Navigation and Mapping in Large-Scale Space

Benjamin J. Kuipers and Tod S. Levitt

In a large-scale space, structure is at a significantly larger scale than the observations available at an instant. To learn the structure of a large-scale space from observations, the observer must build a cognitive map of the environment by integrating observations over an extended period of time, inferring spatial structure from perceptions and the effects of actions. The cognitive map representation of large-scale space must account for a mapping, or learning structure from observations, and navigation, or creating and executing a plan to travel from one place to another. Approaches to date tend to be fragile either because they don't build maps; or because they assume nonlocal observations, such as those available in preexisting maps or global coordinate systems, including active landmark beacons and geo-locating satellites. We propose that robust navigation and mapping systems for large-scale space can be developed by adhering to a natural, four-level semantic hierarchy of descriptions for representation, planning, and execution of plans in large-scale space. The four levels are sensorimotor interaction, procedural behaviors, topological mapping, and metric mapping. Effective systems represent the environment, relative to sensors, at all four levels and formulate robust system behavior by moving flexibly between representational levels at run time. We demonstrate our claims in three implemented models: Tour, the Qualnav system simulator, and the NX robot.

A person finding a way through a building, moving from place to place within a city, or navigating over open terrain is dealing with a *large-scale space*, a space whose structure is at a significantly larger scale than the observations available at an instant. Thus, to learn the large-scale structure of the space, the traveler must necessarily build a cognitive map of the environment by integrating observations over extended periods of time, inferring spatial structure from perceptions and the effects of actions.

Large-scale space and the corresponding cognitive map representation cannot be defined independent of sensory perceptions or motor actions used to observe and move about in this environment. For example, a work bench observed by a laser-bearing robot is not a large-scale space, but the moon is a large-scale space relative to a land-roving robot. A microchip is not large scale relative to an optical inspection system, but a grasshopper ganglion is a large-scale space when observed by an electron microscope.

The cognitive map representation of large-scale space must solve two closely linked problems: *mapping*, or learning the cognitive map from observations, and *navigation*, or creating and successfully executing a plan to travel from one place to another. These problems are important for three reasons.

First, robot navigation and mapping in large-scale space is an important practical problem for applications, including space and undersea exploration, automated manufacturing and transportation systems, and toxic waste cleanup. Second, spatial metaphor is clearly an important aspect of commonsense reasoning, suggesting that the representations for the cognitive map are applied to other domains by structural analogy. Third, because the rate of new observations is constrained by the rate of physical travel, large-scale space can serve as a "cognitive drosophila"—an accessible laboratory domain for the investigation of perception, learning, and problem solving with incomplete knowledge.

Because the development of autonomous mobile robots has become an increasingly important research area, a number of proposals for spatial mapping and navigation algorithms have been presented. However, most of these have been fragile in the sense that they do not build maps or that they depend on accurate preexisting maps of the robot's working environment; active landmark beacons; or geo-locating satellite or other forms of absolute, global coordinate systems.

Robust performance in both mapping and navigation means not only that performance should be excellent when all resources are plentiful but that performance should degrade gracefully when resources are limited. Several types of resources have limitations that are relevant to these tasks.

First is resource-bounded time and memory. Because a robot must act in real time relative to a task environment, it has limited time to assimilate its observations into the map or search a complex map before action is required. At the same time, its computing capacity and working memory can be shared with other pressing tasks. Second is accurate sensory information. Highly accurate observations of the environment can require more computing resources, more favorable observation conditions, or

The TOUR Model	QUALNAV	The NX Robot
Sensorimotor		
Alternating sequence of views and actions	Paths of viewframes: sequences of sets of vectors pointing to landmarks (heading and distance, with uncertainty).	Ring of range-sensors, varying continuously with time. Absolute compass. Two tractor-type chains.
Procedural		
Route procedures built from sensorimotor schemas	Control strategies for crossing a virtual boundary defined by two landmarks.	Control strategies for hill-climbing to define distinctive places, and path-following to define path-segments; selection and execution of strategies.
Topological		
Place-path networks; boundary relations, regions, and skeletal networks.	Network of adjacency relations among orientation regions defined by virtual boundaries.	Network of places and path-segments, as in the TOUR model.
Metrical		
Local geometry at places and along paths. Orientation frames for global frame of reference.	Robust computation of relative position vectors between places with overlapping viewframes (relatively insensitive to uncertainty on distance or angle estimates).	Local geometry at places. Path-segments described as generalized cylinders: axis-length, axis-shape, and cross-section profile.

Table 1. Three Implementations of the Spatial Semantic Hierarchy.

more expensive hardware than is available to the robot. Computational capabilities are third. Either the intrinsic capabilities of the robot or the quality of the data can constrain the type of operations that can usefully be applied to data. Inverse trigonometric operations and scalar multiplication require ratio data, in which a numeric value is calibrated with respect to a true zero. Trigonometric operations can require only interval data on angles, where differences are well defined, but absolute angles are not required. Some control algorithms require only ordinal data, and recognition can sometimes be performed using only nominal data.

How can we achieve robust performance in the face of serious resource limitations? It is worth noting that human mapping and navigation in large-scale space are highly robust (Lynch 1960), even in the face of substantial resource limitations (Miller 1985). The psychological literature, dealing with developmental sequences (Piaget and Inhelder 1967; Siegel and White 1975), individual variations, and the strengths and weaknesses of adult spatial mapping and navigation (Lynch 1960), provide a significant number of useful constraints on the structure of human knowledge representation. Inspired by these empirical results and based on our experience with several successful mapping and navigation programs (Kuipers 1977, 1978, 1979, 1982, 1983, 1985; Levitt et al. 1987a, 1987b; Kuipers and Byun 1987), we propose the following hypothesis.

There is a natural four-level semantic hierarchy of descriptions of large-scale space that supports robust map learning and navigation:

1. *Sensorimotor:* The traveler's input-output relations with the environment.

2. *Procedural:* Learned and stored procedures defined in terms of sensorimotor primitives for accomplishing particular instances of place-finding and route-following tasks.

3. *Topological:* A description of the environment in terms of fixed entities, such as places, paths, landmarks, and regions, linked by topological relations, such as connectivity, containment, and order.

4. *Metric:* A description of the environment in terms of fixed entities, such as places, paths, landmarks, and regions, linked by metric relations, such as relative distance, relative angle, and absolute angle and distance with respect to a frame of reference.

In general, although not without exception, assimilation of the cognitive map proceeds from the lowest level of the spatial semantic hierarchy to the highest, as resources permit. The lower levels of the cognitive map can be created accurately without depending greatly on computational resources or observational accuracy. A complete and accurate lower-level map improves the interpretation of observations and the creation of the higher levels of the map.

For route planning and navigation, the topological and metric levels provide the most powerful problem-solving capabilities but are also the most vulnerable to resource limitations. The sensorimotor and procedural levels are frequently capable of solving navigation problems, although perhaps providing a less optimal or less informative solution.

Of course, a given person's cognitive map consists of many fragments, represented in the different levels of this semantic hierarchy, and additional knowledge at one level can compensate for gaps at another. The semantic hierarchy clarifies the underlying structures of the knowledge but does not correspond straightforwardly to distinctions among individuals.

We applied this semantic hierarchy to the design and implementation of three programs to solve the mapping and navigation problems in large-scale space. The three programs—Tour model (Kuipers 1977, 1978, 1979), the Qualnav model and simulator (Levitt et al. 1987a, 1987b, 1988), and NX (Kuipers and Byun 1987)—represent significantly different instantiations of the semantic hierarchy we defined and solve different aspects of the problem (see table 1). The scope of this article does not permit a detailed comparison of our approaches with others (for example, McDermott 1980; Davis 1983; McDermott and Davis 1984) that focus primarily on the metric level and derive information at other levels from a metric map.

The Tour Model

The Tour model was primarily developed between 1974 and 1979 (Kuipers 1977, 1978, 1979). Its goal is to account for the robust performance and wide variety of incomplete knowledge exhibited by the human cognitive map. The cognitive map has a number of attractive properties as a research domain for the investigation of states of incomplete knowledge.

First, knowledge is acquired over significant periods of time because observations are constrained by the speed of physical travel. Second, cognitive development takes place over a period of a decade or so, and there has been extensive study of the developmental stages of the cognitive map in children (Piaget and Inhelder 1967; Siegel and White 1975). Third, significant individual variation exists among cognitive maps in adults (Lynch 1960), leading one to suspect that there are distinct pathways for the development of a cognitive style. Finally, there is significant variation in cognitive maps as a function of the physical structure of the environment and the type of experience with it (Lynch 1960; Chase 1982), making it easy to test knowledge representation hypotheses with "thought" experiments.

The Tour model was designed to factor the issue of image processing of sensory input from observation assimilation into a description of the large-scale environment. Accordingly, it treats the sensory apparatus as a black box, making only the simplest assumptions about its internal structure. The spatial environment is treated as something that is only indirectly perceived, whose description must be constructed from a person's egocentric sensory input.

The Tour model has been implemented in Lisp several times over the past decade and tested on a variety of small simulated environments, as many as a few dozen places and paths, and perhaps a hundred or so views. Locality of access to the knowledge base ensures that the program runs quickly, and its computational costs are not particularly sensitive to the overall size of the knowledge base. Elliot and Lesk (1982) implemented a route finder that generated an internal map structure from United States Geological Survey data tapes of a large region of northern New Jersey and demonstrated that cost-effective route finding could be done using a version of the boundary heuristic discussed later.

The elements of the Tour model representation are presented here in a declarative, in some cases axiomatic, form, following Marr (1982) and Hayes (1979). They argue for defining the mathematic structure of the knowledge prior to its computational and procedural implementation. However, a little reflection shows that in almost all cases, the declarative form translates straightforwardly into an incremental assimilation rule for acquiring the knowledge from observations.

Sensorimotor Interaction with the Environment

The sensorimotor world of an agent, in this case, a traveler in a fixed environment, is a purely egocentric description of sensory input and motor output and contains no references to fixed features of the external environment. Models of spatial learning in which sensory perception provides direct information about places or objects in the world are, at best, glossing over an important and nontrivial inference step, and more likely, committing a serious category error.

The Tour model (Kuipers, 1977, 1978, 1979) assumes a sensorimotor world consisting of views and actions. A *view* represents the traveler's sensory input at a given instant. Although recognizing that this sensory description might be arbitrarily complex from the point of view of the Tour model, it is treated as an opaque object that can be matched against other views and used as an index for associative retrieval of other structures. An *action* represents motion that the traveler can take within the environment and typically changes the current view. An action is modeled as a change of state, with the current view defined immediately before and after an action but not changing continuously during it.

Views are not necessarily visual. For a blind traveler, the sensory input at an instant can consist of a collection of aural, tactile, and olfactory stimuli, which are assimilated into a cognitive map, much as visual views are by a sighted traveler. The principal requirement is that views be distinctive enough to allow assimilation of the environmental structure. Open ocean and the neophyte's view of desert or forest might fail to satisfy this requirement.

Views and actions are regarded as the internal descriptions provided by the sensorimotor system in response to (recent) past sensations and actions. They are assumed to be correct descriptions of the actual experience, although possibly quite abstracted and incomplete. The sensorimotor input for the traveler, from which the Tour model constructs a cognitive map, is modeled as an alternating sequence of views and actions: V_0, A_0, V_1, A_1, V_2,...V_{n-1}, A_{n-1}, V_n. This discrete simplification of the continuous spatial world makes the research more tractable, of course, but it is also justified by observations that people frequently recall only those places at which decisions are made and not the continuous paths between them (Lynch 1960).

The Tour model includes only two actions, each of which can include optional descriptions of quantitative parameters: (1) (Turn α), where α describes the amount of rotation, and (2) (Travel δ, where δ describes the distance of travel. The extent to which these descriptions of quantities are used and whether they are treated as qualitative or quantitative descriptions varies with the level of the semantic hierarchy.

Procedural Behaviors

The purpose of procedural behaviors is to represent a description of sensorimotor experience sufficient to allow the traveler to follow a previously experienced route through the environment. This function is nontrivial if

purpose	sensorimotor schemas	retrieval use
describe or follow route	$\langle$*goal, situation, action, result*$\rangle$	$(G, V) \rightarrow (A, V')$
travel within the environment	$\langle$*goal, situation, action,* —$\rangle$	$(G, V) \rightarrow A$
record of experience	$\langle$ —, *situation, action, result*$\rangle$	$(V, A) \rightarrow V'$
recognition of familiar place	$\langle$ —, *situation,* —, —$\rangle$	$V \rightarrow$ **true** $\mid$ **false**

Table 2. Sensorimotor Schema Performance under Degradation.

the representation must be robust in the face of resource limitations when information sensed, stored, or retrieved can be incomplete.

The basic element of procedural behaviors is a production-like schema: A *sensorimotor schema* is a 4-tuple <goal, situation, action, result>, where action is an action description, and goal, situation, and result are all views.

The procedural interpretation of a sensorimotor schema is "When attempting to reach goal, if the current view is situation, do action and expect result." The declarative interpretation is "If the current view is situation, then doing action will produce result."

A procedural description of a route is a collection of sensorimotor schemas describing the fragments of the route. For example, suppose that the traveler on the way to goal Vn has the following sensorimotor experience: V_0, A_0, V_1, A_1, V_2,...V_{n-1}, A_{n-1},V_n.

A complete procedural route description would be the following set of sensorimotor schemas: {<V_n, V_i, A_i, V_{i+1}> | i = 0,...n-1}.

A route description can be followed as a procedure for navigating in the environment based on two factors: 1. There must be a complete set of schemas at least at the <goal, situation, action, —> level. 2. The action descriptions must be sufficiently well specified, given the environment, to initiate the appropriate action. (Termination can be environment driven by activating the next schema, with obvious risk of overshoot.)

Even without being complete enough to allow the route-following procedure to be successfully executed, the route description in terms of procedural behaviors is useful for the recognition of landmarks and as a basis for subsequent assimilation. Procedural behaviors are robust because many purposes can be effectively served by collections of incompletely specified sensorimotor schemas (see table 2). The performance of a collection of sensorimotor schemas degrades gracefully as the schemas become less complete.

While acquiring a route description during actual experience traveling in the environment, working memory holds the current view and, likely, the current action. Once an action is initiated, a time interval occurs before the result is observed as the new current view. During this time, distractions can occur, causing some or all of the contents of working memory to be lost. It is relatively easy to create a sensorimotor schema and fill its situation and action components. However, the goal and result components depend on information persisting in working memory over a significant time interval; so, they might frequently be left empty.

The first two uses of sensorimotor schemas in table 2 illustrate the graceful degradation of performance that occurs when resource limitations prevent the schema from being filled. A set of complete schemas allows the route to be followed or described, using the result component as the forward pointer in a linked list to the schema describing the next action and its result. If the result component is missing, the route can still be followed but only within the physical environment, where the environment will produce the result of the action and allow retrieval of the next schema. This level of performance appears to account for a common state of incomplete knowledge of a route, frequently described by the following: "I could take you there, but I can't tell you how."

Preservation of one's goal in working memory during a sequence of actions has obvious survival value. Nonetheless, it is not uncommon to experience capture errors (Norman 1981), when inattention leads a familiar procedure to capture the execution of a less familiar procedure. For example, I might be driving to do an errand on a Saturday morning and inattentively discover myself arriving at work. This phenomenon has a plausible explanation in terms of the interaction between sensorimotor schemas and working memory. If the current goal is dropped from working memory, a given situation can evoke several <??, situation, action, ??> schemas. With no reason to select among them, the most common is selected and executed.

Within procedural behaviors, route finding can take place by retrieving known routes or searching for concatenations of known routes. Although this method is adequate for many purposes, it does not provide any capability to search for novel routes, find shortcuts to optimize familiar routes, or cope without guidance with unknown environments.

The Topological Map

Where the procedural map is defined in terms of the egocentric sensorimotor experience of the traveler, the topological map is defined in terms of the fixed features of the environment—places, paths, and regions—and topological relations among them, such as connectivity, order, and containment.

Once the procedural map is sufficiently complete—that is, there is a stream of sensorimotor schemas of the form < —, situation, action, result>—it becomes possible to assimilate the information in sensorimotor schemas into the topological map. Because it doesn't matter whether such a schema comes from a goal-indexed route description or is simply a record of experience while wandering around, we omit the goal component from our schema notation.

The different topological relations in the map are defined in terms of sensorimotor experience in such a way that the computational structures representing knowledge of these relations can be accumulated incrementally from the stream of observations (see figure 1). Two levels of the topological map exist in the Tour model: (1) a topological network of places and paths and (2) the containment and boundary relations of places and paths with regions.

The Network of Places and Paths. To build a topological map consisting of a network of places and paths, we need only consider actions to be described as "turn" or "travel," ignoring any associated description of magnitude. A turn action leaves the traveler at the same place, even though it might change the current view. A place is identified with the set of views obtainable at it. The relation at (view, place) indicates that view is associated with place:

$<V_1,\text{Turn},V_2> \equiv \exists!P[\text{at}(V_1,P) \wedge \text{at}(V_2,P)]$

Because the place associated with a given view is unique, we can define the function place(V) = place(V') = P. (In the real world, similar or identical views can be associated with different places. A generalization to handle such a situation is not difficult but is beyond the scope of this discussion.)

Similarly, a travel action does not leave the traveler at the same place but on the same path. A *path* is a topologically one-dimensional subset of the environment. If a view lies on a path, we say path = path(view). When a view is on a path, we define the function direction(view,path) -> {+1,-1}, which discriminates between the two directions one might be facing along a path. The connectivity between a place and a path is expressed by the relation on(place,path):

$<V_1,\text{Travel},V_2>$
$\quad\equiv \text{place}(V_1) \neq \text{place}(V_2)$
$\quad\equiv \exists!\text{path}[\text{path} = \text{path}(V_1) = \text{path}(V_2)]$
$\quad\text{on}(\text{place}(V_1), \text{path}(V_1))$
$\qquad\wedge\ \text{on}(\text{place}(V_2), \text{path}(V_2))$
$\quad\text{direction}(V_1,\text{path}(V_1))$
$\qquad= \text{direction}(V_2,\text{path}(V_2))$

We can induce a (partial) order relation on the places lying on each path by defining the function order(place1,place2,path) → {+1,-1,nil},

$<V_1,\text{Travel},V_2> \equiv$
$\quad\text{order}(\text{place}(V_1), \text{place}(V_2), \text{path}(V_1))$
$\qquad= \text{direction}(V_1,\text{path}(V_1)).$

These relations can be naturally translated to rules that respond to a stream of <V,A,V'> schemas and incrementally build a collection of places and paths, asserting connectivity relations (at, on, place, and path), and order relations (direction and order) (see figure 1).

These relations give us a topological network of places and paths which can be used to find novel routes among places using the usual graph-search algorithms. The collection of known <V,A,V'> schemas could also be treated as a graph and searched but at a finer granularity and restricted to schemas describing experienced actions, resulting in less efficiency and less success at finding routes.

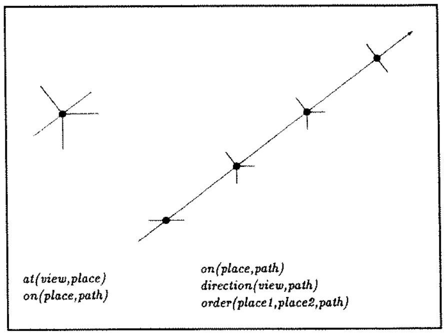


Figure 1. Places and Paths Form a Network of Topological Relations.

Regions, Boundaries, and Containment. As we saw, a path defines an order relation over the places on the path. It also defines a binary classification over the places not on the path: those to the right and those to the left of the path (see figure 2a).

A *boundary* is a sequence of one or more directed paths. Here, we restrict our attention to a single path directed in its +1 direction, but the concept generalizes to sequences of segments, including simple closed curves. A *region* is a set of places; a *boundary region* is the set of places defined to be on one side of a boundary. A path is associated with two boundary regions: right(path) and left(path).

To extract boundary region information from sensorimotor experience and the procedural map, we need a slightly more detailed description of the turn action, including a qualitative description of magnitude: (a) "Turn 0" indicates do nothing, (b) "Turn Around" indicates face the opposite direction along the same path, (c) "Turn Right" indicates turn but not as far as a turn around, and (d) "Turn Left" indicates turn but not as far as a turn around.

Now, suppose we observe a sequence of sensorimotor schemas matching the following pattern:

$<V_1,\text{Travel},V_2>$
$<V_2,\text{Turn Right},V_3>$
$<V_3,\text{Travel},V_4>$

The rules for the place-path network would define the following places and their relations with the observed views (see figure 2b):

$P_1 = \textit{place}(V_1)$
$P_2 = \textit{place}(V_2) = \textit{place}(V_3)$
$P_3 = \textit{place}(V_4)$
$\textit{path}_1 = \textit{path}(V_1) = \text{path}(V_2)$
$\textit{path}_2 = \textit{path}(V_3) = \textit{path}(V_4)$
$\textit{direction}(V_1,\textit{path}_1) = \textit{direction}(V_2,\textit{path}_1)$
$\textit{direction}(V_3,\textit{path}_2) = \textit{direction}(V_4,\textit{path}_2)$

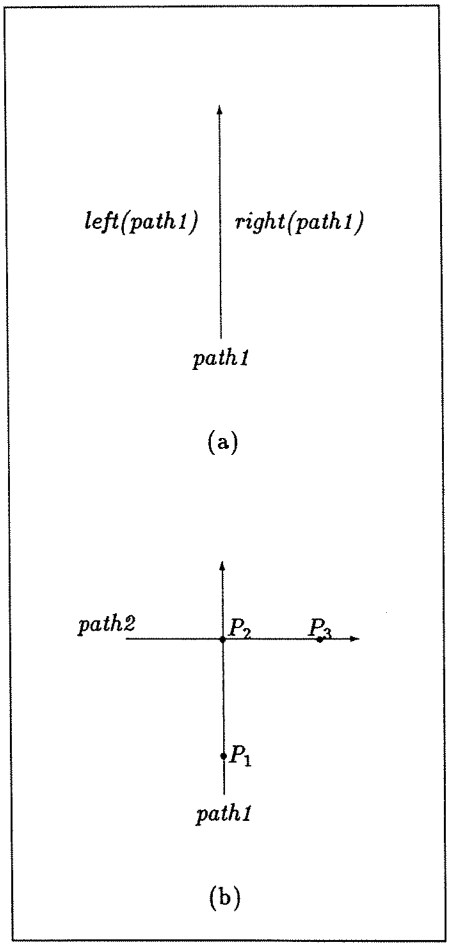


Figure 2.
Boundary regions.
(a) Each Path Divides Space into Right and Left Boundary Regions;
(b) Boundary Relations between Places and Paths Can Be Acquired Incrementally during Travel.

Membership in the boundary regions is then defined by

$$\mathit{direction}(V_2,\mathit{path}_1) = +1 \Rightarrow P_3 \in \mathit{right}(\mathit{path}_1)$$
$$\mathit{direction}(V_2,\mathit{path}_1) = -1 \Rightarrow P_3 \in \mathit{left}(\mathit{path}_1)$$

A similar rule can place P_1 to one side or the other of path_2.

Clearly, for each place and path, the place lies either on the path or in one of its boundary regions. The actual exploration history of the traveler determines which boundary relations are actually observed and assimilated. Once a sufficient number of boundary relations have been accumulated, they provide a useful topological route-finding heuristic. To illustrate, an example boundary heuristic is as follows: To find a route from A to B, if $\exists$ path such that A $\in$ right(path) and B $\in$ left(path), look for routes from A to path and from path to B.

Chase (1982) and others have observed that many human cognitive maps are organized around a "skeleton map" of major streets, within which most problem-solving takes place, with final links from the original start and goal points to the nearest point on the skeleton. The statistical frequency with which routes follow a minority of streets should give these streets a preponderance of the boundary relations and thus determine which streets are treated as part of the skeleton for a given problem.

The Metric Map

Metric descriptions can be added to the map if we consider the quantitative components describing the magnitude of actions: (Turn α), where α describes the amount of rotation, and (b) (Travel δ), where δ describes the distance of travel. For the metric map, the Tour treats these descriptions of magnitude as numbers or intervals.

The metric map extracts this quantitative information acquired during travel and assimilates it into two levels of metric description: (1) local geometry and (2) orientation frames. *Local geometry* includes the locally defined headings of paths radiating from a place and the position of places within a locally defined one-dimensional coordinate system of each path. *Orientation frames* include sets of places and paths whose local geometry has a known relation with a common coordinate frame of reference, supporting the use of two-dimensional vectors to define relative position and orientation.

Local Geometry. The conceptual prerequisite for the local geometry level of the metric map is the network level of the topological map, not the topological map as a whole. The directions radiating from a given place are associated with views. If we can determine the relative angles between views from the magnitudes of particular turn actions, we can incrementally associate an absolute heading with each view, using coordinates locally meaningful at each place:

$$<V_1,(\text{Turn }\alpha), V_2> \equiv \text{localheading}((V_2),\text{place}(V_2)) = \alpha + \text{localheading}((V_1),\text{place}(V_1))$$

The assignment of local headings can take place incrementally if the first request for a local heading of a view at a place assigns this view the arbitrary heading of zero. This association of local headings with views automatically induces an association of local headings with directed paths at each place:

$$\text{localheading}(\text{path}(V), \text{direction}(V,\text{path}(V)),\text{place}(V)) = \text{localheading}(V,\text{place}(V))\,.$$

An analogous relation allows us to assimilate the relative distance between two places and incrementally create a one-dimesional coordinate system associated with each path, giving an absolute one-dimensional coordinate to each place:

$$<V_1,(\text{Travel }\delta), V_2> \equiv \text{localposition}(\text{place}(V_2),\text{path}(V_2)) = \text{localposition}(\text{place}(V_1),\text{path}(V_1)) + \delta * \text{direction}(V_1,\text{path}(V_1))$$

(Recall that $\text{path}(V_1) = \text{path}(V_2)$.)

From local one-dimensional coordinates, we can compute relative distances between places independently of having actually traveled between these two places:

$$\text{pathdistance}(\text{place1},\text{place2},\text{path}) = |\text{localposition}(\text{place1},\text{path}) - \text{localposition}(\text{place2},\text{path})|\,.$$

Local geometry information is useful in problem solving, making it possible to estimate the lengths of routes.

Orientation Frames. It is useful for route finding to be able to estimate the distance and direction from one place to another by integrating the distances and turns along the segments of a route leading from one to the other. This integration requires that relative distances and directions be in a common frame of reference to allow vector addition. Unfortunately, in the local geometry, headings are strictly local to individual places; so, comparisons are impossible.

However, a global orientation frame, such as compass directions, can serve as a common frame of reference for multiple local coordinate systems at individual places. Globally meaningful headings at heading(V,OF) can be defined with respect to an orientation frame OF, with the constraint that the headings are compatible with observed turn information:

$$<V,(\text{Turn }\alpha),V'> \equiv \forall \text{OF}[\text{heading}(V_2,\text{OF}) = \alpha + \text{heading}(V_1,\text{OF})]$$

Note that orientation frames need not be genuinely global frames of reference such as the compass directions. Frequently, an orientation frame is associated with a region of a city that has a regular rectangular pattern of streets, such as Back Bay in Boston, Mass., and ceases to apply at the boundaries of this region. In such cases, the regional orientation frame can be created by local propagation along paths, as neighboring places bring their local coordinate systems into correspondence.

This type of propagation is most likely to take place along straight streets. Incorporating a feature developed for the NX robot, we can generalize the quantitative information in a travel action description to include a description of the net change in heading $\Delta\theta$ experienced during the following action: (Travel δ $\Delta\theta$). For a straight street, of course, $\Delta\theta = 0$ This information imposes a second constraint source on the headings asserted by an orientation frame:

$$<V,(\text{Travel }\delta\,\Delta\theta),V_2 \equiv \forall \text{OF}[\text{heading}(V_2,\text{OF}) = \Delta\theta + \text{heading}(V_1,\text{OF})]$$

It is natural here to use an interval representation for incomplete knowledge of $\Delta\theta$. Travel along a straight street would yield $\Delta\theta$ = [0,0], but a twisty mountain road might give $\Delta\theta$ = [-180, +180].

Vector Addition. It would be useful to be able to manipulate two-dimensional relative position vectors, as people clearly do in some cases. However, the extent of human capabilities along these lines is not at all clear; so, the Tour model takes no firm position on this topic, although several vector addition schemes have been implemented at various times. The problem is to define the functions distance (place1,place2) and direction (place1, place2,OF) whose values represent the vector from place1 to place2 within the orientation frame OF and to provide axioms corresponding to appropriate assimilation rules.

An assimilation process should yield two-dimensional distance and direction relations from experience such as

$<V_1,(\text{Travel } \delta_1\, \Delta\theta_1),V_2>$
$<V_2,(\text{Turn } \alpha),\ V_3>$
$<V_3,(\text{Travel } \delta_2\, \Delta\theta_2),\ V_4>$

Where $\Delta\theta_1 = \Delta\theta_2 \neq 0$, this vector addition is straightforward. Where $\Delta\theta_1, \Delta\theta_2 \neq 0$, the quantitative description of travel requires more detail to capture the net straight-line distance covered by a curved path. It remains unclear whether this kind of knowledge is realistic for a person to accumulate, how the knowledge is inferred, how it is represented, and what its states of partial knowledge are.

Psychological studies of behavior related to vector addition have aroused considerable discussion about the existence of special-purpose spatial analog computational mechanisms in the brain or, at least, in the mind. Part of this discussion was sparked by the dramatic results of Shepard and Metzler (1971) on mental rotation, leading to widespread interest in the properties of mental imagery (see Kosslyn 1980). Another part of this discussion arises from interest in the properties of the hippocampus, a portion of the brain with regular anatomical structures that is closely tied with spatial reasoning and working memory, as shown by studies of cognitive deficits resulting from brain injuries (O'Keefe and Nadel 1978).

Distorted Cognitive Maps. The values of heading(V,OF) are underconstrained by local observations in turn and travel actions. Not surprisingly, then, it is widely observed that human cognitive maps have "rubber sheet" distortions when compared with the actual environment, preserving topological and local metric properties but possibly grossly distorting global metric distance and direction relations (Lynch 1960).

As these relations are implemented with incremental assimilation rules, a number of heuristics, especially for dealing with quantitative observations, can predictably lead to these distortions: (1) In (Turn α), if α is approximately equal to a right angle, assume it is exactly a right angle. (2) In (Travel $\delta\, \Delta\theta$), assume that $\Delta\theta = 0$. (3) Given $<V_1,(\text{Travel } \delta\, \Delta\theta),\ V_2>$, assume that distance(place(V_1),place(V_2)) = δ

The Qualnav Model

The Tour model has a built-in bias toward environments that have approximate network-like structures such as urban networks of places connected by streets. Qualnav takes a substantially different point of view, considering the environment as open, possibly mountainous terrain with significant perceptual events, called

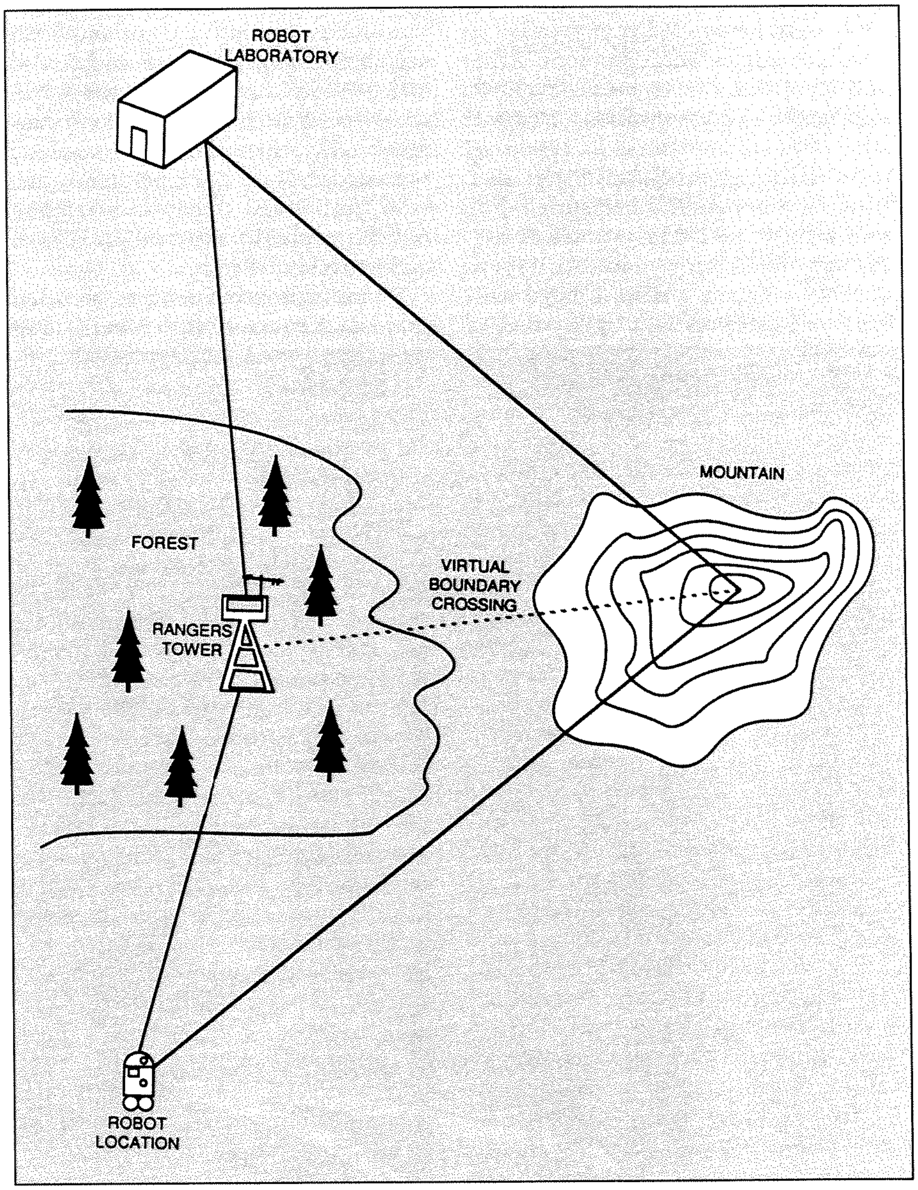


Figure 3. Cross-Country Inference Example.

landmarks, scattered throughout this two-dimensional space. The theory also extends to three dimensions. The spatial semantic hierarchy is successful in providing a robust structure for exploration, mapping, and navigation.

Qualnav is a multilevel representation theory of large-scale space based on the observation and reacquisition of distinctive visual events, that is, *landmarks*. The representation provides the theoretical foundations for visual memory databases and path planning and guidance algorithms, including coordinate-free, topological representation of relative spatial location and smoothly integrating available metric knowledge of relative or absolute angles and distances. Rules and algorithms have been developed that under the assumption of correct association of landmarks on reacquisition (although not assuming landmarks are necessarily reacquired) provide a robot with navigation and execution capability. The ability to deduce or update a map of large-scale space a posteriori is a by-product of the inference process. In order to demonstrate our claims, we built a qualitative navigation simulator, called the Qualnav model, that provides a software laboratory for experimenting with spatial relationships in visual memory and their relationship to path planning and execution.

Robot navigation and guidance has traditionally been quantitative, relying on accurate knowledge of distances, directions, paths traveled, and similar metric data to get from place to place. Existing robot navigation techniques include triangulation (Matthies and Shafer 1986), ranging sensors (Hebert and Kanade 1985), auto-focus capability (Pentland 1985), stereo techniques (Lucas and Kanade 1984; Eastman and Waxman 1985), dead reckoning, inertial navigation, geo-satellite location, correspondence of map data with the robot's own location, and local obstacle avoidance techniques (Moravec 1980; Chatila and Laumond 1985).

These approaches tend to be brittle (Bajcsy, Krotkov, and Mintz 1986; Brooks 1987), accumulate error (Smith and Cheesesman 1985), are limited by the range of an active sensor, depend on accurate measurement of distance and direction perceived or traveled, and are nonperceptual or only utilize weak perceptual models. Furthermore, these theories are largely concerned with the problem of measurement and do not centrally address issues of map or visual memory and the use of this memory for inference in vision-based navigation and guidance.

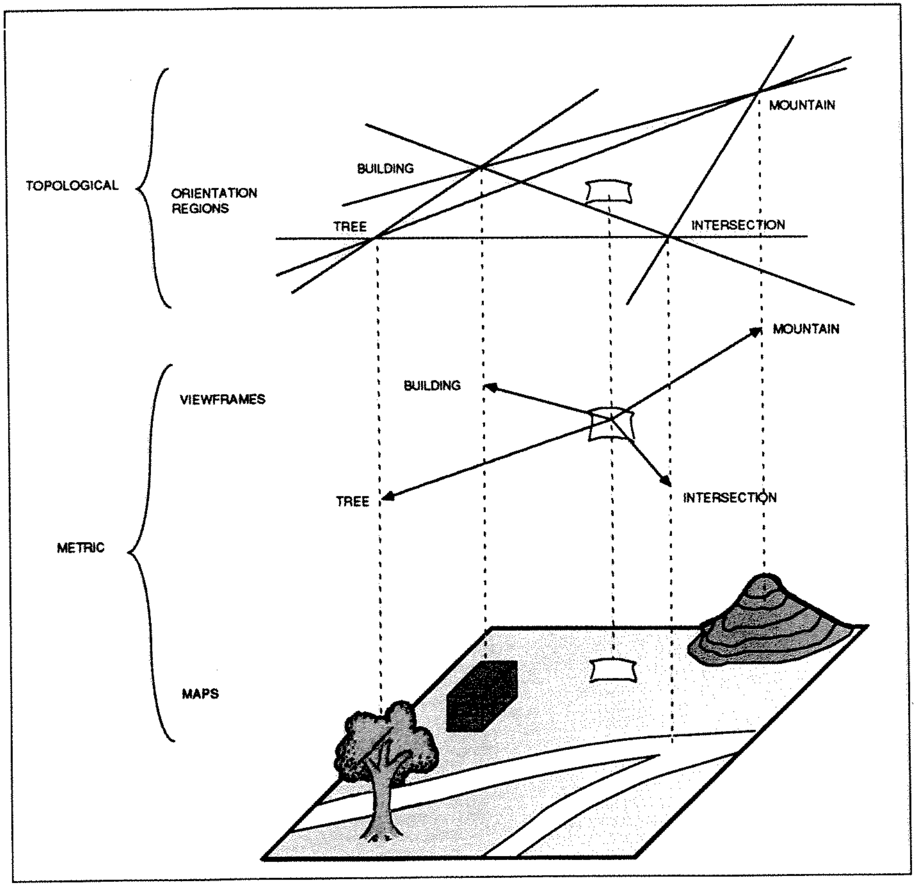


Figure 4. Multiple Levels of Spatial Representation.

The Qualnav model provides a computable theory that integrates qualitative, topological representations of large-scale space with quantitative, metric ones. We contrast traditional robotic spatial reasoning with the qualitative techniques in the following example of a cross-country scenario of figure 3 drawn from the Qualnav model. Here, our solar-powered robot has wandered the long way around the forest (off the left side of the figure) without a map and wants to get home before the sun goes down. The robot sights and recognizes the forest ranger's watch tower and the mountain.

Searching its memory of landmarks viewable from the robot laboratory (the goal location), the robot knows that a watch tower and a mountain can be viewed from there. A traditional triangulation approach to path planning would require that our robot move in a straight line, keeping track of the "crow's flight" distance between its start and end points in order to form a base for a triangle relating its locations with the sighted landmarks. However, the irregular field terrain makes this a difficult and inaccurate task because the robot must constantly avoid local rocks, bushes, ruts, hills, and so on, to traverse the ground.

However, the Qualnav model perceives a hypothetical line joining the landmark pair of watchtower-mountain as a virtual boundary between itself and the robot laboratory. In general, if we draw a line between two (point) landmarks and project this line onto the (possibly not flat) surface of the ground, then this line divides the earth into two distinct regions. If we can observe the landmarks, we can observe which side of this line we are on. The virtual boundary created by associating two observable landmarks together thus divides space over the region in which both landmarks are visible. We call these landmark-pair boundaries (LPBs) and denote LPB constructed from the landmarks L_1 and L_2 by LPB(L_1,L_2).

The robot can draw the inference of which side of the LPB it is on because the watchtower is on its left and the mountain on its right, whereas it recalls that this relationship was reversed when it viewed these landmarks from the robot laboratory. The robot now sets a goal to cross this LPB and executes it by a control feedback procedure depending on continuous sighting and tracking of the landmarks. Based on vision, this algorithm is insensitive to local obstacle avoidance and does not require any numeric estimates of distances to landmarks. Approach to the boundary is assured by measuring the angle between the landmarks; when it reaches 180 degrees, the robot knows it is crossing the LPB. In our example, the robot can now sight and head directly toward the laboratory. Generally, this reasoning is applied recursively to achieve hierarchical, sensory-based, qualitative navigation and guidance.

The robust, qualitative properties and formal mathematic basis of this representation and inference processes are suggestive of the navigations and guidance behavior in animals and humans (Schone 1984). However, we make no claims of biological foundations for this approach.

In the following paragraphs, we discuss the sensorimotor assumptions of the Qualnav model and present the procedural (tracking) behavior of the robot simulator. We then develop the mathematic theory of the viewframe (metric map) representation and the orientation region (topological map) representation. A *viewframe* is a data structure that encodes the observable landmark information in a stationary panorama. An *orientation region* is a data structure representing the area on the ground defined by the set of LPBs in a viewframe. The relationship between the levels of mapping are pictured in figure 4. For each of the viewframe and orientation region maps, we show how these metric and topological concepts relate to visual memory to produce navigation and guidance capability.

Sensorimotor Interaction with the Environment

The Qualnav model simulates a land-roving robot equipped with an omniview visual sensor that produces a continuous, two-dimensional, 360-degree image of the surrounding environment. The Qualnav simulation of the world is an elevation grid that the robot moves over. Vision of landmarks is simulated by marking or numbering discrete points in the elevation grid.

Research by cognitive psychologists (Kozlowski and Bryant 1977; Shepard and Metzler 1971; Pylyshyn 1984) and zoologists (Schone 1984) has clearly demonstrated that humans and animals record distinctive visual landmarks and use the structure inherent in local and temporal relationships between landmarks relative to observer paths of motion to predict and identify places in the world and plan and execute paths between locations. Recent work to close the gap between memory structure and neurobiology for spatial understanding and perception is presented in Foreman and Stevens (1987). Even these preliminary advances strongly support the notions of landmark-based understanding of local environments. Furthermore, humans and animals perform navigation and guidance tasks quite reliably with poor range estimates and coarse angular information.

Ideal landmarks for navigation and map building are uniquely distinguishable points that are visible from anywhere with precisely determined range. The real world, of course, consists of objects that occlude each other and look different under varying viewing conditions. Significant perceptual and cognitive inference is required to recognize the same objects from different places and environmental conditions. For navigation and map building in particular, the changes in object appearance can be considerable because the location-refining power of landmarks depends on relating landmarks from highly separated points of observation.

The Qualnav simulator currently assumes that a landmark will be perceived whenever the robot has a line of sight to it. In the current implementation, all landmark sightings are correctly recognized as having been seen or not seen before. Error in angle and range estimates are randomized around tunable parameters in the simulator. For example, we can set the worst-case range error to 500 percent, with range values to landmarks being randomly generated with 10 percent to 500 percent error. Work is in progress to extend the Qualnav theory to handle mismatched or unobserved landmarks.

Procedural Behaviors

In the Qualnav model, *route headings* are world states that can be created by a robotic action, namely, that of following the heading direction specifier. While a heading is being executed, the vision system builds metric and topological representations of the environment it is passing through. These representations must be sufficient to support future inferencing to plan and execute a path back through this environment or through geographic regions that share common landmarks in view. Termination conditions correspond to computable changes in our location in space, triggering additional spatial inference processes if we have not reached our destination goal.

We define different types of headings and the associated termination conditions. We use visual memory and heading structures to create algorithms that can guide a robot through the world based on visual information. A *heading* consists of a type, destination goal, direction function, and termination criteria.

The type of heading specifies the coordinate system that the direction conditions are computed in. An absolute heading type corresponds to an absolute coordinate system. An absolute heading can be induced by a correspondence of sensor position to an a priori map or grid data from an inertial navigation system, geo-satellite location, a dead reckoning from a known initial position, and so on. A view-frame heading type refers to headings computed between local coordinate systems that share com-

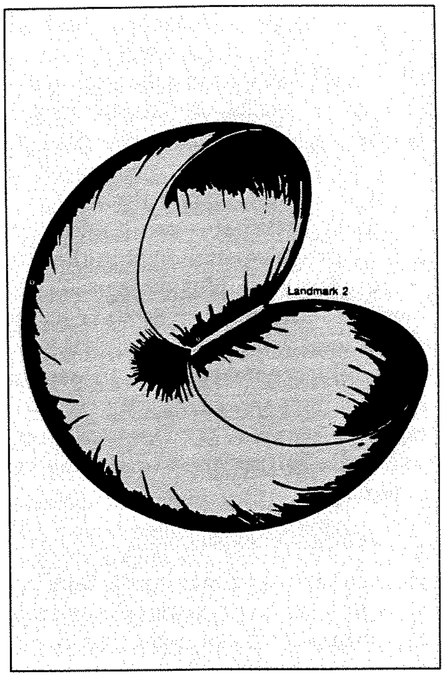


Figure 5.
Constant Angle Toroid.

HEADING TYPE / FIELD	ABSOLUTE	VIEWFRAME	ORIENTATION
destination-goal	polygon expressed in absolute coordinate system	viewframe	orientation region
direction function	distance of current estimated point location to destination polygon	if sensor-centered vector-range set then, difference of current location to vector destination relative to point where heading set or maintain visibility of goal landmarks and hill-climb on relative angles	track until LPB is crossed
termination criteria	estimated error in current location exceeds threshold or goal achieved	if range set then estimated error in relative location exceeds threshold or lose sight of landmarks in goal viewframe or hill climbing to relative angles fails or goal achieved	lose sight of goal LPB landmarks or must reverse a goal LPB to continue or goal achieved

Table 3. Heading Specifications.

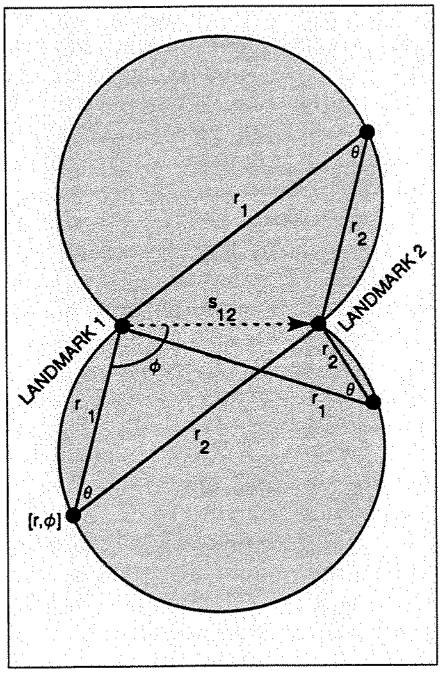


Figure 6 .
Constant Angle Circular Arcs.

mon visible landmarks. An orientation heading type is a heading that is continuously updated based on observed relationships between pairs of landmarks. View-frame and orientation headings are explained in more detail in the following subsections.

Destination goals are descriptions of places that the heading is intended to point the robot or vision system platform toward. A destination goal can be specified, in order of increasingly topological (more qualitative) representation, as a set of absolute (for example, universal transverse mercator [UTM]) world coordinates, a view-frame localization, an orientation region, or a set of (simultaneously visible) landmarks.

Direction functions are functions that accept run-time data at either metric or topological levels of representation and return true if the heading is being maintained and false otherwise. Direction functions are essentially predicates except that they can have side effects, such as updating heading error parameters. Direction functions for absolute headings simply compare the desired heading vector against that returned from direct sensor readings. Directions for view-frame headings are given as relative angles between the heading vector in the sensor-based coordinate system and the observed landmarks. Orientation directions specify a set of simultaneously true conditions indicating passage to the right, left, or between pairs of landmarks.

Termination criteria are run-time computable conditions that indicate that if the heading continues to be maintained, its direction function can no longer return true. This situation can occur because the heading has been fulfilled, meaning that we have reached the destination goal implicit in the direction function. For example, termination occurs if we are at the desired absolute world-coordinate location specified in an absolute heading, we recognize the set of landmarks corresponding to a view-frame heading destination, or we cross the LPBs given in the direction function of an orientation heading. A heading fails if we accumulate too much error in an absolute coordinate system tracking scheme or we lose sight of the set of landmarks required to maintain a view-based or orientation-region heading. Termination criteria can also include feedback conditions from modules outside the vision system, such as a path-planning module that reasons about obstacles, traversability of the ground surface, strategic concealment, and so on. Heading types, destination goals, direction functions, and termination criteria are summarized in table 3.

The top-level loop for landmark-based path planning and following is to (1) determine a destination goal, (2) compute and select a current heading, and (3) execute the heading while building an environmental representation. The destination goals are typically determined recursively, implementing a recursive goal-decomposition approach to perceptual path planning.

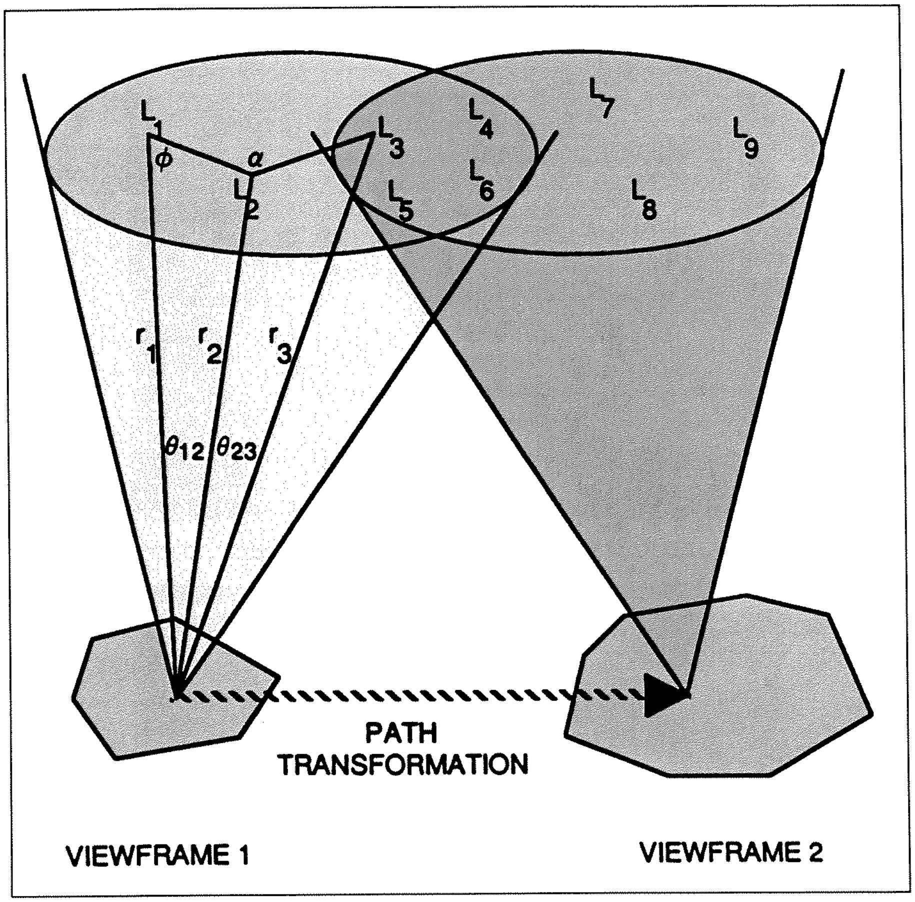


Figure 7. Viewframe Heading.

The Metric Map

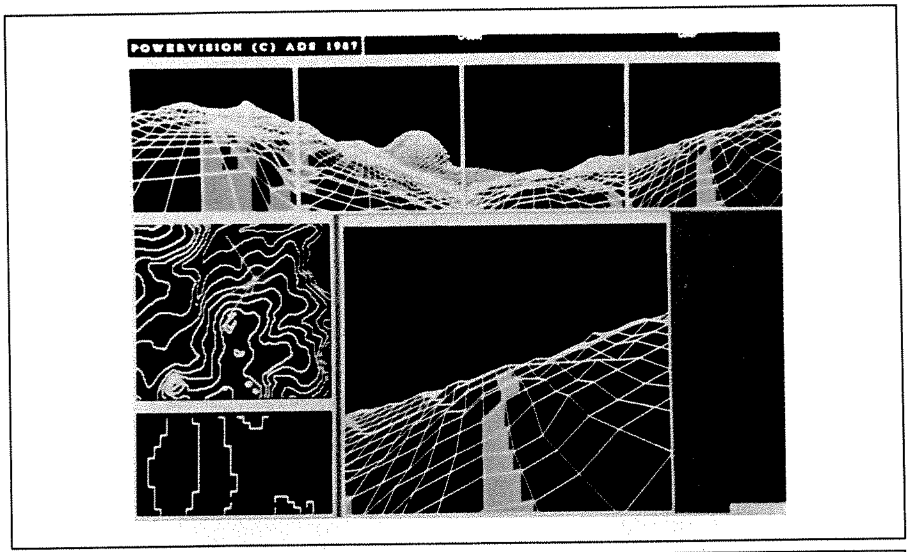
Robot memory consists of sequences or paths of viewframes and relationships between viewframes that have landmarks in common. Because a viewframe encodes observable landmark information in a stationary panorama, we assume the sensor platform is stationary long enough for the sensor to pan up to 360 degrees, tilt up to 90 degrees (or to use an omnidirectional sensor [Cao, Oh, and Hall 1986]), recognize landmarks in its field of view, or buffer imagery and recognize landmarks while in motion.

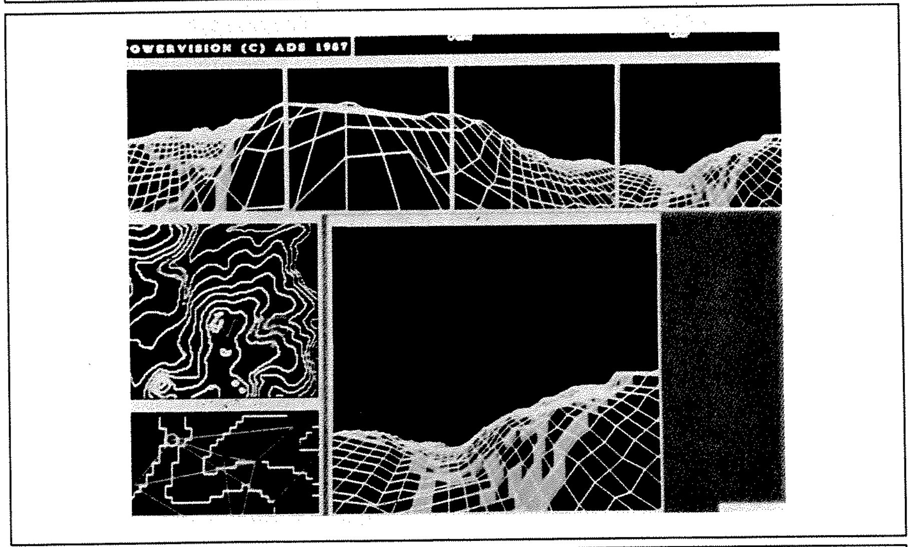
We can pan from left to right, recognizing landmarks, L_i, and storing the solid angles between landmarks in order, denoting the angle between the ith and jth landmarks by Ang_{ij}. The basic view-frame data are these two ordered lists: $(L_1,L_2,...)$ and $(Ang_{12},Ang_{23},...)$. The relative angular displacement between any two landmarks can be computed from this basic list. An error in computing these relative angles is at least as great as the resolution of the vision system and can include cumulative pan-tilt error, angular ambiguity in landmark point localization, or other error sources. The angular error is measured by e_{ij} between landmarks i and j. Finally, range estimates are required for landmarks recorded in viewframes. These estimates can be arbitrarily coarse but finite. We only require that the true range lie between the bounds specified for the estimate. We denote the range interval associated with landmark L_i by $[r_{i1},r_{i2}]$. We now explain how it is possible to make computable local coordinate systems in space relative to these observed landmarks.

We begin by noting that the set of points in three-space from which we can observe an angle of Θ_{ij} between landmarks L_i and L_j is constrained to a closed toruslike surface; a cutaway of this surface is pictured in figure 5. This toruslike surface is more easily observed in a planar cross-section, where the shape is the figure eight cross-section in figure 6.

Now, if the robot observed two landmarks separated by angle Θ, then it must be on a circle determined by Θ and the distances to the landmarks. Because of angular error, the circle of possible locations is thickened to a band. If more than two landmarks are simultaneously observed, then the robot's actual location must be in the intersection of multiple fattened circles. This intersection is called a *viewframe localization.*

Having formulated the view-frame localizations for two viewframes in the common local coordinate system defined by two landmarks and the sensor, we now have two regions expressed in the same coordinate system. Any (affine linear) transformation that maps one viewframe approximately onto the other can be taken as a heading. For example, we can translate the centroid of the first viewframe to the centroid of the second. This translation defines a viewframe heading as a vector pointing between the viewframes and supplies the vision system with an intuitive notion of "head that away." A viewframe heading is generally not the same as an absolute heading because the points in space that the sensor occupied when the viewframes were collected might not be mapped onto each other by the viewframe heading transformation. Figure 7 illustrates the generic situa-





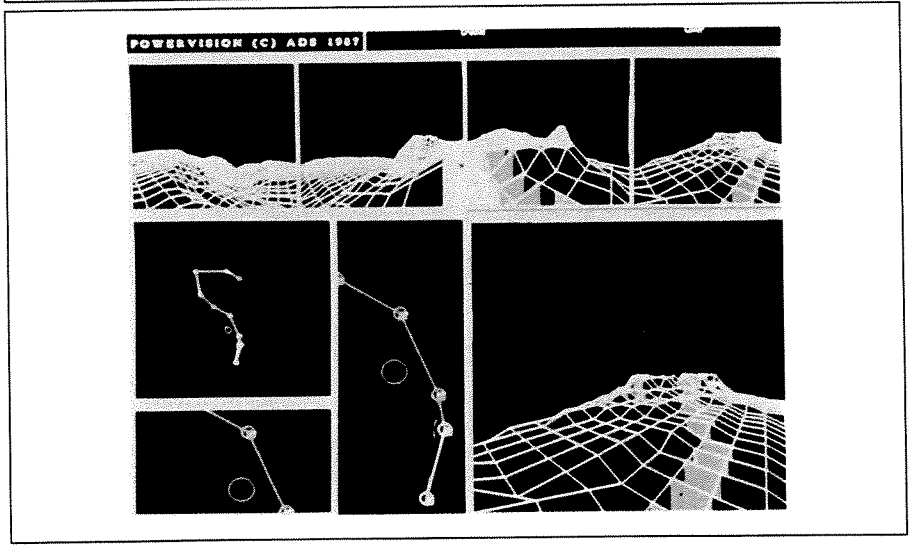


Figure 8 (top). Automatic Viewpath Formation.
Figure 9 (middle).. Linkages in Visual Memory.
Figure 10 (bottom). Initial Plan.

tion in which we can compute a viewframe heading. One viewframe contains the landmarks L_1 to L_6, and the other viewframe contains landmarks L_3 to L_9. Range estimates to L_3, L_4, L_5, and L_6 might be different in the two viewframes. Using the LPB connecting L_3 and L_4, we can assign a local orientation to the vector pointing from L_3 to L_4. This vector, with the implied orientation of the ground plane, defines a local two-dimensional coordinate system.

Inference over viewframes performs path planning and following over a visual memory and, therefore, assumes that viewpaths—that is, the visual memory—have already been collected. If they have not, paths must be planned and followed based on metric data and viewpaths collected in the course of following these paths. A* is used to search along view paths to find a path between start and goal. We show some path planning and execution results in the Qualnav simulator.

The simulator interface defaults to representing a mouse-sensitive menu of options, four three-dimensional-perspective windows capturing the omnidirectional view of the world from the robot's current position, an overhead view of the world, and a zoom window chained to the overhead view. In figure 8, the simulator is also configured with a duplicate, zoomed perspective window of the heading direction.

Because landmarks can be chosen from the perspective display windows, it is especially easy to choose peaks, valleys, ridges, and critical horizon points as landmarks. These distinctive geographic features correspond to naturally occurring landmark choices in cross-country environments.

The menu command to create a new viewpath allows the user to click along an arbitrary path. The simulator then automatically moves along the path, computing and storing the viewframes at each clickpoint, as shown in figure 8. The dark (and dim) blue line corresponds to the user-specified path. The system is midway through calculation of viewframes along the path. The light blue line indicates the current heading along the viewpath.

Building visual memory computes linkages between viewpaths that have view frames with commonly visible landmarks and accounts for a user-specified maximum estimated range between linked viewframes (see figure 9). Here, the dark blue lines are viewpaths in visual memory, and the light blue lines represent linkages between viewpaths built in memory.

Constructing a plan allows the user to specify a start and an end point in the overhead display. The simulator computes a viewframe at these points and searches visual memory for viewframes that shared common landmarks. It uses the recalled range estimates and its current observations (including error) to guess, using viewframe headings, the closest viewframe. A* algorithm is then used over visual memory to get to the viewframe closest to the goal point (computed similarly) (see figure 10).

There is a "step plan" that incrementally executes the following viewframe plan. Figures 11a and 11b show the beginning and end of execution of the plan begun in figure 10. The original plan, created before moving from the start point, is indicated in green. The current plan is in yellow, and the executed path is in white. The robot's current location is shown by a red circle. Its planned heading, that is, the direction it observes its path to be, is indicated by a light blue circle. Dark blue paths (faint in these photos) are viewpaths in memory, and landmarks are shown by red squares. Notice that the executed path is usually much smoother than it is in memory because the robot actually uses the landmarks (with 100 percent range error) to navigate by. Figure 11b highlights the shortcut found by the robot, where its original path took a sharp bend.

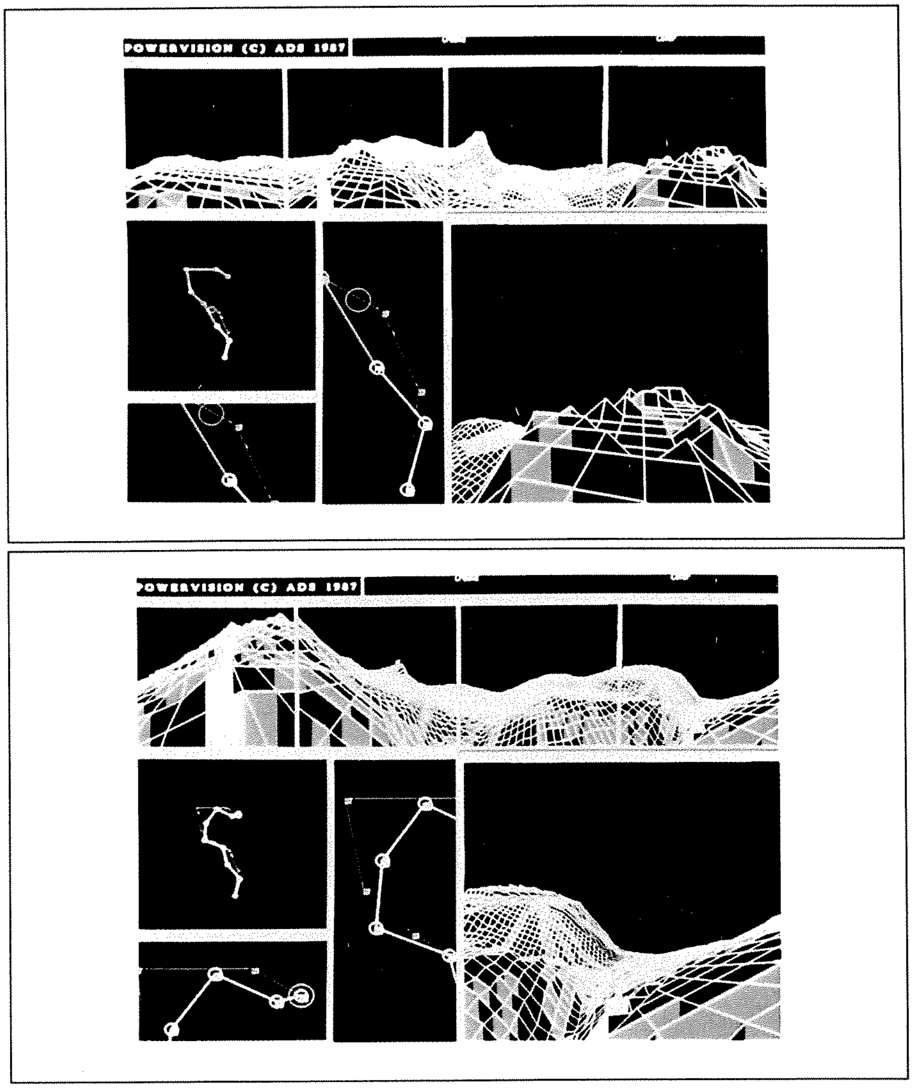


Figure 11. View Fram Plan Execution

The Topological Map

As mentioned earlier, the LPB is constructed from the landmarks L_1 and L_2 by LPB(L_1,L_2). Figure 12a shows LPBs that are implicit in a view frame where landmarks L_1, L_2, L_3, L_4, and L_5 are simultaneously visible. The solid lines are the virtual boundaries created by the landmark pairs. Figure 12a can be misleading in that it seems to imply that the LPBs contain the data to compute the angle-distance geometry of the sensor location relative to the landmarks. Figure 12b shows a representation of the same viewframe. Here, the ranges to the landmarks have been changed, but the ordinal angular relationships between landmarks have not. The angle-distance geometry of our apparent location relative to the landmarks is completely different; however, the topological information, that is, the number of regions, their number of sides and adjacency relations, are preserved.

Roughly speaking, if we observe that landmark L_1 is on our left hand and landmark L_2 is on our right, and the angle from L_1 to L_2 (left to right) is less than radians, then we denote this side of—equivalently, this orientation of—the LPB by (L_1L_2). If we stand on the other side of the boundary, LPB(L_1,L_2), facing the boundary, then L_2 is on our left and L_1 on our right and the angle between them less than π radians; we can denote this orientation, or side, as L_2L_1 (left to right):

$$= \mathrm{sign}(\pi - \Theta_{12}) = \begin{cases} +1 \text{ if } \Theta_{12} < \pi \\ 0 \text{ if } \Theta_{12} = \pi \\ -1 \text{ if } \Theta_{12} > \pi \end{cases}$$

where Θ_{12} is the relative azimuth angle between L_1 and L_2 measured in an arbitrary sensor-centered coordinate system. Here, an orientation of +1 corresponds to the [L_1L_2] side of LPB (L_1,L_2), -1 corresponds to the [L_2L_1] side of LPB(L_1,L_2), and 0 corresponds to being on LPB(L_1,L_2). It is straightfor-

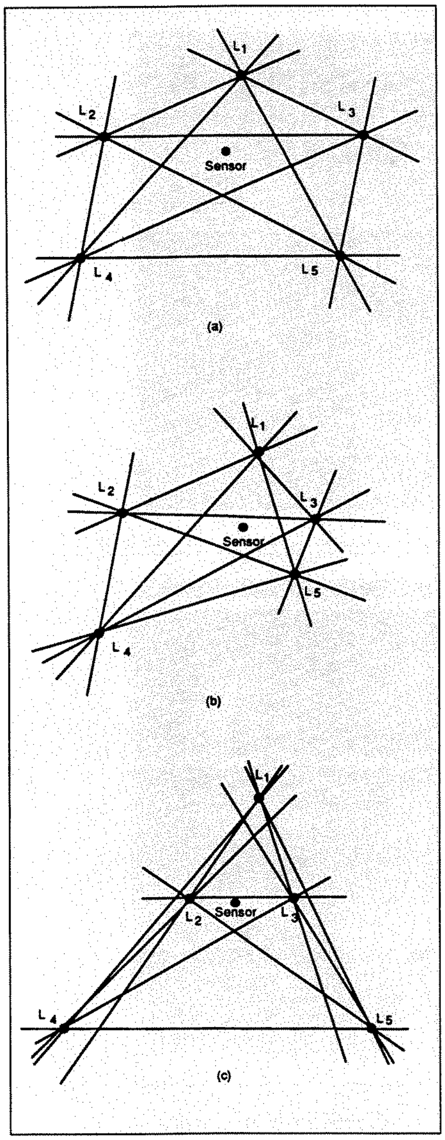


Figure 12.
Viewframe Orientation Regions

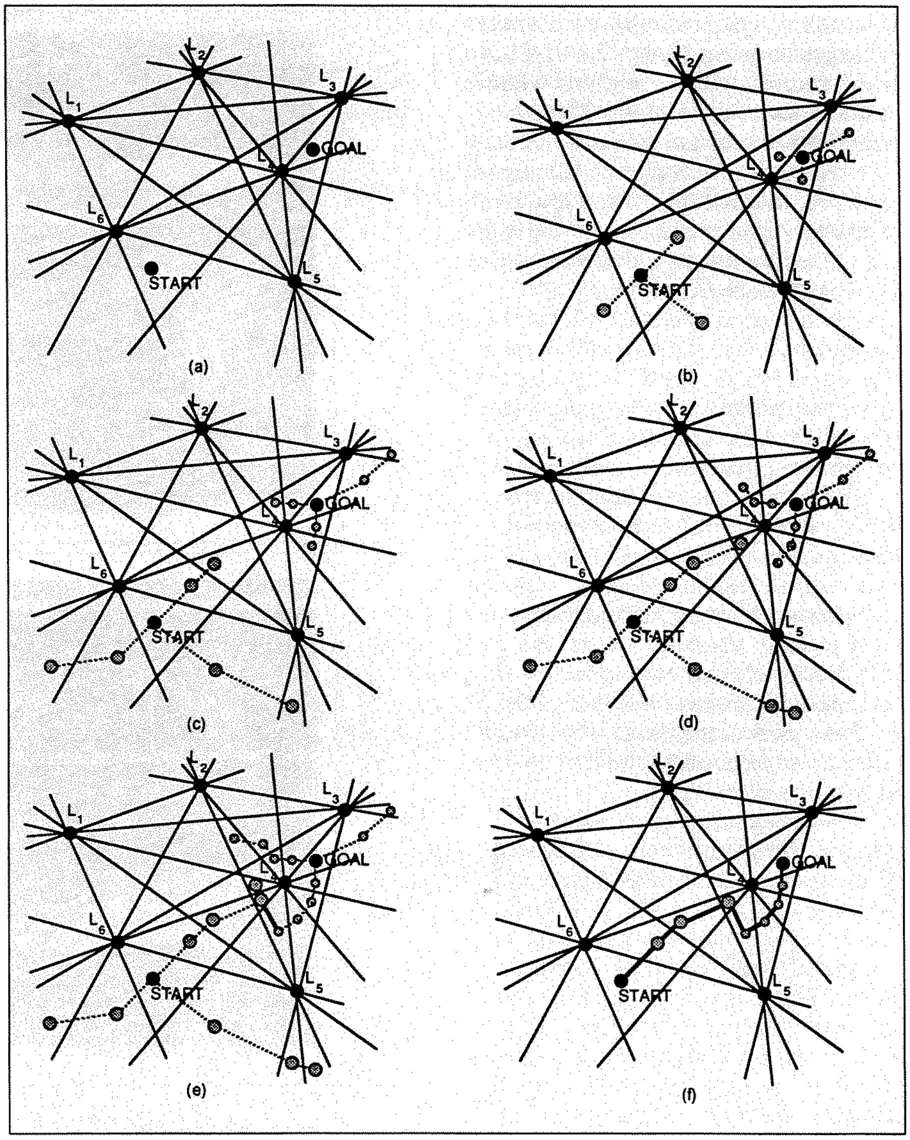


Figure 13.
Qualitative Planning Algorithm.

ward to show that this definition of LPB orientation does not depend on the choice of the sensor-centered coordinate system.

LPBs give rise to a topological division of the ground surface into observable regions of localization, *the orientation regions*. Crossing boundaries between orientation regions leads to a qualitative sense of path planning based on perceptual information.

As a landmark-recognizing vision system moves through large-scale space, it builds a visual memory of the interlocking sequences of orientation regions it has traversed through. Adjacency of orientation regions in visual memory can be determined by sharing a common but opposite orientation LPB. If two regions have a common boundary, it is possible to move between them by tracking the landmarks as we move toward the boundary. Thus, visual memory is an undirected graph where nodes are orientation regions, and arcs join adjacent regions.

The path-following algorithm for qualitative path planning assumes that we carry a compass as we move through the environment and mark the direction north relative to observed landmarks before and after we cross each LPB. Note that this sensor reading is purely local. It is not necessary to record changes in bearing as the sensor moves. Now we have at each node the compass heading of each landmark.

The start and goal points are orientation regions. We want to plan a path in our memory of large-scale space between these regions. We accomplish this task by propagating paths outward from each of the start and goal regions. Each adjacent region to the start or goal region is an approximate compass heading away from the start or goal region, respectively. We store this initial direction for each adjacent region. From each adjacent region, we now choose the single region adjacent to it that is a relative compass heading closest to the initial step from the

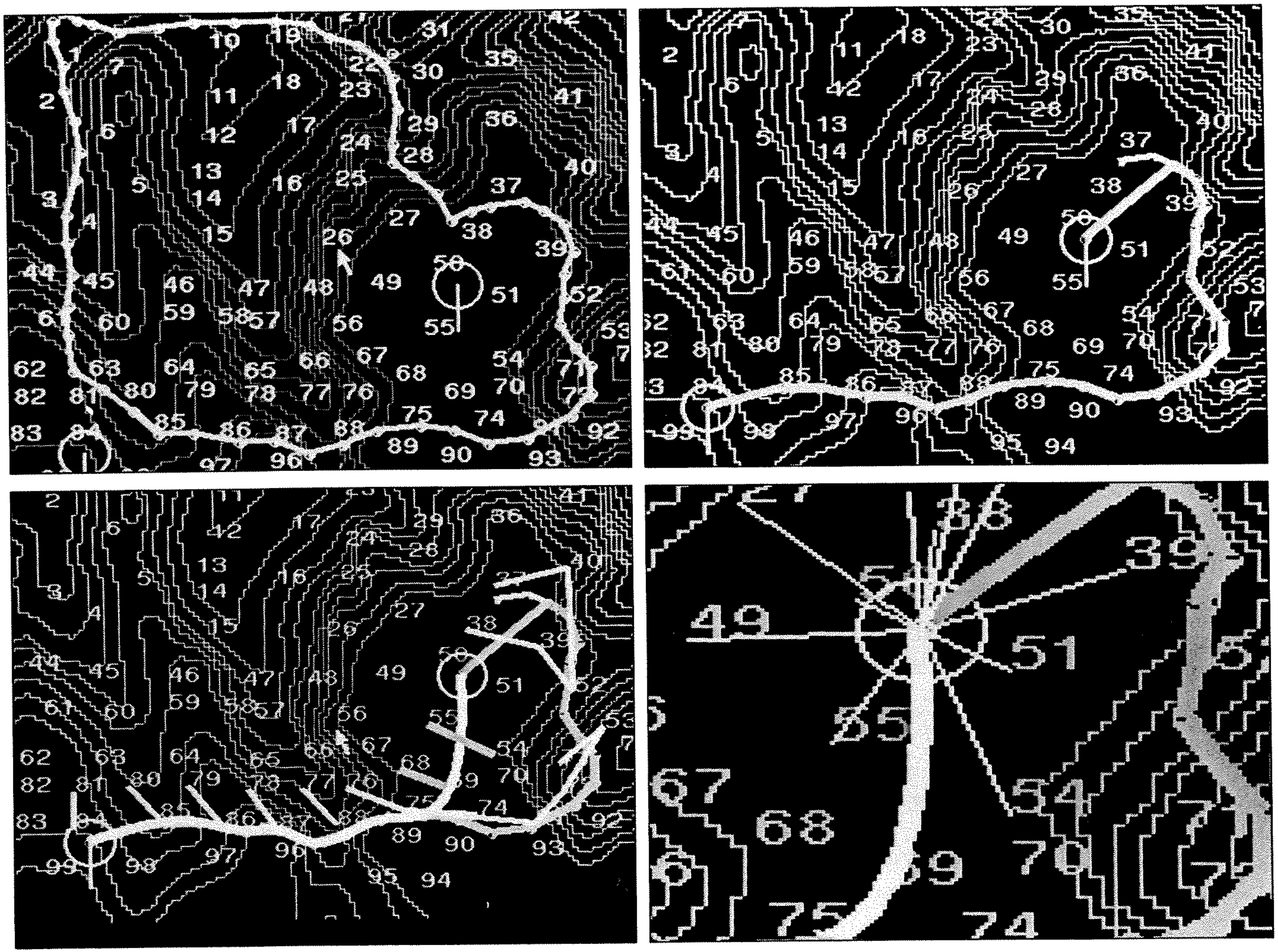


Figure 14. Qualnav Results. (Clockwise from right to left.)
(a) Start, Goal, and Path of Observation Stored in Visual Memory; (b) Path Planned in Visual Memory;
(c) LPBs Planned versus LPBs Crossed during Execution; (d) Vision-Based Shortcut to Goal.

start or goal. Thus, we propagate a single path for each region adjacent to the start or goal region.

Because we minimize the compass difference from the original step from start or goal as we propagate a path, an approximately straight path propagates from each region adjacent to the start or goal node. Because more than one region is adjacent to the start and goal modes, at least four paths exist, not all of which can be parallel. It follows that at least two of these paths, one from the start and one from the goal, must cross in visual memory. We check for crossing at each step by checking (but not propagating a path from) each adjacent region to see if a start or goal path has already crossed there.

This algorithm is illustrated in figure 13. Figure 13a shows the visual memory representation of large-scale space, with start and goal nodes. Figure 13b shows the adjacent regions propagated to, and figure 13c shows the choice from each adjacent region to one that minimizes the heading difference from the initial step. Finally, figures 13d, 13e, and 13f show propagation until path crossing is detected.

This algorithm is clearly order linear in the graph diameter of visual memory of large-scale space. A plan in visual memory is now a sequence of adjacent orientation regions. In path execution, we cross LPBs by tracking landmarks of the boundary LPBs between adjacent orientation regions. If we see landmarks further along our plan, we jump ahead to cross through them, thereby shortcutting the following process opportunistically.

Figure 14 shows a typical run of the Qualnav simulator for orientation region-level planning and execution. Figure 14a shows the start and the goal locations and the path already recorded in visual memory. Note that because of the path taken earlier, the robot does not plan a more direct route to its goal but instead computes the viewframe plan of figure 14b.

Figure 14c shows both LPBs planned and observed and crossed in the course of executing the plan of figure 14b. Notice that because landmarks 54 and 55 are not visible from the same view frame as the goal location (near landmark 50), the plan did not cross LPB (54,55). However, in plan execution, the robot opportunistically observed that it would have to

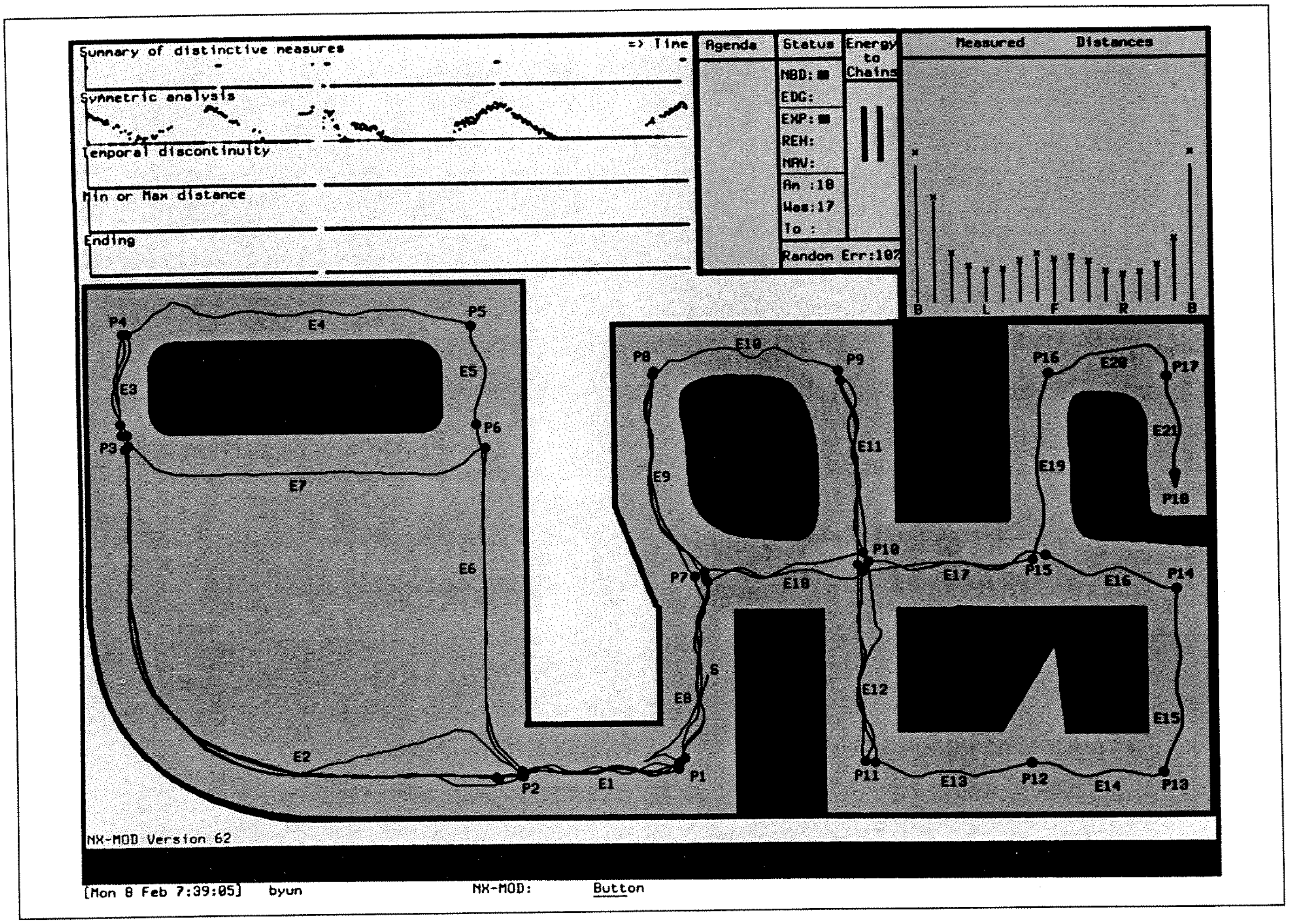


Figure 15. The NX Robot Exploration of an Environment.

be on the (54,55) side of LPB (54,55) to be at its goal location because the last view frame in the plan was on this side of the LPB.

After the robot crosses LPB (54,55), it happens that its goal location is visible; so, it heads directly toward it. Figure 14d shows the vision-based shortcut between the planned and executed paths.

The NX Robot

The NX robot (Kuipers and Byun 1987) extends the basic approach of the Tour model, handling continuous sensory input and continuous motion through the environment. The central problem is to define the places and path segments that allow a discrete, qualitative, network structure to be imposed on a continuous environment.

Figure 15 shows the result of the exploration of a room-and-corridor world by NX. The large dots (labeled P1 through P18) show the points at which places are described in the map, and the thin lines (labeled E1 through E21) show NX's track as it followed the path segments.

NX still follows the basic Tour model approach, oriented toward finding places and paths in the environment rather than including regions as a primary representational element, as in Qualnav.

Sensorimotor Interaction with the Environment

NX is a simulated robot running on the Symbolics 3600 but with a design similar to several laboratory and commercial mobile robots (Brooks 1986; Denning Mobile Robotics 1988). Sensory input comes from a symmetrically arranged ring of 16 range sensors inspired by the Polaroid ultrasonic range finder (Elfes 1986). The simulated range sensors sense actual distance to the nearest obstacle, optionally imposing a specified degree of random error rather than simulating the complex acoustical properties of actual sonar. NX also uses a compass to sense absolute orientation, although dependence on this sense is being eliminated in current work. The block in the upper-right corner of figure 15 shows the current range-sensor input with the robot at P18, including random error ≤ 10 percent. Motor control is by energy sent to two tractorlike chains, permitting straight or curved motion or turning in place.

Procedural Behaviors

In the continuous world, the difficult problem to be solved during exploration is determining where the significant places and paths are and finding them again on returning to the general neighborhood. In NX, these problems are solved at the procedural level by selecting and executing control strategies defined in terms of locally available sensory inputs.

Define a place as the point in the environment that is the local maximum of some distinctiveness measure defined over its neighborhood. As a consequence, once the exploration or navigation strategy brings us into the neighborhood of a place, a simple hill-climbing control algorithm brings us to the place itself, in a way that is relatively insensitive to sensory and motor errors, or to the point in the neighborhood where we began. In figure 15, the upper-left block shows a time-series history graph of the distinctiveness measures (scrolling horizontally).

Exploration of a neighborhood consists of physical motion and analysis of sensory input to determine the best distinctiveness measure defined in this neighborhood. In many cases, a particularly useful distinctiveness measure counts the number of nearby objects and attempts to maximize the similarity (that is, minimize the differences) among their distances. A simple rule-based system examines the qualitative properties of the stream of sensory input to select the appropriate distinctiveness measure.

Define an edge (or path segment) as the control strategy (for example, follow corridor midline) required to move from the neighborhood of one place to the neighborhood of another. Again, the use of control algorithms makes this step relatively insensitive to sensory or motor errors. During exploration, another rule-based system selects the appropriate control strategy for a particular part of the environment based on local sensory cues, determines when the control strategy has failed, and decides whether a different path-following control strategy is appropriate or if the robot is likely to be in a new neighborhood.

Overall exploration is controlled by an agenda of unexplored edges. There are several exploration methods for testing whether a newly encountered place matches a previously known place. These methods are capable of distinguishing places whose sensory image is identical by hypothesizing and testing predictions about the topological connectivity of the place.

The Topological Map

As places and edges are identified and defined in terms of their distinctiveness measures or control strategies, a topological map is created to represent their connectivity, very much like the Tour model's network of places and paths. This topological map is used by NX to plan routes to places that still have unexplored directions on the agenda and to plan routes to test place-match hypotheses.

The Metric Map

The metric map for NX extends the local geometry portions of the Tour model metric map. The local headings at a place are defined to specify directions of edges to explore and the direction and range of nearby objects. When the global compass is used, all headings are defined with respect to the global orientation frame. As we move to a version without a global compass, an orientation frame will be propagated throughout the region during travel. Headings at different places will become tightly coupled with more travel experience between them, as the $\Delta\theta$ for each edge becomes precisely known.

Each edge can be described as a generalized cylinder (Binford 1971) in terms of the length (δ), net orientation change ($\Delta\theta$), and shape of the axis and the profile of cross-sections as a function of position along the axis. This representation allows passability information to be used for route finding with a nonpoint-sized robot and provides additional information for place and edge recognition.

Conclusions

We have described a four-level spatial semantic hierarchy that provides a framework for robust map learning and navigation in large-scale space. The levels are (1) sensorimotor interaction, (2) procedural behaviors, (3) topological map, and (4) metric map. Three programs with different approaches to large-scale space—the Tour model, the Qualnav model and simulator, and the NX robot—demonstrate the successful application of these concepts to the representation of spatial knowledge.

From the perspective of the spatial semantic hierarchy, we can see why the purely metric approach to mapping is often fragile. First, even when only the procedural level representation is immediately required, the full effort to use the metric-level representation, including expensive metric matching between observations and the map, is always necessary. Second, during mapping, when there is insufficient information to specify a unique match between observations and the metric map, either the match must be abandoned (losing an opportunity for knowledge acquisition), or a heuristic assumption is added to force a unique match (causing a certain frequency of serious errors in the map).

When considering human mapping and navigation in real environments, we notice that individual variants exhibiting different subsets of the full semantic hierarchy are easy to identify. One person learns a few important route-following procedures and never creates a topological or metric map at all. Another forces all observations into a preexisting metric framework and is troubled by the incoherence of nonrectilinear environments (for example, Boston Common or Embarcadero in Palo Alto, California). Yet another person moves robustly about in unmarked, cross-country environments, using a compass and distant peaks as landmarks.

The theory and implementation of Tour, Qualnav, and NX demonstrate that these human capabilities are modelable and can lead to practical results in robotics and automation. There is much left to be done, but we claim that the spatial semantic hierarchy offers a coherent organization to guide the development of effective, robust strategies for acquiring and using knowledge of large-scale space.

Acknowledgments

Benjamin Kuipers acknowledges the important contributions of Yung-Tai Byun to the work on the NX Robot. The research on robot exploration is supported in part by NASA Johnson Space Center, under Grant no. NAG 9-200.

Tod Levitt makes special acknowledgment to his research colleague Daryl Lawton. Qualnav work was performed in mutual collaboration. In particular, the perceptual bias and many of the insights in integrating vision and spatial reasoning are due to Lawton (see for example, Lawton [1983]). Thanks are due him not only for his contributions but also for his inspiring dedication to the quest for real-time algorithms that realize theory to achieve automation.

The Qualnav development team also includes Dave Chelberg, Kerry Koitzsch, Phil Nelson, John Dye, and Chris McConnell. Thanks are owed them for their help in developing and verifying the Qualnav algorithms and for the many hours of hard labor involved in implementing the theory, robot simulation environment, and attendant graphics.

This document was prepared by Advanced Decision Systems (ADS) of Mountain View, California, under U.S. Government contract no. DACA76-85-C-0005, for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, and the Defense Advanced Research Projects Agency, Arlington, Virginia, and under internal funds at ADS.

References

Bajcsy, R.; Krotkov, E.; and Mintz, M. 1986. Models of Errors and Mistakes in Machine Perception, Technical Report, MS-CIS-86-26, GRASP LAB 64, Dept. of Computer and Information Science, Univ. of Pennsylvania.

Binford, T. O. 1971. Visual Perception by Computer. In Proceeedings of the IEEE Conference on Systems and Control. Washington D.C.: IEEE Computer Society.

Brooks, R. A. 1987. Visual Map Making for a Mobile Robot. In *Readings in Computer Vision*, eds. M. A. Fischler and O. Firschein, 438-443. Los Altos, Calif.: Morgan Kaufmann.

Brooks, R. A. 1986. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation* RA-2(1): 14-23.

Cao, X.; Oh, S.; and Hall, E. 1986. Dynamic Omnidirectional Vision for Mobile Robots. *Journal of Robotic Systems* 3(1): 5-17.

Chase, W. G. 1982. Spatial Representations of Taxi Drivers. In *Acquisition of Symbolic Skills*, eds. D. R. Rogers and J. A. Sloboda. New York: Plenum.

Chatila, R., and Laumond, J. 1985. Position Referencing and Consistent World Modeling for Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation, 138-170. Washington D.C.: IEEE Computer Society.

Davis, E. 1983. The MERCATOR Representation of Spatial Knowledge. In Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 295-301. Los Altos, Calif.: Morgan Kaufman.

Denning Mobile Robotics, Inc. 1988. Modular Research Vehicles. Product specification sheet on the MRV series of mobile robot platforms. Wilmington, Mass.: Denning Mobile Robotics, Inc.

Eastman, R. and Waxman, A. 1985. Disparity Functionals and Stereo Vision. In Proceedings of the DARPA Image Understanding Workshop, 245-254, DARPA SAIC Report no. 85/1149. McLean, Va.: Defense Advanced Research Projects Agency.

Elfes, A. 1986. A Sonar-Based Mapping and Navigation System. In Proceedings of the IEEE International Conference of Robotics and Automation, 1151-1156. Washington D.C.: IEEE Computer Society.

Elliot, R. J., and Lesk, M. E. 1982. Route Finding in Street Maps by Computers and People. In Proceedings of the Second National Conference on Artificial Intelligence, 258. Los Altos, Calif.: Morgan Kaufmann.

Foreman, N., and Stevens, R. 1987. Relationships between the Superior Colliculus and Hippocampus: Neural and Behavioral Considerations. *Behavioral and Brain Sciences* 10(1): 101-151.

Hayes, P. J. 1979. The Naive Physics Manifesto. In *Expert Systems in the Micro Electronic Age*, ed. D. Michie, 242-270. Edinburgh: Edinburgh University Press.

Hebert, M., and Kanade, T. 1985. First Results on Outdoor Scene Analysis Using Range Data. In Proceedings of the Image Understanding Workshop, 224-231, DARPA SAIC Report no. 85/1149. McLean, Va.: Defense Advanced Research Projects Agency.

Kosslyn, S. M. 1980. *Image and Mind*. Cambridge, Mass.: Harvard University Press.

Kozlowski, L. T., and Bryant, K. J. 1977. Sense of Direction, Spatial Orientation, and Cognitive Maps. *Journal of Experimental Psychology: Human Perception and Performance* 3:590-598.

Kuipers, B. J., and Byun, Y. T. 1987. A Qualitative Approach to Robot Exploration and Map Learning. In Proceedings of the IEEE Workshop on Spatial Reasoning and Multi-Sensor Fusion, 390-404. Los Altos, Calif.: Morgan Kaufmann.

Kuipers, B. J. 1985. The Map-Learning Critter, Technical Report, AITR85-17, Artificial Intelligence Laboratory, The Univ. of Texas at Austin.

Kuipers, B. J. 1983. Modeling Human Knowledge of Routes: Partial Knowledge and Individual Variation. In Proceedings of the Third National Conference on Artificial Intelligence, 216-219. Menlo Park, Calif.: American Association for Artificial Intelligence.

Kuipers, B. J. 1982. The "Map in the Head" Metaphor. *Environment and Behavior* 14(2):202-220.

Kuipers, B. J. 1979. Commonsense Knowledge of Space: Learning from Experience. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 499-501. Los Altos, Calif.: Morgan Kaufmann.

Kuipers, B. J. 1978. Modeling Spatial Knowledge. *Cognitive Science* 2:129-153.

Kuipers, B. J. 1977. Representing Knowledge of Large-Scale Space, Technical Report, TR-418, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Lawton, D. T. 1983. Processing Translational Motion Sequences. *Computer Vision, Graphics, and Image Processing* 22:116-144.

Levitt, T.; Lawton, D.; Chelberg, D.; and Nelson, P. 1987a. Qualitative Navigation. In Proceedings of the DARPA Image Understanding Workshop, 447-465. Los Altos, Calif.: Morgan Kaufmann.

Levitt, T.; Lawton, D.; Chelberg, D.; Nelson, P.; and Dye, J. W. 1987b. Visual Memory Structure for a Mobile Robot. In Proceedings of the IEEE Workshop on Spatial Reasoning and Multisensor Fusion, 92-106. Los Altos, Calif.: Morgan Kaufmann.

Levitt, T.; Lawton, D.; Chelberg, D.; Koitzsch, K.; and Dye, J. W. 1988. Qualitative Navigation 2. In Proceedings of the DARPA Image Understanding Workshop, 319-326. Los Altos, Calif.: Morgan Kaufmann.

Lucas, B., and Kanade, T. 1984. Optical Navigation by the Method of Differences. In Proceedings of the DARPA Image Understanding Workshop, 272-281, DARPA SAIC Report no. 84/1726. McLean, Va.: Defense Advanced Research Projects Agency.

Lynch, K. 1960. *The Image of the City*. Cambridge, Mass.: MIT Press.

Marr, D. 1982. *Vision*. San Francisco: Freeman.

Matthies, L., and Shafer, S. 1986. Error Modeling in Stereo Navigation, Technical Report, CMU-CS-86-140, Computer Science Dept., Carnegie-Mellon Univ.

McDermott, D. V. 1980. Spatial Inferences with Ground, Metric Formulas on Simple Objects, Technical Report, 173, Dept. of Computer Science, Yale Univ.

McDermott, D., and Davis, E. 1984. Planning Routes through Uncertain Territory.

Artificial Intelligence 22: 107-156.

Miller, D. 1985. A Spatial Representation System for Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation, 122-128. Washington D.C.: IEEE Computer Society.

Moravec, H. 1980. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Ph.D. diss., Robotics, Dept. of Computer Science, Stanford Univ.

Norman, D. A. 1981. Categorization of Action Slips. *Psychological Review* 88:1-15.

O'Keefe, J., and Nadel, L. 1978. *The Hippocampus as a Cognitive Map.* New York: Oxford University Press.

Pentland, A. 1985. A New Sense for Depth of Field. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 988-994. Los Altos, Calif.: Morgan Kaufmann.

Piaget, J., and Inhelder, B. 1967. *The Child's Conception of Space.* New York: Norton.

Pylyshyn, W. 1984. *Computation and Cognition: Toward a Foundation for Cognitive Science.* Cambridge, Mass.: MIT Press.

Schone, H. 1984. *Spatial Orientation— The Spatial Control of Behavior in Animals and Man.* Princeton, N.J.: Princeton University Press.

Shepard, R. N., and Metzler, J. 1971. Mental Rotation of Three-Dimensional Objects. *Science* 171: 701-703.

Siegel, A. W., and White, S. H. 1975. The Development of Spatial Representations of Large-Scale Environments. In *Advances in Child Development and Behavior,* vol. 10, 9-55, ed. H. W. Reese. New York: Academic.

Smith, R., and Cheeseman, P. 1985. On the Representation and Estimation of Spatial Uncertainty, Technical Report, Grant ECS-8200615, SRI International Robotic Laboratory.