Development of the NaviGATOR for the 2007 DARPA Urban Challenge

Carl Crane, David Armstrong, Antonio Arroyo, Antoin Baker, Doug Dankel,
Greg Garcia, Nicholas Johnson, Jaesang Lee, Shannon Ridgeway,
Eric Schwartz, Eric Thorn, Steve Velat, and Ji Hyun Yoon

Center for Intelligent Machines and Robotics
University of Florida, Gainesville, Florida, 32611

Abstract

This paper describes the system design developed for Team Gator Nation's submission to the 2007 DARPA Urban Challenge. In this event, vehicles had to navigate on city streets while obeying basic traffic laws. One of the major challenges was interacting with other vehicles such as at intersections. To address these challenges, a hybrid Toyota Highlander was automated and instrumented with pose estimation (GPS and inertial) and object detection (vision and ladar) sensors. A control architecture was developed which integrates planning, perception, decision making, and control elements. The intelligence element implements the Adaptive Planning Framework which was developed by researchers at the University of Florida. This framework provides a means for situation assessment, behavior mode evaluation, and behavior selection and execution. The paper describes this architecture and concludes with lessons learned from participation in the Urban Challenge event.

I. Introduction

In DARPA's vision, "The Urban Challenge features autonomous ground vehicles maneuvering in a mock city environment, executing simulated military supply missions while merging into moving traffic, navigating traffic circles, negotiating busy intersections, and avoiding obstacles." Moving the challenge into an urban setting adds structure and complexity

to the Grand Challenge problem. Previous success relied on a single mode of operation, without interaction with the environment beyond simple traversal. Success in the Urban Challenge required numerous modes of operation and complex interaction with the environment.

The specific problem to be solved is detailed in the Urban Challenge Technical Evaluation Criteria document [1]. Here the problem was organized into four categories, i.e. Basic Navigation, Basic Traffic, Advanced Navigation, and Advanced Traffic, each of which was more complex than the previous. Upon reviewing this document, the authors identified the following set of technical challenges: pavement and lane detection; detection of static obstacles; detection and classification of dynamic objects; environment data representation and sensor integration with noise in sensor systems; localization ; high level mission planning; determination of appropriate behavior mode and smooth transition between modes; and interprocess communication and coordination of multiple threads on multiple computers. Fault tolerance is another obvious concern, but this was only addressed in a limited fashion due to the experimental nature of the vehicle.

Much work has been done in the past twenty years to address many of the technical challenges. Several references [2-7] provide excellent summaries of the advancements made by other teams competing in the 2005 DARPA Grand Challenge. The authors' work related to the 2005 event is published in two references [8-9]. Numerous additional references can be cited for each of the important technical challenges.

The authors believe that the approach presented here makes new contributions primarily with respect to the work associated with the determination of the appropriate behavior mode and the smooth transition between modes. Traditional approaches, such as for example vision

processing algorithms to identify lane markings in an image, are modified as needed and integrated into the system.

II. Overview of System Architecture

A hybrid Toyota Highlander was selected as the base platform for the system. Steering, throttle, braking, and transmission controls were automated and vision, ladar, inertial, and GPS sensors were mounted to provide necessary information about the environment. The vehicle system is shown in Figure 1.

The system architecture is a natural extension of the Joint Architecture for Unmanned Systems (JAUS) Reference Architecture, Version 3.2, which defines a set of reusable components and their interfaces. The actual core software to support the JAUS messaging system was developed and extensively tested for the previous Grand Challenge and supports the current effort with little or no modification required.

At the highest level, the architecture consists of four basic elements, which are depicted in Figure 2. The Planning Element contains the components that act as a repository for a priori data such as the Route Network Definition File (RNDF) which provides the overall database information about the roads, lanes, and intersections, and the Mission Data File (MDF) which provides the set of RNDF waypoints to traverse for a particular mission. This element also performs the high level route planning and re-planning based on that data plus real-time information provided by the rest of the system. The Control Element contains the Primitive Driver that performs closed-loop control on vehicle actuators to keep the vehicle on a specified path. The Perception Element contains the components that perform the sensing tasks required to determine the vehicle's position, to find a road, to find the lanes on a paved road, to locate both static and dynamic obstacles, and to evaluate the smoothness of terrain. Finally, the

Intelligence Element contains the components that work together to determine the best course of action to navigate the vehicle in a complex environment based on the current mission and situation. An overview of a typical sequence of operations of the architecture is presented as follows (reference Figure 2):

(1) The High Level Planner component performs off-line path planning to generate a desired motion path based on the Route Network Definition File (RNDF) and the Mission Data File (MDF).

(2) A tessellated Local World Model (LWM) (300m × 300m grid with 0.5m resolution) is generated based on a priori road network data and the planned motion path. The center point of the LWM is located at the current location of the vehicle as determined from sensor positioning data.

(3) Data from ladar and vision sensors, which identify static obstacles, dynamic objects, smooth terrain, and road lane regions, is integrated as a layer into the LWM.

(4) Based on the a priori data and sensed data stored in the LWM, software components referred to as Situation Assessment Specialists focus on making specific findings (one simple example is the specialist that reports if the lane to the left, or right, is clear of other vehicles or obstacles).

(5) Seven software components referred to as Behavior Specialists then make an assessment of whether their corresponding behavior mode is appropriate at this moment. The six behavior modes are Roadway Navigation, Open Area Navigation, Pass Left and Pass Right, Reverse Direction, Intersection Traversal, Off Road, and Parking.

(6) A software component referred to as the Decision Broker selects the behavior mode for the system based on the recommendations of the Behavior Specialists.

(7) Based on the behavior mode, a software component called the Smart Arbiter then generates a 60m × 60m traversability grid that is formed to elicit a specific response from the vehicle (change lanes is an example).

(8) Finally, the Receding Horizon Controller component plans a suitable path through the grid that was output by the Smart Arbiter. Steering, throttle, and braking commands are generated to execute the planned path.

A description of the components associated with each of the four elements of the architecture, i.e. planning element, sensor element, intelligence element, and control element follows.

III. Planning Element Components

A. High Level Planner

The High-Level Planner (HLP) provides overall guidance to the vehicle. Its functions include:

1. Creating and maintaining a representation of the RNDF that readily allows for efficient data manipulation during route planning,
2. Using the MDF to plan a route through the RNDF representation using an A* algorithm [10],
3. Periodically communicating waypoints to the Local World Model, so it has an accurate record of the immediate planning space,
4. Re-planning a route when obstacles are encountered, and
5. Collecting data from the sensors about the domain as it is explored and populating the RNDF representation with this information so it contains a more accurate representation of the domain.

B. Local World Model

The Local World Model (LWM) has multiple roles within the system architecture. First, it generates a model of the world based on the a priori RNDF. It receives a subset of the RNDF waypoints within a 300m × 300m area of the vehicle from the High Level Planner (HLP) and draws an estimated picture of the world into a rasterized grid using a resolution of 0.5m. This

raster based approach was chosen because the output from the Local World Model can then be easily incorporated into other system components. The grid resolution of 0.5m was chosen from experience in the 2005 DARPA Grand Challenge and was also used in the 2007 DARPA Urban Challenge. Figure 3a shows an example of the 300m × 300m grid. Other components, such as the perception components, which will be discussed subsequently, work with a smaller 60m × 60m grid. Any needed information is extracted from the 300m × 300m grid and can be transmitted to any necessary components. Figure 3b shows such a sub-sampled grid.

After the initial estimate of the world is created from the RNDF, the Local World Model localizes the vehicle position in the world using data from the Global Positioning (GPOS) component as well as lane finding and path finding sensors. The lane finding and path finding sensors are incorporated to account for possible discrepancies between the RNDF and the GPOS data. The Local World Model takes the position of the center of the sensed lane, and adjusts the a priori world map to fit the sensed world. Figure 4 gives examples where adjustment is necessary. In this figure, the black lines represent the actual road as it is sensed relative to the vehicle, the orange lines are based on the RNDF, and the blue rectangle signifies the vehicle position which is based on GPOS. In (a) either GPOS is incorrect or the RNDF points are not in the lane. In (b) the waypoints do not describe the road accurately. Using data from the lane and path finding sensors the Local World Model accounts for these errors. In (c) the RNDF map has been shifted to align the RNDF road and the sensed world. In (d), the Local World Model has added additional waypoints to correct for the discrepancy between the RNDF and the real road.

After localizing the vehicle, the Local World Model tries to determine important facts about where the vehicle is located in the world and about the environment the vehicle is in. Information such as to whether or not the vehicle is on a road, in an intersection, or in an open

area. If the vehicle is on a road, it determines which road the vehicle is on, whether or not there are lanes to the left and right of the current lane, and the direction of travel of those lanes. It also estimates the distance to an intersection, a stop, or an open area. The results of the analysis are shared with the rest of the system through the appropriate Situation Assessment Specialists and Behavior Specialists.

Next, the Local World Model is responsible for characterizing, predicting, and injecting dynamic information into the world model, which can then be propagated throughout the system. A list of objects is received from the Moving Object and Classification sensor which provides the position, velocity, and size of the detected objects. The Local World Model overlays these objects onto the world map and allows the Urban NaviGator to have a better understanding of what is happening in the world. Figure 5 shows the Local World Model output with moving objects placed into the world grid. With the moving object information placed into the world grid, the Local World Model can make estimates on the distance to an object along the road and recommend a speed to the Receding Horizon Controller so that the vehicle will not collide with an object and also maintain a safe separation distance.

Finally, the Local World Model dynamically spools waypoints to the Receding Horizon Controller. After the HLP has planned a path that completes the mission, it provides a rough plan to the Local World Model that contains only the checkpoints, entry points, and exit points that need to be traversed. The Local World Model then takes the rough plan and fills in the intermediate waypoints that need to be traversed to travel from one HLP point to another. This provides the flexibility to modify the waypoints that need to be traversed based upon the current operating behavior without re-planning the overall mission. Figure 6 shows the change in the mission waypoints based upon a change in the operating behavior. In (a) all the mission points

sent by the HLP are shown. This mission involves making a loop around the course and coming to a stop at the segment at the bottom. In (b) a number of intermediate points have been filled in to be sent to the Smart Arbiter component (discussed subsequently). All points up to a set distance away from the vehicle are sent. In (c) the mission points have been shifted to the other lane in order to execute a change lane behavior due to the obstacle (small solid blue rectangle) detected in the same lane.

In summary, the Local World Model provides a detailed 300m × 300m representation of the environment around the vehicle. It receives a priori roadway data from the High Level Planner as well as static and dynamic obstacle and lane information from the perception system. The Local World Model constantly estimates any discrepancies between the a priori and sensed data by calculating a net offset that can be applied to the a priori data in the event that sensed data is momentarily lost. It determines important facts about the vehicle's position in the world in order to help make decisions on the appropriateness of certain behaviors. Lastly, the Local World Model maintains a list of mission goal points that identify the correct lane of travel. This information is transmitted to the Smart Arbiter component for motion execution

IV. Sensor Element Components

A. Localization

Geo-localization is achieved using a GE Aviation North-Finding-Module (NFM) combined with two GPS units and one odometer. The NFM is an inertial navigation system that maintains Kalman Filter estimates of the vehicle's global position and orientation as well as angular and linear velocities.

The system design is predicated on redundancy and relative accuracy when GPS is lost. The GPS signal provided to the NFM comes from one of the two onboard GPS units. They

include a Novatel Propak, V3-HP with Omnistar subscription service, and a Garmin WAAS Enabled GPS 16. An onboard computer simultaneously parses data from the two GPS units and routes the best-determined signal to the NFM. This is done to maintain the best available GPS information to the NFM at all times. The best GPS solution is determined by evaluating each signal with respect to its unit type, mode of operation, HDOP, RMS, number of satellites, and duration of uninterrupted signal among other criteria. The NFM has been programmed to use a different set of tuning parameters in its Kalman Filter depending on what type of GPS signal it is receiving.

In the event that both GPS units lose track of satellites, as seen during GPS outages such as when the vehicle is in a tunnel, the NFM will maintain localization estimates based on inertial and odometer data. This allows the vehicle to continue on course for a period of time; however, the solution will gradually drift and the accuracy of GPOS will steadily decrease as long as the GPS outage continues. Under ideal conditions the GPOS system typically maintains Global position accuracies and repeatability in the range of 0.1 to 0.5 meters. Figure 7 shows five laps around a 0.6 mile test track with GPS (blue lines) and five laps without GPS (red lines). The vehicle was driven as close as possible in the center of the road (road edges are 28’ apart and are marked by green lines) for every lap. Without GPS, the NFM was using only the encoder signals to damp the velocity errors. Under these conditions the GPOS system maintains Global position accuracies less than 5 meters for a distance traveled of approximately 3 miles without GPS.

B. Perception

The sensor packaged deployed on the vehicle includes an array of LADAR and vision sensors. These include six SICK LMS-291 LADARs, two SICK LD-LRS1000 long range LADARs, and six Matrix Vision BlueFox high-speed USB2.0 color cameras. Moreover, many

of the sensors deployed on the vehicle are articulated with one degree of freedom. Figure 8 depicts the sensor configuration.

1. Smart Sensor Concept

The concept of a *Smart Sensor* is to take various sensor inputs and unify them into a generalized format which can be directly added to other such results. To accomplish this broad task, a generalized data representation was designed which serves to unify the generation, transfer, and analysis of sensor information. This representation, known as the *traversability grid* consists of a tessellated grid tied to a geo-spatial location at the time of generation. The grid then serves as the base data structure for all arbitration processes and can easily be spatially added and fused with other data-sources which are also in a generalized grid representation. Figure 9 depicts three example traversability grids and the result of sensor fusion by an arbitration component.

By utilizing a common data representation, developers can work independently of arbitration components. Moreover, the Smart Sensing components can operate asynchronously at a wide variety of rates and grid resolutions due to the spatially located nature of the traversability grid [11]. This is possible due to the spatial mapping of each grid as it is fused with the other available sensor information. Thus, the arbitration process takes into account the geo-spatial offsets between the various Smart Sensor traversability grids when fusing their information so the resulting representation is consistent as the vehicle moves regardless of speed, orientation, or position.

2. Sensor Groups

The main objectives of the perception systems are to characterize terrain, localize obstacles, and provide error information about the vehicle's pose in the lane. To this end, the

development of the sensing systems was split into three main thrusts. The first major thrust concerned terrain characterization. For this, a combination of vision and LADAR based sensors were used to look at the slope, relative height, and regularity of texture of the terrain around the vehicle. The second major thrust was in the area of obstacle localization, specifically Moving Obstacles (MO's). Since the vehicle is intended to operate with other moving vehicles at speeds up to 30 mph, it is necessary to use sensor data with the largest available range to detect and track potential MO's. To this end, the LD-LRS1000 ladar sensors on the front fenders were relied upon to provide obstacle data out to 275m. This long-range data was supplemented by the more common close-range LMS-291 ladar sensors on the front and rear articulated bumper mounts. The last thrust of the sensor package is to characterize the road and to determine the pose of the vehicle within the lane. This process was necessarily heavily vision-based. A series of cameras located on the main sensor bridge both in the center and on each of the wings provided high frame rate images of the area in front of the vehicle and were then used to isolate the visual elements which represented the lane demarcations. From the fusion of this information with some supplemental data from the vertical-fan LADAR's it was possible to estimate the error between the vehicle's current position and the center of the lane. The following sections detail the operation of these three main areas.

3. Terrain Characterization

The process employed for characterizing terrain was fairly well defined in the previous Grand Challenges. To this end, the Urban Navigator utilizes a series of planar oriented articulated LMS-291 ladars as well as a series of statically mounted LMS-291 ladars which are oriented to give scan lines in front of the vehicle at 10 and 20 meters respectively. The point-cloud generated by the sensors is then processed into a tessellated grid representation in which

each grid element is processed to generate a relative height, slope, and best-fit plane. The data gathered by these LADAR's is then fused with vision based path-finder information. The Path-Finder is a component which completes a color and texture based analysis of the scene in the image to determine what type of terrain the vehicle is on and the relative location of other similar terrain. The result is a continuous and cohesive representation of the terrain surrounding the vehicle which is used to populate a traversability grid.

4. Moving Obstacle Detection

This area of development is probably the most critical for success in the Urban Challenge. The largest deviation from the previous competitions is the addition of manned and unmanned moving traffic which represents an entirely new and complex series of problems for unmanned navigation. The first major hurdle is identifying the moving obstacle from the surrounding terrain. This task is made more difficult by the lack of uniformity among competing vehicles both in size, shape, and operating speeds. Furthermore, the various scenarios prescribed by DARPA implied that a high degree of resolution would be required to adequately track vehicle's and provide for persistence through occlusions and other events. To this end, the MO characterization proceeded by developing LADAR based components which could sample the point-cloud of data at a given time-period and generate a cluster of data points which were linearly related. Each cluster then became a candidate for a moving obstacle. Through successive time-steps the velocity state of each candidate object was re-evaluated and corrected for the movement of the vehicle's reference frame. The result was the effective localization and tracking of every significant static and moving obstacle within sensor range. The list of such obstacles was then provided to the Local World Model component which would decide if the obstacles represented other vehicles on the road-network defined in the RNDF.

5. Vehicle Localization

The final development area is that of localizing the vehicle within the lane on a given road. To do so, a series of cameras were used to sample images from around the front and sides of the vehicle as well as the vertically oriented LADAR's mounted on the wings of the sensor bridge. The captured images are then processed to extract color and edge information and the LADAR data for curb detection. The results of each of these individual processes are then sent to an arbitration component to fuse the various corrections and generate a single set of so called 'breadcrumbs' to correct the local world model and maintain the vehicle well posed within the lane.

a) Vision-based Lane-Finder Smart Sensor (LFSS)

The LFSS is a vision based component which utilizes color processing and edge detection using the Hough Transform to isolate linear elements in images captured from the Front and Wing located cameras. These elements are then sorted on both color and spatial properties and clustered to generate a list of lane-demarcation candidates. Each subsequent candidate group is then evaluated to determine which (if any) bound the vehicle at the current time. If so, then the entities are projected into the vehicle reference frame and the relative orientation of the lane is determined at various distances from the vehicle (5, 10, 20 meters). These corrections are then sent on to the Lane Correction Arbiter for sensor fusion.

b) Vision-based Path-Finder Smart Sensor (VPFSS)

The VPFSS is another vision based system which attempts to isolate not the lane demarcations, but the boundaries of the drivable road. It accomplishes this task by using color and textural analysis based on the Expectation Maximization algorithm to estimate which areas of the image represent road or non-road. The results of the process are then projected into the

vehicle reference frame in the form of a traversability grid and directly fused with the terrain evaluation results from earlier and an estimate of the lateral offset of the vehicle from the edge of the road is generated. These results are then forwarded to the Lane Correction Arbiter for sensor fusion.

c) LADAR-based Path-Finder Smart Sensor (LPFSS)

The LPFSS is a LADAR based component which relies on the passenger-side vertically oriented LADAR. The LADAR is actuated, allowing it to rotate about an axis normal to the ground plane and is oriented such that the beam sweeps an arc normal to the ground plane. Moreover, the actuation of the LADAR is behavior based. Among other behaviors, the actuator will in normal roadway navigation attempt to orient the LADAR to track the road-boundary adjacent to the rear axle. The result of processing the LADAR scans is an estimate of the lateral offset from the rear passenger wheel to the edge of the road and serves as a good baseline for all other corrections. Again, the results of this component are forwarded to the Lane Correction Arbiter for sensor fusion.

d) Lane Correction Arbiter Smart Sensor (LCASS)

The Lane Correction Arbiter is a pseudo smart-sensor in that it does not originate any new sensor data through acquisition from sensor hardware; rather it processes other smart-sensor results into a further refined solution. The purpose of the LCASS is to filter, persist, and decay the various lane correction estimates of the other vehicle localization smart sensors into a single "best-fit" solution. To do so the arbiter relies upon an internal vehicle reference traversability grid which is maintained over time and holds the relative lane-center estimates provided by the contributing smart sensors. Each lane-center estimate is given an initial weight and added to a volatile list of estimates. Each time-step, the content of the list which contains both the relative

position of the lane-center and its current value is translated and rotated the inverse of the motion the vehicle frame. In effect, the result is a truly local grid representation of the derived sensor information. The weighted entities depicted in the grid are decayed over time based on their position, the velocity of the vehicle, and the originator of their data. The resulting grid is then used to drive a variable order curve fit ranging (up to 4th order) which attempts to minimize the variance of the fitted curve at regular intervals. Once the curve is generated, a series of vehicle referenced lane correction offsets are generated for 0m, 5m, 10m, 15m, and 25m.

V. Intelligence Element Components

Team Gator Nation has developed and deployed the Adaptive Planning Framework [19] to address the issues associated with behavior mode selection in complex or unstructured environments presented during the DARPA Urban Challenge. It enables the vehicle to intelligently select the most appropriate behavioral characteristics given the perceived operating environment. The framework is scalable to systems of varying complexity and size and is compatible with existing architectures such as JAUS RA-3.2, NIST 4D/RCS, and others. The Adaptive Planning Framework is composed of three principle elements tasked with assessing the situation, determining the suitability and viability of all possible solutions, and executing the most suitable of all recommended solutions. These three component types are multiple Situation Assessment components, multiple Behavior Specialists, and one Decision Broker component.

A. Situation Assessment Specialist Components

Dynamic environment information, originating from any array of sensors is monitored and managed by the Situation Assessment Specialists. Each specialist design is tailored to the sensor or collection of sensors whose data it will be analyzing. While the inputs to the specialist can come from any data source, the output or “finding” must adhere to specific guidelines

outlined by the framework. Findings can be in the form of conditions, state, or events. Once the findings have been generated the information is disseminated to all other components that might need it. An example of a situation assessment specialist would be a software component whose sole function was to determine if it is safe to move to the adjacent lane. This component would monitor sensor data as reported by the Moving Objects sensor and reach a Boolean conclusion that would be stored as metadata for use by other processes.

B. Behavior Specialist Components

The findings rendered by the Situation Assessment Specialists are consumed by the Behavior Specialists. There is a one-to-one mapping of each behavior with a Behavior Specialist. The role of this specialist is to monitor the findings and evaluate the suitability of its behavior under the current perceived operating conditions. An example of a behavior specialist is the Pass Left/Right behavior specialist. This specialist simultaneously monitors the desired travel lane for obstructions as well as adjacent travel lanes. Based on all the inputs the specialist recommends whether or not a lane change is an appropriate and safe option. As with the specialist findings, the default recommendation is unsuitable and must be proven appropriate at every iteration of the program to ensure truth of the results and operating safety. The Behavior Specialists do not possess the ability to activate or deactivate their associated behavior; such authority is only given to the Decision Broker.

C. Decision Broker Component

At the highest level of the framework lies the Decision Broker. Its role is to monitor all Behavior specialist recommendations. It assumes ultimate authority over how the Urban NaviGator will operate while in autonomous mode. Like the other entities within the framework the Decision Broker can base its conclusions on not only the recommendations and findings of

other specialists, but it may also look at data from any other pertinent source. Team Gator Nation's implementation of the Adaptive Planning Framework centralizes all the Decision Broker functionality within the JAUS Subsystem Commander and has the added responsibility of selecting which component receives control of the vehicle's JAUS Primitive Driver. The framework architecture employs an asynchronous, iterative, forward chaining reasoning approach to decision making.

D. Behaviors Used during the Urban Challenge

The Urban NaviGator is programmed with seven operating behavior modes where each behavior is comprised of a series of sub-behavior modes. Some sub-behaviors may be optional, depending on the mission plan or ambient conditions. Vehicle performance is denoted by a sub-behavior status indicator. A failure protocol is incorporated into each sub-behavior should sufficient environmental changes warrant the current vehicle operation inappropriate or unsafe. In most cases the vehicle is able to recover to a default safe operational state. However, in some cases, such as a catastrophic system failure or an excessively hostile environment, the safest course of action is for the vehicle to pause and wait for more favorable conditions. The corresponding behavior specialist constantly evaluates the appropriateness of its behavior mode and the decision broker determines which mode will have operation of the vehicle. The seven behavior modes are described subsequently.

1. Roadway Navigation

The Roadway Navigation behavior is the primary driving behavior deriving commands to be sent to the vehicle actuators while the objective is lane following. This behavior will allow the vehicle to navigate the roadway within the lines of its desired lane. The default sub-behavior

is to maintain a safe following distance behind any vehicles ahead. Other sub-behaviors include lane changes on a multi-lane road in order to pass through a mission goal point.

2. Open Area Navigation

Open area navigation is a behavior that should only be needed in special circumstances during the Urban Challenge event. This behavior allows the vehicle to move towards a goal location without striking any object, while avoiding any rough terrain. This is in effect the only behavior mode that was required in the 2005 DARPA Urban Challenge. It will be useful in the Urban Challenge when the vehicle is in an open area such as a parking lot or an obstacle field. The associated sub-behaviors are Enter Open Area, Exit Open Area, Enter Parking Space, and Exit Parking Space. Thus if the mission plan for the open area does not contain parking spaces the system would transition from Enter Open Area to Exit Open Area sub-behavior.

3. Pass Left and Pass Right

The Pass Left and Pass Right maneuvers will be used in passing situations where a static obstruction impedes progress in the desired lane but there exists an adjacent available travel lane. Successful Pass Left Behavior execution entails a Lane Change Left sub-behavior, Passing Vehicle sub-behavior, and Lane Change Right sub-behavior. This behavior implies a momentary lane change for obstacle avoidance.

4. Reverse Direction

This behavior is called whenever it is determined that the current lane is blocked and there is no alternate clear lane available for passing. It will also be applicable in cases where the vehicle has entered a ‘dead end’ road that it must ‘escape’ to reach a mission goal point. The default sub-behavior is to execute an N-point turn sub-behavior protocol.

5. Intersection Traversal

The intersection traversal behavior will be applicable when the vehicle enters the vicinity of an intersection. This is one of the most complicated behavior modes in that the system must rely on a series of situation assessment specialists to safely navigate the intersection. This behavior mode must handle queuing, stopping at the stop line, determining right of way, and ultimately traveling through the intersection while avoiding other vehicles. These steps are compartmentalized into five sub-behaviors: Queue to Intersection, Stop At Intersection, Queue Turn, Clear Intersection, and finally Traverse Intersection. It should be noted that if there is no stop at the intersection the sub-behavior will transition from Queue to Intersection to Queue Turn.

6. Off Road

This behavior is called when a sparse waypoint problem is identified or when the MDF indicates an unmarked or dirt road. The default sub-behavior is Defensive/Reflexive. In this sub-behavior the vehicle operates in a heightened state of cautiousness. The Subsystem Commander enforces more stringent speed limits based on inertial measurement sensor feedback, other perception algorithms are retuned for path finding as opposed to lane finding and line following, and sensor grid maps are arbitrated to give more freedom to the A-star vehicle path planner for reflexive obstacle avoidance.

7. Parking

This behavior must deal with the problems that arise in the parking lot scenario where precise motion is necessary. When the vehicle approaches the vicinity of an assigned parking space, precise path planning will be initiated to align the vehicle as required. Situation

assessment specialists monitor the near surroundings of the vehicle to center the vehicle in its parking space while avoiding any static or dynamic objects.

D. Smart Arbiter Component

The purpose of the Smart Arbiter component is to generate a 60m × 60m traversability grid, centered at the vehicle's current position, which is used to implement a desired behavior. Motion execution, which is discussed in the next section, is accomplished via an A* search through this grid to determine the least cost path. In most cases, the least cost path will be obvious as the grid has been constructed to accomplish a desired action. An important feature of this entire approach is that specific behavior modes can be changed with smooth continual control of the vehicle.

The Smart Arbiter obtains inputs from the Terrain Smart Sensor, the Lane Finding Smart Sensor, the Path Finding Smart Sensor, and the Local World Model and builds its grid based on the current behavior mode of the system. For example, if the system is in the Roadway Navigation behavior, then the grid cells corresponding to the positions of the line on the edge of the lane as identified by the Lane Finding Smart Sensor will be marked as non-traversable regions in the Smart Arbiter grid. The cells corresponding to the road lane will be marked as highly traversable. This will prevent the planner from planning outside the current lane.

VI. Control Element Components

A. Receding Horizon Controller Component

Low level path planning on the Urban NaviGator is carried out for the basic road following, intersection, passing, and open area behaviors using an A* search algorithm in the Receding Horizon Controller component. This makes use of a modified form of Model Predictive Control [20-21]. This procedure expands a state-space search tree through the

arbitrated traversability grid by means of a vehicle kinematic model. The objective is to find the optimal trajectory through the grid to a goal point which minimizes the cost function $J = \Phi(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt$. The goal points are calculated from a list of path segments constructed by the High Level Planner and Local World Model. The algorithm is modified by feeding the vehicle position forward in time to account for a small system lag, and the first generation of the search is expanded such that the previous steering command is always included as the central node. Once the optimal path is found, the trajectory is traced back to the first state change to find the initial steering command to be sent to the actuator. The output steering command is in the form of a positive or negative propulsive rotational effort. A special feature of the planner is the ability to plan forward while the vehicle transmission is in drive, and backward while in reverse using the inverted vehicle kinematics. This is utilized when backing out of a parking space or backing away from a roadblock. Closed-loop steering control is accomplished by repeating the algorithm at 40 Hz with position and actuation feedback.

Figure 10 shows a traversability grid that was generated by the Smart Arbiter component during Roadway Navigation behavior. The vehicle position is at the center of the figure and the vehicle is heading towards the upper right corner. Low cost cells which identify the lane are colored black, traversable higher cost cells are colored orange, and non-traversable cells are colored red. The short black lines show paths that were considered while the brown line identifies the path that was selected.

The speed of the Urban NaviGator is maintained using a P.I.D. controller. The controller selects the desired speed as the lowest of recommendations coming from the path segment list based on RNDF and MDF speed limits, from the Subsystem Commander based on the current behavior and perceived driving conditions, and from the Local World Model based on proximity

to intersections and moving or static obstacles in the lane. The controller can further limit the speed based on path curvature and traversability of the chosen path. A speed command, s, is then calculated using appropriate acceleration or deceleration values specific to the Urban NaviGator, and is passed to the controller. The output of the controller is in the form of a propulsive or resistive linear effort represented by

$$linear\_effort = K_{FF}s + Bias_{FF} + K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt} \tag{1}$$

where $e(t)$ is the error between the desired speed and the actual speed. Again, the control loop is closed by repeating the algorithm at 40 Hz using feedback from the Velocity State Sensor. The rotational and linear efforts are then sent to the Primitive Driver component, which converts the efforts to hardware commands and passes them to the actuators.

B. Primitive Driver Component

The Primitive Driver component (P.D.) closes the loop between the Receding Horizon Controller and the vehicle. The wrench commands and shifter commands generated by the local path planner and the P.I.D. speed controller are converted from their native values to hardware specific values. The steering and shifter commands are converted to absolute motor positions for the motors connected to the steering column and shifting mechanism. The motors have an internal P.I.D. controller which is tuned to accurately achieve required positions in a timely manner. Throttle and brake wrench commands are converted to voltages, which are fed into the Urban NaviGator's pre-existing throttle and brake drive-by-wire Electronic Control Units (E.C.U.'s). The P.D. is also responsible for monitoring the states of all the motors and kill-switch. Lastly, the P.D. collects feedback from the motors and E.C.U.'s, and reports the implemented control back to the Receding Horizon Controller.

VII. Results and Lessons Learned

The performance of the implemented architecture at the DARPA Urban Challenge was in most part satisfactory, but less than desired with respect to certain scenarios. The system performed most subtasks well, but failed to fully realize the potential of the design. The qualification event was comprised of missions planned on three courses. The vehicle ran on all three courses with some success.

The Adaptive Planning Framework correctly managed the system's behavior with respect to the sensed scenario. Low level control of the vehicle was maintained during imposed behaviors by the architecture, leading to smooth continuous driving behavior.

Course A exposed a deficiency in the persistence of moving objects in the implementation. This course simulated a two way traffic circle that the autonomous vehicle had to merge into and out of. Sometimes traffic vehicles became occluded by others, leading in these cases to the autonomous vehicle incorrectly determining it had right of way and could proceed. An attempt to tune this deficiency's effect down was not successful, mostly due to the lack of testing.

Course B exposed an error in the search methodology for the open area behavior. This test involved navigating through a large open area to a road network to complete a mission. The search space the algorithm considered for the open area was uniform in costs related to traversability, leading to an ill conditioned optimization problem. This situation was not handled, and the component controlling the vehicle became non responsive.

Courses A and C exposed a "ground strike" problem with moving object detection. Course C was designed to test intersection precedence and re-planning. In both A and C, ground strikes from the LADAR sensors were detected as fixed objects that had to be considered in the

intersection and roadway navigation behaviors. These false detections lead to less than desirable behaviors for the scenarios encountered.

The Lane Correction Arbiter Smart Sensor (LCASS) concept provided accurate information concerning lane center relative to the vehicle location. The utilization of this information in a grid resolution of 0.5 m proved to be problematic. Typical lane widths encountered at the DUC site were 3+ m to 4 m. Typical lane center corrections were often smaller than the grid resolution, leading to situations where the vehicle left the lane due to the lack of precision in lane representation. These problems were observed in roadway navigation on lanes less than 5 m.

Most failures observed involved component implementation errors. The overall architecture worked as designed, given the performance of the components. The implementation does not have significant simulation capacities. Testing was performed with the system deployed in a suitable environment. The development of hardware in the loop simulation for the system could have allowed many of the shortcomings of the implementations to be identified and fixed in a shorter time.

VIII. Conclusion

The performance requirements identified in the Urban Challenge Technical Evaluation Criteria were challenging. The system had to be able to detect and model its environment and then plan and execute appropriate actions in real time.

The approach described in this paper was generated after careful consideration of the design requirements. The central concept is the integration of a priori and sensed information in a raster format in the Local World Model. Based on this information, an appropriate behavior is

selected via arbitration. The behavior is executed by generation of a navigation grid coupled with metadata.

The primary new contribution of this approach is that related to solving the technical challenges of (a) the determination of the appropriate behavior mode, and (b) the smooth transition of vehicle control between behavior modes.

IX. Acknowledgments

The members of Team Gator Nation would like to gratefully acknowledge the sponsorship and support of GE Aviation, the Air Force Research Laboratory at Tyndall Air Force Base, and the University of Florida.

X. VI. References

[1]DARPA Urban Challenge Technical Evaluation Criteria, 16 Mar 2006.

[2]Urmson, C., et al., "A Robust Approach to High-Speed Navigation for Unrehearsed Dessert Terrain," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[3]Trepagnier, P., et al., "KAT-5: Robust Systems for Autonomous Vehicle Navigation in Challenging and Unknown Terrain," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[4]Mason, R., "The Golum Group/ University of California at Los Angeles Autonomous Ground Vehicle in the DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[5]Lakhotia, A., et al., "CajunBot: Architecture and Algorithms," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[6]Travis, W., et al., "SciAutonics-Auburn Engineering's Low-Cost High-Speed ATV for the 2005 DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[7]Miller, I., et al., "Cornell University's 2005 DARPA Grand Challenge Entry," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[8]Crane, C., et al., "Team CIMAR's NaviGATOR: An Unmanned Ground Vehicle for the 2005 DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, Issue 8, Aug 2006.

[9]Touchton, R., Galluzzo, T., Kent, D., and Crane, C., "Perception and Planning Architecture for Autonomous Ground Vehicles," *IEEE Computer*, Dec. 2006, pp. 40 – 47.

[10]Coppin, Ben, *Artificial Intelligence Illuminated*, Jones and Bartlett Publishers, Sudbury, MA, 2004, ISBN 0-7637-3230-3, pp. 108-109.

[11]Touchton, R., et. al, "Planning and modeling extensions to the Joint Architecture for Unmanned Systems (JAUS) for application to unmanned ground vehicles," *SPIE Defense and Security Symposium*, Orlando, Mar 2005.

[12]Solanki, S., "Development of Sensor Component for Terrain Evaluation and Obstacle Detection for an Unmanned Off-Road Autonomous Vehicle," Ph.D. Disertation, Univ. of Florida, 2007.

[13]Nanning, Q., and Cheng, H.. "Springbot: a Prototype Autonomous Vehicle and Its Algorithms for Lane Detection." *IEEE Trans. on Intelligent Transportation Systems 5*, (2004).

[14]Odagiri, K., Kobayashi, K., Watanabe, K., Umino, H., and Nakaho, N., "Development of Active Contour Extraction for Autonomous Vehicle Lane Detection at Outdoor Environment." Society of Automotive Engineers, 2001.

[15]Matthews, John H., *Numerical Methods for Computer Science*, Engineering and Math, Englewood Cliffs, New Jersey: Prentice-Hall, 1987, 224-252.

[16]Lee, J., Crane, C., Kim, S., and Kim, J., "Road Following in an Unstructured Desert Environment using Monocular Color Vision as Applied to the DARPA Grand Challenge," *Proceedings of the International Conference on Control, Automation, and Systems (ICCAS 2005)*, Gyeong Gi, Korea, June 2005

[17]Gonzalez, R.C. and Woods, R.E., *Digital Image Processing*, Prentice Hall, 2002.

[18]Viola and Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*. 2001.

[19]Touchton, R. "An Adaptive Planning Framework for Situation Assessment and Decision-Making on an Autonomous Ground Vehicle," Ph.D. Dissertation, University of Florida, 2006.

[20]Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality," *Automatica*, Volume 36, Issue 6, June 2000, pp. 789-814.

[21]Scokaert, P.O.M, Mayne, D.Q., Rawlings, J.B., "Suboptimal Model Predictive Control (Feasibility Implies Stability)," *IEEE Transactions on Automatic Control*, Volume 44, Issue 3, March 1999, pp. 648-654.

[22]Hart, P.E., Nilsson, N.J., Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, SSC4 (2), (1968), pp. 100-107.

Figure 1: Team Gator Nation NaviGATOR

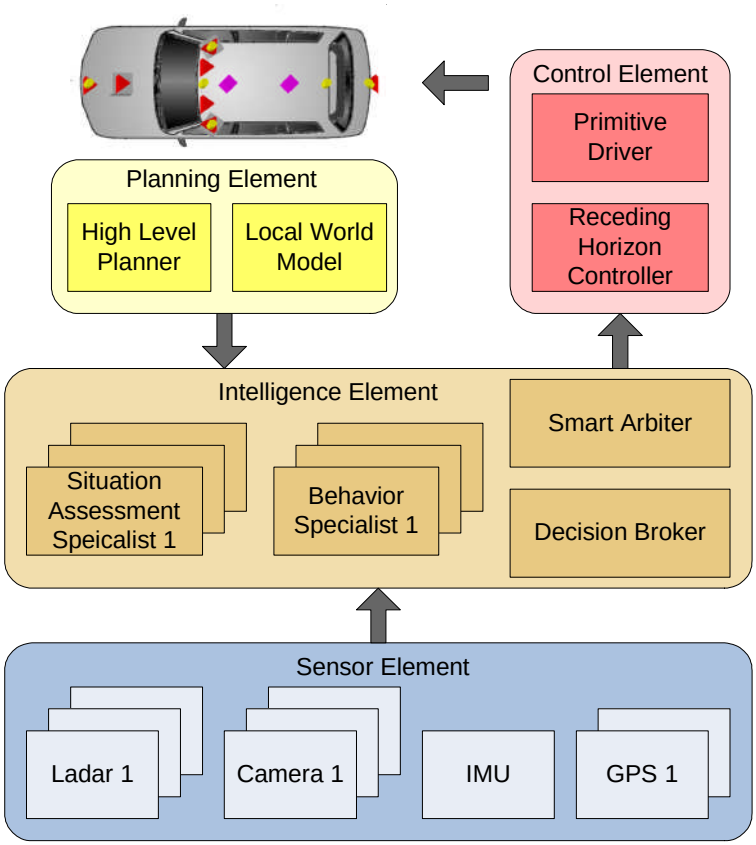
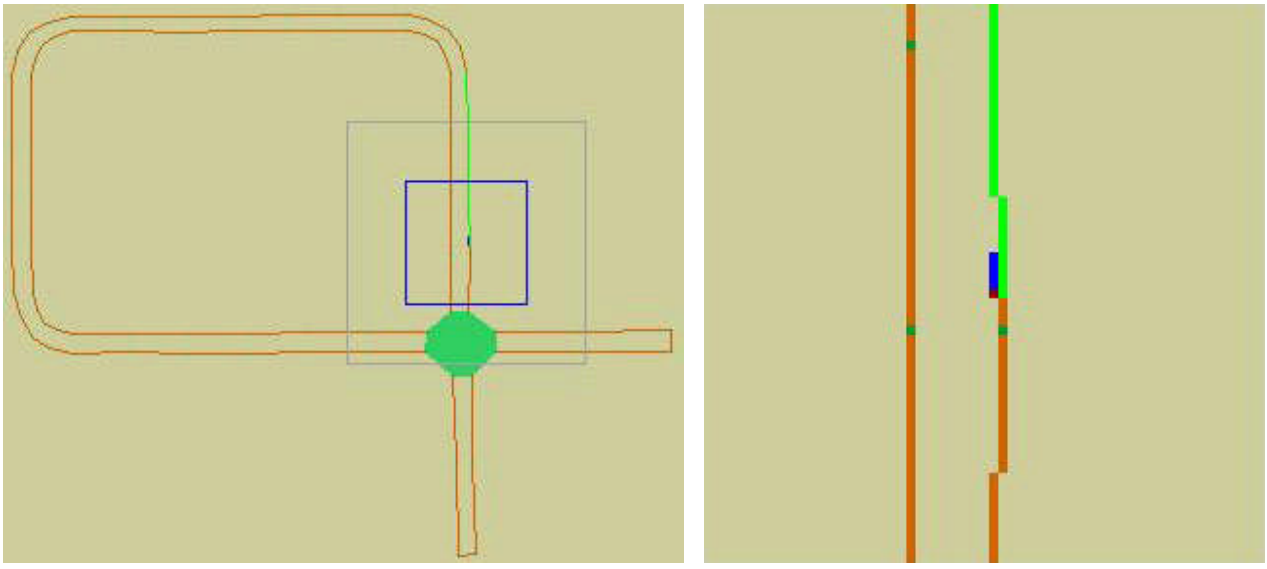
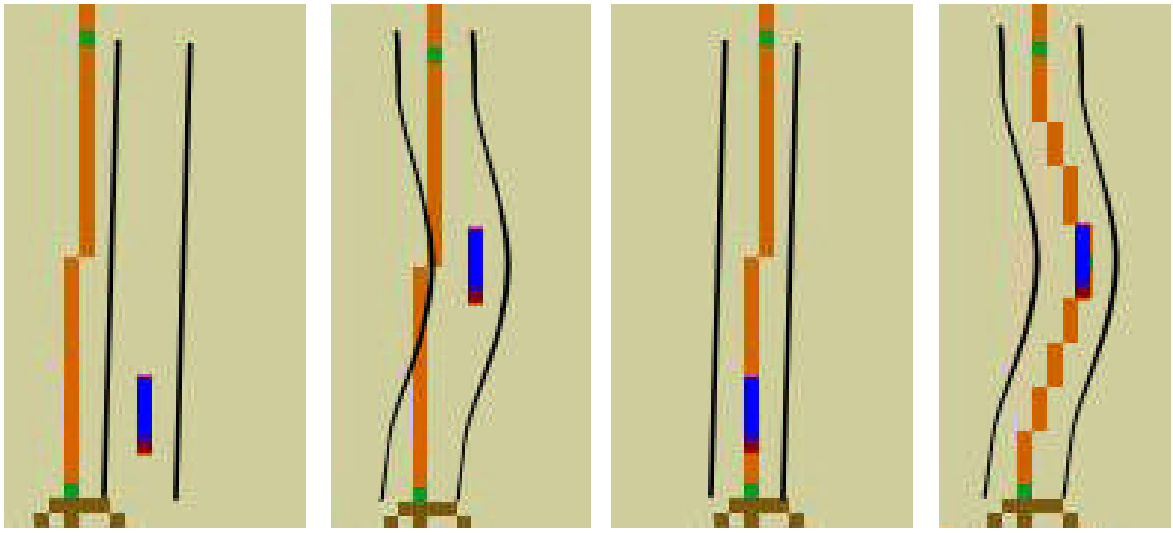


Figure 2: System Architecture



(a) (b)

Figure 3: (a) 300m × 300m Raster Local World Model ;
(b) Sub-sampled 60m × 60m Grid.



(a) (b) (c) (d)

Figure 4: Arbitration of Discrepancy in GPOS Data,
RNDF Data, and Sensed Data

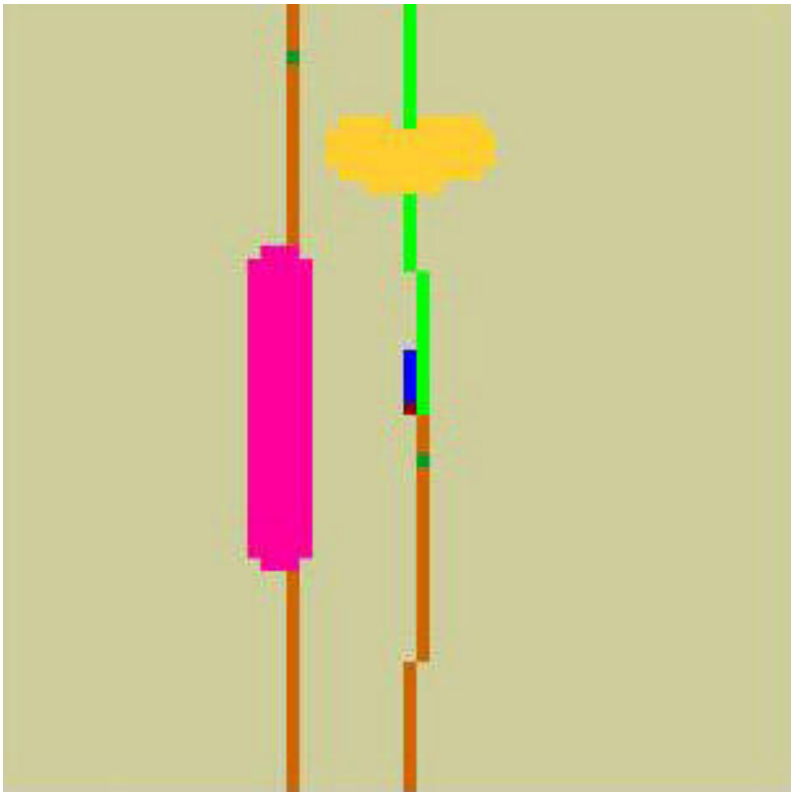
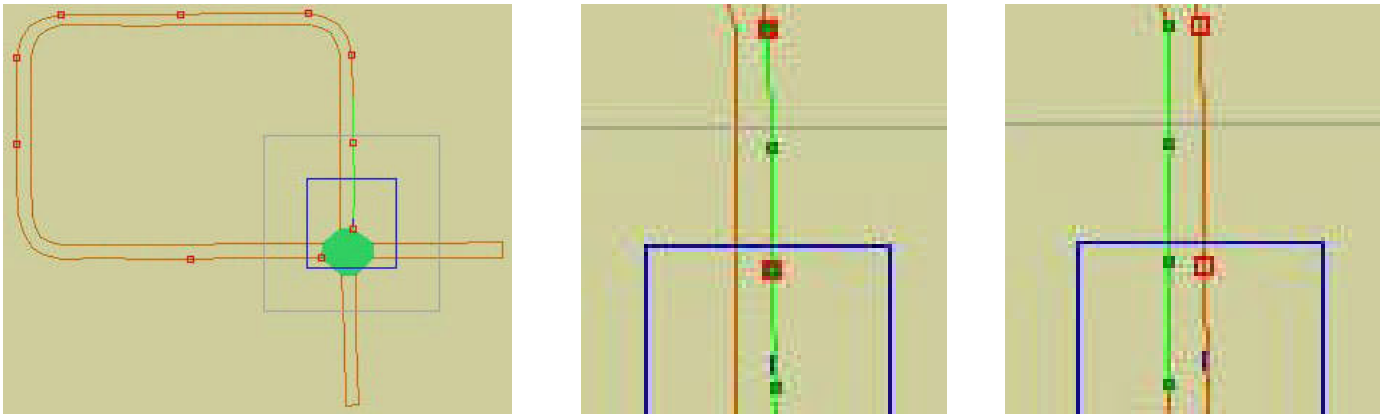


Figure 5: Moving Obstacle Depicted in Local World Model



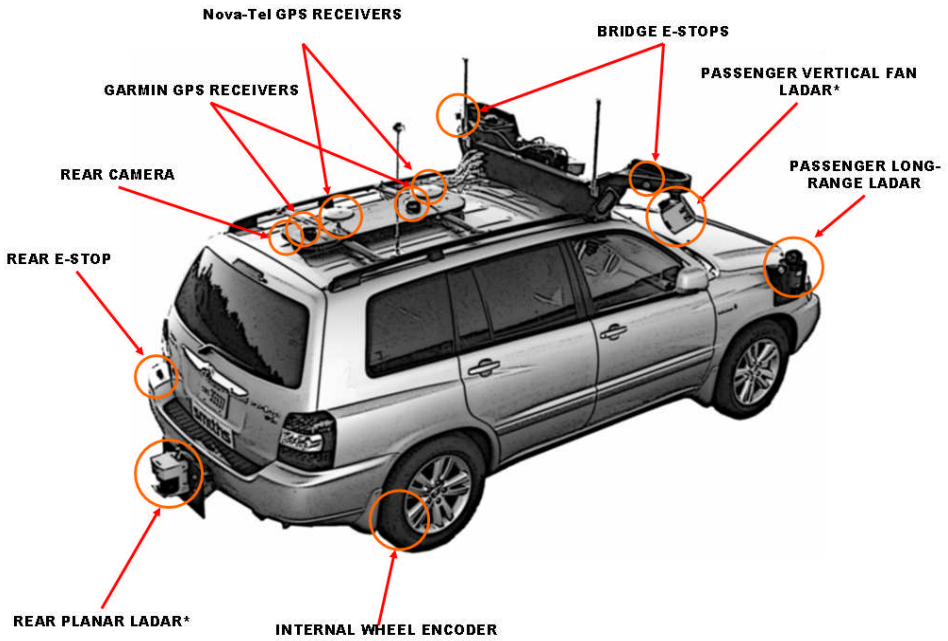
(a) (b) (c)

Figure 6: Dynamic Update of Intermediate Points

(a) (b)

Figure 7: (a) GPOS Repeatability Data (b) Magnified View

(a)



(b)

Figure 8: Sensors

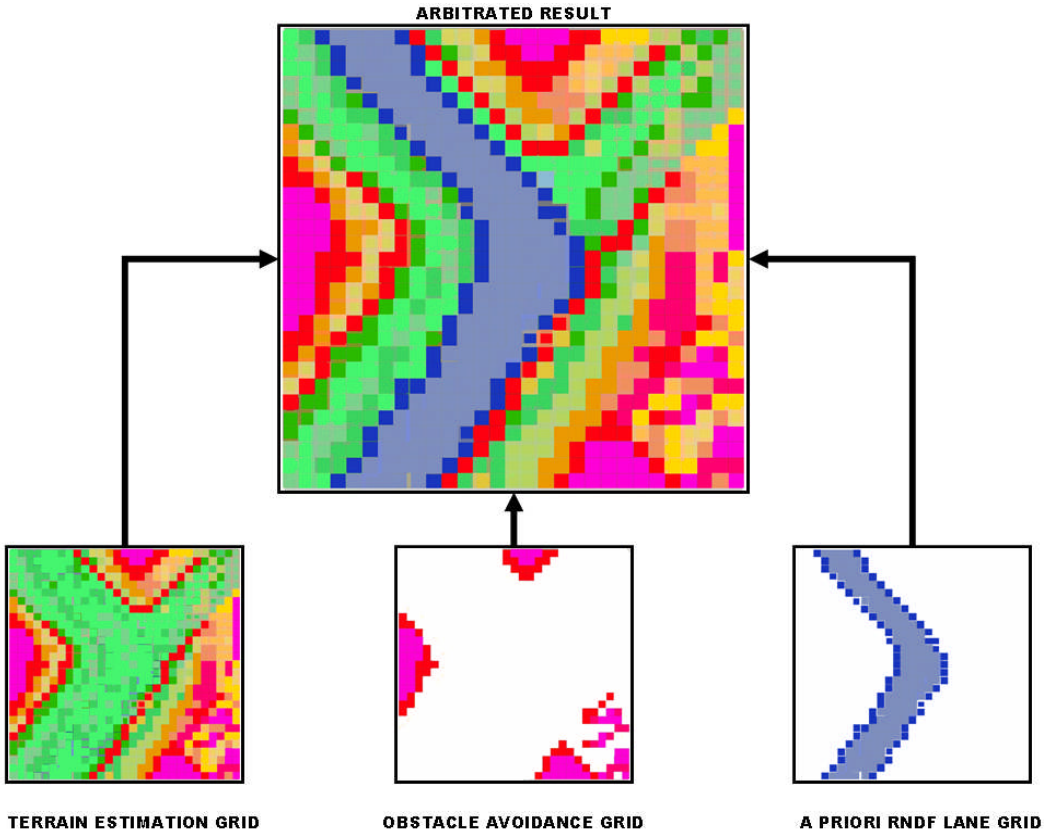


Figure 9: Smart Sensor Traversability Grid Fusion

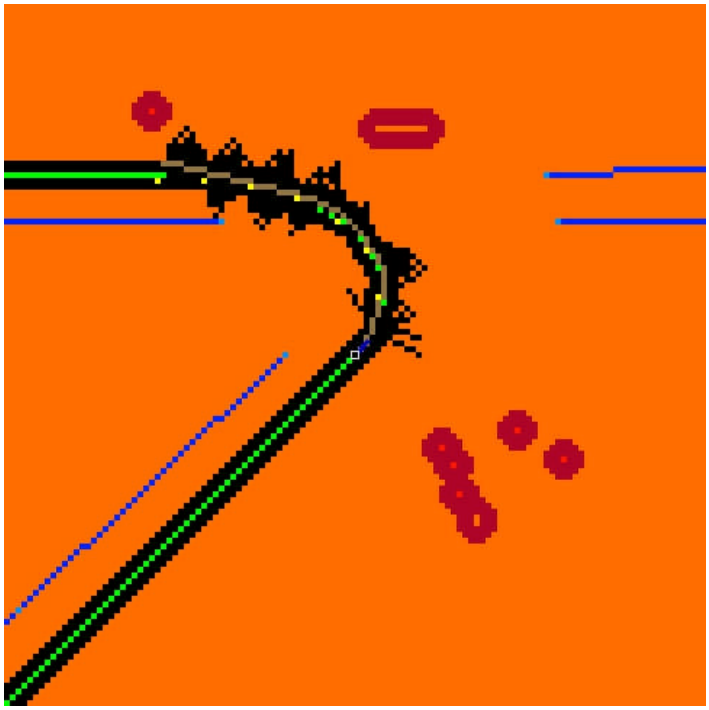


Figure 10: Path Search Algorithm