Mapping in unknown graph-like worlds

Gregory Dudek[†], Paul Freedman[‡], Souad Hadjres

[†] Centre for Intelligent Machines,

School of Computer Science,

McGill University,

3480 University St., Montreal, Québec, Canada H3A 2A7

[‡] Centre de recherche informatique de Montréal

1801 McGill College Avenue, Suite 800, Montréal, Québec, Canada H3A 2N4

Abstract

We consider the problem of constructing a map of an unknown environment by an autonomous agent such as a mobile robot. Because accurate positional information is often difficulty to assure, we consider the problem of exploration in the absence of metric (positional) information. Worlds are represented by graphs (not necessarily planar) consisting of a fixed number of discrete places linked by bidirectional paths. We assume the robot can consistently enumerate the edges leaving a vertex (that is, it can assign a cyclic ordering). A mobile robot is assigned the task of creating a topological map, i.e. a graph-like representation of the places in the world and their connectivity, by moving from place to place along the paths it encounters. It can detect edges and count them, but cannot directly sense the labels associated with a place or an edge. In principle, this types of representation could be used for non-spatial environments such as computer networks.

We present an approach to the exploration of unknown environments for which local sensing information alone is insufficient to distinguish distinct places, based simply on the structure of the world itself. Places are identified by a non-unique *signature* and by using a collection of such non-unique local signatures, an *extended* signature may be constructed which determines the robot's position (although in certain 'degenerate' worlds additional information is required). We describe the "exploration tree" as a representation of a collection of alternative descriptions of the environment. In addition, heuristics are presented that can accelerate the search for the correct world model.

1 Introduction

We are interested in the problem of automatically learning a map of an unknown environment using an autonomous agent such as a mobile robot. Because positional information can sometimes be difficult to maintain, we consider the feasibility of map construction in the complete absence of quantitative positional information. In the research to be described here, worlds are represented by graphs (not necessarily planar) consisting of a fixed number of discrete places linked by bidirectional paths. A mobile robot is assigned the task of creating a topological map, i.e. a graph-like representation of the places in the world and their connectivity, by moving from place to place along the paths it encounters. Note that topological representations can also be used to represent the connectivity of environments where quantitative spatial information is not natural, such as the connectivity of computer networks.

A prerequisite for map construction is ability to solve the *place identification* problem: when visiting a given place in the world, how can the robot determine whether or not it was already visited and, if so, to what previously-seen place does it correspond?

This task appears to be simple given the idealized robot with error-free (albeit limited) perceptual capacities that is reasonably common in the theoretical literature. In practice, the progressive accumulation of positional error, even with sensor feedback, makes the construction of a map based on an absolute metric coordinate system problematic. Furthermore, the definition of a map in terms of specific places or landmarks facilitates person-machine interaction; it is much easier to specify a task in terms of a place (eg. take the mail down the hall to the third office, perform a measurement on the North side of the lander, etc.) than at "x-y coordinate (976,436)".

Various approaches to robot localization using sensory information have been developed but most either depend on a good prior estimate of the robot's position and/or a sufficiently feature-rich environment [Moravec 1988; Cox 1989; Elfes 1987; MacKenzie and Dudek 1994; Dudek, Romanik and Whitesides 1995; Dudek and Zhang 1996]. In practice, some places may be difficult to uniquely and accurately localize (due to sparse or noisy sensor data, occlusion or interference with respect to GPS systems, accuracy considerations, etc.) and so it is important that autonomous systems be capable of functioning in the presence of non-unique signature information. As such, we have chosen to study an extreme case of sensing error in the context of exploration and mapping by assuming that the robot is equipped with limited sensing capabilities such that local sensing information alone is insufficient to uniquely distinguish one place from another, making localisation problematic.

In this paper, we describe ongoing work [Dudek et al. 1988; Dudek et al. 1991a; Dudek et al. 1993] that addresses exploration by assuming that our mobile robot can obtain little or even *no* metric positional information whatsoever. In a sense, this is a "worst-case" scenario for metric mapping. In particular, we demonstrate that by associating with each place the (non-unique) local signatures of its neighbours (and their neighbours, etc.) called an *extended* signature, it becomes possible in many but not all cases to answer the place identification problem correctly [Corneil and Kirkpatrick 1980; Dudek et al. 1991b]. In this way, a map of the world may be obtained which faithfully models the places and their connectivity.

Figure 1: (a) A geometric map of a world. (b) A depiction of its associated topological map or graph.

2 Specification

We assume that the world to be explored and mapped is composed of a finite number of distinguishable *places* connected by bi-directional paths. Such a world may be represented as a graph where *vertices* correspond to places and *edges* correspond to paths[1]. For example, the world shown in Figure 1 may be represented by a graph $G = (V, E)$ where V is the set of vertices $v_1, v_2, .., v_6$ and E is the set of edges where $e_{i,j}$ or (v_i, v_j) denotes the edge connecting vertices v_i and v_j.

In addition, we shall assume that at each place, the robot is able to enumerate the incoming/outgoing edges (i.e. paths) in a systematic way (eg. clockwise), relative to the edge

[1]For simplicity of presentation, we will consider only graphs with at most a single edge between any pair of vertices and without an reflexive edges.

by which it arrived at the place: the edge is referred to as the *reference edge* (Figure 2(a,b)). This edge enumeration is also known as a *cyclic edge ordering.* For planar graphs this is closely related to the specification of am *embedding.*

We suggest that many environments of current interest for mobile robots may be characterized as unstructured 2D environments identified by geographical landmarks whose characterization may not by unique. For example, intersections in an office environment would easily confuse a mail carrying robot unless additional information such as the (presumably unique) numbers of adjacent office doors can also be perceived. Similarly, a security robot touring a warehouse might be directed to check for the presence of intruders at specific places without being able to associate specific (video) camera images with such places (since image data is sensitive to camera position and orientation, ambient lighting, etc.).

We refer to the set of landmarks which are 'visible' (i.e., which may be perceived) at a place, along with any other (local) identifying characteristics as the perceptual **signature** of that place [Kuipers and Byun 1991].

To simplify the exposition in the context of graph-like environments, we will consider a specific instance of a signature function, that being the *degree* of the corresponding vertex (another example might associate signature with vertex colour). In general, however, the signature refers to some arbitrary collection of non-unique measurements associated with a location in space.

The robot is situated in some vertex at any time. The *reference edge* is the edge by which the robot entered the vertex (or an arbitrary edge for the initial vertex). The robot and can choose to move to another vertex by edge specified by its position in the enumeration relative to the reference edge.

As the robot performs the exploration, it records all the information obtained whenever any action, sensing or motion (path traversal), is performed. By "remembering" all motion sequences, the robot may retrace any previously performed motion.

Consider, for example, the scenario shown in Figure 2(c) using a counter-clockwise edge ordering. The robot has entered the vertex v_i by the edge $e_{l,i}$ and left it to reach v_j by the edge $e_{i,j}$ which is r edges after $e_{l,i}$, i.e. the r^{th} exit from $e_{l,i}$, and then left v_j for v_k by the edge $e_{j,k}$ which is s edges after $e_{i,j}$. When the robot returns to v_j by $e_{j,k}$, it can reach v_i by the edge $e_{i,j}$ which is $-s$ edges after $e_{j,k}$ (or s edges before it).

We define the transition (or motion) function δ as follows:

$$\delta(v_i, e_{i,j}, r) = v_j \tag{1}$$

means leave vertex v_i by the edge which is r edges after the reference edge $e_{i,j}$ and this takes us to vertex v_j. We can retrace the previous sequence of movements since:

if $\delta(v_i, e_{l,i}, r) = v_j$ and $\delta(v_j, e_{i,j}, s) = v_k$ then $\delta(v_j, e_{j,k}, -s) = v_i$.

3 Related work

Basic work on path planning and navigation began by presupposing a complete map of the environment. In order to move between two points, the robot would first plan a path joining

Figure 2: Examples of: (a) clockwise and (b) counterclockwise edge enumeration. (c) An example of how the robot retraces its movement

those two points using the map provided, and call upon its sensors to try to follow this path.

Subsequent research began to look at how the robot might go about learning and mapping its (unknown) environment. Typically, the learning is performed via exploration, i.e. by moving the robot around, new information is gradually acquired and integrated, in order to update a world model. The sensing data collected are typically positions of obstacles in the world computed according to a given *absolute reference* or based on multiple *relative reference frames*. The absolute reference might be a Cartesian coordinate system or some other absolute system built according to some fixed features existing in the world. The resulting map, called the *geometric map*, is in general a two or three-dimensional data structure which describes the geometric structure of the world and from which a graph generally known as the *road map* representing the connectivity of the free space, can be constructed [Latombe 1991; Davis 1986]. We note that many different kinds of models have been proposed in the literature for defining geometric maps and in particular free space, such as convex polygons, regular grids, generalized cones (freeways), Voronoi diagrams [Latombe 1991; Arkin 1990].

For a variety of reasons, the representation of map information in a metric or geometric form alone may not be suitable for long-range planning. Some of the reasons include the following.

- Long-term goals are often expressed in terms of semantic tokens or places, rather than specific coordinates (i.e. going to a valley, a route or a room rather than a specific coordinate).
- Absolute coordinate systems are typically very difficult to accurately maintain at every scale.
- Complete metric representations may involve very large mounts of data.
- Changes in the environment and the correspondence between objects may be difficult to establish in a purely metric representation.

For these and other reasons there has been some interest in the use of more abstract maps. A *topological* map can be defined [Davis 1986] as a map including all fixed entities in the world such as distinguishable places and regions, linked by topological relations eg. connectivity, containment. Advantages of such an approach include its qualitative nature and attractive links to theories of human cognition and mapping. Such a map is often represented as a graph where vertices are places and edges their adjacency relations. For example, when planning a trip from one place to another, most systems (including people) usually begin by constructing a map of possible routes.

Metric and topological information can be hierarchically related, within the context of a multilevel representation theory of a *large scale* space [2] based on the observation and re-acquisition of distinctive visual events called *landmarks* [Chatila and Laumond 1985]. Topological and metric representations can be associated via an abstraction hierarchy. In this spirit, Chatila et. al. [Chatila and Laumond 1985; Chatila 1986] have defined three levels of description for the world:

1. a 2D geometric level obtained from perceptual data using polygonal approximations of the perceived obstacles,
2. a topological level, and finally
3. a semantic level, obtained by attributing to each element of the topological model a label representing its functional property.

In contrast, the multi-level 'Tour' model of Kuipers and Levitt proposes that a metric representation be constructed *from* a topological one [Kuipers and Byun 1988] such that the successive layers of the system are: 1) sensorimotor (action primitives), 2) procedural (primitives for place-finding and route-following), 3) topological, 4) metric (metric relations). This difference stems, in part, from the manner in which places are defined in the two alternative approaches.

In more recent work, the same authors propose the 'Qualnav' Model [Kuipers and Byun

[2] A "large-scale" space is a space whose structure is at a significantly larger scale than the observations available at any instant [Kuipers and Byun 1988].

1991] for the simulation of a land-roving robot equipped with an omni-view visual sensor that produces a continuous two-dimensional 360 degree image of the surrounding environment. Here the topological map is simply a spatial decomposition defined in terms of landmarks and projections (on the ground) of straight lines which connect adjacent landmarks to form 'landmark-pair-boundaries'. Metric data is added to define the relative orientation between adjacent landmarks. In this way, the Qualnav Model provides a computable theory integrating qualitative, topological representations of a large scale space with the quantitative metric ones. Nonetheless, its applicability is limited to those environments which may be characterized in terms of landmarks and to those robots having specific sensing devices (for perceiving landmarks). In addition, the robot cannot be considered as completely autonomous since it requires some prior knowledge concerning landmarks viewable from the goal places.

It is often assumed that distinctive places can be robustly found, that they are not too numerous, and that no two places can be confused [Schwartz and Yap 1987; Leonard and Durrant-Whyte 1991]. Clearly, this last assumption is an idealization of a real robot exploring a real world; not making it leads to serious complications [Basye and Dean 1990].

Indeed, the notion that no two places can be confused (based on the available sensing data) is precisely the key hidden assumption in the NX Robot work [Kuipers and Byun 1987; Kuipers and Byun 1991]. Unlike the Tour Model for which a place is pre-defined by a set of "unrealistic" views, the NX robot must define in an autonomous way places and connecting paths in order to create a discrete qualitative description of its a necessarily continuous environment. In this model, A *place* is defined as a point which maximizes some distinctiveness measures allowing it to be locally distinctive within its immediate neighbourhood. Such distinctiveness measures are defined in terms of The sensory features, the distinctiveness measures and the feature values maximized at a place together define the *signature* of that place.

While the authors suggest that any robot equipped with a sensorimotor system which provides sufficiently rich sensory input and moves in sufficiently small steps through the environment can use their approach, there are several important assumptions which are made which mitigate such usefulness: places must be distinguishable, the robot must be able to find them robustly and repeatably, and the places are not too numerous (a summary of the Tour, Qualnav, and NX models may be found in [Kuipers and Levitt 1988]).

In contrast to this simulation work, Toto [Mataric 1990; Mataric 1992] is a real robot that creates a topological map (a graph) as it explores a real world. Here the *sensory interaction* with the world is defined in terms of three basic sensors: current sensors on the base motor to detect stalling (non-movement); a ring of 12 Polaroid ultrasonic ranging sensors that covers the entire 360 degree area around the robot; a flux gate compass. As landmarks are detected, they become nodes in the graph along with their qualitative properties, i.e. type (left wall, right wall, corridor) and associated compass bearing. A clever "truth maintenance" protocol is invoked to ensure that the same landmark does not become multiple nodes in the graph.

While Toto can truly explore and map unknown worlds it is subject to the constraints related to its sensing and motor control systems. Its very success is due to the care with which the underlying processing (robot 'behaviors') has been tailored to the kinds of environments for which it was designed.

Finally, we turn to previous work associated with exploration and mapping using markers (movable beacons) [Dudek et al. 1988; Dudek et al. 1991b]. This work deals with the exploration of graph-like worlds, or the validation of maps of such worlds, defined in the same manner as the environments considered in this paper. In addition, however, the robot is equipped with at least one recognizable marker which can be put down or picked up. This physical marker makes v_i distinctive relative to the other vertices, and thus establishes a "temporary" unique signature of that vertex. Such worlds can be fully explored and described in limited complexity using a single movable marker a pebble) even if there are no spatial metrics and limited no sensory ability on the part of the robot.

A practical instance of marker-based navigation is proposed in [Deveza et al. 1994] whereby a mobile robot deposits a short-lived chemical maker on the ground to indicate the path it has already followed. In this way, the robot 'records' information about where it has been in the environment itself as do certain insects (and Hansel and Gretel).

This marker-based analysis requires that the robot be able to reliably place, identify and recover the markers it uses for exploration. In this paper, we discuss how mapping can be accomplished without such markers even though individual places may not be uniquely identifiable.

4 Building a map

We presuppose that the robot is able to define and detect locations associated with vertices in the graph-like representation. Further, it can move between these locations in order to follow edges as specified by the transition function δ. Large scale exploration and mapping algorithm then proceed from this basis.

Over time we maintain a set S that contains the models of the environment that are consistent with the percepts acquired by the robot. This set of solutions is called the "*solution universe*". While the exploration takes place, the robot constructs a data structure called the "*exploration tree*" which is used to compute the set S.

If S contains more than one model, then the robot must rely on additional knowledge about the world such as the total number of places, information about the probability distribution of place signatures, or perhaps some compass measurements, to identify that model which best represents the connectivity information in the world.

4.1 The exploration tree

The exploration tree refers to the collection of possible partial maps the serve as hypotheses about the world the robot is exploring. It is incrementally constructed while the exploration takes place. The *root* of the tree is a map containing only the initial place from which the exploration began. Each successive *level* in the tree corresponds to elaborating the maps in the level above using information from the traversal of a previously unexplored edge. The *nodes* belonging to a given level of the exploration tree represent possible partial models of the world.

Nodes corresponding to the current level in the exploration tree (as the exploration proceeds) are called *frontier* nodes. *Leaf* nodes represent possible models (complete configurations) of world connectivity and are the elements of S. A given node in the exploration tree is considered to be a leaf node (i.e. a possible model) if there are no paths still to be traversed. Our notation is as follows:

- vertices corresponding to places in the world are denoted by v_1,v_2, $\ldots$
- vertices associated with nodes in the exploration tree are denoted by v_1^j, v_2^j, $\ldots$, where v_i^j corresponds to the j^{th} visit to place v_i. Since a given place may be visited several times as part of the exploration, we can have several vertices in a model (one exploration tree node) for a single vertex in the world arising from different visits to that vertex.

A correct model is characterized by the fact that when the robot visits a given place multiple times, it 'recognizes' that these are all visits to the *same* place. That is, there exists a correspondence between v_i^1 (the first visit to the place corresponding to vertex v_i) and v_i^k (the kth visit) for all k, and an absence of other (incorrect) correspondences. To guarantee successful exploration, i.e. exploration leading to the creation of a solution universe S which necessarily contains a model of existing connectivity in the world, two problems must be addressed:

1. How can the robot know when all of the places in the world have been visited ?
2. When a place is visited, how can the robot know whether or not it represents a place previously visited and is therefore already present in the exploration tree ? We shall

call this the *place identification* problem.

Many possible exploration strategies are possible (even stochastic ones) but for simplicity, we will process new edges in a a FIFO (first-in first-out) manner, based on breadth-first traversal in order to guarantee that all edges will be explored, although perhaps not optimally.

Lemma 1 *The breadth first traversal (BFT) of a finite graph will visit all its vertices after at most depth d, d being the graph diameter*[3].

BFT may be represented by a tree[4] where the root is the starting vertex v_s (where the robot starts the exploration), and a level i in the tree contains the i^{th} neighbours of v_s (neighbours will be defined later); see Figure 3.

In Figure 4, we illustrate how the robot explores the world shown in Figure 1, by showing the robot motions and the associated actions to construct the exploration tree. In this example, the exploration tree has only one branch and the solution universe contains just one possible solution which is the leaf of the tree, the node enclosed in a dotted rectangle.

The second problem is more complex since place identification must be performed with very limited information. Indeed, by associating the signature of a place with vertex degree, the robot cannot always know when it is visiting a place for the first time or not. In short, we can be sure that a BFT will visit every node in a graph, but without node labels how can we know when this has taken place? For example, in a world which contains cycles, the

[3]The diameter of a graph G is the maximum distance between any two vertices of G.

[4]The BFT tree associated with a graph G is different from its exploration tree since the BFT tree deals with *labelled* vertices and, further, a single level in the BFT tree corresponds to many levels in the exploration tree.

Figure 3: An example of a breadth first traversal tree.

robot will inevitably re-visit some places. That is, when the robot visits a given place, it could either be the first visit to a new (previously un-visited) place, or a re-visit to any of the places that have the same signature (i.e. that appear the same). Thus, when the robot visits a place, it must consider all possible ways of adding vertices to the frontier nodes in the exploration tree.

Figure 4 shows how the robot explores the world described by the graph G (of Figure 1), by showing the robot motions, and the associated actions to construct the exploration tree. In this example, the universe of possible solutions contains only one branch and one possible solution which is the leaf of the tree (the framed node). We note that a given node (world model) of the exploration tree is considered a possible solution if there are no edges still to be traversed. In the exploration tree, we never know whether or not there are extra places to visit in the environment, only whether or not there are extra exits of a place which we have

Figure 4: The exploration tree for the world shown in Figure 1

not visited yet.

Branches in the exploration tree are created as a result of modelling the true topological structure of the world, or by making one or more correspondence errors of different types. In most cases, branches arising from errors eventually terminate due to inconsistencies resulting from the incorrect topology induced by the error(s).

4.2 Algorithm details

The exploration algorithm functions by incrementally expanding a graph (intended to model the world) for which there exists a known map. Initially, this is simply the robot's starting place. As the exploration proceeds, this map is gradually expanded by adding new vertices and edges. When a place is visited and a new vertex in the map is postulated, its relationship to the set of known vertices must be established and the correspondence (if any) with any other vertex must be verified (using the signature and extended signature analysis).

When the robot visits a place corresponding to a new vertex u, it starts by ordering the incident edges e_i according to the "reference" edge e_0 by which it arrived at u (eg. using a clockwise ordering). (Note that the "reference edge" is only defined by the robot's own history – the reference edge is not perceptible in the graph itself.) The robot then examines these edges (except e_0) sequentially by traversing each one to visit the vertex at the other end. The process of traversing the ith edge e_i may be described as follows:

The process of traversing the successive edges e_i of a new vertex u proceeds as follows:

1. Traverse edge e_i to reach the other vertex v and compute the signature of v.

2. Verify if edge e_i is already connected to another vertex w in the map (for each node in the current level of the exploration tree). If yes, verify the validity of the connection by comparing the extended signatures of v and w: $Sig(v)$ and $Sig(w)$; if they are different, then reject the proposed connection.

3. If edge e_i is 'free', i.e. not already connected to another vertex, compute $C(e_i)$, the set of all possible connections, which includes:

   - connections to previously visited vertices v_i with $Sig(v_i) = Sig(v)$ and with a 'free' incident edge. For each such v_i found, create a new node in the exploration tree by adding an edge connecting v to v_i.

   - a connection to a new vertex w. A new node is created in the exploration tree by adding an edge connecting v to w.

The cost of this exploration process in terms of actual edge traversals by the robot (mechanical complexity) depends only on the search strategy used by the robot in moving through the world. The algorithm is compatible with almost any strategy that progressively traverses new edges. Breath-first search provides a simple example.

Further details on the algorithm and its implementation along with some complete examples may be found in [Dudek, Freedman and Hadjres 1994].

4.3 Algorithm behaviour

For example, if the robot incorrectly constructs a model that connects two vertices v_1 and v_2 during the exploration of v_1 it may observe that there is an inconsistency during a subsequent

Figure 5: An example of an error in edge correspondence.

visit to v_1 that actually takes it to v_2 instead (see Figure 5).

As shown later, exploration tree will always contain a branch for which no errors are committed, i.e. a branch leading to a leaf which faithfully describes the connectivity in the world.

Some of the possible complications are illustrated in the degenerate example shown in Figure 6 (note every location in this world is indistinguishable from every other). For example, at the fourth level of the exploration tree, during the visit to the vertex v_3 coming from v_2, the robot finds the local signature (degree) of this vertex to be equal to the degree of another visited but not yet explored, vertex v_3^1, and therefore the robot has to consider 2 cases : (i) v_3 is "fused" with v_3^1, labelled node 'a' at level 4, (ii)v_3 is a new vertex v_3^2 labelled node 'b' at level 4.

We can describe the mapping errors that can occur at two levels: local correspondence errors at a given vertex of edge, and global structural errors that ensue as the result of one or more local errors. We can divide the possible local errors into a few basic types. These occur

Figure 6: Problems with constructing the exploration tree.

when the robot arrives at (and maps) a vertex v_i, as shown in Figure 7. (Errors associated with paths (edges in the exploration tree) are subsumed within this classification.) These are as follows.

E1 Errors of type OLD-LOOKS-NEW. A vertex v_i is assumed to be a new vertex even though it has been visited before (i.e. a failure in correspondence). In this case, an additional vertex is added to represent the current place even though a vertex for the current place has already been created.

E2 Errors of type MIS-CORRESPONDENCE. There are two sub-cases to consider: (a) VERTEX MIS-CORRESPONDENCE a vertex v_i is "recognized" as a known vertex v_j $(j \neq i)$ even though, in reality, it is another old vertex v_k (i.e. the robot has confused two existing nodes); (b) EDGE MIS-CORRESPONDENCE v_i is indeed v_j but the edge ordering is wrong. Thus, an incorrect connection between vertices is established in the model.

E3 Errors of type NEW-LOOKS-OLD. A vertex v_i is assumed to be a previously visited vertex even though it is new. In this case, the map will have a missing vertex relative to the real world and incorrect connectivity.

The consequences of these errors on the exploration tree are illustrated in Figure 8. A succession of failures in correspondence (E1 errors) leads to the creation of an infinite branch (succeeding node M9), corresponding to the case where each time a place is visited, it is always considered as a new vertex. A possible model where no correspondences occur (possibly erroneously) occurs in every exploration tree. As we show, in addition to such erroneous

Figure 7: Possible errors which may appear during the exploration.

branches, the exploration tree will always contain a branch which leads to a leaf describing the real world, where no errors are committed (eg. model $M8$ of Figure 8).

Theorem 1 *The solution universe S produced by the exploration of a graph $G = (V, E)$ with a cyclic edge ordering always contains at least one model $M \in S$ which is isomorphic to G (i.e. which describes its structure). This model is obtained after m traversals of new edges where m is the cardinality of E. (We will see later that the exploration may lead to multiple models $\{M_1, ...\} \in S$ which all faithfully represent the real world.)*

Proof: All nodes $v_i \in V$ and all its outgoing edges must be visited as part of the exploration (Lemma 1)). Now, $\forall v_i \in V$ and all edges $e_{i,k} = (v_i, v_k) \in E | v_k \in V$ the traversal of e_i involves the creation of $C(e_i)$ which contains all locally consistent of adding this edge to the partial world models obtained thus far. One of these must, of necessity, include a model M which faithfully represents the part of the real world explored thus far since that is, by definition a possible combination of edges and vertices. This means that after each traversal of a new edge (as opposed to backing up along an edge previously traversed), a new real world connection is added to M (and to all the other partial models). Therefore, after m such new edge traversals, M has been augmented in such a way as to faithfully represent the real world. □

Corollary 2 *The solution universe produced by the exploration of a finite graph is never empty, since it must contain at the very least the real world model.*

4.4 The extended signature

Typical exploration trees usually include branches that are subsequently pruned (i.e. they develop inconsistencies before they lead to a complete model). This can be observed in the tree shown in Figure 8. The major reason for this is the weakness of the signature information used by the robot for addressing the place identification problem; incorrect hypotheses regarding vertex correspondences cannot be avoided based on local perceptual input. To make the exploration more robust and effective, we shall now exploit non-local information by defining an *extended signature* incorporating signature information about a place's neighbours.

Neighbours are defined with respect to potential forward exploratory motion of the robot. Thus, each vertex has a predecessor (except for the root) and zero or more immediate neighbours (or successors). Specifically, the initial (zero'th) neighbours $N_0(u)$ of a vertex u as follows:

$$N_0(u) = \{u\} \quad (2)$$

Figure 8: An example of an exploration tree illustrating the three types of errors.

Then its immediate neighbours $N_1(u)$ may be defined as follows:

$$N_1(u) = \{v \in V | (u, v) \in E\} \tag{3}$$

We can then define the immediate "outgoing" neighbours $N_2(u)$ of those vertices in $N_1(u)$ as follows:

$$N_2(u) = \{v \in V \mid v \in \{N_1(N_1(u)) - u\}\} \tag{4}$$

Note that we take care to exclude u from this set,since it is the predecessor of the vertices in $N_1(u)$ but not a successor of any vertex in $N_1(u)$. In a similar manner we define $N_m(u)$, the m^{th} neighbours of u, as the union of the successors of the nodes in $N_{m-1}(u)$.

Examples of neighbourhood computations are shown in Figure 9. Note that this definition of a neighborhood does not preclude the repeated occurrence of a vertex in different neighborhoods of a vertex; a vertex could be in both $N_m(u)$ and $N_{m+1}(u)$ (unlike an alternative "wavefront" definition). Thus, for example, every node in a clique is the m^{th} order neighbour of every other node for all $m > 0$.

To make the vertex matching in the exploration tree more robust, we will now go beyond local signature information (vertex degree) to consider an *extended signature* defined in terms of the signatures of the neighbours of a place. For example, suppose that we have two vertices u and v and we wish to establish whether they refer, in fact, to the same physical place. If they have identical signatures (degrees), then we consider their immediate neighbours, $N_1(u)$ and $N_1(v)$ as a cue to disambiguating them. If the immediate neighbours also have identical

Figure 9: Some neighbourhood computations.

signatures and appear in the same sequence or configuration, then we also consider *outgoing* neighbours $N_2(u)$ and $N_2(v)$, and so on.

For a generic signature defined by an arbitrary sensor, we define the m^{th} order extended signature of a vertex u in terms of the signatures of its neighbours up to a given neighborhood distance m as follows:

$$Sig_0(u) = (sense(u)) \tag{5}$$

$$Sig_m(u) = (Sig_{m-1}(u), (sense(v))) \quad \forall v \in N_m(u) \tag{6}$$

where $sense(v)$ denotes sensory information obtained at vertex v to identify it. For example, for the graph in Figure 1 with a sensor that returns the vertex degree, the extended signatures

Figure 10: An example of signature tree.

of the vertex v_1 are as follows:

$$Sig_0(v_1) = (2) \tag{7}$$

$$\begin{aligned} Sig_1(v_1) &= (2, (3, 2)) \\ Sig_2(v_1) &= ((2, (3, 2)), (1, 1, 1)) \end{aligned} \tag{8}$$

This extended signature may be also be viewed as a *signature tree*, where the root represents the degree of vertex u and the nodes belonging to level i represent the degrees of the i^{th} neighbours of u, i.e. $N_i(u)$ (see Figure 10).

Note that a node's extended signature is generally only unique with respect to a specific reference edge. Consider as root a vertex with degree d, with d possible extended signatures. In Figure 11, we illustrate how any extended signature may be obtained from any other extended signature by circularly re-ordering its edges. We define two extended signatures as isomorphic if one can be obtained from the other by a cyclic shift of the edges at the root. Consequently, when comparing two vertices, the robot must take into consideration all cyclic re-orderings at the root in evaluating their extended signatures. If the extended signatures are *isomorphic*, then the two vertices may correspond to the same place.

4.5 Some special kinds of graphs

Despite the availability of an extended signature, ambiguity may still remain in place identification. As a result, the universe of possible solutions S may contain various models which are equivalent insofar as the extended signature is concerned, of which just one faithfully reflects the connectivity in the world (for example a simple cycle of either three or four vertices). Additional information of various sorts can be used. For different classes of graphs special-purpose strategies or cues can be used to resolve this difficulty [Dudek, Freedman and Hadjres 1994].

We can structure graphs into several classes depending on the kind of additional information we require in order to solve the exploration problem such that obtain a single world model. These are as follows.

1. Graphs which may be reliably explored (i.e. to obtain a correct model of the unknown world) just by using the extended signatures over a distance sufficiently large to allow for the visiting of all vertices.

2. Graphs for which alternative models can be disposed of by foreknowledge of the number

Figure 11: An example illustrating the effect on the signature tree of cyclic re-ordering at the root (i.e. via selection of a different reference edge). The graph G is shown in the upper left. The other three trees show 4^{th} order signatures of vertex v_2.

Figure 12: A taxonomy of graph topologies and appropriate methods for finding a unique model.

of nodes in the real world (i.e no alternative model has the same number of nodes as the correct model). (Planar graphs may be in this class.)

3. Graphs for which the presence of a single uniquely *distinguishable vertex* can resolve any ambiguity in the solution universe. In many cases vertices can be uniquely identified by sequence of turns along a path between them and this distinguished vertex (i.e by the sequence of edge numbers along this path). How such a vertex could be optimally located is an iteresting problem.

4. Graphs for which additional information is required.

Further work on how this taxonomy of graphs could be elaborated remains to be carried out. Note, however, that an appropriate selection of cues could provide information how to augment an environment to avoid confusion by a robot navigator (in the context of navigation as well as exploration).

We not also that a distinguished vertex *along with an orientation cue* at that vertex provides a very powerful cue. Thus, painting an arrow on the floor while exploring is much more informative than simply leaving a mark or a bread crumb.

5 Experimental results and heuristics

The analysis described in the preceding sections of this document relates primarily to the feasibility of the algorithm and its worst-case behaviour. In order to examine its performance with respect to various graph-like worlds that might arise in practice, we developed an implementation that would allow experimental performance assessment. It can be used to simulate exploration of any connected graph to generate the set of output models.

We now briefly consider empirical considerations drawn from running this program on a large number of examples representative of important different classes of environment. The computational cost of the mapping algorithm is a function of the number of nodes (possible world models) generated in the exploration tree; the actual generation, maintenance and comparison operations for extended signatures have low-order polynomial complexity. The number of nodes in the exploration tree depends on the distinctiveness of the perceptual information extracted at each place. When places are clearly distinguishable, the exploration tree grows only as a linear function of the number of new locations visited. When the locations observed in the world are not uniquely distinguishable, the exploration tree can grow more rapidly as a consequence of the ambiguity. Thus, when insufficient perceptual information is available to constrain the growth in the exploration tree various pruning or deferred expansion strategies become attractive.

Despite the availability of an extended signature, ambiguity may still remain in place identification. As a result, the universe of possible solutions S may contain various models which are equivalent insofar as the extended signature is concerned, of which just one faithfully reflects the connectivity in the world (for example a simple cycle of either three or four vertices). This, in turn, can lead to a large number of candidate models being developed and leads to a question of when to terminate the exploration and accept the map that has already been established. Figures 13 and 14 illustrate and example of a correct and alternative (incorrect and larger) solution found for the same world – note that with the impoverished data available to the robot, there is no way to establish which model is correct, i.e. both are consistent with the available sensor data; it is possible the the larger model could, in fact, be the correct one. Vertices in the input (correct) model are uniquely numbered while vertices in the larger model (Figure 14) which are duplicates are marked with an apostrophe in a shaded ellipse.

A natural pruning strategy is to limit the exploration tree or terminate search is based on the observation that errors of type $E2$ (MIS-CORRESPONDENCE) and $E3$ (NEW-LOOKS-OLD)

Figure 13: A correct solution for a 6x6 node map (see following figure).

Figure 14: An alternative (more complex) model for the prior 6x6 graph.

typically lead to *inconsistency* in the hypothetical map (see Figure 7) while errors of type $E1$ (OLD-LOOKS-NEW) alone lead to a map which is far larger then it should be, but such maps may be self-consistent (eg. Figure 14). That is, errors of type $E1$ alone lead to a map with duplicated sub-graphs but without incorrect "fusion" of non-equivalent nodes. Thus, the incorrect models that are not rapidly eliminated tend to be far larger then the correct model of the world. In short, it is the manifestation of Ockham's razor: the real model tends to be the most concise. We have examined this experimentally by constraining the search process to avoid retaining hypotheses (maps) that are much more complex than the current simplest model. Results to date suggest that, for realistic environments, this leads to major performance improvements. An example illustrating the effects of pruning the models of size greater than $(\gamma s + 2)$ where s is the current largest *incomplete* model and γ is 1.05 and 1.15 is shown in Figure 15 where a few key nodes have been eliminated. It illustrates that even a small amount of pruning has a major effect.

6 Summary

In this paper, we have described how a robot with limited perceptual capacities may explore and faithfully map an unknown graph-like world. Such a world can be used to represent the topology or connectivity of a real metric environment if a lower-level subsystem can extract the places and routes associated with vertices and edges.

Our approach is based on aggregating non-local information to compensate for potentially ambiguous local perceptual information. Locations in the world are identified by a non-

Figure 15: Effects of pruning. Number of exploration tree nodes as a function of level in the tree with and without pruning. Note that the correct model is found early.

unique “signature” that serves as an abstraction for a percept that might be obtained from a robotic sensor. While the signature of any single place may not be unique, under appropriate conditions the distinctiveness of a particular set of signatures in a neighborhood increases with neighborhood size. By using a collection of non-unique local signatures we can thereby construct an **“extended signature”** that uniquely determines the robot’s position (although in certain insufficiently rich worlds additional information is also required). The algorithm makes use of no metric information such as the distances of the paths traversed, but the availability of such measurements would simplify the mapping problem.

The worst case behaviour of the algorithm is clearly problematic. For example, there can be multiple embeddings of the same graph, leading to multiple models of the unknown world. For example, for regular graphs every place is identical to every other and the number of possible models grows initially as $O(k!)$ for level k of the exploration tree (although keeping only one or two models is sufficient to express both all the structure that the robot has observed and all that it can accomplish given the limited percepts it has made). This initial explosive growth is reduced once the tree depth exceeds the vertex degree (i.e. very early for planar graphs). This difficulty is not surprising since under such circumstances we are attempting to construct a map from no knowledge about where we are or how we are moving – anything is possible. Thus the difficulty is not intrinsic to this algorithm but rather to the impoverished stimuli.

In realistic worlds where more information is available and various places can be distinguished from one another using sensory data, the tree grows *much* more slowly. Further, metric information that constrains the location of several nodes (places in the world), even approximately, typically greatly simplifies the problem. In simulations we have carried out, the correct map is usually found almost immediately. In the extreme, position information constraining every place in the world reduces the problem to a fairly simple one.

To conclude, we now suggest some additional ways of reliably exploring unknown worlds when both the extended signature and the existence of a uniquely distinguishable place are not sufficient.

In some worlds, there might be multiple interchangable distinguishable vertices (places in the world) instead of just one. As a result, the robot may associate extra information with each vertex, since we can define and record multiple edge sequences, one from each distinguishable vertex.

It becomes logical to now consider under what conditions the existence of even two distinctive vertices is sufficient for the robot to uniquely identify any other place in the environment. Let v_{d1} and v_{d2} be two distinctive vertices which can be confused with one another. Two different vertices u and v can be confused by the robot if they have the same extended signatures and if v_{d1} and v_{d2} occupy the same positions in the tree representations of the extended signatures. This is illustrated in Figure 16.

In practice, the robot can simply decide to stop the exploration after a certain number of world models are obtained, and then pick one model from among the possible candidates. Typically robots have access to at least limited metric information that could be used to choose the correct model of the real world. Unfortunately, such a strategy can not be guaranteed to succeed in all cases. One way to accomplish this would be to augment the graph with *qualitative* perceptions based on the metric data. We note that fuzzy set theory has been used with some success for such qualitative modelling [Freedman and Liu 1993].

Finally, the robot might have access to probabilistic information about, for example, the existence of places with a specific degree, the average and the standard deviation, etc. Such information might be available thanks to previous exploration of 'similar' worlds. In appears that such information could be used to assign probabilities to alternative models in

Figure 16: An example of graph where two vertices may be confused despite the existence of two distinguishable vertices.

the exploration tree, as well as to prune the tree as it is being developed.

In conclusion, in the absence of metric data the exploration and mapping of worlds at a topological level (in terms of a graph) can be carried out even when the local sensing is highly ambiguous. The use of a non-local sensor signature can greatly reduce the difficulty of the problem although, in general, multiple alternative world models may be possible except in worlds described by acyclic graphs. In practice, the alternative models often differ greatly in their complexity (as measured by the number of places) and hence simply selecting a parsimonious model may be sufficient.

7 Acknowledgements

The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council and the Centre de recherche informatique de Montréal.

References

Arkin, R. 1990. *Mobile Robots in Knowledge Engineering, Vol II, Applications (Hojjat Adeli, Editor)*, chapter 5, pages 116–160. McGraw Hill.

Basye, K. and Dean, T. 1990. Map learning with indistinguishable locations *Uncertainty in Artificial Intelligence 5*. Lemmer, M. H. L. N. K. J. F., editor, pages 331–340. Elsevier Science Publishers.

Chatila, R. 1986. Mobile robot navigation: space modelling and decisional processes *Robotics Research 3*. Faugeras and Giralt, editors, pages 373–378. MIT Press.

Chatila, R. and Laumond, J. 1985. Position referencing and consistent world modelling for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 138–170.

Corneil, D. G. and Kirkpatrick, D. G. 1980. A theoretical analysis of various heuristics for the graph isomorphism problem. *SIAM J. Computing*, 9(2):281–297.

Cox, I. 1989. Blanche: Position estimation for an autonomous robot vehicle. In *Proceedings of the IEEE/RSJ International Workshop on Robots and Systems (IROS)*, pages 432–439.

Davis, E. 1986. *Representing and Acquiring Geographic Knowledge*, Pitman and Morgan Kaufmann Publishers, Inc., London and Los Altos, California.

Deveza, R., Thiel, D., Russell, A., and Mackay-Sim, A. 1994. Odor sensing for robot guidance. *International Journal of Robotics Research*, 13(3):232–239.

Dudek, G., Freedman, P., and Hadjres, S. 1994. Signatures and extended signatures for topological mapping using graphs. CIM-94-34, Centre for Intelligent Machines, McGill University.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. 1988. Robotic exploration as graph construction. Technical Report RBCV-TR-88-23, University of Toronto (Dept. of Computer Science), Toronto.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. 1991a. Robotic exploration as graph construction. *IEEE Trans. Robotics and Automation*, Vol. 7, NO. 6:859–865.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. 1991b. Robotic exploration as graph construction. *Transactions on Robotics and Automation*, 7(6):859–865.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. 1993. Map validation and self-location in a graph-like world. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence*, pages 1648–1653, Chambery, France. Internation Joint Conf. on Artificial Intelligence Inc.

Dudek, G., Romanik, K., and Whitesides, S. 1995. Localizing a robot with minimum travel.

In *Proc. Sixth ACM-SIAM Symposium of Discrete Algorithms (SODA)*, pages 437–446, San Francisco, CA.

Dudek, G. and Zhang, C. 1996. Vision-based robot localization without explicit object models. In *Proc. International Conference of Robotics and Automation*, Minneapolis, MN. IEEE Press.

Elfes, A. 1987. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265.

Freedman, P. and Liu, J. 1993. Using uncertain measurements to make plausible perceptual hypotheses: application to autonomous navigation using sonar. In *Proc. of the Int. Conf. on Intelligent Autonomous Systems: IAS-3*, Pittsburgh, PA.

Kuipers, B. and Byun, Y.-T. 1991. A robot exploration and mapping strategy based on a semantic hierachy of spatial representations. *Robotics and Autonomous Systems*, 8:46–63.

Kuipers, B. and Levitt, T. 1988. Navigation and mapping in large-scale space. *AI Magazine*, pages 25–43.

Kuipers, B. J. and Byun, Y. 1987. A qualitative approach to robot exploration and map-learning. In *IEEE Workshop on Spatial Reasoning and Multi-sensor Fusion*, pages 390–404, Los Altos, California.

Kuipers, B. J. and Byun, Y.-T. 1988. A robust qualitative approach to a spatial learning mobile robot. In *SPIE Advances in Intelligent Robotics Systems*.

Latombe, J.-C. 1991. *Robot Motion Planning*, Kluwer Academic Publishers, Standford University.

Leonard, J. J. and Durrant-Whyte, H. F. 1991. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382.

MacKenzie, P. and Dudek, G. 1994. Precise positioning using model-based maps. In *Proceedings of the International Conference on Robotics and Automation*, San Diego, CA. IEEE Press.

Mataric, M. J. 1990. A distributed model for mobile robot environment-learning and navigation. Technical Report AI-TR 1228, MIT Artificial Intelligence Laboratory, Artificial Intelligence Laboratory, 545 Technology square Cambridge, MA 02139 USA.

Mataric, M. J. 1992. Integration of representation into goal-driven behavior-based robots. *IEEE Trans Robotics and Automation*, Vol. 8,NO. 3:304–312.

Moravec, H. 1988. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74.

Schwartz, J. and Yap, C.-K. 1987. *Algorithmic and Geometric Aspects of Robotics*, Lawrence Erlbaum Assoc., Hillsdale, N. J.

List of Figures