the danger and imminence of collision. For example, a road block at one kilometer ahead is less dangerous than another vehicle passing at three meters ahead.

Another notion added to the CM is the aggressivity of the driver. An aggressive driver may associate a lower risk with its proximity to other drivers and conforming to the speed limit. Thus, the aggressivity has an important role on the behavior of a driver on the road. The aggressivity is reflected in the algorithm by a coefficient applied to the costs, such that the more aggressive the driver, the lower the cost for performing an action. Examples of costs are shown in Figure 13.

Quick Acceleration (QA)	4
Slow Acceleration (SA)	2
Keep the same Speed (KS)	1
Slow Deceleration (SD)	2
Quick Deceleration (QD)	3
Changing Lane (CL, CR)	15
Opposite direction	300
Collision (CO)	100

Fig. 13. Example Cost Model values

Given the fact that a higher cost represents a higher danger, the probabilities of an action is inversely proportional to the cost. So a high cost is converted to a low probability.

4.4. Predicted Vehicle Trajectory

A Predicted Vehicle Trajectory (PVT) represents the possible movements of a vehicle throughout the time period being analyzed. The PVT is represented by a trajectory.

Fig. 14. Predicted Vehicle Trajectory

The PVT (Figure 14) is built from the origin position (x_{IP}, y_{IP}, t_{IP}=0) at time=0 to the predicted position (x_{PP}, y_{PP}, t_{PP}=t_{Pred}) where t_{Pred} is the predetermined time in the future for the prediction process. Also contained within the PVT is the action-cost and action-probability information.

The PVT is used to determine if potential collision will occur. Because a PVT represents a trajectory of one predicted position (initial to predicted), to obtain the collision information between two vehicles, the possible intersection between two PVT has to be checked.

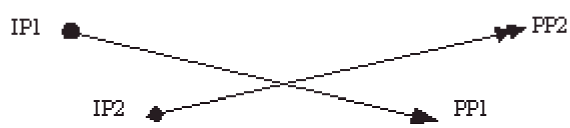

Fig. 15. Crossing PVT

Thus, no collision occurs when two PVTs are not crossing. But when two PVTs do cross, there is a probable collision (Figure 15). When two PVTs cross, it is important to know when (where) they cross. This information can be obtained by using a parametrization of each PVT.

The parametrization is:

$$\begin{aligned} x_1(u_1) &= x_{PP1}\ u_1 + x_{IP1}\ (1-u_1) \\ y_1(u_1) &= y_{PP1}\ u_1 + y_{IP1}\ (1-u_1) \end{aligned} \qquad \text{where } u_1 \in [0, 1] \tag{1}$$

$$\begin{aligned} x_2(u_2) &= x_{PP2}\ u_2 + x_{IP2}\ (1-u_2) \\ y_2(u_2) &= y_{PP2}\ u_2 + y_{IP2}\ (1-u_2) \end{aligned} \qquad \text{where } u_2 \in [0, 1] \tag{2}$$

where u_1 and u_2 are the parameters of each PVT.

The two equations (1) and (2) create a linear system (3) which, after using the Theorem of Cramer, can be used to determine u_1 (4) and u_2 (5):

$$(x_{PP1} - x_{IP1})\, u_1 + (x_{IP2} - x_{PP2})\, u_2 = x_{IP2} - x_{IP1}$$
$$(y_{PP1} - y_{IP1})\, u_1 + (y_{IP2} - y_{PP2})\, u_2 = y_{IP2} - y_{IP1} \quad (3)$$

$$u_1 = \frac{\begin{vmatrix} x_{IP2} - x_{IP1} & x_{IP2} - x_{PP2} \\ y_{IP2} - y_{IP1} & y_{IP2} - y_{PP2} \end{vmatrix}}{\begin{vmatrix} x_{PP1} - x_{IP1} & x_{IP2} - x_{PP2} \\ y_{PP1} - y_{IP1} & y_{IP2} - y_{PP2} \end{vmatrix}} \quad (4)$$

$$u_2 = \frac{\begin{vmatrix} x_{PP1} - x_{IP1} & x_{IP2} - x_{IP1} \\ y_{PP1} - y_{IP1} & y_{IP2} - y_{IP1} \end{vmatrix}}{\begin{vmatrix} x_{PP1} - x_{IP1} & x_{IP2} - x_{PP2} \\ y_{PP1} - y_{IP1} & y_{IP2} - y_{PP2} \end{vmatrix}} \quad (5)$$

So the two vehicles will cross each other at two different times ($u_1\, t_{Pred}$) for the first vehicle and ($u_2\, t_{Pred}$) for the second vehicle. For a small difference, the collision is probable or certain. Conversely, for a large difference, the collision is improbable. Thus if the PVTs cross and the difference of time is less than a predetermined time (T), we use Equation 6 to determine the collision cost:

$$\text{CollisionCost} = \text{CO}\left(T - (t_{Pred}\, |u_1 - u_2|)\right) \quad (6)$$

where CO is the predetermined maximum cost that can occur when colliding with a specific object (Figure 13) and T is the predetermined time difference in which a cost for collision will be incurred.

5. HOW THE MOP OUTPUT CAN BE USED

In Section 4, we described how we associate probabilities to the possible future location of objects in the environment. In this section, we will describe how the outputs of the MOP algorithms are expected to be used by a planner. Before we do so, we need to describe the expected output format of the MOP algorithms.

Each time the algorithms are run, the following information will be provided for each possible future location of every pertinent moving object in the environment.

Time Step In The Future	Vehicle ID	Vehicle Type ID	XPosition	YPosition	Probability
1	10	2	10.5	11.5	40
1	10	2	11.5	11.5	20
1	10	2	10.5	10.5	30
1	10	2	11.5	10.5	10
1	11	2	10.5	12.0	30
...	...	...	...	...	...

Fig. 16. MOP Output

The MOP output (Figure 16) is composed of a list of time steps in the future, external vehicle information (ID and type of the vehicle), all the possible future locations (XPosition, YPosition), and probability information.

Every predicted location has an associated probability to represent the probable occupancy of the vehicle on the road at a certain time. Some of these predicted positions are not relevant due to a low probability, so a threshold can be applied to ignore those locations under the threshold value.

To obtain all the future positions within the prediction horizon (Figure 14), we use 1) the action sequence (Section 4.2.2.) used to create the PVT, 2) the velocity, and 3) a parameterization of the PVT.

	Initial Position	**Intermediate Predicted Positions**			**Predicted Position**
Action Sequence		QA	QA	SA	KS
Velocity ($m.s^{-1}$)	20	22	24	25	25
Parameter	0.00	0.23	0.48	0.74	1.00
XPosition (m)	0.0	0.7	1.4	2.2	3.0
YPosition (m)	10.0	32.1	56.1	81.0	106.0

Fig. 17. Example of a PVT parameterization

Figure 17 is an example of a PVT parameterization, which is used to obtain all the intermediate predicted positions within the prediction horizon. The action sequence shows the chain of actions that the vehicle is predicted to perform. The velocity shows the expected velocity that the vehicle is expected be at after performing each action. The parameter is the ratio of the distance the vehicle has traveled (after performing the action) as compared to the prediction final position of the entire PVT. The XPosition and YPosition are the corresponding location of the vehicle as determined by the parameter value. By using this, we can find where the vehicle is expected to be at each time step within the prediction horizon.

It is expected that a planner will use the probability information from the MOP to determine the damage potential of occupying a location in space at a given time. Specifically, this damage potential will be based on the object it will encounter and the probability that the object will be there. For example, if the MOP algorithms determine that a HMMWV (High-Mobility Multipurpose Wheeled Vehicle) (which we assume has a maximum damage potential of 200) has a 40% chance of occupying a point in space, then the planner may associate a damage potential (due to the presence of moving objects) of 80 (40% of 200) when determining the most optimal path.

6. EXPERIMENTAL RESULTS

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 (Figure 18).

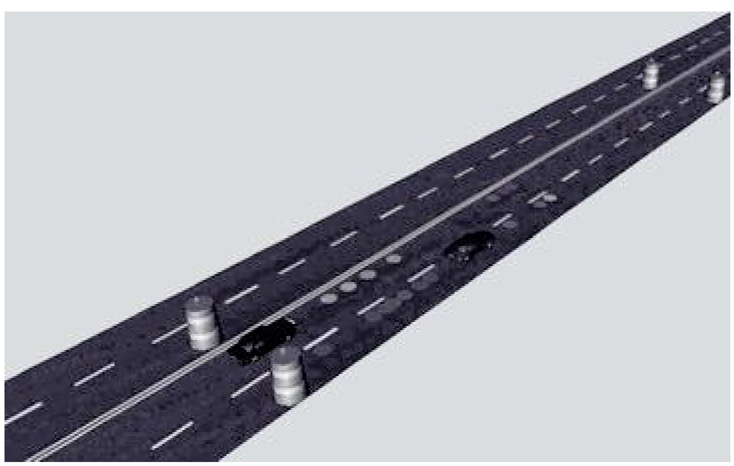

Fig. 18. Two Vehicles passing obstacles

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: 1) proximity to other objects in the environment as a function of the necessary stopping distance, 2) exceeding or going below the speed limit by a given threshold, 3) changing lanes, 4) not being in the right most lane, 5) rapidly accelerating or decelerating, and 6) 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 objects. Using these costs, we were able to predict up to five seconds into the future at a rate of five predictions per second.

Results show that predicting less than 3-4 seconds in the future does not properly account for sequence of actions (e.g., passing). As such, the prediction algorithms must look at least 3-4 seconds in the future to be able to properly predict and model higher level driving maneuvers.

The purpose of these algorithms are to work in real time, the experimental results show that after seven seconds in the future, the algorithms become jerky and lose their real time functionality (CPU: P4 1.8 Ghz, memory: 512 MB).

7. CONCLUSION/FUTURE WORKS

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-theory 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. In this paper, we have described the higher-level prediction algorithms in detail.

Even 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 of vehicles we encounter. For the situation-based probabilistic approach, we need 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.

ACKNOWLEDGEMENT

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Mobile Autonomous Robot Software (MARS) program (PM. D. Gage).

REFERENCES

1. Albus, J. & al, *4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems*, NISTIR 6910, National Institute of Standards and Technology, Gaithersburg, MD, 2002.

2. Balakirsky, S., *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Spaces*, IOS Press, Berlin, 2003.

3. Champion, A., Espie, S., and Auberlet, J., "Behavioral Road Traffic Simulation with ARCHISM", *Proceedings of the Summer Computer Simulation Conference*, USA, 2001.

4. Dickmanns, E. D., "An Expectation-Based Multi- Focal Saccadic (EMS) Vision System for Vehicle Guidance", *Proceedings of the 9th International Symposium on Robotics Research (ISRR'99)*, Salt Lake City, 1999.

5. Espie, S., Saad, F., and Schnetler, B., "Microscopic traffic simulation and driver behavior modeling: the ARCHISM project", *Proceedings of the Strategic Highway Research Program and Traffic Safety on Two Continents*, Lille, France, 1994.

6. Hoseini, S., Vaziri, M., Shafahi, Y., “Combination of Car Following and Lane Changing Models as a Drivers’ Optimization Process”, *Applications of Advanced Technologies in Transportation Engineering*, Reston, ASCE and T&DI, 0-7844-0730-4, pp. 601-605 , 2004.

7. Kalman, R., "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME Journal of Basic Engineering*, **Vol. 82**, **No. Series D**, pp. 35-45, 1960.

8. Madhavan, R. and Schlenoff, C., "The Effect of Process Models on Short-term Prediction of Moving Objects for Unmanned Ground Vehicles", *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, Washington DC, 2004.

9. Schlenoff, C., Ajot, J., Madhavan, R., "PRIDE: A Framework for Performance Evaluation of Intelligent Vehicles in Dynamic, On-Road Environments", *Proceedings of the Performance Metrics for Autonomous Systems (PerMIS) 2004 workshop*, 2004.

10. Schlenoff, C., Madhavan, R., and Barbera, T., "A Hierarchical, Multi-Resolutional Moving Object Prediction Approach for Autonomous On-Road Driving", *Proceedings of the 2004 ICRA Conference*, 2004.

11. Shoemaker, C. and Bornstein, J. A., "Overview of the Demo III UGV Program", *Proceedings of the SPIE Robotic and Semi-Robotic Ground Vehicle Technology Conference*, **Vol. 3366**, pp. 202- 211, 1998.

12. Scrapper, C., Balakirsky, S., and Weiss, B., "Autonomous Road Driving Arenas for Performance Evaluation", *Proceedings of the Performance Metrics for Autonomous Systems (PerMIS) 2004 workshop*, 2004.