Robotica (2008) volume 26, pp. 189–203. © 2007 Cambridge University Press
doi:10.1017/S026357470700375X Printed in the United Kingdom

Topologically-directed navigation

David Rawlinson* and Ray Jarvis

Intelligent Robotics Research Centre, Monash University, Melbourne, Australia.

(Received in Final Form: December 4, 2006. First published online: September 10, 2007)

SUMMARY

Recent advances in simultaneous localization and mapping permit robots to autonomously explore enclosed environments and, subsequently, navigate to selected positions within them. But, for many tasks, it is more useful to immediately navigate to goals in unexplored environments, without a map. This is possible if a human director can describe the ideal route to the robot using grounded symbols that both parties can perceive directly.

In this paper, a mobile robot is autonomously navigated to many locations in a cluttered laboratory environment by a variety of routes. A series of topological navigation instructions are provided in advance by the director, in a form that can be expressed verbally and translates easily to software representation. The instructions are based on the perception of spatial affordances available to the robot, namely nearby junctions and edges in a pruned Generalized Voronoi Diagram. The operator can generate the instructions by viewing or imagining the environment without any measurements. Only three to five instructions are needed to navigate anywhere in our laboratory. The instructions contain only topology. No spatial measurements or environmental data such as landmarks are provided to the robot.

KEYWORDS: Goal-directed; Navigation; Topological; Exploration.

1. Introduction

This paper presents a new approach to navigation in an unknown environment by exploiting knowledge readily available to a human director, communicated to the robot using a simple topological language. The scenario is as follows. First, the human director views or imagines the environment in question. The director then plans a route to the goal, and explains this plan to the robot. Finally, the robot executes the plan autonomously, with no further assistance from the director and no additional information about the environment, such as a map.

The viability of this concept depends on the language chosen for communication and interpretation of the plan by the robot. The language must be topological to make it feasibly efficient to describe a long route. Importantly, topological localization to the next landmark in a planned sequence eliminates accumulated error, whereas the incremental interpretation of a metrical plan (e.g., move 10 m forward, then 5 m right...) becomes increasingly uncertain.

It must also be easy for people to generate accurate plans in the chosen language. Research into machine-generated navigational instructions for human execution (for example, car satellite navigation systems[1]), reveals that humans prefer to communicate topological plans, matching their invariably topological internal models of space.[2–4] People find estimation of distances difficult, particularly from memory of a place, so ideally topological plans would not rely on assessments of distance (inaccurate) or measurements (inconvenient).

A critical feature of the chosen language must, therefore, be topological landmarks, that must be perceived reliably in the real world, first by the human director, and later, during navigation, by the robot. It is crucial that perception of landmarks is mutual—in other words, both parties must perceive the same landmarks, without omissions or additions.

Finally, for efficient planning and communication, the landmark features must be sparse. However, landmarks must naturally occur with sufficient frequency in all environments you wish to navigate. This paper proposes a simple language that meets all these criteria (Table III).

Since, in our scenario, the robot must execute topological plans autonomously, it must detect topological landmarks using only the data that it incidentally accumulates during navigation. It has no prior map, and it cannot waste time mapping the world before executing the plan. This explains why empirical demonstration of topologically directed navigation in a realistic environment could not occur before the development of hybrid (metric-topological) autonomous mapping systems capable of sensing topological landmarks given only local spatial information. Only recently have systems matching this specification been demonstrated.[5–8] There is also evidence that people maintain egocentric representations of the world,[9] reconstructed incrementally during navigation.

1.1. Exploration and mapping

This work stems from recent successes in Simultaneous Localization and Mapping (SLAM) using stochastic algorithms typically derived from the Kalman filter[10–13] or Expectation-Maximization (EM).[14] Using these algorithms, robust results were achieved in a wide variety of scenarios and, more recently, attention has turned toward improvements in the accuracy or speed of mapping[7,15–19] and improvements in exploration technique[20–23], so that the environment can be mapped more efficiently. So,

* Corresponding author. E-mail: david.rawlinson@eng.monash.edu.au

success in SLAM has turned the spotlight on other issues, including how to direct exploration. It is widely but, perhaps, erroneously accepted that exploration must precede goal-directed navigation.[24]

Currently, many mapping experiments are conducted using semiautonomous or manual navigation to ensure that the desired locales are built into the map.[25–27] Often, the human navigator will employ his or her foresight to ensure small loop closure and the avoidance of long uninteresting loops and dead-ends. Manual navigation also fails to show that the achieved mapping is sufficient for autonomous navigation. In many autonomous exploration experiments, unless the environment is enclosed, autonomous mapping will not be completed within a reasonable period of time. This is sometimes avoided by modifying the environment.[16,21]

Behind all SLAM lies the assumption that useful activities will be undertaken once mapping is complete, which is largely true: Within finite yet reasonably large, at least partially known environments, efficient navigation can be achieved using existing planning algorithms.[28,29] Of course, in some situations, the mapping is itself a useful outcome.

Originally this work was motivated by limitations in existing SLAM techniques, that cannot exploit human guidance concerning where to explore. In many cases, mapping a complete environment is not desirable and modification of the environment to enclose the robot is inconvenient. Furthermore, in situations where mapping is only a precursor to the desired task, it is preferable to map only the route necessary for completion of the task.

Conventionally this sounds impossible, since how can the ideal task-achieving route be known before the map is complete? In fact, humans are able to navigate in unknown environments very efficiently by exploiting the assistance of better-informed individuals, via a shared language that allows complex directions to be transferred verbally, or visually, using signs. Even if the goal is mapping, it would be useful to be able to describe the areas that need mapping in advance, to permit efficient autonomous exploration.

1.2. Hybrid topological mapping

Hybrid (metric/topological)[20,30–22] and (feature/topological)[26,33–35] mappings were developed because although planning is efficient in topological models, continuous global localization is required to build accurate and consistent global maps using stochastic methods. In most of these works, the topological model is continually rebuilt from geometric data as the accuracy of the latter improves. Before completion of the global geometric map, the topological model may be poorly estimated.[32,36] A few researchers have recently demonstrated global topological mapping without global geometric models.[5–7] Instead, local or egocentric geometric data is used to model local topology. It is a small step further to use language to transfer topological information from human to robot, perceive geometric data locally, and omit the need for exploration altogether.

1.3. Existing goal-directed navigation strategies

Articles in the literature concerning goal-directed navigation usually define goals as positions in a robot's internal global geometric map[13,37] or vertices of a robot's global topological or hybrid map.[16,20,35] The robot can then plan a path to the goal in its internal map.[13,28,38] Alternatively, a series of waypoints defined by externally corrected odometry or Global Positioning System (GPS) signals may describe the path to the goal[39,40] rather than the goal itself. The robot is, then, responsible for execution of the supplied geometric plan; when following the specified route, significant diversions may be needed to avoid unexpected obstacles.

In this work, the robot is given topological directions to the goal—equivalent to an incomplete topology—but does not have a global topological map or any geometric information with which to ground the provided topology. It must, instead, acquire geometric data during navigation in order to perceive topological landmarks and execute the plan. The lack of geometric data in provided topological instructions is deliberate—it is easy for people to perceive and communicate topology verbally, but hard to do the same with geometric models. In the literature, there are no examples of topologically directed navigation in realistic environments without a global map, although autonomous execution of verbally delivered topological directions in a simplified environment with artificial topological landmarks was demonstrated in the IJCAI robot competitions of 1994 and 1995.[41,42] In these examples, the robots were given a global topological map as well as instructions.

The robot described here localizes topologically, but not continually or relative to geometric landmarks. This allows operation without global geometric data. The most similar works to date are fully-autonomous global–topological and local-geometric hybrid mapping techniques such as in refs. [6], [7], [16] and [35], but these associate local geometric data with global topological landmarks to allow continuous localization and recognition of particular topological vertices. The geometric data increases the difficulty of communicating these maps.

1.4. The shared perception of spatial affordances

In this paper, we propose a simple language for topologically directed navigation, based on only two concepts—junctions and edges of the GVD as popularized by Choset.[43–45] However, the GVD topology that is used in this paper is heavily pruned and simplified, compared to accurate GVD models. Only edges that the robot can physically traverse are retained. Smaller gaps between obstacles, and minor variations in obstacle contours, are not represented in the topological model. In the next part of the paper, a method of easily computing this pruned GVD is presented.

Use of the GVD can be imagined as an inversion of the landmark selection problem. Concrete landmarks, such as doorways, may not occur in all environments. But a suitably pruned GVD can be considered a map of the navigational options available to the robot, which, by definition, are always available in all environments. By explicitly modeling the effects of external features on the navigator, rather than the features themselves, generality is assured. GVD-like structures are often known as *roadmaps*.

Another crucial factor for the selection of the GVD as a topological representation is that nearby edges and vertices of the GVD can be perceived in the robot's immediate

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

surroundings. This means that the robot can incrementally and incidentally perceive topological landmarks—namely GVD vertices—as it follows the topological plan provided by the human director, without having to divert from this task to detect landmarks with extra mapping. Other topologies, particularly,[6,7] employ complex characterizations of places that may require dedicated navigational activity to be accurately identified.

The GVD has other appealing characteristics. Where navigation is required beyond the edges of the GVD, it is by definition trivial since there will be no intervening obstacles. From anecdotal experience, people can easily understand and perceive the pruned GVD, in the real world, unassisted. It has already been shown that robots can perceive the GVD in their surroundings using a local or egocentric map.[5–7]

Using Gibson's notion of affordances,[46] we can describe all the parameters for construction, pruning, and use of the GVD in terms of the robot's characteristics. From a human perspective, a GVD pruned to remove all the edges the robot cannot traverse is more intuitive and easily described without technical definitions.

Humans are practiced at estimating the physical affordances of other people, machines, and animals. It is also very useful to have an overall principle to refer to when deliberating the topology the robot *should* perceive. The notion of affordances gives a consistent definition despite variations in the accuracy and resolution of the robot's sensors, and changes in the robot's physical size.

The high angular resolution of typical scanning-range sensors can yield a very complex GVD. There will be many gaps between sensed obstacles, and irregular surfaces with concavities, into which the GVD extends branches. Typically, dilation (see the following text), thinning,[32] or pruning[7,45] is applied to this structure to simplify it. The combination of the robot's own physical properties and the existing graph density determine the amount of pruning or dilation required.

The robot cannot pass through a gap smaller than itself, thereby defining what is and is not a viable junction and/or edge. If two junctions are close together, the robot's size means that it can occupy more than one GVD junction simultaneously, effectively creating a single vertex with four or more incident edges (see Fig. 1). Given infinite accuracy, it is unlikely that any vertex of the GVD would have more than three edges.

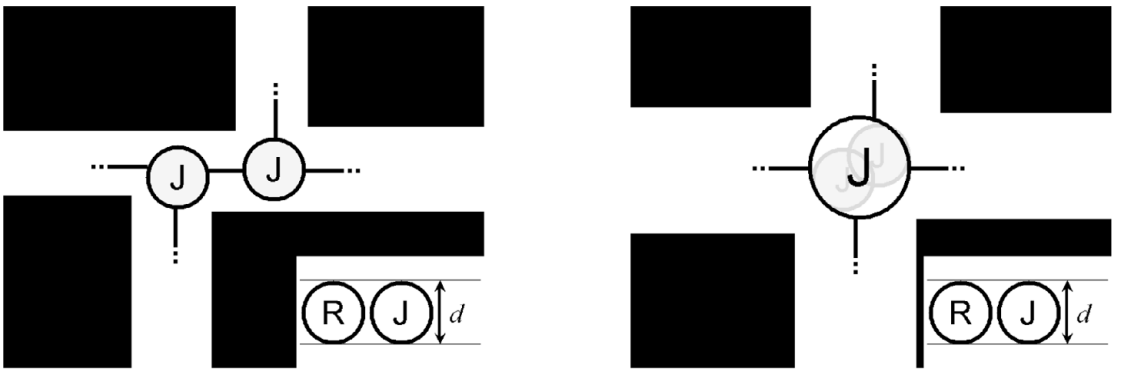
If vertices are merged, two ambiguities are eliminated. The first is the order in which vertices are visited, which cannot be determined unless the robot's movement is highly constrained and the unmerged vertices are precisely positioned. The second ambiguity is which unmerged vertex the robot occupies if parts of the robot are at different vertices. However, a third ambiguity is created—whether two vertices are merged, given the distance between them. In the reported experimental trials, the environment was altered to cause this type of ambiguity, and a method of resolving it is demonstrated.

Although it is not possible to eliminate all types of ambiguity, errors can be managed by choosing when they occur. In order to minimize the use of physical constants and metric parameters, all vertices are considered to have an area equal to the diameter of the robot. Overlapping vertices are merged. It is easiest for the director to assess a scene in terms of the robot's size, and what it is possible for the robot to do at that location. This also makes it easier to explain the direction language concepts to a lay user. In summary, we define the topology of spatial affordances as a selectively pruned and merged GVD, in which the size of the robot determines the outcome of pruning and merging.

1.5. Are topological directions a map?

So far, we have established that there exists a navigationally useful common language that can be shared by humans and robots. Given this shared symbology, the next step is to show that knowledge transfer is possible, in other words, that useful navigational tasks can be achieved by the robot using a human director's knowledge of the world. Communication between both parties will be based on concepts that both can perceive locally and directly.

It is arguable that a series of navigational instructions actually constitutes a topological map. This may be true; but if a person can generate a map of this type from memory and transmit it verbally it constitutes an enormous increase in utility over traditional map types that must be drawn, digitised, calculated using measuring devices, or generated through robot exploration. The issue of whether topological



(a) Two separate junctions; the area of each junction is defined by the diameter d of the robot.

(b) Junctions so close that they overlap are considered a single larger junction with multiple exits.

Fig. 1. Ambiguity from junction merging. If two junctions only slightly overlap sensor error will determine whether they are perceived as one or two junctions.

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

directions constitute an incomplete topological map is, in comparison, insignificant.

2. Implementation

2.1. Mobile robot

The experiments reported in this paper were conducted using an ActivMedia Pioneer 3 All-Terrain mobile robot featuring a skid-steer drive mechanism. Due to the topological nature of the navigation method proposed, the resultant poor odometry is not problematic. A Hokuyo URG scanning laser-range sensor is mounted on the robot for obstacle detection and the perception of spatial affordances, as detailed later.

2.2. Computing the GVD in an egocentric metric map

An Egocentric map is always centered on the robot, and has limited scope. Only the area around the robot is mapped. When the robot moves, the mapped area moves like a scrolling window, keeping the robot at its center, as in refs. [6] and [7]. However here, odometry is used without correction to translate the map to account for robot motion. In this work, Egocentric maps are represented by images, i.e., as a regular grid of cells. To compute the GVD of free space around the robot, it is first necessary to know which cells are occupied by obstacles.

Since both sensors and odometry are inaccurate, stochastic methods such as Bayes' update rule[47] are necessary to infer cells' occupancy probability, given multiple sensor readings. Stochastic reasoning is also necessary to remove dynamic or transient obstacles such as passing people. The Occupancy Grid[48] is a suitable stochastic model for Egocentric mapping because distant obstacles roll off the edge of the map and large loops cannot occur. Similarly, scan-matching is unnecessary because cumulative errors are not significant. Therefore, the authors describe the model detailed later as an *Egocentric Occupancy Map* (EOM).

To calculate the probability that a given cell in the egocentric grid is obstructed, it is necessary to know the prior probability of occupancy $P(H)$ and the probability of obtaining current sensor evidence if the cell is occupied $P(E|H)$. In ref. [49] $P(E|H)$ is obtained from an *occupancy probability profile* for a particular sensor. In this work, since the geometric accuracy of the EOM is not important, a two-dimensional profile was created that was required to account for all types of error—dynamic obstacles, odometric error in translation of the map, and sensor error.

The two-dimensional profile is built from three Gaussian kernels—one to support the hypothesis that cells around the sensed location are occupied; second, a strong refutation of the occupancy of intervening cells along the line of sight of the sensor, and, last, a mild refutation in the case where nothing is sensed. Note that priors default to 0.5 as they are added at the edge of the map, or are inherited from cells' previous occupancy probabilities using uncorrected odometry to account for robot translation. To avoid interpolation of priors, the orientation of the egocentric map is fixed.

Since a scanning laser is used, the evidence from all N readings is summed and normalized. In Eqs. (1), and (2), a hierarchy of forks determines which kernel will be used to generate an inference for a particular (pixel, sensor reading) combination. Function $f(i, j, n)$ is an arbitrary name for the second level of this hierarchy. Refutation is maximized along the sensor's line of sight and rotational odometric error dominates; hence, the difference between sensor bearing θ_n and pixel bearing $\theta_{i,j}$ is used as a distance measure for these kernels. Support is based on the Euclidean distance $x_{i,j,n}$ from the sensed obstacle position. The support model is invoked if the distance from the sensor to the pixel $d_{i,j}$ is greater than a threshold $t_n = d_n - 3\sigma_x$ where d_n is the sensed distance and σ_x is the standard deviation of expected error. It is assumed that due to the Gaussian profile, support beyond $3\sigma_x$ would be insignificant; the authors found ad-hoc tuning of σ_x acceptable, given that this model should account for multiple sources of error that cannot easily be modeled, and accuracy is not crucial. The strength of refutation is determined by two parameters, α and β. At insignificant loss of precision, the output for all possible configurations of sensor and pixel can be precomputed and stored in a lookup-table. It is worth mentioning that if greater accuracy were required, this component could be replaced with more accurate Occupancy Grid or SLAM methods.

$$P(E|H)_{i,j} = \eta \sum_{n=0}^{N} \begin{cases} f(i, j, n) & \text{valid reading} \\ \alpha \cdot \text{refute}(\theta_{i,j}, \theta_n) & \text{otherwise} \end{cases} \tag{1}$$

$$f(i, j, n) = \begin{cases} \beta \cdot \text{support}(x_{i,j,n}) & d_{i,j} > t_n \\ \gamma \cdot \text{refute}(\theta_{i,j}, \theta_n) & \text{otherwise} \end{cases} \tag{2}$$

$$\text{support}(x) = 1 + \left(\frac{1}{\sqrt{2\pi}\sigma_x}\right) e^{-\frac{x^2}{2\sigma_x^2}} \tag{3}$$

$$\text{refute}(\theta, \bar{\theta}) = 1 - \left(\frac{1}{\sqrt{2\pi}\sigma_\theta}\right) e^{-\frac{(\theta-\bar{\theta})^2}{2\sigma_\theta^2}}. \tag{4}$$

In this paper, an approximation of the GVD is created from the EOM using the method in ref. [5], which has linear complexity in the number of map pixels and does not require pruning of the resultant graph. Alternatives would include the morphological thinning method proposed by ref. [32], clustering of obstacle features[50], pruning of the true GVD as suggested by refs. [7], [31] and [45] and pruning of the Medial Axis Transform.[51] The morphological thinning approach has the drawback that the mapped region must be enclosed. In a local, incomplete or egocentric map enclosure will cause spurious or misplaced junctions and edges between actual obstacles and the edge of the map. Pruning the graph model is complex[7,31,45] and, therefore, not convenient.

The GVD is approximated by first thresholding the egocentric occupancy probability map P at 0.51, giving a map of confirmed obstacles O. The chosen threshold ensures that without evidence, cells are modeled as unoccupied. These and subsequent maps are stored and processed as images. No gap smaller than the diameter d of the robot can be traversed; therefore, the obstacles in O are dilated by $d/2$ giving image D. Dilation is achieved using the Distance Transform[52] of the background or free space region,[53] thresholding at the pixel-scale equivalent of $d/2$.

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

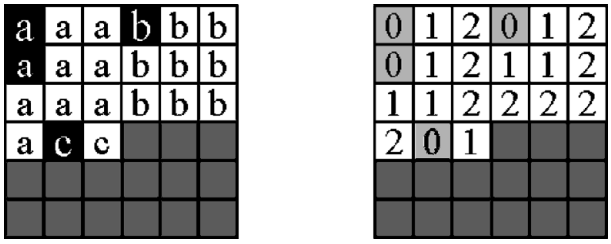


Fig. 2. Simultaneously propagating *distance to* and *label of* the nearest obstacle. This figure displays two images N (left) and M (right) half-way through the first of the two passes of the proposed custom distance transform algorithm. Three obstacles (black pixels a, b, c) are visible in N, and in all other pixels, the label of the closest obstacle. Obstacle a consists of two pixels—the connected component transform has already been applied to ensure that all connected obstacle pixels share the same label. The right image (M) stores distance to these obstacles. Note that at this point part way through the first pass, many pixels are undetermined, and that distance and obstacle labels are propagated from top-left to bottom-right; in the second pass, labels and distance are propagated in the opposite direction. The reader should note that unlike the actual algorithm, integer distances are used in this example.

Subsequently, a single pass of D is sufficient to uniquely label all 8-way connected feature regions.[54] The labels are stored as image L. At this stage, a modified Distance Transform is applied to L so that the label of the closest feature is propagated with accumulating distance. This results in two images, one containing distance to the nearest dilated obstacle (M), the other containing the label of the nearest dilated obstacle (N) (see Figure 2).

An image V approximating the GVD may be computed in a single further pass of N: For each pixel in N, count the number of distinct labels n among its eight neighbours: If $n = 2$, the pixel lies on an edge of the GVD; if $n > 2$, the pixel is located at a junction in the GVD.

$$
\begin{aligned}
O &= \text{Threshold}(P, 0.51);\\
T &= \text{DistanceTransformOf}(O);\\
D &= \text{Threshold}(T, t_{\text{Dilate}});\\
L &= \text{LabelConnectedComponentsIn(D)};\\
M,N &= \text{DistanceTransformOf}(L);\\
V &= \text{CountDistinctNeighbours}(N).
\end{aligned}
$$

Note that this method of approximating a pruned GVD is unusual because it does not produce "weak meet points," spurs, or other navigationally irrelevant skeletar features. Treatment of obstacles as labeled connected components ensures that topology is unaffected by changes in obstacle boundary or shape. Topological variations are only caused by the appearance or disappearance of an entire obstacle.

2.3. *Coastal navigation of open areas*

The finite range of typical laser and sonar sensors makes it difficult to accurately and replicably move across large open areas. When no obstacles are detected, the robot must rely on internal sensing (odometry) to maintain course. Without corrective feedback from the world, odometric error will accumulate. Although, in this work, odometry is not corrected using laser data, obstacles still have the effect of constraining motion and, hence, make navigation more consistent.

For these reasons, an Extended-GVD model may be preferred, similar to that proposed by Beeson and Kuipers.[6–8] (See the cited works for illustrations of Extended-GVD topology). In the Extended-GVD, edges are formed around the perimeter of obstacles when no other obstacles are nearby. By following this perimeter and always keeping an obstacle in view, the robot may navigate more predictably. This approach is sometimes known as *coastal navigation*.

An Extended-GVD can be computed using the method presented in this paper for approximating the GVD by propagating the labels of connected components. A new threshold (t_{Free}) is required to define the distance of the new edges from obstacles. Figure 3(b) shows that t_{Free} can be configured so that two alternative topologies generate an edge in the same place. When only one obstacle is detected, the dotted edge in Fig. 3(b) is followed. This edge is a perimeter edge. When another obstacle is detected (e.g., a black circle in Fig. 3(b)), an edge is, instead, formed between the two obstacles.

Erratic behavior can occur when obstacles are slightly too far apart to be consistently detected. However, if t_{Free} is set to twice the sensory horizon, the edges created by the two alternative models coincide, and the behavior is consistent despite intermittent detection of one or the other obstacle.

$$
\begin{aligned}
O &= \text{Threshold}(P);\\
T &= \text{DistanceTransformOf}(O);\\
S &= \text{Threshold}(T, t_{\text{Free}});\\
D &= \text{Threshold}(T, t_{\text{Dilate}});\\
D &+= S;\\
L &= \text{LabelConnectedComponentsIn}(D);\\
M,N &= \text{DistanceTransformOf}(L);\\
V &= \text{CountDistinctNeighbours}(N).
\end{aligned}
$$

2.4. *Selectively navigating GVD edges*

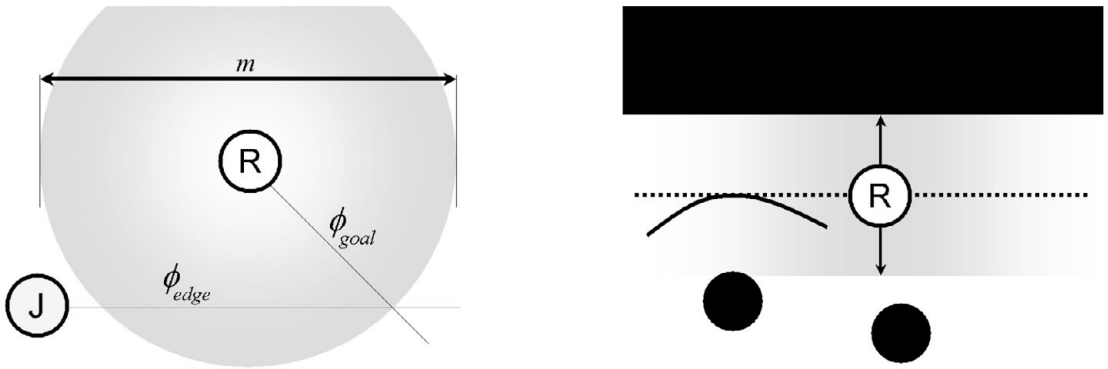
The robot is controlled by manipulation of the first derivative of the robot's rotational velocity, using a Behavioral Dynamics approach for the coordination of multiple behaviors, as described by Althaus and Christensen.[55] The reader is directed to the cited work, or to ref. [56], for a detailed description of this methodology.

However, here the GOTO behavior in the cited work is replaced with an EDGE TRAVERSAL behavior and the DOOR PASSING, CORRIDOR FOLLOWING, and WALL AVOIDANCE behaviors are removed. The OBSTACLE AVOIDANCE behavior is also redundant if the robot is navigating only on or near the edges of the approximated GVD.

Since only one or two behaviors are used, the primary benefit of the Behavioral Dynamics is elegant generation of the control signal for the robot, its rotational velocity $\dot{\phi}$. Since this is generated from the egocentric map images, an obvious and popular alternative would be further image-based processing to generate artificial *potential fields*,[57, 58] which would then determine robot motion.

The robot used for this work has an embedded control algorithm that accepts a desired rotational velocity that it attempts to realize. As the robot's physical response is slower than the rate of updates to $\dot{\phi}$, the robot follows a continuously curving path. The accuracy of the robot's adherence to the theoretical path is not critical to this paper, which is,

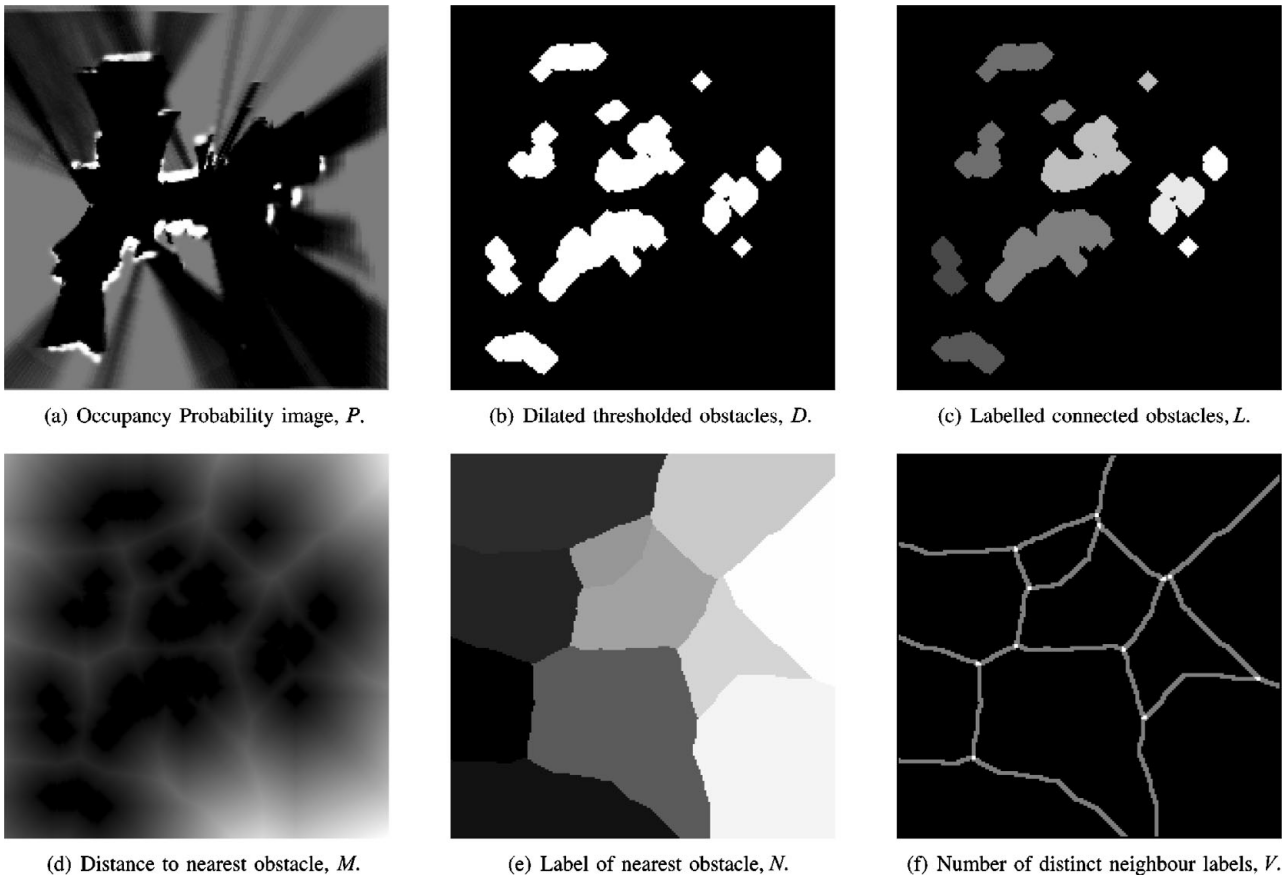
https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press



(a) Calculating ϕ_{goal}. The goal vector originates at a junction J, at an angle ϕ_{edge}, that was computed by averaging edge pixel bearings when the robot was at J. The robot has since moved to position R, forced away from the goal vector by obstacles. The current vector of travel, at angle ϕ_{goal}, is set to intersect the goal vector at a point that is maximally distant to J and at a distance r from the robot, where r is the radius of the egocentric map (diameter m). This method provides a good balance between obstacle avoidance and continued progress along the goal vector, rather than simply in the intended direction of travel.

(b) If a coastal navigation (wall-following) approach is used in open spaces, the robot may rapidly alternate between competing topological models when obstacles occur at the limit of sensor range. One model, indicated by the dotted line earlier, is applied when only the wall (coast) is visible. The alternative model (solid curved line) is the edge caused by the detection of another obstacle, in this case one of two circular obstructions. If the wall-following distance is set correctly, these two topologies overlap each other, stabilizing the robot's behavior regardless of which model is perceived.

Fig. 3. Careful handling of problematic situations prevents erratic behavior when traversing egocentrically modeled edges. Obstacles may force the robot off its intended vector of travel, or suddenly alter the perceived positions of topological edges.



(a) Occupancy Probability image, *P*.

(b) Dilated thresholded obstacles, *D*.

(c) Labelled connected obstacles, *L*.

(d) Distance to nearest obstacle, *M*.

(e) Label of nearest obstacle, *N*.

(f) Number of distinct neighbour labels, *V*.

Fig. 4. Computing an egocentric approximation of the GVD. The robot is always at the centre of these images. Note that higher probability of occupancy is associated with brighter pixels in P. Priors for P are retained from previous iterations using odometry to account for robot translation. Odometric errors are not estimated. Egocentric map orientation is invariant to robot orientation. Brighter pixels in M are more distant from obstacles. In V, white pixels are junctions with three or more different neighbor labels. Grey pixels have two different neighbor labels. Black pixels have 1.

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

instead, concerned with defining, identifying, and making navigational decisions.

EDGE TRAVERSAL is a development of the GOTO behavior attractor in ref. [55]. Instead of generating a single attractor in a particular goal direction, all GVD edge pixels in the egocentric image V are attractive to the robot. The strength of individual pixel's attraction reduces with increasing distance from the robot, and with increasing angular distance from the goal direction. Thus, GVD edge pixels close to the center of the map, in the desired direction of travel, have the greatest influence. Since EDGE TRAVERSAL navigates the robot only along traversable GVD edges (the graph is already pruned via dilation), this behavior also effectively performs obstacle avoidance. As a result, the goal direction need only be approximately specified. In the event that the current edge is very curved, the behavior will still drive the robot along the current edge, i.e., edge following takes priority over goal direction.

Let ϕ_{robot} represent the current heading of the robot. $V_{i,j}$ is a pixel at column i and row j of the Voronoi image V. The output of the EDGE TRAVERSAL function is $\dot{\phi}$, the robot's desired turn rate:

$$\dot{\phi} = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} \begin{cases} \dot{\phi}_{i,j} & \text{iff } V_{i,j} = \text{edge pixel} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\dot{\phi}_{i,j} = (-\lambda \sin \phi_{\text{relative}}) \cdot f_{\text{radial}}(i, j) \cdot f_{\text{angular}}(i, j) \tag{6}$$

$$f_{\text{radial}}(i, j) = \mathrm{e}^{-\gamma \sqrt{(c-i)^2+(c-j)^2}} \tag{7}$$

$$f_{\text{angular}}(i, j) = \pi - |\phi_{i,j} - \phi_{\text{goal}}| \tag{8}$$

$$\phi_{\text{relative}} = \phi_{\text{robot}} - \phi_{i,j} \tag{9}$$

$$\phi_{i,j} = -a \tan 2(c - j, i - c). \tag{10}$$

Note that the function is computed as the sum of output for all pixels within V and biased toward the global goal direction ϕ_{goal}. The term $f_{\text{radial}}(i, j)$ ensures that closer edge pixels are more influential. γ is a constant chosen to scale pixel distances within V. $f_{\text{angular}}(i, j)$ reduces the strength of attractors not in the direction of the goal. λ scales the output of EDGE TRAVERSAL to match the physical capabilities of the robot.

2.5. *Developing topological instructions*

Each topological instruction permits selective navigation of a single edge of the GVD. Therefore, normally each instruction is initialized at a GVD junction, and guides the robot to an adjacent GVD junction. This allows instructions to be executed in series. At the moment the current instruction is completed, the robot's pose is taken to be a reference point for the next instruction. The instructions provided in this paper only allow travel to junctions; once close to the goal, it is assumed that application-specific behaviors will be used instead.

By terminating each instruction at a reference point perceived in the world—i.e., a GVD junction—rather than at an absolute position, error is not accumulated between instructions. (Of course, a topological error such as transiting the wrong edge will immediately introduce significant error). It is also useful to define a perceived, rather than absolute, reference direction: Angular error has a more significant effect than positional error. So, in this paper, the direction from which the robot approached the current junction (hereafter, "back-trail") is used as the reference direction ϕ_{ref}.

The back-trail bearing ϕ_{ref} is computed as the circular mean bearing of all egocentric map pixels the robot has recently occupied. If the robot is at a GVD junction, ϕ_{ref} is effectively computed relative to that junction. The robot's heading ϕ_{robot} at the moment it reaches a GVD junction is not a good indication of the direction it has come from, since the edges of the GVD are usually highly curved near junctions. ϕ_{ref} gives a better long-term estimate of the direction from which the robot has approached the junction. This has the added benefit that it makes more sense to the human director.

The output of a topological instruction initialized at junction J is ϕ_{edge}, the orientation of a vector Z originating at J. The input to the EDGE TRAVERSAL behavior, ϕ_{goal}, is derived from ϕ_{edge}. EDGE TRAVERSAL interprets the parameter ϕ_{goal} liberally, causing the robot to move along edges in this direction while avoiding obstacles. Where multiple edges are available, the edge whose bearing is closest to ϕ_{goal} will be followed. Since the robot will not move accurately along the specified vector Z, ϕ_{goal} is continually recomputed and set to values that would cause the robot to intercept Z at the edge of the egocentric map [see Fig. 3(a)]. Two different instruction types are described later. They differ in the way that ϕ_{edge} is calculated.

2.6. *Relative bearing instructions*

The "relative bearing" instruction specifies that the robot should move in a straight line originating wherever the instruction is initialized at an angle ϕ_{edge}. The parameter ϕ_{offset} is a specified offset to the robot's direction of travel at initialization; therefore, $\phi_{\text{edge}} = \phi_{\text{ref}} + \pi + \phi_{\text{offset}}$. The additional π transforms the back-trail into the robot's direction of travel.

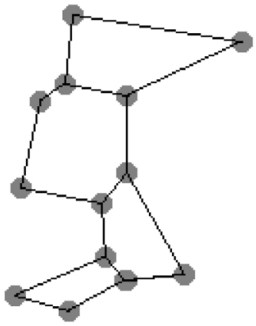
This instruction need not be initialized at a junction, if the robot's current heading ϕ_{robot} is used in place of the robot's direction of travel ($\phi_{\text{ref}} + \pi$). It is, therefore, suitable for the first part of a navigation exercise. As with hand-waving human convention, the first instruction is based on the instructee's pose. The relative bearing instruction is completed when the robot arrives at any GVD junction that is at least n meters from the point of initialization. Thus, to summarize, two parameters are required, n and ϕ_{offset}. These are supplied by the director.

2.7. *Edge enumeration instructions*

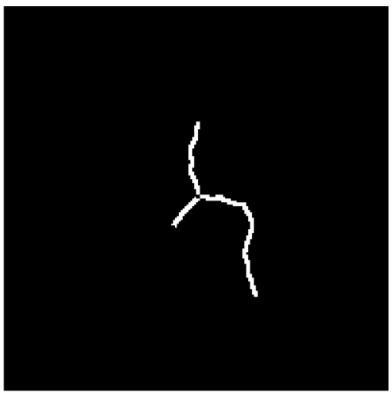
The "edge enumeration" instruction is an enhancement of the "relative bearing" instruction. The motivation behind this instruction type is that selection of edges by bearing is somewhat ambiguous when two or more edges are

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

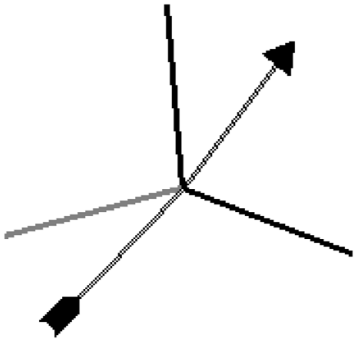
(a) Dilated obstacles, D.



(b) Resultant egocentric Voronoi graph approximation

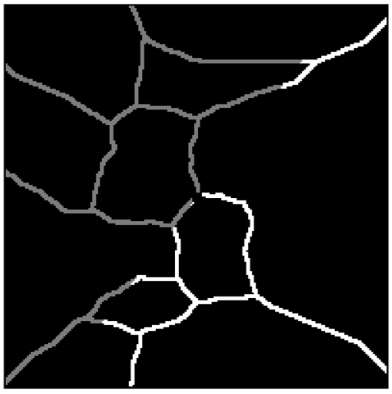


(c) Edges within image V that are incident on the vertex closest to the robot.



(d) The bent arrow represents the robot's current heading (the head) and back-trail (the end with the flights). Thick lines denote edges incident at the local vertex; the reference edge is grey.

(e) Magnitude of the output of the EDGE TRAVERSAL function, for each pixel in the egocentric map. The robot is approaching a junction, and has no preference for either path as yet.



(f) Sign of the output of the EDGE TRAVERSAL function, for each pixel in the egocentric map.

Fig. 5. Enumerating, selecting, and navigating edges of the egocentric Voronoi graph. Note that these images represent a different scenario to those in Fig. 4.

close. The reference bearing may also be difficult to estimate beforehand, especially if the junction position is not perceived accurately. Moreover, it is easier and more natural for a human director to specify edges by counting from a reference edge. For example, when driving a car and navigating a roundabout, one might "take the second exit."

When initialized, the edge enumeration instruction computes the circular mean bearing of GVD edge pixels in V relative to the junction closest to the robot, for each edge k incident at that junction. This is illustrated in Figs. 5 and 6. Note that these bearings are calculated relative to the position of the junction, not the robot. One edge is assumed to be the edge the robot has just traversed. This is identified by minimizing the circular angular difference between edge mean bearings ϕ_{edge_k} and the back-trail bearing ϕ_{ref}. Once the reference edge bearing $\phi_{\text{edge}_{\text{min}}}$ is known, it is a simple matter to count clockwise or anticlockwise to find the selected edge bearing $\phi_{\text{edge}_{\text{selected}}}$. Thus, $\phi_{\text{edge}} = \phi_{\text{edge}_{\text{selected}}}$, and the selected edge will be traversed even if further along it subsequently twists around to a new direction.

In all other respects, the Edge Enumerating instruction is identical to the Relative Bearing instruction. The instruction is completed when the robot reaches a GVD junction. Since the Edge Enumerating instruction relies on identification of the previously transited edge and the robot being at a GVD junction when initialized, it cannot be the first instruction in a series.

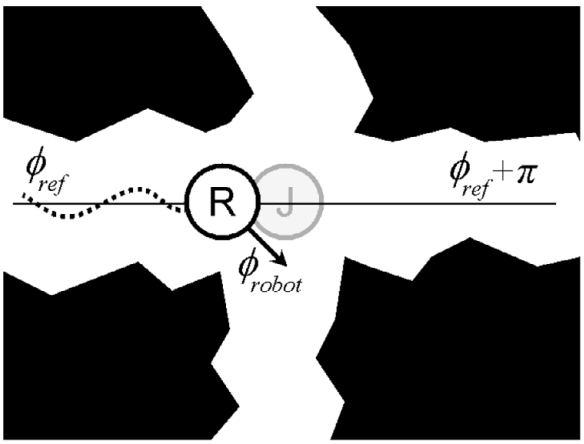
2.8. Other instructions

More complex and natural instructions could be created, such as "take the second left"—meaning go straight over the first junction, and left at the second. But these can be created by chaining the atomic instructions already discussed. Useful and more sophisticated instructions could be created by expanding the vocabulary to include recognition of landmarks that are distinctive, if not, otherwise, navigationally relevant. However, recognition of landmarks described using an abstract language is a problem domain in its own right, and beyond the scope of this paper.

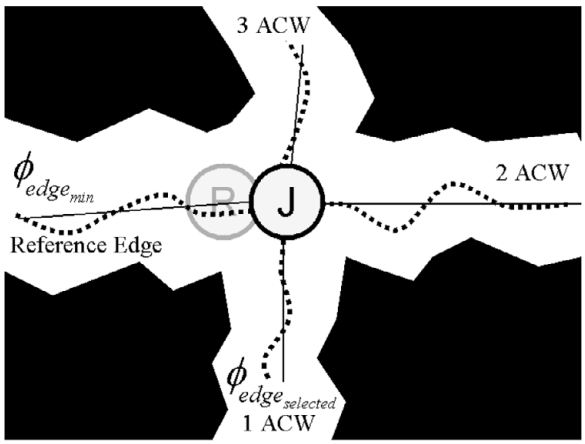
3. Experiment

In 21 trials, the robot was tasked with navigating from the author's desk to every other workplace in the laboratory, via a number of different routes, using different combinations of relative bearing and edge enumeration instructions provided by the author and other students. The laboratory environment is not a single cluttered concavity, so a combination of goal vector and obstacle avoidance behavior would not result in successful or efficient navigation (see Figs. 7, and 8). The

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

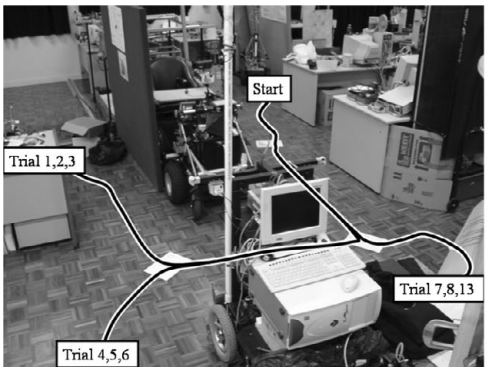


(a) ϕ_{ref} is computed by averaging the bearing of pixels in the egocentric map that the robot has recently occupied, relative to the robot. The dotted line represents these pixels and the solid line depicts ϕ_{ref}. Rotating ϕ_{ref} by π transforms it into the robot's current direction of travel. This measure is more robust than the robot's current orientation (ϕ_{robot}) which varies significantly in the short term, especially near GVD junctions.



(b) At the same junction, the bearing of GVD edge pixels relative to the junction is computed. The mean junction-relative bearing of all edges' pixels is then calculated. Here, four edges are incident at a junction. The edge whose average pixel bearing $\phi_{\text{edge}_{\text{min}}}$ is closest to ϕ_{ref} is found and labeled as the *reference edge*. All other edges are ordered by average pixel bearing so that one can be selected, giving us $\phi_{\text{edge}_{\text{selected}}}$.

Fig. 6. Enumerating the edges around a junction, and the direction of travel of the robot. The aim is to define a goal vector with an origin at the junction and orientation equal to $\phi_{\text{edge}_{\text{selected}}}$. Here, we select the first edge anticlock wise (ACW) of the reference edge.



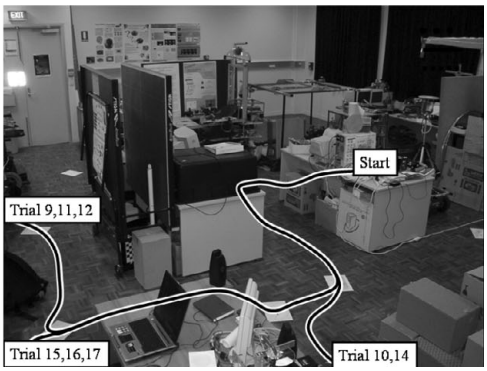
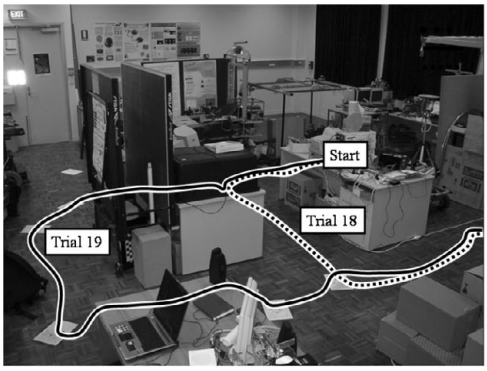
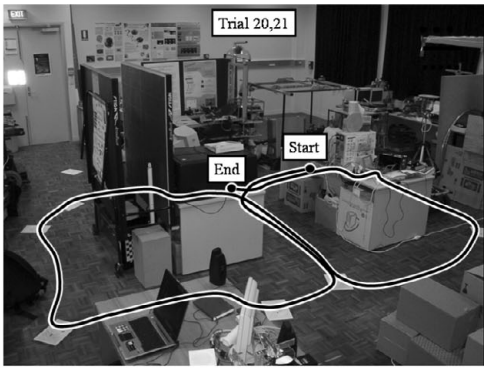


Fig. 7. Approximate path of robot when undergoing trials, overlaid on photographs of the experiment environment. In each trial, the robot navigated from the start point (a variable position close to the author's desk) to a goal location. Junctions of the GVD are marked with sheets of white A4 paper on the laboratory floor. This is purely for the author's convenience in presenting these results in the paper and demonstrating the robot.



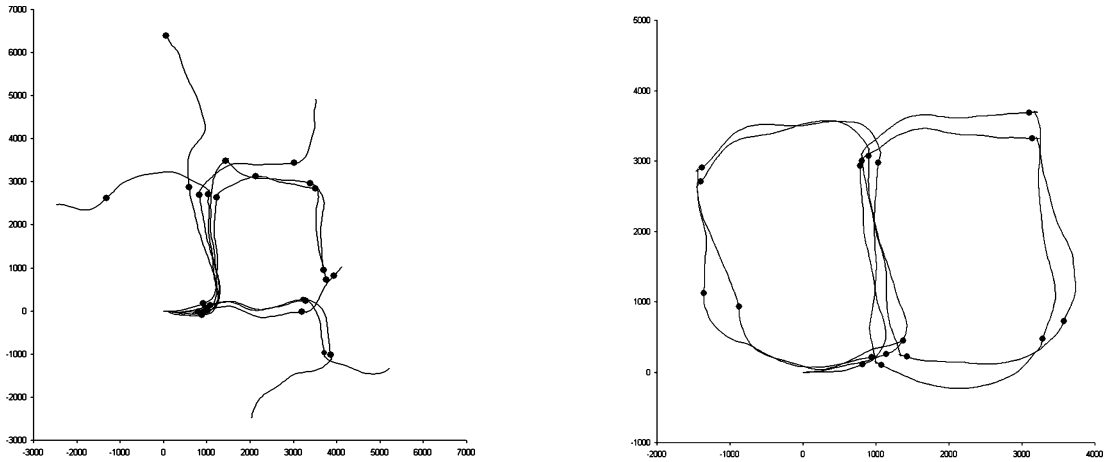
(a) Alternate paths to laboratory exit.



(b) Figure-8 route.

Fig. 8. Approximate path of robot in longer trials.

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press



(a) Trials 1, 4, 7, 11, 12, 14, 16, and 18. (b) Figure-8 route, trials 20 and 21.

Fig. 9. Robot paths and perceived junctions, from odometry. Axes show distance in millimetres. Note that along each route, each time a junction is reached, it is detected exactly once. No false junctions are detected. No junctions are omitted, except in the deliberately ambiguous case where two junctions are sometimes merged.

final two instruction sets specified a Fig. 9 route. The robot moved without pause during the trials at a wheel speed of approximately 10–25 cm/s. The robot was not required to detect or recover from erroneous instruction, or to recover from erroneous interpretation of accurate instructions. These problems are beyond the scope of this paper and have been previously researched (e.g., refs. [41] and [42]).

Table I enumerates instruction types implemented for these trials. Table II details the use of these instructions in the trials. Note that the Relative Bearing instructions can make use of reasonably accurate bearing information, but we have chosen to limit precision to $\pi/2$, i.e., Left, Right, or Forward, to show that the human director need not estimate GVD edge angles accurately.

4. Results

Figures 7, and 8 show the routes covered by the robot and the experimental environment. In these images, the GVD junctions are marked on the laboratory floor for the benefit of the reader—they are not used by the robot. They were also used to explain the GVD concepts to other students who subsequently demonstrated the ability to direct the robot.

Twenty of the 21 trials resulted in successful navigation. The total distance covered was just over 200 m. Trial 10 failed due to the engineered ambiguity of the GVD, as illustrated in Fig. 1. At one location on the robot's path, either one or two junctions are perceived, since the distance between them is exactly the threshold required for the junctions to be separate. This was achieved by using cardboard boxes to modify the environment. In such situations, sensor error determines the actual mapping. In trials 11 onwards, the same area is navigated successfully using "relative bearing" instructions and approximately specifying the minimum distance to the junction at which the next instruction should be applied. In trial 11, one junction is perceived at the ambiguous location, with four edges incident. In trial 12, two junctions are perceived here, each having three edges.

Overall, the robot was successfully navigated to all accessible areas of the laboratory and, in trials 20 and 21, a Fig. 8 route was executed to demonstrate a longer series of instructions (46 m total for the Fig. 8 routes). One trial was unsuccessful. Since it was not necessary to measure the environment to provide the instructions and the robot does not use SLAM, no metric map of the environment is available. However, Fig. 9 shows the robot's path from odometry and detected junction positions, for selected trials. Before each trial, the robot was positioned near the author's desk facing approximately in the direction of the first goal junction: precise positioning was unnecessary.

Some problems were experienced during testing prior to reported trials due to the limitations of the scanning laser-range sensor. In particular, planar sensing assumes that the shape of obstacles does not vary greatly with height and some surfaces are not detectable. However, traversal of the GVD minimizes such problems since distance to all obstacles is maximised. Also, the method of GVD junction detection is robust to variations in sensor performance and odometric error. In the few places where scanning sensing was inadequate, artificial obstacles were added below higher projections, to prevent collision. Similarly, a few surfaces that do not reflect the laser signal were covered with cardboard to ensure the robot could sense its environment satisfactorily. Note that these modifications were made to mitigate sensor limitations, not processing flaws. In several trials, the robot encountered dynamic obstacles in the form of passing people, but navigation was unaffected due to application of Bayes' rule in the egocentric occupancy map. A video of experiments similar to trials 20 and 21 is available at `http://thecyberiad.net/videos/` to better illustrate the operating conditions.

5. Conclusions

The results show that it is possible for robots and humans to share useful topological navigational concepts that both can perceive directly. This enables reliable and efficient navigation of the robot in a realistic environment without a prior mapping operation, which is very useful. The topological instructions provided to the robot are

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

Table I. Instructions.

Shorthand	Type	Description
F*n*	Relative bearing	$\phi_{\text{offset}} = 0$. Follow edges closest to the direction the robot is initially facing, for at least n meters, until a junction is reached. Default $n = 2r$, where r is the diameter of the robot.
L*n*	Relative bearing	$\phi_{\text{offset}} = \pi/2$. Follow edges closest to the direction the robot is initially facing $+ 90^{\circ}$, otherwise as previously.
R*n*	Relative bearing	$\phi_{\text{offset}} = -\pi/2$. Follow edges closest to the direction the robot is initially facing $-$ 90 degrees
*m*CW	Edge enumerating	Count n edges clockwise from the back-trail edge; follow the selected edge until a junction is reached, e.g., 1CW means take the first edge clockwise from the back-trail edge.
*m*ACW	Edge enumerating	As previously, but counting anticlockwise.

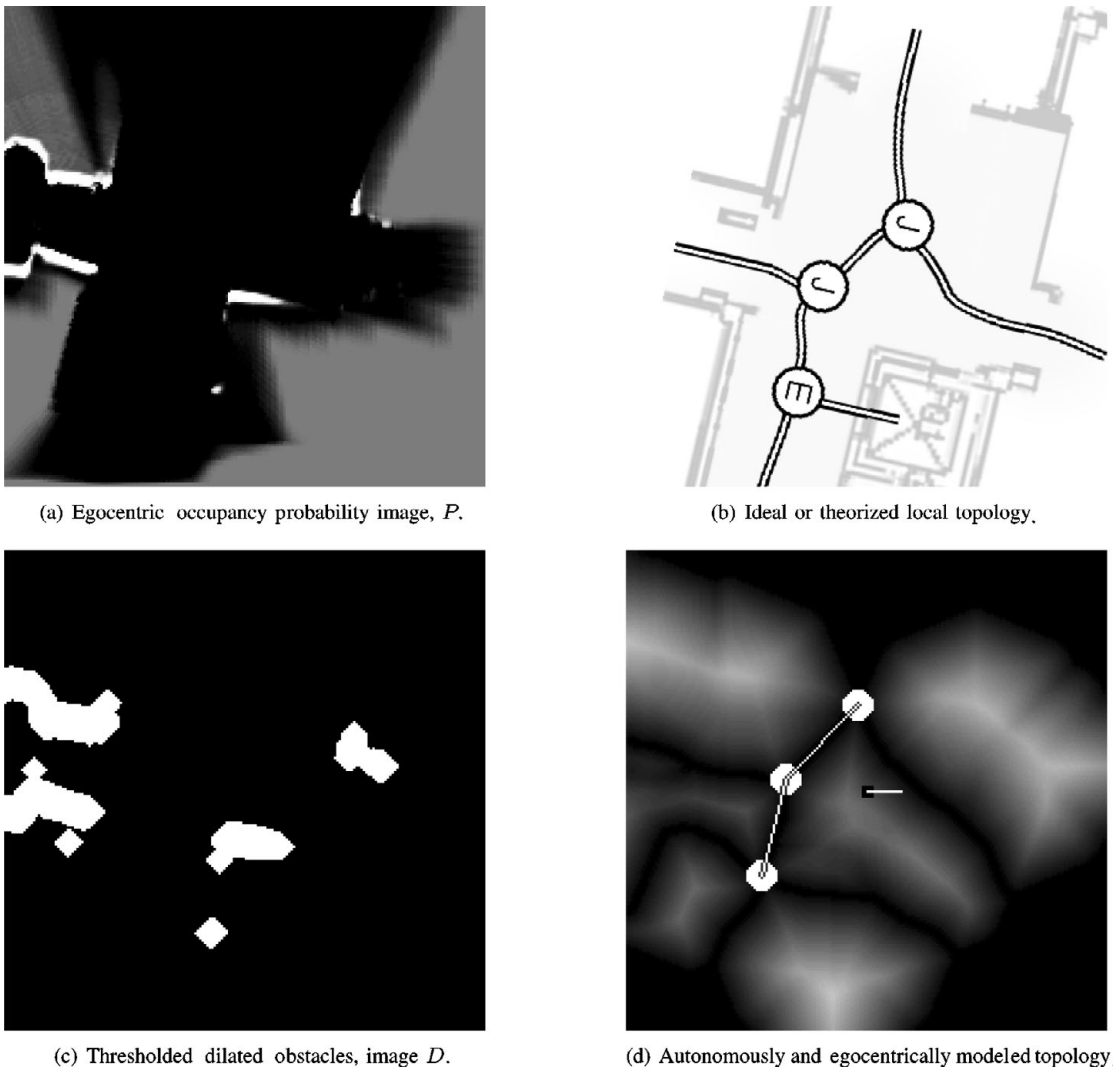
Table II. Experimental trials.

Trial	Goal	Instructions	Comments
01	P's desk	F, 1ACW, 1ACW, 1ACW	Forward until junction, take first path on right, until junction . . . and so on
02	P's desk	F, 1ACW, 1ACW, 1ACW	Repeat
03	P's desk	F, F, R, R	Same route as trials 01 and 02
04	S's desk	F, F, R, L	First part same as trial 03, then diverges
05	S's desk	F, 1ACW, 1ACW, 1CW	Same route as 04
06	S's desk	F, 1ACW, 1ACW, 1CW	Repeat
07	F's desk	F, 1ACW, 1CW, 1ACW	Short route
08	F's desk	F, 1ACW, 1CW, 1ACW	Repeat
09	F's desk	F, 1CW, 1ACW, 1ACW	Longer route
10	F's desk	F, 1CW, 1ACW, 1ACW	Longer route, failed: Went to G's desk
11	F's desk	F, 1CW, R1.5, 1ACW	Longer route like trial 10, better instructions
12	F's desk	F, 1CW, R1.5, 1ACW	Repeat trial 11, different modeling of ambiguous junction, still success
13	F's desk	F, F, L, R	Short route, different instructions
14	G's desk	F, 1CW, F1.5	To show can still get here
15	Z's desk	F, 1CW, R1.5	
16	R's desk	F, 1CW, R1.5, 1CW	
17	R's desk	F, 1CW, R1.5, 1CW	Repeat
18	Laboratory exit	F, 1CW, L, 1ACW	Short route
19	Laboratory exit	F, 1ACW, 1CW, 1CW, F, 1ACW	Longer route
20	Author's Desk	F, 1CW, R1.5, 1ACW, 1ACW, 1ACW, 1CW, 1CW, 1CW	Figure-8 route back to start
21	Author's Desk	F, 1CW, R1.5, 1ACW, 1ACW, 1ACW, 1CW, 1CW, 1CW	Repeat

Table III. Essential features of a topologically directed navigation language.

Feature	Comments
Mutual perception	To conveniently exchange navigational information, both parties (robot and human) must be able to perceive the same topology reliably and directly.
Local perception	Many topological mapping systems generate erroneous topology until the map is largely complete and accurate. To follow directions without a complete map, local sensing must be sufficient.
General relevance	It must be possible to efficiently describe routes in all conceivable situations to all goal locales, because the chosen topological landmarks are present in all environments.
Efficiency	The chosen topological landmarks must occur frequently enough to specify routes accurately, but not so frequently that the plan becomes cumbersome or complicated.

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press



(a) Egocentric occupancy probability image, P.
(b) Ideal or theorized local topology.
(c) Thresholded dilated obstacles, image D.
(d) Autonomously and egocentrically modeled topology.

Fig. 10. An example of egocentric modeling in a different environment—here the robot crosses the foyer of our building. The robot's perception of local topology agrees with the author's estimation, shown overlaid on the building floor plan. Two junctions (J) exist in the foyer, a third due to a path into the elevator (E). Note that despite detecting only fragments of the foyer walls and nearby corridors, the robot has still modeled local topology accurately. Topological modeling errors occur only if a significant obstacle is entirely undetected. Small perturbations in obstacle shape, and erroneous small gaps between obstacles, have no effect on topology. Small variations in the shape or position of topological edges have a momentary effect on robot motion, but do not affect the incremental interpretation of topological directions.

parsimonious and intuitive, especially when explained in terms of the robot's spatial affordances. The instructions correspond directly with the type of directions that humans give each other when navigating unknown environments, albeit the latter employ a richer vocabulary of symbols (not just junctions and edges). For comparison, topological navigation instructions generated expressly for the purpose of easy human comprehension are discussed in this paper.[1] Since the instructions use minimal or no metric data, it is possible for a human director to produce them from memory of the scene, greatly increasing the utility of the system.

The pure topological approach to navigation demonstrated here is different to other goal-directed navigation in the literature. Lack of detailed metric information does make the system vulnerable to gross error. The same attribute simultaneously offers great convenience. It is interesting that complex robot perception, rather than more accurate mapping, makes possible a new mode of navigation in unmapped environments.

The experiments reported in this paper are not simply GVD following because the robot does not have a map of the global GVD and the physical structure that generated it. Instead, it is given only the part of the topology that it must transit, and must incrementally perceive and ground equivalent features in the real world, simultaneously navigating directly to the goal. This is more difficult than localization in a fully mapped environment and has not previously been demonstrated.

5.1. Sources of error

The proposed method by which the topological features—junctions and paths—are perceived is robust to most variations in sensing, but two types of failure are possible. Firstly, total failure to sense *any* part of a large obstacle dramatically changes the perceived position, or existence, of associated junctions. However, sparse and/or inaccurate detection of parts of an obstacle is sufficient to localize junctions with reasonable accuracy, as demonstrated in our experiments (See Fig. 10).

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

The second source of error or ambiguity is merging of adjacent junctions in the GVD that may be considered a single junction, if close enough. The source of error is the definition of "close enough." In the reported experiments, one such situation (see Fig. 8) was artificially created with cardboard boxes, but, in fact, it was still possible to reliably navigate this part of the laboratory by specifying the approximate minimum distance to the junction at which the next instruction should be applied.

It is worth mentioning that misinterpretation of the instructions does lead to a gross navigational error that is undetectable. This also happens to people, who recover either by realizing that the goal has not been reached or by using more detailed localization based on a richer vocabulary of landmarks.

5.2. Future directions

Navigation may also be unsuccessful if the topology of the world changes between the moment the robot is instructed, and completion of navigation. Topological changes may be caused by simple acts such as opening and closing doors.

Possible solutions to this problem include use of extra geometric information to confirm topological interpretations, and querying the human director during navigation. But, ideally, the robot should be fully autonomous after initial instruction, and geometric information is inconvenient to obtain.

Instead, in later experiments, the authors used visual landmarks to identify specific GVD junctions by appearance, ignoring intermediate vertices. Provided that visual landmarks are not missed, this makes the system robust to changes in topology. However, a large vocabulary of natural landmarks is required to allow instruction of the robot in a variety of environments. These results will appear in a future paper.

5.3. Summary of prerequisites

A number of key features must be in place in order that the robot can execute a series of meaningful topological navigation instructions in an unknown environment. The first is robust local mapping and planning given incomplete knowledge of the environment. This was achieved using an egocentric metric mapping approach to computation of the GVD, conceptually similar to Beeson and Kuipers' Local Perceptual Mapping (LPM).[6, 7] Note that the proposed algorithm only makes judgements about the characteristics of the robot's immediate environment, for which maximal information is available. Without accurate sensing of local topology, global mapping is necessary before goal-directed navigation can occur.

A second key feature is the choice of shared symbols that are perceived and grounded by both the robot and the operator, in this case GVD junctions and edges. These are present in all environments[44] unlike doorways, corners, and other more specific geometric landmarks. The GVD concepts modeled as spatial affordances are also highly relevant to navigation, and, in this paper, defined with reference to the physical capabilities of the robot (any passage through which the robot can physically pass is an edge, and any position at which two paths are available is a junction). It is easy for an operator to make good judgements about where the robot can and cannot travel, since the same skills are necessary for a number of everyday tasks such as driving and tool use.

A third key feature is the near-absence of metric information, instead relying on topology that humans can readily perceive. If it were necessary to add accurate metric information, considerable labor would be required to provide such data and it would be laborious to translate into an acceptable digital format without the use of a computer. In contrast, using a standard speech analysis program, the proposed system could easily interpret spontaneous verbal instructions using the robot's onboard computer. It is interesting that a remote operator can also generate navigation instructions, from memory of the environment. Many of the instruction chains provided for the reported experiments were generated by the author while not viewing the experimental environment.

Finally, the fact that the robot's internal topological map formed during navigation is in the same format as the original topological directions means that repeat navigation to any previously visited location may be requested.

5.4. Applications

In some applications, complete and accurate mapping is effectively part of the mission, but, in many situations, it is merely necessary to move to a goal location as quickly as possible. In any nonenclosed environment, a complete mapping will necessarily take an unreasonable amount of time. Many SLAM experiments are conducted by manually driving the robot, or by altering the environment to close off areas that are not desired to be explored. The proposed approach, in conjunction with existing SLAM techniques, would permit guided autonomous exploration experiments to focus on the areas that do indeed require mapping, without modification of the environment.

Acknowledgment

This work was supported by the IEEE.

References

1. W. Maass, P. Wazinski and G. Herzog, "Vitra Guide: Multimodal Route Descriptions for Computer Assisted Vehicle Navigation," *Proceedings of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)* (1993), pp. 144–147.
2. M. Levit and D. Roy, "Interpretation of spatial language in a map navigation task," *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **37**(3), 667–679 (2006).
3. S. Blisard and M. Skubic, "Modelling Spatial Referencing Language for Human–Robot Interaction," in *Proceedings of the 2005 RO-MAN Workshop* (2005) pp. 698–703.
4. I.-P. Park and J. R. Kender, "Topological direction-giving and visual navigation in large environments," *Artif. Intell.* **78**(1–2), 355–395 (1995).
5. D. Rawlinson and R. A. Jarvis, "Simple yet Effective Visuo-Spatial Topological Mapping," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2006) pp. 2766–2771.
6. P. Beeson, N. K. Jong and B. Kuipers, "Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2005).

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

7. B. Kuipers, J. Modayil, P. Beeson, M. MacMahon and F. Savelli, "Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2004) pp. 4845–4851.
8. P. Beeson, M. MacMahon, J. Modayil, J. Provost, F. Savelli and B. Kuipers, "Exploiting Local Perceptual Models for Topological Map-building," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Reasoning with Uncertainty in Robotics (RUR)* (2003) pp. 15–22.
9. R. F. Wang and E. S. Spelke, "Updating egocentric representations in human navigation," *Cognition* **77**(3), 215–250 (2000).
10. R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," **In**: Autonomous Robot Vehicles (I. J. Cox and G. T. Wilfong, eds.) (Springer-Verlag, 1990) pp. 167–193.
11. P. S. Maybeck, *Stochastic Models, Estimation and Control* (Academic, New York, 1979).
12. S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *Int. J. Robot. Res.* **20**, 335–363 (2001).
13. S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hhnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot Minerva," *Int. J. Robot. Res.* **19**(11), 972–999 (2000).
14. A. Dempster, N. Laird and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc., Ser. B* **39**(1), 138 (1977).
15. M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," *Proceedings of the AAAI National Conference on Artificial Intelligence* (2002) pp. 593–598.
16. R. S. N. Tomatis and I. Nourbakhsh, "Simultaneous Localization and Map Building: A Global Topological Model with Local Metric Maps," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2001) pp. 421–426.
17. C. Stachniss, D. Hahnel and W. Burgard, "Exploration with Active loop-closing for fastSLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2004) pp. 1505–1510.
18. R. Sim and G. Dudek, "Examining Exploratory Trajectories for Minimizing Map Uncertainty," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Reasoning with Uncertainty in Robotics (RUR)* (2003) pp. 69–76.
19. R. Grabowski, P. Khosla and H. Choset, "Autonomous Exploration via Regions of Interest," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2003) pp. 1691–1696.
20. A. Morris, D. Silver, D. Ferguson and S. Thayer, "Towards Topological Exploration of Abandoned Mines," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2005) pp. 2117–2123.
21. B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robot. Auton. Syst.* **8**, 47–63 (1991).
22. C. Stachniss, O. M. Mozos and W. Burgard, "Speeding up Multi-Robot Exploration by Considering Semantic Place Information," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2006) pp. 1692–1697.
23. C. Stachniss, G. Grisetti and W. Burgard, "Information Gain-Based Exploration Using Rao-Blackwellized Particle Filters," *Proceedings of Robotics: Science and Systems* (RSS) (2005).
24. M. Salichs and L. Moreno, "Navigation of mobile robots: Open questions," *Robotica* **18**, 227–234 (2000).
25. J. E. Guivant, F. R. Masson and E. M. Nebot, "Simultaneous localization and map building using natural features and absolute information," *Robot. Auton. Syst.* **40**, 79–90 (2002).
26. J. Folkesson and H. I. Christensen, "Robust SLAM," *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles (IAV)* (2004).
27. Y. Wang, M. Huber, V. N. Papudesi and D. J. Cook, "User-Guided Reinforcement Learning of Robot Assistive Tasks for an Intelligent Environment," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2003).
28. A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (1994).
29. P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.* **2**(4), 100–107 (1968).
30. B. Kuipers, "The spatial semantic hierarchy," *Artif. Intell.* **119**(1–2), 191–233 (2000).
31. S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.* **99**(1), 21–71 (1998).
32. B.-Y. Ko and J.-B. Song, "Real-Time Building of a Thinning-Based Topological Map with Metric Features," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2004).
33. B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis and H. Choset, "The hierarchical atlas," *IEEE Trans. Robot. Autom.* **21**(3), 473–481 (2005).
34. J. Modayil, P. Beeson and B. Kuipers, "Using the Topological Skeleton for Scalable Global Metrical Map-Building," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2004).
35. N. Tomatis, I. Nourbakhsh and R. Siegwart, "Hybrid simultaneous localization and map building: A natural integration of topological and metric," *Robot. Auton. Syst.* **44**, 3–14 (2003).
36. A. Poncela, E. Perez, A. Bandera, C. Urdiales and F. Sandoval, "Efficient integration of metric and topological maps for directed exploration of unknown environments," *Robot. Auton. Syst.* **41**, 21–39 (2002).
37. W. Burghard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artif. Intell.* **1**(53), 3–55 (2000).
38. S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja and K. Schwehr, "Recent Progress in Local and Global Traversability for Planetary Rovers," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2000).
39. C. Urmson, J. Anhalt, M. Clark, T. Galatali, J. P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P. Koon, K. Peterson, B. Smith, S. Spiker, E. Tryzelaar and W. Â. Whittaker, "High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-04-37 (2004).
40. A. S. P. Tompkins and D. Wettergreen, "Mission-level path planning and re-planning for rover exploration," *Robot. Auton. Syst.* **54**, 174–183 (2006).
41. I. Nourbakhsh, R. Powers and S. Birchfield, "Dervish: An office-navigating robot," *Artif. Intell. Mag.* **16**(2), 53–60 (1995).
42. D. Hinkle, D. Kortenkamp and D. Miller, "The 1995 robot competition and exhibition," *Artif. Intell. Mag.* **17**(2), 31–45 (1996).
43. H. Choset and J. Burdick, "Sensor based planning. part i: The generalized Voronoi graph," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (1995) pp. 1649–1655.
44. H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized Voronoi graph," *Int. J. Robot. Res.* **19**(2), 96–125 (2000).

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press

45. H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, **17**(2), 125–136 (2001).
46. J. Gibson, *The Ecological Approach to Human Perception* (Houghton-Mifflin, Boston, MA, 1979).
47. T. Bayes, "Studies in the history of probability and statistics: Ix. Thomas Bayes' essay towards solving a problem in the doctrine of chances," *Biometrika* **45**, 296–315 (1763/1958) [Bayes' essay in modernized notation].
48. A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robot. Autom.* **3**, 249–265 (1987).
49. A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Comput.* **22**(6), 46–57 (1989).
50. R. Mahkovic and T. Slivnik, "Constructing the generalized local Voronoi diagram from laser range scanner data," *IEEE Trans. Syst., Man, Cybern. A, Syst. Hum.* **30**(6), 710–719, (2000).
51. H. Blum, *A Transformation for Extracting New Descriptors of Shape* (MIT Press, Cambridge, MA, 1979).
52. A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. Assoc. Comput. Mach.* **13**, 471–494 (1966).
53. R. A. Jarvis, "Collision-Free Path Planning in Time Varying Environments," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (1989), pp. 99–106.
54. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision* (Addison-Wesley Longman, Reading, MA, 1992) vol. 1.
55. P. Althaus and H. Christensen, "Behaviour coordination in structured environments," *Adv. Robot.* **17**(7), 657–674 (2003).
56. G. Schner and M. Dose, "A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion," *Robot. Auton. Syst.* **10**, 253–267 (1992).
57. O. Khatib, "Real-time obstacle avoidances for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–98 (1986).
58. E. Rimon and D. Koditscheck, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.* **8**(5), 501–518 (1992).

https://doi.org/10.1017/S026357470700375X Published online by Cambridge University Press