Reduction of User Interaction by Autonomy

Arin Morfopoulos, Michael McHenry, and Larry Matthies

Jet Propulsion Laboratory
4800 Oak Grove Drive, Pasadena, CA 91109
{arin.morfopoulos, michael.mchenry, larry.matthies}@jpl.nasa.gov

Abstract

This paper describes experiments that quantify the improvement that autonomous behaviors enable in the amount of user interaction required to navigate a robot in urban environments. Many papers have discussed various ways to measure the absolute level of autonomy of a system; we measured the relative improvement of autonomous behaviors over teleoperation across multiple traverses of the same course. We performed four runs each on an "easy" course and a "hard" course, where half the runs were teleoperated and half used more autonomous behaviors. Statistics show 40-70% reductions in the amount of time the user interacts with the control station; however, with the behaviors tested, user attention remained on the control station even when he was not interacting. Reducing the need for attention will require better obstacle detection and avoidance and better absolute position estimation.

Keywords: User interaction, attention, autonomy, robot.

1. Introduction

The usefulness of mobile robots is a strong inverse function of the amount of user interaction required to control them; hence, the value of autonomous behaviors is partly a function of how much they reduce the amount of user interaction required. Thus, quantifying user interaction as a function of the available robot control modes is necessary for measuring progress. Quantitative user interaction experiments can also reveal where existing robot behaviors have trouble dealing with the environment, and therefore are useful for prioritizing further development.

In this paper, we describe results of user interaction experiments with a Packbot robot equipped with stereo cameras, a single axis scanning LADAR, and a variety of behaviors ranging from teleoperation to waypoint following with obstacle avoidance. We counted user button clicks and mouse drags in a series of trials employing different behaviors over two courses, an "easy" course and a "hard" course. Two trials on each course used just teleoperation, two used more autonomous behaviors. We did not measure the amount of time the user was watching the control station (ie. user attention) when he was not physically interacting with it. This is an important additional step needed in future work. As we will show, however, current autonomous capabilities still require considerable user attention even though user interaction *per se* may be small, because robots are likely to get in trouble if the user is not watching an image stream from the robot.

The rest of this paper is organized as follows. First, section 2 reviews related prior work. Then we describe our experimental setup and how we collected the data (Section 3). The actual data collection follows this with all the attendant graphs and tables (Section 4). We use these results to highlight key areas where more development is needed to reduce both user interaction and user attention. The most significant areas are position estimation and path planning capabilities that enable autonomous traverses beyond line of sight from the robot (Section 5) and obstacle detection and avoidance capabilities that can cope with negative obstacles and moving objects, such as cars and people (Section 6).

2. Related Work

Goodrich, Crandall et al [1,6,7,8] have done a series of user interaction and attention studies. Their data lies chiefly in robotic simulation, which allowed them to force the user's attention to another task and to quantify the effect of user inattention upon performance. They call this "neglect tolerance". We could not afford to quantify this, because real robots currently face too much risk of damage from unseen obstacles and moving traffic when the user is not attending to downlinked imagery. Goodrich, Crandall et al have also defined a general model of teleoperation, waypoints, and scripted waypoints in terms of interaction vs. performance. This is a theoretical model that closely describes the behaviors we use. Teleoperation requires constant interaction whereas waypoints require more work to initiate, but once started can be left alone for a much longer period.

Tejada [3] discussed a model for a 3-D graphical user interface to use with urban search and rescue teams. Our experiment did not quantify the effect of different Operator Control Units, since our focus was on the effect of different behaviors in the robot.

Frost [4] discussed the difficulties of pure teleoperation, but only as general observations. No measured times to complete a course or accomplish an objective are given. This paper is particularly relevant because it describes the same robotic platform with which we conducted our tests. The autonomy used was entirely different, but the chassis and thus mechanical ability was the same.

Bruemmer [5] used a robotic system to work in a nuclear disposal facility. Their work described the difficulties of a teleoperated system using only visual feedback for control information. They were forced to place cameras inside each room to have enough data to allow safe navigation because the view from the robot's cameras was not sufficient. As a result of these difficulties they devised an autonomous system to assist in control and to prevent the user from endangering the robot or the environment. This is an excellent example of where it would be valuable to quantify the benefit of new behaviors. They tried teleoperation, found it lacking, created autonomy to assist, but have not yet measured the improvement enabled by that autonomy.

3. Experiment Setup

The robotic platform in our experiments ("Urbie") was an iRobot Packbot chassis with an electronic payload developed by JPL under the Tactical Mobile Robotics (TMR) and Mobile Autonomous Robot Software (MARS) programs sponsored by DARPA. Urbie's sensors included a black and white stereo camera pair, a SICK LADAR, 3-axis accels and gyros, a magnetometer with pan/roll/tilt axes, and track encoder feedback. Robot state was maintained by a Kalman filter which estimated the orientation (roll/pitch/yaw) of the robot. The KF orientation estimate was combined with the track encoder data to produce an estimate of robot x/y position via dead reckoning.

Behaviors used included visually designated waypoint following, stairclimbing, and teleoperation. An Obstacle Detection and Avoidance (ODOA) module could be dynamically incorporated via an arbiter system to make any of the above behaviors become Safeguarded. To use "visually designated waypoint following" a user selected a heading from the robot's field of view and defined a desired distance; the robot would then attempt to hew to that line as closely as possible (while avoiding obstacles should ODOA be enabled). Stairclimbing allowed autonomous traverses of single flights of stairs.

A single experienced user controlled the robot over each run. The user had feedback from one of the stereo cameras allowing a keyhole view of the environment from the robot's perspective. Location of the robot's current position relative to the starting position was recorded but not viewed during the run.

Two courses were used: one fairly easy run taking place entirely on road surfaces and one comparatively hard run which involved driving on grass, sidewalks, and stairs as well as roads. Data was collected for four runs over each of the two courses. For each course two of the four runs used only teleoperation and the remaining two used autonomous behaviors. The hard course consisted of positive and negative obstacles, two flights of stairs and narrow pathways between positive and negative obstacles. The easy course was twice the distance but involved only paved roads with few obstacles. The user was familiar with each course and had walked them beforehand. Each course was designed as a navigational course rather than an exploration effort, in the sense that the user knew where he wanted to go and roughly what obstacles he would face.

During each run all mouse clicks and drags were recorded and time stamped along with position. This allowed us to relate user interaction with the robot's location.

4. Data and results

For each course we will show an overhead photo as well as graphs of a teleoperation run and an autonomous run. The graphs relate user interaction to the location at which that interaction occurred. The path of the robot is drawn in blue, and any user interactions are denoted by an 'X' superimposed on the path where the interaction took place. The robot's location was recorded for every 20 cm traveled, and the size of the 'X' reflects the number of seconds the operator spent interacting with the control unit during that 20 cm segment. Specifically, marker size is proportional to the cube root of the interaction seconds per segment. Thus a point drawn 3 times the size of another point represents 9 times as much interaction.

The space between X's indicates no user interaction. This indicates distance traveled using waypoints or stairclimbing, where the robot was entirely autonomous.

After discussing each course we will show combined statistics gathered over all 8 runs.

4.1 Easy Course

Fig. 1 is an overhead image of the Easy Course with the robot path drawn in red. Note that the entire path is along paved roads.

Fig 1. Overhead image of the Easy Course

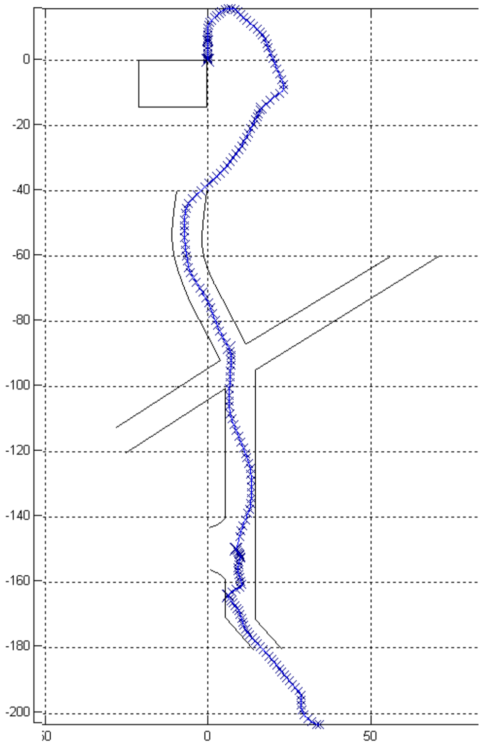

Fig 2.Teleoperation over easy course

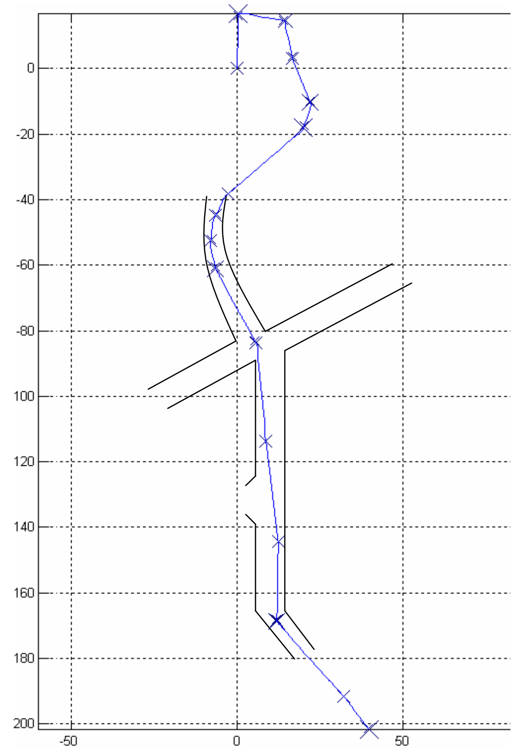

Fig 3. Waypoint over easy course

Fig. 2 and Fig. 3 show the graphs of user interaction as a function of position. Streets are overlain as hand-drawn lines.

During the teleoperation run the user was interacting for the entire run, constantly commanding a direction. During the waypoint run the user could interact for brief intervals to set up a new waypoint and then wait.

Each waypoint was set to 30 meters and could only be selected with the current image from the robot. This forced the user to intervene in the curved portion of the first half of the path to change directions, whereas in the second half the user could wait for the 30 meters to be reached before needing to interact again.

4.2 Hard Course

Fig. 4 displays the overhead image of the Hard Course with the chief obstacles labeled. Unlike the Easy Course, the path is over grass, sidewalks, streets, and up two flights of stairs. Negative obstacles forced the user to pay close attention during teleoperation and waypoints. Cars on the street required special concern and attention. Several situations were too complex for the autonomy on board and the user had to manually switch to teleoperation and guide the robot over the difficult sections.

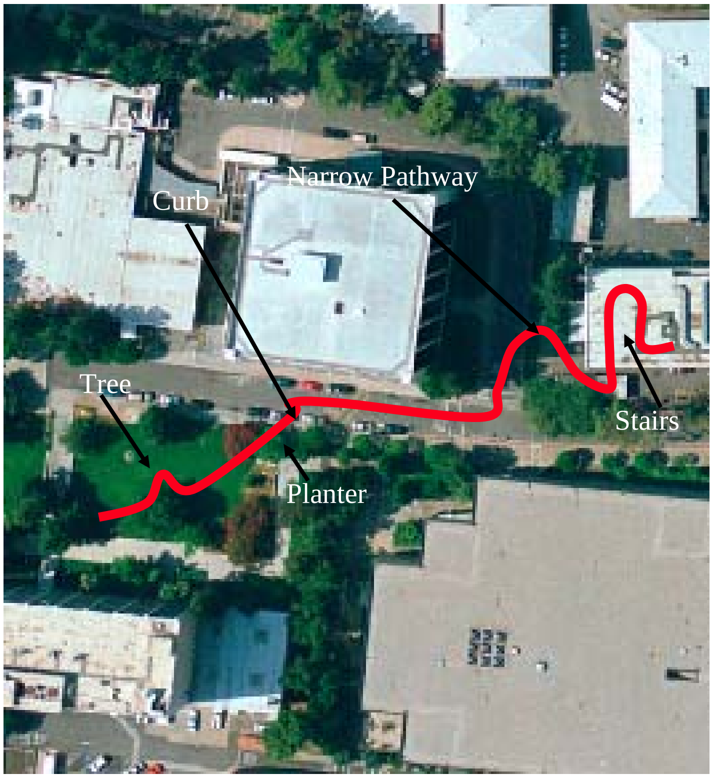

Fig 4. Overhead image of the Hard Course

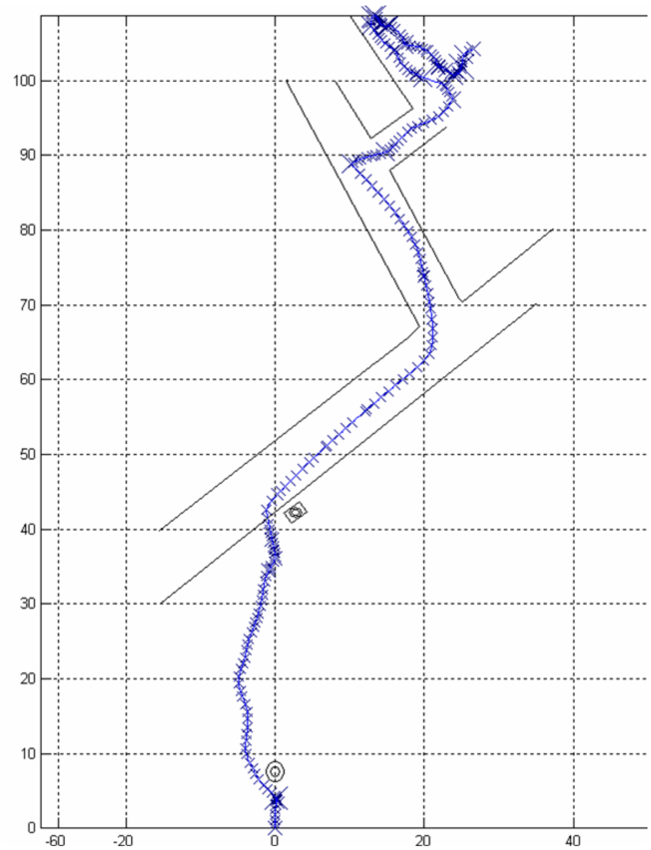

Fig 5. Teleoperation over Hard Course.

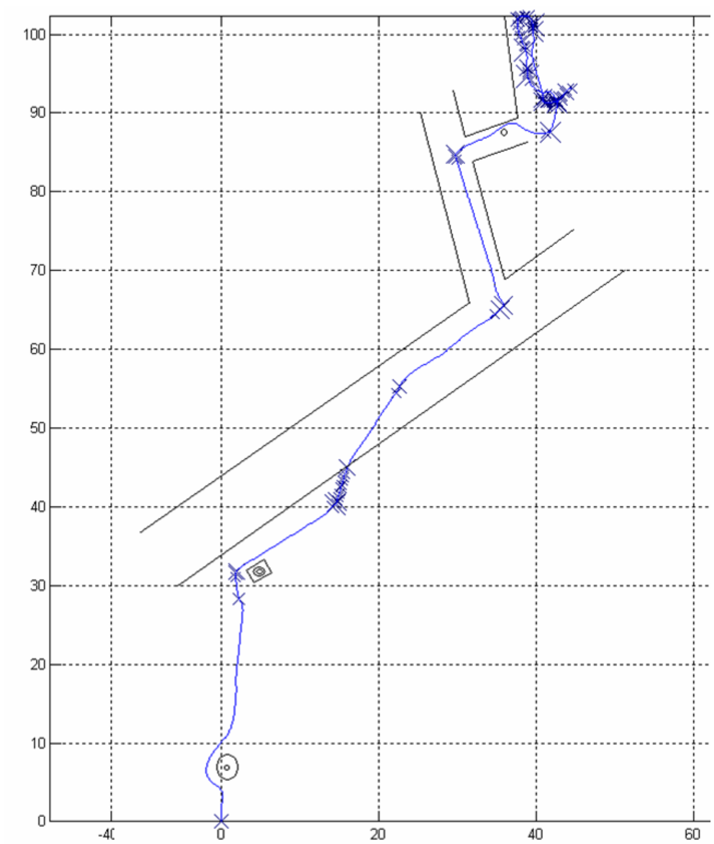

Fig 6. Autonomous and Teleoperation behaviors

Obstacles along the path (in the order they appear) are a
- Tree
- Negative obstacle (planter on the right shown as a small square).
- Negative obstacle requiring traversal (curb).
- Moving obstacles (cars)
- Narrow pathway with a pole in the middle
- Stairs

Fig 5. shows a sample run with only teleoperation. Notice that the points of interaction become denser in the difficult areas near the curb, stairs and narrow pathways. Hand drawn on the image are two obstacles (the tree and planter) as well as the street edges.

Fig 6. illustrates the mix of autonomous behaviors and teleoperation used for the Hard Course.

A few areas should be emphasized:
- The tree was avoided autonomously using ODOA.
- The planter was avoided by manually aborting a waypoint and selecting a new waypoint. The ODOA capability on Urbie would not have avoided this negative obstacle.
- The robot could not go down the curb because the path was blocked by a car (not shown in the figure); we followed the curb, bounded by negative obstacles on either side.
- Once past the car, the user was forced to teleop down the curb, as waypoint mode drives too fast to be safe.
- The user could not do a line of sight waypoint directly to the base of the stairwell; instead he used 4 waypoints, one after the other.
- The robot avoided the pole autonomously (shown as small circle in the middle of the narrow pathway).

Recall that there was no interaction on any path segment where no X's appear. Any smooth, unblemished line was traveled autonomously.

In the Fig. 6 the stairwell took so much interaction over such a small distance that it appears no autonomy was used. In fact, waypoints were used to the base of the stairwell and autonomous stairclimbing was used up both flights of stairs. The landings required manual teleoperation to maneuver in such tight quarters. Fig 7. illustrates this with a closer view (the robot enters from the bottom of the graph, finishes the run to the right).

	Easy Course Metrics				Hard Course Metrics			
	Teleop		Waypoint		Teleop		Waypoint	
Length (meters)	275	272	270	274	169	159	155	154
Time of Run (seconds)	264	380	355	370	423	459	448	488
Average Speed (m/sec)	**1.24**	**0.72**	**0.76**	**0.74**	**0.40**	**0.35**	**0.35**	**0.32**
Percent of time user interacting	**99%**	**92%**	**30%**	**26%**	**81%**	**79%**	**51%**	**45%**
User Interaction per meter (sec/m)	**0.79**	**1.29**	**0.40**	**0.35**	**2.04**	**2.27**	**1.46**	**1.44**
Percent of Distance Traveled Autonomously	**0**	**0**	**99%**	**99%**	**0**	**0**	**80%**	**78%**
Ave time of Autonomous Drive (sec)	**0.00**	**0.00**	**14.65**	**18.93**	**0.00**	**0.00**	**9.55**	**13.11**

Table 1. Interaction data over all 8 runs

A quick aside is necessary to explain Fig 7 properly. Only the first half of the first flight of stairs was traveled autonomously because a power fluctuation caused by the extreme driving conditions caused the communication link to be dropped. The user did not resume the stairclimbing behavior but simply teleoperated to the next flight.

A second oddity is that the second flight of stairs appears longer than the first, which is a result of extreme track slippage. Recall that the robot's orientation estimate was combined with the track encoder data (which is unreliable on stairs) to produce an estimate of robot x/y position via dead reckoning.

At the top of the second flight of stairs the robot traveled through a doorway inside the building using teleoperation.

4.3 Quantitative Measures

Table 1 the sum of all quantitative measurements over the 8 runs (with 4 runs per course). Attention should be directed to five areas in particular.

1. Percent of time user interacting is not 100% during teleop runs because the user had to pause occasionally to observe the surroundings, during which time he is not actually issuing any commands to the robot.
2. During the first teleop run on the Easy Course the user drove very aggressively and a little unsafely. The speed during this run is higher than the other 3 over the Easy Course. The user also never stopped to get his bearings during that run.
3. The user was forced to use teleoperation over the Hard Course during a few difficult segments. The Percent of Distance Traveled Autonomously was below 100% during those runs as a result.
4. The average time of autonomous segments on the Easy Course was longer than on the Hard Course because the path was generally straighter on the Easy Course.
5. The time to complete a course did not improve as a result of autonomy, even though the speed during autonomy (not shown) was higher than during teleoperation.

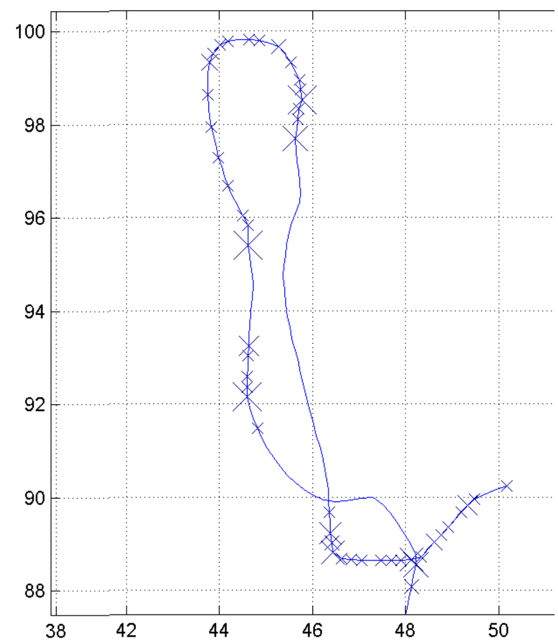

Fig 7. Close-up of the stairwell.

5. Autonomous Traverses Beyond Line of Sight

One interesting result is that autonomy did not reduce the overall time of the run even though the velocity during the autonomous waypoints was usually higher than a human driver would have been comfortable with in teleoperation. The setup time and piece by piece selection of waypoints took up the remaining time; each waypoint could only be selected from the robot's current field of view, which only allows short traverses before being occluded by obstacles.

The overall interaction time could have been much reduced if the user was able to select a series of waypoints at a time instead of waiting for the robot to finish each leg before commanding the next. Alternatively, the user could have selected a single goal, far in the distance, and trusted the robot get there autonomously. We call the first ability Scripted Waypoints and the second ability Single Point Goal Selection.

5.1 Scripted Waypoints

The challenge with scripted waypoints is knowing the absolute position of the robot in a global reference frame. The problem is really twofold: a) how does the user know where he wants to go before the robot can see it and b) translating this point to a coordinate frame the robot can use to navigate to.

The first part of the problem can be solved by providing the user with an overhead image or map of the terrain. In theory the user would be able to select points on the overhead image on the fly and have the robot drive from one to the next.

The second part of the problem is translating the points that the user clicks on in an overhead image to a reference frame the robot can use to autonomously navigate to. The predominant global localization method is using GPS with a known relation between GPS coordinates and the overhead image.

In general, obtaining GPS data in a dense urban setting can be difficult, at times impossible. The buildings to either side of the robot prevent the acquisition of enough satellites to provide meaningful data. In other types of terrain with significant sky coverage, GPS is a viable option and not to be discounted, but in many urban environments GPS availability can not be assumed.

An attractive alternative is to use the buildings themselves as landmarks to help localize the robot. Recent work at JPL has shown that it is possible to use vision algorithms to extract features from onboard imagery, to match those features to known buildings derived from the overhead image, and thereby to estimate the position of the robot in the global reference frame. This option is under ongoing development in the MARS program and would be a great boon to autonomous capabilities for terrain types where GPS is unavailable.

This option has the additional benefit of being viable within buildings, where GPS is nearly guaranteed to fail. The feature template could be a rough blueprint of the site or a sequence of pictures taken by hand during some reconnaissance run a priori. In either case the robot could continue to calculate its position in an absolute reference frame, allowing better state estimation and bounding absolute error.

5.2 Single Point Goal Selection

An autonomous behavior above and beyond scripted waypoints would be a single waypoint, very distant, with enough path planning ability on board to get there without further advice. The two greatest challenges to do this are a sophisticated path planner and a very accurate state estimator.

The path planner would have to be capable of escaping cul-de-sacs, navigating narrow path ways and capable of broad goal definitions.

Even with a perfect path planner, accurate position in the global reference frame is still needed. GPS could provide this global frame, but as we discussed it may not be available. Without a global position sensor, any state estimator, such as the Kalman Filter we used on our robot, will drift over time. Although usually very accurate, our filter did not tolerate climbing stairs or driving off of curbs very well. The sudden acceleration and deceleration, as well as extreme track slippage and skidding, brought about a slight error in pose every time. Compounded over another 50 meters of travel this slight error would grow to several meters of error in the pose estimate. As discussed in 5.1, matching vision sensor data to a prior map obtained from overhead imagery is a promising approach to this problem.

6. User Attention

The user looked at the robot feedback during the entire course of each run. He never looked directly at the robot, but neither could he look away from the screen to do anything else. In any real application, this is a critical flaw. The main reason that constant attention was needed was that the robot's current ODOA capability was not adequate to keep it out of trouble that could damage the robot or terminate the mission. The main types of trouble that could be encountered were negative obstacles and moving objects.

6.1 Negative Obstacles

Urban settings are rife with negative obstacles: planters, stairs, curbs, and walkway edges. Rails that prevent humans from walking off pathways are often placed too high to be spotted by ODOA and allow the robot to travel without resistance into a crevasse. Some negative obstacles such as curbs are traversable, but only at low speeds. Although negative obstacle detection has been addressed for years for cross-country navigation, even there the problem is not completely solved, and we have not yet integrated such capabilities into Urbie. Moreover, the urban domain has unique characteristics that could profit from algorithms designed specifically for it.

The inability of our autonomous system to guard against negative obstacles forced a great amount of vigilance on the part of the user. Referencing back to the Hard Course, the site of greatest interaction besides stair climbing was driving off the curb slow enough to be safe. The autonomous behaviors could not identify the ledge as dangerous, so the user was forced to manually switch to teleoperation and drive off slowly. Only when safely down could the user switch back into an autonomous mode.

6.2 Moving Obstacles

Since much of the navigation was on roads, the user needed to keep an eye on the traffic during autonomous drives. The robot could drive directly into the path of a car, not being able to calculate that the car was in fact in motion. There has been work on moving object detection with LADAR and with stereo vision [9,10]; incorporating this into obstacle avoidance algorithms is a critical next step. Street driving is the easiest of all autonomous traverses: relatively straight, on a solid surface, with few holes or cliff edges, and the few obstacles that exist are large (cars, trucks). This is where autonomy can shine, but without some form of moving object ODOA, the best autonomy will be able to aspire to is fewer user interactions, with no hope of reducing user attention.

7. Conclusions

We did a series of experiments to quantify the effect of our system's autonomy upon the amount of user interaction required. We did this to gain insight into where the system would need to improve to reduce the strain on the user.

We found that great amounts of improvement could be achieved with the ability to command traverses effectively beyond the current field of view. On simple terrain this could lead to almost zero interaction necessary for hundreds of meters at time-- although for complicated terrain this would require a very clever path planner and state estimation accuracy beyond what is currently available. Reducing attention on top of interaction would require a robust negative and moving obstacle detection behavior.

8. References

[1] D. R. Olsen and Michael A. Goodrich. "Metrics for evaluating human-robot interactions". *Performance Metrics for Intelligent Systems Workshop(PerMIS)*, 2003.

[2] M. Tambe P. Scerri, and D. V. Pynadath. "Towards a theory of adjustable autonomy". *Journal of Artificial Intelligence Research*, volume 17,pages 171–228, 2002.

[3] S.Tejada, E. Normand, and S. Sharma. "Virtual Synergy An Interface for Human-Robot-Agent Interaction". *Workshop on Humans and Multi-Agent Systems*, 2003,

[4] T. Frost, C. Norman, S. Pratt, and B. Yamauchi. "Derived Performance Metrics and Measurements Compared to Field Experience for the PackBot". *Performance Metrics for Intelligent Systems Workshop(PerMIS)*, 2002

[5] D. Bruemmer, J. Marble, D. Dudenhoeffer, M. Anderson, and M. McKay. "Intelligent Robots for Use in Hazardous DOE Environments". *Performance Metrics for Intelligent Systems Workshop(PerMIS)*, 2002

[6] M. A. Goodrich and E. R. Boer. "Model-based human-centered task automation: A case study in ACC system design". *IEEE Transactions on Intelligent Transportation*, part A: Systems and Humans, Vol. 33, No. 3, May 2003

[7] M. A. Goodrich and D. R. Olsen, Jr. "Seven Principles of Efficient Interaction". *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, October 5-8, 2003, pp. 3943-3948.

[8] J. W. Crandall. *Towards Developing Effective Human-Robot Systems*. M.S. Thesis, Brigham Young University, 2003.

[9] E. Prassler, J. Scholz, M. Schuster, D. Schwammkrug. "Tracking a large number of moving object in a crowded environment", *IEEE Workshop on Perception for Mobile Agents*, Santa Barbara, CA, June 1998.

[10] A. Talukder and L. Matthies. "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow", *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, Sendai, Japan, October 2004.