TerraMax™: Team Oshkosh Urban Robot

Yi-Liang Chen, Venkataraman Sundareswaran, and Craig Anderson
Teledyne Scientific & Imaging
Thousand Oaks, California 91360
e-mail: ylchen@teledyne.com, sundar@teledyne.com, canderson@teledyne.com

Alberto Broggi, Paolo Grisleri, Pier Paolo Porta, and Paolo Zani
VisLab, University of Parma
Parma 43100, Italy
e-mail: broggi@ce.unipr.it, grisleri@ce.unipr.it, portap@ce.unipr.it, zani@ce.unipr.it

John Beck
Oshkosh Corporation
Oshkosh, Wisconsin 54903
e-mail: jbeck@oshkoshcorp.com

Received 22 February 2008; accepted 28 August 2008

Team Oshkosh, composed of Oshkosh Corporation, Teledyne Scientific and Imaging Company, VisLab of the University of Parma, Ibeo Automotive Sensor GmbH, and Auburn University, participated in the DARPA Urban Challenge and was one of the 11 teams selected to compete in the final event. Through development, testing, and participation in the official events, we experimented and demonstrated autonomous truck operations in (controlled) urban streets of California, Wisconsin, and Michigan under various climate and traffic conditions. In these experiments TerraMax™, a modified medium tactical vehicle replacement (MTVR) truck by Oshkosh Corporation, negotiated urban roads, intersections, and parking lots and interacted with manned and unmanned traffic while observing traffic rules. We accumulated valuable experience and lessons on autonomous truck operations in urban environments, particularly in the aspects of vehicle control, perception, mission planning, and autonomous behaviors, which will have an impact on the further development of large-footprint autonomous ground vehicles for the military.

Journal of Field Robotics 25(10), 841–860 (2008)
Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.20267

In this article, we describe the vehicle, the overall system architecture, the sensors and sensor processing, the mission planning system, and the autonomous behavioral controls implemented on TerraMax. We discuss the performance of some notable autonomous behaviors of TerraMax and our experience in implementing these behaviors and present results of the Urban Challenge National Qualification Event tests and the Urban Challenge Final Event. We conclude with a discussion of lessons learned from all of the above experience in working with a large robotic truck.

1. INTRODUCTION

Team Oshkosh entered the DARPA Urban Challenge with a large-footprint robotic vehicle, TerraMax™, a modified medium tactical vehicle replacement (MTVR) truck. By leveraging our past experience and success in previous DARPA Challenges, the combined multifaceted expertise of the team members, and the support of a DARPA Track A program award, we demonstrated various autonomous vehicle behaviors in urban environments with excellent performance, passed through many official tests at the National Qualification Event (NQE), and qualified for the Urban Challenge Final Event (UCFE). TerraMax completed the first four submissions in mission 1 of the UCFE before being stopped after a failure in the parking lot due to a software bug. We brought TerraMax to the UCFE test site in Victorville in December 2007, where TerraMax completed successfully three missions totaling more than 78 miles in 7 h and 41 min.

Team Oshkosh is composed of Oshkosh Corporation, Teledyne Scientific and Imaging Company, VisLab of the University of Parma, Ibeo Automotive Sensor GmbH, and Auburn University. Oshkosh provided the vehicle, program management, and overall design direction for the hardware, software, and control systems. Oshkosh integrated all the electrical and mechanical components and developed the low- and midlevel vehicle control algorithms and software. Teledyne Scientific and Imaging Company developed the system architecture, mission and trajectory planning, and autonomous behavior generation and supervision. The University of Parma's VisLab developed various vision capabilities. Ibeo Automotive Sensor GmbH provided software integration of the LIDAR system. Auburn University provided evaluation of the global positioning system/inertial measurement unit (GPS/IMU) package.

Although substantial hurdles must be overcome in working with large vehicles such as TerraMax, we feel that large autonomous vehicles are critical for enabling autonomy in military logistics operations. Team Oshkosh utilized a vehicle based on the U.S. Marine Corps MTVR, which provides the majority of the logistics support for the Marine Corps. The intention is to optimize the autonomous system design such that the autonomy capability can be supplied in kit form. All design and program decisions were made considering not only the Urban Challenge requirements, but eventual fielding objectives as well.

Our vehicle was modified to optimize the control-by-wire systems in providing a superior low-level control performance based on lessons learned from the 2005 DARPA Grand Challenge (Braid, Broggi, & Schmiedel, 2006; Sundareswaran, Johnson, & Braid, 2006). Supported by a suite of carefully selected and military-practical sensors and perception processing algorithms, our hierarchical state-based behavior engine provided a simple yet effective approach in generating the autonomous behaviors for urban operations. Through the development, testing, and participation in official events, we have experimented and demonstrated autonomous truck operations in (controlled) urban streets of California, Wisconsin, and Michigan under various climate conditions. In these experiments, TerraMax negotiated urban roads, intersections, and parking lots and interacted with manned and unmanned traffic while observing traffic rules.

In this article, we present our experience and lessons learned from autonomous truck operations in urban environments. In Section 2 we summarize the vehicle and hardware implementation. In Section 3 we present the overall system architecture and its modules. In Section 4 we describe TerraMax's sensor and perception processing. In Section 5 we present TerraMax's autonomous behavior generation and supervision approach. In Section 6 we discuss TerraMax's field performance and experience in the NQE and the UCFE. We comment on lessons learned in Section 7.

Figure 1. TerraMax: the vehicle.

2. TERRAMAX: THE VEHICLE AND HARDWARE SYSTEMS

2.1. Vehicle Overview

The TerraMax vehicle (see Figure 1) is a modified version of a standard Oshkosh MTVR Navy tractor,[1] which comes with a rear steering system as standard equipment. The MTVR platform was designed for and combat tested by the U.S. Marine Corps. We converted the vehicle to a 4×4 version by removing the third axle and by shortening the frame rail and rear cargo bed. The TAK-4™ independent suspension allowed rear axle steering angles to be further enhanced to deliver curb-to-curb turning diameters of 42 ft, equivalent to the turning diameter of a sport utility vehicle. In addition to the enhancement of turning performance, Oshkosh developed and installed low-level controllers and actuators for "by-wire" braking, steering, and powertrain control. Commercial-off-the-shelf (COTS) computer hardware was selected and installed for the vision system and autonomous vehicle behavior functions.

[1]Oshkosh MTVR. http://www.oshkoshdefense.com/pdf/Oshkosh_MTVR_brochure_07.pdf

2.2. Computing Hardware

We opted for ruggedized COTS computing platforms to address the computing needs of TerraMax. Two A-Plus Mobile A20-MC computers with Intel Core Duo processors running Windows XP Pro were used for autonomous vehicle behavior generation and control. The four Vision personal computers (PCs) use SmallPC Core Duo computers running Linux Fedora. One PC is dedicated to each vision camera system (i.e., trinocular, close-range stereo, rearview, and lateral). Low-level vehicle controller and body controller modules are customized Oshkosh Command Zone® embedded controllers and use the 68332 and HC12X processors, respectively. To meet our objectives of eventual fielding, all the computing hardware was housed in the storage space beneath the passenger seat.

2.3. Sensor Hardware

2.3.1. LIDAR Hardware

TerraMax incorporated a LIDAR system from Ibeo Automobile Sensor, GmbH, that provides a 360-deg field of view with safety overlaps (see Figure 2). Two ALASCA XT laser scanners are positioned on

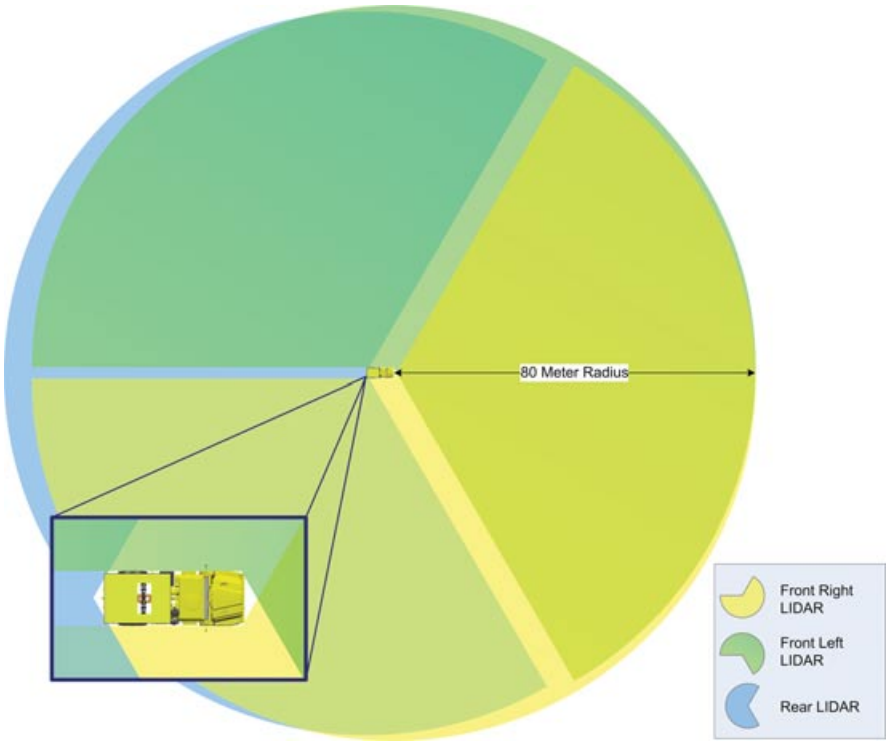

Figure 2. LIDAR coverage of TerraMax (truck facing right).

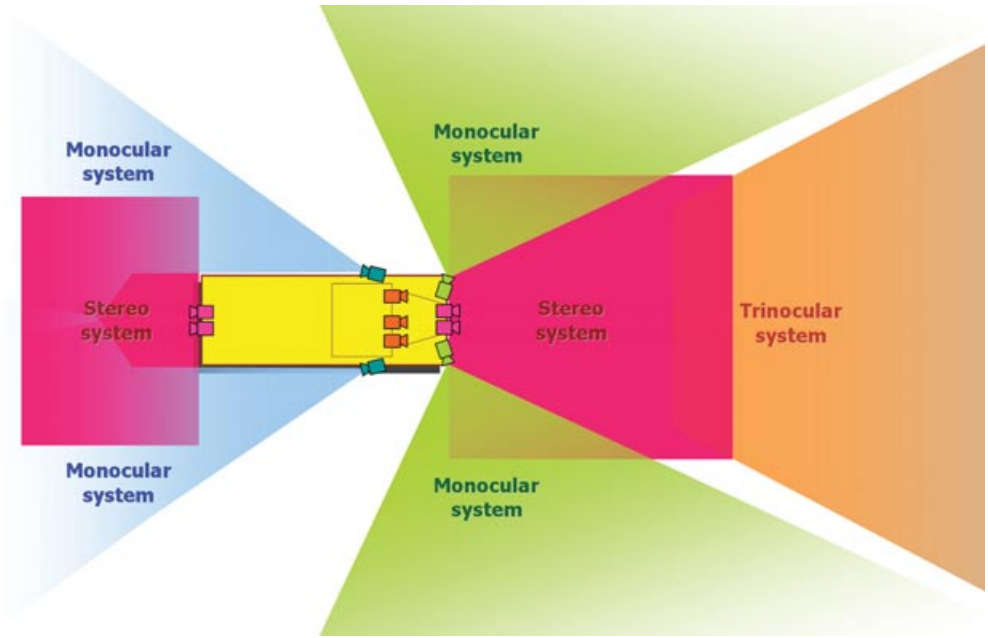

Figure 3. Vision coverage of TerraMax (truck facing right). Systems displayed: Trinocular (orange) looking forward from 7 to 40 m, stereo front and stereo back (purple) monitoring a 10 × 10 m area on the front of the truck and 7 × 5 m in the back, Rearview (blue) monitoring up to 50 m behind the truck, and lateral (green) looking up to 130 m.

the front corners of the vehicle, and one ALASCA XT laser scanner is positioned in the center of the rear. Each laser scanner scans a 220-deg horizontal field. Outputs of the front scanners are fused at the low level; the rear system remained a separate system. The LIDAR system native software was modified to operate with our system architecture messaging schema.

2.3.2. Vision Hardware

Four vision systems are onboard: trinocular, stereo, rearview, and lateral. Figure 3 depicts the coverage of these vision systems. Table I summarizes the functions and components of these vision systems.

Each vision system is formed by a computer connected to a number of cameras and laser scanners, depending on the application. Each computer is connected through an 800-Mbps, FireWire B link to a subset of the 11 cameras [9 PointGrey Flea 2, sensor: charge-coupled device (CCD), 1/3″, Bayer pattern, 1,024 × 768 (XGA); and 2 Allied Vision Technologies Pike 2, sensor: 1″, Bayer pattern, 1,920 × 1,080 pixels high-definition TV (HDTV)] mounted on the truck, depending on the system purpose.

2.3.3. GPS/INS

Using a Novatel GPS receiver with Omnistar HP corrections (which provides 10-cm accuracy in 95% of cases) as a truth measurement in extensive tests under normal and GPS-denied conditions, we selected Smiths Aerospace Inertial Reference Unit (IRU) as our GPS/inertial navigation systems (INS) solution based on its more robust initialization performance and greater accuracy in GPS-denied conditions.

3. SYSTEM ARCHITECTURE

On the basis of a layered architecture design pattern (Team Oshkosh, 2007), we designed the software modules as services that provide specific functions to the overall system. These services interact with each other through a set of well-defined asynchronous messages. Figure 4 illustrates these software modules and the relations among them. As illustrated, there are two main types of services: autonomous services, whose modules provide functionalities for autonomous vehicle behaviors, and system services, whose modules support the reliable operations of the vehicle and the mission. We summarize the main functionalities of these software modules in the following description.

Table I. Vision system components.

Vision system	Trino	Stereo	Lateral	Rearview
Cameras	3x PtGrey Flea2 (XGA)	4x PtGrey Flea2 (XGA)	2x Allied Vision Technologies Pike 2 (HDTV)	2x PtGrey Flea2 (XGA)
Camera position	Upper part of the windshield, inside the cab	2 on the front camera bar, two on the back of the truck, all looking downward	On the sides of the front camera bar	External, on top of the cab, looking backward, and downward, rotated by 90 deg
Linked laser scanner	Front	Front, back	Not used	Back
Algorithms	Lane detection, stereo obstacle detection	Lane detection, stop line detection, curb detection, short-range stereo obstacle detection	Monocular obstacle detection	Monocular obstacle detection
Range	7 to 40 m	0 to 10 m	10 to 130 m	−4 to −50 m
Notes	3 stereo systems with baselines: 1.156, 0.572, 1.728 m	2 stereo systems (front and rear)	Enabled when the truck stops at crossings	Overtaking vehicles detection

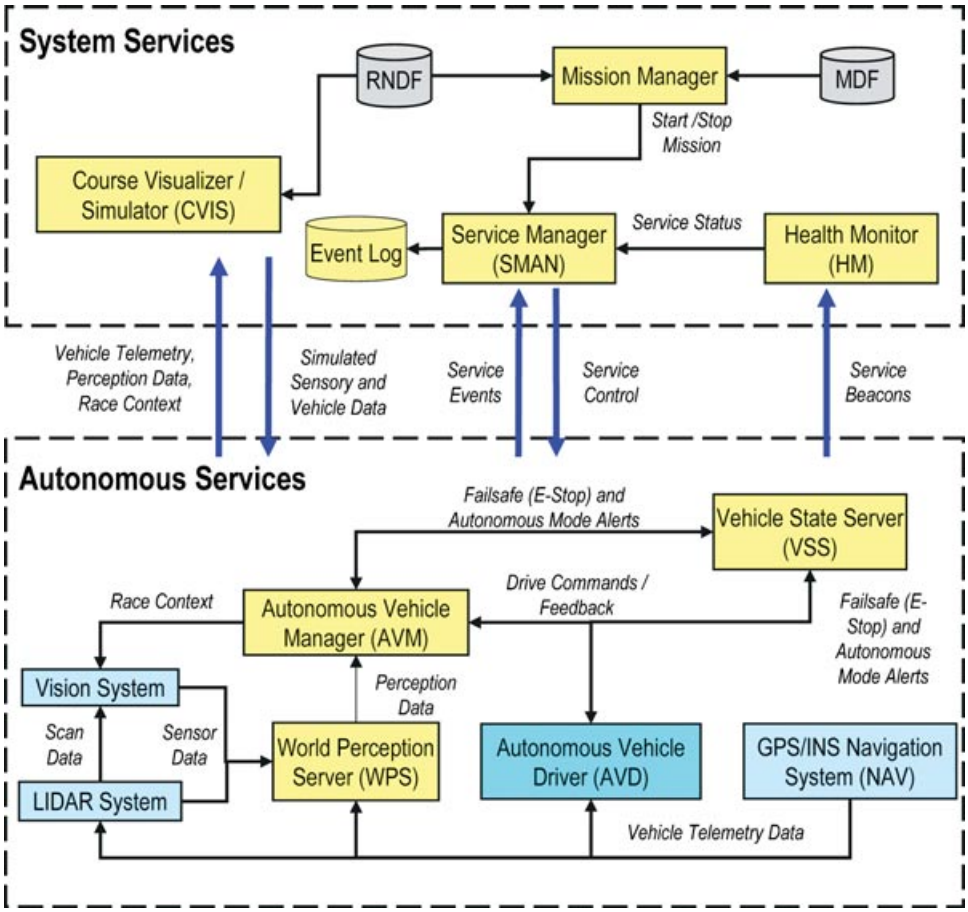

Figure 4. Software deployment architecture.

3.1. Autonomous Services

Autonomous vehicle manager. The autonomous vehicle manager (AVM) manages the high-level autonomous operation of the vehicle. It is primarily responsible for performing route planning, trajectory planning, and behavior management. The AVM receives perception updates from the world perception server (WPS) and uses this information to track the current vehicle state and determine the current behavior mode. The AVM continuously monitors perceived obstacle and lane boundary information and issues revised trajectory plans to the autonomous vehicle driver (AVD) through drive commands.

Autonomous vehicle driver. The AVD provides vehicle-level autonomy, such as waypoint following and lateral, longitudinal, and stability control by accepting messages from the AVM and commanding the lower level control-by-wire actuations.

World perception server. The WPS publishes perception updates containing the most recently observed vehicle telemetry, obstacle, and lane/road boundary information. The WPS subscribes to sensory data from the LIDAR and the vision system (VISION). The WPS combines the sensory data with the vehicle telemetry data received from the navigation service (NAV). Obstacles detected by the LIDAR and VISION are further fused in order to provide a more accurate depiction of the sensed surroundings. The AVM consumes the perception updates published by the WPS and uses this information to determine the next course of action for the vehicle.

Vision system. VISION publishes processed sensory data and metadata from different groups of

cameras. The metadata may contain information such as detected driving lane/path, lane boundary and curb marking, and/or obstacles. These sensory data and metadata are sent to WPS for distribution.

Other autonomous services include the vehicle state server (VSS), which monitors and manages low-level control for transitions from manual to autonomous operations, detects any low-level faults, and attempts to recover the system into fail-safe mode, if needed; the LIDAR system (LIDAR), which fuses and publishes obstacle information provided by the native obstacle detection and tracking functionalities from different laser scanners; and the NAV service that manages communications to the GPS/INS.

3.2. System Services

Course visualizer. The course visualizer is the prime interface to allow human operators/developers to observe the internal operations and status of the autonomous systems during a run. During an autonomous run, it provides real-time, two-dimensional visualization of the course data (i.e., road network Definition File; RNDF), vehicle telemetry data, metadata from sensors (e.g., lane updates and obstacles), and status/results of autonomous behavior generation (e.g., internal logics of a particular autonomous mode, results of a particular behavior algorithm). It can also serve as the main playback platform to enable postoperation analysis of the data log. Incorporated with a simplistic vehicle model, it also serves as a rough simulation platform to allow early testing and verification for developing or adjusting behavioral algorithms.

Other system services include the mission manager, which provides the user interface for configuring autonomous services, loading mission files, and starting the autonomous services/modes; the health monitor, which monitors service beacons from other services and alerts the service manager if an anomaly occurs; and the service manager, which manages the initialization, startup, restart, and shutdown of all autonomous services.

4. SENSOR PROCESSING AND PERCEPTION

In previous Grand Challenge efforts we used a trinocular vision system developed by VisLab at the University of Parma for both obstacle and path detection, coupled with laser scanning systems for obstacle detection. We used several SICK laser scanners and one Ibeo laser scanner for obstacle detection. The Grand Challenge generally involved only static obstacles, so sensing capabilities focused on the front of the vehicle. Urban driving introduces a new dynamic—obstacles move (i.e., other moving vehicles), and the vehicle must respond to these moving obstacles, resulting in a much greater need for sensing behind and to the sides of the vehicle. The AVM needs more information about the obstacles, requiring their velocity as well as their location, and it also needs great precision in detecting stop lines and lane boundaries. To meet these goals, we enhanced capabilities in both laser scanning and vision.

Ibeo provided three of their advanced ALASCA XT laser scanners and fusion algorithms for an integrated 360-deg view. The new TerraMax vehicle utilizes multiple vision systems, with perception capabilities in all the critical regions.

4.1. LIDAR

The Ibeo laser scanners have two roles on TerraMax. First, they provide processed object data (Kaempchen, Bühler, & Dietmayer, 2005; Wender, Weiss, Dietmayer, & Fuerstenberg, 2006) to the WPS. Second, they provide scan-level data used by the vision system to improve its results. The onboard external control units (ECUs) fuse the data from the two front LIDARs[2] acting as a single virtual sensor in the front.

4.2. Vision System

4.2.1. Software Approach

All the vision computers run the same software framework (Bertozzi et. al., 2008), and the various applications are implemented as separate plug-ins. This architecture allows hardware abstraction, while making a common processing library available to the applications, thus making algorithm development independent of the underlying system. Each vision system controls its cameras using a selective autoexposure feature. Analysis of each image is focused on a specific region of interest.

All the vision systems are fault tolerant with respect to one or more, temporary or permanent, sensor

[2]Ibeo laser scanner fusion system. http://www.ibeo-as.com/english/technology_d_fusion.asp

failure events. The software is able to cope with FireWire bus resets or laser scanner communication problems and to reconfigure itself to manage the remaining sensors.

4.2.2. The Trinocular System

Driving in urban traffic requires detailed perception of the environment surrounding the vehicle: for this, we installed in the truck cabin a trinocular vision system capable of performing both obstacle and lane detection up to distances of 40 m, derived from Caraffi, Cattani, and Grisleri (2007). The stereo approach has been chosen because it allows an accurate three-dimensional (3D) reconstruction without requiring strong a priori knowledge of the scene in front of the vehicle, but just correct calibration values, which are being estimated at run time.

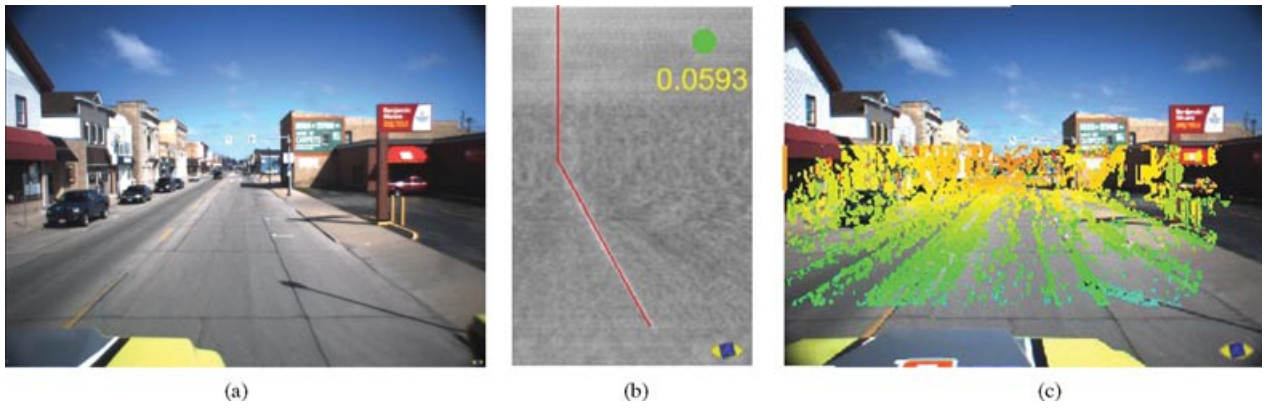

Figure 5. A frame captured from the right camera; (b) corresponding V-disparity map, where the current pitch is shown in yellow text and the detected ground slope is represented by the slanted part of the red line; and (c) disparity map (green points are closer; orange ones are farther away).

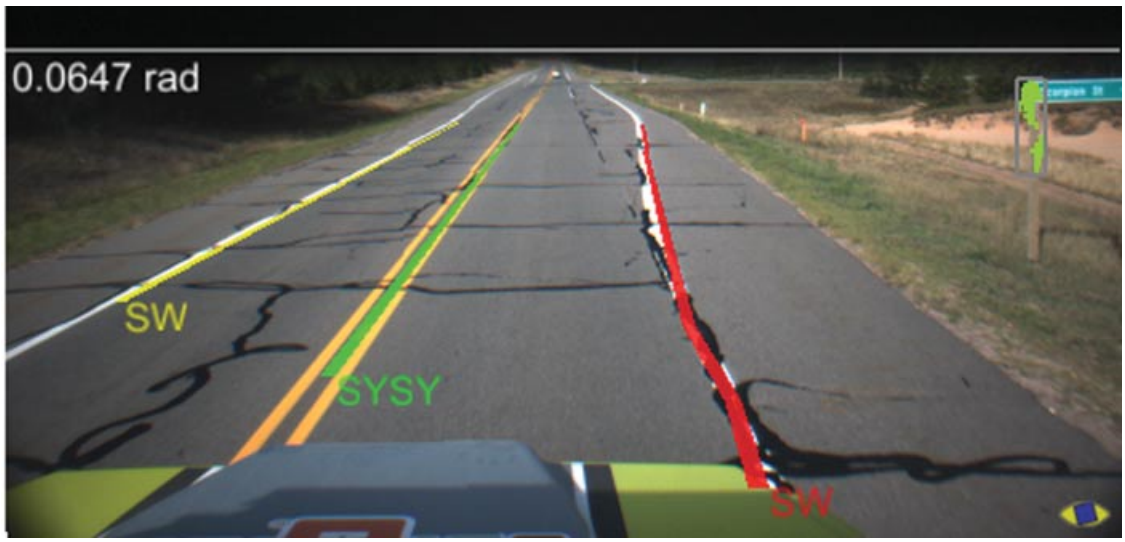

Figure 6. Results of lane detection algorithm projected on the original image. From right to left, the red line represents a right boundary, green a left boundary, and yellow a far left boundary. In this image, the right line is detected although it is not a complete line.

The three cameras form three possible baselines (the baseline is the distance between two stereo cameras), and the system automatically switches between them depending on the current vehicle speed; at lower speeds it is thus more convenient to use the shorter (and more accurate) baseline, whereas at higher speeds the large baseline permits the detection of obstacles when they are far from the vehicle.

Images are rectified, so that the corresponding epipolar lines become horizontal, thus correcting any hardware misalignment of the cameras and allowing for more precise measurements. The V-disparity map (Labayrade, Aubert, & Tarel, 2002) is exploited to extract the ground slope and current vehicle pitch, in order to compensate for the oscillations that occur while driving [Figures 5(a) and 5(b)].

The next step is to build a disparity map from the pair of stereo images: this operation is accomplished using a highly optimized incremental algorithm, which takes into account the previously computed ground slope in order to produce more accurate results and to reduce the processing time [Figure 5(c)].

The disparity map, along with the corresponding 3D world coordinates, is used to perform obstacle detection. After a multistep filtering phase aimed at isolating the obstacles present in front of the truck, the remaining points are merged with the ones from the front LIDAR and are initial values for a flood-fill expansion step, governed by each pixel disparity value, in order to extract the complete shape of each obstacle. This data fusion step ensures good performance in poorly textured areas, while ensuring robustness of the vision-based obstacle detection algorithm against LIDAR sensor failures. Previously identified obstacles are removed from the full-resolution image used by the lane detection algorithm, lowering the possibility of false positives, such as those introduced by poles or vehicle parts. Figure 6 shows a typical scene and the lanes detected by the algorithm.

4.2.3. Stereo System

Navigation in an urban environment requires precise maneuvers. The trinocular system described in the preceding section can be used for driving only at medium to high speeds, because it covers the far range (7–40 m). TerraMax includes two stereo systems [one in the front and one in the back, derived from Broggi, Medici, and Porta (2007)], which provide precise sensing at closer range. Using wide-angle (fish-eye, about 160 deg) lenses, these sensors gather information over an extended area of about 10×10 m; the stereo systems are designed to detect obstacles and lane markings with high confidence on the detection and position accuracy.

Obstacle detection is performed in two steps: first the two images, acquired simultaneously, are preprocessed in order to remove the high lens distortion and perspective effect, a thresholded difference image is generated and labeled (Bertozzi, Broggi, Medici, Porta, & Sjögren, 2006), and then a polar histogram-based approach is used to isolate the labels corresponding to obstacles (Bertozzi & Broggi, 1998; Lee & Lee, 2004). Data from the LIDARs are clusterized so that laser reflections in a particular area can boost the score associated with the corresponding image regions, thus enhancing the detection of obstacles.

The chosen stereo approach avoids explicit computation of camera intrinsic and extrinsic parameters, which would have been impractical, given the choice of using fish-eye lenses to cover a wide area in front of the truck. The use of a lookup table (generated using a calibration tarp) to remap the distorted input images to a bird's-eye view of the scene thus results in improved performance and reduced calibration time

Short-range line detection is performed using a single camera, to detect lane markings (even along a sharp curve), stop lines, and curbs. As the camera approaches the detected obstacles and lane markings, the position accuracy increases, yet the markings remain in the camera field of view due to the fish-eye lens. A precise localization of lane markings enables the following: lane keeping despite large width of the vehicle, stopping of the vehicle at close proximity to the stop line at intersections, accurate turning maneuvers at intersections, and precise planning of obstacle avoidance maneuvers. Figure 7 shows a frame with typical obstacle and lane detection.

4.2.4. Lateral System

We employ lateral perception to detect oncoming traffic at intersections (Figure 8). During a traffic merge maneuver, the vehicle is allowed to pull into traffic only when a gap of at least 10 s is available. For this, the vehicle needs to perceive the presence of oncoming traffic and estimate vehicle speeds at range. The intersecting road might be viewed at an angle other than 90 deg; therefore the lateral system

Figure 7. A sample frame showing typical obstacle and lane detection.

must be designed to detect traffic coming from different angles. We installed two high-resolution AVT Pike cameras (1,920 × 1,080 pixels) on TerraMax—one on each side—for lateral view, together with 8-mm Kowa lenses. With this configuration each camera can cover a 90-deg angle and is able to see objects at high resolution up to distances greater than 130 m.

The lateral camera image is processed using a robust, ad hoc background subtraction–based algorithm within a selected region of interest, with the system being triggered by the AVM when the vehicle stops at an intersection, yielding to oncoming traffic. This approach allows us to handle the high-resolution imagery with a simple, computationally effective approach by leveraging the semantic context of vehicular motion.

4.2.5. Rearview System

When driving along a road, in both urban and rural environments, lane changes and passing may occur. The rearview system is aimed at detecting passing vehicles (Figure 9). This solution has proven to be very robust, while keeping processing requirements low; the onboard camera setup (with cameras placed on top of the cab, looking backward) ensures good visibility, because oncoming traffic is seen from a favorable viewing angle. The rearview system processing is based on color clustering and optical flow. The first stage of processing performs data reduction in the image: a category is assigned to each pixel depending on its color, resulting in blobs that represent objects or portions of objects of uniform color. In the second

Figure 8. Lateral system algorithm results (detected vehicles are marked red).

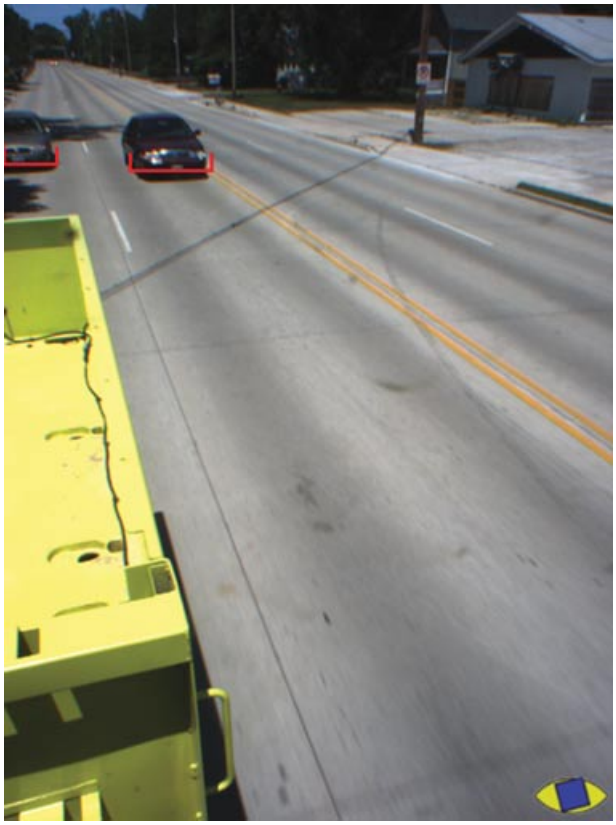

Figure 9. Rearview system algorithm results (detected vehicles are marked in red).

(optic flow) stage, blobs of uniform color are analyzed and tracked, to estimate their shape and movement.

Obstacles found using optical flow are then compared with those detected by the LIDAR: because the latter has higher precision, the position of obstacles estimated by vision is refined using the LIDAR data, if available. The fusion algorithm thus performs only position refinement and does not create/delete obstacles, in order to isolate the detection performance of the vision system from that of the LIDAR.

4.3. Obstacle Fusion and Tracking

We adopted a high-level approach for fusing (both dynamic and static) obstacle information. The high-level fusion approach was favored for its modularity and rapid implementation. It was also well-suited for our geographically dispersed development team.

In this approach, obstacles are detected locally by the LIDAR and vision systems. Detected obstacles are expressed as objects that contain relevant information such as outline points, ID, velocity, height (vision only), and color (vision only). The obstacles from LIDAR and vision are fused in the WPS based on their overlap and proximity.

Similarly, we relied on the native functions of the LIDAR (Wender et. al., 2006) and vision systems for obstacle tracking (through object IDs generated by these systems). This low-level-only tracking approach proved to be effective for most of the situations. However, it was inadequate in more complex situations in which a vehicle is temporarily occluded by another (see discussions in Sections 6.2 and 7).

To predict the future position of a moving vehicle, the WPS applies a nonholonomic vehicle kinematic model (Pin & Vasseur, 1990) and the context of the vehicle. For example, if the vehicle is in a driving lane, the WPS assumes that it will stay in lane. If the vehicle is not in a lane, the WPS assumes it will maintain its current heading.

5. PLANNING AND VEHICLE BEHAVIORS

In this section, we describe our approach for vehicle behavior generation and route/trajectory planning.

5.1. Overview of Vehicle Behaviors

We adopted a goal-driven/intentional approach to mission planning and generation of vehicle behaviors. The main goal for the mission and behavior generation is to navigate sequentially through a set of checkpoints as prescribed in the DARPA-supplied mission definition files (MDF). Functionality related to autonomous vehicle behaviors is implemented in the AVM.

Main components in the AVM (as depicted in Figure 10) include the mission/behavior supervisor (MBS), which manages the pursuit of mission goal (and subgoals) and selects and supervises the appropriate behavior mode for execution: the mission/route planner, which generates (and regenerates as needed) high-level route plans based on the road segments and zones defined in the RNDF; the behavior modes & logic, which contains a set of behavior modes, the transitional relationship among them, and the execution logic within each behavior mode; the event generators, which monitor the vehicle and environment state estimation from the WPS and generate appropriate events for the behavioral modes when a prescribed circumstance arises

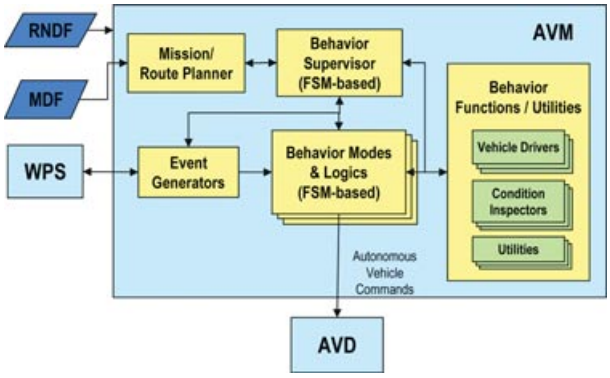

Figure 10. Major function blocks in the AVM.

(e.g., an obstacle in the driving lane ahead); and the behavior functions/utilities, which provide common services (e.g., trajectory generation) for different behavior modes.

5.2. Behavioral Modes and Supervision

We adopted a finite state machine (FSM)–based discrete-event supervisory control scheme as our primary approach to generate and execute autonomous vehicle behaviors. The FSM–based scheme provides us with a simple, yet structured, approach to effectively model the race rules/constraints and behaviors/tactics, as opposed to the conventional rule-based approaches or behavior-based approaches (Arkin, 1998). This scheme allows us to leverage existing supervisory control theories and techniques (Cassandras & Lafortune, 1999; Chen & Lin, 2001b; Chung, Lafortune, & Lin, 1992; Ramadge & Wonham, 1987) to generate safe and optimized behaviors for the vehicle.

We model autonomous behaviors as different behavior modes. These behavior modes categorize potential race situations and enable optimized logic and tactics to be developed for the situations. We implemented 17 behavior modes, shown in Figure 11, to cover all the basic and advanced behaviors prescribed in the Urban Challenge. Examples of the behavior modes include lane driving, in which the vehicle follows a designated lane based on the sensed lane or road boundaries; and inspecting intersection, in which the vehicle observes the intersection protocol and precedence rules in crossing intersections and merging with existing traffic.

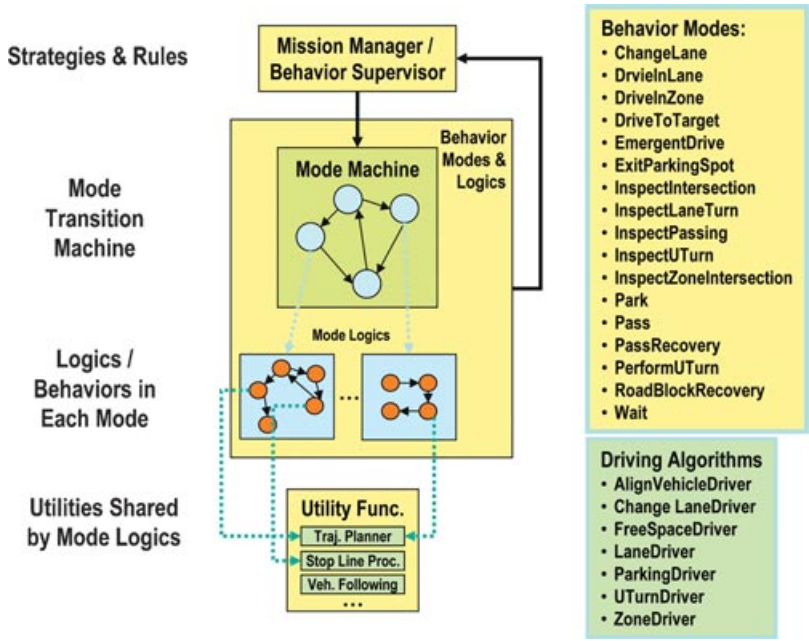

Figure 11. Components for behavior generation and execution.

For each behavior mode, a set of customized logic is modeled as an extended FSM [e.g., a FSM with parameters (FSMwP) (Chen & Lin, 2000)], which describes the potential behavior steps, conditions, and actions to be taken. The behavior logic may employ different utility functions (e.g., trajectory planners/driving algorithms and stop-line procedure) during execution.

Transitions among behavior modes are modeled explicitly as an FSMwP (Chen & Lin, 2000), named mode transition machine (MTM), in which guard conditions and potential actions/consequences for the transitions are expressed. The MTM is used by the behavior supervisor in MBS in determining the appropriate behavior mode to transition to during execution.

The generation and control of the vehicle behavior may be formulated as a supervisory control problem. We adopted the concepts of safety control (Chen & Lin, 2001a) and optimal effective control (Chen & Lin, 2001b) for FSMwP in which the traffic rules and protocols are formulated as safety constraints and current mission subgoal (e.g., checkpoint) as the effective measure to achieve. However, to improve the real-time performance during execution, we manually implemented a simplified supervisor that does not require explicit expansion of supervisor states (Chung et. al., 1992) by exploiting the structure of the MTM.

Our FSM-based behavior generation scheme is intuitive and efficient. However, it may suffer from several potential drawbacks. Among them are the reduced robustness in handling unexpected situations and the lack of "creative" solutions/behaviors. To mitigate the potential concern in handling unexpected situations, we included an unexpected behavior mode (RoadBlockRecovery mode) and instituted exception-handling logic to try to bring the vehicle to a known state (e.g., on a known road segment, or zone). Through our field testing and participation at official events, we found this unexpected behavior mode to be generally effective in ensuring the robust autonomous operation of the vehicle. A more robust unexpected behavior mode based on "non-scripted" techniques, such as behavior-based approaches (Arkin, 1998), may be introduced in the future to handle the unexpected situations. This hybrid approach would strike the balance between simplicity/consistency and flexibility/robustness of behaviors.

5.3. Route Planning

The objective of the route planning component is to generate an ordered list of road segments among the ones defined in the RNDF that enables the vehicle to visit the given set of checkpoints, in sequence, at the least perceived cost of the current sensed environment. We implemented the route planner as a derivative of the well-known Dijkstra's algorithm (Cormen, Leiserson, & Rivest, 1990), which is a greedy search approach to solve the single-source, shortest-path problem.

We used the estimated travel time as the base cost for each road segment, instead of the length of the road segment. This modification allowed us to effectively take into account the speed limit of the road segments (both MDF-specified and self-imposed due to the large vehicle's constraints) and the traffic conditions the vehicle may experience through the same road segment previously traveled (during the same mission run). Meanwhile, any perceived road information, such as road blockage, and (static) obstacles are also factored into the cost of the road.

5.4. Trajectory Planning

The trajectory planner generates a sequence of dense and drivable waypoints, with their corresponding target speeds, for the AVD to execute, given a pair of start/end waypoints and, possibly, a set of intermediate waypoints. Alternatively, the trajectory planner may also prescribe a series of driving commands (which include steering direction and travel distance). Instead of using a general-purpose trajectory/motion planner for all behavior modes, we implemented the trajectory planning capabilities as a collection of trajectory planner utility functions that may be called upon by different behavior modes depending on the current vehicle and mission situation. Our approach exploited specific driving conditions in different behavior modes for efficient and consistent trajectory planning.

We implemented four different types of trajectory planners:

- **Lane-following trajectory planner** utilizes detected/estimated lane and road boundaries to generate waypoints that follow the progression of the lane/road. Targeted speed for

each waypoint is determined by considering the (projected) gap between TerraMax and the vehicle in front, if any, dynamics of TerraMax (e.g., current speed, limits of acceleration/deceleration), and the speed limit of the road segment. Curvatures among waypoints are further checked, adjusted, and smoothed using a spline algorithm (Schoenberg, 1969) to ensure that the waypoints are drivable within TerraMax's dynamic constraints.

- **Template-based trajectory planners** are a set of trajectory planners that can quickly generate trajectory waypoints based on instantiation of templates (Horst & Barbera, 2006) with current vehicle and environment state estimates for common maneuvers such as lane changing, passing, swerving, and turning at intersections. A template-based trajectory planner determines first the targeted speed for (each segment of) the maneuver, using the method similar to that for the lane-following trajectory planner, and applies the targeted speed to the parameterized trajectory template in generating the waypoints.
- **Rule-based trajectory planners** utilize a set of simple driving and steering heuristic rules (Hwang, Meirans, & Drotning, 1993) that mimic the decision process of human drivers in guiding the vehicle into a certain prescribed position and/or orientation, such as U-turns or parking. Because the rule-based trajectory planners are usually invoked for precision maneuvers, we configured the targeted speed to a low value (e.g., 3 mph) for maximum maneuverability.
- **Open-space trajectory planner** provides general trajectory generation and obstacle avoidance in a prescribed open space where obstacles may be present. We adopted a two-level hierarchical trajectory planning approach in which a lattice/A*-based, high-level planner (Cormen et al., 1990) provides a coarse set of guiding waypoints to guide the trajectory generation of the low-level real-time path planner (RTPP), which is based on a greedy, breadth-first search algorithm augmented with a set of heuristic rules. Similar to that for the rule-based trajectory planner, we configure the target speed of the open-space trajectory planner to a low value (e.g., 5 mph) for maximum maneuverability.

6. FIELD PERFORMANCE AND ANALYSIS

In this section, we discuss TerraMax's performance during testing and participation in the Urban Challenge and related events.

6.1. Basic and Advanced Behaviors

TerraMax successfully demonstrated all the basic and advanced behavior requirements set forth by DARPA. In the following, we comment on our experience in implementing some key autonomous behaviors.

Passing: Initially, the trajectory generation of the passing behavior was handled by a template-based passing trajectory planner, which guided the vehicle to an adjacent lane for passing the detected obstacle in its current lane and returned the vehicle back to its original lane after passing. We added a *swerve* planner to negotiate small obstacles (instead of *passing* them). This modification resulted in smooth and robust passing performance.

U-turn and parking: Through testing, we found that a rule-based approach outperformed a template-based approach for U-turns and parking. Therefore we employed a rule-based parking maneuver, which performed flawlessly in the NQE and UCFE.

Merge and left turn: We adopted a simple approach to inspect traffic in the lanes of interest and determine whether there is a safe gap for executing merge or left-turn behaviors. We employ a simple vehicle kinematic model (Pin & Vasseur, 1990) to predict the possible spatial extent that a moving vehicle in the lane may occupy in the near future (e.g., in the next 10 s) and apply an efficient geometry-based algorithm to check whether the spatial extent intersects with TerraMax's intended path. To reduce false (both positive and negative) detections of traffic in a lane, we further fine-tuned LIDAR sensitivity, employed a multisample voting scheme to determine whether a vehicle is present based on multiple updates of the obstacles reported by the LIDAR, and verified our modifications through an extended series of controlled live-traffic tests. The enhancements resulted in near-perfect merging and left-turn behaviors.

6.2. Performance at the NQE

TerraMax participated in multiple runs in Test Areas A, B, and C during the NQEs. During these runs, TerraMax successfully completed all the key

autonomous maneuvers as prescribed in each Test Area. Specifically, in Test Area A, TerraMax demonstrated merging into live traffic and left-turn maneuvers; in Test Area B, TerraMax completed leaving the start chute, traffic circle, zone navigation and driving, parking, passing, and congested road segment maneuvers; in Test Area C, TerraMax demonstrated intersection precedence, queuing, roadblock detection, U-turn, and replanning behaviors.

In Table II, we summarize the performance issues we experienced during the early runs in the NQE and our actions in resolving/mitigating these issues for the subsequent runs in the NQE and the UCFE. We next discuss each of the performance items in detail.

Obstacles protruding from the edge of a narrow road could interfere with safety spacing constraints and cause path deviation: During our first run of Test Area A, TerraMax did not stay in the travel lane but drove on the centerline of the road segment. The K-rails on the road closely hugged the lane boundary, appearing to be inside the lane. This prompted obstacle avoidance, and due to our earlier safety-driven decision to never approach an obstacle closer than 0.5 m, the AVM decided to cross the centerline to pass.

Unreliable path/lane updates could cause incorrect travel lane determination: During the first run of Test Area B, TerraMax pulled into a driveway of a house and, after a series of maneuvers, successfully pulled out of that driveway but led itself right into the driveway next door, where it stalled and had to be manually repositioned. This time-consuming excursion was primarily triggered by an incorrect path update and subsequent incorrect travel lane selection. We resolved the faulty path update and readjusted the relative importance of the different information sources used to determine the travel lane.

Dust clouds and other false-positive "transient" obstacles could result in unexpected behaviors: Unexpected

Table II. Performance issues and resolutions in NQE.

Performance issue	Cause	Impact on performance	Mitigating action(s)	Lessons learned
Minor obstacles (K-rails) causing planned path deviation	Safety parameter setting	Vehicle rode centerline	Reduce clearance to obstacle Widen path by moving K-rails	Narrow roads are harder to navigate for a large truck
Path updates incorrectly processed	False positives for road edges	Vehicle got stuck in driveway	Corrected path updates and readjusted path determination method	Road edge detection and processing can be complex and unreliable
Parking twice in the same spot	Dust clouds behind vehicle perceived as obstacles	Total time taken increased	None required	Temporary dust clouds need to be treated differently from other obstacles; high-level obstacle fusion desirable
Traveling too close to parked cars	Real-time path planner not activated in time	Brushing a parked car	Modified trajectory planning transition logic	Navigating congested narrow road segments needs sophisticated supervisory logic
Entering intersection prematurely	Timer bug	Incorrect precedence at intersection	Fixed timer bug	Simple bugs could cause large-scale performance impact
Incorrect tracking at intersections	Vehicle hidden by another	Incorrect precedence at intersection	Implemented obstacle caching; better sensor fusion and tracking	Simple sensor processing is inadequate in complex settings
Queuing too close/erratic behaviors	LIDAR malfunction	Nearly ran into the vehicle in front	Fixed bug in sensor health monitoring	Critical system services have to be constantly monitored

behaviors observed in dirt lots of Test Area B can be attributed to dust clouds having been sensed as obstacles. These behaviors included parking twice in both runs, a momentary stop, and "wandering around" in the dirt lot. On all of these occasions, the system recovered well and did not result in failures other than the superfluous excursions.

More sophisticated free-space trajectory planners (other than simple template-based ones) are needed to negotiate highly congested areas: While navigating a segment where many obstacles were set up on a curvy road during the first run of Test Area B, TerraMax slowly knocked down several traffic cones and brushed a parked vehicle with the left corner of its front bumper. Our postrun study revealed that the RTPP was triggered much less often than desired. To prevent TerraMax from hitting obstacles, especially in the congested road segments as we experienced, we revised AVM's transition logic and processes between normal driving modes and RTPP. With the revisions, the supervisor invoked RTPP in more situations.

Persistent vehicle tracking needed for complex situations at intersections: During the second run of Test Area C, TerraMax entered the intersection prematurely on one occasion. This was due to the lack of proper tracking of the obstacles/vehicles in our overall system. When the first vehicle entered the intersection, it momentarily occluded the second vehicle. The AVM could not recognize that the newly observed obstacle was in fact the vehicle that was there before. We mitigated this problem by refining the obstacle caching and comparison mechanism.

Proper response should be designed to prepare for subsystem malfunction: In the second run of Test Area C, TerraMax started to behave erratically after one-third of the mission was completed. It drove very close to the obstacles (and K-rails) at the right side of the lane, and it almost hit a vehicle that queued in front of it at the intersection. In analysis, we discovered that the front-right LIDAR had malfunctioned and did not provide any obstacle information. This, combined with the fact that the stereo obstacle detection had not been enabled meant that TerraMax could not detect obstacles in the front right at close range. Fortunately, there was some overlap coverage from the front-left LIDAR, which was functioning correctly. This overlap coverage from the front-left LIDAR picked up the vehicle queued in front of TerraMax at the intersection so that the supervisor stopped TerraMax just in time to avoid hitting this vehicle. Stereo obstacle detection was not turned on because the team did not have time to tune the thresholds before the start of this run.

6.3. Performance at the UCFE

TerraMax completed the first four submissions in Mission 1 of the UCFE with impressive performance. However, TerraMax was stopped after a failure in the parking lot (zone 61). Table III summarizes TerraMax's performance statistics prior to the parking lot event.

The arrival into the parking lot was normal. No detected obstacles in the path of TerraMax were detected, and therefore an s-shaped (farmer) turn was issued that brought TerraMax into alignment with the target parking space [Figure 12(a)]. TerraMax continued the parking maneuver successfully and pulled out of the parking space without error [Figure 12(c)]. TerraMax backed out of the parking space to the right so that it would face the exit of the zone when finished exiting the parking spot.

There were two separate problems in the parking lot. The first problem was that TerraMax stalled in the parking lot for a very long time (approx. 17 min). The second problem was that when TerraMax eventually continued, it no longer responded to commands from the AVM and eventually had to be stopped.

Table III. Performance statistics for UCFE prior to parking lot.

	Mission 1 (first 4 submissions)	
Total time	0:40	
Top speed	21 mph	
Oncoming vehicles encountered	47	
Oncoming autonomous vehicles encountered	7	
	Pass	Fail
Int. precedence	9	0
Vehicle following	0	0
Stop queue	1	0
Passes	1	0
Road blocks/replan	0	0
Zone	1	0
Park	1	0
Merge[a]	1	1

[a]Failed cases indicate that traffic flow was impeded by TerraMax during merge.

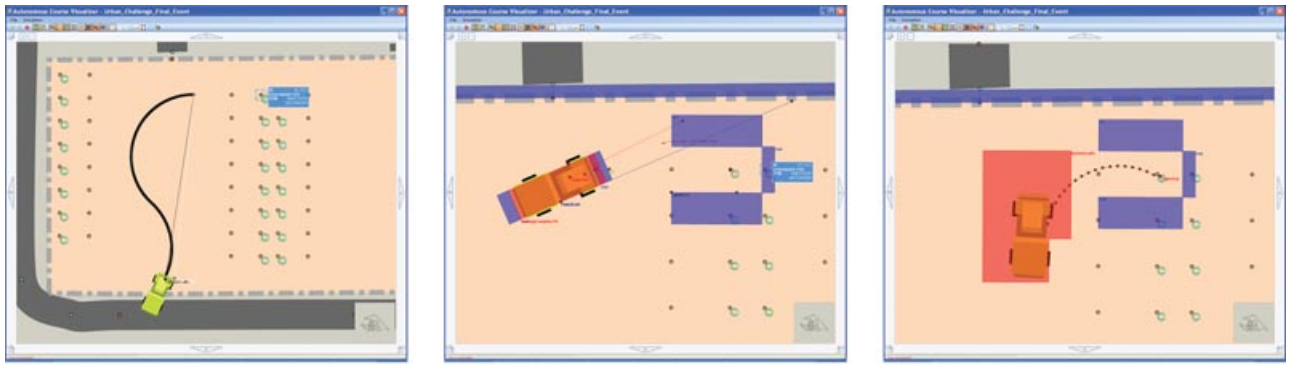

Figure 12. UCFE parking lot: Successful parking maneuvers.

Stall condition. The RTPP produced paths that contained duplicate waypoints while driving in a zone, resulting in a stall. A bug was introduced in the RTPP during prerace modifications. This bug was quickly corrected after the UCFE and tested when we retested at Victorville in December 2007.

Unresponsive vehicle. TerraMax recovered after stalling for more than 17 min. GPS drift "moved" the position of TerraMax to where the RTPP returned four points (two pairs of duplicate waypoints). The open-space trajectory planner then commanded TerraMax to drive at 5 mph in a straight path toward the parking lot boundary. However, the order of the commanded duplicate waypoints in the drive command caused the AVD service to fault at the point where the vehicle had already accelerated to approximately 1 mph. At this point, TerraMax became unresponsive to subsequent commands and continued to drive in a straight line toward a building until an E-stop pause was issued by DARPA officials.

6.4. Return to Victorville

We were unable to acquire full performance metrics during the UCFE due to the premature finish. Therefore we brought TerraMax back to Victorville on December 13, 2007, for a full test. Although we were not able to use the entire UCFE course, we used the UCFE RNDF and limited the missions to the housing area and added a parking spot in Zone 65. The MDF created for these missions used the speed limits from the MDF for the first mission at the UCFE. TerraMax ran with the software version used in the UCFE. No revisions were made. The team ran three missions totaling more than 78 miles in 7 h and 41 min, for an average speed of 10.17 mph. Six test vehicles driven by team members acted as other traffic. One safety vehicle followed TerraMax with a remote e-stop transmitter at all times during autonomous operation. We list performance statistics for the three missions in Table IV.

Table IV. Performance statistics for Victorville test missions.

	Mission 1		Mission 2		Mission 3	
Total distance	24.9 miles		19.5 miles		33.8 miles	
Total time	2:28		1:55		3:18	
Average speed	10.09 mph		10.17 mph		10.24 mph	
Oncoming vehicles encountered in opposite lane	42		92		142	
Intersections	84		108		193	
	Pass	Fail	Pass	Fail	Pass	Fail
Int. precedence[a]	17	3	8	0	20	2
Vehicle following	3	0	1	0	2	0
Stop queue	2	0	1	0	1	0
Passes	5	0	3	0	1	0
Road blocks/replan	0	0	1	0	2	0
Zone	5	0	4	0	7	0
Park	3	0	2	0	2	0
Merge[b]	7	1	5	0	10	1

[a]In all cases, failed intersection precedence was due to the test vehicle(s) being more than 0.5 m behind the stop line.
[b]Merge results include left turns across opposing lane and left and right turns into traffic from a stop with other vehicles present. Failed cases indicate that traffic flow was impeded by TerraMax during merge.

7. CONCLUDING REMARKS

Team Oshkosh entered the DARPA Urban Challenge with the intention of finishing the event as a top contender. Despite not completing the final event, the team believes that TerraMax performed well and safely up to the point the vehicle was stopped. TerraMax proved to be a very capable contender and is arguably the only vehicle platform in the competition that is relevant for military logistics missions.

Through the development, testing, and official events, we experimented and demonstrated autonomous truck operations in (controlled) urban streets of California, Wisconsin, and Michigan under various climate conditions. In these experiments, TerraMax exhibited all the autonomous behaviors prescribed by DARPA Urban Challenge rules, including negotiating urban roads, intersections, and parking lots, interacting with manned and unmanned traffic while observing traffic rules, with impressive performance. Throughout this endeavor, we learned valuable experience and lessons, which we summarize in the following.

Course visualizer/simulator efforts truly paid off. Learning from our experience in DARPA Grand Challenge 2005, we invested time and effort up front to develop a graphic tool with a two-dimensional capability for visualizing the RNDF and MDF. We later expanded the functionality of the tool to include mission data log playback, sensor information display, built-in debugging capability that displays results of various autonomous mode status, logic and calculations in the AVM, and simple simulation of TerraMax operations.

This tool served as a true "force-multiplier" by allowing team members in widely dispersed geographic locations to verify different functionality and designs, experiment with different ideas and behavioral modes, pretest the software implementation prior to testing onboard TerraMax, and perform post-run analysis to resolve issues. The tool not only sped up our development and testing efforts, but also enabled us to quickly identify the causes for issues encountered during NQE runs and to promptly develop solutions to address them.

Simplicity worked well in U-turn and parking. Instead of using a full-fledged trajectory planner/generator for maneuvers such as U-Turn and parking, we opted to search for simple solutions for such situations. Our experiments with U-turn prior to the site visit clearly indicated the performance, simplicity, elegance, and agility superiority of a rule-based approach over a template-based one. The rule-based approach, which mimics a human driver's actions and decision process in performing those maneuvers, was our main approach for both U-turn and parking maneuvers and performed flawlessly in all our site visit, NQE, and UCFE runs.

Better persistent object tracking capability is needed. In the original design, object tracking was to be performed at a high level. However, due to time constraints, the object tracking responsibility was delegated to the processing modules of the individual sensor elements. As demonstrated in our first two runs of Test Area C, a persistent object tracking capability is required to handle situations in which objects may be temporarily obstructed from observations. However, this persistent object tracking mechanism should focus only on the objects of both relevance and importance for efficiency and practicality. Though we implemented a less-capable alternative to maintain persistent tracking of vehicles at the intersection that yielded satisfactory results, a systematic approach to address this shortfall is needed.

Our sensor technology proved capable, but additional work is required to meet all the challenges. The use of passive sensors is one of the primary goals of our vehicle design. LIDAR sensors produce highly accurate range measurements; however, vision allows cost-effective sensing of the environment without the use of active signals (Bertozzi, Broggi, & Fascioli, 2006; Bertozzi et. al., 2002; Broggi, Bertozzi, Fascioli, & Conte, 1999) and contains no moving parts, which are less desirable in operational environments. The calibration of vision systems needed special care because many of them were based on stereo technology whose large baseline precluded attachment of the two cameras to the same rig. Nevertheless, the calibration of many of them turned out to be absolutely robust, and there was no need to repeat the calibration procedure in Victorville after it was performed a few months before. The calibration of the trinocular system, which is installed into the cabin, survived a few months of test and many miles of autonomous driving. The front stereo system was uninstalled for maintenance purposes a few times and a recalibration was necessary, including in Victorville. The back stereo system did not need any recalibration, whereas the lateral and the rearview system, being monocular, relied on only a weak calibration, which was just checked before the race. This is a clear step forward in usability and durability of vision systems for real-world unmanned systems applications.

Performing low-level fusion between vision and LIDAR data has brought considerable improvements in distance measurement for the vision systems, especially at greater distances. Robustness and persistence of results are additional improvements realized by means of this technique. Despite these improvements, LIDAR was used as the primary sensor for obstacle detection and vision the primary sensor for road boundary detection during the Final Event.

Lane detection provided consistent data and was able to localize most of the lane markings. Some problems were encountered when lane markings were too worn out and in situations in which the red curb was misinterpreted as a yellow line. The specific algorithm used to identify yellow markings was not tested in correspondence to red curbs, which showed the same invariant features that were selected for yellow lines.

Improved road boundary interpretation is needed. Although the sensor system detected lanes and curbs in most situations, problems were encountered in situations in which sensed data differed significantly from the expected road model obtained by interpreting the RNDF data. As a result TerraMax would cross the centerline, cut corners, or drive off the road in order to reach the next RNDF waypoint.

Test for perfection. The team had tested TerraMax extensively in a variety of environments and scenarios; the NQE and UCFE differed from our testing situations sufficiently, however, so that we were required to make last-minute revisions to the system that were not extensively tested. Unfortunately, these last-minute revisions for the NQE adversely impacted performance in the UCFE.

The DARPA Urban Challenge, as all DARPA Grand Challenges, has proven to be an excellent framework for the development of unmanned vehicles for Team Oshkosh. We have experimented and developed many elegant solutions for practical military large-footprint autonomous ground vehicle operations in urban environments. The system developed by Team Oshkosh for the Urban Challenge has been shown to be robust and extensible, an excellent base to which additional capabilities can be added due to the modular architecture.

REFERENCES

Arkin, R. C., (1998). Behavior-based robotics. Cambridge, MA: MIT Press.

Bertozzi, M., Bombini, L., Broggi, A., Cerri, P., Grisleri, P., & Zani, P. (2008). GOLD: A complete framework for developing artificial vision applications for intelligent vehicles. IEEE Intelligent Systems, 23(1), 69–71.

Bertozzi, M., & Broggi, A. (1998). GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Transactions on Image Processing, 1(7), 62–81.

Bertozzi, M., Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., & Porta, M. (2002). Artificial vision in road vehicles. Proceedings of the IEEE—Special issue on Technology and Tools for Visual Perception, 90(7), 1258–1271.

Bertozzi, M., Broggi, A., & Fascioli, A. (2006). VisLab and the evolution of vision-based UGVs. IEEE Computer, 39(12), 31–38.

Bertozzi, M., Broggi, A., Medici, P., Porta, P. P., & Sjögren, A. (2006). Stereo vision-based start-inhibit for heavy goods vehicles. In Proceedings of IEEE Intelligent Vehicle Symposium 2006, Tokyo, Japan (pp. 350–355). Piscataway, NJ: IEEE.

Braid, D., Broggi, A., & Schmiedel, G. (2006). The TerraMax autonomous vehicle. Journal of Field Robotics, 23(9), 655–835.

Broggi, A., Bertozzi, B., Fascioli, A., & Conte, G. (1999). Automatic vehicle guidance: The experience of the ARGO vehicle. Singapore: World Scientific.

Broggi, A., Medici, P., & Porta, P. P. (2007). StereoBox: A robust and efficient solution for automotive short range obstacle detection. EURASIP Journal on Embedded Systems—Special Issue on Embedded Systems for Intelligent Vehicles, Vol. 2007.

Caraffi, C., Cattani, S., & Grisleri, P. (2007). Off-road path and obstacle detection using decision networks and stereo. IEEE Transactions on Intelligent Transportation Systems, 8(4), 607–618.

Cassandras, C., & Lafortune, S. (1999). Introduction to discrete event systems, 2nd ed. Boston: Kluwer.

Chen, Y.-L., & Lin, F. (2000). Modeling of discrete event systems using finite state machines with parameters. In Proceedings of the 9th IEEE International Conference on Control Applications, Anchorage, AK (pp. 941–946). Piscataway, NJ: IEEE.

Chen, Y.-L., & Lin, F. (2001a). Safety control of discrete event systems using finite state machines with parameters. In Proceedings of the 2001 American Control Conference (ACC), Arlington, VA (pp. 975–980). Piscataway, NJ: IEEE.

Chen, Y.-L., & Lin, F. (2001b). An optimal effective controller for discrete event systems. In Proceedings of the 40th IEEE Conference on Decision and Control (CDC), Orlando, FL (pp. 4092–4097). Piscataway, NJ: IEEE.

Chung, S.-L., Lafortune, S., & Lin, F. (1992). Limited look-ahead policies in supervisory control of discrete event systems. IEEE Transactions on Automatic Control, 37(12), 1921–1935.

Cormen, T., Leiserson, C., & Rivest, R. (1990). Introduction to algorithms. Cambridge, MA: MIT Press.

Horst, J., & Barbera, A. (2006). Trajectory generation for an on-road autonomous vehicle. In Proceedings of SPIE: Unmanned Systems Technology VIII, Vol. 6230.

Hwang, Y. K., Meirans, L., & Drotning, W. D. (1993). Motion planning for robotic spray cleaning with

environmentally safe solvents. In Proceedings of the IEEE International Workshop on Advanced Robotics, Tsukuba, Japan (pp. 49–54). Piscataway, NJ: IEEE.

Kaempchen, N., Bühler, M., & Dietmayer, K. (2005). Feature-level fusion for free-form object tracking using laserscanner and video. In Proceedings of the 2005 IEEE Intelligent Vehicles Symposium, Las Vegas, NV (pp. 453–458). Piscataway, NJ: IEEE.

Labayrade, R., Aubert, D., & Tarel, J. P. (2002). Real time obstacle detection in stereo vision on non-flat road geometry through V-disparity representation. In Proceedings of the IEEE Intelligent Vehicles Symposium, Versailles, France (Vol. II, pp. 646–651). Piscataway, NJ: IEEE.

Lee, K., & Lee, J. (2004). Generic obstacle detection on roads by dynamic programming for remapped stereo images to an overhead view. In Proceedings of ICNSC 2004, Taipei, Taiwan (Vol. 2, pp. 897–902). Piscataway, NJ: IEEE.

Pin, F. G., & Vasseur, H. A. (1990). Autonomous trajectory generation for mobile robots with non-holonomic and steering angle constraints. In Proceedings of the IEEE International Workshop on Intelligent Motion Control, Istanbul, Turkey (pp. 295–299). Piscataway, NJ: IEEE.

Ramadge, P. J., & Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. SIAM Journal on Control and Optimization, 25(1), 206–230.

Schoenberg, I. J. (1969). Cardinal interpolation and spline functions. Journal of Approximation Theory, 2, 167–206.

Sundareswaran, V., Johnson, C., & Braid, D. (2006). Implications of lessons learned from experience with large truck autonomous ground vehicles. In Proceedings of AUVSI 2006, Orlando, FL. Arlington, VA: AUVSI.

Team Oshkosh. (2007). DARPA Urban Challenge technical report. Oshkosh Corp. www.darpa.mil/grandchallenge/TechPapers/Team_Oshkosh.pdf.

Wender, S., Weiss, T., Dietmayer, K., & Fuerstenberg, K. (2006). Object classification exploiting high level maps of intersections. In Proceedings of the 10th International Conference on Advanced Microsystems for Automotive Applications (AAMA 2006), Berlin, Germany. New York: Springer.