Deictic Primitives for General Purpose Navigation*

Jill D. Crisman, Ph.D.
Robotic and Vision Systems Laboratory
Departent of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

Abstract

We are investigating visually-based deictic primitives to be used as an elementary command set for general purpose navigation. Each deictic primitive specifies how the robot should move relative to a visually distinctive target. The system uses no prior information about target objects (e.g. shape and color), thereby insuring general navigational capabilities which are achieved by sequentially issuing these deictic primitives to a robot system.

Our architecture consists of five control loops, each independently controlling one of the five rotary joints of our robot. We show that these control loops can be merged into a stable navigational system if they have the proper delays. We have also developed a simulation which we are using to define a set of deictic primitives which can be used to achieve general purpose navigation. Encoded in the simulated environment are positions of visually distinctive objects which we believe will make good visual targets. We discuss the current results of our simulation.

Our deictic primitives offer an ideal solution for many types of partially supervised robotic applications. Scientists could remotely command a planetary rover to go to a particular rock formation that may be interesting. Similarly an expert at plant maintenance could obtain diagnostic information remotely by using deictic primitives on a mobile platform. Moreover, since no object models are used in the deictic primitives, we could imagine that the exact same control software could be used for all of these applications.

1. Introduction

We are developing a robot architecture which uses a natural deictic interface that allows the user to point out targets to the system. To operate a deictic mobile robot, the user would select a target in a video image and then issue a command such as "approach that" or "pass to the right of that" where 'that' is the target selected in the video image. In this paper, we describe the robot architecture that we are using for this deictic system. We also describe our simulation environment that we are developing to explore the definition of a set of deictic primitives to be used for general purpose navigation.

This work is important since the elementary deictic primitives give researchers a novel way to think about programming robot systems. Most robots are controlled by specifying a target in geometric terms, for example as a Cartesian position and orientation (e.g. 'go to 20m, 12m, and face 10 degrees') or as a location on a map. On the other hand, deictic primitives would involve a user pointing out a sequence of

---

* This work is supported by the National Science Foundation under grant number IRI-921056. This work is also aided by the donation of a Cognex 4400 Machine Vision System by Cognex Corp., Needham, MA.

visual targets and the robot moving relative to those targets. We believe that this type of programming interface is more natural for humans since people tend to move relative to what they perceive. For example, we would 'walk to the doorway' rather than 'walk forward 10 feet'. As our work progresses in the future, we will add object models so that our system would be able to 'approach the doorway'. Therefore, we believe that deictic commands would be a more natural method for people to interact with a mobile robot system.

This deictic interface is very different than interfaces to traditional mobile robots. Many robots are controlled by specifying a target location in geometric Cartesian coordinate with respect to an initial robot location. In this case, the robot must keep track of its location in order to know if it has reached the goal location. Other mobile robots navigate with respect to a map of the environment where goal locations are specified by a geometric coordinate on the map. The robot must continually track its position with respect to the map to determined if it has obtained its goal. Still other robots navigate to target objects which have pre-stored models so that the robot can identify landmarks. In all of these traditional approaches to interfacing with the robot, environmental knowledge must be encoded geometrically for the system to operate.

Our deictic system is very different in that the robot only needs to keep track of the destination object in it video field. Since target tracking is more robust than object identification, the processing time of our system is decreased. The robot does not need to keep track of its location with respect to a global map, therefore our system is not susceptable to position tracking errors. We take advantage of movable camera systems to simplify our robot control architecture.

This deictic interface for semiautonomous robots has many applications, especially in exploratory robots. Scientists can control a planetary rover by selecting a location of interest in the video screen and commanding the robot to go to that area. Underwater robots can be controlled with lower bandwidth communications than is typically necessary for remotely operated vehicles. Moreover, semi-autonomous robots have applications in aids for the handicapped.

In this paper, we overview the robot architecture which uses five feedback control loops to control the motion of the robot. We show that with the time constants on the feedback loops that this system can provide smooth and stable motion of all joints of the robot. We also present our initial work on a simulator for exploring the definition of a set of deictic primitive commands. We show the results of this simulation for a series of approach commands.

2. Related Work

Developing mobile robot systems based on traditional computer vision and robotics paradigms requires the use of an *a priori* object model for the goal and a reference coordinate frame [16] [20]. The vision system identifies the goal in the scene by using the *a priori* object model provided. The object positions and orientations are perceived in the camera coordinate frame and must be transformed into the reference coordinate frame and added to the world model. Other sensor modules add information to the world model. Motion decisions for the robot system are made by a path planning module using the most recent information from the sensors which has been integrated into the world model. As the robot moves, the system must record and update the robot's position within the world model. This system has been used in many robotic systems including [21] [11]. This traditional solution is somewhat limited since it assumes that prior object models are available, which is often not the case in applications such as planetary exploration and household robotics.

Similar systems, for example [13], construct a world model without having the *a priori* object models. However, the world model construction process is computationally very

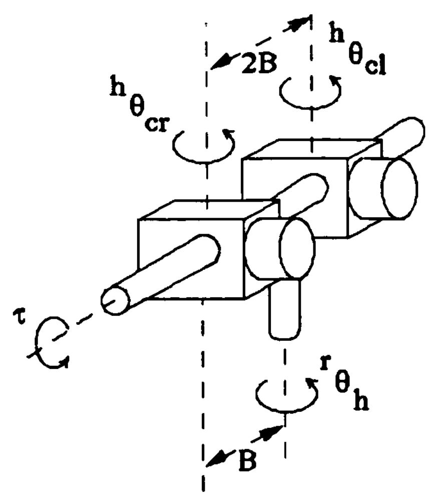

Figure 1: Robot Head. Our robot head has four joints. The first joint controls ${}^r\theta_h$, the pan of the head with respect to the robot base. The second joint controls the tilt, τ, of the cameras. The third and fourth joints control the pan of the cameras

expensive. These systems require calibration between the camera system and the robot, a localization routine so that the robot can identify its location with respect to the local map (so that the world model can be integrated over time), and a good kinematic and dynamic model of the robot system. The calibration, kinematic, and dynamic models always have associated with them some approximation errors. Motion planning, which is done on the world model, can become difficult as the robot modeling errors accumulate.

Visual servoing techniques have been proposed to eliminate the geometric dependence of the motion commands. Rather than directing the robot to a destination location, the robot is instructed to maintain its visually apparent position with respect to an object using dynamic visual feedback. Robot manipulators with a camera mounted on the arm can now track specific objects in 3-D space [22] [10] and navigation systems can track pathways [6] [9]. These systems work in real-time by tracking a specific visual feature rather than reconstructing a complete 3D description of the world.

Other researchers have abandoned traditional methods and instead have promoted behavior-based robotic architectures and local path planning algorithms [1] [3] [4] [12] [19]. These systems tend to use a distributed computer system to acheive tightly coupled control loops between the sensing and actuation. Therefore these systems have better reaction times in the presence of moving objects. Ultrasonic sensors are a common choice to provide fast obstacle detection [2] [14].

Our system currently uses a simple and fast method for determining the motion of the robot and most closely resembles these behavior based systems. Therefore our system is able to react quickly to a moving or newly detected obstacle. We use a visual servoing technique to position the gaze of each camera directly at the target. The mobile robot then moves in the gaze direction of the cameras if the pathway is clear of obstacles. Otherwise it moves around the obstacle and continues seeking the target.

3. Mobile Robot Hardware

Our experimental equipment consists of a mobile robot base with a ring of ultrasonic sensors, an active robot head, and a high speed video processor. The active robot head has four controllable motions. The robot head carries two cameras and controls the pan of each camera individually and it controls the tilt and pan of the pair of cameras, as shown in Figure 1. This platform is similar to those described in [5], [15], and [17]. The platform was constructed such that the pan and tilt of the cameras occur approximately about the focal point of the cameras. A Cognex 4400 Machine Vision system is currently handling the real-time video processing of the cameras. The active camera head is mounted on a mobile robot platform with a ring of 24 ultrasonic sensors. Each ultrasonic sensor can determine the distance to the closest object in a 30° field of view.

4. System Architecture

Our goal is to achieve fast, reliable pursuit of a target while avoiding obstacles in the path. Our system includes three components: a target tracker, obstacle detector, and mediator as shown in Figure 2. The target tracker follows the target location selected by the user and reports the angle and distance of the target to the mediator. The active robot head is used to simplify the target tracking task. The obstacle detector reports the measurements from the ultrasonic sensor ring. These measurements are the distance to the closest object within the field-of-view of each sensor as a function of angle from the robot. The mediator then determines the speed and steering angle of the robot. In the following subsections, we describe in more detail the three components of this system.

4.1. Tracker

The tracker is responsible for reporting the angle and distance to the target. Since we are focusing on a video interface, we will be using targets from video images from the stereo cameras. We are using stereo cameras to determine the distance to the target. While determining the distance to a stationary target is possible from a moving platform with a known motion, we do not assume that the target is stationary nor that the motion of the target is known. As the robot and target are moving, the tracker must determine the location of the target in the image. Since the target can easily move outside of the field of view of the cameras, we use an active robot head to keep the target in sight and thus to simplify the tracker.

The tracker operates as four independent controllers, one for each motion of the camera head: right camera pan, left camera pan, head pan and tilt (see Figure 1). The target is first located independently in each stereo image. The camera pans, θ_{cl} and θ_{cr}, and the head tilt τ are used to move the cameras such that the position of the target appears in the center of the stereo images. The head pan is independently controlled to try to face the cameras directly at the target. The angle to the target can then be directly measured from the pan of the robot head. The angles of the stereo cameras with respect to the robot head can be used to compute the distance to the target. For more details of this controller see [7] and on video tracking [8].

4.2. Obstacle Detection

The sonar system is responsible for reporting the locations of obstacles surrounding the vehicle. In a typical ultrasonic system, each

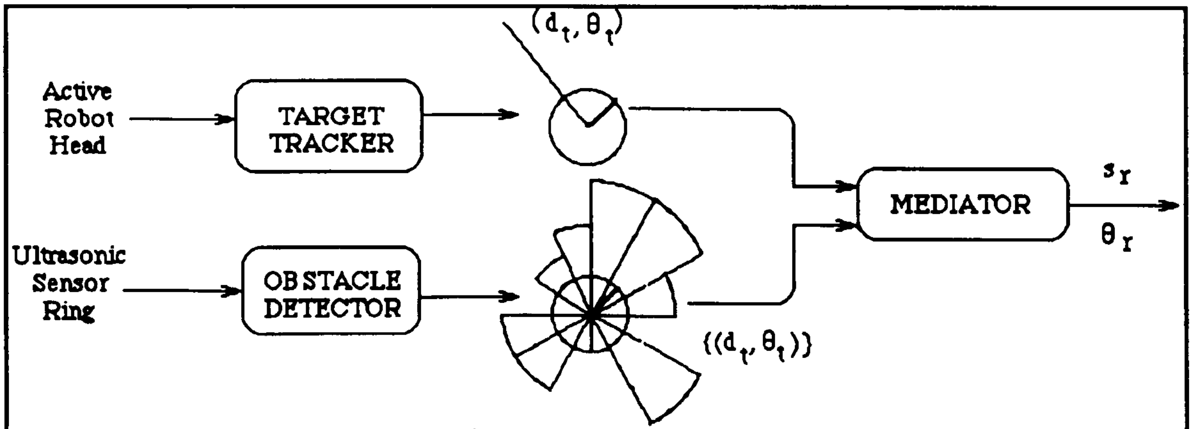

Figure 2: System Overview. Target tracking uses the active robot head to report the direction and distance of the target relative to the mobile robot base. Obstacle detection reports the distance to the closest object within the field-of-view of each sonar sensor. The mediator picks the best speed and steering angle commands for the mobile robot base.

sonar covers a 30° field-of-view. The object which is closest within this field is detected by the sonar. The sonars are spaced in a ring around our platform. The mediator receives the result of each sonar individually. These readings can be thought of as the cost of the robot traversing in that direction.

4.3. Mediator

The mediator decides the steering and speed commands that will be sent to the mobile robot. The tracker reports to the mediator the current direction and distance to the target. The obstacle detector determines a radial map of distances to obstacles surrounding the vehicle (see Figure 2). Interestingly, we found that the mediator need not be complex to steer the robot successfully.

Consider that the robot can only steer within the resolution that it can sense. Therefore, to track the target in an image, the robot can steer according to the resolution of the pixels in the image. However, if obstacles are detected, the robot only knows that an obstacle appears within a 30° field-of-view. Therefore, the robot can only steer in 30° increments. Each ultrasonic reading corresponds to a steering direction. If an ultrasonic sensor detects an obstacle, then the robot should not steer into the 30° field-of-view of the detecting sensor.

If there are no obstructions in the direction of the target, then the robot pursues the target direction. If there is currently an obstruction in the direction of the target, the mediator will select the closest open steering angle to the target.

The mediator also considers the closest obstacle and the distance to the target when selecting the vehicle speed. The speed is inversely proportional to the distance to the closest object. We pursue the target to within a fixed distance. For safety reasons, the robot's speed is also clipped to a maximum value.

4.4. Simulation

To show the competence and stability of the system we have simulated a robot motion model to test our navigation algorithms. To ensure a realistic simulation, we have modeled each motion of the robot as a second-order system. The motion of the robot joints is modeled as a damped response to the desired motion commands issued by the mediator.

At each step in our simulation, two camera images and 24 ultrasonic measurements are taken of the environment. We assume that these measurements are relatively accurate. We completely model the limited field of view of the cameras and the quantization of the camera measurements. We also add random noise to these measurements. The ultrasonic measurements also have noise added and we model a 30° field-of-view of the ultrasonic sensors.

The simulation keeps track of the motion of the target and the motion and orientation of the robot with respect to a world coordinate frame. Notice that in our architecture, the robot does not know about a world coordinate frame since it has no world model. The robot only concentrates on pursuing the target location and it considers its location in the world irrelevant. For the purpose of display and sensor input computations, we represent locations of objects, targets, and the robot with respect to a world coordinate frame. Our simulation is two-dimensional, ignoring the z axis. Therefore, the tilt of the camera head is not simulated.

In the following subsections, we describe the simulation of the camera input, the sonar readings, and the motion model of the robot.

4.4.1. Camera Pan and Tilt Simulation

For our simulation, we currently do not model projection, back projection, and camera measurements. Instead, we compute the desired angle for the camera pans by transforming the position of the target to the camera frame. The transformation between the camera frame and the world coordinate frame is updated as the robot moves.

4.4.2. Ultrasonic Measurement Simulation

The obstacles in our simulation are represented by their corner locations. For each corner of an object, the position of each corner is transformed to the coordinate frame of the robot. We then compute the angle to this location to determine in which of the ultrasonic measurements this corner will appear. If the new distance, with additive noise, is less than the current minimum distance know by that sensor, then the sensor measurement is updated Given the range ultrasonic sensor in the ring effected by each object allows us to compute the intermediate sonar values.

4.4.3. Motion Control

We model each joint motion as a second-order system. We assume that the joint controller is critically damped and that the discrete inputs from the computer controller are modelled by step input functions. This type of motion is achieved by using a proportional-derivative (PD) controller. These PD controllers have been successful in controlling the vergence of stereo cameras on a robot platform [18]. The motion response to the desired input is shown in Figure 3. The equations of the response function is:

$$\theta(t) = \theta^d \, (1 - \exp(t/\tau))$$

where t is reset to zero when θ^d changes. θ^d is the desired angle of the joint that is computed by our joint motion algorithms described previously. θ^d is a piecewise step function since it is being computed by a discrete controller. τ is the time constant of the system which controls how fast the joint can track the desired input. We also limit the velocity of each joint and we insure that the motion of each joint stays within its range.

Our current parameter values for the time constant and maximum velocity for each joint is summarized below:

$\tau_{cr} = 50$	$\|\omega_{cr}\|_{max} = 90$ deg/sec
$\tau_{cl} = 50$	$\|\omega_{cl}\|_{max} = 90$ deg/sec
$\tau_h = 10$	$\|\omega_h\|_{max} = 60$ deg/sec
$\tau_r = 5$	$\|\omega_r\|_{max} = 30$ deg/sec

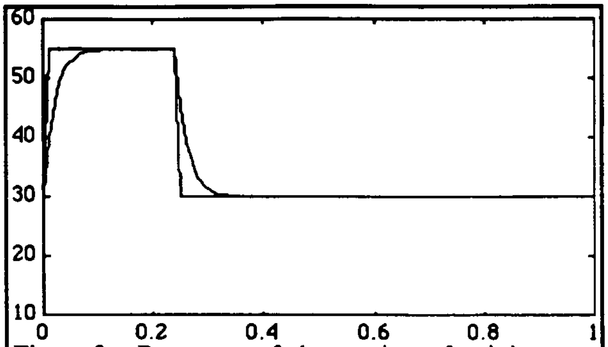

Figure 3: Response of the motion of a joint to a specified input.

4.5. Results

We have run the simulator on numerous examples and we show a couple of results here. In all attempted scenarios, we have successfully arrived at the target location without colliding with obstacles. In the first example, we assumed a stationary target at location (10,7) with respect to the initial robot frame (see Figure 6.) Recall that the x coordinate of the robot frame specifies its direction of motion. Since our slowest time for processing a single frame was 100 milliseconds, we used this time as the sampling period of the system. We assumed that the vehicle could travel a maximum of 3 meters/second.

We present a test sequence where the target is at the limit of the cameras' field-of-view. Therefore, the desired pan of the cameras will be at its largest possible value. We demonstrate to show that the system is stable and controls the head and robot motions smoothly even given the largest step input to the system.

Figure 4 show the motion of the left and right cameras with respect to time. As the robot begins it journey, the cameras first notice that the target is about 40° to the left of the robot. The cameras begin to pan to the target and the head begins to pan to face the cameras toward the target. The system normalizes when the angle of the head and the cameras is small. In this case, the angles between the left and right cameras will become equal in magnitude and opposite in sign. This occurs at about 1 second. This angle magnitude remains close to zero while the target is far away, but as the robot approaches the target the cameras begin to verge. The magnitudes of the two camera angles are still about equal which indicates that the pan of the head is still correctly facing the target. When the mobile robot arrives at the target location at about 4 1/2 seconds the left and right camera angles are verged at -60° and 60° respectively. This angle can be used to compute the distance to the target. When the simulation was allowed to run to acquire the target, the camera angles became -90° and 90° respectively.

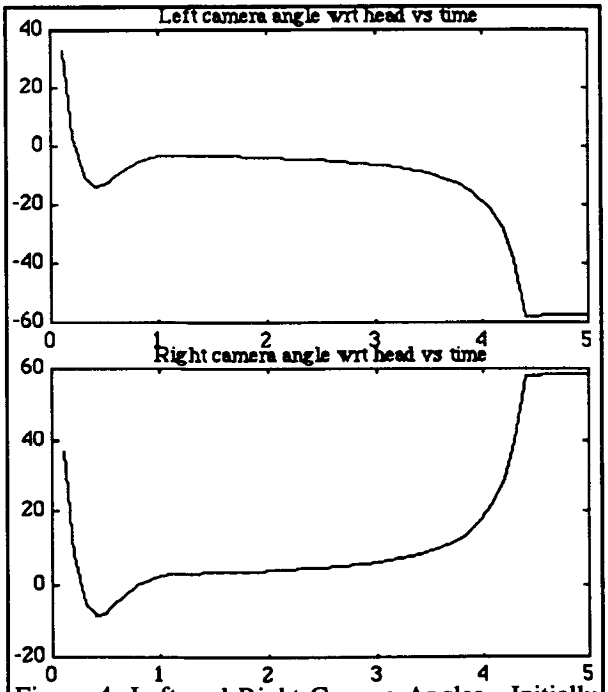

Figure 4: Left and Right Camera Angles. Initially the robot and the camera head are facing away from the target at about an angle of -40°. The cameras and pan stabilize on the stationary target location at about 1 second. From then on the magnitudes of the camera angles are approximately equal. The robot arrives close to the target at approximately 4.5 seconds.

Figure 5 shows the angle of the camera head over time. Confirming what we noticed in the camera angles, the pan motion becomes zero as the cameras are stabilized on the target location at about 1 second. Notice that when the cameras first observe that the target is at 40° the robot head begins to pan to face the cameras toward the target. The pan of the head never gets all the way to 40° since the robot itself also turns in the direction of the pan. As the system stabilizes, the pan of the head is zero since the robot is facing the target.

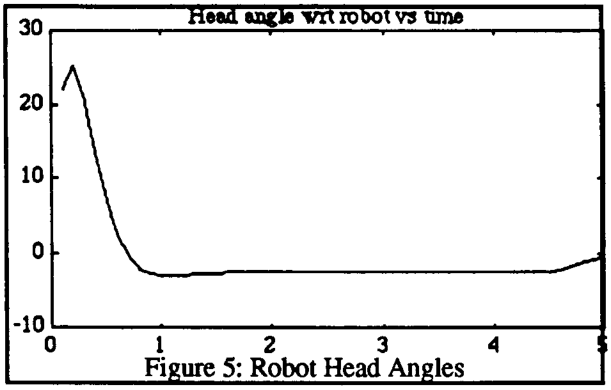

Figure 5: Robot Head Angles

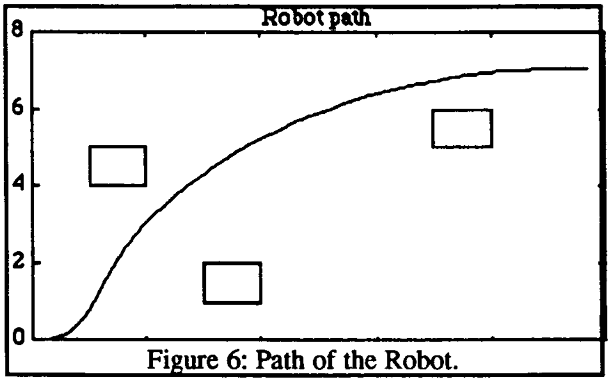

Figure 6: Path of the Robot.

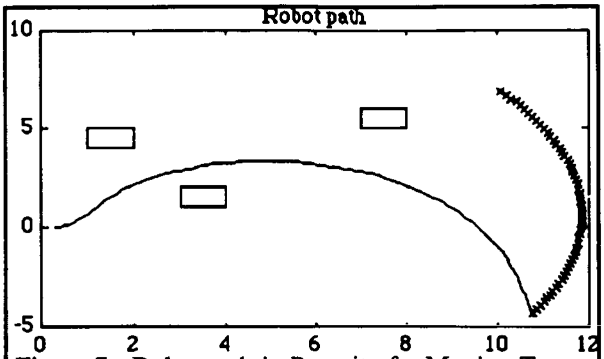

Figure 7: Robot path in Pursuit of a Moving Target

Figure 6 shows the path of the robot to the stationary target at (10,7). The robot avoids a couple of obstacles that were placed close to the straight line path to the goal. Notice that the motion of the robot corresponds to smooth forward trajectories that would be possible with a nonholonomic robot that would be steered similarly to an automobile.

Finally, Figure 7 show the path of the robot tracking a moving target. The target is following a circular path with a changing radius. The target locations, denoted by an 'x', begin at position (10,7) and end at position (10.4, -4.75). The interesting thing is that even though the robot is not estimating the motion of the target, the path developed by the visual pursuit algorithm seems to anticipate the new location of the target and correctly intercepts it.

4.6. Discussion

In our system, the motion of the camera head, panning the two cameras toward the target, is a redundant motion with the steering of the robot. This motion is necessary to allow the robot to freely manuever around obstacles without allowing the target to move outside the field-of-view of the cameras at the maximum camera angles. This gives the robot the freedom to track a target that may even move behind the robot.

The architecture is very simple and provides for much of the navigational and path planning abilities necessary in the system. Unlike other path planning research, we are not focusing on singular conditions in the path planning (e.g. trapping in 'U' shaped obstacle on path to the goal.) This is because our system inherently has a human in the loop, who can select a new intermediate target to move the robot away for the trap.

We discovered that the all the joint motions will oscillate if the response times of the camera pans, head pan, and robot turning are the same. Smooth paths were generated and smooth positioning of the cameras were obtained only if the response of the camera pans are faster than the response of the head pan which in turn is faster than the response of the robot.

5. Deictic Command Simulation

We have also extended our previously described simulation to explore the deictic primitives that are necessary to perform a general purpose navigation. Our goal is is to catalog a large number of environments and the visually interesting or trackable features of the environment. Each environment also has a set of possible goal locations. Using this simulator, we test if the robot can traverse from all starting locations to all possible goals using deictic commands in reference to the visually distintive to the targets.

We read polygonal environment descriptions from an input file. We also mark on these files, objects in the environment which we feel are easily trackable by our video system. We currently have descriptions of a standard living room and the third floor corridors of one of the buildings at Northeastern University.

Currently, we have implemented an approach command where the robot directly approaches the target location. We show examples of paths taken by our robot when commanded to approach a sequence of targets. The data depicts the corridors of the Northeastern University engineering building and we navigate to targets which we feel are trackable by video systems in the corridors. In Figure 8, we show the robot navigating in the corridors from just outside the elevators on the third floor of our Snell building to the doorway between Snell and Dana. The robot is issued three approach commands: The first target is the sign on a vending machine near the end of the first corridor. The second commands approaches a doorknob on the door at the start of the second corridor. The final command approaches the sign on the door at the end of the corridor.

In Figure 9, the robot goes to an office in Snell, again from outside the elevators. The robot first approaches the fire alarms mounted on the wall to the left near the end of the first corridor. Then it approaches a sign on a door office to round the corner. A second alarm becomes the next target, and finally, the poster in the office is used to navigate the robot into the office.

6. Conclusions and Future Work

Our initial work on integrating an active robot head into a navigation scenario has been extremely promising. We have shown that a simple, 'follow your eyes' scenario is sufficient for tracking a moving target. In our situation, we do not plan extensive paths through the field of obstacles but we rely on a low resolution sonar sensor to detect obstacle locations. The motion of the joints on the robot head is smooth and can react to step changes in the target location. We enforce in our simulation a reasonable model of the response of the mechanical systems and the limitations of velocity and acceleration. Because of this modeling of the robot motion latency, the simulation produces realistic paths of the robot.

We are implementing our algorithms on our hardware platform and intend to develop algorithms for obstacle detection using the active robot head. We will test this algorithm extensively to determine what steps we will need to improve the algorithm to acheive better performance in many environments. We will also begin working on vision algorithms that can robustly track many targets. We want to develop a number of visually directed commands useful for general navigation. Later, we will extend this work to include targets and orientation constraints. We hope to eventually develop a set of visual commands for manipulation as well.

Not only does this system provide solutions in current semi-autonomous applications, it is also an alternative philosophy for developing fully-autonomous, general-purpose mobile robot systems. Many researchers are developing autonomous mobile robots which can navigate in limited situations, for example road-following or corridor tracking. Their philosophy is to merge autonomous systems performing specific tasks and to derive a general purpose autonomous system. We, on the other hand, are developing a robust mobile robot which can navigate in general situations. To make general mobility possible, our system will rely on more human interaction than typical mobile robot systems. Over time we will decrease the amount of user interaction by adding general environmental knowledge to the system thereby increasing the autonomy of the system. This will result in systems that are easily configured to a number of applications including underwater and space exploration, flexible manufacturing, and robotic wheelchairs.

References

[1] R.C. Arkin, "Motor Schema-Based Mobile Robot Navigation," *Inter. Journal of Robotics Research*, vol. 8, no. 4, 99-112, Aug 1989.

[2] R.C. Arkin, "Navigational Path Planning for a Vision-Based Mobile Robot," *Robotica*, vol. 7, 49-63, 1989.

[3] J. Borenstein and Y. Koren, "Teleautonomous Guidance for Mobile Robots," *SMC*, vol. 20, no. 6, 1437-1442, Nov/Dec 1990.

[4] R. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Trans. Robotics and Automation*, vol. RA-2, no. 1, 14-23, 1986.

[5] C.M. Brown, "The Rochester Robot," Tech. Rep., Computer Science, 257, Sept 1988.

[6] J.D. Crisman, "Color Vision for the Detection of Unstructured Roads and Intersections," Ph.D. thesis, Carnegie Mellon University, Dept. of Electrical and Computer Engineering, May 1990.

[7] J.D. Crisman, Y. Du, and M. Cleary, "Adaptive Control of Camera Position for Stereo Vision," in *Optics, Illumination, and Image Sensing for Machine Vision VIII*, SPIE, Boston, MA, September 1993, invited paper.

[8] J.D. Crisman and Y. Du, "Generic Target Tracking Using Color," in *Intelligent Robotics and Computer Vision XII: Active Vision and 3D Methods*, SPIE, Boston, MA, September 1993, invited paper.

[9] E.D. Dickmanns, B. Mysliwetz, and T. Christians, "An Integrated Spatio-Temporal Approach to Automated Visual Guidance of Autonomous Vehicles," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, no. 6, 1273-1284, Nov/Dec 1990.

[10] J.T. Fennema and O.W. Mitchell, "Vision Guided Servoing with Feature-Based Trajectory Generation," *IEEE Trans. Robotics and Automation*, vol. 5, no. 5, 691-699, Oct. 1989.

[11] C. Fennema, A. Hanson, E. Riseman, J.R. Bevride, and R. Kumar, "Model Directed Mobile Robot Navigation," *SMC*, vol. 20, no. 6, 1352-1369, Nov/Dec 1990.

[12] R.J. Firby, "Adaptive Exectution in Complex Dynamic Worlds," Ph.D. thesis, Yale University, 1989.

[13] M. Hebert, T. Kanade, E. Krotkov, and I.S. Kweon, "Terrain Mapping for a Roving Planetary Explorer," in *Proc. IEEE Robotics and Automation Conf.*, Scottsdale AZ, May 1989, pp. 997-1002.

[14] B.H. Krogh, "A Generalized Potential Field Approach to Obstacle Avoidance Control," in *Proc. Robotics Inter. Robotics Research Conv.* , Bethlehem PA, 1984.

[15] E. Krotkov, *Active Computer Vision by Cooperative Focus and Stereo*. Springer-Verlag, 1989.

[16] D. Marr, *Vision*. W. H. Freeman and Company, 1982.

[17] H.K. Nishihara, "Practical Real-Time Imaging Stereo Matcher," *Optical Engineering*, vol. 23, no. 5, 536-545, 1984.

[18] T.J. Olson and D.J. Coombs, "Real-Time Vergence Control for Binocular Robots," Tech. Rep., Computer Science, 348, June 1990.

[19] D.W. Payton, "Internalized Plans: A Representation for Action Resources," *Robotics and Automonmous Systems*, vol. 6, 89-103, 1990.

[20] L.G. Roberts, "Machine Perception of Three-Dimensional Solids," in *Optical and Electro-Optical Information Processing*, Cambridge MA: MIT Press, 1965, chap. 9, pp. 159-197.

[21] C. Thorpe and J. Gowdy, "Annotated Maps for Autonomous Land Vehicles," in *Proc. DARPA IUS Workshop*, Sep 1990, pp. 765-771.

[22] L.E. Weiss, A.C. Sanderson, and C.P. Neuman, "Dynamic Sensor-Based Control of Robots with Visual Feedback," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 5, 404-417, October 1987.

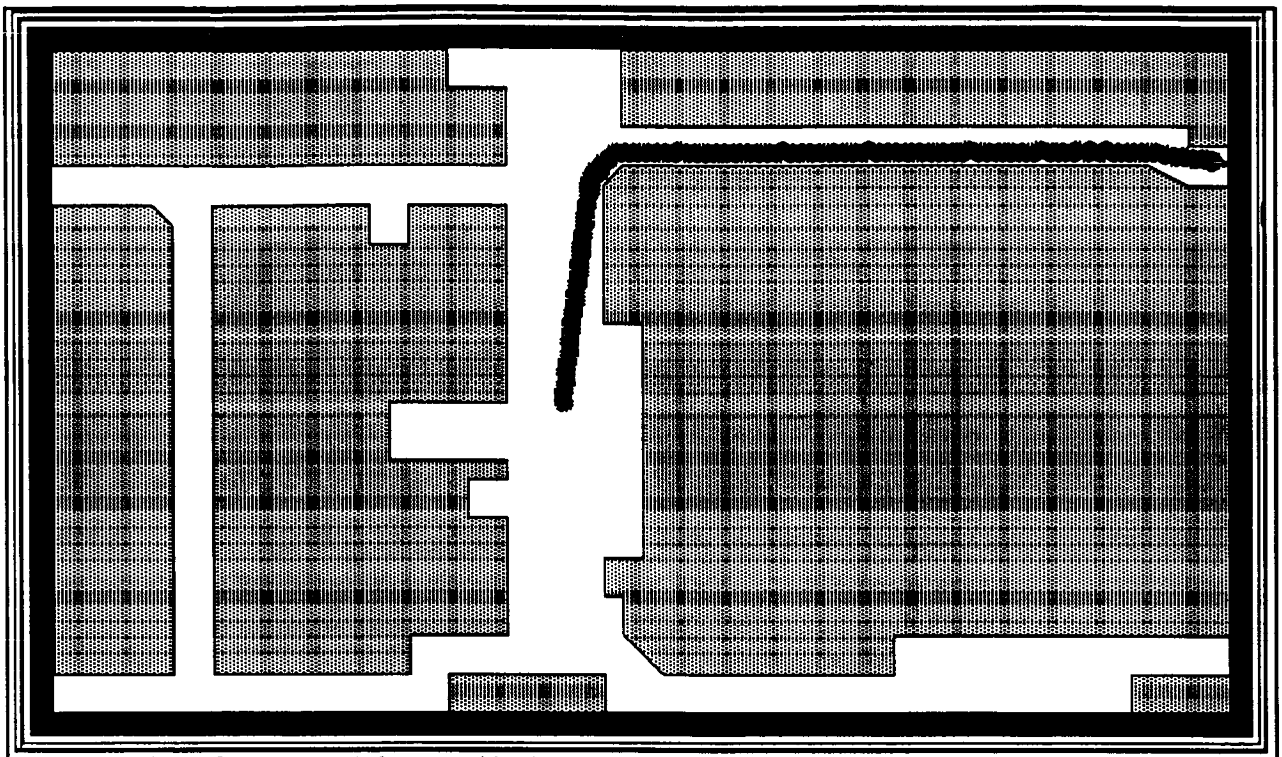

Figure 8: Robot path from outside elevator to the door between the Snell and Dana buildings.

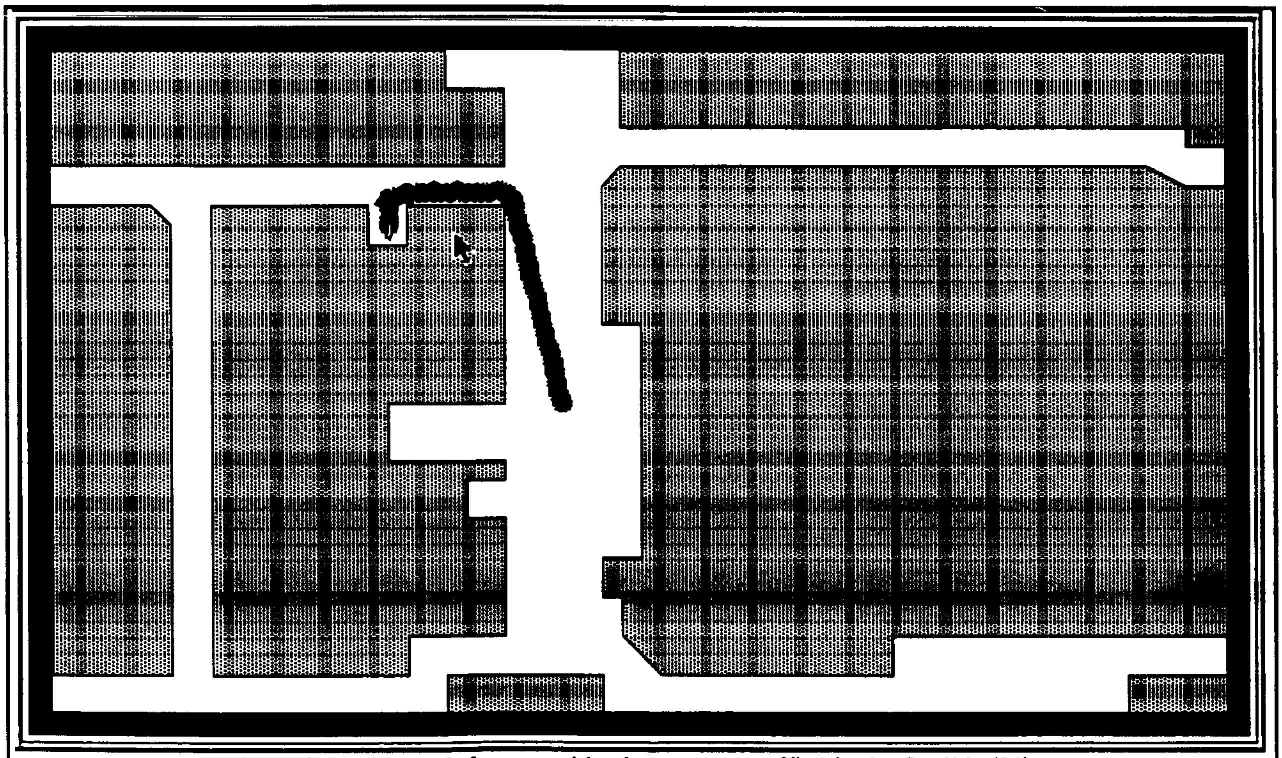

Figure 9: Robot path from outside elevator to an office in the Snell building.