Human-Robot Team Navigation in Visually Complex Environments

John Carff[1], Matthew Johnson[1], Eman M. El-Sheikh[2], and Jerry E. Pratt[1]
[1]Florida Institute for Human and Machine Cognition
[2]Department of Computer Science, University of West Florida

***Abstract*—Current fully autonomous robots are unable to navigate effectively in visually complex environments due to limitations in sensing and cognition. Full teleoperation using current interfaces is difficult and the operator often makes navigation mistakes due to lack of operating environment information and a limited field of view. We present a novel method for combining the sensing and cognition of a robot with that of a human. Our collaborative approach is different from most in that we address bi-directional considerations. It provides the human a mechanism to supplement the robot's capabilities in a new and unique way and provides novel forms of feedback from the robot to enhance the human's understanding of the current state of the system and its intentions.**

***Index Terms*—Mobile Robots, Navigation, Cognition, Complex Environments**

I. Introduction

Robots are currently unable to autonomously operate in visually complex environments due to limitations in sensing and cognition. Once outside the structured laboratory environment, the current sensors provide inadequate information for successful autonomous operation. Even the best 3D range finders can miss critical pieces of information and place a large computational burden on the system. However, if a perfect sensor did exist, the lack of cognitive ability to interpret the sensory information would still be a major barrier to navigation. Object recognition is still in its infancy and scene interpretation is still a far off dream. Therefore, to date, the vast majority of robots deployed in urban environments have been fully teleoperated (e.g. Talon and Packbot), placing a large burden on the human operators who are often hindered by poor interfaces [1].

Consistent with much of the recent work being done in this area [2-6], we believe that with a good system design and an effective user interface, a human-robot team navigation system can be faster, more accurate, and more efficient than a purely teleoperated system or a purely autonomous system. In this paper, we describe the design goals for a human-robot team navigation system, methods for achieving those goals, and the implementation of such a system. Section II provides a brief review of other work related to the scope of this project. In Section III, we discuss the design goals and methods for a human-robot team navigation system. In Section IV, we present our implementation of such a system. The paper concludes with a discussion of the uses, benefits, and limitations of our approach, and directions for future work.

A. Navigation Challenges for Purely Autonomous Systems

Despite the advances made in autonomous navigation, navigating in visually complex environments remains a challenge. In such environments, there are many instances of Turing Test style navigation problems, which seem to require human-level cognition to solve.

Figure 1 shows just a few examples of scenes that are simple for humans to interpret, but would thwart today's robotics systems. On the left is an automatic sliding glass door adjacent to glass windows. To a human familiar with sliding doors, the proper way to enter the building is obvious. This knowledge would be difficult to include in an autonomous system and the shadows and reflections make this a daunting image recognition problem. The image on the right is a fence with a narrow gate that is covered in vines. Discriminating between fixed and moveable obstructions such as the vines is another challenge out of reach of today's navigation systems.

Like typical Artificial Intelligence systems, we could add specific rules for these and other cases but would soon find that there are too many special cases to account for, and that no matter how many special rules we add to our system, we would soon encounter novel situations.

Fig. 1. Typical scenes that are trivial for a human to interpret, but challenging for an autonomous system. On the left, an automatic glass sliding door. On the right, a narrow gate with vines.

In addition, in a hostile environment, it would be very easy for adversaries to add "Navigation Captchas" to the environment to prevent movement of machines while not hindering access to humans. These could include signs that are difficult for machines to interpret, as well as obstructions that can easily be moved, but that a machine might not deduce as being movable (such as beads in a doorway).

No matter what advances are made in autonomous navigation, we believe that a Human-Robot Team Navigation system that combines human and robot capabilities will be more effective than either a purely autonomous or purely teleoperated system.

II. Related Work

Over the past several years, extensive work has been done in the field of robot navigation. Advances in this technology have grown rapidly, producing many diverse systems. However, most systems deployed in complex urban environments are still teleoperated, such as the Talon [7] and the Packbot [8]. While teleoperation provides greater control over a robot, it also places a heavy burden on the operator, who often loses situational awareness and has difficulties navigating in cluttered urban environments [1]. Alternate to teleoperated systems are autonomous control systems such as those used on robots entered into the DARPA Grand Challenge [9]. Our research focuses on an improved navigation system that incorporates some of the best aspects of both teleoperated and autonomous systems.

There have been several approaches that provide improved information about the robot's environment to the human operator. Sugimoto and colleagues developed the Time Follower's Vision [10]. In this system, a delayed video stream and virtual robot avatar are used to give the operator the ability to operate the robot from a third person's view. Using this view, it is easy to acquire situational awareness of the robot and be very precise in movements. Our system incorporates a similar feature, but also allows for the user to place multiple views around a room from any point that the robot has previously been and view the robot from many different angles as shown in Figure 7. We also allow for the robot to navigate behind objects and be occluded by them due to the 3D mapping of the environment.

There have also been approaches that provide a mechanism for the human to provide information to the robot. For example, in Chronis and Skubic's Hand-Drawn Maps for Robot Navigation System [11], the user creates 2D maps for the robot to use. Our system enhances that approach by allowing the operator to create 3D maps consisting of virtual objects for greater environmental control. Instead of relying on the robot's interpretation of the environment, we rely on the operators spatial reasoning skills to extract 3D information from two dimensional images. Using this system, the operator can more precisely control what the robot knows about the environment and even inform the robot about information that no other autonomous system could detect. Fong's collaborative control poses questions to the user when it has trouble with perception or cognition [12]. However, it does not provide a method to interject unsolicited information into the robot's sense-plan-act sequence.

Recently, systems have been appearing that allow a user to interact with the system's autonomous processes. Google Maps has incorporated one such feature in their online mapping system. This feature not only allows for routes to be presented to the user, but also allows for the user to manipulate those routes by dragging control points on the path that was generated. Our system has a similar feature. Instead of fully relying on the robot to generate a path, we allow the user to adjust this path to avoid sensitive areas or take a different route entirely. Also much like Google Earth, we provide the capabilities to allow the operator to open up many different viewports. These viewports can contain real time video stream overlaid with virtual information, virtual environments that the operator can fly around in, top down views for enhanced navigation, and rewind views for external robot perspectives.

A main focus of our approach is to be highly collaborative, providing bi-directional information exchange. The human receives enhanced information through the interface that includes a grid overlay and a mixed reality robot overlay providing a three person view. Additionally the human can get a preview of intentions in this same display. The robot receives information from the human that supplements or enhances its capabilities. An added advantage of this approach is that the human gets feedback through observation of the real environment (actual video) on how well the system is working. If deviations between the virtual elements and the real elements occur, they are obvious to the human and indicate a need to adjust the robot's model of the world. This feature is very helpful in predicting the robot's competence.

III. Design of a Human-Robot Team Navigation System

In this Section, we describe the design goals for a Human-Robot Team Navigation System, and the methods for achieving those goals.

A. Design Goals for a Human-Robot Team Navigation System

The main goal of a Human-Robot Team Navigation System is to be more effective and efficient than a system that uses human or machine cognition only, by utilizing and integrating the capabilities of both.

Some of the sub-goals that can help achieve this goal are:

- Quickly and continuously provide situational awareness to the human operator.
- Provide awareness of autonomous processes to the human operator to minimize automation surprise.
- Utilize the best aspects of currently available fully autonomous systems. Allow these autonomous

subsystems to operate despite their limitations, knowing that their results can be filtered by a human.
- Allow a human to input goals more quickly than through teleoperation, knowing that the robot will be safe due to the capabilities of the underlying autonomous system.
- Provide tools and interfaces that place a minimal burden on the human operator.
- Enhance trust of the system by providing appropriate alerts and requiring human confirmation at appropriate decision points.

B. Methods for Achieving an Effective Human-Robot Team Navigation System

In this Section we, describe some methods that are useful for an effective Human-Robot Team Navigation System. In Section IV, we describe the implementation of a system that incorporates these methods.

1) Mixed Reality Displays and Virtual Viewports

By using mixed reality displays, we can superimpose real-time imagery with virtual processed imagery to give the operator a sense of the data, goals, and intentions of the system, improving situational awareness. We can provide various virtual viewports to the operator, such as an overhead view and views to the various sides of the robot. Of course, without a real sensor placed overhead, that view will only be able to display virtual objects. However, any view from a real sensor will be able to overlay both virtual and real imagery from that sensor. In fact, the sensor could be on a different robot [13], or could even be recorded from a different vantage point at a different time, as described in the "Rewind" Section below.

Of course, it will be very important for the human to be able to easily differentiate between the real information and the virtual information in the mixed reality displays. As we develop future systems it will be interesting to see if there are any psychological problems that arise that make such displays confusing and/or disconcerting.

2) Human-Assisted Image Processing and Scene Interpretation

Ideally, the human would use as natural a user interface as possible, perhaps even describing a scene using natural language. The user could label objects, mark their boundaries, describe their properties, etc. In the simplest case, the operator could simply determine what areas are keep-out regions and what areas are passable. For example, consider the image in Figure 2. A human can easily see the books on the shelves, the immovable pillar, and the moveable table and chairs. A user could label this room as a library, label some of the objects, and indicate to the system that there is likely a free path around the left side of the pillar as it is highly probable that the pillar has a square cross section and does not extrude all the way to the wall. An automated system might not be able to make that deduction without first exploring to gather more information to the left side of the pillar.

A very simple interface for allowing human processing and interpretation is to allow the human to place keep-out regions in the scene, using a mixed-reality display. For example, in Figure 2, a human could place a keep out region along the bookshelf, a square upward extrusion at the pillar, and a box over the table and chairs. This is the approach we take in the system we implemented, described in Section III.

3) Play-Forward

A "Play-Forward" feature would allow one to observe the intention of a robot by displaying the expected action of an ongoing or upcoming plan. The user could then terminate or modify the plan if desired, or continue work on something else with the confidence that the robot will perform as expected without having to continuously monitor the robot. This feature can greatly enhance the level of trust of the system by reducing automation surprises.

Fig. 2. Example image showing a library with table and chairs and a pillar. A human can quickly identify the objects in the scene, distinguish between the movable and immovable objects, and determine that there is likely a path to the left of the pillar.

As the example in Figure 3 illustrates, one could use graphical overlays on a camera image and an overhead view to show the expected path of a humanoid robot around an object and down a corridor. Figure 3a shows the path (green), the specific footsteps (dark blue) and a virtual representation of the robot. The robot's path could be animated so that the user could watch the robot "walk" through the environment. This approach would be similar for a wheeled vehicle except without the footsteps. With spatial information from objects or keep-out regions in the environment, the robot could be occluded as appropriate to convey passing behind objects. Similar information can be presented from alternate views, such as an overhead view as in Figure 3b.

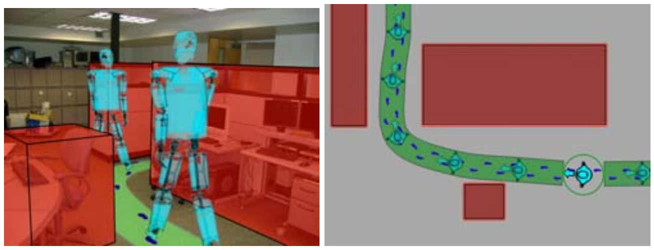

Fig. 3. Example of Play-Forward: a) mixed reality image b) virtual 2D overhead view.

4) Manual Adjustment of Automatically Generated Sensor Processing or Plans

Due to limitations in autonomous navigation, a fully autonomous system can often suffer from sensor processing or planning mistakes. Rather than completely bypassing the autonomous system, and relying on pure teleoperation of the robot, a more efficient method might be to let the system do the best it can and then have a human manually adjust the results of the autonomous system.

These adjustments could be applied directly to the output of the system, for example, by moving waypoints on a plan. Alternatively, they could be applied to the inputs. For example, a keep-out region could be added to an area to force the autonomous system to re-plan a route around the keep-out region. They could be applied after the autonomous system has finished doing the best that it can, or they could be part of a continuous interaction between the system and the human.

Play-Forward of a plan would allow the operator to see potential errors or misinterpretations prior to execution. The operator could then interact with the plan in various ways. One interaction would be to increase the standoff range for obstacles. For example, if the path passes too close to obstacles, the user may simply increase the desired standoff range and the path will be automatically updated, as shown in Figure 4.

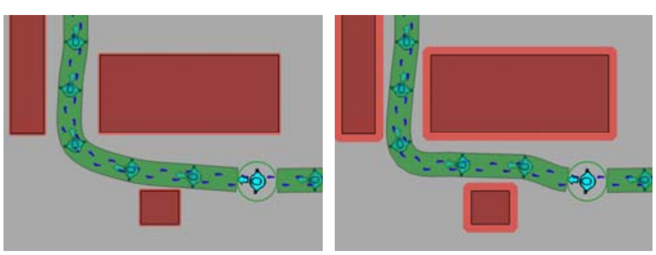

Fig. 4. Example of standoff adjustment to increase the distance that the robot stays away from the obstacles.

A second method is to "grab and drag" portions of the plan to modify it. This would be appropriate when the standoff range is fine, but there is a particularly sensitive object you would like to give more room to or a direction you would prefer to favor. The path generated by the system has handles that allow the user to grab and manipulate the path. Once grabbed, the user is presented with the allowable range of dragging. As the user drags, the plan is recalculated and redisplayed. In addition, the user can add additional keep-out or stay-in regions to the 3D model to cause the plan to change.

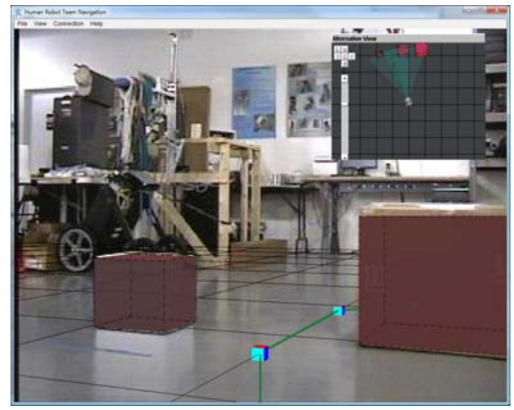

Fig. 5. The view from the robot after various objects have been marked off by the operator and a path has been planned through them by the robot. The colored cubes represent the waypoints the robot has generated.

Figure 5 shows a view from our robot after the operator has marked off the obstacles critical to the robot's navigation. The robot has planned a path through those obstacles and has presented the operator with the path. In this figure, there are colored cubes connected by a green line on the ground. These colored cubes are waypoints that the robot has generated for navigating through the boxes. To change the robot's generated path, the operator may drag these waypoints around to change the robots path or add additional keep out regions and have the robot re-plan.

5) Rewind and Viewports from Recorded Imagery

A "Rewind" feature can be used to help reacquire situational awareness and avoid errors. Such a feature could allow the user to see a short replay of the movement prior to the current location using graphical overlays and actual imagery that was recorded as the robot moved to its current location. The user could also "zoom out" a short distance, providing a different viewpoint. This would help to rapidly situate the user and help reduce the problems associated with context switching and limited field of view that are typical in teleoperation interfaces.

This rewind feature can be achieved by saving a brief history of images and objects and display a virtual representation of the robot from the viewpoint of its previously recorded data. This will allow the user to "see" their immediate surroundings better. For example, Figure 6a shows a view from our robot's camera at its current position. This view shows a doorway that can be difficult to enter using a robot with a limited field of view. To make the process of navigating through a doorway easier, an operator can set up a rewind view at the robot's current location.

Figure 6b shows the rewind view from the original location of the robot after key obstacles have been marked off. Using this rewind view, and a virtual representation of the robot, the operator has a view from behind the robot and was easily able to navigate the robot through the doorway without colliding with any obstacles.

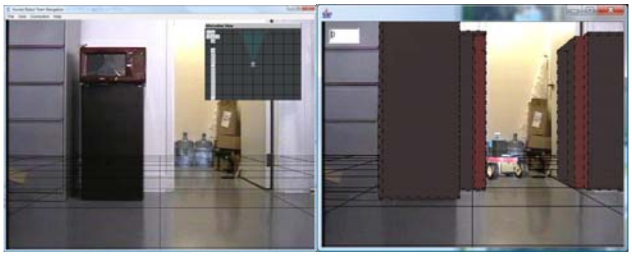

Fig. 6. Example of Rewind feature: a) Robot's view showing a doorway that has not been marked off by the operator. b) Rewind view with the doorway marked off and the robot, shown virtually, beyond the doorway.

In current systems based on typical teleoperation interfaces, the operator is burdened with maintaining a detailed cognitive map of the environment, while providing control for the robot. A Rewind capability would allow the user to replay a short history and provide viewpoints similar to those shown in Figure 7. Here you get a glimpse of the robot situated in its environment from multiple viewports and can see an obstacle that is outside the robot's view in its current position.

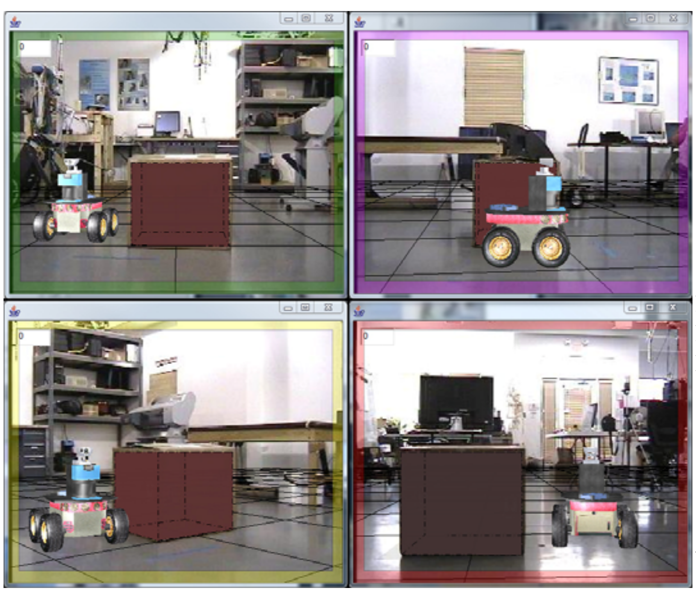

Fig. 7. Multiple mixed reality viewports showing external perspective and the virtual robot.

This idea can be generalized to include mixed reality viewports from any previously recorded source at different locations and times. For example, upon entering a room, the robot could move side to side, looking at the room from different perspectives. Then, as the robot navigated through the room, various viewports could be provided to show the view of the robot from the entrance of the room.

Of course, in order to effectively mix old imagery with new data, correspondence between data sources must be maintained, and the environment must not change significantly. Depending on how dynamic the environment is, viewports from older recordings could indicate the staleness of their data and/or disappear as they become too old.

IV. Implementation of a Human-Robot Team Navigation System using a Wheeled Robot, Dead Reckoning, and Single Camera

In this Section we describe our first implementation of a Human-Robot Team Navigation system that achieves many of our design goals. In this system, we use a very basic robot system, shown in Figure 8. This system consists of a Pioneer 3AT wheeled robot from Mobile Robots, equipped with a compass, a Sony PTZ camera with pan and tilt, and a SICK Laser. The robot has adequate dead reckoning so that we can maintain good correspondence between the virtual and real objects in our mixed reality displays. With this simple platform we are able to demonstrate many features of Human-Robot Team Navigation.

Fig. 8. Platform used for our first implementation of a Human-Robot Team Navigation System. The robot is a Pioneer 3AT from Mobile Robots, equipped with a pan-tilt camera and a laser range sensor.

A. Interface Development for Single Camera

We developed a novel interface that makes use of single camera images. A simple camera is still a tremendous untapped resource for robots. Image interpretation is very difficult for fully autonomous systems, which is why laser range finders have recently been a more favored sensor. However, a human can extract significant information from a single camera image, including inferring 3D information, guided by experience with urban environments. Rather than simply displaying the image to an operator for teleoperation, we use some creative techniques to allow the operator to provide information to the robot and vice versa.

Figure 9 shows two views from the user interface. On the left is a mixed-reality camera view, and on the right is a virtual overhead view. The main camera view displays what the robot sees, along with the information the robot knows about the environment. The virtual overhead view displays the location of known objects and the robot's location in the world. The main camera view is also what the human operator uses to mark off areas of interest to the robot. We see that the user has marked off a box in the middle of the room by placing a virtual keep-out region around the box. After an object is marked off through simple mouse commands, the object is added to the virtual world and seen in the virtual overhead view. Using this overhead view, an operator can easily navigate the robot around obstacles, even if the obstacle is not in the field of view of the camera.

Given a single camera image, the cognitive capabilities of a human can be used to determine things such as what surfaces are horizontal and vertical, what is most likely located in an occluded area, what objects are moveable, where doors are and how they are opened, what objects are safe to step on, what objects can be bumped into and which must be avoided, and what signs say and what they mean. We exploit these cognitive capabilities first by creating a simple 3D model of the environment that consists mainly of keep-out and stay-in regions represented by simple geometric shapes. We then automatically plan in this model and allow the user to interact with the system to improve the planning process.

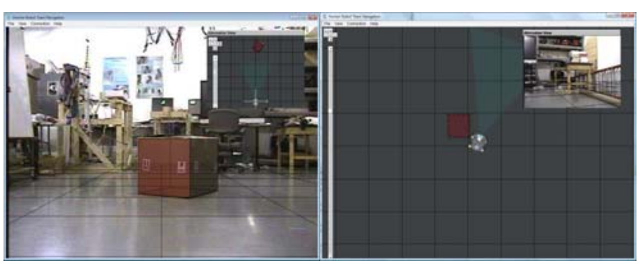

Fig. 9. a) A screen shot of the user interface showing a box in a room that has been marked off by the human operator. b) The interface showing the overhead view and the camera view swapped for easy navigation. The overhead shows the location of the box in relation to the robot after the robot has driven to it.

There are three main aspects to the interface design: 1) building a simple 3D model of the world, 2) projecting robot intentions using the 3D model, and 3) using historical information to improve the operator's situational awareness.

Using knowledge about the camera's location and orientation and identifying orthogonal surfaces, the user can assist the robot in mapping out its 3D environment using only a camera. The user can specify if a point in the view is on the ground. The user can also specify if a point is vertically above a given point. In these cases, the system can determine the 3D coordinates of the location signified by the user while only using a 2D image.

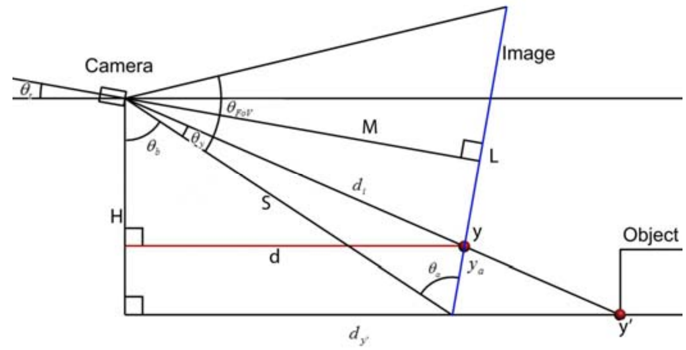

Fig. 10. Side view diagram for calculating distance from camera to ground objects on the y-axis.

In Figure 10, *y'* is the actual location of an object in the environment and *y* is its corresponding location in the static image. The pixel height and width of the image is known. The height of the camera, *H*, and the angle that the camera is tilted, θ_t, is also known. Knowing the exact height and rotation of a camera in an environment, one can calculate the distance to any floor point in a static image. With this, if you know the location of *y* in the image you can find the location *y'* in the environment.

To find angle θ_a, the angle in the isosceles triangle that is created with the image and the camera, we use the camera's horizontal field of view that is represented by θ_{FoV},

$$\theta_a = \frac{\pi - \theta_{FoV}}{2} \tag{1}$$

Using the field of view, θ_{FoV}, and the angle that the camera is tilted, θ_t, we are able to calculate θ_b, the angle from the camera to the point that the image plane intersects with the ground. θ_t is negative if the camera is angled down.

$$\theta_b = \left(\frac{\pi}{2}\right) - \left(\frac{\theta_{FoV}}{2}\right) + \theta_t \tag{2}$$

Now that we have θ_b, we can use the height of the camera, *H*, to find the direct distance from the camera to the bottom of the image, *S*,

$$S = \frac{H}{\cos(\theta_b)} \tag{3}$$

Using *S* and the field of view, we can find the distance from the camera to the center of the image, *M*,

$$M = \cos\left(\frac{\theta_{FoV}}{2}\right) S \tag{4}$$

Using these two distances, *S* and *M*, we can find the height of the image, *L*,

$$L = 2\sqrt{\left(S^2 - M^2\right)} \tag{5}$$

We use the pixel location of the object, *y*, the height of the image in pixels, i_h, and the actual height of the image, *L*, to calculate the distance from the bottom of the image to the location of the object in the image y_a,

$$y_a = \frac{(yL)}{i_h} \tag{6}$$

We can then calculate d_i, the direct distance to the object in the image from the camera location.

$$d_i = \sqrt{{y_a}^2 + S^2 - 2y_a S\cos(\theta_a)} \tag{7}$$

Once we have d_i, we can calculate the angle from the bottom of the image to the object location in the image, θ_y.

$$\theta_y = \arcsin\left(\frac{[y_a \sin(\theta_a)]}{d_i}\right) \tag{8}$$

Using the height of the camera, *H*, we can calculate $d_{y'}$, the distance on the floor from the camera's location to the object location along the y-axis,

$$d_{y'} = \tan(\theta_b + \theta_y)H \tag{9}$$

d is the horizontal distance to the object in the image from the camera location that we will use to calculate the *x* distance of the object from the image.

$$d = \sin(\theta_b + \theta_y)d_i \tag{10}$$

In Figure 11, *x'* is the actual location of an object in the environment and *x* is its corresponding location in the static image. The angle the camera is rotated, θ_p, and the pixel location, *x*, are known. With this, if you know the location of *x* in the image, you can find the location *x'* in the environment.

We use the width of the image in pixels, i_w, the height of the image in pixels, i_h, along with *L*, the height of the image in meters, to calculate *w*, the width of the image in meters.

$$w = \frac{i_w}{i_h}L \tag{11}$$

With *w* we can calculate x_a, the *x* length in meters from the center of the image to the location of the object in the image.

$$x_a = \frac{xw}{i_w} \tag{12}$$

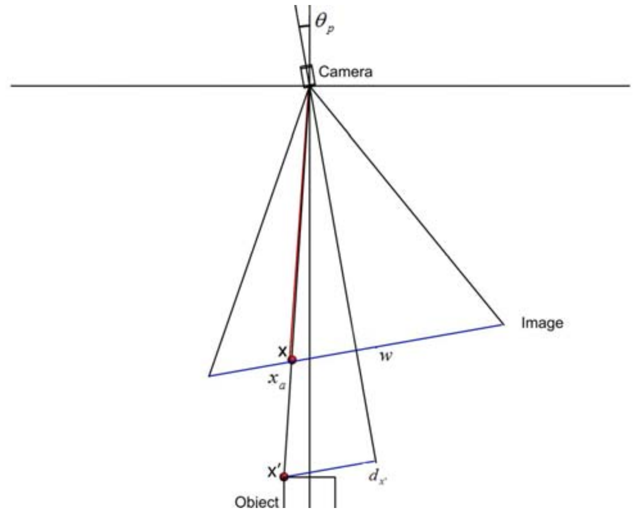

Fig. 11. Top view diagram for calculating distance from camera to ground objects on the x-axis.

We can then calculate $d_{x'}$, the distance in meters on the floor from the camera to the object along the x-axis.

$$d_{x'} = \left(x_a - \frac{w}{2}\right)\left(\frac{d_{y'}}{d}\right) \tag{13}$$

To find x' and y', the actual location of the object, we rotate the location $d_{x'}$ and $d_{y'}$ by the angle the camera is rotated around the z axis,

$$x' = d_{x'}\cos(\theta_p) + d_{y'}\sin(\theta_p) \tag{14}$$

$$y' = -d_{x'}\sin(\theta_p) + d_{y'}\cos(\theta_p) \tag{15}$$

As the objects are identified and mapped out by the user, they are color coded for easy visibility and to confirm for the user what the robot "sees." The user is also presented a 2D map based on the information generated by our algorithm. To keep the interface simple, the user only needs to identify relevant objects in the world. In our cluttered image in Figure 12, the user only needs to identify the partitions defining the exit and the chair that is likely to be an obstacle on the way to the exit. The color coding makes for clearly defined objects for the user.

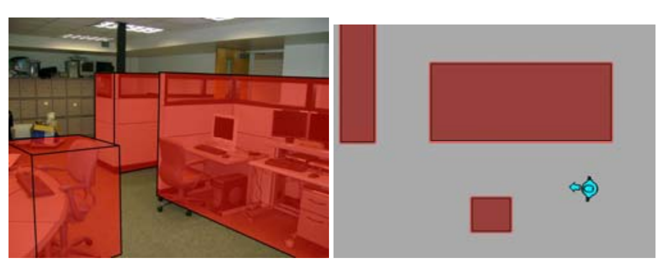

Fig. 12. Marked up image and map generated from markup.

In addition to implementing the system on the real robot, we also test the system in a purely simulated environment, in which the locations of all obstacles are perfectly known. This is useful for quick development and allows for verifying the correctness of our algorithms.

V. CONCLUSION

Despite advances made in autonomous navigation, navigating in visually complex environments remains a challenge. In such environments, there are many types of navigation problems that require human-level cognition to solve. Navigation mistakes are also common in fully teleoperated systems, due to the operator's lack of operating environment information and limited field of view. A Human-Robot Team Navigation system that combines human and robot capabilities will be more effective than either a purely autonomous or purely teleoperated system.

In this paper, we presented a novel method for combining the sensing and cognition of a robot with that of a human in a way that will both increase the speed and accuracy of navigating a mobile robot in a complex urban environment, while lessening the burden placed on a human operator by conventional teleoperated systems. We have described the design and development of a Human-Robot Team Navigation System. The system integrates several methods to achieve effective navigation, including mixed-reality displays, virtual viewports, human-assisted scene interpretation, and manual adjustment of automatically generated plans. The system also implements a Play-Forward feature, which allows the operator to observe the intention of a robot by displaying the expected action of an ongoing or upcoming plan, and a Rewind feature, which allows the operator to see a short replay of prior movement using graphical overlays and actual recorded imagery as the robot moved to its current location.

The combination of these methods and features enables the system to perform more accurately and effectively than a purely autonomous or a purely teleoperated system. The system also incorporates a novel graphical user interface that provides flexibility in determining the best course of actions for successful navigation. In addition, the design methodology facilitates use of the Human-Robot Team Navigation System in more complex, urban environments.

Future work includes

- incorporating the laser range finder and other sensors,
- including more advanced autonomy, such as a SLAM system and developing interfaces for the user to interact with it,
- developing different types of displays and interfaces for interacting with the system,
- expanding the system to robots requiring 3D navigation, such as humanoids that can step over obstacles,
- and performing evaluations to validate the improvements in speed and accuracy of navigating a mobile robot in a complex urban environment.

REFERENCES

[1] J. L. Burke, R. R. Murphy, M. D. Coovert, and D. L. Riddle, "Moonlight in Miami: a field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise.," *Human-Computer Interaction*, vol. 19, pp. 85-116, 2004.

[2] G. Ahuja, G. Kogut, E. B. Pacis, B. Sights, D. Fellars, and H. R. Everett, "Layered Augmented Virtuality," 2007.

[3] D. Bruemmer, D. Few, H. Hunting, M. Walton, and C. Nielsen, "Virtual camera perspectives within a 3-d interface for robotic search and rescue," 2006.

[4] C. W. Nielsen and M. A. Goodrich, "Comparing the Usefulness of video and map information in navigation tasks," 2006, pp. 95-101.

[5] C. W. Nielsen, "**Using Augmented Virtuality to Improve Human-Robot Interactions** ." PhD. Brigham Young University, 2006.

[6] B. Ricks, C. W. Nielsen, and M. A. Goodrich, "Ecological displays for robot interaction: a new perspective,", 3 ed 2004, pp. 2855-2860.

[7] P. Wells and D. Deguire, "TALON: a universal unmanned ground vehicle platform, enabling the mission to be the focus,", 5804 ed 2005.

[8] B. Yamauchi, "PackBot: A Versatile Platform for Military Robotics ,", 5422 ed 2004, pp. 228-237.

[9] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge* Springer Tracts in Advanced Robotics, 2007.

[10] M. Sugimoto, G. Kagotani, H. Nii, N. Shiroma, F. Matsuno, and M. Inami, "Time Follower's Vision: a teleoperation interface with past images," *Computer Graphics and Applications, IEEE*, vol. 25, no. 1, pp. 54-63, 2005.

[11] M. Skubic, S. Blisard, A. Carle, and P. Matsakis, "Hand-drawn maps for robot navigation," 2002.

[12] T. Fong, "Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation." PhD. Carnegie Mellon University, 2001.

[13] M. W. Kadous, R. K. M. Sheh, and C. Sammut, "Controlling Heterogeneous Semi-autonomous Rescue Robot Teams,", 4 ed 2006, pp. 3204-3209.