A Brief History of PRIDE

Z. Kootbally, C. Schlenoff and R. Madhavan
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD, USA
Email: {zeid.kootbally, craig.schlenoff, raj.madhavan}@nist.gov

Abstract — PRIDE (PRediction In Dynamic Environments) is a framework that provides an autonomous vehicle's planning system with information that it needs to perform path planning in the presence of moving objects. 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.
This paper presents the chronology of the development of the PRIDE framework. We describe the different prediction algorithms developed for moving object predictions. We provide details on different work performed specifically for each prediction algorithm and how these algorithms are used together to give better predictions. The chronology also relates the successive simulation packages and testbeds[1] used in each step of the development of the PRIDE framework.

Keywords: *4D/RCS, aggressivity, autonomous vehicles, critical time points, long-term prediction, moving object prediction, PRIDE, short-term prediction, integration methodology.*

I. Introduction

The field of autonomous ground vehicles has made prominent strides during the last decade. Advancements have been made in methods for autonomous navigation of autonomous vehicles in dynamic environments. 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 off-road driving [17]. 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 also been experiments performed with autonomous vehicles during on-road navigation. Perhaps the most successful has been that of Dickmanns [2] as part of the European Prometheus project in which the autonomous vehicle performed a trip from Munich to Odense ($>$ 1600 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 near future, considering the vehicle's current velocity and acceleration. The agent architecture AUTODRIVE [19] simulates the generation and execution of a driver's plan to reach a destination safely while taking account of other road users and obeying traffic signs and signals. The selection of appropriate goals is made through a process of "dynamic goal creation" that causes the continual run-time creation and modification of sub-goals.

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 dysfunctions [1], [3], [7]. Our research interest bears upon 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, they often have expectations of how each object in the environment is expected to move according to 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 on these assumptions about the behavior of other vehicles.

To address this need, we have developed a multi-resolutional, hierarchical 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 [12], [15]. 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.

This paper presents the chronology of the development of the PRIDE framework, starting back in 2003 when the

[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.

initial concept called Moving Object Representation, Prediction, and Planning System (MORPPS) was first introduced using a Kalman filter-based prediction approach. In 2004, we started using the AutoSim simulation package to provide higher resolution simulations of moving objects and on-road driving. We also introduced a second set of prediction algorithms that predicted at longer timeframes (seconds into the future as opposed to tenths of seconds). The term PRIDE appeared in 2005 and looked at using the outputs of the two prediction approaches to strengthen/weaken the results of the other. PRIDE was also applied to simulate realistic traffic patterns during on-road driving by using the longer-term prediction algorithms to control individual vehicles on a crowded roadway. More recently, in 2006 and 2007, work has been performed to determine the future time horizons when the different prediction algorithms give the best results. We also started incorporating driver aggressivity into the longer-term algorithms, and determined how the perceived aggressivity of a driver in the environment affected the future position of the vehicle they were driving. During this same time, we ported the PRIDE algorithms over to the Mobility Open Architecture Simulation and Tools (MOAST) and the Urban Search and Rescue Simulation (USARSim) framework [16], which provided a higher-fidelity simulation platform with a physic-based engine.

This paper is organized as follows: Section II presents the initial concept called Moving Object Representation, Prediction, and Planning System (MORPPS), which explored logic-based motion prediction while using different prediction algorithms for different environments. Section III provides an overview of the PRIDE framework. Section IV gives details on the short-term prediction approach along with the description of LAser Detection And Ranging (LADAR) noise models. Section V describes the second prediction approach, the long-term, cost-based, probabilistic moving object prediction algorithms. Section VI provides information on different works performed on the integration of the long-term and short-term predicted estimates. Section VII discusses the role of aggressivity in PRIDE and describes how it is addressed using MOAST and the USARSim simulation environment. Section VIII concludes the paper and gives an overview on future work.

II. The Days of MORPPS

The initial moving object framework called MORPPS (Moving Object Representation, Prediction, and Planning System) [14] was developed in 2003. This framework provides a mechanism to apply appropriate prediction algorithms and representational approaches in order to fully capture the information needed to navigate in the presence of moving objects.

A. Logic-Based Motion Predictions in Constrained Environments

The framework explores logic-based prediction algorithms for use in constrained environments. The purpose of these algorithms is to predict the probability that an object will occupy a given location in space at a given time by taking into account: a) the constraints that are placed on the object's motion and b) the influencing factors that would cause it to take a given action over another at specific times.

In the case of on-road driving, vehicles must stay on the road and as such, the road network provides the constraints dictating the bounds in which a vehicle may travel. A database structure [4] has been developed to capture detailed information about the road network, which includes information about the curvature of lanes, road interconnectivity, signage and traffic control, lane marking, etc.

The rule-based prediction approach requires that one discretizes the possible actions that a moving object may take. In the case of a vehicle driving on-road, we limit the actions of the vehicle to be: remain at a constant velocity in the current lane, slowly accelerate in the current lane, rapidly accelerate in the current lane, slowly decelerate in the current lane, rapidly decelerate in the current lane, change to a lane on the left, change to a lane on the right, turn to a lane on the left (at an intersection), turn to a lane on the right (at an intersection), make a U-Turn (at an intersection).

B. Constraints on Motion and Influencing Factors

Different factors can affect the probabilities associated with the possible actions that a vehicle may take while driving on-road. There are two classes of factors that we must consider. The first are factors that limit the possibilities of where the vehicle is able to reach. In other words, by considering these factors, we can eliminate certain portions on the maps that are not reachable by the vehicle. We call these *constraints on motion*. An example of a constraint on motion is *a priori* road network information, where the road network limits the possible locations that the vehicle can possibly attain.

The second are factors that influence which of the possible actions the vehicle is likely to perform out of those that are available to it. We call these *influencing factors*. An example of an influencing factor can be the weather and the environmental conditions. Weather and environmental conditions affect the visibility and slickness of the road surfaces. As the weather and environmental conditions worsen, the probability often increases that the vehicle's velocity will decrease.

III. The PRIDE Framework

Many efforts on the framework led to the second generation of MORPPS called PRIDE (PRediction In Dynamic Environments) that was conceived in 2004. From this time, we consider PRIDE as a multi-resolutional, hierarchical framework that provides an autonomous vehicle's planning system with information required to perform path planning in the presence of moving objects. 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 understand the way that PRIDE was developed and the functionality that it is intended to provide, it is important to understand the 4D/RCS architecture [5] on which it was based. 4D refers to the four dimensions (three dimensions of space and one dimension of time), and RCS stands for Real-time Control Systems. The 4D/RCS architecture provides a reference model for unmanned vehicles on how their software components should be identified and organized. It defines ways of interacting to ensure that high-level objectives can be met. To achieve this, the 4D/RCS reference model provides well defined and highly coordinated sensory processing, world modeling, knowledge management, cost/benefit analysis, behavior generation, and messaging functions, as well as the associated interfaces.

The 4D/RCS conceptual framework spans the entire range of operations that affect intelligent vehicles, from those that take place over time periods of milliseconds and distances of millimeters to those that take place over time periods of months and distances of thousands of kilometers. The 4D/RCS model is intended to allow for the representation of activities that range from detailed dynamic analysis of a single actuator in a single vehicle subsystem to the combined activity of planning and control for hundreds of vehicles and human beings in full dimensional operations covering an entire theater of battle. In order to span the wide range of activities included within the conceptual framework, 4D/RCS adopts a multilevel hierarchical architecture with different range and resolution in time and space at each level, as shown for a military environment in Figure 1 [5] and described below.

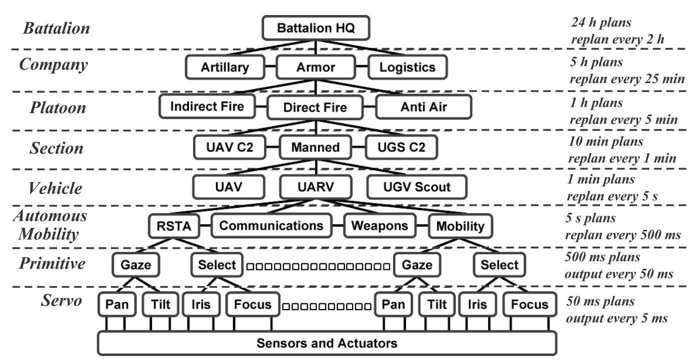

Fig. 1. A high level block diagram of a typical 4D/RCS reference model architecture.

At the Servo level, commands to actuator groups are decomposed into control signals to individual actuators. Outputs to actuators are generated every 5 milliseconds (ms). Plans that look ahead 50 ms are regenerated for each actuator every 5 ms. Plans of individual actuators are synchronized so that coordinated motion can be achieved for multiple actuators within an actuator group. At the Primitive level, multiple actuator groups are coordinated and dynamical interactions between actuator groups are taken into account. Plans look ahead 500 ms and are recomputed every 50 ms. At the Autonomous Mobility level, all the components within an entire subsystem are coordinated, and planning takes into consideration issues such as obstacle avoidance and gaze control. Plans look ahead 5 seconds (s) and replanning occurs every 500 ms. At the Vehicle level, all the subsystems within an entire vehicle are coordinated to generate tactical behaviors. Plans look ahead 1 minute (min) and replanning occurs every 5 s. At the Section level, multiple vehicles are coordinated to generate joint tactical behaviors. Plans look ahead 10 min and replanning occurs about every minute. At the Platoon level, multiple sections containing a total of 10 or more vehicles of different types are coordinated to generate platoon tactics. Plans look ahead an hour (h) and replanning occurs every 5 min. At the Company level, multiple platoons containing a total of 40 or more vehicles of different types are coordinated to generate company tactics. Plans look ahead 5 h and replanning occurs every 25 min. At the Battalion level, multiple companies containing a total of 160 or more vehicles of different types are coordinated to generate battalion tactics. Plans look ahead 24 h and replanning occurs at least every 2 h.

The PRIDE framework was developed to provide moving object predictions to planners running at any level of the 4D/RCS hierarchy at an appropriate scale and resolution. The underlying concept of the PRIDE framework is based on a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. At the higher levels of the framework (Vehicle level and above, as shown in Figure 1), 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”. Once objects of interest are identified, we use the long-term prediction approach presented in section V to predict where those objects will be at various time steps into the future. At the lower levels (Autonomous Mobility level and below, as shown in Figure 1), we utilize estimation theoretic short-term predictions using sensor data as described in section IV to predict the future location of moving objects with an associated confidence measure.

IV. Implementing the Short-term Prediction Algorithm

Details on the development of a combined probabilistic object classification and estimation theoretic framework to predict the future location of moving objects, along with an associated uncertainty measure can be found in [11]. The framework proposed adopts a more generalized view of moving object representation and prediction in concurrently integrating multiple knowledge representation approaches from disparate sources to completely model the information necessary for dynamic planning.

A. The OneSAF Testbed (OTBSAF)

In this approach, the prediction algorithms are tested using the OneSAF (OTBSAF) testbed as the virtual sensor. OTBSAF is a simulation package used for integrating, testing and user feedback of technology developments into the OneSAF Objective System. It provides operational environments useful for identifying, developing, prototyping, demonstrating, and testing of enabling technologies and entity behaviors. As a simulated environment, OTBSAF is able to represent moving objects. By querying OTBSAF, we can retrieve an object's location and velocity at the current time. To validate the testbed and prediction algorithms, we are initially using this retrieved data to serve as our processed sensor data.

B. LADAR Noise Model

In this work, the LADAR sensor is the primary source of sensor data. The data retrieved from OTBSAF is perfect sensor data. In other words, when we ask for the location or dimensions of the object, we are presented with the exact location and the exact dimensions without any associated uncertainty. Although convenient, this does not represent the information that we expect to get from sensors on the actual vehicle. To compensate for this, we have introduced a noise model into the data retrieved from OTBSAF [11].

C. Prediction of Moving Objects

An Extended Kalman Filter (EKF) is employed to predict (estimate) the position and velocity of the moving object at a future time instant. Kalman's prediction theory allows the computation of the best estimate of a future system state by using the most recent estimates of system state along with the system dynamic model. With appropriate interpretation, covariance analysis inherent in the Kalman filtering techniques serves as a confidence measure indicative of the uncertainty in the predicted system states. The EKF thus provides a convenient measure of prediction accuracy through the covariance matrix. The EKF employs a nonlinear model derived from equations based on the kinematics of the moving objects (vehicles) to be predicted.

The EKF is a well established recursive state estimation technique where estimates the states of a nonlinear system are obtained by *linearization* of the nonlinear state and observation equations. Within the PRIDE framework, short-term prediction of objects moving at variable speeds and at given look-ahead time instants (every one-tenth of a second) are predicted using the EKF. It should be noted here that, in contrast to the long-term predictions, the estimation-theoretic short-term prediction algorithm does not incorporate *a priori* knowledge such as road networks and traffic signage and assumes uninfluenced constant trajectory. More information on the short-term prediction algorithm can be found in [10].

V. Implementing the Long-term Prediction Algorithm

The long-term (LT) situation-based probabilistic prediction approach was implemented in AutoSim in 2004 [12]. Autosim was developed by the Advanced Technology Research Corporation and was used to provide higher resolution visualizations of moving objects and on-road driving. AutoSim is a high-fidelity visualization 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.

The LT prediction approach is used to predict the future location of moving objects for longer time horizons. Figure 2 graphically shows the overall process flow.

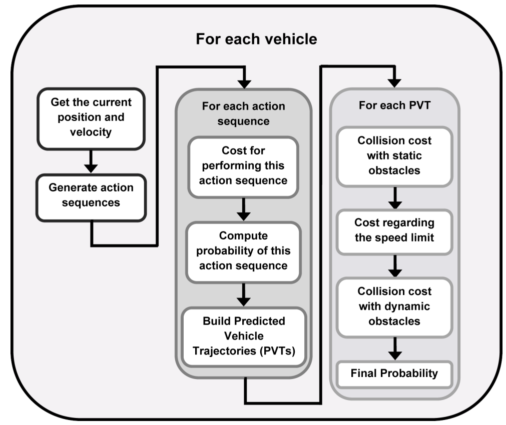

Fig. 2. The situation-based probabilistic (long-term) prediction process.

The output of this loop is a list of locations with associated probabilities showing where a vehicle is expected to be at specific times in the future. Using these probabilities, we can create traffic patterns in one of two ways:

- Control the vehicle to move to the location with the highest probability. For example, if the vehicle has a 40 % chance of being at location A, a 30 % chance of being at location B, a 20 % chance of being at location C, and a 10 % chance of being at location D, the vehicle will always be commanded to move to location A.
- Control the vehicle to move to a location whose likelihood is proportional to the probability that it is expected to be there. One approach would be to use a random number generator. In this way, a vehicle's movement would be closely tied to the probabilities coming out of the moving object predictor, as opposed to always moving to the location with the highest probability.

Independent of the approach used to control the vehicles, the output of these algorithms result in realistic traffic patterns involving one to many vehicles that can be used as a basis to evaluate the performance of autonomous vehicle within simulated on-road driving scenarios.

A. Possible Vehicle Actions

The process of predicting several time steps into the future consists of a series of continuous actions which constitute a driving procedure. Each action is accomplished in one time step, thus, for a time of prediction n, n actions will be completed. The long-term prediction algorithms use different types of actions. The first type of actions consists of a set of speed profiles: Quick Acceleration (QA), Slow Acceleration (SA), Keep the same Speed (KS), Quick Deceleration (QD), Slow Deceleration (SD). The second type of actions concerns the changing of lanes: a vehicle has the possibilities of staying in its lane (SL), changing to the right lane (CR), changing to the left lane (CL). The last type of action pertains to intersections, a vehicle has the possibility to turn left, to turn right or to go straight through an intersection.

At this step, for each vehicle on the road, the algorithm computes all possible sequences of actions, regarding the current velocity and location. Some actions may not be possible due to the vehicle's current velocity (for example, a vehicle moving slowly cannot change lanes in one second during a deceleration). In this case, those actions are not considered. Each sequence of actions is generated in a realistic way using rules. Presently, 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 at most by one. An example of action sequences and their associated validity is shown in Table I.

TABLE I

EXAMPLE OF VALID AND INVALID SEQUENCES OF ACTIONS.

Actions				*Validity*	*Description*
SD	SD	SD	SD	Valid	
QD	QD	QA	QA	Invalid	QD to QA illegal

B. Cost Model

The sequences of actions are deemed finite, and the probabilistic LT prediction algorithms use an underlying cost model that simulates the danger that a driver would incur by performing an action or occupying a state [15]. These costs are being used by multiple efforts within the program that this effort is a part of. Thus, there is value of building the probabilities directly from these costs to allow for synergy with other efforts. These costs can be separated in two different categories:

1) The cost representing the vehicle's actions: This cost represents the penalties for performing 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 lane is greater.
2) The cost representing the vehicle's state on the road: The proximity to other static and dynamic objects on the road is assigned to a cost of collision with these objects. Examples of static objects on the road are road blocks, debris, etc. 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.

Examples of costs are shown in Table II.

TABLE II

EXAMPLE OF ACTIONS WITH THEIR CORRESPONDING COSTS.

Action	*Cost*
Quick Acceleration (QA)	5
Quick Deceleration (QD)	5
Changing lane (CL, CR)	20
Opposite direction	500
Collision (CO)	1000
Being under the speed limit (US)	5
Being over the speed limit (OS)	5

C. Predicted Vehicle Trajectory

Costs of collision between vehicles are computed using Predicted Vehicle Trajectories (PVTs) which represent the possible movements of vehicle throughout the time period of prediction being analyzed. A PVT is a vector whose origin represents the current position of the vehicle ($x_{IP}, y_{IP}, t_{IP} = 0$) at $time = 0$ and its extremity represents 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.

A collision is detected when PVTs cross each other, the location and time of the collision is determined using a parametrization of each PVT. This information can be obtained by using a parametrization of each PVT as represented in the following equations.

$$\begin{cases} x_1(t_1) = x_{PP_1} t_1 + x_{IP_1}(1 - t_1) \\ y_1(t_1) = y_{PP_1} t_1 + y_{IP_1}(1 - t_1); \ t_1 \in [0, 1] \end{cases} \tag{1}$$

$$\begin{cases} x_1(t_2) = x_{PP_2} t_2 + x_{IP_2}(1 - t_2) \\ y_1(t_2) = y_{PP_2} t_2 + y_{IP_2}(1 - t_2); \ t_2 \in [0, 1] \end{cases} \tag{2}$$

where t_1 and t_2 are the parameters for each PVT. Equations (1) and (2) create a linear system where t_1 and t_2 can be solved using Cramer's rule:

$$t_1 = \frac{\begin{vmatrix} x_{IP_2} - x_{IP_1} & x_{IP_2} - x_{PP_2} \\ y_{IP_2} - y_{IP_1} & y_{IP_2} - y_{PP_2} \end{vmatrix}}{\begin{vmatrix} x_{PP_1} - x_{IP_1} & x_{IP_2} - x_{PP_2} \\ y_{PP_1} - y_{IP_1} & y_{IP_2} - y_{PP_2} \end{vmatrix}}$$

$$t_2 = \frac{\begin{vmatrix} x_{PP_1} - x_{IP_1} & x_{IP_2} - x_{IP_1} \\ y_{PP_1} - y_{IP_1} & y_{IP_2} - y_{IP_1} \end{vmatrix}}{\begin{vmatrix} x_{PP_1} - x_{IP_1} & x_{IP_2} - x_{PP_2} \\ y_{PP_1} - y_{IP_1} & y_{IP_2} - y_{PP_2} \end{vmatrix}}$$

The two vehicles will cross each other at two different times, (t_1, t_{pred}) for the first vehicle, (t_2, t_{pred}) for the second vehicle. For a small difference between the two times, 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 (τ), we use Equation (3) to determine the collision cost:

$$Collision\ Cost = CO\left(\tau - (t_{pred}|t_1 - t_2|)\right) \quad (3)$$

where CO is the predetermined maximum cost than can occur when colliding with a specific object (Table II) and τ is the predetermined time difference in which a cost for collision will be incurred.

D. From Cost to Probability

As discussed previously, the PRIDE algorithms compute n realistic sequences of actions with an associated cost. Based on this cost, we can determine the probability that the vehicle will perform that sequence of actions in the following way. The first step is to create a ratio of the cost for performing a given sequence of actions to the sum of all of the costs for performing n sequences of actions:

$$ratio_i = \frac{\sum_{j=1}^{n} cost_j}{cost_i}, \forall\ i \in [1, n]$$

We then normalize the ratio of each sequence of actions by dividing it by the sum of all of the ratios, as shown in Equation (4):

$$proba_i = \frac{ratio_i}{\sum_{j=1}^{n} ratio_j}, \forall\ i \in [1, n] \quad (4)$$

Equation (4) computes the normalized probability of a given sequence of actions occurring as compared to all sequences of actions that are possible at that time.

VI. Integration of the Long-term and Short-term Predictions

One key component of the PRIDE framework is the ability to integrate the predictions from the two algorithms described in Sections IV and V. With this integration, we are able to increase or decrease the confidence of the results of each of these prediction algorithms based upon how well the predictions align. The methodology used to integrate the long-term prediction estimates with those provided by the short-term prediction algorithm is detailed in [15].

A. Significance of Critical Time Points

We define critical time points as those that lie between time periods when both ST and LT provide useful estimates. This is important as it provides opportunities for leveraging the predictions when both prediction algorithms provide valuable estimates during these times. To facilitate discussion, we define t_{bp} as the *break-off point* beyond which the ST estimates are of little value.

When an exteroceptive sensor observation becomes available, the innovation and the innovation covariance (which is a 2×2 matrix as we are considering x_v and y_v), are checked to determine if the EKF updates are to be performed with that observation. The following two conditions are checked to determine if the observation falls within 2σ (95 %) bounds:

$$\left|\left(\frac{\nu(1)}{\sqrt{\mathbf{S}(1,1)}}\right)\right| < 2.0 \quad \text{and} \quad \left|\left(\frac{\nu(2)}{\sqrt{\mathbf{S}(2,2)}}\right)\right| < 2.0$$

If the above conditions are not satisfied, the ST estimates will no longer be bounded (the covariances of the position estimates grow without bounds) and accordingly their consistency cannot be guaranteed. The time instant at which this occurs is termed the break-off point, t_{bp}.

B. Experimental Results

The integration of the predictions from the two algorithms has been performed in different ways through important efforts.

In 2005, work was performed to apply the integration methodology on a straight line with obstacle avoidance [15]. The resulting prediction estimates showed that while the ST predictions provide accurate position estimates within a shorter time horizon, the quality of the predictions degrade considerably as the time horizons get longer. Conversely, the LT prediction algorithms specifically address this shortcoming by providing realistic estimates at longer time horizons that are amenable for autonomous on-road driving. The probabilistic scaling methodology was used to integrate the two prediction algorithms more tightly, such that the results of the ST prediction can help to validate those of the LT prediction and vice-versa.

In 2006, a new way to apply the integration methodology was implemented [10]. To analyze the performance of the prediction algorithms and to determine the window in which both the ST and LT algorithms provide reasonable results, we let the vehicle traverse the track until the first break-off point occurs. As mentioned earlier, the break-point occurs when the ST estimates are no longer consistent. The integration methodology is used on the ST and LT estimates belonging to the time period $[0 - t_{bp}]$ by varying the speed of the vehicle and the time of prediction.

During the same year, in our last effort using the integration methodology, we have tested the performance of the ST and the LT prediction algorithms with several data sets of varying data rates, speeds and prediction intervals on a closed-track [8]. The results have consistently demonstrated that ST estimates are superior to LT estimates in the time period $[t_0 - 0.25t_{bp}]$ and the LT estimates are to be preferred in the time period immediately after t_{bp} until $2t_{bp}$ especially when no external corrections are available for ST prediction. Subsequently, $[0.25t_{bp} - t_{bp}]$ is the most desired time period for the integration of the ST and LT estimates. We compare the results of the integration methodology performed in the two mentioned time periods along a closed-track. We use the last LT estimates from the previous integration to find the next break-off point, and we repeat the same process until the last break-off point of the track.

VII. Driver Aggressivity

The addition of aggressivities is the latest enhancement to the PRIDE framework. The term aggressivity in this context refers to the following description [18]:

A driving behaviour is aggressive if it is deliberate, likely to increase the risk of collision and is motivated by impatience, annoyance, hostility and/or an attempt to save time.

The aggressivity feature was developed after the integration of the PRIDE framework with the Open Architecture Simulation and Tools (MOAST) and the Urban Search and Rescue Simulation (USARSim) simulation environment [16]. This effort provides predictions incorporating the physics, kinematics and dynamics of vehicles involved in traffic scenarios.

A. Mobility Open Architecture Simulation and Tools (MOAST)

MOAST is a framework that provides a baseline infrastructure for the development, testing, and analysis of autonomous systems that is guided by three principles: 1) Creation of a multi-agent simulation environment and tool set that enables developers to focus their efforts on their area of expertise, 2) Creation of a baseline control system which can be used for the performance evaluation of the new algorithms and subsystems, and 3) Creation of a mechanism that provides a smooth gradient to migrate a system from a purely virtual world to an entirely real implementation.

MOAST implements a control technique which decomposes the control problem into a hierarchy of controllers with each echelon (or level) of control, adding more capabilities to the system. Module-to-module communications in MOAST is accomplished through the Neutral Message Language (NML) [6], based on a message buffer model.

B. Urban Search And Rescue Simulation (USARSim)

USARSim is a high-fidelity physics-based simulation system that provides the embodiment and environment for the development and testing of autonomous systems. This is an open source simulation environment that is based on Epic Games Unreal Tournament 2004. Originally developed to study human robotic interactions in multi-agent environment in an Urban Search And Rescue (USAR) environment [9], USARSim is expanding its capabilities to provide realistic simulation environments to assist in the development and testing of cognitive systems, autonomous nautical vessels, and autonomous road driving vehicles.

USARSim utilizes the Karma Physics engine and high-quality 3D rendering facilities of the Unreal game engine to create a realistic simulation environment that provides the embodiment of a robotic system.

C. System architecture of the MOAST/USARSim and PRIDE Frameworks

The embedded client-server architecture (Figure 3) of the Unreal game engine enables USARSim to provide individualized control over multiple robotic systems through discrete socket interfaces. The interfaces provide a generalized representation language that enables the user to query and control the robots' subsystems. All the communications between the clients (Unreal client and the Controller) and the server are performed through the network. The Unreal Server includes the Unreal Engine, Gamebots to bridge the Unreal Engine with outside applications, the maps and the models (robot models, victims, etc). MOAST first connects to the Unreal Server, then it sends commands to USARSim to spawn a robot. At this step, MOAST listens to the sensor data and sends commands to control the robot.

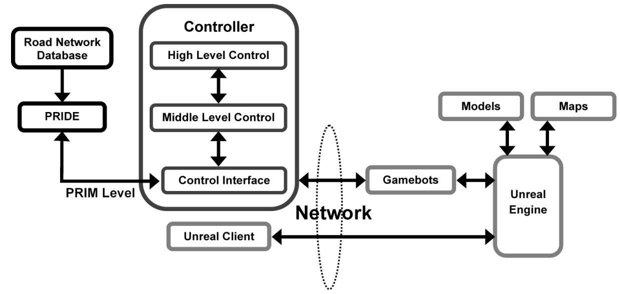

Fig. 3. System architecture of USARSim, MOAST and PRIDE.

As depicted in Figure 3, PRIDE uses a Road Network Database [4] to retrieve the information about road networks for the moving object prediction process. The purpose of the Road Network Database is to provide the data structures necessary to capture all of the information necessary about road networks so that a planner or control system on an autonomous vehicle can plan routes along the roadway at any level of abstraction. The PRIDE framework assumes knowledge of the current position and the velocity of the vehicles on the road to predict their future locations. The PRIDE algorithms retrieve the status (position and velocity) of every vehicle by querying their corresponding navigation channel. At this step, the information from the Road Network Database is used to compute the future positions of the moving objects. The data commands are sent to MOAST through the Primitive level.

D. Modeling Aggressivity within PRIDE

Unlike other approaches that use an underlying static cost model for activities such as path planning, this approach introduces the concept of a dynamic cost model, where the costs are vehicle specific and are a function of what is perceived in the environment. As explained in section V, we associate underlying costs to various actions and states. We then sum the costs that are associated with a specific driving maneuver and use that overall cost to determine the probability that a vehicle will perform that maneuver; the higher the cost to perform the maneuver, the lower the probability that it will occur. However, different drivers have different driving behaviors, and thus have different underlying costs model. One driver may be very conservative, only changing lanes when absolutely necessary, never exceeding the speed limit, etc. On the other hand, another driver may drive very aggressively, weaving in and out of lanes, greatly exceeding the speed limit, and tailgating other drivers. In most cases, one would experience both kinds of drivers on any trip (along with many drivers that fall somewhere in the middle), and a moving object prediction framework needs a mechanism to account for all such circumstances.

When a driver is first encountered, it is extremely rare that one can instantaneously determine the perceived aggressivity of the driver. This information is often determined after observing the driver for a certain amount of time, characterizing their driving behaviors, and assigning an aggressivity. The aggressivity that is assigned greatly impacts PRIDE's predictions as to where that driver will be at times in the future. For example, we would likely assume that a conservative driver will remain in their lanes whenever possible and stay a safe distance behind the vehicle in front of him. An aggressive driver would have a higher probability of changing lanes. We may also find that the aggressivity of the driver may change over times. There are times when one can observe a driver for many seconds at a time. In this case, the driver's aggressivity may change, perhaps they are very aggressively trying to get to a certain lane but become more passive when they get there.

The PRIDE framework addresses all of these driver types and all of the situations mentioned above. Experiments and corresponding results performed on the aggressivity can be found in [13].

VIII. Conclusions and Future Work

The utility of algorithms of predictions has proved to be particularly important with emphasis on complex path planning for autonomous vehicles in dynamic environments. This paper presented the chronology of the development of the PRIDE framework, a hierarchical, multi-resolutional approach for moving object prediction during autonomous on-road driving. We discussed the different concepts used during each step of the development of PRIDE. We described the different prediction algorithms, how they can be used to predict the future location of moving objects. We then showed the features within PRIDE and how they individually make the strength of each algorithm. We also detailed how the short-term and long-term algorithms can be unified to provide better predictions and we gave an overview of different efforts using the integration methodology. We provided an overview of the successive simulation packages used to accomplish more complex traffic situations and used to implement the set of features that constitute this framework today.

Although substantial progress has been made in designing and implementing the PRIDE framework, there is still much to be done. In order to have more complicated traffic situations, we plan on using multiple vehicles in more complex road networks, even though PRIDE is not limited algorithmically to deal with multiple vehicles. In future papers we will tape a real traffic scenario and compare the results to those provided by PRIDE, in this way we can analyze how well PRIDE predicts the future location of the vehicles. PRIDE aims to integrate fuzzy logic for traffic negotiation at intersections and for identification of object of interests. We also plan to upload a release of PRIDE on sourceforge once a stable version is available.

References

[1] A. Champion, S. Espie, and J. Auberlet. Behavioral Road Traffic Simulation with ARCHISM. In *Proceedings of the Summer Computer Simulation Conference*, 2001.

[2] E. Dickmanns. The Development of Machine Vision for Road Vehicles in the Last Decade. In *Proceedings of the International Symposium on Intelligent Vehicles*, pages 644–651, 2002.

[3] S. Espie, F. Saad, and B. Schnetler. Microscopic Traffic Simulation and Driver Behavior Modeling: The ARCHISM Project. In *Proceedings of the Strategic Highway Research Program (SHRP) and Traffic Safety on Two Continents*, 1994.

[4] C. Schlenoff et al. The NIST Road Network Database: Version 1.0. Technical Report NISTIR 7136, National Institute of Standards and Technology, July 2004.

[5] J. Albus et al. 4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems. Technical Report NISTIR 6910, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2002.

[6] V. Gazi, M. Moore, K. Passino, W. Shackleford, and F. Proctor. *The RCS Handbook: Tools for Real-Time Control Systems Software Development*. John Wiley & Sons Inc., New York, NY, June 2001.

[7] S. Hoseini, M. Vaziri, and Y. Shafahi. Combination of Car Following and Lane Changing Models as a Drivers' Optimization Process. *Applications of Advanced Technologies in Transportation Engineering*, 0-7844-0730-4:601–605, 2004.

[8] Z. Kootbally, R. Madhavan, and C. Schlenoff. Prediction in Dynamic Environments via Identification of Critical Time Points. In *Proceedings of the IEEE Workshop on Situation Management (SIMA), Military Communications Conference*, Washington DC, October 2006.

[9] M. Lewis, K. Sycara, and I. Nourbakhsh. Developing a Testbed for Studying Human-Robot Interaction in Urban Search and Rescue. In *Proceedings of the International Conference on Human Computer Interaction (HCI)*, pages 270–274, Crete, Greece, June 22–27 2003.

[10] R. Madhavan, Z. Kootbally, and C. Schlenoff. Prediction in Dynamic Environments for Autonomous On-Road Driving. In *Proceedings of the IEEE International Conference on Control, Automation, Robotics and Computer Vision (ICARCV)*, pages 1690–1695, December 05–08 2006.

[11] R. Madhavan and C. Schlenoff. Moving object prediction for off-road autonomous navigation. In *Proceedings of the SPIE Aerosense Conference*, volume 5083, pages 134–145, Orlando, FL, 2003.

[12] C. Schlenoff, J. Ajot, and R. Madhavan. PRIDE: A Framework for Performance Evaluation of Intelligent Vehicles in Dynamic, On-Road Environments. In *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, 2004.

[13] C. Schlenoff, Z. Kootbally, and R. Madhavan. Driver Aggressivity Analysis within the Prediction In Dynamic Environments (PRIDE) Framework. In *Proceedings of the 2007 SPIE Defense and Security Symposium*, volume 6561, May 2007.

[14] C. Schlenoff, R. Madhavan, and S. Balakirsky. An Approach to Predicting the Location of Moving Objects During On-Road Navigation. In *Proceedings of the Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments: World modeling, planning, learning, an communicating at the International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 71–79, 2003.

[15] C. Schlenoff, R. Madhavan, and Z. Kootbally. PRIDE: A Hierarchical, Integrated Prediction Framework for Autonomous On-Road Driving. In *Proceedings of the 2006 International Conference on Robotic Applications (ICRA)*, pages 2348–2353, 2006.

[16] C. Scrapper, S. Balakirsky, and E. Messina. MOAST and USARSim: A Combined Framework for the Development and Testing of Autonomous Systems. In *Proceedings of the 2006 SPIE Defense and Security Symposium*, volume 6230, Orlando, FL, April 17–21 2006.

[17] C. Shoemaker and J. Bornstein. Overview of the Demo III UGV Program. In *Proceedings of the SPIE Robotic and Semi-Robotic Ground Vehicle Technology Conference*, pages 202–211, 1999.

[18] L. Tasca. A Review of the Literature on Aggressive Driving Research. In *Aggressive Driving Conference*, 1996.

[19] S. Wood. *Planning in a Rapidly Changing Environment*. Phd thesis, School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK, 1990.