

MANIPULATOR AUTONOMY FOR EOD ROBOTS

Josh Johnston, Joel Alberts, Matt Berkemeier, John Edwards
Autonomous Solutions, Inc
Petersboro, UT, 84325

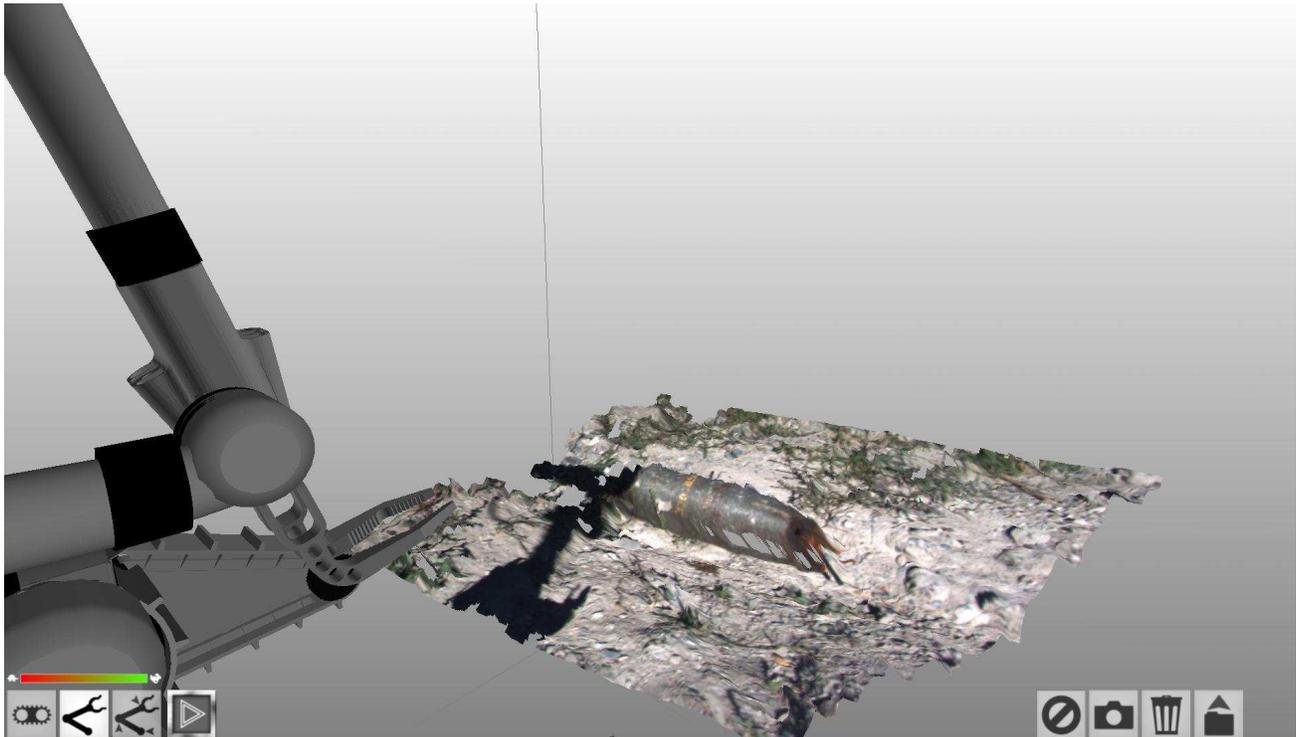


Figure 1: 3D rendering of a partially-buried artillery shell.

ABSTRACT

This paper presents research that enhances the effectiveness of Explosive Ordnance Disposal (EOD) robots with autonomy and 3D visualization. It describes an approach where autonomous behaviors like Click and Go, Drag for Wire, and Click and Grasp are rapidly formed by combining foundational technologies like Resolved Motion, Inverse Kinematics, and 3D Visualization. Also presented is a flexible, JAUS-based architecture that supports new autonomous behaviors and future work that applies these manipulation advancements to mobility and navigation.

1. INTRODUCTION

Increasing the level of manipulator autonomy is imperative for improving the effectiveness of EOD robots, but there are many potential autonomous functions and many different ways of implementing them. The specific implementations of autonomy (ie, what

information does the human receive, what information does the robot receive from the human, and what does the robot do as a result) vary greatly, but all depend on a few key technologies which allow for numerous implementations.

Under current programs with the Navy's EOD Technology Division (NAVEODTECHDIV), US Army TARDEC, and the Navy SPAWAR, ASI has advanced the state of the art in 3D visualization and resulting arm control and demonstrated 'fly the gripper' functionality integrated with 3D visualization (Figure 1) as well as 'click and go' functionality on an EOD Packbot allowing the user to command the gripper to an object in 3D space.

This paper presents research demonstrating the following integrated technologies on EOD-class platforms:

- Resolved motion routines (enabling 'fly the gripper' and coordinated manipulation and mobility),
- Technology for sensing and visualizing the robot's 3D environment

- Autonomous ‘click and go’ technology allowing a user to indicate a point in 3D space to which the manipulator is to travel (ie, rapidly and accurately approaching manipulation targets),
- Autonomous pick and place technology for robotic manipulators.

These user-level functionalities depend on more basic underlying technologies which enable a host of other future functions. This paper includes a description of a software architecture enabling future integration and open development of manipulator autonomy.

Though mostly covering demonstrations that showcase EOD applications, this paper will also present route-clearing combat engineering applications currently under construction.

1.1 Teleoperation, Full Autonomy, & Partial Autonomy

A fully autonomous robot is conceived as one which can respond to very high level human-like commands (ie, ‘pick up that can over there’) consistently and correctly under any conditions. By contrast, a teleoperated robot is one in which every robotic movement is directly commanded by a human operator. There are many advantages of a hypothetical ‘fully autonomous’ robot and one way of illustrating them is shown in Figure 2

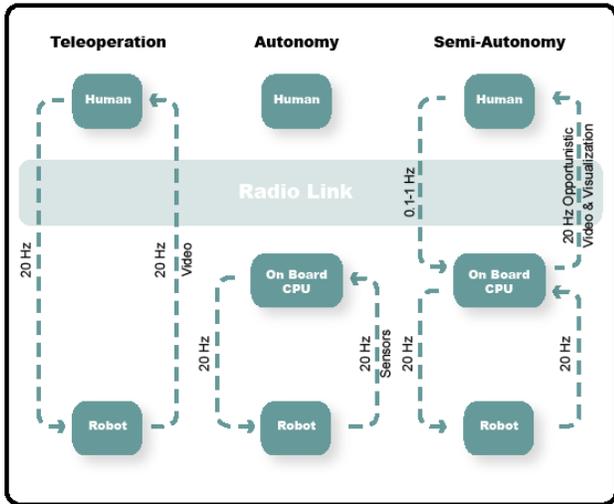


Figure 2: Teleoperation vs Full Autonomy vs Semi-Autonomy. In this perspective, teleoperation requires the human to close the control loop on the robot’s actions, demanding his full attention to respond constantly and rapidly. Full autonomy moves the closed loop control to the robot side, freeing up the human’s time and attention. Semi-autonomy still requires human oversight and intervention, but at a much lower rate since some lower-level actions are completed by the robot autonomously.

From the perspective of Figure 2, it is clear that a fully autonomous system places the fewest demands on

the operator, who is now no longer required to ‘close the control loop’ through the RF link based on limited 2D visual information. Full autonomy also places the least stringent demands on the RF communications system, a constant problem in a military environment.

Unfortunately, full autonomy in an arbitrary environment with a rich command set is still at a very low TRL. Full manipulator autonomy depends on many underlying technologies, including such higher-TRL technologies as fast, high-resolution 3D sensing, dynamic controls, complex path planning, and often lower-TRL technologies such as various geometry recognition and scene understanding algorithms. Some of these underlying technologies are at higher TRL’s than others, which enables Autonomous Solutions to implement some autonomous behaviors, thus achieving ‘semi-autonomy’. Implementing semi-autonomy lowers the demands on the operator, reduces dependence on RF linkage, and improves system performance.

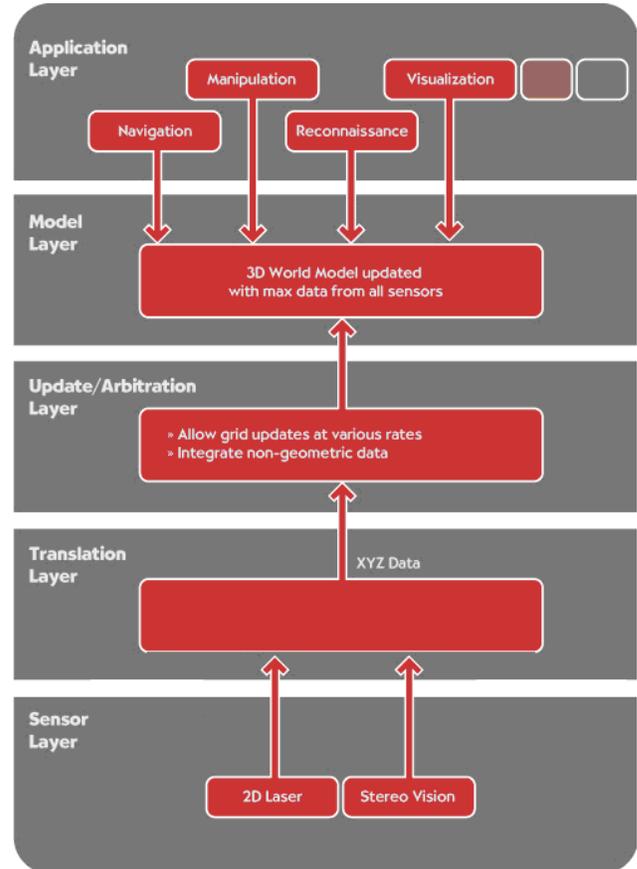


Figure 3: Autonomous Solutions’ 3D visualization and automation architecture. Sensor data is translated, merged, and stored in a repository that is available to multiple consumers simultaneously. This means that visualization and automation applications can operate on data from the same sensors.

1.2 Architecture for Autonomy

Autonomous Solutions uses a layered software architecture shown as Figure 3 which translates the raw data from an arbitrary set of sensors into a 3D world model through a standard interface. This 3D world model is available to an arbitrary set of high level autonomy applications through another standard interface. In this way, perception sensors and autonomous software applications are ‘plug and play’.

The lowest architecture layer contains the raw sensors available for robotic perception, like stereovision, planar lidar, and flash lidar. These sensors provide data with different coordinate frames, formats, densities, update rates, and other parameters. The next higher layer, the translation layer, translates the data from each sensor into a common format, such as an xyz point cloud. The update/arbitration layer decides how and when the data from each sensor gets used to update the full 3D world model. The model layer contains the most recently updated full model available from the sensor data in a standard format. It is accessed by a variety of applications which implement autonomous behaviors, such as those shown.

The benefits of this approach are:

- Any set of sensors can be used to collaboratively create a single 3D world model
- Sensor sets can be chosen to optimize a common model, allowing for comparison and optimization against a common standard
- Common control and autonomy applications (such as navigation algorithms) can be made independent of the particular sensor set used
- Both sensors and autonomy applications, being independent of each other, can be replaced as technology is improved or as applications require.
- The application layer can exist off-robot on an Operator Control Unit (OCU) which minimizes requirements for the vehicle processor.
- Sending 3D points and still camera images across a radio link requires less bandwidth than a video channel.

2. FLY THE MANIPULATOR

Operating a manipulator with several degrees of freedoms is a complex task that becomes cumbersome when controlling each joint individually. To ease the demands of the user, a “fly-the gripper” mode of arm operation is developed and implemented on the robot. In this mode, the user is able to specify x , y , and z gripper velocity components using a joystick, and joint velocities are computed and then executed to achieve the

desired gripper velocity. In order to accomplish this, the forward kinematic map and Jacobian matrix for the arm need to be computed.

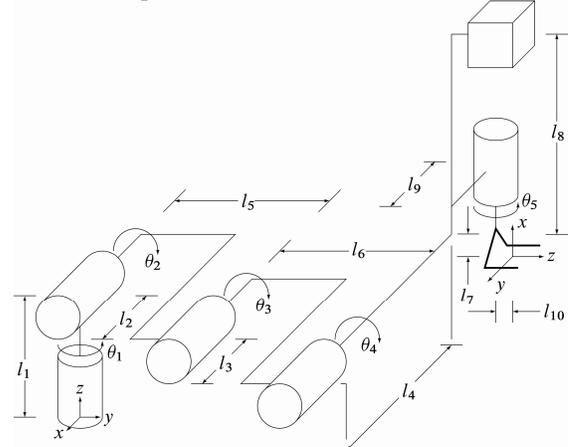


Figure 4: Packbot arm configuration. The arrows indicate the directions the angles increase.

Autonomous Solutions has developed “fly the manipulator” algorithms for several platforms including an iRobot PackBot under funding from NAVODTECHDIV. A schematic of the PackBot arm configuration is shown in Figure 4. The arm is shown in its reference configuration which means all joint angles are zero.

2.1 Forward Kinematics

The product of exponentials method (Murray, et al 1993) was used to determine the forward kinematic map for the PackBot arm. Since the arm has five degrees of freedom, the product of exponentials formula takes the form

$$g_{bt}(\theta) = e^{\xi\theta_1} e^{\xi\theta_2} e^{\xi\theta_3} e^{\xi\theta_4} e^{\xi\theta_5} g_{bt}(0)$$

with b denoting the base frame and t denoting the tool frame.

For fly the manipulator, the interest is in the position of the gripper; rather than its orientation. The gripper position is given by the rightmost column of the g matrix. These expressions are simplified slightly by setting $\theta_4 = \theta_5 = 0$ and $l_7 = l_{10} = 0$. More explicitly,

$$p = f(\theta) = \begin{bmatrix} l_{2349}c_1 - s_1(l_5c_2 + (l_{10} + l_6)c_{23} - l_7s_{23}) \\ l_{2349}s_1 + c_1(l_5c_2 + (l_{10} + l_6)c_{23} - l_7s_{23}) \\ l_1 - l_7c_{23} - l_5s_2 - (l_{10} + l_6)s_{23} \end{bmatrix} \quad (1)$$

where,

$$s_i = \sin \theta_i, s_{ij} = \sin(\theta_i + \theta_j), c_i = \cos \theta_i, c_{ij} = \cos(\theta_i + \theta_j)$$

$$l_{2349} \text{ is shorthand for } l_2 + l_3 - l_4 - l_9.$$

2.2 Jacobian Matrix

The gripper position is given by the 3×1 vector $p = f(\theta)$, where θ represents the joint angles $\theta_1, \theta_2, \theta_3$ (Equation 1). Let v be the gripper velocity. Differentiating f provides the map from joint velocities to gripper velocities:

$$v = \frac{\partial f}{\partial \theta}(\theta) \dot{\theta} = J(\theta) \dot{\theta}$$

where the element in the i th row and j th column of $J(\theta)$ is given by $\frac{\partial f_i}{\partial \theta_j}(\theta)$

$J(\theta)$ is a 3×3 matrix, which is a function of the joint angles. To determine joint velocities from Cartesian velocities, one must solve for the pseudo inverse matrix of the Jacobian [Strang 1988]: $\dot{\theta} = J^+(\theta)v$

Using the pseudo inverse, the joint velocities can now be determined from the desired gripper Cartesian velocities. With this implementation, operators can perform manipulator tasks more easily and rarely need joint by joint control. This simple control method also complements the usefulness of the OCU visualization discussed in Section 3.

3. 3D VISUALIZATION

Autonomous Solutions has developed a suite of sensors and software to enable an EOD technician to have a real-time three-dimensional (3D) view of the target environment and the robot's position and orientation in it (Figures 1, 5, 6, and 7). This view can be manipulated by the user at video frame-rates to observe the robot and its environment from any angle and distance.

After the 3D model of the world is generated, the rendered robot position can be updated using information from a pose source like GPS or odometry. This telemetry is much smaller than streaming video but still enables teleoperation of the vehicle, including obstacle avoidance. Therefore, the 3D modeling is a way to continue operating the vehicle despite lapses in high bandwidth radio communication. When extra bandwidth is available, the 3D model can be extended and updated.

The research has identified several methods of displaying texture on 3D models.

3.1 Point Cloud Visualization

The simplest and fastest 3D visualization method uses point sprite particle visualization. Each 3D point returned from the sensors is assigned a color. The color is

either passed with the point if a stereo vision camera is used, or is assigned using mapping from a single camera image if the point contains no inherent color. Each sprite is rendered with its one color, regardless of the size at which it is rendered on the screen. Using sprites, a point cloud of up to and exceeding one million points can be visualized at video frame-rates.



Figure 5: ASI's Packbot modified for 3D visualization.

The point cloud visualization is effective when the 3D data density is about the same as the texture resolution and the operator doesn't need to zoom too close to the surface. Additionally, if the sensor data is updated frequently, such as is the case with lidar sensors capturing data at rates of 100,000 points per second or more, then the point cloud visualization method gives the best capture-to-display rate.



Figure 6: ASI's stereovision system mounted to the TALON robot using a quick-release bracket (silver camera on left of image).

When visualizing the data from a distance, the point sprite method effectively approximates a surface visualization. When viewing at large scales, however, the individual points begin to visibly resolve.



Figure 7: ASI's 3D visualization on a TALON OCU (upper left quad view). All processing is performed on the robot and the 3D model is displayed as a video stream that replaces a camera

3.2 Triangulation

ASI has developed a novel triangulation method that more closely approximates the physical surfaces being sensed without sacrificing frame rate. When the 3D points are captured by the sensors, triangles are generated using an optimized raster-walking algorithm. Both color and spatial coordinates are used as arguments to the decision function which determines whether sequences of three points lie on a physical surface or not. The triangles are displayed with a two-dimensional (2D) color raster image projected onto them. Once the triangles are generated and initially displayed, the visualization system can easily display them at interactive frame-rates.

The triangulation method displays triangles at near frame-rate speed after the initial calculations. It is optimal for cases in which a higher quality display image is desired or when sensor data is not being collected at high rates.

3.3 Quad Visualization

The third visualization option is similar to point cloud display, but uses quads with image projection. Using this technique, each quad displays a piece of the image, rather than a single color as with the point cloud technique.

As computation of each quad's texture is relatively expensive, a level of detail (LOD) control using octrees limits the number of quads shown at a given time.

The quad visualization method approximates a surface display when viewed at a distance. With low resolution 3D points, this surface can look better than point clouds since the quad size can be increased without significantly increasing the "blocky" look of the data. However, the cost of display, even using decimated data, is higher than that of point clouds. The capture-to-display rate is better using quads than triangles, but the subsequent interactive frame-rate of quad display is significantly lower than triangles after the overhead triangulation step is complete.

4. AUTOMATED MANIPULATION

4.1 Click and Go To

Interacting with the 3D world using a 2D interface has traditionally meant that depth or distance cannot be indicated. Methods such as visual servoing can approach a target, but the operator cannot determine the distance to that target or discriminate between conflated objects, such as when a further object is occluded by a closer one.

ASI has developed a technique to select locations and objects in 3D space using a 2D interface. The user views a camera image or an arbitrary perspective of the sensed 3D data. This flat representation is a mathematical projection of the 3D world into the 2D image plane. Therefore, when a user clicks on a location in this flat view, it expands into a line perpendicular to this view in 3D space. The different points along the line represent the depth ambiguity of this selection (see Figure 8).

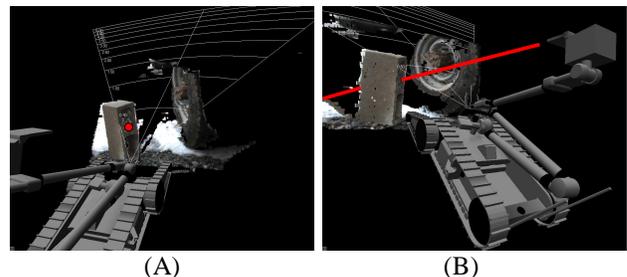


Figure 8: The point selected in (A), shown by a red dot, is not a unique position in 3D space. The set of possible points corresponding to the selection is represented by the red line in (B).

With a 2D camera-based approach like visual servoing, this range ambiguity cannot be solved. Since ASI has a full 3D model of the world, however, there are two ways to fix the actual point of selection.

The intersection of a line and a surface is one or more points. Where the ambiguity line first intersects the world model surface is the best interpretation of the user's desired selection point, because the other potential intersections are occluded. ASI has found that this is the

best way to select a single point on the world model surface.

Often, it is desirable to select a volume instead of a single point. By selecting the same object in two or more views of the world, the user can indicate the precise 3D volume of interest using a 2D viewer.

dragging action along the ground in between. While the gripper is being autonomously dragged over the ground, the operator can devote full attention to video feedback to check if wires are caught or objects are moved.

This is another example of how a few underlying technologies enable a variety of possible behaviors, the

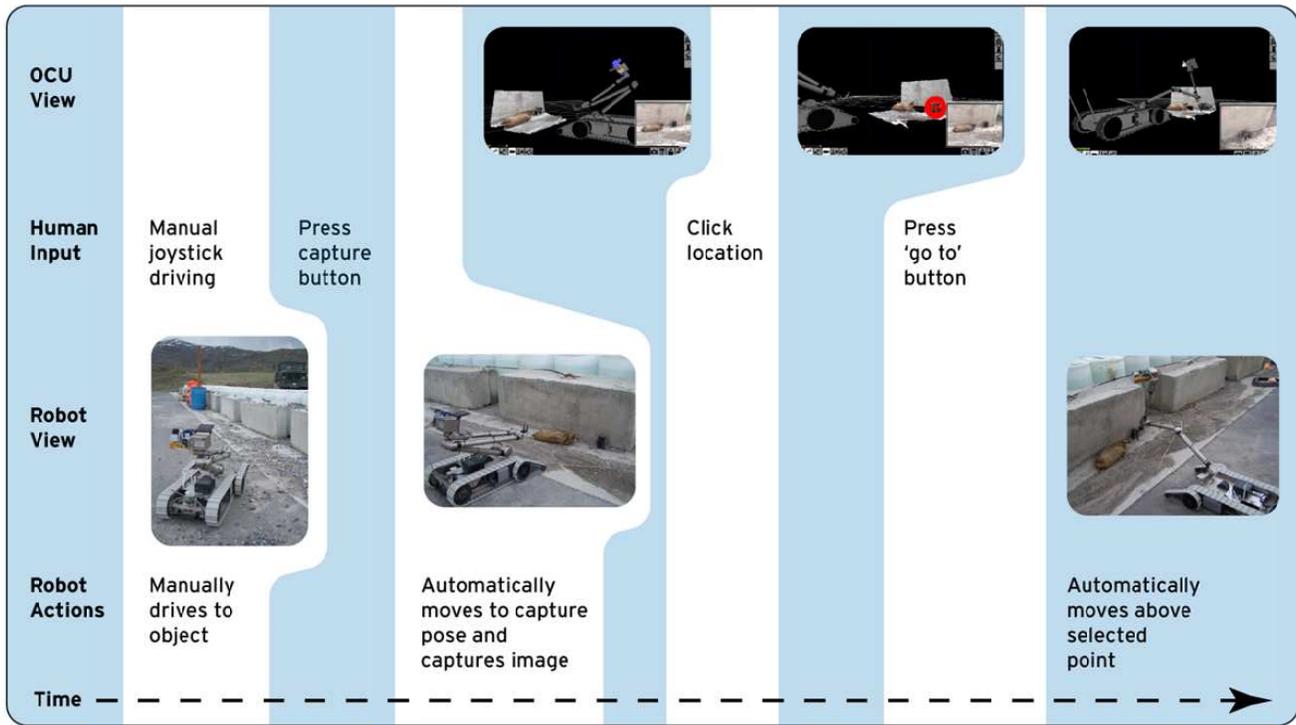


Figure 9. “Click and Go” manipulator implementation from the user’s perspective.

Click and go behavior is implemented by combining this 3D selection with the fly the manipulator technique (Figure 9). The user drives as normal to an object of interest and captures a 3D model of the relevant area. Clicking a point in the model indicates the user’s desired manipulator location, to which the gripper autonomously travels once the user presses the ‘go to’ button.

4.2 Click and Grasp/Click and Drop

These behaviors combine “click and go” with basic grasp planning to pick up or drop an object. They rely on the foundational technology of 3D visualization to indicate an object in full world coordinates as well as the inverse kinematic path tracker to achieve these positions and orientations.

4.3 Drag for Wire

Another semi-autonomous behavior implemented by ASI is “drag for wire” (Figure 10). The operator indicates two points on the ground and the robot executes a

specifics of which can be driven by needs peculiar to a given mission type. In this case, the behavior is dependent on 3D visualization to select the start and end points and fly the manipulator to autonomously follow the resulting path.

5. COORDINATED MANIPULATION AND MOBILITY

Under an SBIR Phase II contract with TARDEC, ASI has been developing a coordinating manipulation and mobility method using an omnidirectional platform previously developed for TARDEC (Figure 11). The purpose of coordinated control is to command the manipulator tool, with six degrees of freedom, using the three degrees of freedom in the robot and three degrees of freedom in the manipulator. Two manipulator joints will be locked for coordinated control.

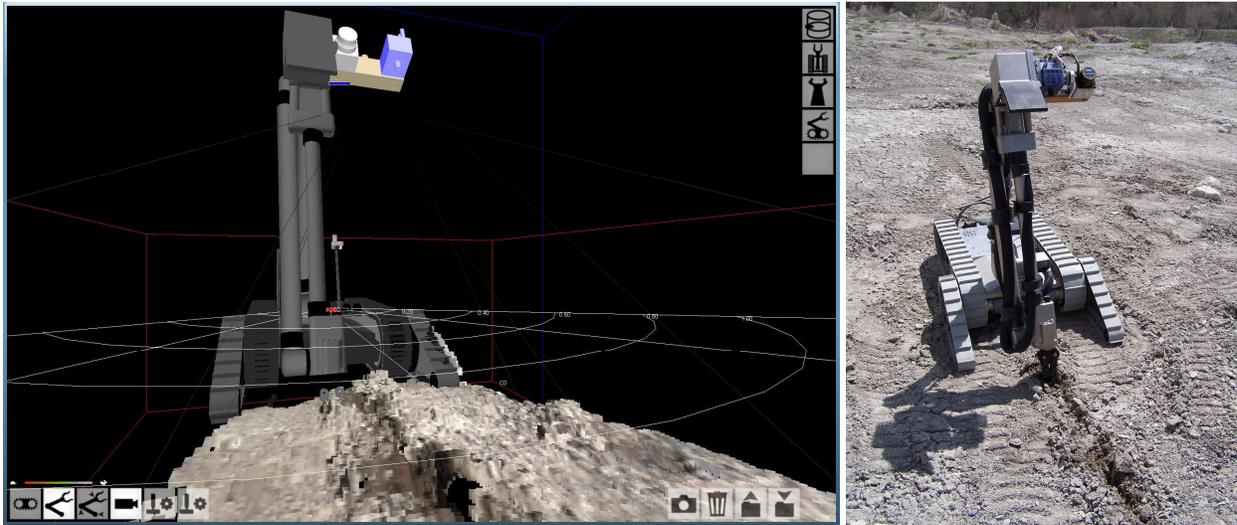


Figure 10: 3D visualization while digging. The rendering of the end effector and ground data verifies the depth of ground penetration

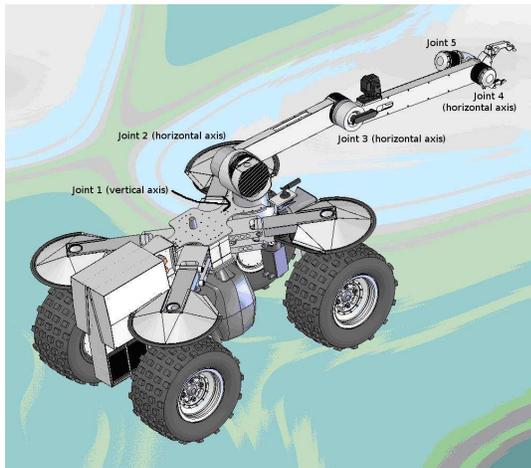


Figure 11: The Wile E. vehicle used for coordinated manipulation and mobility research

6. FUTURE WORK

Under funding from NAVEODTECHDIV and SBIR Phase II funding from TARDEC, ASI is developing further extensions to small robot control. One approach is 2D blueprint-style mapping of the environment using a small planar laser (Figure 12).

Another approach extends the 3D generation to full environments using an automatically detail-scaling display that will provide seamless zoom from full terrain (Figure 14) to local scenes (Figure 13). This system uses a Velodyne HDL-64 lidar system and a hemispherical camera.

This larger world will extend the “click and go” from manipulation (as described above) to mobility. The prior Digital Elevation Map (DEM) and long range of the lidar

(120m) will provide a 3D model that the operator can use to indicate a path downrange. As opposed to visual servoing, this path can contain curves and maneuvers to avoid obstacles. When it is time for a retrotraverse, the operator can indicate a path back through the world that was built while the vehicle drove downrange.

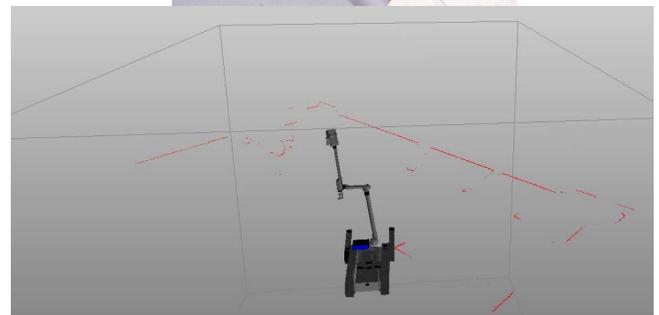


Figure 12: Mapping with a 2D Hokuyo laser scanner. This capability is added to the above 3D visualization to improve navigation.

Modeling the area of traversal in 3D will enable combat engineers to more effectively search for ordnance, clear routes, and determine terrain traversability.

Reducing the operator load will mean that the robot can operate for longer stretches while increasing the efficiency and speed of the task.

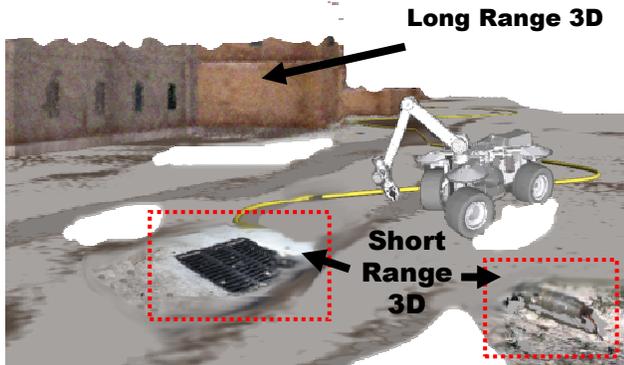


Figure 13 Conceptual diagram of a world building display.

CONCLUSIONS

With the proper architecture, diverse autonomous and semi-autonomous behaviors can be built atop certain foundational technologies. Adding behaviors or extending existing ones to new domains, like from manipulation to mobility, can be performed without revamping the enabling technologies.

There is great concern that we cannot predict the operational challenges of future EOD robotics missions and therefore cannot define current research priorities. As shown by the implementations in this paper, specific autonomous behaviors don't need to be selected to identify the foundational technologies they will require. Advancing the TRL of path planning, object recognition,

force feedback, and 3D model-building will enable future autonomous and semi-autonomous behaviors that will be quickly implemented in response to changing mission requirements.

ACKNOWLEDGEMENTS

The authors are grateful for support from TARDEC, NAVEODTECHDIV, and SPAWAR.

REFERENCES

- Berkemeier, M., Poulson, E., Aston, E., Johnston, J., Smith, B., 2008: Development and Enhancement of Mobile Robot Arms for EOD Applications, *Proc. of SPIE*, Vol. 6962, 69620P-69620P-12.
- Gonzalez, R., and Woods, R. *Digital Image Processing*, Addison-Wesley Publishing Company, 1992, pp 518 - 519, 549.
- Murray R., Li, Z., and Sastry, S.. *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1993.
- Strang, G. *Linear Algebra and Its Applications*. Harcourt Brace Jovonich College Publishers, 1988.

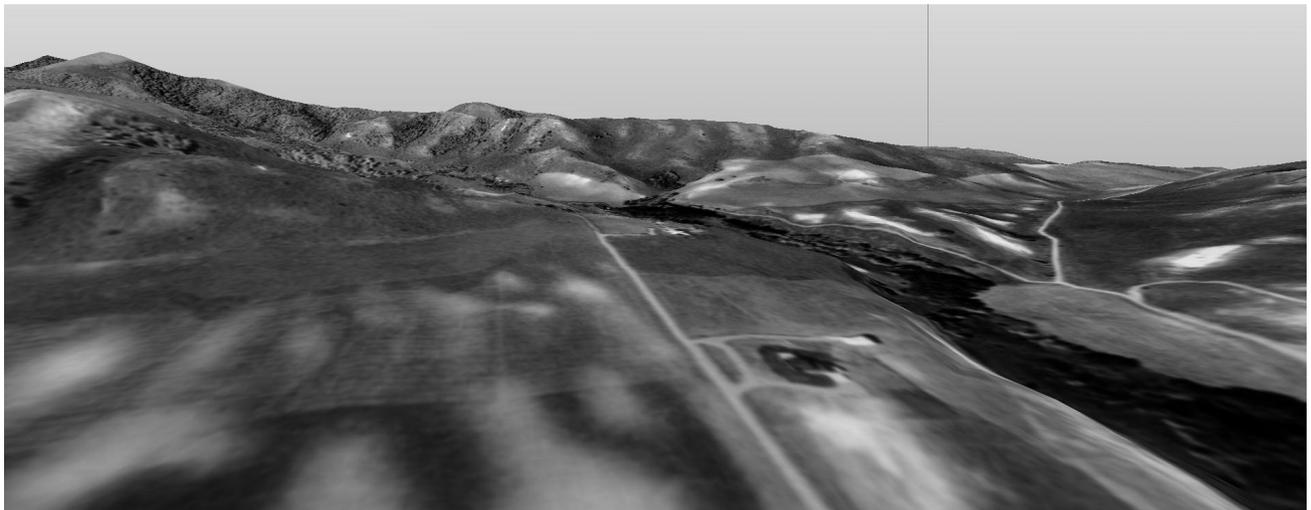


Figure 14: Terrain visualization of USGS DEM with applied Landsat full-spectrum imagery. Using a new octree-enabled technique, zooming will be possible between this level of detail and the robot's immediate vicinity