

# The Crusher System for Autonomous Navigation

Anthony Stentz, John Bares, Thomas Pilarski, and David Stager

National Robotics Engineering Center

Carnegie Mellon University

Pittsburgh, Pennsylvania 15201

Email: {axs,bares,temp,cop}@rec.ri.cmu.edu

Phone: (412) 681-6900

URL: <http://www.rec.ri.cmu.edu>

## Abstract

*We present the Crusher system for autonomously navigating complex off-road terrain. In this paper, we describe the Crusher system's three-pronged approach for safely and reliably moving between widely-spaced waypoints. First, the system automatically interprets aerial map data to assess mobility risk and plans trajectories that move from way point to way point. Second, the system uses a ladar- and camera-based perception system to detect and avoid hazards that are not discernable in the map data or that appear after the area is mapped. Third, the Crusher vehicle breaches hazards too difficult for on-board sensors to detect. The autonomy software is adaptive for operation on a wide range of terrain types. To date, the Crusher system has been tested on terrain with dense trees, long washes, deep ditches, steep slopes, thick brush, and large rocks. In early 2007, the system was tested at Ft. Bliss in Texas, where it drove over two hundred fifty kilometers autonomously.*

## INTRODUCTION

We define autonomous navigation as the task of driving a ground vehicle, equipped with sensors for measuring properties of the terrain, computers for interpreting the data, and actuators for steering, accelerating, and braking the vehicle, from way point to way point across a given terrain, such that the vehicle operates safely and within the correct performance envelope for the operation, but with little or no human involvement or intervention.

The vehicle may be assisted by a map of the terrain, if one is available, but it is able to drive successfully without such a map. The way points may be closely spaced or widely separated—the latter implies that the vehicle may need to plan and follow routes, in addition to avoiding obstacles. Ideally, there is no human in-

volvement at all. If an intervention is required, remotely driving out of trouble is considered far less severe than stopping the vehicle to prevent catastrophe.

## **TECHNICAL APPROACH**

The research described in this paper significantly pushes the state of the art for autonomous navigation in complex terrain. Progress in the field requires a complete *system* effort, pushing the limits of many component technologies such as obstacle detection and avoidance, route planning, position estimation, vehicle control, sensor fusion, and map interpretation. We list the key elements of our approach below. The Crusher system is the only cross-country navigator to embody all of these elements in a single system:

*Multi-Sensor, Multi-Feature, Multi-Viewpoint, Multi-Range/Resolution Ground and Aerial Perception:* for complex terrains, it is very difficult to distinguish hazards (e.g., rocks) from non-hazards (e.g., bushes). The Crusher system uses multiple sensor modalities, including red/green/blue and near infrared cameras, and lidar range and remission data to maximize the chances that there will be signature data present. Numerous features are extracted from the data, including local shape, range pixel density, normal vector orientation, the difference between visible red and near infrared light, and others to reduce the data bandwidth without reducing the information content. The data is acquired from multiple viewpoints (i.e., air and ground) as well as multiple ranges and resolutions. The system reasons about which sensor data to believe given sensor type, range, resolution, and hazard/non-hazard type. This approach has enabled Crusher to detect hazards such as steep slopes, ditches, ravines, holes, trees, boulders, and rocks, man-made obstacles (e.g., vehicles and other machinery), and areas with poor traction.

*Learning Algorithms, Risk Assessment, and Data Inferences:* given the large variety of both hazards and non-hazards on natural terrain, it is not possible to fully program a system to correctly classify all of them; instead, we advocate a learning approach to assist in the process. Our algorithms learn from human-provided examples as well as logged vehicle data, both off-line and on-line. These algorithms have been instrumental in significantly improving our performance on non-hazards. The Crusher system is capable of distinguishing hazardous vegetation (e.g., stout bushes, bramble, and thick branches) from non-hazardous vegetation (e.g.,

grass, weeds, sparse bushes). In cases where the level of hazard is uncertain, the system assigns a risk value (i.e., continuous cost) so that the planner is able to trade-off driving over the terrain feature with taking an alternative route. Furthermore, the Crusher system is able to infer the presence of hazards it cannot see, such as those occluded by vegetation cover. The approach includes slowing the vehicle down for shadowed areas and “filling in” missing data to infer holes and estimate the supporting ground plane.

*Obstacle Avoidance and Maneuvering for Difficult Environments:* the Crusher system assumes that the vehicle will need to thread its way around obstacles, squeeze through tight spaces, and turn on surfaces with poor traction. The system is able to plan and execute the trajectories necessary to perform these actions, including coupling forward and backward motion and making use of a calibrated vehicle model to minimize tracking error.

*Continuous Route Re-planning:* at every point in its traverse, the Crusher system selects the best route given the latest and most accurate information available. Thus, the system re-plans continuously as new information is received. It makes use of sketchy obstacle information acquired at a long range and then updates its route as the vehicle draws near and acquires more accurate information. The system represents map data at the highest resolution available and fuses information from multiple passes over the same area to ensure the vehicle has an accurate picture of the options available to it. This approach is essential for driving in cluttered environments where few safe options may be available. Furthermore, it has also enabled the Crusher system to navigate with any amount of terrain uncertainty, since the system does not require a prior map but makes the best use of the information available to it.

The Crusher system uses a three-pronged approach to perform a “mission”, consisting of a sequence of widely spaced way points on a given terrain. First, if a map is available, the system analyzes it for navigable areas and hazards and selects a route that moves the vehicle safely from one way point to the next. Second, the system scans the terrain with its sensors as the vehicle drives to detect hazards too small to be resolved from map data or which appeared after the terrain was mapped. The system avoids the hazards or re-plans an

alternative route as needed. Third, the Crusher vehicle is designed to breach/survive some hazards that were missed by the perception system and would have been avoided otherwise.

## CRUSHER VEHICLE

### High Mobility Platform

Crusher is a 6,800 kg, six-wheeled, hybrid-powered robot which is extremely capable in the most severe terrain. Figure 1 lists the vehicle's performance specifications.

Weight (no payloads, 30gal fuel)	6,071 kg	13,356 lb
Payload+armor weight	2,384 kg	5,244 lb
Operating vehicle weight (GVW)	8,455 kg	18,600 lb
Normal operating height (with 16" ground clearance)	1.5 m	60 in
Minimum crouched height (belly on ground)	1.1 m	44 in
Overall length	5.1 m	201 in
Overall width	2.6 m	102 in
Tire diameter	1.3 m	50 in
Max speed	41.6 kph	26 mph
Acceleration (0-42kph)	7.5 sec	
Min turning radius	Pivot steer capable	
Slope climbing (traction limited)	89 %	40 deg
Side slope (traction limited)	66 %	30 deg
Step crossing	1.2 m	48 in
Gap /trench crossing	2.0 m	80 in
Engine	Turbo-diesel	
Engine-generator output (@120F ambient)	58 kW	78 hp
Onboard battery storage	18.7 kW-hr	
Traction motor power (cont duty, sum all mtrs)	210 kW	282 hp
Power/weight ratio (continuous)	25 kW/ton	30 n
Peak torque (sum of 6WD)	54,302 N-m	40,051 ft-lb
Tractive Force/GVW ratio	1.26	
Payload volume (total internal)	1.63 m <sup>3</sup>	57.7 ft <sup>3</sup>
Payload power	28V, 5kW	
Control modes	RC control Teleoperation control Waypoint following Fully autonomous (w/ add'l equip)	

**Figure 1: Crusher performance specifications.**

High-strength aluminum tubes and titanium nodes make up Crusher's space frame hull design. Crusher can comfortably carry over 8000 lbs. of payload and armor. The hybrid electric system allows the vehicle to move silently on one battery charge over miles of extreme terrain. A 60 kw turbo diesel engine maintains charge on a high-performance lithium ion battery module. Engine and batteries work intelligently to deliver power to Crusher's six-wheel motor-in-hub drive system.

Crusher's advanced suspension supports 30 inches of travel with selectable stiffness and reconfigurable ride height. This suspension system provides a smooth ride for the navigation sensors over rough surfaces even at speeds up to 12 meters per second and enables the vehicle to climb obstacles and cross gaps (see Figure 2).



**Figure 2: Crusher breaching an obstacle (left) and crossing a ditch (right).**

A suspended and shock-mounted skid plate made from high-strength steel enables Crusher to shrug off massive, below-hull strikes from boulders and tree stumps. This skid plate enables Crusher to survive driving over rocks that are hidden in vegetation—an obstacle that is hard for an autonomy system to detect without a foliage penetrating sensor.

Crusher's bumper was designed to absorb the impact energy from major frontal collisions with trees and rocks. This bumper allows the autonomy system to confidently interact with the terrain to determine if an object is crushable or not (e.g., vegetation, boulder, cactus, tree, etc.). In short, the Crusher vehicle can survive an autonomy system that makes an occasional mistake. Furthermore, the autonomy system can learn from such mistakes without sacrificing the vehicle.

### **Navigation Sensors**

Crusher uses laser rangefinders to measure the geometry of the terrain. The geometry is important for determining the supporting surface, positive obstacles such as trees, and negative obstacles such as ditches. Crusher uses cameras in the red/green/blue/near infrared range to measure the appearance of the terrain. The appearance provides important clues about the material properties of the terrain, for example, which portions

are vegetation and which portions are solid objects. In order for the cameras to be usable outdoors, we developed a high-dynamic range (HDR) system that fuses images with multiple exposure times to compensate for a wide range of illumination.

Crusher operates in complex off-road terrain and thus requires a wide sensor field of view, since the vehicle must see in many directions to negotiate a cluttered, confined area. Figure 3 shows the sensor pod configuration on Crusher. The sensor pod is made up of laser rangefinders and cameras that give a total field of view of 180 degrees. The sensor pod laser rangefinders consist of six SICK LMS lidar sensors, each with a 90 degree field of view, 0.5 degree beam spacing, 181 points per scan, and 75 Hz update rate. The six lidars scan independently and generate a total of 81,450 range points per second. The four forward-looking lidars cover a 122 degree field of view in front of the vehicle. Each side-looking lidar covers a 48 degree field of view. The sensor pod includes four camera cubes, each containing four cameras. In a cube, two of the cameras capture red, green, and blue light, while the other two capture visible and near-infrared light to compute a difference relevant to vegetation detection. Each camera in the pod has a field of view of 70 degrees x 53 degrees with 1024 x 768 pixel resolution and a 15 frame per second capture rate. The two forward-looking and two side-looking cubes collectively cover the lidar fields of view.

The sensor pod produces camera images and colorized laser points that are used by various perception system modules. Both the camera images and the colorized laser points are time and pose tagged. The time tag represents the time the sensor data was acquired and the pose represents the pose of the vehicle at the time the sensor data was captured.

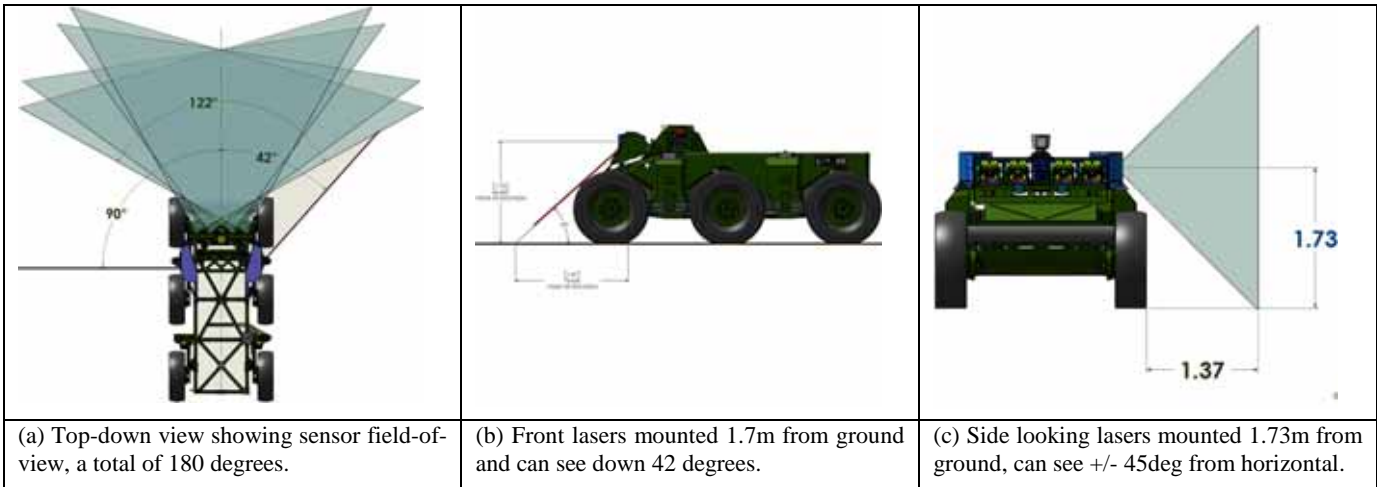


Figure 3: Crusher autonomy sensor pod configuration and fields-of-view.

## AUTONOMY ARCHITECTURE

### Software Architecture

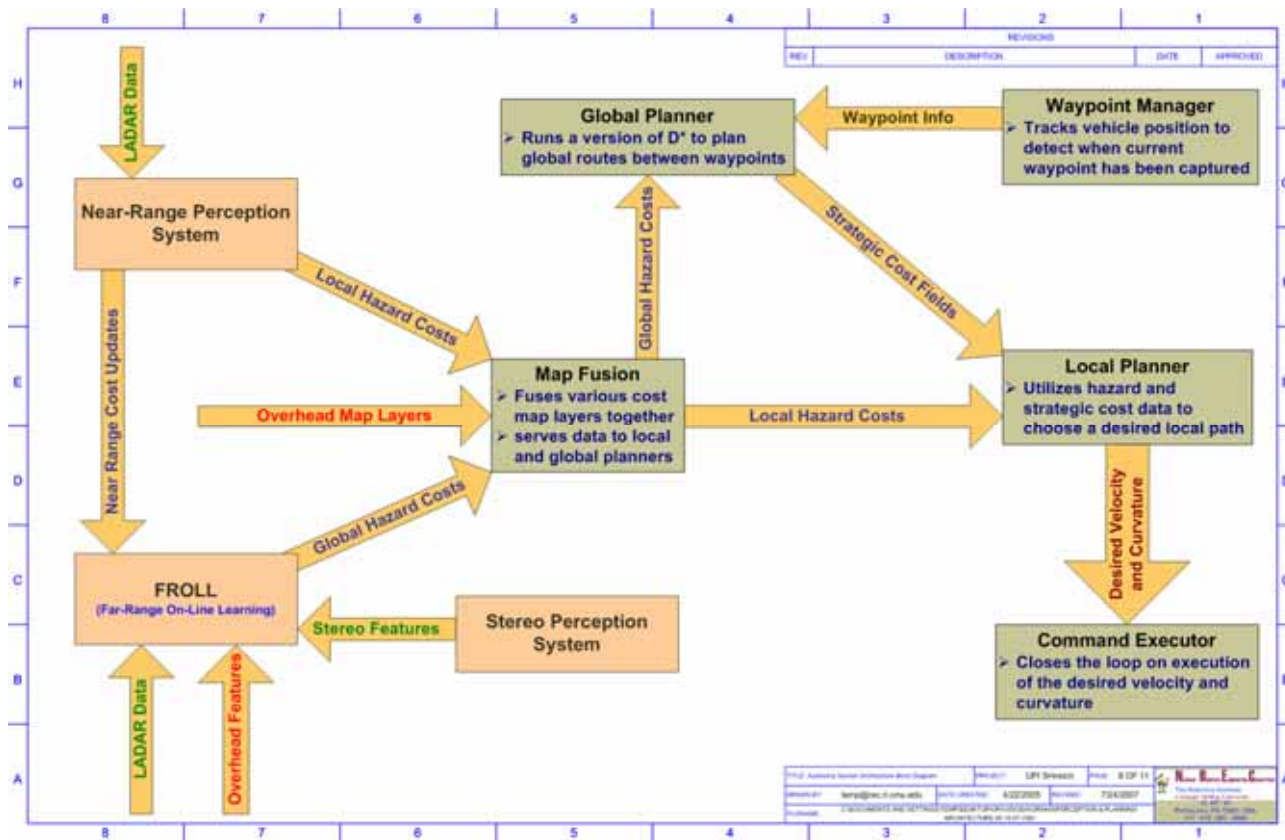


Figure 4: UPI Software architecture.

In order to support autonomous driving, we designed an architecture that is deliberative enough to support the strategic mission of moving the vehicle from one way point to the next, and reactive enough to respond to the many surprises encountered along the way. The Crusher architecture is an improvement on that developed for Demo II [4] and extended for PerceptOR [3], supporting multiple sensors operating at different ranges, greater concurrency, mobility and tactical maps, learning, and better sensor/map fusion.

The Crusher software architecture resembles a hierarchical architecture such as 4D-RCS [1] in that it uses computational nodes (i.e., perception-planning-control loops) at multiple time scales. It is different in that the spatial organization matches the sensor fields of view, and the spatial resolution is uniformly high at all levels. The architecture resembles a behavioral framework [2] in that it is organized around specific capabilities and is quick to respond, and it is different in that each module may make extensive use of maps and models of the environment.

The Crusher architecture is based on the principle that the modules are organized around specific capabilities, employing whatever spatial and temporal resolutions are necessary to accomplish the task, even if the result is not a uniform abstraction across space and time from bottom to top. The modules may be fast and simple, or they may be slow and complex—whatever it takes to do the task.

The architecture is shown in Figure 4. The Waypoint Manager is given the mission in the form of a sequence of way points. It monitors the vehicle's progress and "checks off" the way points as each one is achieved. The Waypoint Manager sends the next way point to achieve to the Global Planner. The Global Planner generates an initial route to the way point using the prior map data. This map data may include just mobility cost data (e.g., obstacles) or it may also include tactical data (e.g., concealed regions for stealth). If no prior map data is available, the Global Planner uses a map initialized to uniform cost.

With an initial route planned, Crusher begins driving. Its sensors scan the terrain in front and to the sides of the vehicle for obstacles. The Near-Range Perception System (NRPS) is the module primarily responsible for obstacle detection, operating out to about 20 meters (i.e., greater than the vehicle stopping distance). The

NRPS receives position- and time-tagged colored lidar points from the sensors and aggregates them into a three-dimensional voxel map. The NRPS classifies each voxel in the vicinity of the vehicle as a positive obstacle, vegetation, or ground surface and assigns a mobility (i.e., hazard) cost to the corresponding terrain cell. The processing time varies as a function of the type of obstacle detected, typically ranging from 50 ms to 250 ms. The NRPS sends the changed cells to Map Fusion at a rate of 10 times per second.

Map Fusion is responsible for fusing hazard data from multiple sources into a single, consistent representation and providing that data to the modules that use it. Using a local (i.e., relative) position estimate for the vehicle, Map Fusion accumulates the NRPS hazard costs into a local hazard map. Using a global (i.e., absolute) position estimate for the vehicle, Map Fusion fuses these local hazards with the global map data. For a given terrain cell, the fusion rule selects one estimate over another based on factors such as age and resolution of the data, obstacle type, viewing perspective, and distance from the sensor. Map Fusion sends the changed cells to both the Local Planner and the Global Planner at a rate of 20 times per second. In the latter case, Map Fusion combines the hazard data with the tactical data before sending.

The Global Planner is responsible for providing a route to the Local Planner that is current given all map and perception data. The Global Planner repeatedly updates its copy of the prior map with fused hazard costs and re-plans the route from the vehicle's current location to the next way point. The Global Planner stores its map information at the resolution of NRPS (i.e., 20 cm), so that no information is lost if/when the vehicle must plan through a previously driven area. Given the nature of the algorithm, the Global Planner calculates paths from most cells in the vicinity of the vehicle. It sends changed strategic (i.e., path) costs to the Local Planner at a rate of once per ten seconds up to ten times per second, depending on the number of updated cells and the difficulty of the terrain.

The Local Planner is responsible for driving the vehicle around hazards. It receives hazard data from Map Fusion and stores it in a scrolling map centered on the vehicle. At a rate of 5 to 20 times per second, the Local Planner simulates the vehicle driving along a set of candidate trajectories and scores each trajectory based

on the hazards encountered. Each score is added to an estimate of the remaining path cost provided by the Global Planner, and the trajectory with the best score is selected for driving. The candidate trajectories are either circular arcs or S-turns (each is at least as long as the stopping distance of the vehicle) and may include backward driving in addition to forward driving, depending on the difficulty of the terrain encountered. For the simulation, the Local Planner uses a higher fidelity vehicle model than that used by the Global Planner. Typically, the vehicle drives a small fraction of a trajectory before another is planned.

The Local Planner sends a steering arc and speed to the Command Executor to drive the selected trajectory. The Command Executor is responsible for actuating the wheels on the left and right sides of the vehicle to implement the selected steering angle and speed. The Command Executor cycles at a rate of 20 times per second.

Finally, the Far-Range On-Line Learning (FROLL) module extends the range of perception to 70 meters in front of the vehicle. At that distance, the data is sparse, blurry, and difficult to interpret. Rather than program algorithms to interpret far-range data, we developed learning algorithms that use Near-Range Perception System results as ground truth. FROLL stores the far-range data for a given patch of terrain and learns how to interpret it once NRPS is close enough to observe it. (NRPS sends its interpretations to FROLL.) FROLL then applies this learned interpretation to new far-range data. At a rate of once per second, FROLL sends its hazard costs to Map Fusion for the range 20 to 70 meters in front of the vehicle. FROLL can similarly interpret far-range stereo data as well as overhead map data using NRPS hazard data as ground truth and send the resulting hazard costs to Map Fusion.

## **Hardware Architecture**

The Crusher autonomy system has high computational requirements in order to support high-speed driving and rapid, intelligent decision making in complex off-road terrain. Fortunately, much of the software architecture is decomposable into separately executing modules, pipelined (e.g, NRPS) or running concurrently (e.g., local and global planning). We chose a hardware computing system to support this software modularity. The autonomy computing system consists of a ladar controller machine and a blade server. The ladar con-

troller consists of a 2 GHz Pentium M processor, 2 GB of memory, Ramdisk boot, and a dual gigabit Ethernet interface. The blade server consists of eight dual-processor dual-core blade machines, AMD Op-  
 teron 200 series 2.4 GHz Model 280 processors with 1 MB on-die cache, 4-8 GB of memory on each blade, central 1 TB RAID 6 disk for logging of high bandwidth data, ramdisk boot, and dual gigabit Ethernet inter-  
 face.

Figure 5 shows a mapping of the autonomy software system modules onto the computing hardware. Also shown in the figure are the number of CPU cores that the software modules on that board consume and the total amount of memory that the modules use. Each of the camera blade machines interfaces to two cameras, while the ladar controller machine interfaces to all of the SICK ladar sensors.

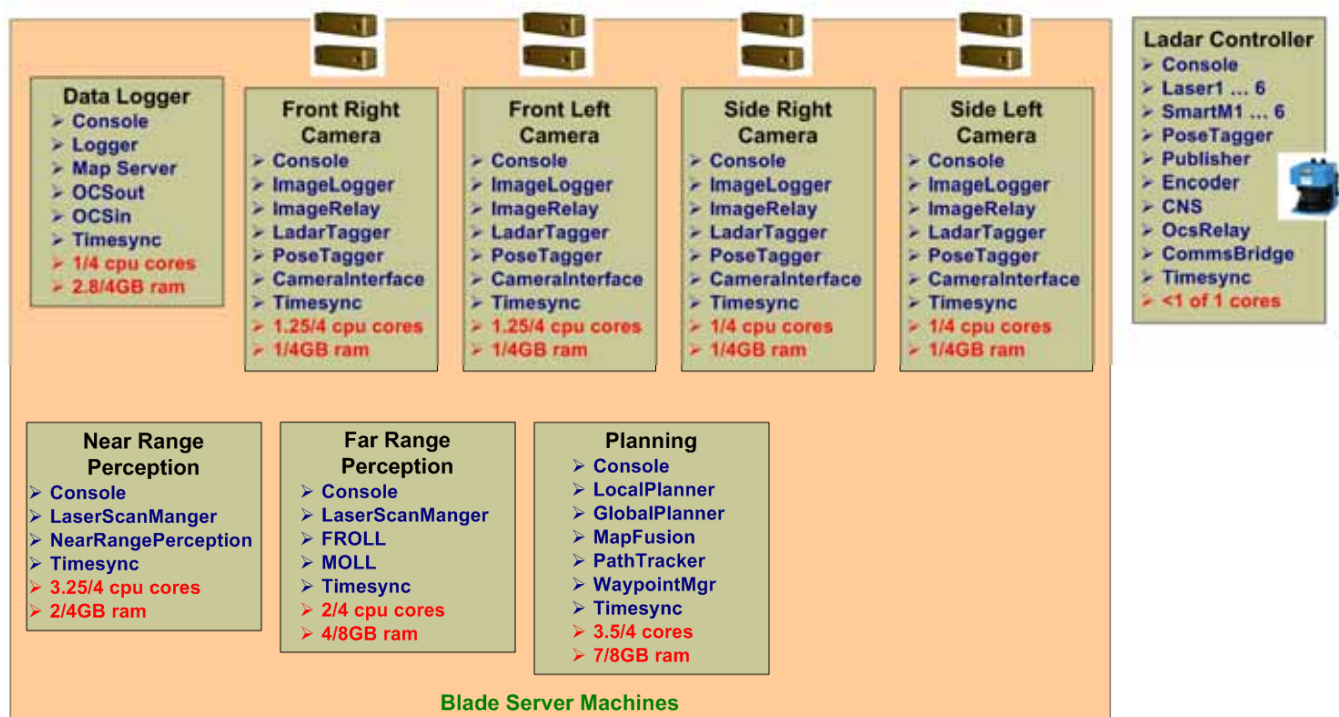


Figure 5: Mapping of Crusher autonomy system software modules onto the computing hardware.

## EXPERIMENTAL RESULTS

Crusher participated in unrehearsed quarterly field experiments at sites in Washington State, Texas, Colorado, and Pennsylvania on complex terrain with steep slopes, washes, ravines, trees, rocks, and thick vegetation. Prior to each field experiment the government team spent several days scouting the terrain to identify numerous challenging test courses to run during the two week long field experiment. These courses were not

disclosed to the Crusher team until it was nearly time to run the course—thus precluding any opportunity to practice or to tune the system to the specifics of a given course. We present results from the experiment performed at Fort Bliss, Texas in January/February of 2007.



**Figure 6: Typical terrain found at Ft. Bliss.**

For this experiment, the government team tracked four metrics during each of the course runs: total distance traveled, total time elapsed, adjusted speed, and number of operator interventions. The total distance traveled is the sum of the distance traveled in the forward and reverse directions. The elapsed time is the total time from the start of the course until completion and includes all pauses at the way points (i.e., to plan the next route). The adjusted speed is the total forward distance traveled divided by the elapsed time. This is not equivalent to the average speed of the vehicle during the run since the adjusted speed penalizes the system for backing up (i.e., counts the time but not the distance). The number of operator interventions is the number of times the field safety operator or the autonomy control station operator had to intervene to get the vehicle out of trouble.

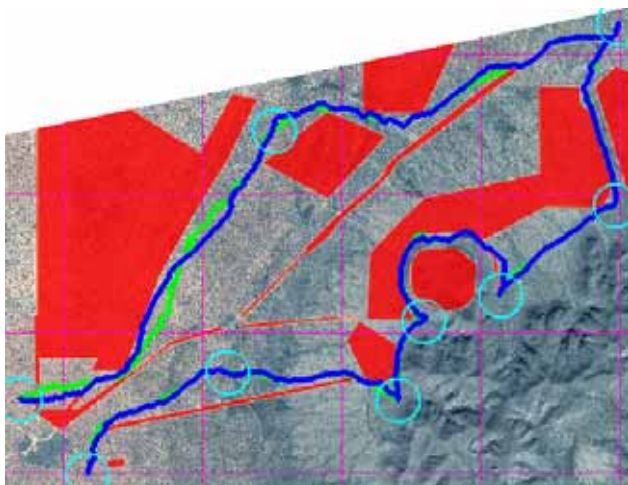
Over an eleven day experiment, Crusher performed 119 course runs (on 6 different courses) at Fort Bliss. The autonomy metrics for all runs combined resulted in a total distance traveled of 258.44 km (160.59 miles) with an adjusted speed of 3.14 m/s (7.03 miles/hour) while averaging 1.0 intervention per 10 km of travel. In the figures below, we illustrate results from three runs on a 20 km course. As shown in Figure 8, the course consisted of eight way points that ranged from 400 to 1500 meters apart. The terrain included various types

of vegetation, numerous wash crossings, navigation over a saddle point between two hill tops, and navigation through a berm wall. Figure 7 is a sequence of images that shows Crusher crossing one of the washes on the course. The vehicle was not given any speed limits or safety corridors to navigate within, that is, the vehicle was allowed to select a safe speed and to choose a suitable route to each way point.

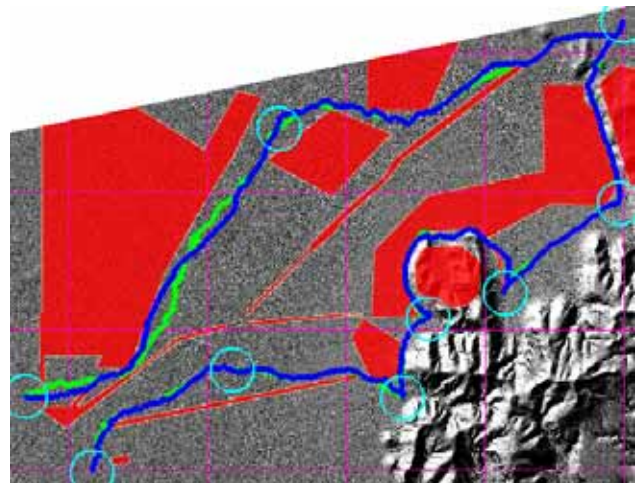
Figure 9 shows the data for all three runs performed on this course. Between the runs, we varied only the resolution level of Digital Terrain Elevation Data (DTED) map in order to gauge the effect resolution has on performance. The only performance difference between runs was an intervention that occurred during the DTED Level 3 (10 meter) run at one of the larger wash crossings. When using the DTED Level 4 and 5 (3 and 1 meter respectively) maps, Crusher drove down a gentle slope to get through the wash. Conversely, when using the DTED Level 3 map, Crusher drove directly at the steep part of the wash. The vehicle approached the steep wash slope, and the safety operator took control immediately before the autonomy system instructed the vehicle to back away from the wash. This resulted in the only intervention for the run.



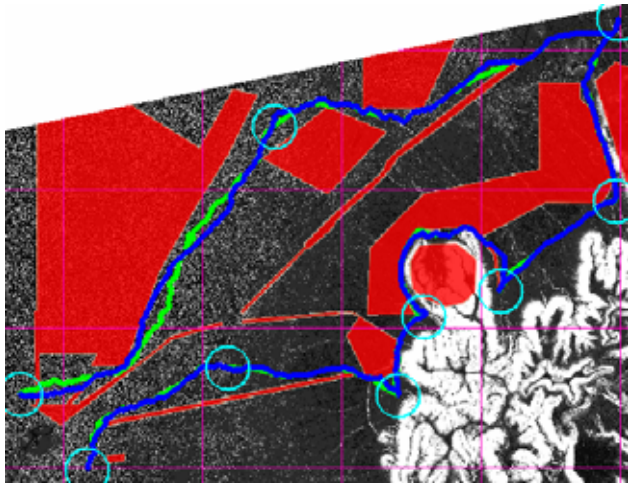
**Figure 7: Crusher autonomously crossing one of numerous ditches on the 20 km course.**



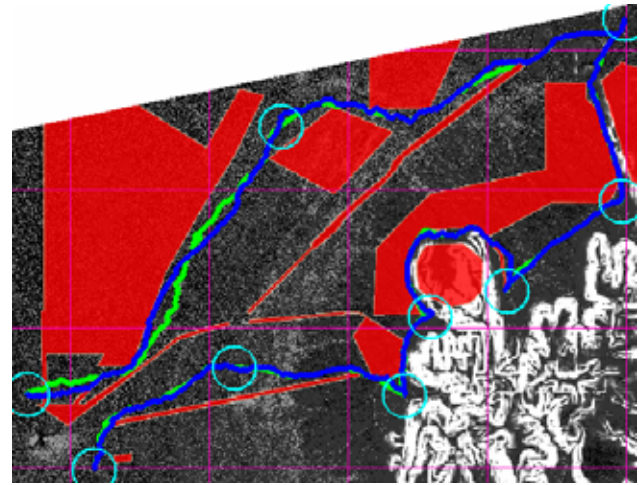
(a) paths overlaid on satellite imagery



(b) paths overlaid on DTED 5 shaded relief image



(c) paths overlaid on DTED 5 Engineered cost map



(d) paths overlaid on DTED 3 Engineered cost map

**Figure 8: Example of different routes driven on the 20 km course: DTED 5 (green), DTED 3 (blue). Red zones are tactical keep out zone for this course. The course runs from far left waypoint clockwise around to lower left waypoint.**

Run ID	Prior Data Settings	Total Distance (m)	Total Time (sec)	Adjusted Speed (m/sec)	Number Of Interventions
L-1	DTED 5 Eng	20485.50	5817.30	3.52	0
L-2	DTED 4 Eng	20423.90	5665.60	3.60	0
L-3	DTED 3 Eng	20801.91	6183.86	3.36	1

**Figure 9: Autonomy settings and metrics for the various runs on the 20 km course.**

## CONCLUSIONS

As indicated by the Ft. Bliss experiment, the Crusher system has matured to the point where it is able to drive nearly fully autonomously on complex terrain containing rocks, washes, ditches, steep slopes, bushes, and other vegetation. We continue to advance the technology, with an emphasis on adaptive approaches, to push Crusher into even more complex terrains.

## ACKNOWLEDGMENTS

The authors thank the entire UPI project team, including Drew Bagnell, Michael Bode, Roger Boulet, David Bradley, Charles Corr, Constantine Domashnev, Cris Dima, Colin Green, Paul Haser, Alonzo Kelly, Joseph Manojlovich, Eric Meyhofer, Clifford Olmstead, John Olson, Donald Salvadori, Michael Sergi, Mark Si-

benac, David Silver, Boris Sofman, Mark Waldbaum, and Carl Wellington. We also thank our DARPA sponsors Larry Jackel, Mike Perschbacher, and Jim Pippine.

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract "Unmanned Ground Combat Vehicle - PerceptOR Integration" (contract number MDA972-01-9-0005). This work used component technologies sponsored by the Army Research Laboratory, NASA, and the National Science Foundation under separate projects. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

## REFERENCES

- [1] Albus, J.S., "4D/RCS: a Reference Model Architecture for Intelligent Unmanned Ground Vehicles," Proc. of SPIE Aerosense Conference, 2002.
- [2] Arkin, R.C., "Behavior-Based Robotics," The MIT Press, Cambridge, Massachusetts, 1998.
- [3] Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., Happold, M., Herman, H., Mandelbaum, R., Pilarski, T., Rander, P., Thayer, S., Vallidis, N., Warner, R., "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," International Journal of Robotics Research, vol. 25, issue 5-6, May-June 2006.
- [4] Stentz, A., and Hebert, M., "A Complete Navigation System for Goal Acquisition in Unknown Environments," Autonomous Robots, vol. 2, no. 2, August 1995.