CHAPTER VII

SPATIAL UNCERTAINTY MANAGEMENT

Mobile robots have difficulties that are not found in more conventional robot systems. Robotic manipulators, through the use of inverse kinematics, and the fact that their position relative to the workspace is typically known to a high degree of accuracy (on the order of fractions of millimeters) using high resolution encoders, can in many cases ignore the uncertainty in the robot's position itself. That is not to say that uncertainty is a solved problem for this domain; quite the contrary. Most uncertainty in assembly-oriented tasks arises from the relationship of the manipulated parts to the modeled world rather than that of the robot to the world.

Automatic guided vehicles, such as wire or stripe following robots, have more uncertainty to contend with, but the problem is essentially one-dimensional. As the robot must maintain adherence to a line, the only uncertainty is in where the robot currently is located along that line. By embedding optical landmarks along the path or through the use of infrared beacons, the robot's world position can be readily confirmed.

Mobile robots are plagued with uncertainty problems. Uneven traction due to the terrain or tire inflation, and drift due to problems within the drive train can rapidly lead to disorientation of the robot relative to its modeled world. The mobile robot must contend with a minimum of three degrees of freedom (with significant uncertainty in each) – 2 degrees of translation, which can be represented as x and y coordinates in a Cartesian world model, and 1 degree of rotation (assuming a planar world), the actual heading of the robot relative to known compass headings.

Most mobile robot systems concerned with uncertainty management have handled this problem in the context of environmental acquisition, where the robot's world is not modeled ahead of time, but instead is built from sensor observations, typically sonar. This chapter describes an approach to uncertainty management that uses an explicit representation of the robot's positional and angular uncertainty relative to an *a priori* model of the world. Vision is the principal means for reducing this uncertainty.

The goal of the spatial uncertainty management system is to guide expectations for sensory modules in AuRA, reducing the amount of processing required to determine the robot's whereabouts. The world representation used is the meadow map described in Chapter 4. Embedded within this map are data essential for uncertainty management. This includes relevant terrain data for a statistical approach to uncertainty growth, and visual landmarks which are used to guide perceptual processing for ultimate recognition and resultant reduction of uncertainty.

The remainder of this chapter is divided into the following sections. Section 1 presents a review of related work in the field of mobile robot uncertainty management. Section 2 describes AuRA's cartographic spatial uncertainty management subsystem and its relationship to other components of the AuRA architecture. The specific representations used and the process that manages the growth and reduction of uncertainty, based on the vehicle's motion in the world coupled with visual feedback, is detailed in Section 3. The process of landmark selection is described in Section 4 and that of landmark recognition in Section 5. Simulation results are presented in Section 6, while actual experimentation using the mobile vehicle is presented in Chapter 8. A summary and evaluation of the uncertainty system concludes the chapter.

§1. Related work in uncertainty management

A hallmark paper addressing uncertainty in robotics, more concerned with assembly than with navigation, was written by Brooks [28]. His discussions of visual map making [25,29], where he argues against the use of a global map for a system which acquires (learns) its environmental model solely from vision, are more pertinent to mobile robotics.

A fundamental problem, Brooks states, is that worse case error must be used with an absolute coordinate system. There are significant limitations if the entire world is to be modeled in a single world-oriented frame of reference. AuRA proposes two frames of reference: first, an egocentric model which is the basis for sensor data acquisition; and second, a world model which is used to represent *a priori* knowledge (a partially modeled world). A model based on knowledge of the terrain reduces the dependency on worst case analysis.

Polyhedral models are to be avoided, according to Brooks, due to their poor time-performance and tendency to break down in real world situations (as the real-world is not polyhedral). INRIA's HYPER system [17], described below, argues against this. In AuRA, successful counter-arguments can also be made to this claim by recognizing particular classes of landmarks, subdividing the processing over multiple active schemas, and searching for landmark features in restricted portions of the image. Finally the desirability of vision over less informative sensors such as sonar is shared by both Brooks and myself.

Brooks' system uses, as does AuRA, shaft encoder data, visibility analysis, and visual landmark recognition. His representations include both "freeways" (generalized cylinders), a free-space representation method, and meadows (circles in [29], convex regions in [25]). His treatment of uncertainty involves the generation of 3D solid "uncertainty manifolds" arising from an uncertain transform. Brooks' approach combines ("cascades") these uncertainty manifolds, which arise from sequences of uncertain transformations, to provide information about the robot's current location relative to sensed landmarks. Projections of the uncertainty manifolds are represented in 2-dimensional space as circles that grow as the robot moves when there is no feedback available from landmark recognition. AuRA instead uses a convex polygon representation (circles represent the worst case analysis) due to the asymmetric nature of motion error. A fundamental difference in AuRA's approach lies in the backprojection of the uncertainty into world coordinates (i.e. the construction of the spatial uncertainty map) which is then overlaid on top of the absolute world model (the meadow map) to provide expectations of previously unseen landmarks whose whereabouts are known only in world coordinates.

Chatila and Laumond [33] independently developed an approach called "fading" that is similar to Brooks' method. This technique employs circular approximations for uncertainty and is used in a sensor-acquired (learned) meadow map, closely related in structure to AuRA's; indeed, this work for HILARE provided an incentive for the extended meadow map representation of *a priori* knowledge used for world modeling in AuRA. HILARE is concerned with acquiring its own world model and hence needs to associate a new frame of reference with each newly discovered landmark. AuRA assumes the existence of landmarks in its meadow map, added by the cartographer from available *a priori* knowledge. In HILARE, landmark recognition is used to update the robot's model of the world as much as its own position relative to the perceived world. It is entirely possible that the landmark's position, and not just the robot's, must be updated. Focus of attention mechanisms, as found in AuRA, are not treated either in Brooks' paper or this one.

Smith and Cheeseman [118] have developed a basis for the handling of spatial uncertainty in the mobile robot domain. Drawing on Kalman filter theory, they include methods for merging (combining evidence from independent parallel measurements to improve the certainty over any single measurement) and compounding (chaining sequential uncertainty transformations). Smith and Cheeseman's paper, as in the two cited above [29, 33], handles the transformations over multiple frames of reference, each associated with the vehicle's position at the time of its observation. The goal is to describe landmark observations in terms of previously sensed, but uncertain, landmarks rather than to represent newly acquired data in terms of a world model. The choice of which landmarks to use is guided by the uncertainty in previous landmark recognitions. Although an elegant mathematical technique is developed for this purpose, it relies heavily on fully independent sensings and thus does not derive, in our estimation, the full benefit available from landmark tracking.

Active sensing using infrared beacons placed at strategic locations, (e.g. available for the commercial Denning Sentry and other similar robots) can be used to reduce positional uncertainty. Infrared beacons are also used in HILARE [52].

Fukui [49], in one of the first systems to use passive vision for positional uncertainty management, used a special landmark to position a robot in interior scenes. This early work placed an artificial diamond-shaped landmark of high contrast in locations that favored its detection so as to reduce the spatial uncertainty of the vehicle.

One of the more sophisticated systems developed thus far for maintaining the spatial uncertainty of a mobile vehicle arises from work performed at the University of Maryland [4]. The system is composed of three separate modules. The MATCHER identifies landmarks in an image by using a Hough transform based on an edge template of the landmark in question. The FINDER controls the pan, tilt, and zoom mechanism for the camera based on available spatial uncertainty data. The SELECTOR chooses good landmarks from a database that enables the vehicle to reduce its positional uncertainty.

A circular uncertainty region, called a disk, is used to model the positional uncertainty of the vehicle in global coordinates. These data, in conjunction with the actual structure of the landmark, the angular uncertainty in the vehicle, and the uncertainties in the pan-tilt and zoom mechanisms, are used to constrain the direction and focal length of the camera. Landmarks are actively sought by the system and are not derived as a by-product of other available images. The entire landmark must be present in the image for recognition with the Hough transform.

Although the Hough transform method is fundamentally different from the visual strategies currently used in AuRA (see Chapter 6), the geometrical development of the FINDER and AuRA's Expecter is similar, since both are used to predict where a landmark will occur in an image. In the FINDER, this information is used to mechanically drive the camera to actively seek out the landmark, whereas in the Expecter it is used to provide appropriate subwindows of the image. Maryland's SELECTOR chooses from all of the available landmarks in the database and is not restricted to consider only those lying in one particular field of view. AuRA's Expecter selects only those landmarks that are expected to be encountered in the direction of the robot's current motion. The Maryland system is a triangulation oriented system while AuRA's Expecter can use to advantage information derived from a single landmark. This is largely due to the asymmetry available within AuRA's spatial uncertainty map. Although it is desirable to control the focal length of the camera while searching for specific landmarks, no provision is made for this in the current implementation of AuRA. The fundamental reason for this is that if a single image sequence is to be used for path following, landmark identification, obstacle avoidance, and other tasks, it is not feasible to optimize the image for any single perceptual schema. By judicious selection of landmarks, multiple tasks (including multiple landmark recognition) can proceed concurrently without altering the orientation or focal length of the camera.

A final distinction between the Maryland system and AuRA arises from the source of uncertainty. AuRA's empirical approach for terrain modeling serves as the basis for the growth of the spatial uncertainty map. Maryland's system appears to be largely based on previous identifications of landmarks alone to constrain the positional uncertainty. Convex regions, similar to the spatial uncertainty map, are used in the development of the spatial uncertainty, but they arise from the triangulation effects of multiple landmarks. This polygon is then approximated by a circle for later use. A convex polygonal representation is also retained for AuRA's representation of spatial uncertainty. Additionally a point center within the convex region is maintained by AuRA to indicate the likeliest location of the robot.

Work performed at INRIA in the development of the HYPER system [17], used to guide a robot arm to pick up occluded or poorly illuminated parts, develops important ideas for extension to the mobile robot domain. The use of polygonal representations for part (in our case landmark) recognition as well as scene modeling is stated to offer several advantages. These include:

- local information (in contrast to conventional robot vision measures that use global numerical features)
- Low storage requirements (compact)
- A general method is available for diverse parts (landmarks)
- Position and orientation sensitive (hence their recovery is feasible)
- Simple and fast vision operations

HYPER's success in part recognition provides a justification for polygonal landmark representation. The most important contribution of HYPER, however, is related to the rating of potential matches through a hypothesis evaluation mechanism using a quality metric.

Work at Yale [37,82] regarding the representation of spatial uncertainty in SPAM (spatial module) is of interest. McDermott and Davis represent spatial uncertainty with fuzziness, and particular locations of environmental objects relative to each other with fuzzboxes. The reduction of uncertainty is termed fuzz constriction. Although their work has been directed toward route planning, several of the concepts, including fuzzboxes (generally rectangular but allowed to have other shapes as well) to represent spatial uncertainty, appear generalizable. The other proclaimed strength is the support of multiple frames of reference through *frobs*, a hybrid representation for an object in a perceived frame of reference. SPAM does not specifically address the uncertainty associated with visual landmark recognition, but the representations and methodology it uses seem extendible.

Specific systems dealing with the uncertainty associated with particular visual strategies are also worth mentioning. At CMU [80,81], a stereo based system has been employed for visual guidance of robot motion. The triangulation uncertainty method constrains the position of the vehicle, using a Kalman filter approach (similar to Smith and Cheeseman above). Two models are maintained; a local, moving, robot-centered frame and a global coordinate system. Correlation between the perceived landmarks in the local frame and the known position landmarks in the global frame constitute the task.

Uncertainty analysis for depth-from-motion image sequences has been studied by Snyder [119] at the University of Massachusetts, and is being applied in AuRA (Chapters 6 and 8). Limitations on the image displacements anticipated from frame to frame constrain the search space for match points and thus expedite the depth extraction process.

§2. Uncertainty subsystem - An overview

The bulk of the spatial uncertainty management subsystem (UMS) lies within the cartographer's responsibility in the overall AuRA architecture (Fig. 2). A block diagram of the UMS appears in Figure 66. The UMS is tied to other components of AuRA through the clipboards, vehicle interface, navigator, pilot and motor schema manager.

The UMS consists of both data structures (rectangles in Fig. 66) and processes (rounded rectangles). The relevant data structures represented include the spatial uncertainty map itself, an identified landmark buffer, data from long-term memory (LTM) including terrain characteristics and landmarks, the schema database, specific clipboard data containing positional reports, and the command buffer within the vehicle interface. The processes include the uncertainty map manager, the components of the pilot concerned with **find-landmark** schema instantiation, and the Expecter that is used to predict landmark position in incoming images.

The overall flow of control within UMS can be described as follows. The pilot first receives information that the robot is to traverse a specific leg of the overall global path developed by the navigator (see Chapter 4). Available within the cartographer is an express representation (the spatial uncertainty map) of the robot's current position (bootstrapped at start-up or known from previous legs) including both the uncertainty in heading and spatial location relative to the global meadow map. Available in STM,

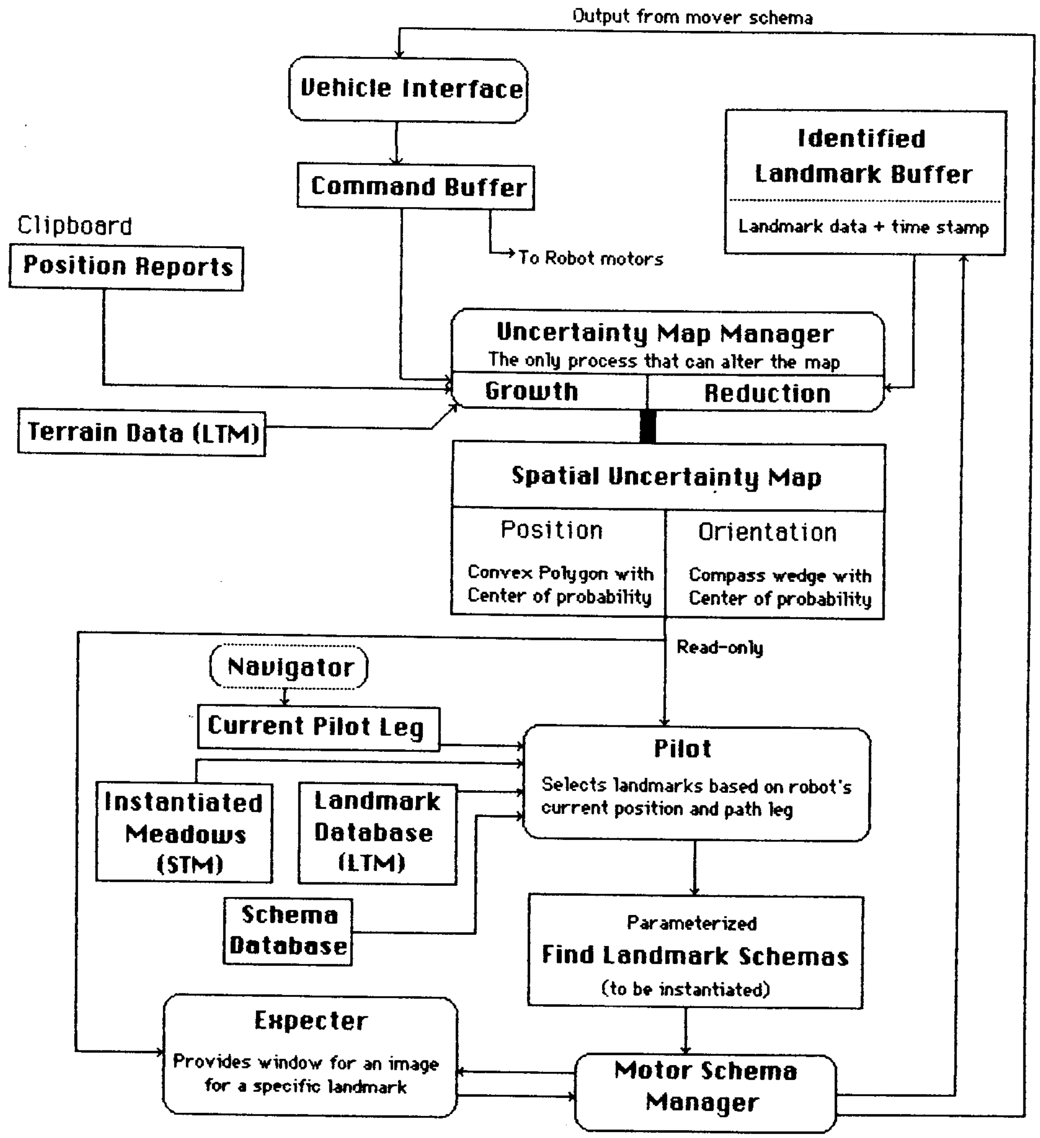


Figure 66: Uncertainty Management Subsystem

provided by another cartographic process, are instantiated meadows, those portions of the LTM map which are of concern to the vehicle during the piloting for this particular leg. This component of STM contains pointers to landmarks that are of potential value during this portion of the robot's journey. The pilot, acting on the available information, parameterizes **find-landmark** schemas obtained from the schema database and passes them to the motor schema manager for instantiation. There they remain, waiting for specific visual and positional reports to trigger their activation.

The robot generally begins its motion as a consequence of a **move-ahead** schema. As it moves, positional reports from the shaft encoders are fed to the uncertainty map manager. The uncertainty growth routines within the uncertainty map manager act on both this information and the characteristics of the terrain to increase the extent of spatial and angular uncertainty as the robot travels. This is usually done at the end of a leg or when a landmark is recognized. If no landmarks are recognized and the robot continues to move, eventually the spatial uncertainty of the robot would fill the entire map. It is essential that landmark recognition be accomplished to produce effective uncertainty management.

Based upon the robot's current uncertainty, the **find-landmark** schemas, when activated, make requests to the Expecter process to predict where in the image a landmark feature should occur. This restricts the perceptual processing associated with each landmark to reasonable limits. After the appropriate perceptual schema is run on that window of the incoming image, the result is passed to an evaluation function which determines whether or not the landmark has been recognized (i.e. exceeded its recognition threshold). Once a **find-landmark** schema has recognized the position of the landmark in the image, it posts its results in the identified landmark buffer. The uncertainty map manager uses this time-stamped information, after updating the growth of the uncertainty map based upon the likewise time-stamped position reports, to reduce the extent of the positional and/or orientation uncertainty of the robot (described in Section 3).

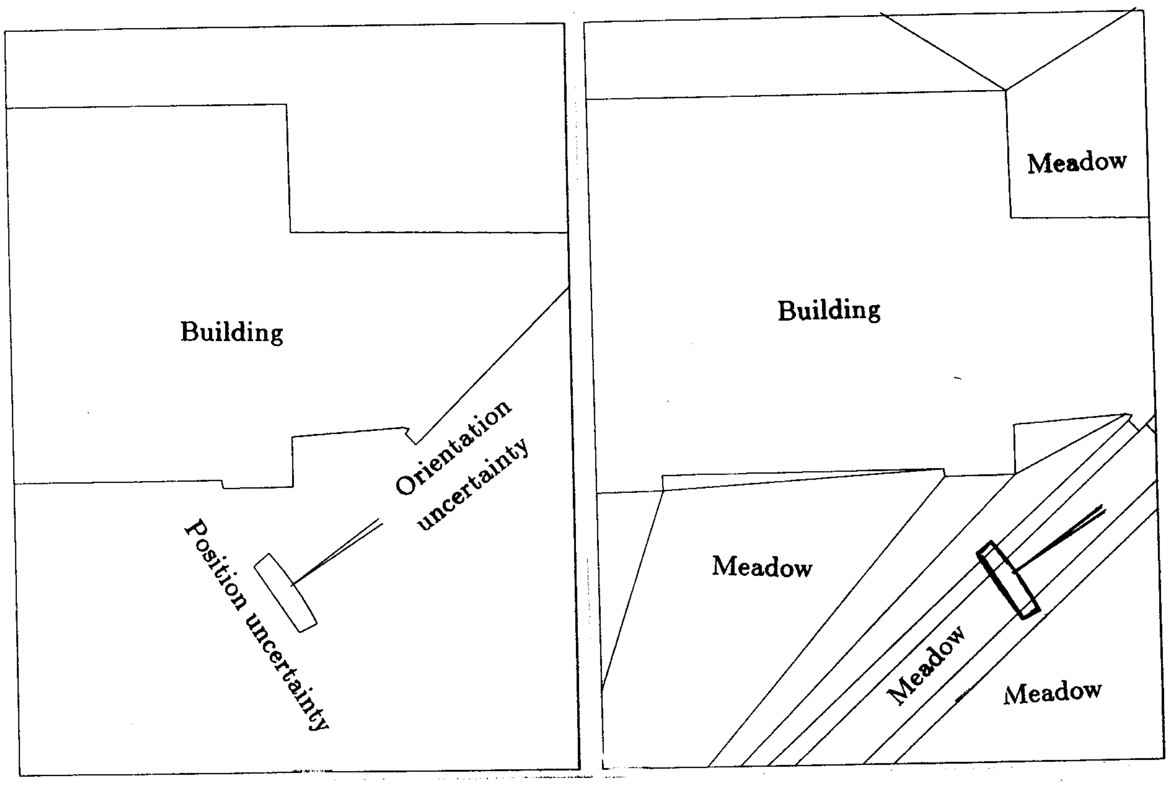
A feedback loop is achieved by the establishment of expectations based upon the current spatial uncertainty map and the subsequent recognition of landmarks within those established image boundaries, modifying the spatial uncertainty map. If no landmarks are recognized even though several have been predicted, the robot declares itself lost, stops, and then starts searching larger windows (and even rotating if necessary) in an effort to encounter something familiar and recognizable relative to its world map. In the normal sequence of events, however, the robot does not change the camera pan, tilt or focal length during leg traversal (in contrast to the Maryland system [4] described in Section 1).

The two frames of reference that need to be reconciled are the egocentric perceptual representation provided by the video images and the global meadow map itself. The spatial uncertainty map provides the mapping from one frame to the other. UMS uses an approach to uncertainty growth based on empirical terrain statistics. Consequently, there is a finite, but relatively small, probability that the robot will be located outside of the bounds predicted by the uncertainty map. Back-up re-orientation procedures are important if the robot is to regain its bearings if this occurs.

Thus far, we have assumed that the uncertainty of objects located within the global map is nil. Although this is technically an invalid assumption, as there will always be some non-zero amount of error in the positional representation of a landmark, it is safe to assume that if these data came from accurate blueprints or maps that the amount of uncertainty is small to the point of being negligible when compared to the error resulting from the robot's motion. Nevertheless, it is feasible to explicitly represent each landmark's positional uncertainty relative to the global map and to use that information in the Expecter process and in the uncertainty reduction techniques within the uncertainty map manager. This will be discussed further when these processes are described in Sections 3 and 5 respectively.

The **find-landmark** schemas also play a role in producing motor behavior (see Chapter 5). In particular a **move-to-goal** motor schema needs a **find-landmark** schema to recognize the goal towards which it is directing the movement of the robot. In this instance the **find-landmark** schema serves a twofold purpose - to provide perceptual guidance for a specific motor behavior while concurrently providing information to reduce positional uncertainty by the UMS.

Figure 67 illustrates the relationship of the spatial uncertainty map, representing both position and orientation uncertainty, and the global map. Other examples are presented in Section 6. The sections that follow describe the details of the data structures and processes that are found with UMS.



(a) (b)

Figure 67: Spatial uncertainty map in context of global map

a) Convex polygon represents robot's positional uncertainty. Two lines extending from center of maximum likelihood of robot's location indicate extent of rotational uncertainty.

b) Close up of same view showing meadows in meadow map.

§3. Spatial uncertainty map and uncertainty map manager

The spatial uncertainty map and uncertainty map manager represent and maintain the spatial uncertainty present in the robot's position relative to the modeled world. The spatial uncertainty map manager is the only process which can alter the uncertainty map itself, although the uncertainty map has read-only availability for other processes such as the Expecter and pilot.

§3.1 *Spatial uncertainty map*

The uncertainty map consists of two components, one representing the spatial extent of positional uncertainty, the other the limits of heading uncertainty. The positional component is modeled as a convex polygon which is produced by the repeated application of uncertainty transforms on earlier uncertainty maps. The coordinate system for the points of the polygon is the same as that of the global map against which it is matched by processes like the Expecter. The spatial extent of the positional area represents the likelihood that the robot's position is located within this region to a fixed probability (typically 95-99% - two or three standard deviations assuming a Gaussian distribution). Although it is theoretically possible to model this region as a three-dimensional surface, this was not done, both for reasons of computational and mathematical tractability as well as the lack of a perceived advantage to such an approach. Nonetheless, a center point representing the robot's single most likely position is maintained within this uncertainty map.

The positional component described above tells us nothing about the direction that the robot is facing within that area, only the likelihood that it is to be found there. The second component of the spatial uncertainty map represents the uncertainty in orientation. A compass wedge indicating the limits of heading uncertainty relative to the global map constitutes this model. As before, a center of probability is maintained, in this case representing the most probable orientation. The wedge is modeled by assigning limits to the extent of both clockwise and counter-clockwise rotational uncertainty from this center point.

Previous approaches have typically used circular disks to model positional uncertainty (see Section 1). This was deemed inappropriate due to the asymmetric nature of uncertainty growth and landmark recognition as described in the following subsections. To use a disk would require worst case uncertainty modeling for all situations. An example which illustrates this point nicely is the uncertainty in position as a robot moves down a path (Fig. 68). Visual feedback indicating that the robot is located on the road substantially restricts the amount of spatial uncertainty in the direction perpendicular to the road. Protracted movement causes large amounts of uncertainty to accumulate along the dimension parallel to the road itself.

Differences in the image window produced by landmark location predictions are quite pronounced if a disk model rather than a polygonal model is used. Expectations produced from the uncertainty map for landmark recognition (described below) can result in much smaller image windows if an asymmetric map is used (Fig. 69). Significant computational savings can be achieved by restricting the search space for a particular landmark to as small a region as possible. By properly orienting these windows based on the relationship of the asymmetric uncertainty map and the landmark information available in LTM, these savings can be realized.

For the remainder of this section we first describe the growth procedure used by the spatial uncertainty map manager. This is followed by an analysis of the types of landmarks used and their application in the reduction of uncertainty by the spatial uncertainty map manager. These two techniques complete the feedback cycle used for the establishment of expectations for landmark position, and after identification, reduction in the size of the uncertainty region that was used to produce those expectations.

§*3.2* *Uncertainty map growth statistics*

The spatial uncertainty map's growth arises from the difference between the commanded motion of the robot and the distance it actually traversed. The extent of growth (i.e. the amount of slippage and drift) is dependent to a high degree on the terrain that the robot is traveling on. Experimental data have been collected for each of the terrain types in the robot's current world (tile, concrete, grass, and gravel) and are presented in Table 3. It should be recognized that these data will vary based on daily conditions (e.g. long wet grass will have different values than newly mown dry grass, similarly a waxed floor will have values different from a dirty floor). If a more extensive table is built based on these varying conditions, a more representative error model of the world

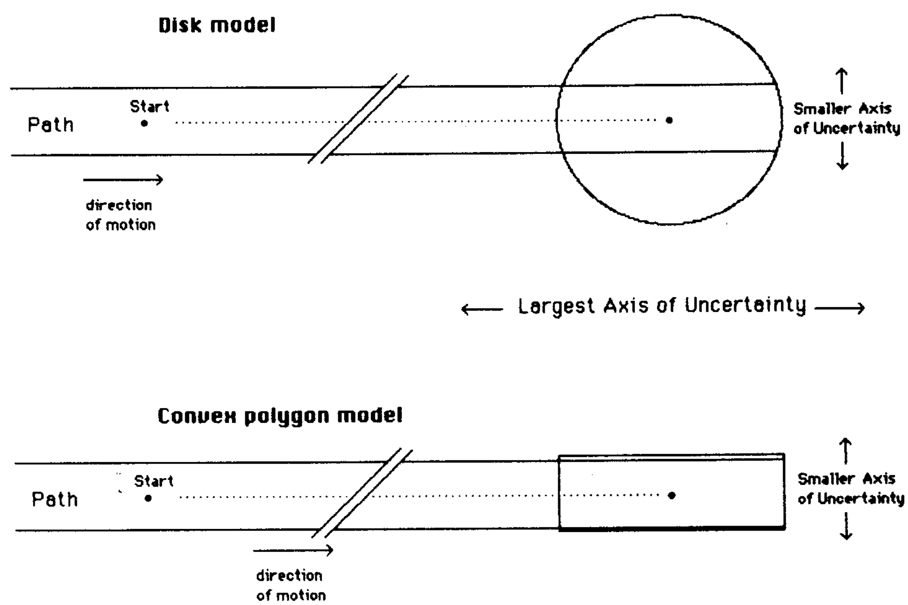


Figure 68: Disk model versus convex polygon model

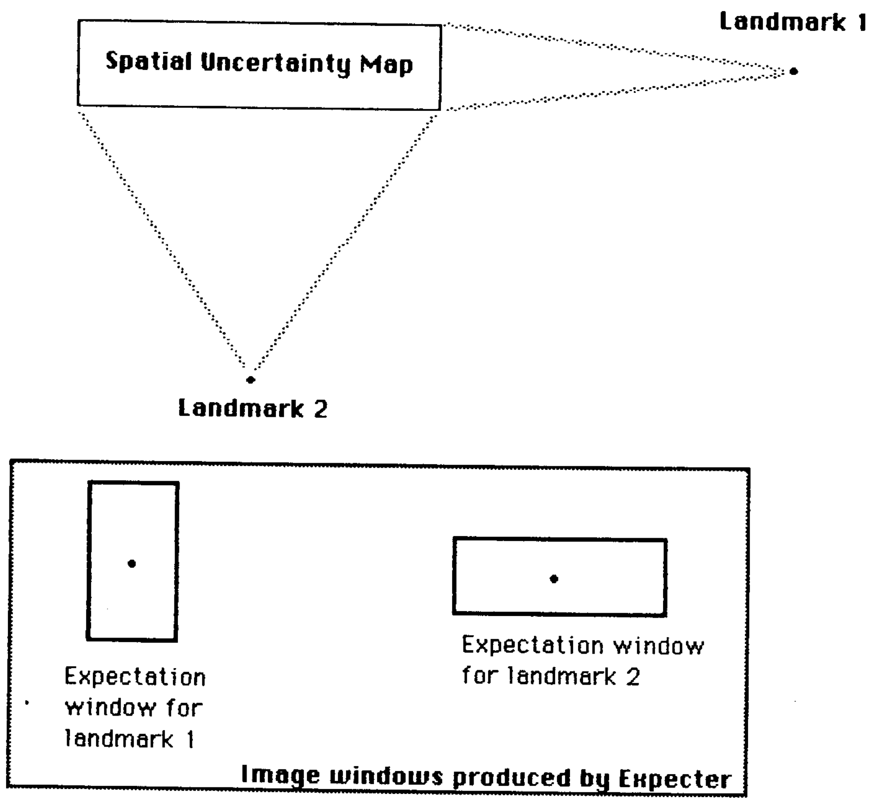


Figure 69: Image windows from convex model

For a polygonal map, smaller windows can be obtained for expected landmark location than for a circular model. Assuming the angular uncertainty is the same in all cases, Landmark 1 produces a window with a greater vertical extent, while Landmark 2's is more horizontal.

The window shape for a circular disk would generally be square instead of rectangular, and although the window itself would be simpler to compute, the overall computational cost would be greater due to the larger area required to be processed by the perception and matching algorithms.

would be available. For the sake of this dissertation however, we will assume that the statistics presented in this table are valid for all runs and all conditions for the particular terrain type concerned.

The data collected includes the following:

- Translational error (for straight-ahead motion)
  - mean translational error (loss)
  - mean inertial translational error (independent of distance traveled)
  - standard deviation of translational error
  - drift (degrees/foot)
- Rotational error (for turn commands about the robot's axis)
  - mean rotational error
  - standard deviation of rotational error
  - skitter (feet/degree)
  - standard deviation of skitter

The contribution to the uncertainty of the values used for the mean translational or rotational error depends on the distance the robot travels or turns. In all cases, an assumption is made that the robot will never go further than it is commanded. This implies that the robot does not slide significantly when it is told to stop (an invalid assumption for high-speed braking). For example, if the mean translational loss (error typically due to wheel slippage) is 0.10 (10%) and the robot is commanded to move 20.0 feet, the mean distance actually traveled would be 18.0 feet (similarly for degree errors in turning commands). The standard deviation serves to limit the likelihood of the robot being found within a given region to a specific probability (assuming a Gaussian distribution).

Inertial losses, which arise from slippage during the robot's acceleration and braking, are independent of the distance traveled. This quantity is a small fixed value that is independent of whether the robot has traveled 1 foot or 100 feet.

Table 3: Uncertainty data

	Tile	Concrete	Gravel	Dry grass
Translational Errors				
Mean Translational error (ft/ft)	0.042	0.044	0.038	0.026
Mean Inertial error (ft)	0.0084	0.0084	0.0084	0.0084
Total Standard deviation (ft/ft)	0.0028	0.0019	0.0017	0.0042
Mean Drift error (degrees/ft)	0.27	0.30	0.29	0.28
St. deviation Drift error (deg/ft)	0.12	0.14	0.14	0.14
Rotational Errors				
Mean Total Rotational error (deg/deg)	0.018	0.017	0.016	0.016
Total Rot. Standard deviation (deg/deg)	0.006	0.003	0.003	0.003
Skitter error (ft/deg)	0.0013	0.0014	0.0014	0.0012
St. deviation Skitter error (ft/deg)	0.00013	0.00029	0.00014	0.00014

The terrain statistics gathered are certainly dependent on the speed and acceleration of the robot over a given terrain type. For the relatively slow velocities used with HARV, the observed differences were negligible.

The problem of drift arises both from inaccuracies in the drive train of the vehicle and the terrain itself. As the vehicle moves, it "pulls" to the left or right depending on the particular configuration of the wheels. Drift is dependent on distance traveled. Although the drift value can be made a function of the current orientation of the wheels and handled by a simple look-up table, by overestimation we can treat this quantity as a fixed factor.

Skitter, the last error factor and the rotational analog of drift, is the robot's tendency to translate across the floor when given a command to turn. This quantity is dependent on the amount of rotation commanded. It is also orientation specific; the robot skitters (translates) to the right during a clockwise rotation (when viewed from the rear) and to the left during counterclockwise rotation. Although this asymmetry can be preserved in the growth procedures described below, the skitter component is quite small when compared to the translational error and is treated as a symmetric error.

A final comment on the validity of the statistics is in order. Although many man-days were spent on the collection and analysis of the statistical data presented in Table 3, it should be recognized that the accuracy of these figures is limited. A major effort would be required to fully model terrain characteristics. The time limitations for the development of this dissertation precluded this. Any errors in these figures have no effect whatsoever on the theoretical development of the UMS and the underlying growth routines for the uncertainty map. They have also served adequately for the experiment described in Chapter 8. If this system of uncertainty management is used for applications outside of the UMASS environment, a comprehensive study of the terrain characteristics of the domain in question is recommended.

§*3.3 Uncertainty growth procedure*

The uncertainty growth procedure resides within the uncertainty map manager. This routine draws on the statistical data described above and the robot's commanded motion in applying an uncertainty transform to the current uncertainty map. Each transform consists of a "turn and run" movement. If no rotation is commanded, the turn component is zero. If no move command is issued, the run component is zero. In general, most robot commands will take the form of first turning to a new direction and then proceeding a given distance.

Let us assume that the uncertainty map is initially a point. Figure 70 shows an application of an uncertainty transform to a point. This point is transformed by the average translational and rotational errors (plus the inertial components) to yield a new center of uncertainty. The positional extent of the uncertainty map grows depending on the standard deviation of translational error. The assumption is made that overshoot is negligible. This sector-shaped region is then enclosed in a polygon, with any skitter or drift component added. The new uncertainty map's positional component is represented in Figure 70.

The orientation component is handled in a similar manner (Fig. 71). The center of orientation uncertainty is updated based on the amount of turn commanded and the rotational losses. The standard deviation is used to asymmetrically update the clockwise or counterclockwise uncertainty limits. Any orientation drift due to the translation is then added to these limiting values.

We now have a polygonal approximation of the positional uncertainty and a compass wedge representing orientation uncertainty. The technique for a new application of the uncertainty transform based on the next "turn and run" movement is straightforward. For positional uncertainty, the geometric transform as described above for a point, is applied to each of the points of the newly formed uncertainty map polygon. The center of uncertainty is updated exactly as before. A convex hull algorithm [44] is applied to the resultant set of points and a new positional uncertainty map results. The same approach is applied to the orientation compass wedge, but since the wedge is only one-dimensional, only the maximum value for each uncertainty limit need be retained. This process is illustrated in Figure 72.

The initial approximation of the robot's position need not be a point, but can be a bounding polygon. This is desirable as it is difficult to be certain of the exact location of the starting point of the robot in the global map unless we use a surveyor's transit.

In summary, the growth procedure operates as follows. For each "turn and run" movement, an uncertainty transform is applied to the existing position and orientation components of the spatial uncertainty map (the center point and all vertex points of the

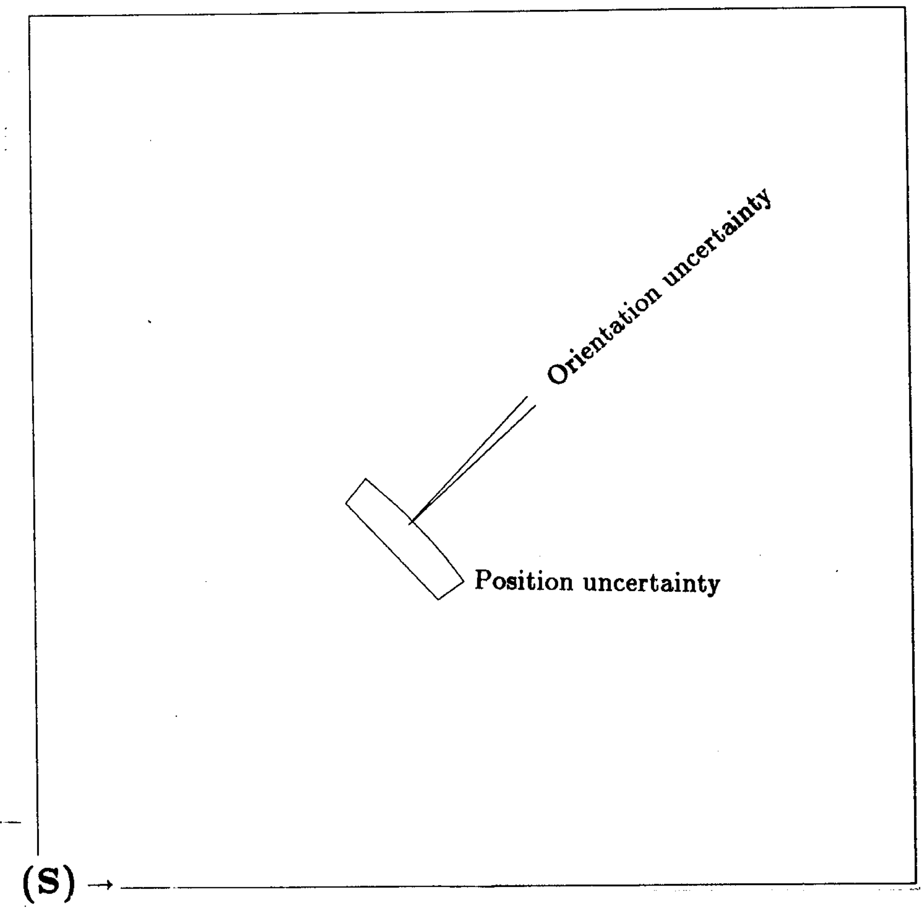


Figure 70: Point uncertainty transform

The robot starts at point S at the origin, facing directly to the right. A 45 degree turn is executed followed by a 30 foot move. This causes the spatial uncertainty map to grow and introduces orientation uncertainty as well.

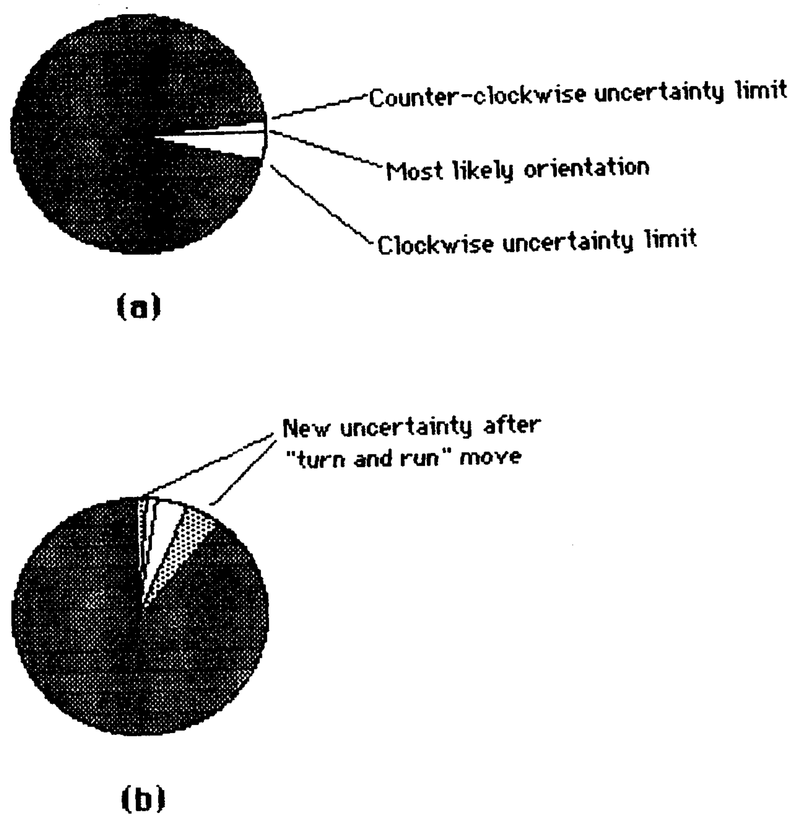
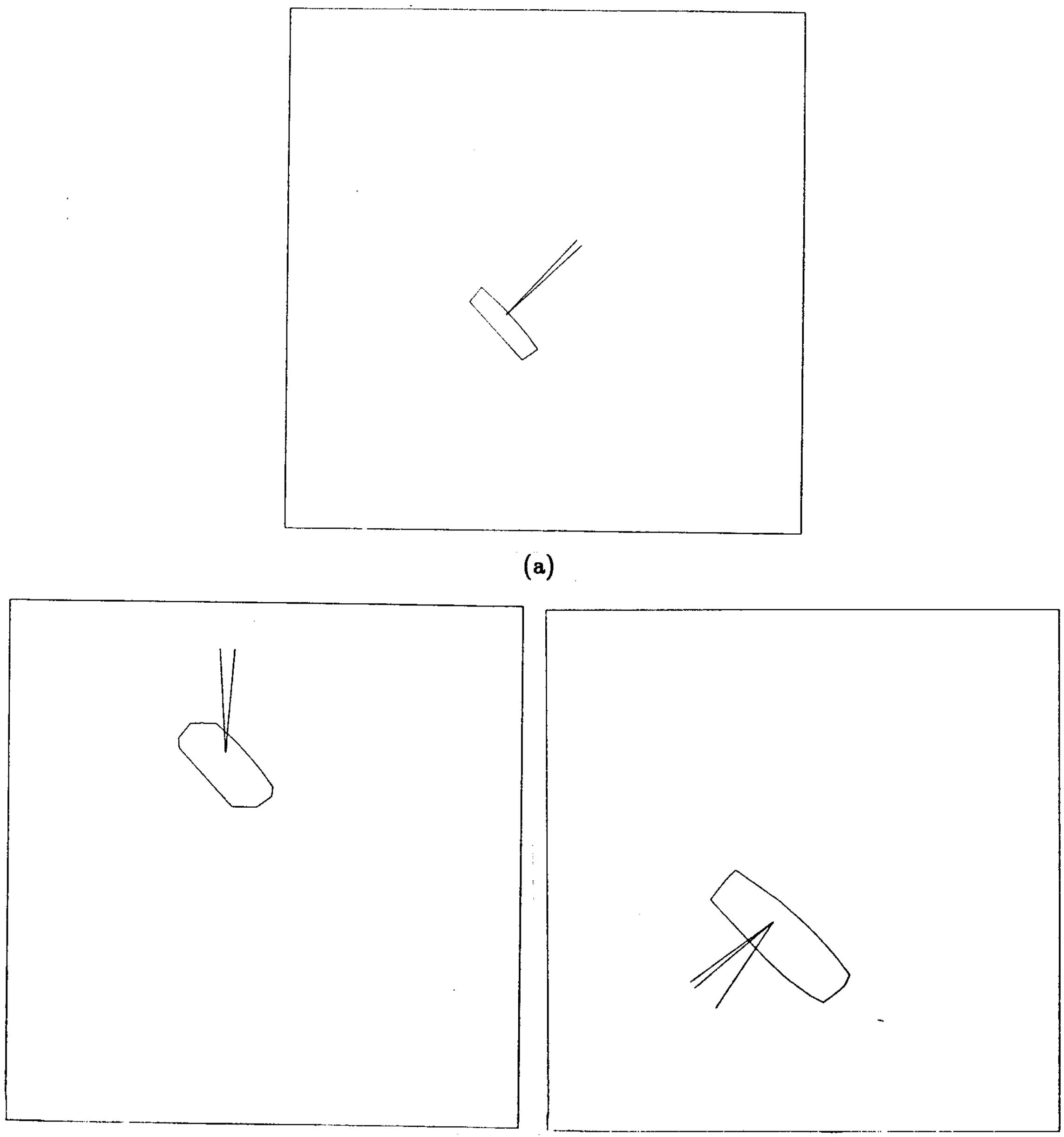


Figure 71: Angular uncertainty

The circle represents a compass with the unfilled areas representing possible headings of the robot relative to the global map.
a) Original uncertainty. The center line is the most probable orientation while the two side lines limit the uncertainty to a known probability. The center line reflects the accumulated mean errors in rotation while the side lines are produced from the standard deviations of the rotational error.
b) New uncertainty after a "turn and run" command. The direction of the turn was 90 degrees in the counter-clockwise direction, in this case producing a large increase in clockwise uncertainty. The small increase in counter-clockwise uncertainty is due to the drift that occurs during the run.



(b) (c)

Figure 72: Map uncertainty transform
a) Starting spatial uncertainty map (from Fig. 70).
b) 45 degree turn and run applied to spatial uncertainty map.
c) 180 degree turn and run applied to spatial uncertainty map.

map itself). This transform uses as inputs: the current terrain characteristics represented in a statistical format; the vehicle commands sent to the motor controllers of the robot; and the current spatial uncertainty map. It produces as output a new uncertainty map that has been increased in uncertainty in a manner reflecting the input data. This in turn can serve as input for the next "turn and run" movement.

Without environmental feedback, this map would grow to eventually fill the global map. The following subsection describes the methods that can be used to reduce both components of the spatial uncertainty map based on successful landmark recognition.

§*3.4 Uncertainty reduction procedures*

The uncertainty represented in the spatial uncertainty map is reduced by means of landmark recognition. As AuRA is predominantly a vision-based system, the types of landmarks used are those capable of being extracted from video images. Other uncertainty reduction techniques can be applied to landmarks recognizable by other sensors, but they will not be considered here.

Three distinctive landmark classes are available to the spatial uncertainty map manager for the reduction of uncertainty. The first class consists of landmarks, typically walls or poles, that produce vertical image lines that have been truncated by the top of the image. The depth to these lines is not directly ascertainable from the image data. Although their location in world coordinates is known, the distance from the robot to the landmark cannot be computed from this information alone. There is no way to tell if the fraction of the line detected in the image is a major or minor portion of the landmark's total physical extent. What can be obtained is a ray from the landmark's world position to the robot. The robot must be located somewhere along this ray. This ray thus can be used to constrain the spatial uncertainty map. The second class of landmarks consists of identifiable image features that can be used, when combined with *a priori* knowledge of the physical landmark's height and location available from LTM (i.e. their world coordinates), to tightly constrain the uncertainty map. The third class consists of long lines typically found in the ground plane such as road edges. These lines help control the uncertainty in the direction of motion. These three classes are illustrated in Figures 73-75 and are discussed in more detail below.

Reiterating, Class I landmarks are typically based on the recognition of a vertical

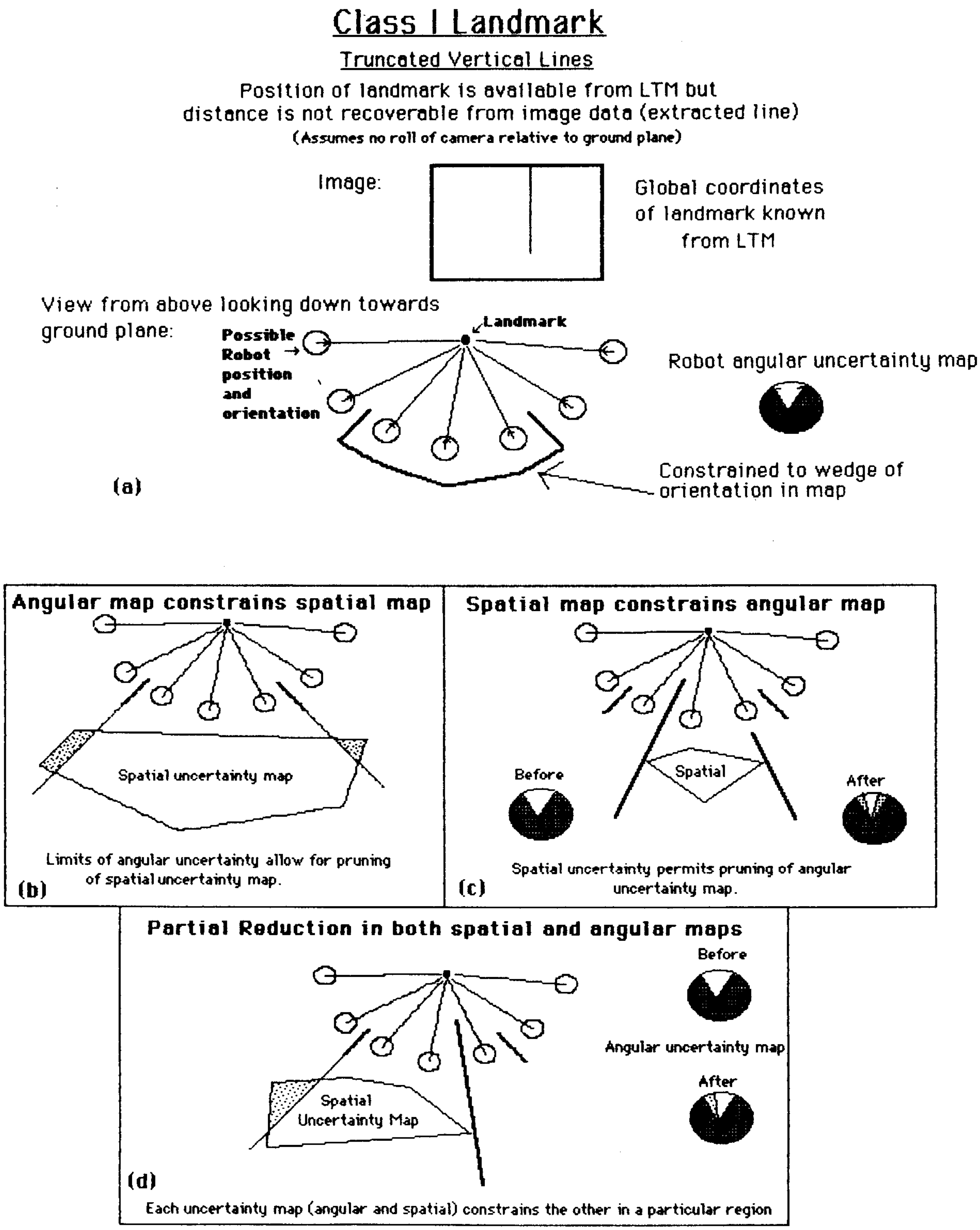


Figure 73: Class I landmarks - Vertical lines

Actual landmark yields a truncated vertical image line.

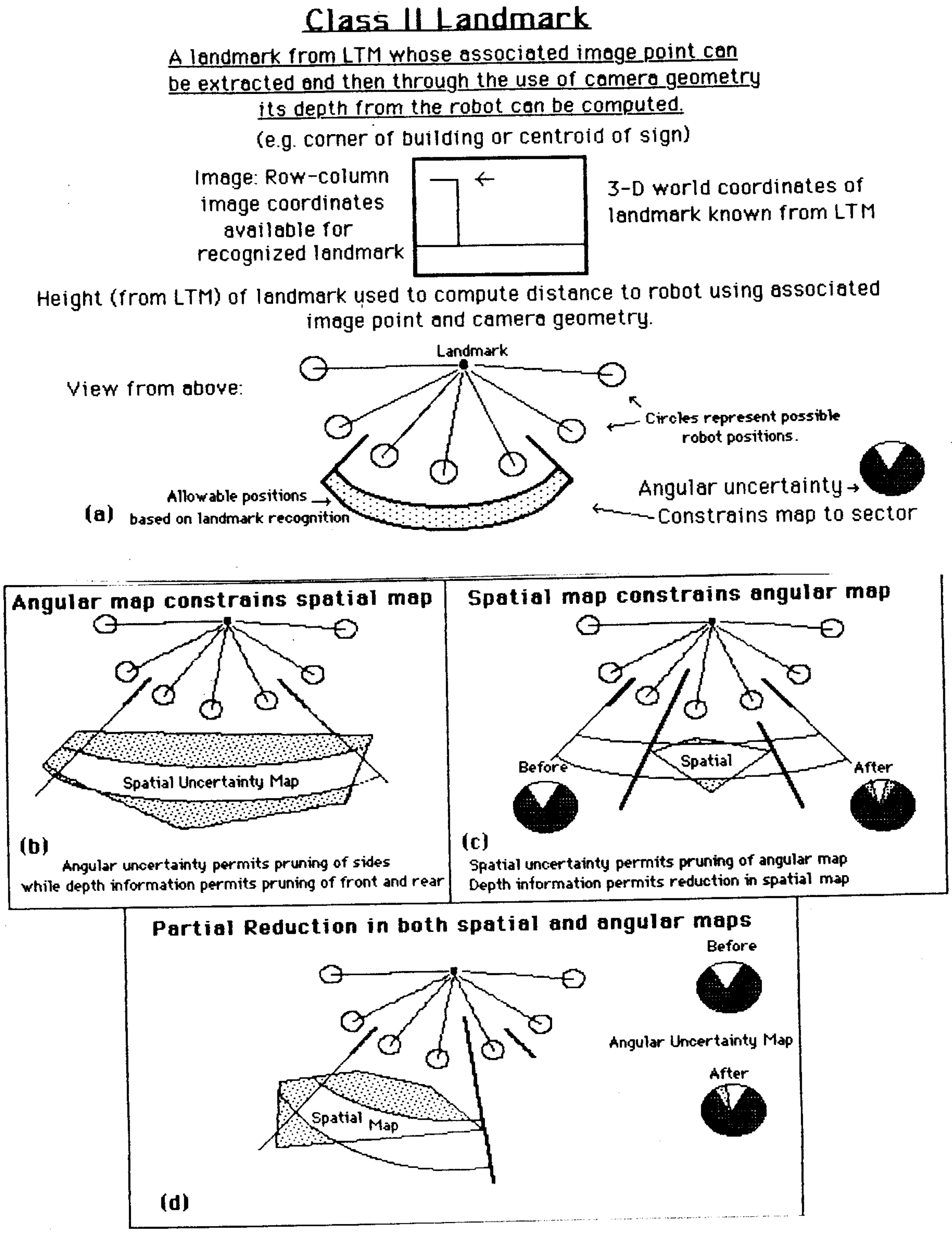


Figure 74: Class II landmarks - Points
Match made between 3D actual landmark world coordinates and 2D image plane coordinates of landmark's image.

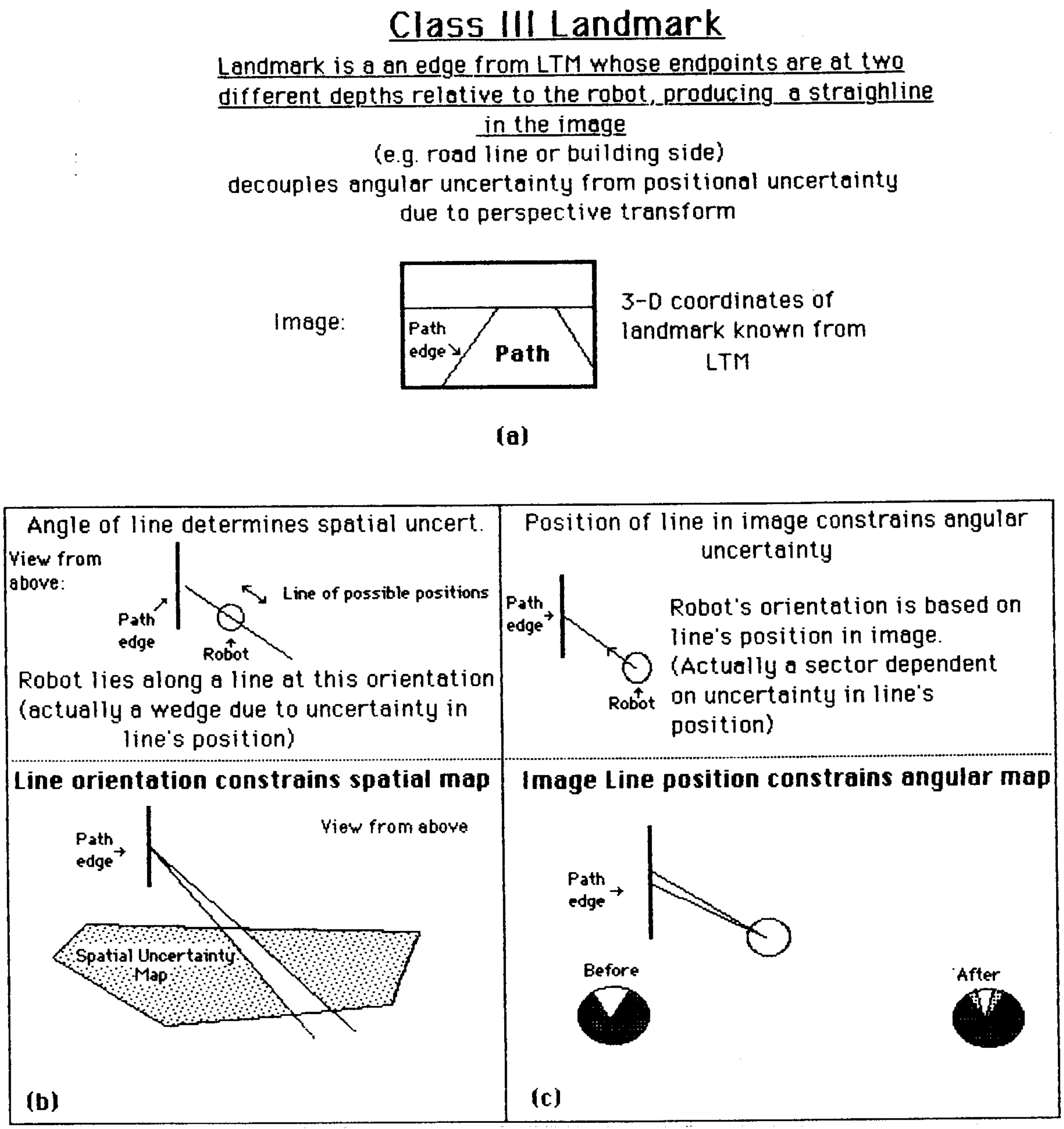


Figure 75: Class III landmarks - Ground Plane Line
The actual landmark produces a line, typically in the ground plane, that contains two endpoints at significantly different distances from the robot.

image line that has been cut off at the top of the image. This type of landmark appears, for example, with a closeby doorway in an interior scene or a nearby building outdoors. Even with *a priori* knowledge of the height of the associated feature (the doorway or building), it is impossible from this evidence alone to extract the distance to the landmark feature from the image. Only a ray emanating from the associated world map point can be used as information for spatial uncertainty map pruning purposes. This is a consequence of the fact that we do not know how much of the image line represents the total actual side of the door or building; if we are close, it may be a small fraction; if far, it may be almost the entire side. Although depth information is lacking, by combining the information available in the spatial uncertainty map regarding possible positions and orientations of the robot, uncertainty can be significantly reduced. The evidence obtained from a class I landmark consists of a ray emanating towards the camera. The position of the ray in the image plane (Fig. 73) restricts the possible orientations of the robot relative to it. The positional component of the uncertainty map when combined with the known global position of the recognized landmark restricts the possible locations of the vehicle. In some cases the extent of orientation uncertainty enables a reduction in the positional component of the uncertainty map (Fig. 73b). In others the spatial map allows reduction of the orientation uncertainty (Fig. 73c). Finally, in some instances, a reduction in both position and orientation uncertainty can be made (Fig. 73d).

Class II landmarks are landmarks whose image produces a recognizable point that can be directly matched to a landmark feature in LTM, yielding information regarding its world coordinates, height, etc. Class II landmarks (Fig. 74) provide significantly more information than do class I. Typical class II landmarks include building corners, road signs, or any recognizable feature that can be represented with its 3D coordinates in the meadow map. Both the 3D world coordinates and the matched 2D image plane coordinates are used in position estimation. Through the use of camera geometry and the perspective transform, the distance from the robot to the recognized landmark can be established within some known error limit (assuming the vehicle's relationship to the ground plane is understood). The distance error is based on camera calibration error, digitization resolution, actual landmark location uncertainty, etc. The detected landmark's relationship to the spatial uncertainty map (and hence robot) is best described as a fuzzy point location, as the actual location of the landmark is known only within limits and not exactly. It is not a sharp point due to the inherent uncertainty in the imaging process and in the actual location of the landmark. The net effect is that with this approximate depth information, we can accomplish everything that a class I landmark offered, but also reduce the forward and rearward components of spatial uncertainty (Fig. 74b-d).

Class III landmarks (Fig. 75) are similar to class I landmarks in that they do not directly provide depth information. This class of landmarks arises from landmarks in LTM that are typically located in the ground-plane in a direction that is oriented away from the camera (i.e. not parallel to the image plane) and that produce lines in the 2D camera image. The best example is a path edge. With this information it is possible, as a consequence of the perspective projection, to decouple orientation and positional errors. The angle of the line in the image plane corresponds to the relative translational position of the robot to the line (in a ray-like manner). The position of the line in the image (left or right) provides feedback on the orientation of the robot. This same type of information is used to servo the robot for path-following based on line finding as described in Chapters 6 and 8. The reductions in uncertainty that are possible with class III landmarks are illustrated in Figures 75b-c.

An important feature in the use of these classes of landmarks is that triangulation (the recognition of two landmarks in widely separated locations) is not required to improve the vehicle's estimate of its position and heading. Triangulation certainly can be subsumed by this method (e.g. identification of two class I or II landmarks). It is a goal of this system however to provide concurrent landmark recognition without forcing the camera to search through the countryside, using a pan-tilt-zoom mechanism. In this manner only relevant landmarks are sought in the direction of the robot's motion. If the robot becomes sufficiently disoriented, as recognized by exceeding a certain area threshold for the spatial uncertainty map or by failing to detect several predicted landmarks, it can then stop and look around for familiar and easily discernible landmark features.

The actual coded implementation of uncertainty reduction involves slicing the uncertainty map as specified in Figures 73-75. In the case of class II landmarks, where sectors are produced (e.g. Fig. 74b), the curves produced are enclosed with lines, maintaining the convex polygon representation for spatial uncertainty as described in Section 3.3.

How these landmarks are specified, selected, sought after, and recognized within the

UMS is the topic of the next two sections.

§4. **Find-landmark** schema selection

Uncertainty reduction cannot be accomplished without the availability of recognizable visual landmarks. These landmarks must be stored in memory and selected for possible recognition when necessary. Perceptual recognition strategies must then be associated with each chosen landmark and activated when appropriate to complete successful landmark recognition. This section describes the roles of long-term memory (LTM), short-term memory (STM), the pilot, and the motor schema manager in the context of uncertainty management. All of these AuRA components also serve other useful functions: long-term and short term memory for planning purposes, the pilot and the motor schema manager for motor behavior selection and activation. These systems will only be discussed here in the context of uncertainty management.

Long-term memory

Landmark information must be stored somewhere. As AuRA generally assumes the existence of a partially modeled world, it is a logical consequence to embed landmark data in LTM. Two choices are possible:

1. The landmarks could be created automatically based on available 3D world model data and a visibility analysis.

2. Specific landmarks can be chosen by the designer of the system and consist only of those landmarks that are expected to be particularly useful (i.e. a tuned subset of the collection in 1).

Several advantages of the second method are apparent. First, the landmarks are precompiled for a particular region and so it is merely a memory access operation to obtain them, which does not burden the system with additional computation. It is possible that this precompiling could be automated from available 3D world models. Second, and perhaps of more significance, it is also easier to ensure proper selection of landmarks and their activation ranges for the experimental testing of the system. In addition, the number of **find-landmark** schemas can be kept down to a reasonable number. AuRA stores a 3D model of the world (that is a partial wire frame world view), so it is feasible to implement a visibility search to determine which landmarks could be of value. In the interest of a reasonable completion date for this thesis however, a specific subset of landmarks was manually selected and tied into each meadow.

Each meadow contains a pointer to a landmark list. This list contains pointers to useful landmarks visible from that meadow, not only those within that meadow. For some meadows this list will be empty. For others there will be one or more landmarks available. The information stored will depend on the specific landmark but typically includes some of the following:

- Symbolic class (e.g. lamppost)
- Symbolic label (e.g. Lamppost_107)
- Landmark Class (I,II,or III)
- Activation criteria
  - Minimum and maximum effective recognition distance of the robot to the landmark
  - Acceptable viewpoints - maximum and minimum heading
- Means of identification - perceptual strategies to use
- Spectral data (if applicable)
- 3D structure (if applicable)
- Positional uncertainty of landmark in global map
- Pointers to subparts
- Pointer to parent object

A single landmark may be present in the landmark list of several meadows. The landmark need not be located in the meadow, but only visible and potentially identifiable from some point within that meadow.

Short-term Memory

When the navigator specifies a particular leg for the pilot to execute, the cartographer recognizes this and instantiates a group of related meadows from long-term memory into STM. These instantiated meadows consist of the meadows that the robot is expected to traverse during this particular leg of its journey. Adjacent and other nearby meadows (where nearness is measured by the proximity to the computed navigational path) are also instantiated. This information, which is chosen by the cartographer, restricts the number of landmarks for the pilot to search in its quest for suitable **find-landmark** schemas.

Pilot

Prior to initiating motion, the pilot accesses the landmark lists from the instantiated meadows in STM. Using the available schema library, the pilot parameterizes the **find-landmark** schemas with information available from the landmark data. It also sets activation criteria based on each landmark's location and identifiability. Appropriate perceptual schemas are parameterized with values suitable for the landmark in question. For example, if the line finder is to be used, filters and/or buckets will be tuned to specific line orientations. If the region segmentation algorithm is to serve as the basis for identification, the spectral and region characteristics will be set by the pilot. The pilot then passes this set of slot-filled schemas to the motor schema manager for instantiation.

Motor schema manager

The **find-landmark** schema instantiations (SIs) are created by the motor schema manager immediately upon their receipt. In general, many are immediately placed in a state of hibernation until the robot moves into range for potential identification. At that time they are activated and make specific calls to the Expecter process to determine their anticipated position in the image. This portion of the image is then processed by the **find-landmark** SI. If the landmark is deemed identified (see next section) that fact is placed in the identified landmark buffer with a time-stamp for the uncertainty map manager reduction processes to use. The **find-landmark** schema continues to make periodic reports, tracking the landmark over multiple frames.

The role of the Expecter process and the means for landmark identification are described in the following section.

§5. Landmark identification

Actual landmark identification consists of matching the information acquired by the perceptual schemas, using the expectations established by the Expecter process, and the landmark model itself. This section first describes the operation of the Expecter process and then the identification mechanisms available within UMS for landmark recognition.

§5.1 *Expecter*

The Expecter process restricts the search for a landmark to a particular region in the incoming image. This reduces the possibility of erroneous identifications and affords better utilization of available computing power. The initial implementation uses a single Expecter process outside the realm of the motor schema manager. It is also possible to create individual expectation schemas for each and every **find-landmark** schema running within the motor schema manager. This strategy was not employed in AuRA's first-pass implementation as the time required to produce the expectations for a perceptual schema is small compared to that for the processing of the perceptual schema itself.

By correlating the spatial uncertainty map against the known position of a landmark in LTM, a window in the image can be established which, to a known probability, contains the landmark's image projection. This is accomplished by analyzing the spatial extent and angular uncertainty of the uncertainty map in light of the landmark's global position (see Fig. 76). Worst case analysis establishes the window. The top of the window is determined by computing where the landmark would appear in the image if the robot was located at the closest point on the spatial uncertainty map, while the farthest point is used to determine the bottom of the expectation window. The leftmost and rightmost boundaries of the image window are determined by applying the clockwise and counter-clockwise uncertainties in heading respectively to the individual spatial uncertainty map vertices and determining where the landmark would appear in each case. The predicted image locations furthest to the left and right complete the rectangular window bounds. The resultant window must be adjusted so as to include an adequate area to produce the intermediate representations of image features necessary for the landmark identification processes described below. For example, if the corner of a building is to be located using the line finder, an adequate window size must be provided. This would involve a window much larger than the corner so that accurate lines could be extracted to determine the intersection that yields the corner sought for (Fig. 77a-b). On the other hand, if the Moravec operator is to be used for the same task, a much smaller window can be established for this perceptual schema (Fig. 77c-d). Figure 78 shows typical windows produced for different landmark classes.

In order for the Expecter process to reliably predict the landmark location, it must have information obtained from camera calibration procedures. The computed perspective transform is then applied to the position of the landmark relative to the possible locations of the camera in the world as determined by the uncertainty map. This yields the image window to be searched. The calibration process is described in Chapter 6.

§5.2 *Landmark discovery and tracking*

It is difficult to determine just when a landmark has been positively identified in scenes as unconstrained as those to be found in AuRA. A judicious choice of landmarks that produce relatively unambiguous data under normal circumstances is a key factor for successful recognition. Two distinct phases for landmark recognition are present: discovery and tracking. These two phases bear a strong relationship to the bootstrap and feedforward methods described in Chapter 6 in the context of perceptual analysis.

The discovery phase uses the model provided by LTM to locate a landmark within the image window. This phase is vulnerable to failure due to obscuration, poor or changing lighting conditions, etc. It is advisable that discovery of a landmark be confirmed over several images to ensure that a transitory event did not produce a mismatch in a single frame. The schema activation level mechanism readily accommodates this. Nevertheless, discovery is the most difficult component of the landmark identification process.

The tracking phase adjusts the LTM model of the landmark used for discovery to provide a newer model (within the image) that is used to guide landmark identification in images acquired after the initial phase. The model is continually adjusted as successive images are acquired. The newly produced model is compared with the original LTM version to ensure that the landmark was not incorrectly identified in the first place. Tracking generally assumes reliable discovery. If a landmark is shown to be incorrect during the tracking phase, the robot must assume that its uncertainty map is partially invalid and institute special methods to regain its bearings. The concepts involved in

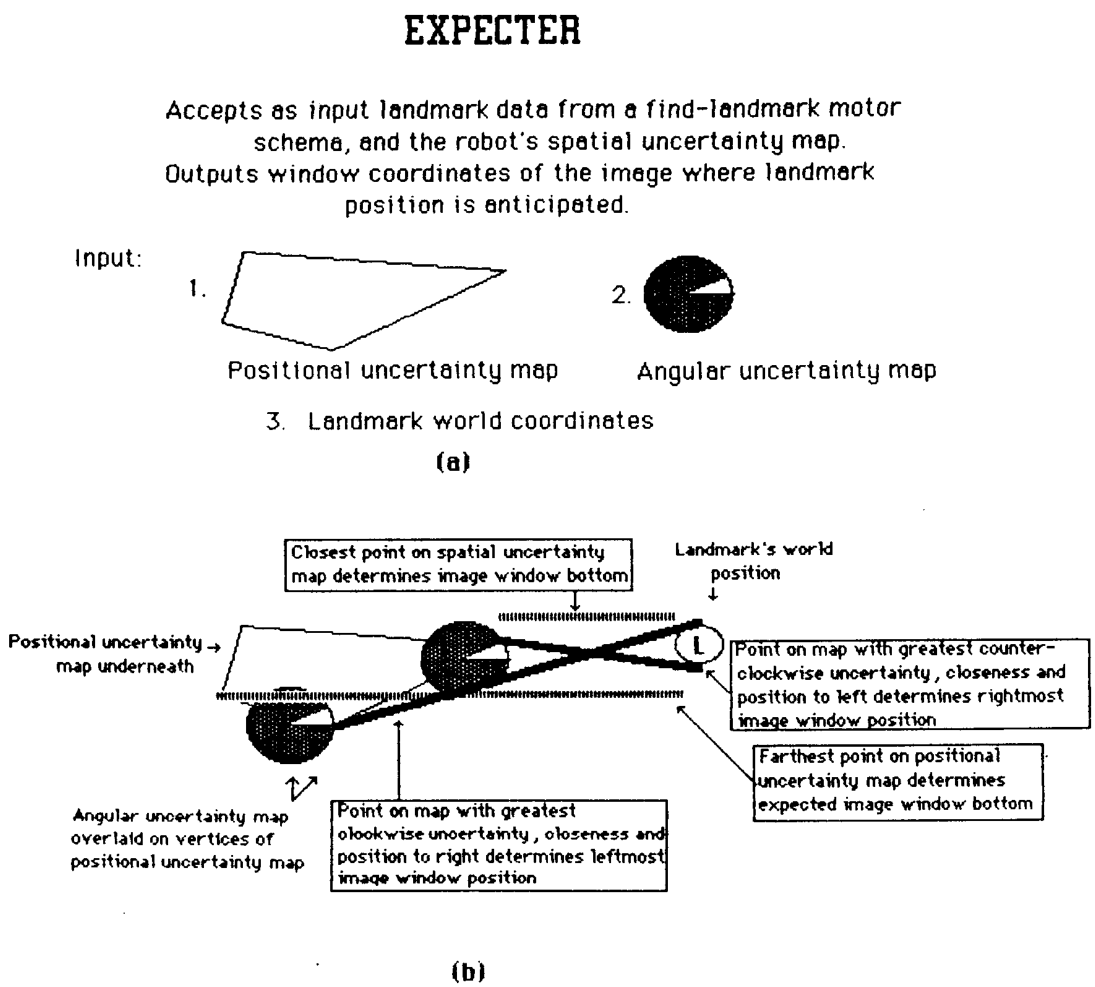


Figure 76: Expecter

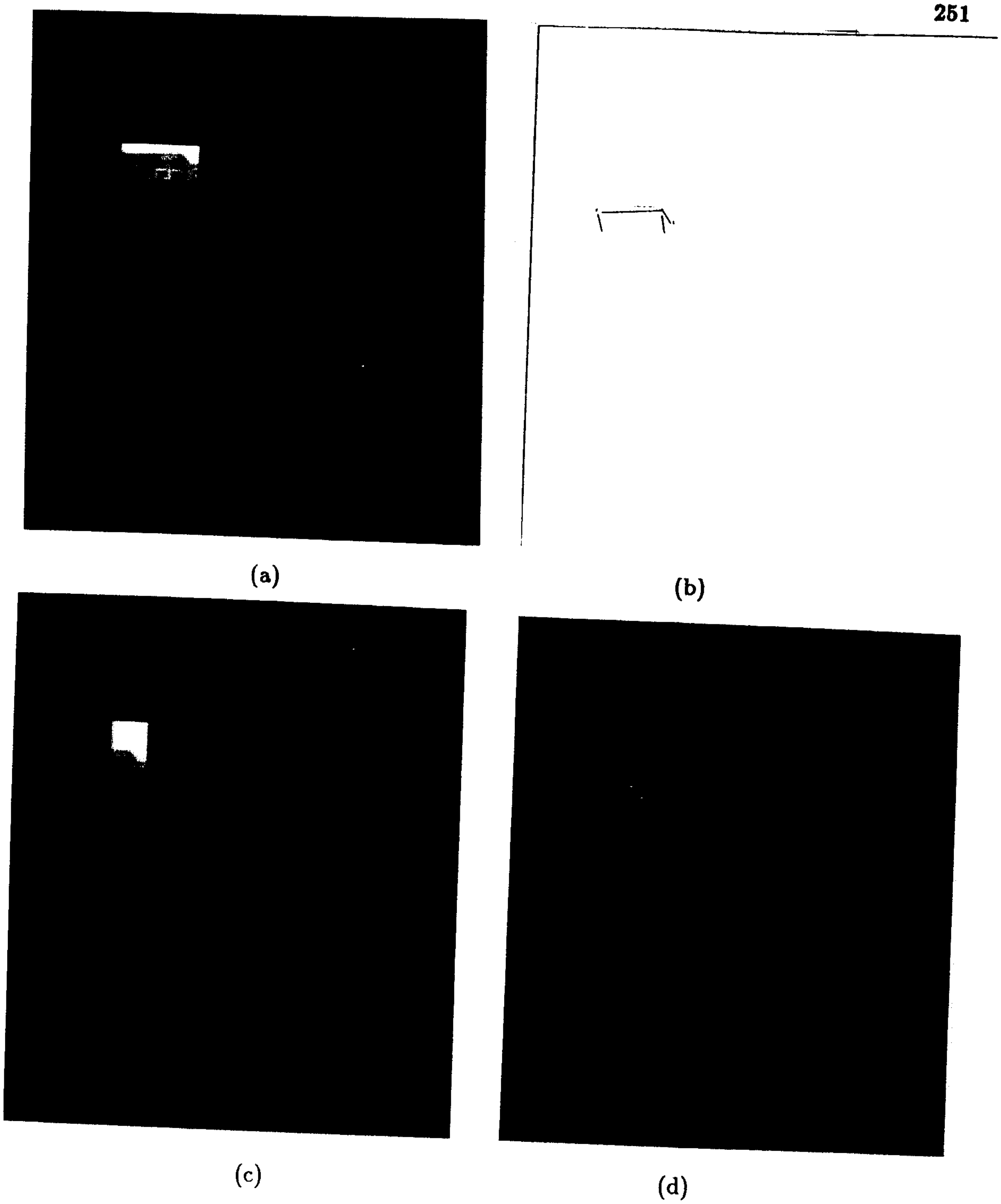
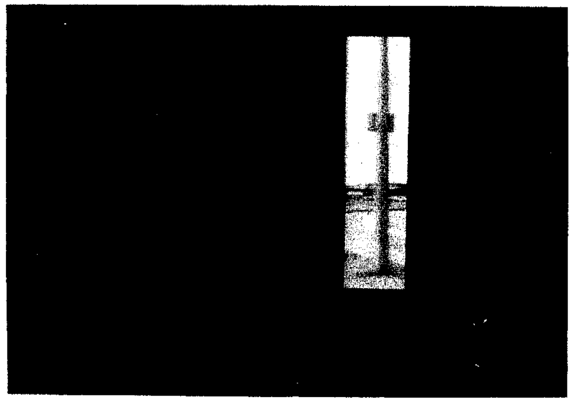
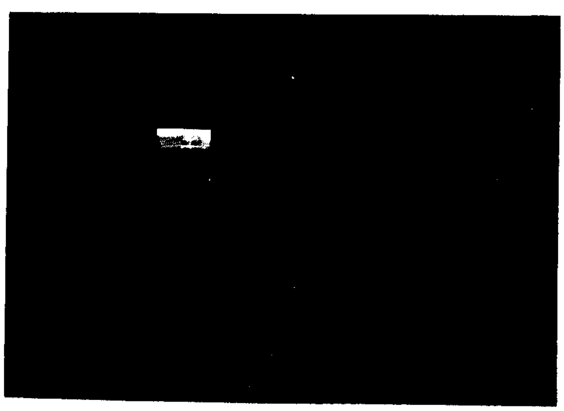


Figure 77: Expecter windows

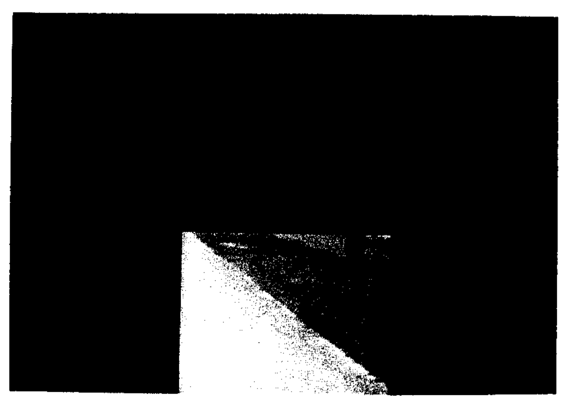
a) Window size (search area) for building corner using fast line finder.
b) Results of fast line finder within window a).
c) Window size (search area) for building corner using Moravec operator.
d) Results using Moravec operator within window c). The interest point furthest to the top and right identifies the corner.



(a)



(b)



(c)

Figure 78: Typical landmark class windows
a) Class I - Lamppost
b) Class II - Corner of building
c) Class III - Path edge.

tracking (feedforward) are explained in Chapter 6. The remainder of this section will deal with the more difficult problem of discovery.

A separate thesis could be written about the problem of discovery of landmarks in natural scenes. The UMASS schema group [42] is addressing the problem of object recognition which closely overlaps landmark discovery. Other projects, described in Section 1, report possible mechanisms to handle this difficulty [4,17]. Burns and Kitchen [32] are developing means for recognizing 3D objects in 2D images using prediction hierarchies for potential use within UMS. "Generic views" serve as a means for compiling 3D information into a useful format for 2D recognition. Techniques for indexing into a large database of objects and their object views are being developed. For the implementation version of AuRA's UMS, somewhat naive approaches will be used for the discovery process. These include line matching, region extraction, and corner identification.

Class I landmarks, typified by strong vertical lines, are identified through the use of the fast line finder described in Chapter 6. The fast line finder is run within the window produced by the Expecter. If a sufficiently strong line of proper orientation is encountered, where strength is measured by length and gradient magnitude, the landmark is deemed identified and the **find-landmark** schema converts to the tracking phase.

Class II landmarks, which yield the depth of a modeled landmark, are extracted using the line finder, the region segmenter, the interest operator, or some combination of these perceptual strategies. The multiplicity of algorithms available allows the opportunity to explore cooperating schemas as a confirmation mechanism. Landmark discovery is declared when specific certainty thresholds for a single algorithm are exceeded and/or mutual concurrent discovery occurs from different vision algorithms.

Class III landmarks, most commonly path edges, are identified by the fast line finder path-edge extraction method and/or by the region segmenter (see Chapter 6). Whenever a landmark is identified, either by discovery or tracking methods, its time-stamped location relative to the robot is stored in the identified landmark buffer. This is done independently of the class.

The techniques used for actual landmark discovery in the current version of AuRA do not use sophisticated 3D models for landmarks. Although this limits the current versatility of the system, in particular during conditions that produce partial or complete obscuration of landmarks, more sophisticated strategies for landmark discovery can be easily embedded in AuRA when they become available. This is achieved by using the feature editor (Chapter 4) for the meadow map as the mechanism for the addition of new models and by taking advantage of the modularity afforded by the use of schema-based control (Chapter 5). Ultimately, as the VISIONS system approaches real-time scene interpretation with the advent of the UMASS IMage Understanding Architecture, more robust methods for landmark discovery will be available.

§6. Examples and implementation

We describe here examples showing the growth of the spatial uncertainty map as a consequence of motion over differing terrain types, and its resultant pruning due to landmark recognition. The examples in this section do not deal specifically with the landmark discovery process. Landmarks are entered in the simulations below through the identified landmark buffer, without concern for the specific perceptual processing that placed them there.

The implementation of UMS is in Common LISP and runs on a VAX-750. The supporting cartographic processes and representations are coded in C.

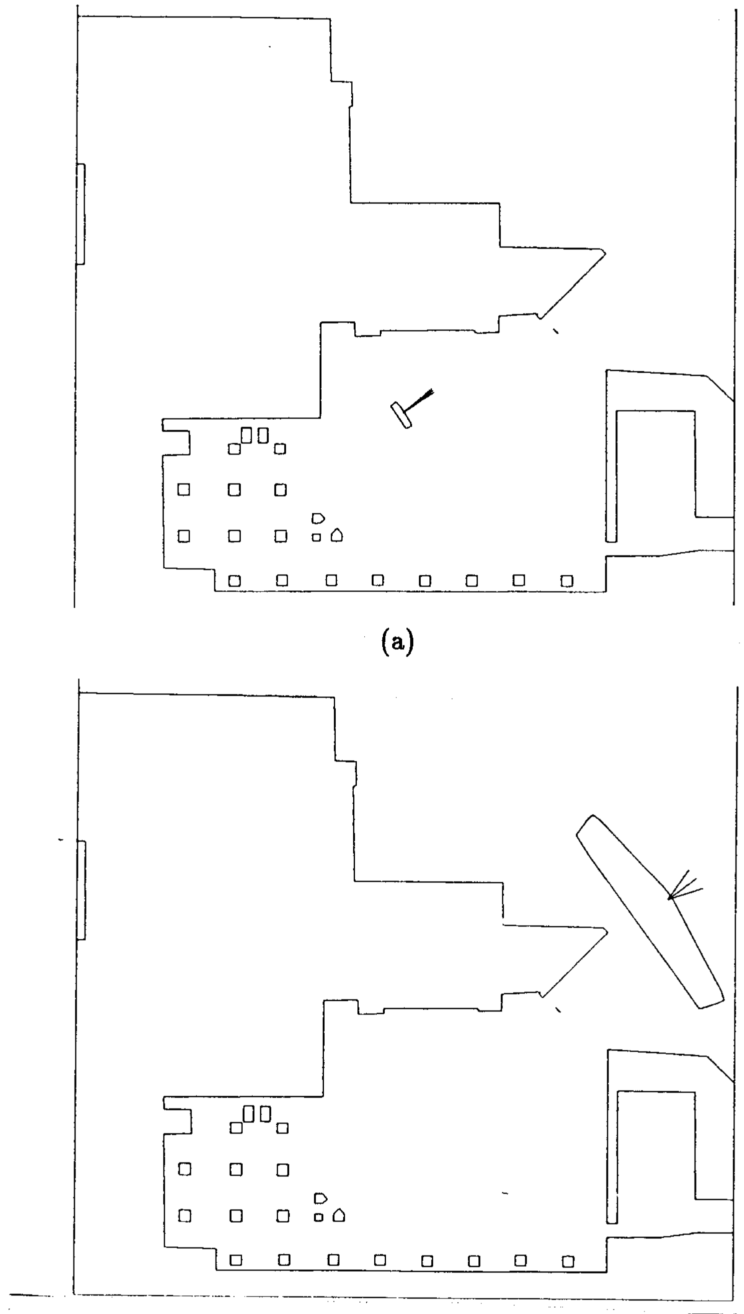
Figure 79 shows the growth of the spatial uncertainty map during repeated movement of the robot without landmark recognition feedback. Note that the unchecked growth rapidly covers a large portion of the global map.

Figure 80 shows the importance of landmark recognition in controlling the spatial and orientation uncertainty over the same path. Marked reductions in both positional and rotational uncertainty are apparent upon inspection of the figures.

Chapter 8 presents a localization experiment involving the spatial uncertainty map using images acquired from our mobile robot HARV.

§7. Summary

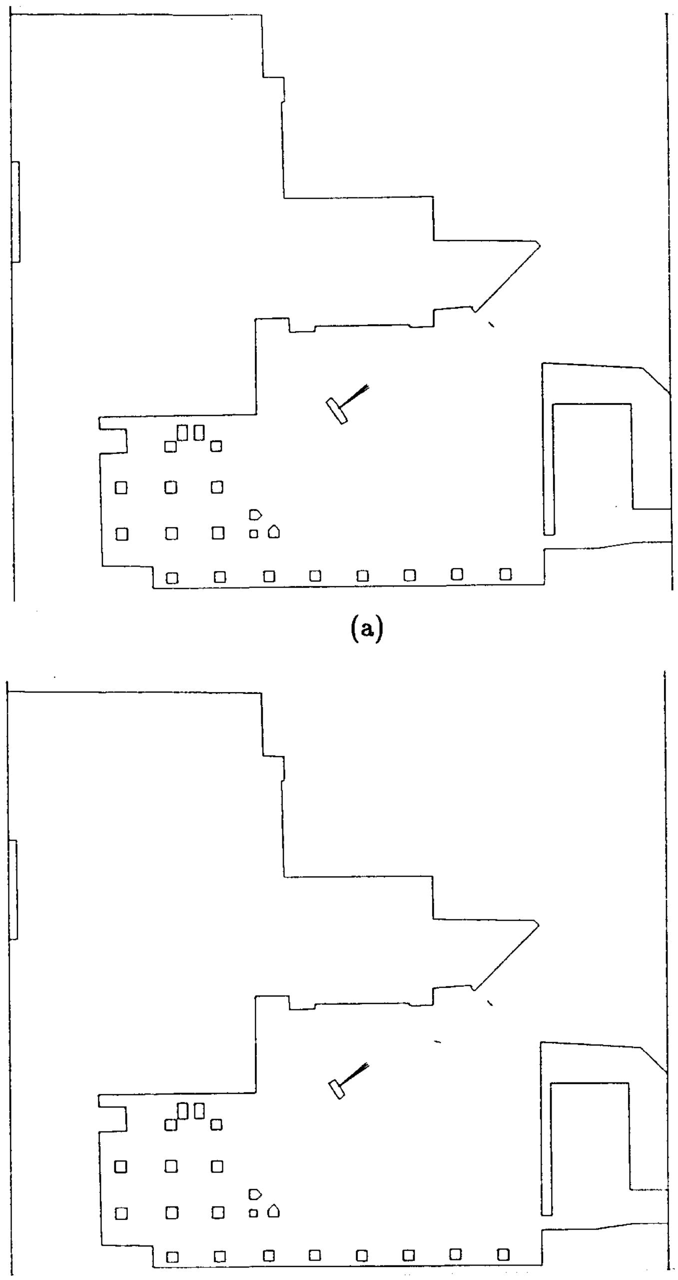
An Uncertainty Management System has been developed for AuRA to provide for the efficient use of computational resources in the guidance of perceptual processing, and as a means to ensure successful navigation of a mobile robot within a partially modeled environment. To accomplish this, a spatial uncertainty map representing both



(a)

(b)

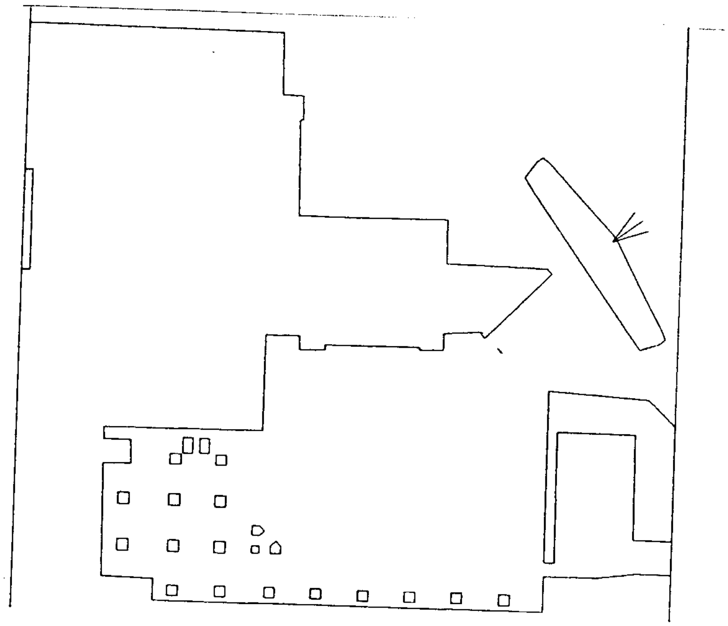
Figure 79: Uncertainty map growth without recognition feedback
a) Small amount of movement produces small uncertainty map.
b) Continued motion without landmark recognition produces large spatial uncertainty map.



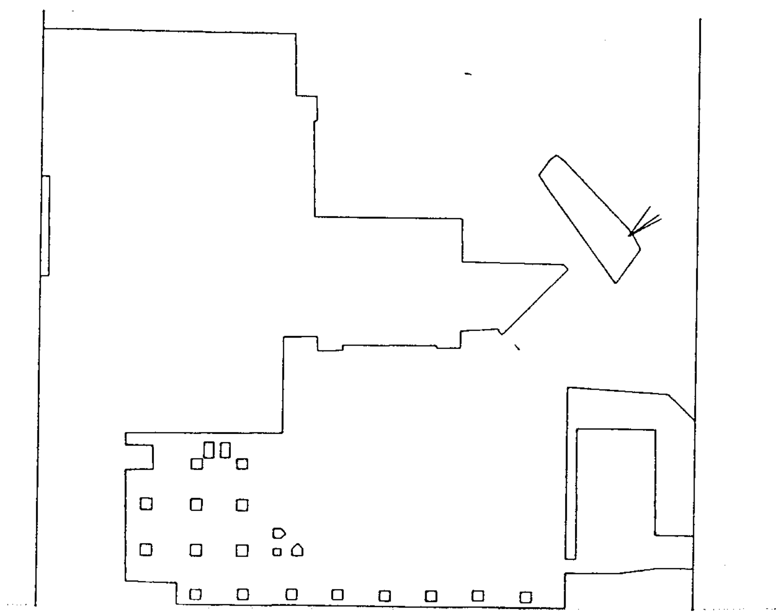
(a)

(b)

Figure 80: Uncertainty map with recognition feedback
a) Original small uncertainty map.
b) After pruning based on identification of building edge.
(Figure continued on following page).



(c)



(d)

Figure 80 continued.

c) After next movement but before next landmark recognition.

d) Reduction in both spatial and angular uncertainty due to recognition of lamppost. This case corresponds to Figure 73d, where uncertainty reduction is made to one side of the positional component (the rightside) and to one side of the orientation component (clockwise uncertainty).

positional and orientation uncertainty has been created. Specialized processes, used for both the growth and the reduction of this uncertainty map, have been implemented. Landmark selection to provide appropriate feedback for the control of uncertainty is handled by the pilot, based on the current navigational goals. An Expecter process is used to guide perceptual schemas in their interpretation of image events by restricting those interpretations to specific portions of the image.

AuRA's approach to uncertainty is based on the tenet of action-oriented perception. The advantage of this approach lies in the ability to restrict the computational needs of perceptual processes by limiting their operation to only portions of incoming images. This is important when many different processes are performing different perceptual tasks.

This particular approach for uncertainty management is somewhat narrow in focus and is not claimed to be generalizable. It is geared specifically for mobile robot architectures that contain significant amounts of reliable *a priori* knowledge. It would not work well in systems that acquire their world maps dynamically. The designer must be careful in the accuracy of his world map regarding landmark position. Significant errors in the *a priori* world map would force the robot to stop and initiate more costly means for localization (e.g. scene interpretation).

Most of the implementation of the UMS is complete, although it is not fully integrated with the navigational components of the AuRA architecture. Landmark discovery is one area that remains to be worked on. Hopefully when the UMASS Image Understanding Architecture is completed and running the VISIONS system, this problem will be easily manageable.

Another area for growth is the replacement of the shaft encoders with inertial navigation, eliminating the need for terrain modeling and restricting uncertainty growth to be based on the drift and other cumulative errors found with this more costly piece of hardware. Predictions from the Expecter would then be tighter, resulting in even lower processing demands.

Uncertainty management at the mission planner level has been largely ignored in this treatment as it addresses only spatial issues. Symbolic reasoning for dealing with unpredicted events will need to be studied when the mission planner takes on a fuller implementation.

Additional work on more sophisticated vision algorithms that recognize expected landmarks from their 3D models is an important area of future research. By using available knowledge from the spatial uncertainty map and long-term memory, limitations on each landmark's pose and distance relative to the robot can be obtained. This scale and orientation information, when applied to the landmark's 3D model, can then be used to invoke the perceptual strategies that are most appropriate for the identification task.