

C H A P T E R VIII

EXPERIMENTS

§1. Introduction

This chapter describes the various experiments that were used to demonstrate the different components of AuRA. The test domains included the second floor of the Graduate Research Center (Fig. 12) and its immediate outdoor surroundings (Fig. 13).

Several different types of experiments were performed. Some involved the testing of individual components of the architecture in isolation. In particular, the relatively slow vision algorithms were exercised in this manner. Other components were tested in a system framework, e.g. the ultrasonic/encoder schema-based navigation. Management of the robot's uncertainty was also experimentally tested. The last experiment tests several of the different AuRA components working on the same task.

§2. Motor schema-based navigation

A real-time schema experimentation/demonstration system was developed based on sensing using ultrasonic and encoder data. The hooks for tying in visual sensing are in place but are currently not implemented (due to the slow processing speeds for vision).

Five different motor schemas have been implemented: **move-ahead** (encoder based), **move-to-goal** (encoder based), **avoid-static-obstacle** (ultrasonic based), **noise** (sensor independent), and **follow-the-leader** (ultrasonic based). The user is able to select the collection of motor schemas to use and associate a perceptual schema with each. After schema selection is complete, robot motion is initiated. The vehicle then behaves in an intelligent manner in response to its environmental stimuli. Several of the more interesting behaviors are described below. Although the available videotape of mobile robot behavior

shows more than words allow, the following subsections and still photograph sequences describe these behaviors as well as we can. It should be noted that the schemas for this system emulate distributed processing, but actually are evaluated sequentially.

§2.1 *Avoidance*

By instantiating the **avoid-static-obstacle** schema with ultrasonic perception, the robot manifests an interesting behavior. The schema instantiation can be controlled by altering both the gain, which affects the velocity of the vehicle, or the sphere of influence of detected obstacles, which increases its sensitivity to the environment. When activated, the robot seeks out a potential field minimum and remains there unmoving (or slightly oscillating in cluttered environments due to the limited sampling rates and noise in the sensor data).

If a change in the environment occurs, (e.g. a person approaches), the robot is repulsed and seeks out a new potential minimum. The robot can be “herded” by following behind it, forcing it to move to a desired location. It avoids obstacles during its journey and then settles into its new location when the environment stabilizes. Figure 81 illustrates this process.

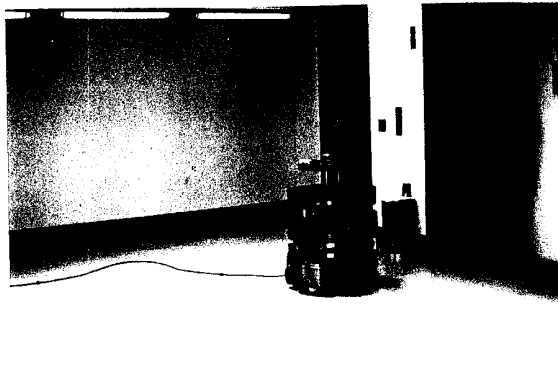
§2.2 *Exploration*

By combining the **noise** schema with the **avoid-static-obstacle** schema, exploration behavior can be observed. The **noise** schema’s gain (strength) and persistence (how frequently the direction changes) can be set at startup by the experimenter. The robot meanders about the lab exploring different regions while avoiding collision with the obstacles. (Fig. 82).

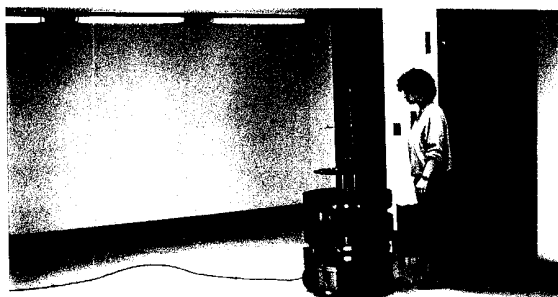
The robot responds quite well in response to changes in its environment, as when people surround it during a demonstration. The biggest problem HARV faces is the slow sensor sampling which makes its reflexes quite slow. The vehicle can also be herded when running in this behavioral mode, but it is not quite as obedient, moving in the general direction forced upon it but occasionally making some slight sidesteps due to the presence of noise. Actually this form of herding is more reminiscent of an animal’s behavior, and HARV has been likened to a sheep by some observers when it’s running in this mode.

Figure 81: Avoidance behavior

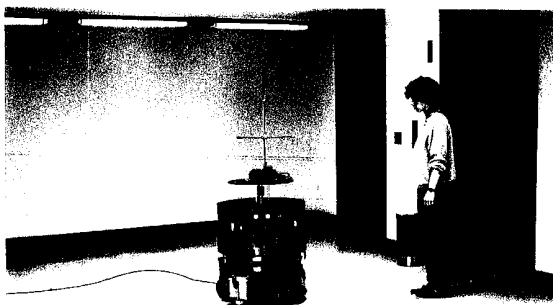
In this sequence, the robot is initially stationary (a). When the robot is approached (b), it is repulsed and moves to its new position and then stops (c). This behavior is produced by the instantiation of an avoid-static-obstacle schema.



(a)



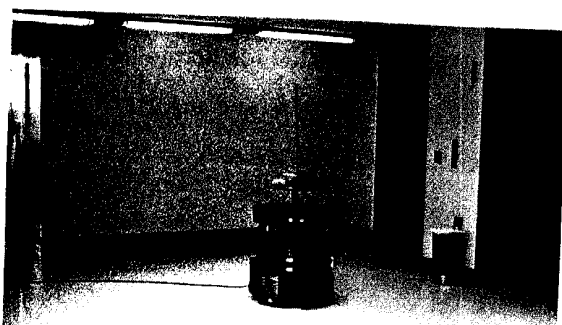
(b)



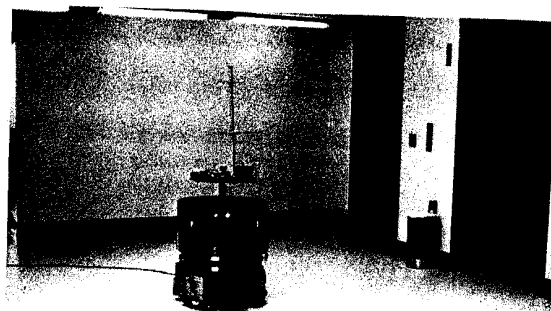
(c)

Figure 82: Exploration

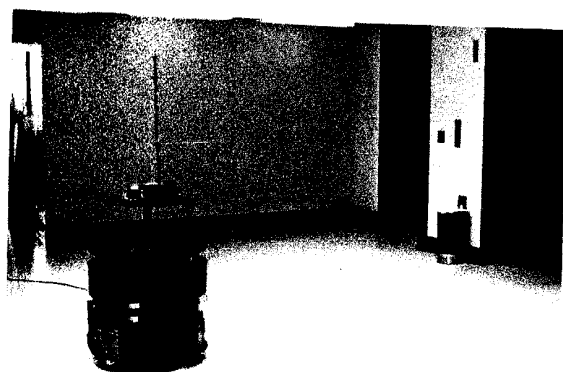
Exploration behavior is produced by combining avoid-static-obstacle and noise schemas. The robot wanders randomly while avoiding collisions. (a-c). The last 3 photos (d-f) show the robot being "herded" towards the wall.



(a)



(b)



(c)



(d)



(e)



(f)

§2.3 *Hall following*

By instantiating the **move-ahead** schema with an **avoid-static-obstacle** schema, the robot is able to safely travel down a hall (Fig. 83). Although there is no model of the hall (at the schema level), the robot senses the two sides as obstacles and positions itself near the center. The **move-ahead** schema drives it forward.

HARV responds to changes in the hall structure itself such as pillars, seeking out the middle of the hall. Small obstacles placed near the wall's sides are interpreted as part of the wall by the ultrasonic sensors. Open doorways pose no problem as long as the **move-ahead** direction is roughly parallel to the hall. If this direction points steeply into the wall, the robot might pass through an open door (see door entry below), but since the angle into the wall typically must be at least 45 degrees or more for normal doorway entry, this allows a latitude of about 90 degrees (± 45) of error in the **move-ahead** direction for successful navigation of halls using this method.

§2.4 *Navigation in the presence of obstacles*

The same combination of schemas allows for navigation in cluttered hallways or outdoor situations. The **move-to-goal** schema can be substituted freely for the **move-ahead** schema if desired. Figures 84 and 85 show the robot's course through a series of cluttered obstacles both indoors and outdoors. The **move-to-goal** schema relies on the shaft encoders to move the robot to a particular point. This is accomplished by specifying the distance to and direction towards the goal and then monitoring the encoder data to provide the input to the **move-to-goal** SI.

The robot's minimum detectable ultrasonic sensor reading is 0.9 feet. For this reason, whenever the ultrasonic sensor returns a value of 0.9, the robot must consider a collision to be imminent. This effectively increases the robot's diameter by almost two feet, making it more difficult to squeeze through tight spaces. This is particularly evident in the indoor hallway examples. A consequence of this fact, when coupled with the slow sampling rates for ultrasonic data (2-3 seconds for a complete scan, transmission, and interpretation by the VAX), is the production of motion oscillations when the robot is operating under continuous motion in constricted areas. The vehicle moves from side to side in a dance-like motion as it squeezes through the congested spot. The robot still

Figure 83: Hall following

This 5-photo sequence shows the robot moving down the hall under the control of move-ahead and avoid-static-obstacle SIs. Note that it detects and steers away from obstacles along the way. The robot maintains its center position both by virtue of the move-ahead SI and the fact that the walls of the hallway itself are repulsive due to the avoid-static-obstacle SI.



(a)



(b)



(c)



(d)



(e)

Figure 84: Navigation (Indoor)

In a more complex environment than that of Figure 83, the robot still succeeds in its navigational task of moving down the hall while avoiding collisions in the considerably more cluttered environment. (The balls on top of the cones make them better sonar targets, as the height of the traffic cones is just short of the height of our ultrasonic sensors in their particular geometric configuration).

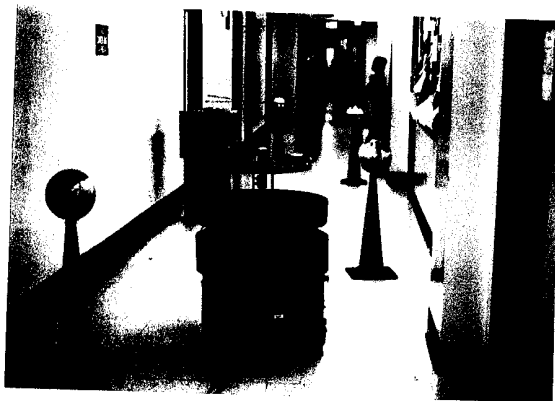


Figure 85: Navigation (Outdoor)

The same type of navigation can be performed outdoors as well. In this case, a move-to-goal SI (shaft-encoder based) was used instead of a move-ahead SI, the goal being a location near the distant cone that does not have a ball on it. The robot's course winds through the obstacles while making its way towards the goal.



attains its navigational goal and does not crash into obstacles, but it takes more time than would otherwise be necessary. This problem will vanish when the processing and communication speeds are improved. On the other hand, if the velocity of the robot is increased over the 0.3-0.4 feet per second used for these examples, the problem would be exacerbated. To attain higher velocities, faster processing is necessitated.

The experimental schema system offers three types of motion: step-by-step mode where each motor step must be approved by the operator (used chiefly for debugging new schemas), lurch mode where the robot waits approximately 2 seconds per step while sensor processing is completed between moves, and continuous motion where the robot acquires sensor data while moving. The problem with continuous motion is that the vehicle changes its position in the two or three seconds it takes to process the sensor data. Generally this is not a problem, but it can give rise, especially in tight quarters, to the oscillatory situations described above. All of the behaviors shown in this section work well in continuous motion with the possible exception of door entry (described below), again due to the tight quarters.

§2.5 *Single wall following*

A "drunken sailor" walk can be produced by directing a **move-ahead** schema into a wall with the obstacle avoidance behavior active (Fig. 86). The robot slides along a repulsive field a specified distance from the wall, reacting to obstacles as it moves. This allows the robot to enter doorways if the vector pointing into the wall is sufficiently large in magnitude and its angle is sufficiently steep. (Fig. 87).

With normal doorways, it is difficult to get the robot to enter when it is operating in continuous mode. As stated above, this is a consequence of the data being old relative to the robot's current position. It also requires some finesse in proper selection of gains and angle of attack to produce smooth door entry behavior. With a 45 degree angle of attack and 1.3 gain (1.0 is baseline) for the **move-ahead** schema, and a sphere of influence of 2.5 feet and gain of 0.7 for the **avoid-obstacle** schema, good results have been obtained.

§2.6 *Impatient waiting*

The potential fields methodology is strongly in evidence when the robot moves into a box canyon. The robot ends up in a potential well and is not able to make meaningful

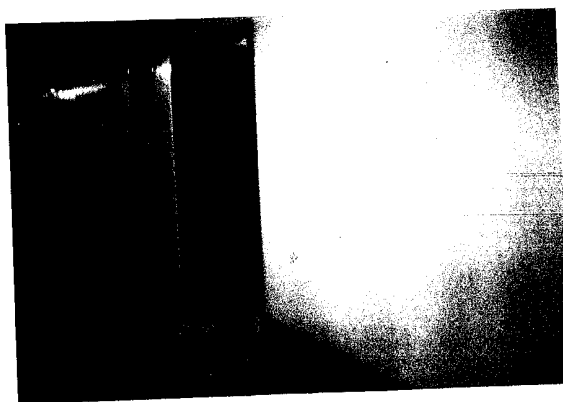
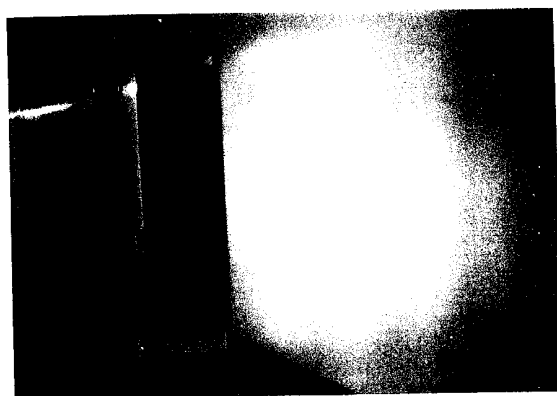
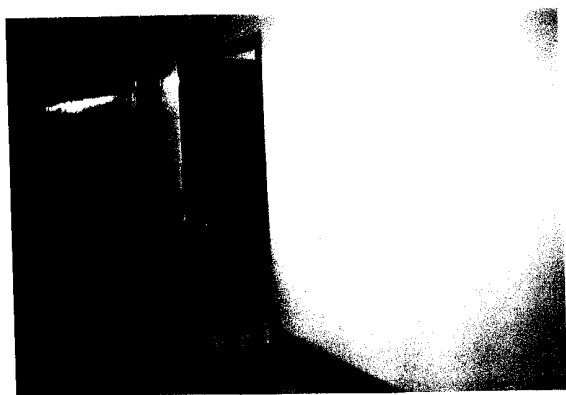
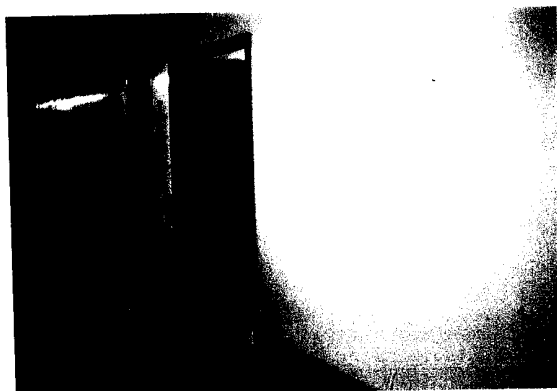
Figure 86: Wall following (“drunken sailor”)

This behavior is produced by the combination of a move-ahead SI directed obliquely into the wall while keeping the obstacle avoidance behavior active. In the sequence shown, the robot follows the single wall, moving around the obstacle, and then staggers towards the staircase. The behavior appears as if HARV was leaning against the wall while moving forward, conforming to its contours. HARV would have fallen down the stairs (to the right), if it was not shut off by our technician (classic “drunken sailor” behavior).



Figure 87: Door entry

The tendency of the robot to move through openings using the combination of behaviors described in Figure 86 can also be used to advantage in going through doors. In this sequence, the robot follows the wall and then enters the open door.



progress. Normally, after a time-out occurs based on a hard real-time deadline, the pilot and/or navigator would be reinvoked to compute an alternate path. Suppose, however, the blockage is temporary, due to closed doors at the end of a hall or an elevator. The robot can wait, rocking around in its potential well, until the obstruction is removed. This behavior is similar to what might be evidenced by a fly at a window. When the path becomes unblocked, the robot continues ahead, trying to satisfy its initial goals. Figure 88 depicts this process.

It is also possible to embed a temporal delay schema that turns off the robot's motion for a specified time interval, letting the robot deliberately wait before restarting. This incorporates aspects of patient waiting in addition to the impatient waiting described above.

§2.7 *Follow the leader*

HARV can track a moving object (with additional obstacle avoidance behavior to aid maneuvering in tight situations if desired). The appearance is similar to walking the robot on an invisible leash (Fig. 89). It turns and tracks the nearest object within a given ultrasonic sensor spread, moving more rapidly as the distance between robot and object increases, but avoiding contact with the tracked object. The potential well (the ideal distance separating the tracked object and the robot) is user specifiable as is the separation distance at which the robot abandons following. Vision algorithms, well suited for this particular type of tracking, will hopefully be exploited for this purpose when suitable real-time image processing hardware arrives.

§2.8 *Motor schema experiment summary*

The motor schema experimentation/demonstration system enables visualization of motor schema-based potential field navigation. The greatest drawback lies in the relatively slow sampling rates for processing the ultrasonic data on a shared VAX-750 (typically 2-3 seconds for 24 sensor readings) causing minor oscillations of the vehicle in highly dynamic or extremely cluttered environments when continuous motion is allowed, since the sensor data becomes old relative to the robot's position.

Enhancements would include the addition of other perceptual schemas using vision, particularly if they can be made to function more rapidly than the current implementa-

Figure 88: Impatient waiting (“fly at a window”)

This sequence shows what can occur when the robot enters a potential well. A move-ahead SI directs the robot to the end of the hall, while the avoid-static-obstacle behavior keeps it from colliding with the door. The robot roves impatiently at the end of the corridor until the door is opened. At that point, it moves through the door and continues on its way.

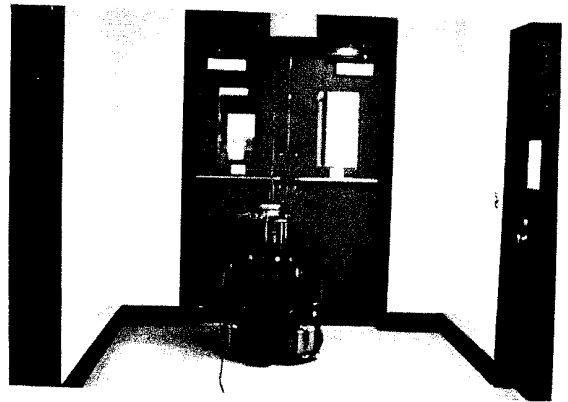
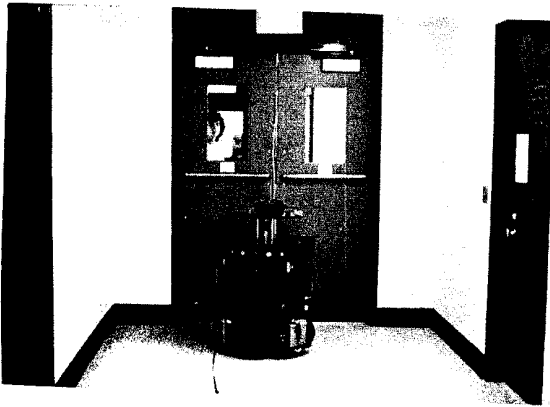
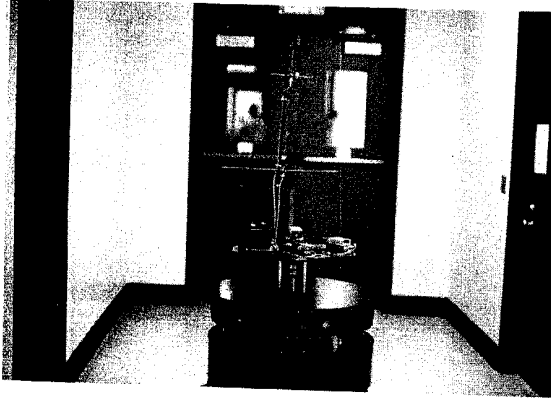


Figure 89: Follow-the-leader behavior

Although difficult to show in a sequence of still photos, the robot can be seen to follow Robyn across the floor, turning when she turns. When she stops (d-e), the robot settles into a potential well a short distance from her and then stops. This behavior mimics walking a dog, but with an invisible leash.



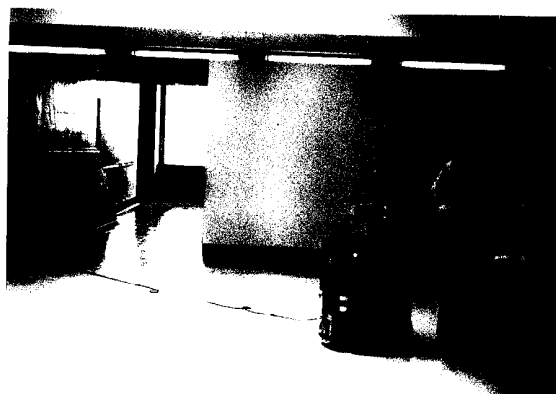
(a)



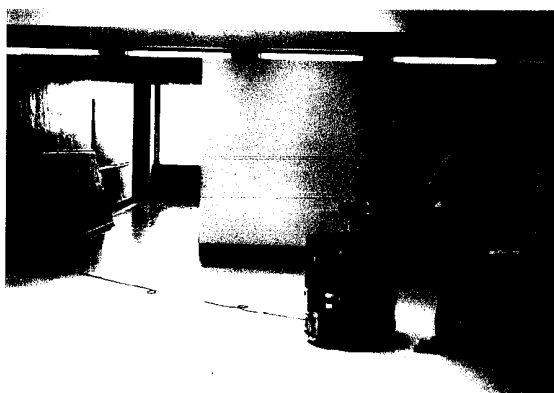
(b)



(c)



(d)



(e)

tions of the vision algorithms described below. Converting the system to a distributed environment will reduce the delays incurred by the sequential evaluation of schema velocity vectors when using the VAX, further enhancing the robot's sensitivity to its world. It can be seen however, that by using combinations of relatively primitive behaviors, sufficiently intelligent action can be undertaken by the robot to successfully navigate through its world.

§3. Visual navigation experiments

The vision algorithms were tested individually, principally because of the longer times required for their processing. Three algorithms were tested including path following using the fast line finder, path following using the fast region segmenter, and obstacle detection using the depth-from-motion algorithm. Path following cannot be implemented using the DRV's ultrasonic sensors, thus the visual path detection algorithms significantly increase the robot's ability to navigate in an outdoor environment. The details of these algorithms are described in Chapter 6 and Appendix C. This section presents the experimental results.

§3.1 *Path following using fast line finding*

The robot was tested on both an interior hallway and an asphalt sidewalk outside the GRC. In both cases the robot successfully extracted the path edges and servoed itself to remain on the path. (See Figs. 90-93). The bootstrap position of the path edges was provided for the the robot by the experimenter in each case. All the feedforward path edge positions were computed automatically. In the indoor run (Figs. 90-91), the robot was initially started at the side of the hall, quickly found the centerline, moved to the hall center, and then continued to follow the hall successfully. Near the end of the corridor, the image contained little path edge for the robot to follow and the run was terminated.

In the outdoor run (Figs. 92-93), similar behavior was observed. Since the UHF transmitter was not operational for these runs, the testing was limited to those paths that were accessible using our 500 foot cable tether. Whenever a path had reasonably well defined edges, the fast-line finder was able to provide the information necessary for visual servoing. It failed however for the case of the gravel path which had quite

Figure 90: FLF path following - indoors

This sequence of photos shows the robot finding the centerline of the hallway and then maintaining its position on the centerline. The algorithms described in Chapter 6 (FLF, line fragment grouping, centerline computation, servoing, etc.) produce this well-defined behavior. Figure 91 shows the robot's view of the hall.

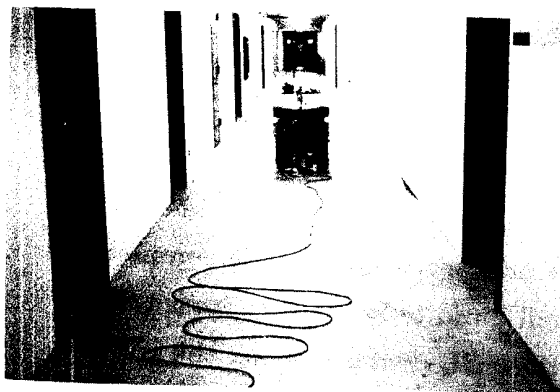


Figure 91: FLF path following - indoors (continued)

The lefthand images show the results of the fast line finder with the buckets tuned to the anticipated hall edges, and the righthand images show the extracted left and righthand path edges and the computed centerline used for servoing (see Fig. 90). Images (a-b) show the initial position of the robot, images (c-d) show the view after the first servo sequence is completed, (now in the center of the hall), and (e-f) show an image taken from much further down the hall, demonstrating the robot's continuing ability to extract the path edges.



(a)



(b)



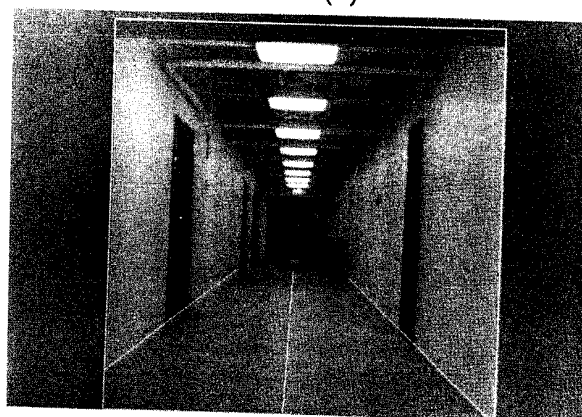
(c)



(d)



(e)



(f)

Figure 92: FLF path following - outdoors

The robot, using the same algorithm as in Figures 90-91, navigates along an asphalt sidewalk. The robot is started off to the left, then after finding the path, moves to the center of the sidewalk, and then remaining in the center of the sidewalk for the rest of the run.

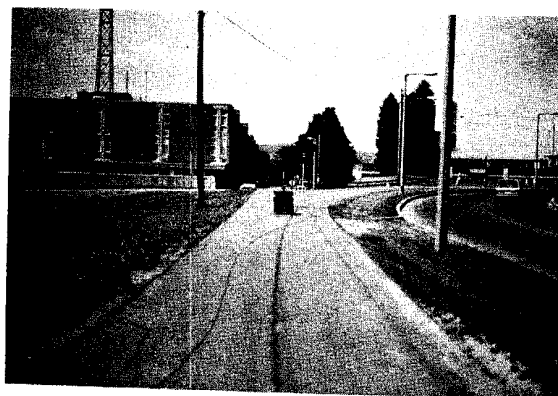
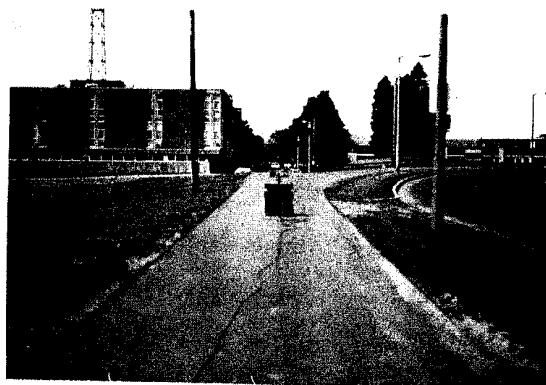
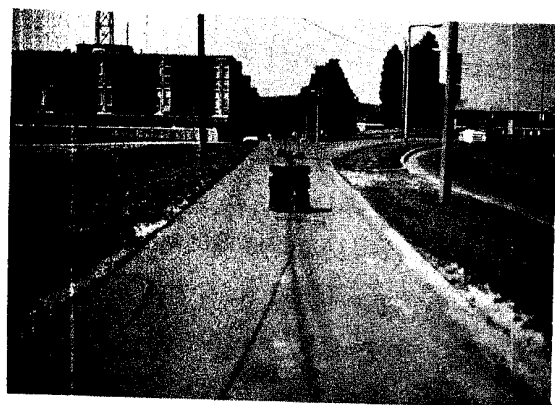
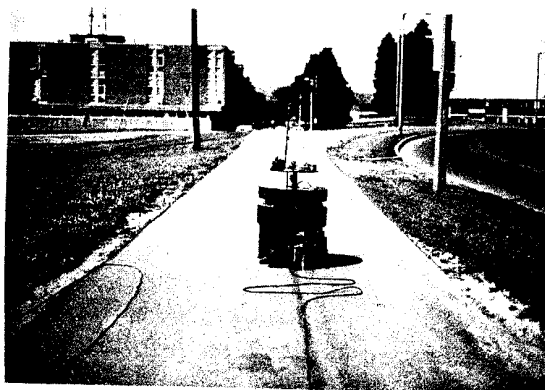
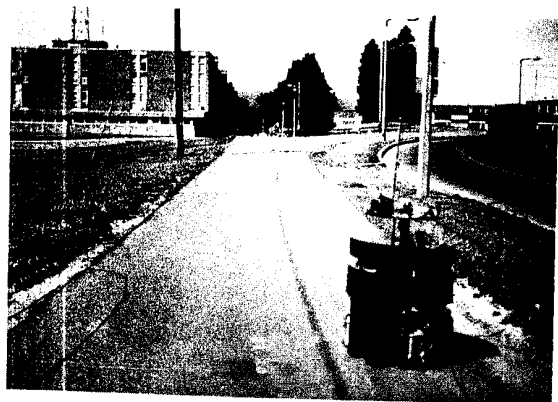


Figure 93: FLF path following - outdoors (continued)

The lefthand images show the results of the fast line finder with the buckets tuned to the anticipated sidewalk edges, and the righthand images show the extracted left and righthand path edges and the computed centerline for servoing (see Fig. 92). Figures (a-b) show an image taken from the initial position of the robot (off to the side of the path) that is used for the first step in path following, Figure (c-d) shows the view after the first visual servo sequence is completed and the second step of FLF processing done, (the robot is now in the center of the sidewalk) and Figures (e-f) show the results from an image taken from much further down the sidewalk, showing the robot's continuing ability to extract the path edges.



(a)



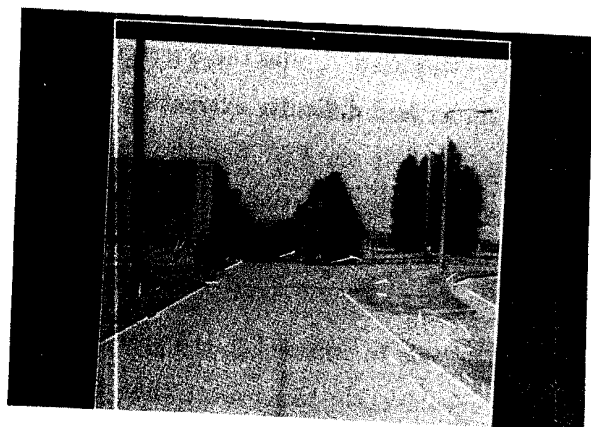
(b)



(c)



(d)



(e)



(f)

indistinct edges. Visual path following was still attainable for this case however, using the fast region segmenter as the perceptual process (see section following).

Topography plays a role in outdoor navigation using the FLF. If the ground drops off due to a change in the terrain's slope, the vanishing point of the path appears above the actual horizon. As a consequence, lines may be extracted in trees, buildings, etc., which are added to the line fragment collection and may cause the extracted path edges to deviate from their actual position. This problem can be obviated by pointing the camera downwards, so the horizon is above the field of view, but this will make landmark recognition from the same image difficult, as the road becomes the most prominent feature. Another alternative is to use high-level knowledge that reflects the topography to lower the horizon and remove any extraneous line fragments before combining them to produce the path edges. A more sophisticated approach would involve geometric grouping of long collinear lines as in [143].

The FLF algorithm as applied to path following, however, is quite robust and requires little tuning to produce good results. For the runs above, a 40 degree range for line orientation proved more than adequate. The expectations in general do not have to be highly constrained to produce good, working results. This robustness is largely a consequence of the use of gradient orientation as opposed to gradient magnitude as the means for producing the line fragments.

§3.2 *Path following using fast region segmentation*

The robot was placed in a similar test environment as for fast line finding. In this case, however, a gravel path with quite indistinct path borders was used. Under these irregular conditions, at the image resolution used, the fast line finder had difficulty extracting the path borders (smoothing the image prior to processing may help the FLF find the path). Using the fast region segmenter (FRS), the robot successfully negotiated this course (Fig. 94). Figure 95 shows the operation of the FRS algorithm on the images used during this run.

This algorithm, being sensitive to the gradient magnitude as opposed to the gradient orientation as in the case of the fast line finder, is considerably less robust than the FLF to changing lighting conditions and thus is not particularly well suited for outdoor navigation. Changing lighting conditions (e.g. passing clouds) make this algorithm, as

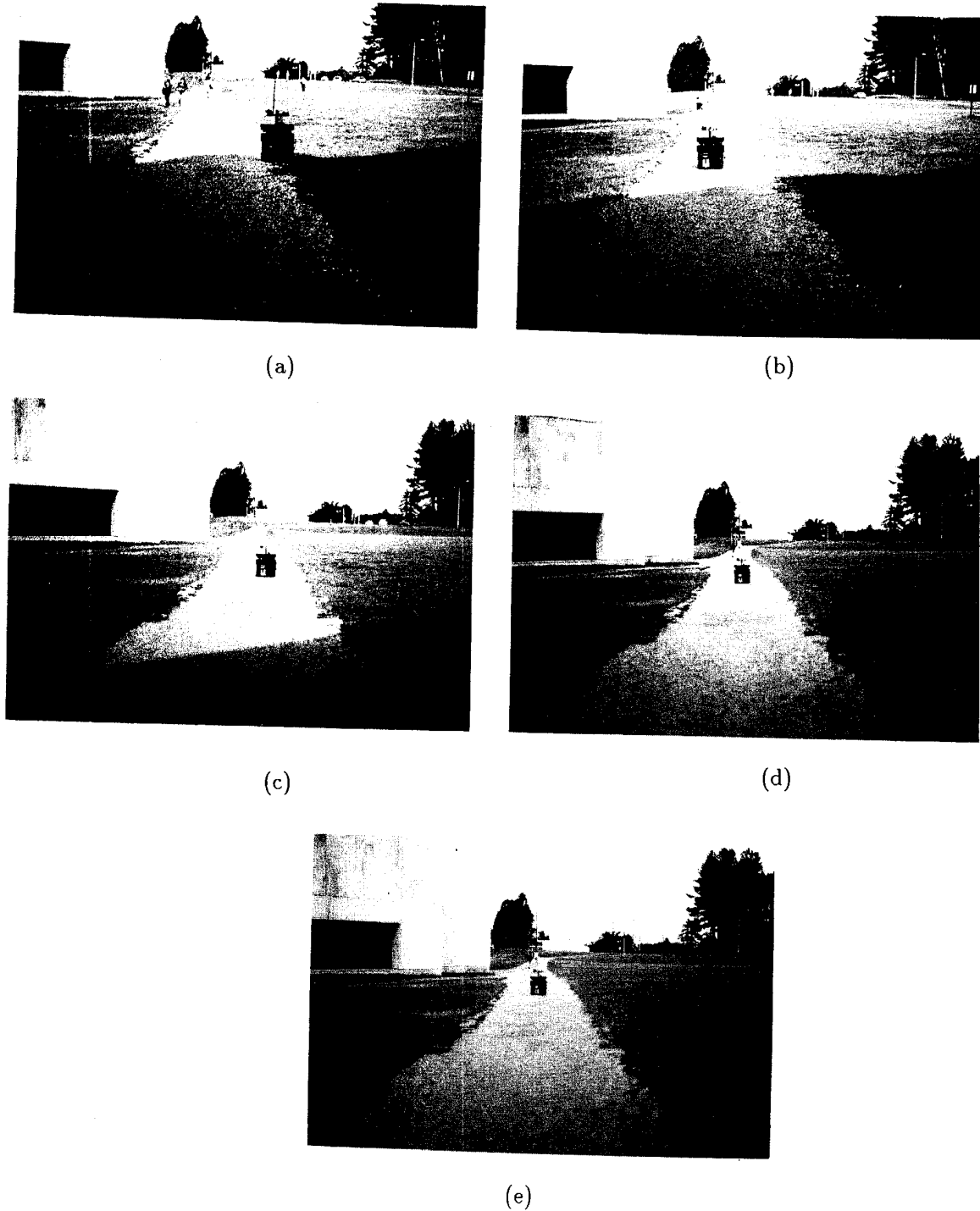
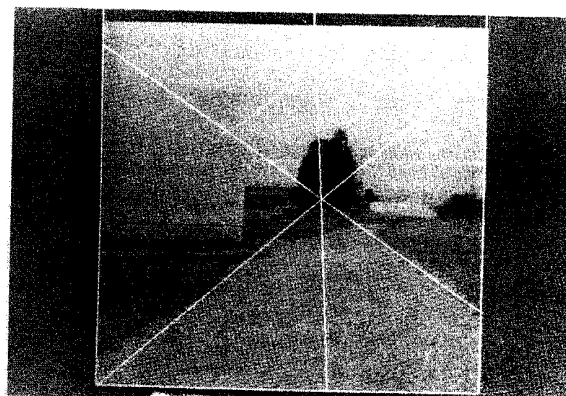
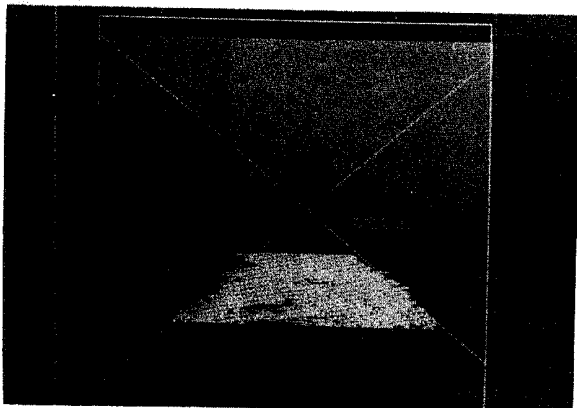
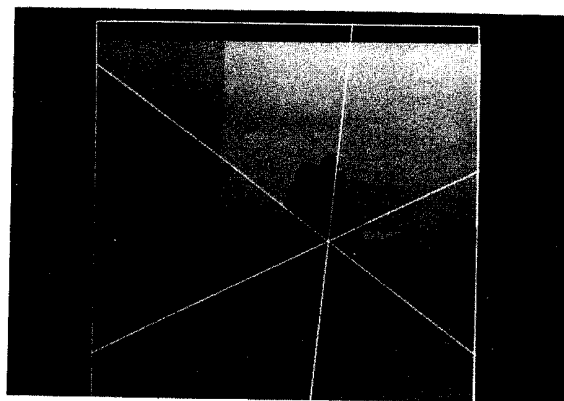
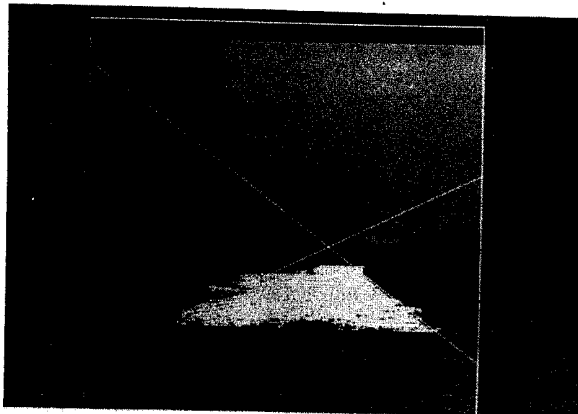
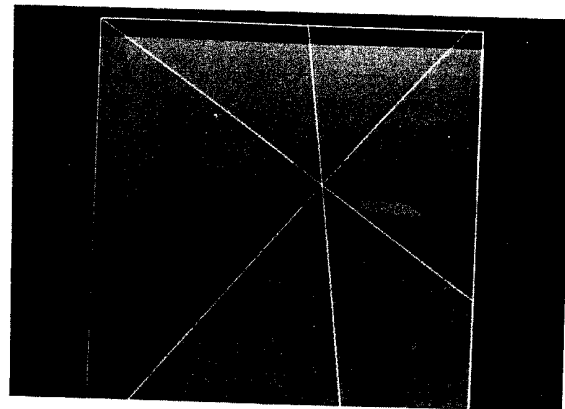
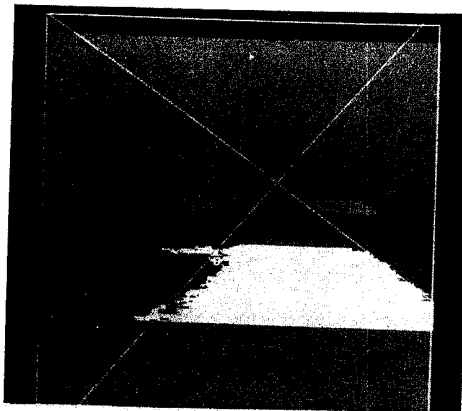
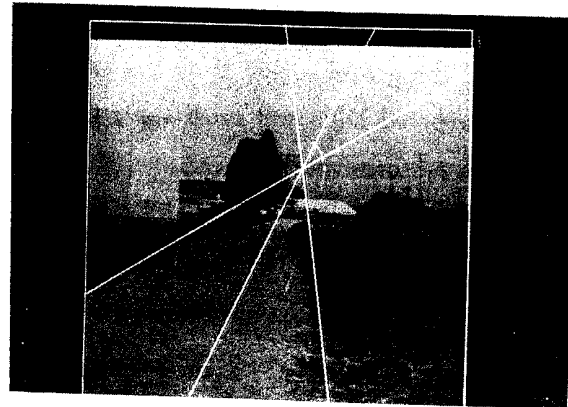
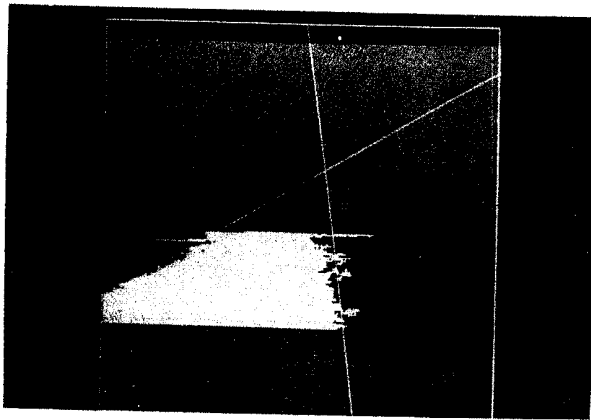


Figure 94: FRS path following
This sequence shows the robot's path when navigating under the fast region segmentation algorithm over a gravel path.

Figure 95: FRS path following - (continued)

The lefthand images show the results of the fast region segmentation. Lines are fitted to the left and right sides of the region using the least squares method. The righthand images show the computed centerline used for servoing purposes. Only the central portion of the path was extracted to obtain more consistent and uniform intensity values for the segmentation.



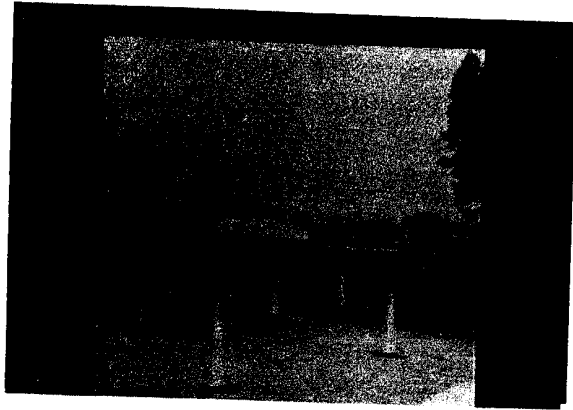
it currently stands, susceptible to failure. The use of color images would significantly increase its versatility as additional spectral information would aid in the separation of the gravel region from the surrounding grassy areas. For indoor environments with carpeted halls and predictable lighting conditions, the FRS algorithm can be expected to perform well.

§3.3 *Depth-from-translational-motion results*

The depth-from-motion algorithm was tested with images acquired from HARV, but was not used to drive the vehicle. This algorithm in its current hardware implementation is far too slow to provide reasonable control of the robot. Two image sequences were used, the first outdoors and the second indoors. The outdoor images were hand-registered (an automated registration process is being developed [106]) while the indoor sequence was used as is (no registration was performed). Proper image registration is crucial to accurate recovery of depth. Although successful point tracking can occur even when image sequences are not well registered, the actual depth values may be inaccurate. In both cases the FOE was automatically extracted. The three image sequences themselves are shown in Figures 96 and 99. The tracked points and extracted obstacles are shown in Figures 97 and 100. Depths to several of the extracted and tracked obstacle points are tabulated in Tables 4 and 5. Motor schemas were then attached to the obstacles in the outdoor case in a simulation run and produced the path shown in Figure 98.

The major problem associated with accurate depth recovery, as stated in Chapter 6, is the difficulty in obtaining accurately registered images. HARV acquired the outdoor image sequence while traveling over a concrete pad used as a parking lot at the GRC. The pad itself is not completely level and there is considerable gravel and dirt covering it. This caused the camera to roll somewhat from one image acquisition to the next. The manual registration process used for the outdoor sequence did not attempt to correct for this rotation in the image plane. Only image translations were used to minimize the misregistration. Pitch and yaw movements of the camera can only be partially removed by image translation [106]. Roll of the camera, nonetheless, still remains the biggest problem in registration.

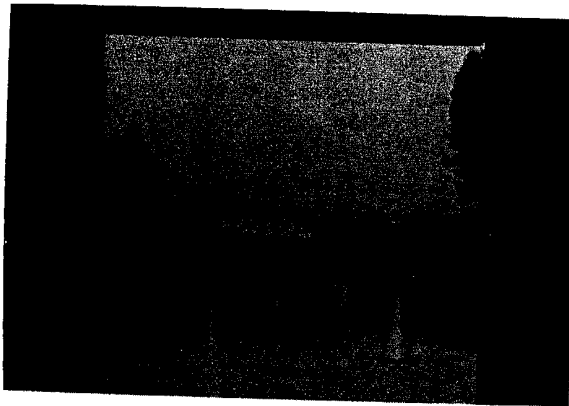
The numerical results (Tables 4,5) are somewhat discouraging. Anomalous behavior, such as the depth to an object increasing as the robot approaches, can be seen in the



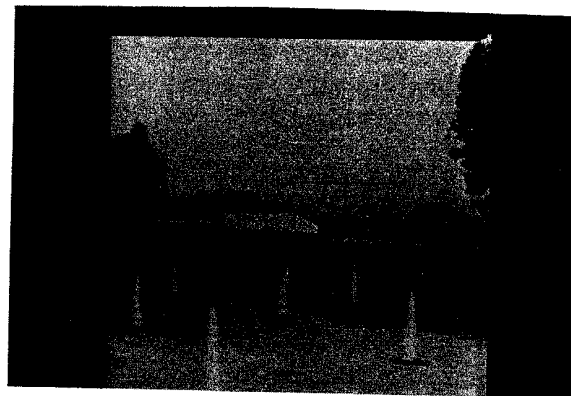
(a)



(b)



(c)



(d)

Figure 96: Outdoor depth-from-motion image sequence



(a)



(b)

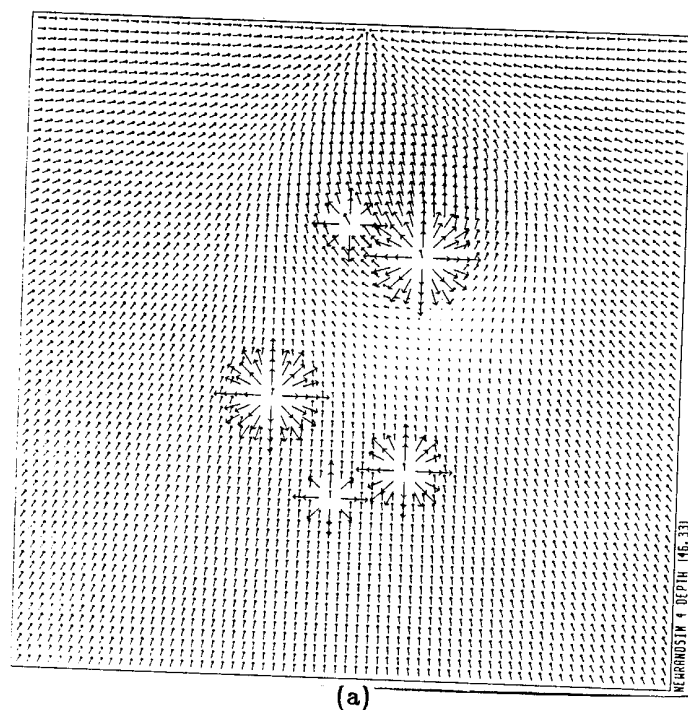
Figure 97: Outdoor depth-from-motion obstacles

(a) The tracked points, associated with each cone top through all four frames, are shown superimposed on the first frame.

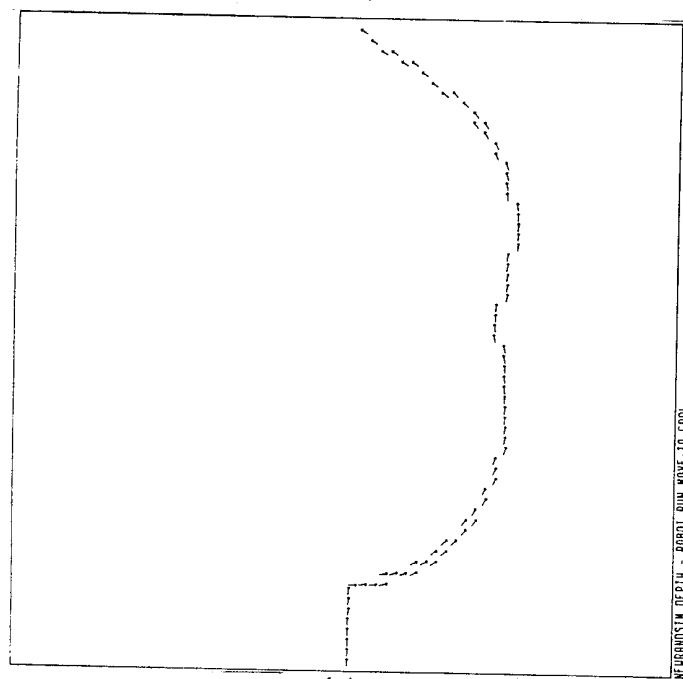
(b) The results of the obstacle extraction algorithm (thresholded at 50% of the points above the least squares-line fit - see Chapter 6). The three closest cones are successfully detected as obstacles.

Table 4: Outdoor depth-from-motion results

object	feature	nominal depth			true depth (feet)		
		1-2	2-3	3-4	$frame_2$	$frame_3$	$frame_4$
cone1	0	33.1	64.4	65.6	35.1	33.2	31.3
	1	30.3	55.3	66.2	35.1	33.2	31.3
	4	36.5	31.1	38.4	35.1	33.2	31.3
	10	36.3	38.0	33.0	35.1	33.2	31.3
	18	33.7	—	—	35.1	33.2	31.3
cone2	9	37.1	41.7	39.8	45.1	43.2	41.3
	11	38.8	46.7	39.7	45.1	43.2	41.3
	15	46.2	51.3	49.3	45.1	43.2	41.3
cone3	5	47.6	54.6	51.4	55.1	53.2	51.3
	6	51.8	56.1	58.2	55.1	53.2	51.3
	14	61.9	60.8	56.1	55.1	53.2	51.3
cone4	8	90.8	85.7	100.1	65.1	63.2	61.3
	13	83.2	88.6	101.1	65.1	63.2	61.3
cone5	2	73.9	85.3	109.4	75.1	73.2	71.3
	12	88.9	107.5	94.6	75.1	73.2	71.3
cone6	3	107.1	106.9	98.2	85.1	83.2	81.3
	7	113.2	146.4	—	85.1	83.2	81.3
	23	94.0	126.8	92.7	85.1	83.2	81.3



(a)



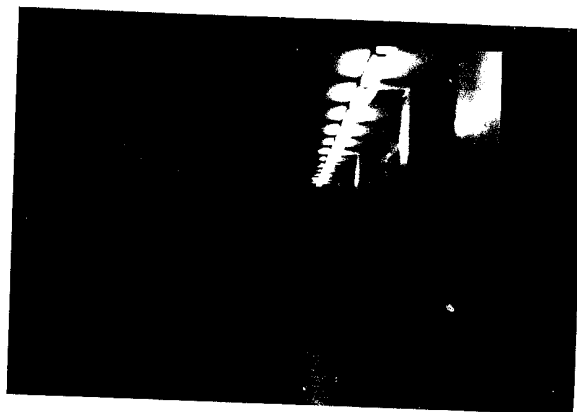
(b)

Figure 98: Depth-from-motion schema simulation

- a) The position of 5 of the tracked obstacles plus a goal produce the velocity field shown here. The robot is located at the bottom of the vector field with the goal at the top.
- b) The robot's course through the obstacle field to the goal.



(a)



(b)



(c)

Figure 99: Indoor depth-from-motion image sequence

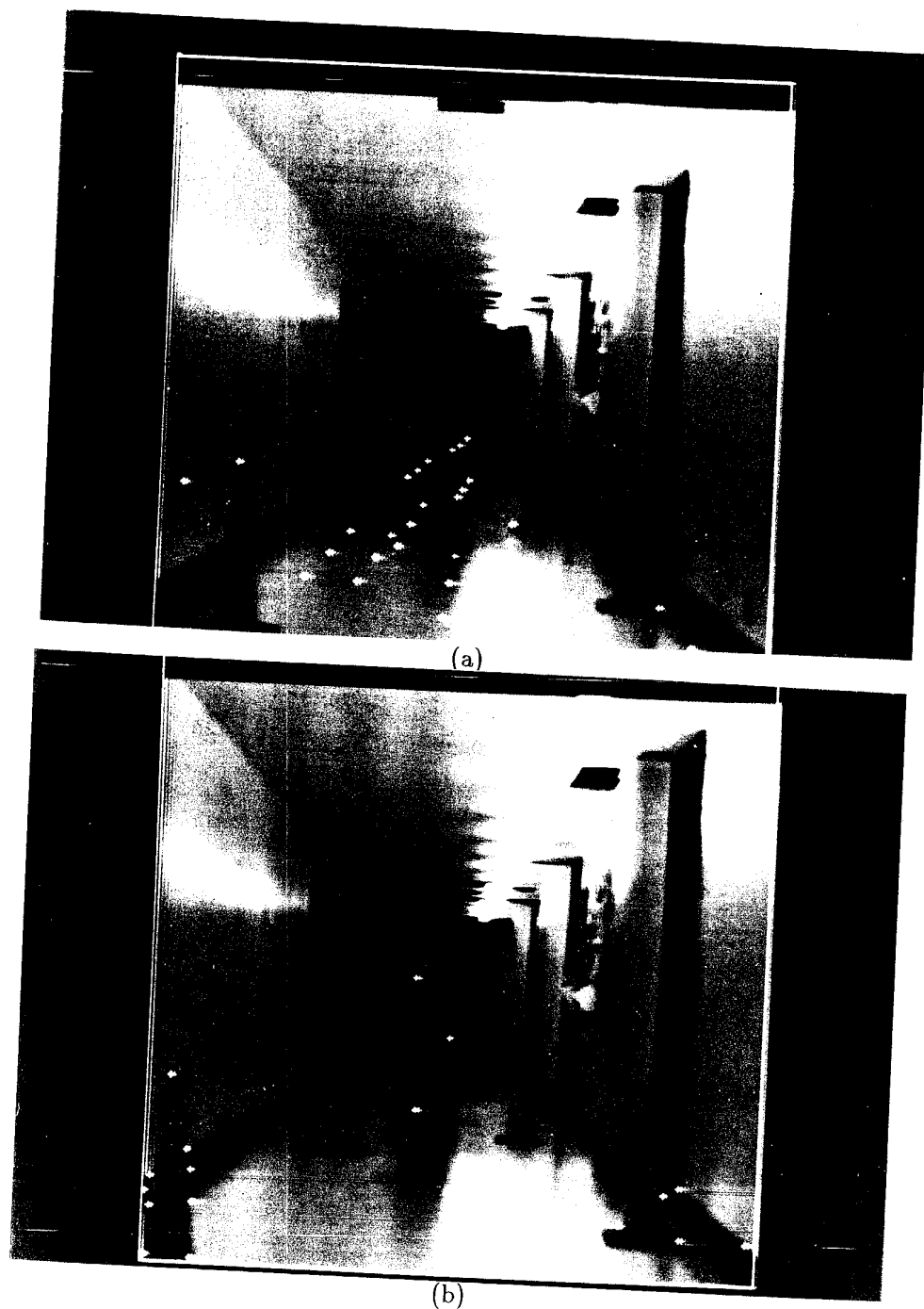


Figure 100: Indoor depth-from-motion obstacles

(a) Tracked points, associated with several obstacles through all three frames, are shown superimposed on the first frame.

(b) The results of the obstacle extraction algorithm (thresholded at the top 50% of the points above the least squares line fit)). The two closest cones, the closest can, an additional cone, and the door were all detected as obstacles.

Table 5: Indoor depth-from-motion results

object	feature	nominal depth		true depth (feet)	
		1-3	3-5	$frame_3$	$frame_5$
cone1	18	25.1	—	16.1	12.2
	42	23.0	—	16.1	12.2
	45	23.2	18.7	16.1	12.2
	55	23.5	—	16.1	12.2
	68	22.4	24.4	16.1	12.2
	71	22.2	—	16.1	12.2
	72	22.3	—	16.1	12.2
	91	22.5	25.8	16.1	12.2
	92	22.6	33.3	16.1	12.2
cone2	11	26.4	36.5	21.1	17.2
	15	24.3	25.9	21.1	17.2
	16	28.7	29.1	21.1	17.2
	46	26.4	23.0	21.1	17.2
	50	27.1	26.3	21.1	17.2
	63	28.0	28.4	21.1	17.2
cone3	41	49.9	44.5	31.1	27.2
	44	39.8	38.8	31.1	27.2
	47	41.5	40.2	31.1	27.2
cone4	13	51.1	49.2	36.1	32.2
	43	50.6	54.3	36.1	32.2
cone5	33	57.5	82.3	41.1	37.2
	51	71.7	85.5	41.1	37.2
	58	57.1	61.3	41.1	37.2
cone6	60	64.6	84.4	56.1	52.2
	65	66.1	84.2	56.1	52.2
can1	3	38.476	34.34	26.1	22.2
	8	40.827	38.9	26.1	22.2
	9	38.171	34.5	26.1	22.2
can2	19	54.2	63.9	51.1	47.2
	32	52.0	62.1	51.1	47.2
	52	54.0	66.5	51.1	47.2

analysis of later frames in a sequence. However, the tracking of the object features themselves is quite robust and leaves some basis for optimism in further development of the algorithm (Figs. 97 and 100). The causes for the numeric error become apparent on inspection of the tracked points. For example, note the two leftmost cones in Fig. 97a. Curvature of the point tracks can clearly be seen. This is attributed to rotation in the image plane, a known problem that can produce errors such as those above.

For the indoor sequence (Fig. 100) the images were not manually registered as the floor was level, smooth, and clean. Drift of the vehicle and eccentricities in its drivetrain are sufficient in themselves to produce registration errors nonetheless. Hand registration was impractical since there are no truly distant points to use as invariants from frame to frame, and only those points extremely close to the FOE can be assumed not to undergo displacement as motion proceeds (these are the points needed to register the images accurately). However, the FOE cannot be accurately found unless the images are registered and there is no easily extracted set of points available from this monocular image sequence with which to carry out the registration process. The most distant trackable points were clustered at the end of the hall and would not work well with the software registration algorithm described in [106].

Although the numeric results were not as accurate as we would have liked to see, many of the obstacles were still extractable. In the outdoor sequence (Fig. 97b) the tips of the three closest cones were detected as obstacles. For the indoor run (Fig. 100b) the two closest cones, the nearest can, one further cone, and the door were extracted. Although the numeric values for depth returned were poor (indoor case), the obstacle detection algorithm worked well enough to extract the closer obstacles. The depth values themselves are not usable in this current form. Partial blame can be assigned to misregistration. The results obtained for the example in Chapter 6 were much better.

An interesting observation is in the tracking of points formed by occlusions of cones with the path edge or the road edge. The point trackings are generally well behaved but their image displacements are not dependent entirely on either the depth to cone or to the road. An occlusion of the edge of a nearby cone with a nearly vertical and distant line in the image could produce large pixel displacements for relatively small vehicle translations due to the slope of the cone's side, giving rise to substantial errors in depth. Fortunately, most of the points tracked in these examples did not exhibit this type of behavior.

What is of the most significance for this algorithm is that although the depths recovered were not always as accurate as hoped for, the tracking of the feature points was in general reliable (Figs. 97a and 100a). By improving on our FOE detection and image registration methods, particularly by introducing a stabilized platform for image acquisition, the accuracy in the depth recovery is predicted to increase dramatically. Research on these issues is continuing at the University of Massachusetts for application to HARV. This collection of algorithms is being refined and will be sent to Carnegie-Mellon University for use on their NAVLAB vehicle. There we expect to be able to obtain images that have additional hardware support for solving the registration difficulties described above.

§4. Localization and uncertainty management

Localization, correlating the robot's position with the long-term memory representation, was tested by placing the robot in a known outdoor location and using the information available in the environmental model, in conjunction with the terrain measures and distance traveled, to control the spatial uncertainty map. The motion of the robot through its world map is depicted in Figure 101.

The robot started at a position known to an accuracy of ± 1 foot relative to the landmarks. This was probably an overestimate of the positional accuracy. Four images were captured (Fig. 102), first from the robot's starting point and then as it moved 10 feet, then 20 feet and finally 40 feet over the sidewalk. Figure 103 shows how the spatial uncertainty map changes as the robot moves from position to position. In this particular case the orientational uncertainty is reduced appreciably after each landmark recognition.

Three landmarks (a telephone pole, a concrete lamppost, and the corner of the East Engineering building) were used. All were treated as Class I landmarks. The building corner initially was treated as a Class II landmark, but it consistently appeared lower in the image than it was predicted to. On further analysis, this was found to be a consequence of the failure of the ground plane assumption in two ways. First, the corner's height from the ground as determined from architectural drawings did not indicate to us that the building's foundation was about 10 feet lower than the ground plane of the path that HARV was on. After compensation was made for this (only an approximation),

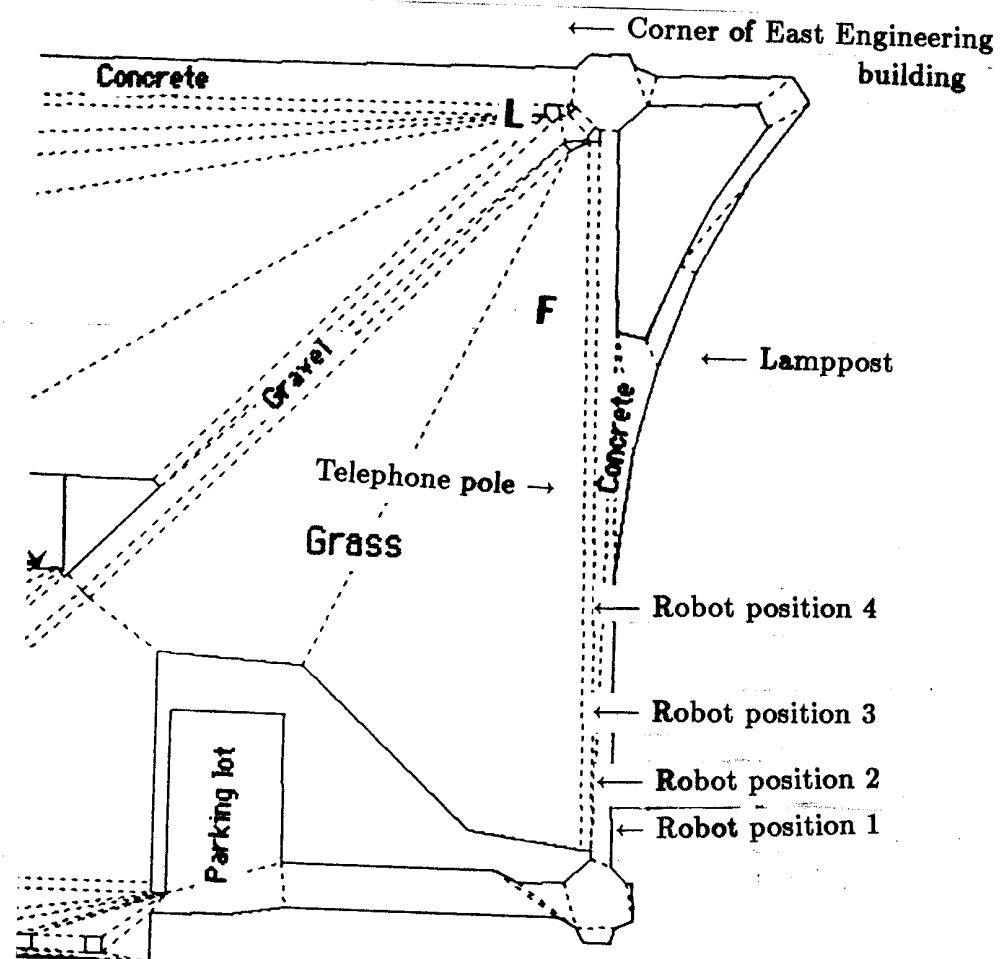
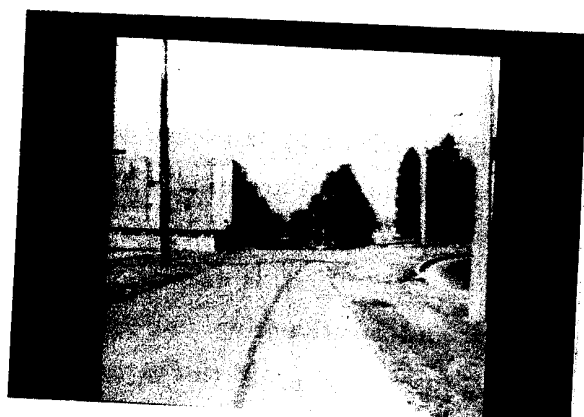
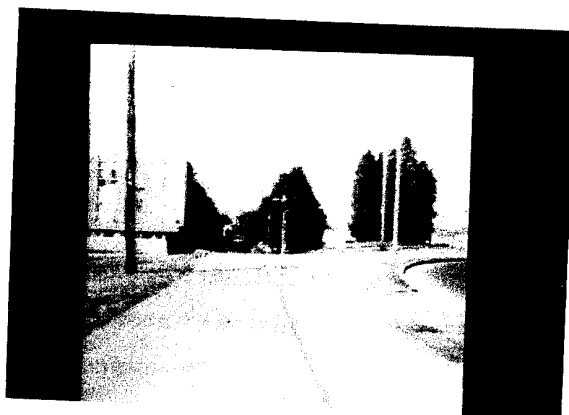


Figure 101: Localization experiment - Robot's position in LTM



Position 1



Position 2



Position 3



Position 4

Figure 102: Localization experiment image sequence

Figure 103: Localization experiment spatial uncertainty map

Here the spatial uncertainty map is pruned via landmark recognition. Most of the improvement occurs in orientation uncertainty which is reduced considerably. The left hand column shows the map before recognition and pruning. The right hand column shows the reduction accomplished by landmark recognition. The concrete pole landmark is used to reduce the uncertainty for maps (a) and (e), the telephone pole is used to reduce the uncertainty in map (c), and the corner of the East Engineering building is used for map (g).

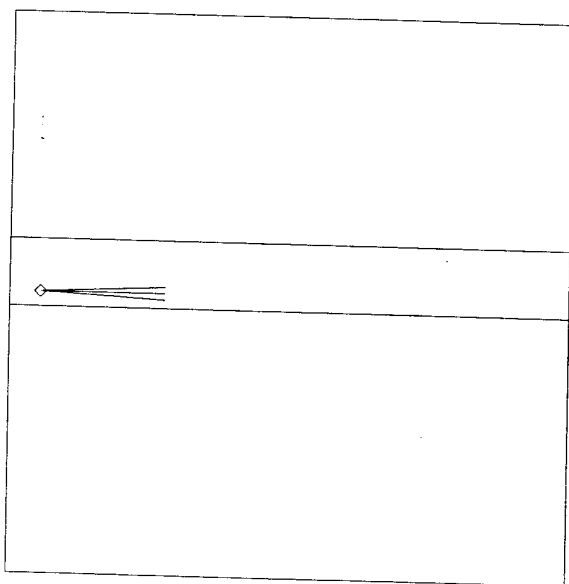
(a-b) Position 1, before and after pruning.

(c-d) Position 2.

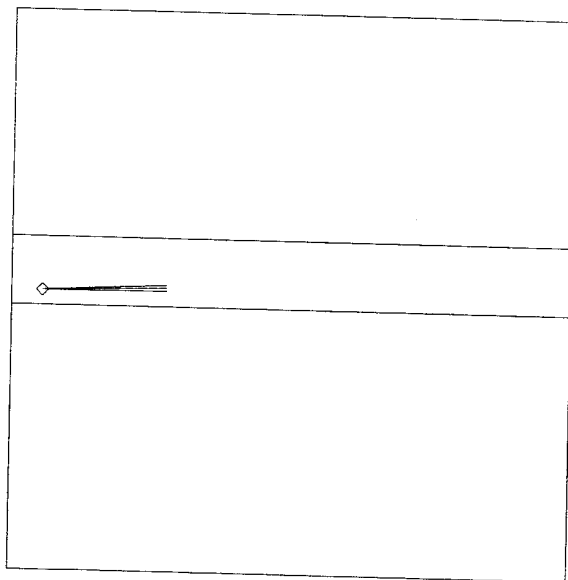
(e-f) Position 3.

(g-h) Position 4.

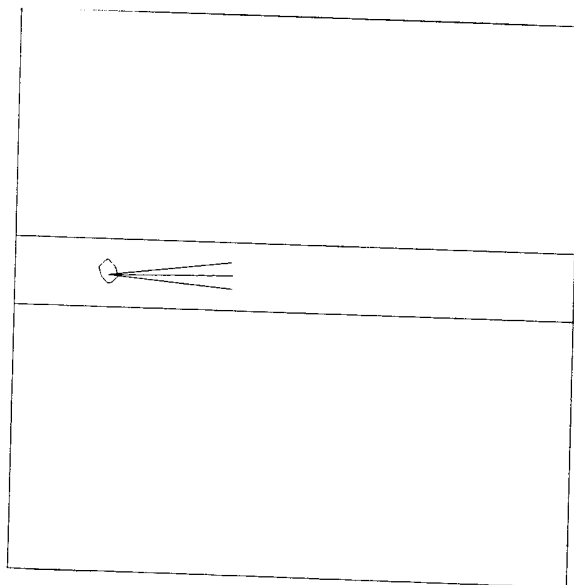
(Figure continued on following page).



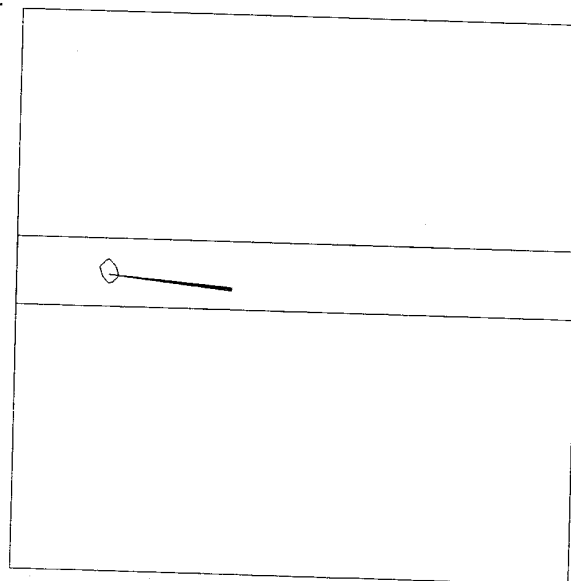
(a)



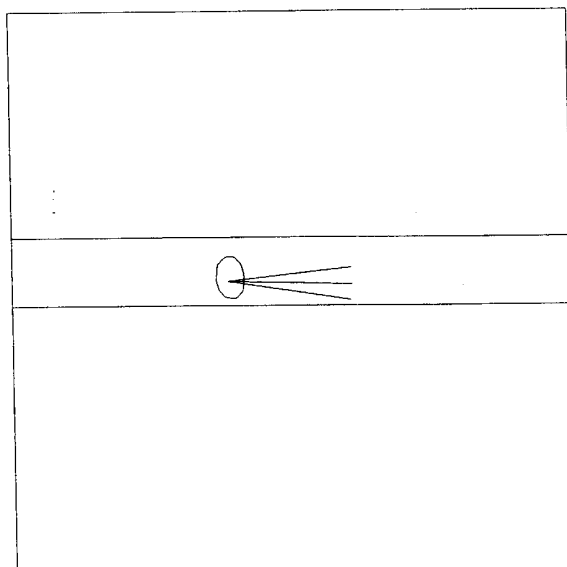
(b)



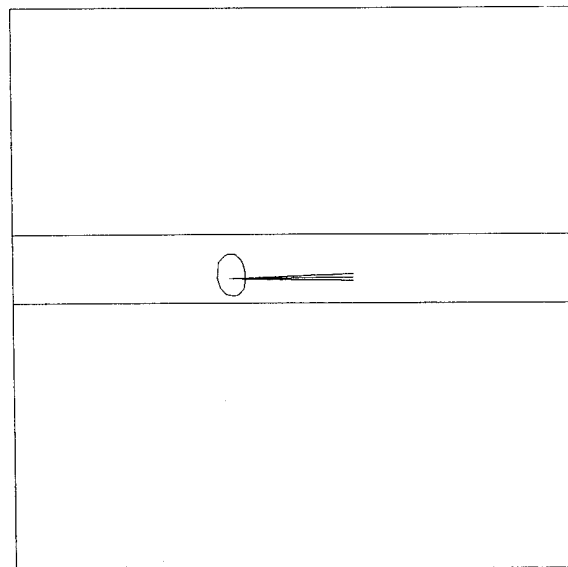
(c)



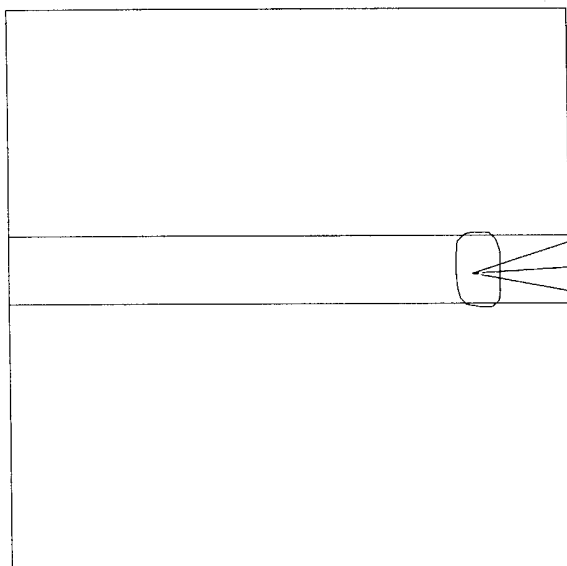
(d)



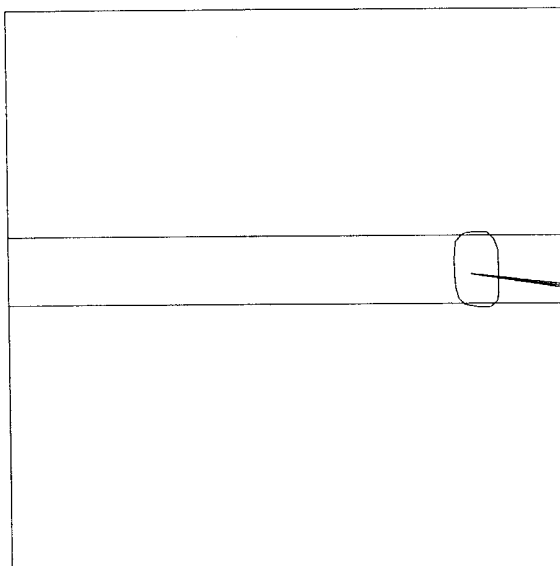
(e)



(f)



(g)



(h)

Figure 103 continued.

the pitch of the vehicle (due to the crowning of the sidewalk surface) still distorted the landmark's position in the image. By this time, without any inclinometer data or a stabilized platform, we had lost confidence in the ability to directly extract the depth of this feature. The roll was minimal (by observation) so we used this feature as a Class I landmark for the rest of the experiment.

Expectation windows were produced and used to restrict the search for the particular landmark being sought. If no pruning was accomplished the expectation windows would be larger than if pruning was undertaken. Figure 104 shows the size of the expectation windows for three landmarks when pruning is performed. The production of these windows involved the use of the camera calibration matrix described in Chapter 6. Although the line finder was run on these windows (Fig. 105) and produced identifiable structures related to the landmarks, the actual image position of the landmark was determined manually. After identification occurred, the landmark image coordinates were input along with its world coordinates to automatically reduce the spatial uncertainty. Only orientation uncertainty reduction was in evidence as the uncertainty in the world coordinates of the landmark and the error in the calibration matrix permitted only coarse adjustments. By *surveying* the three-dimensional world accurately, and using inclinometers or inertial navigation to give an estimate of the roll, pitch, and yaw of the camera relative to the world, greater control of the uncertainty could be produced.

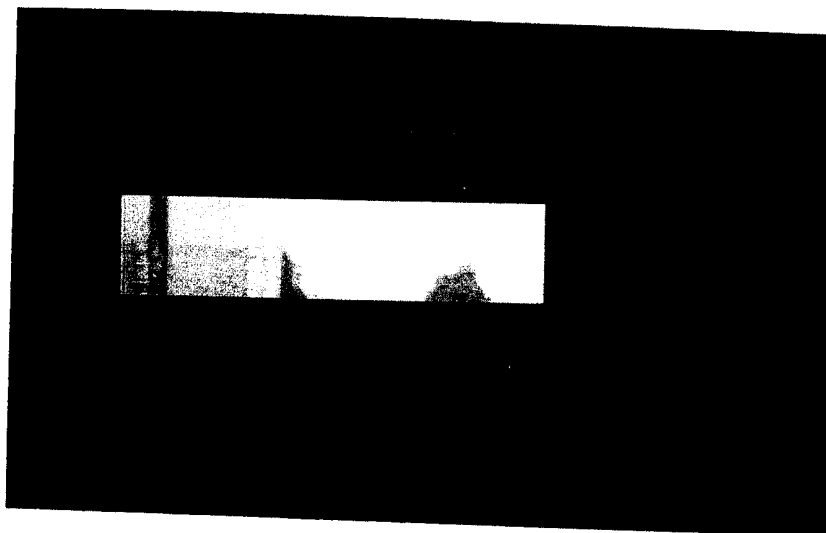
§5. Multi-component test

The multi-component test combines motor schema based navigation, ultrasonic obstacle avoidance, path following using vision, and STM building. The navigation was performed on-line, while the short-term memory representation was generated off-line. The STM context is shown for both levels: the representation for the instantiated meadows (Fig. 112) and the sonar-filled regular grid (Fig. 110).

This test initially involved navigation using the fast-line finder, with the robot moving towards the center of the walk. Figure 106 shows the robot's course. While performing line finding, the robot monitored its heading for obstacles using ultrasound. When a blockage was detected as evidenced by at least one of the three leading ultrasonic sensors returning a value below a specified threshold (3.5 ft), the robot switched over automat-



(a)



(b)

Figure 104: Localization experiment expectation windows

These are the windows produced for the image taken at position 3 by the spatial uncertainty map incorporating the reductions in orientation uncertainty as the robot moves.

- a) Window for telephone pole and the concrete lamppost.
- b) Window for building corner.

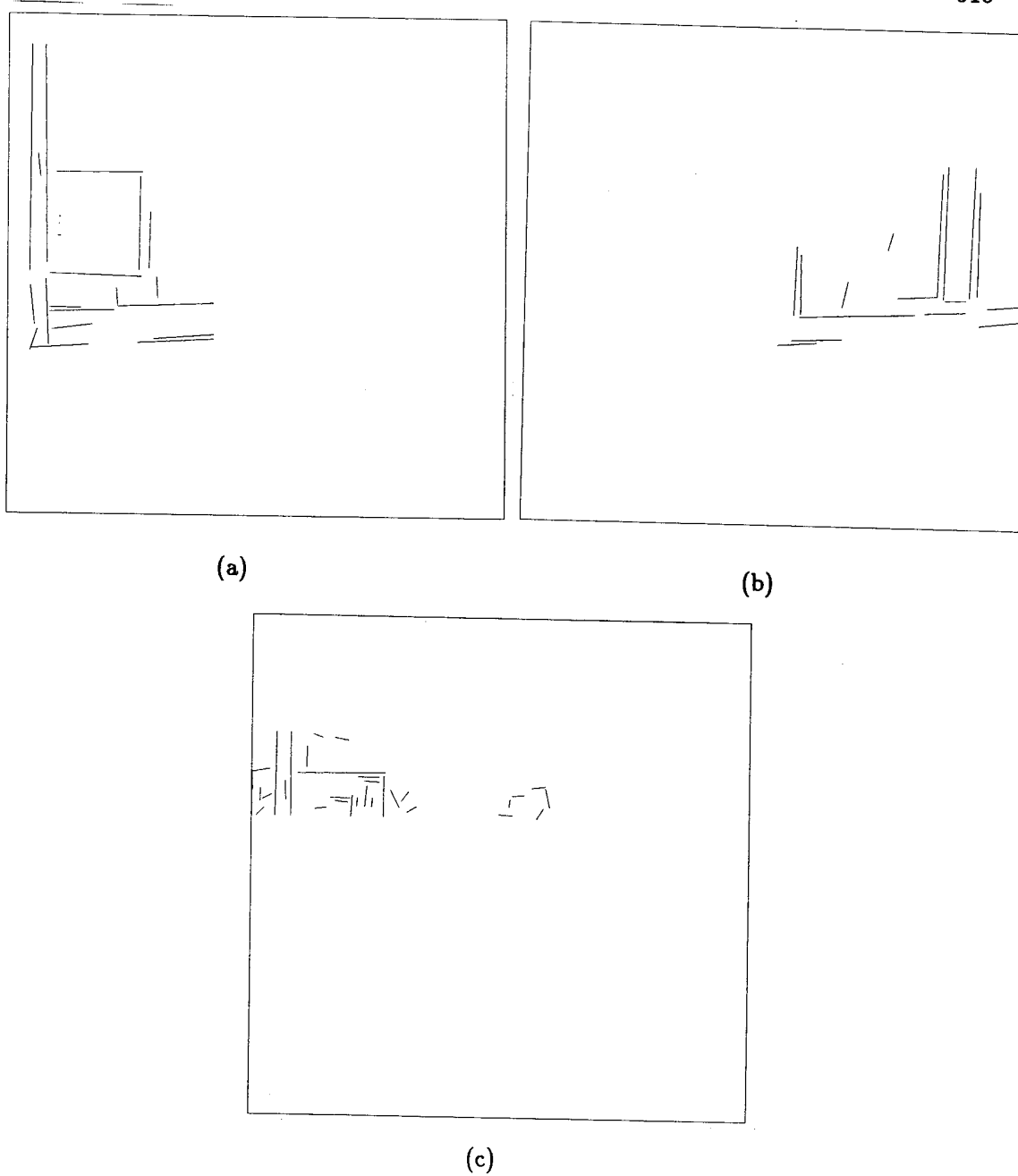


Figure 105: Line finder results on landmark windows

These are the results for the image taken from position 3.

a) Results for telephone pole.

b) Results for concrete lamppost.

c) Results for building corner.

ically to schema-based navigation, using **move-ahead** (encoder-based), **stay-on-path** (encoder-based), **avoid-static-obstacle** (sonar-based), and **noise** SIs in the framework of the schema-based navigational testbed (described in Section 1) to negotiate the course. The obstacles were successfully circumnavigated, at which time the robot reverted to line-finding navigation. The run went extremely well, with the exception of the robot's desire to strangle itself on its camera tether. The cable tether, due to the frequent rotations during the obstacle avoidance phase, became entangled around the robot's upper (rotating) platform and the lower (stationary) body. When the UHF transmitter becomes operational this will no longer pose a problem. Aside from this difficulty, the run went smoothly.

Figures 107-109 show the fast line finding component in action. It is interesting to note that the line being extracted on the right side is actually a change in the surface type of concrete rather than a separate path. This surface feature is distinctive enough for the line finder to operate on.

Figure 110 shows the short-term memory representation built up from the sonar data. The robot built this model as it passed amongst the middle of the cone area. The scale of this grid is 32 feet by 32 feet. Whitespace denotes areas highly probably to be unoccupied while occupied area probability is denoted by numeric values. This information was not needed for this run as the motor schema based navigation system handled the obstacles without any difficulty. As mentioned in Chapters 4 and 5, the reliance on STM should be relatively rare and is only needed when the potential field methodology fails.

Figure 111 shows the planned course through LTM. The information in this map is used by the STM meadow instantiator process to extract only those meadows relevant for this particular navigational leg. Figure 112 shows the instantiated meadow component of STM for this run.

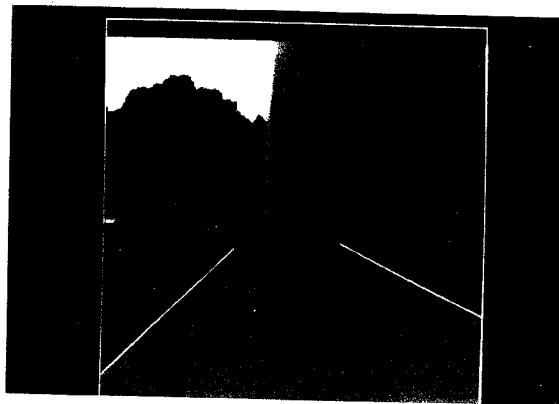
§6. Summary

HARV's experiments have given a practical demonstration of many of the concepts developed in earlier chapters. As expected, some experiments were more successful than others. Good results were obtained with motor schema-based navigation, the path following using the fast line finder and the multi-component experiment. The fast region

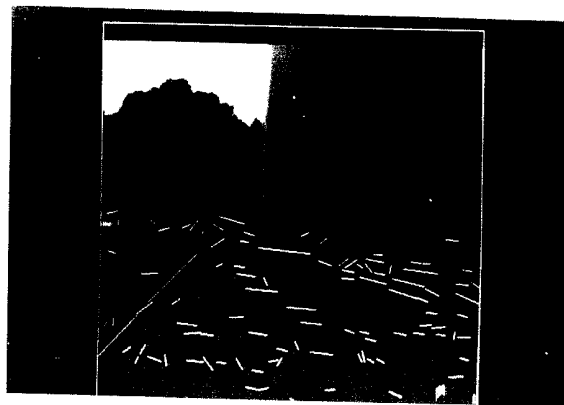
Figure 106: Multi-component test run

The robot initially moves under control of the FLF path following algorithm. When it detects the obstacle in frame 3 using ultrasound, it switches to schema-based navigation, using the shaft encoders to enforce the stay-on-path schema. After successful completion of the obstacle course, it reverts to FLF path following.

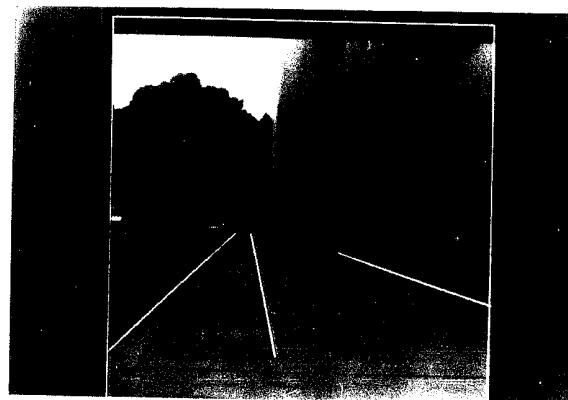




(a)



(b)



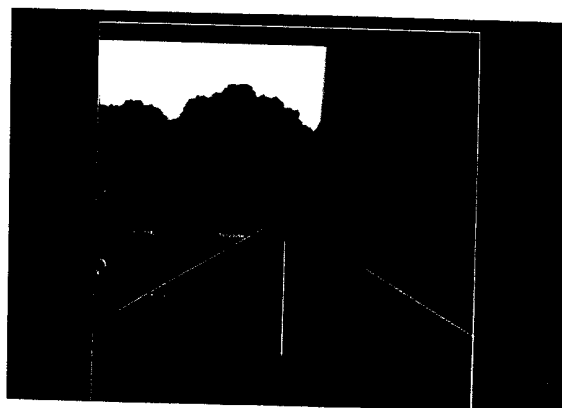
(c)

Figure 107: Multi-component fast line-finding (first image)

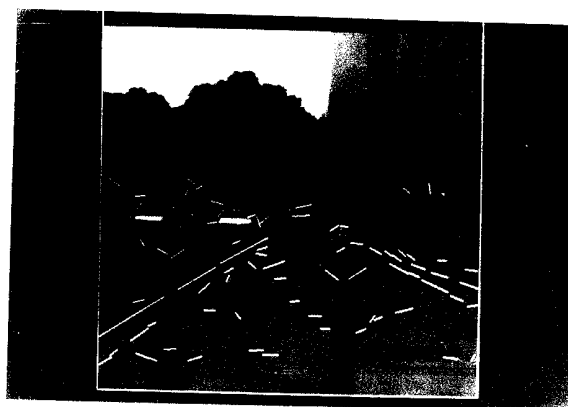
a) The bootstrap lines used to tune the buckets for the image. Note there is a significant discrepancy between the expectations and the actual image, but the lines are found anyway, indicating the robustness of this approach.

b) Extracted lines using orientation tuned buckets.

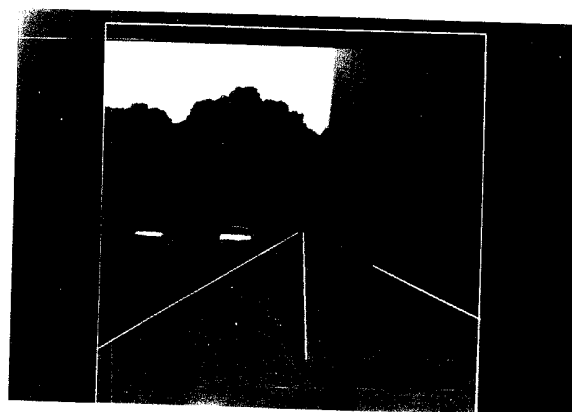
c) Resulting road edges and computed centerline for servoing.



(a)



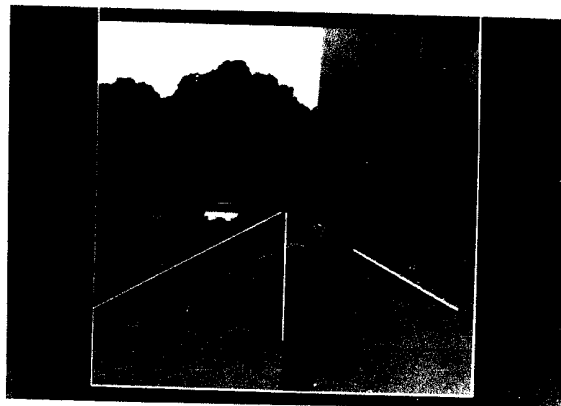
(b)



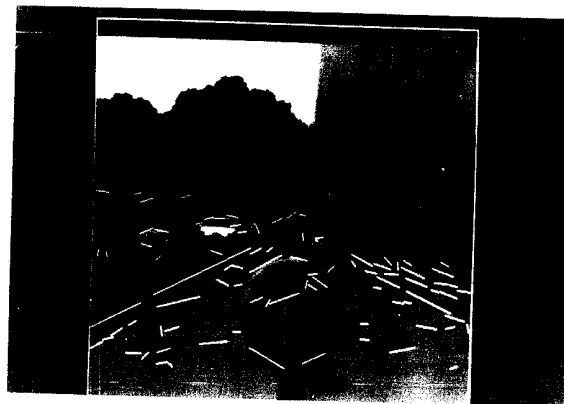
(c)

Figure 108: Multi-component fast line-finding (second image)

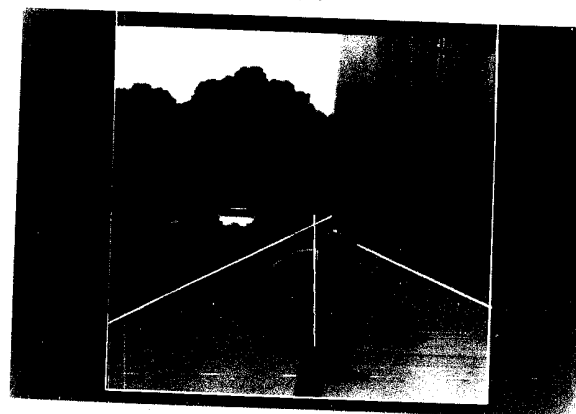
- a) The feedforward lines produced from the first image that are used to tune the buckets for the image. Note again the discrepancy between the expectations and the actual image events.
- b) Extracted lines using orientation tuned buckets.
- c) Resulting road edges and computed centerline for servoing.



(a)



(b)



(c)

Figure 109: Multi-component fast line-finding (third image)
a) The feedforward lines used to tune the buckets for the image.
b) Extracted lines using orientation tuned buckets.
c) Resulting road edges and computed centerline for servoing.

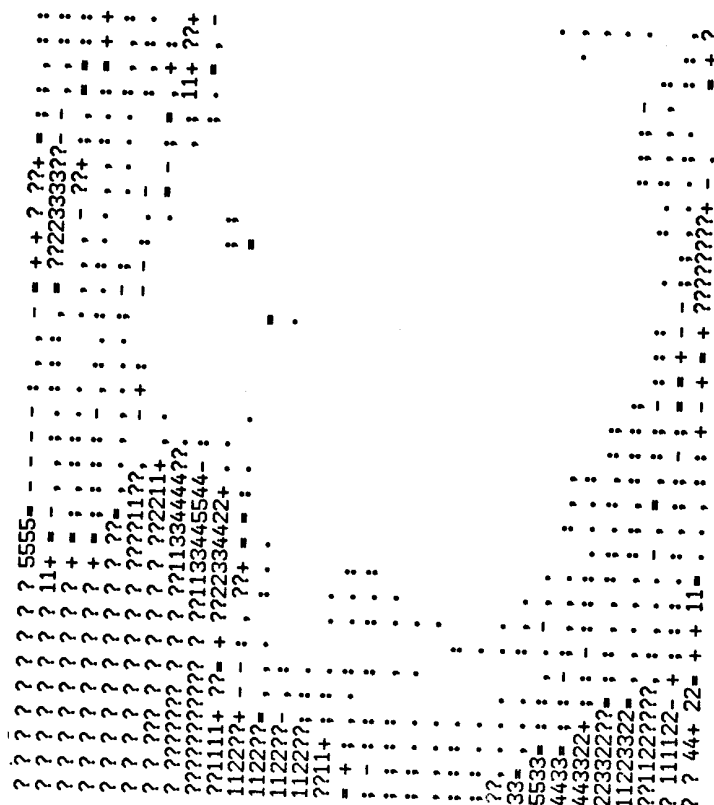


Figure 110: Multi-component STM building

A modified version of the Moravec-Elves code [93] is used to construct the top-level grid representation for STM (Chapter 4). Here the robot is in the free space amidst the obstacles (the view is from above). The clusters of numbers represent positions likely to be occupied by an obstacle (i.e. cone) while whitespace denotes areas likely to be unoccupied. (The scale of the grid is 32 feet by 32 feet).

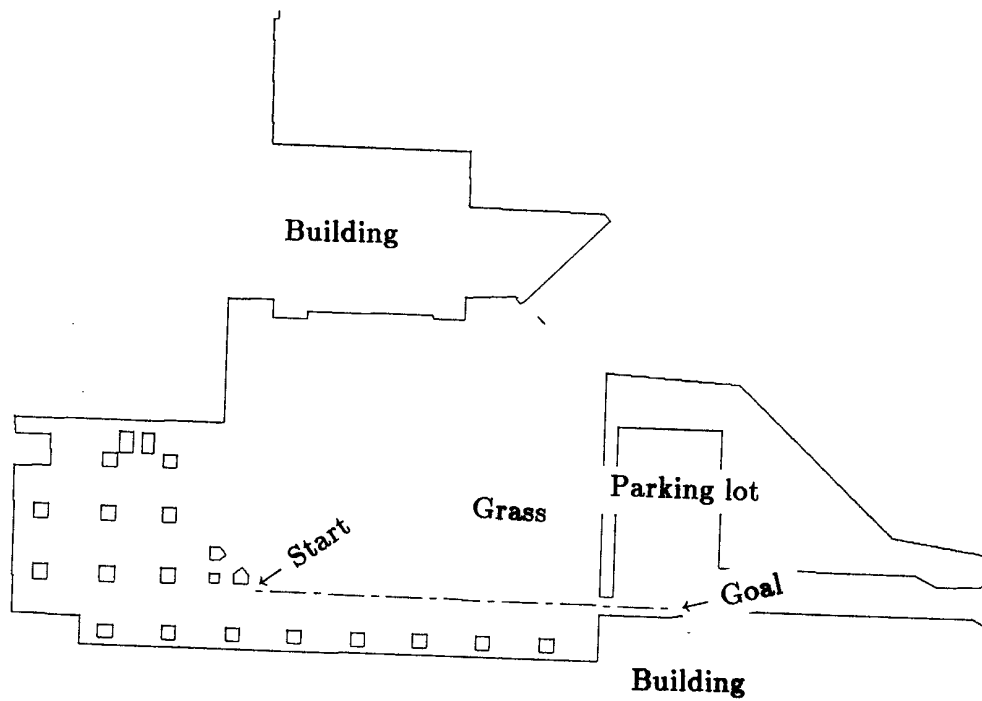


Figure 111: Multi-component LTM path

The robot's path takes it along the sidewalk past the GRC's concrete pillars. Note the unmodeled obstacles are *not* present during the path planning process.

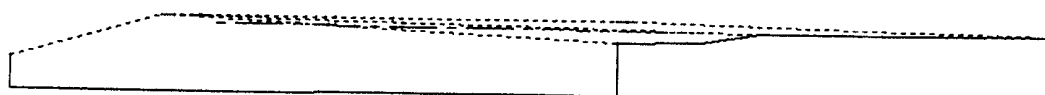


Figure 112: Multi-component STM instantiated meadows

These are the meadows selected by the cartographer to provide the context for STM (base level) during the robot's journey.

segmenter and depth-from-motion system produced less than perfect results. In all cases, the experiments yielded valuable insights into the problems of navigation in relatively unconstrained environments.

Schema-based navigation using ultrasound showed a high degree of reliability. Combinations of the primitive motor behaviors manifested the predicted more complex behaviors. When the vision algorithms become faster, it would be desirable to embed them as well as perceptual algorithms in this experimental testbed, especially for goal recognition and path following.

The vision algorithms were tested modularly and in one instance, in the multi-component test, combined with other navigational components of AuRA. The fast line finding system behaved particularly well for path following in both indoor and outdoor situations under a variety of environmental conditions. This algorithm will continue to be tested under more difficult conditions such as roads and paths streaked with shadows.

The fast region segmentation path following system is less robust, but this is expected. Using gradient magnitude for outdoor scenes only using intensity images limits severely what can be accomplished. Nonetheless, successful path following was accomplished using this algorithm. Color image acquisition is essential, in our estimation, to improve the reliability of this algorithm when applied to path finding.

The depth-from-motion algorithm was a mixed bag, yielding both successes and disappointments. The successes arose from the ability to track points reliably from frame-to-frame in a very high percentage of the cases. The disappointments came from the inability to produce highly accurate numerical depths. The reasons for this are believed to be well understood: misregistration and inaccurate focus-of-expansion recovery. These problems, although difficult, are not considered insurmountable and continuing research is being applied to the depth-from-motion system to obtain greater accuracy.

The role of spatial uncertainty management in AuRA was illustrated through the localization experiment. The Expecter, using camera calibration data, is used to predict where in an image a particular landmark will occur. Although this experiment used manual landmark discovery, the results of the fast line finder on these windows showed automation of this process is possible. Ultimately, landmark discovery will be relegated to the VISIONS system.

An interesting problem, especially for point landmarks (Class II - see Chapter 7),

occurs when the ground-plane assumption fails. This occurred in two ways. First, during the initial knowledge acquisition phase, it was insufficient to know the height of the building or other landmark feature above the ground. What was required was the height of the ground feature above the ground-plane the robot was currently located on. The LTM landmark representation must incorporate height relative to some global plane (e.g. sea level) if the robot is not working on a truly planar area. Secondly, it is important that the robot's pitch and roll relative to the ground plane be known. Inclometers, currently unavailable on HARV, can provide this information. If the pitch or roll is non-zero relative to the ground plane, the vehicle is actually on a different plane relative to the LTM ground plane, and compensation must be made in the predictions issued by the Expecter as to where the landmark will appear in an incoming image. If the robot is pitching upwards towards the front, a landmark will appear lower in the image than otherwise expected. To further worsen matters, when the recognized landmarks's position is backprojected into LTM it will then appear farther away than it actually is producing errors in the spatial uncertainty map. As sidewalks and roads are typically crowned for drainage reasons this can pose real difficulties in obtaining accurate results.

The multi-component test demonstrates the relationships between many of the individual components of the AuRA architecture, including LTM, STM, navigational planning and schema-based navigation. This experiment went quite smoothly with all the different components integrating quite well. Future enhancements would involve the conversion of the visual FLF path following component to a schema and its integration into the motor schema testbed. This would avoid the somewhat artificial changeover from visual path following to ultrasonic and encoder based navigation and then back to vision again.

Many lessons were learned from these experiments. One of the most important is the value of performing the experiments themselves and not relying too heavily on simulations. In some cases the move from simulation to experiment was smooth as in the schema-based navigation system. In others, such as the depth-from-motion system, whose simulations worked well on synthetic images [21], many types of unanticipated difficulties appeared, such as misregistration and FOE changes. All in all, however, considerable progress has been made in having a mobile robot explore both indoor and outdoor environments given a partial model of the world.