Exploiting Local Perceptual Models for Topological Map-Building*

Patrick Beeson **Matt MacMahon**† **Joseph Modayil**
Jefferson Provost **Francesco Savelli**‡ **Benjamin Kuipers**

Intelligent Robotics Lab
Department of Computer Sciences
University of Texas at Austin

Abstract

The Spatial Semantic Hierarchy (SSH) provides a robot-independent ontology and logical theory for building topological maps of large-scale environments online. Existing SSH implementations make very limited use of perceptual information and thus create many candidate maps. Metrical mapping implementations capture detailed knowledge about local small-scale space but do not handle large environments well due to computational limitations and global metrical uncertainty. In this paper, we extend the SSH to utilize better sensory information by incorporating information derived from local metrical models into the large-scale space framework. This new extension of the Spatial Semantic Hierarchy uses local topology obtained from local perceptual models to constrain a global topological map search.

1 Introduction

Modern research developments provide robots with precise metrical models of the local surrounding environment. These models, including occupancy grids, obstacle maps, and even some vision based representations, permit a robot to perceive, plan, and move efficiently in small-scale space. All suffer uncertainty when scaling to large-scale space due to global localization problems in sufficiently complex environments. We term local, bounded implementations of these methods *local perceptual models (LPMs)*.

The Spatial Semantic Hierarchy (SSH) provides a framework for building topological maps of large-scale environments [Kuipers, 2000]. An axiomatic formalization of the SSH [Remolina and Kuipers, 2001; In Press] specifies how topological maps can be built during exploration. Following this specification, there exists an implementation which allows an autonomous robot to build correct topological maps of large-scale environments online. This implementation is well-suited to resolving map ambiguities when exploring large-scale space.

The SSH was designed to be robot independent. It defines weak requirements for building robust cognitive maps. To demonstrate this, existing implementations make very limited use of perceptual information. In this paper, we extend the SSH to utilize better sensory information in order to build a complete local perceptual model of the sensory surround. The addition of LPMs allows the Spatial Semantic Hierarchy to construct complete local topology descriptions from small-scale space observations, greatly improving the efficiency of topological mapping in large-scale space.

Our approach allows us to factor spatial uncertainty into distinct components, controlled in distinct ways. First, movement uncertainty is controlled by the behavior of feedback-driven motion control laws. Second, pose uncertainty is controlled by incremental localization within the local perceptual model. Third, topological ambiguity about the large-scale structure of the environment is controlled by the abduction process that builds the topological map. Fourth, global metrical uncertainty is controlled by relaxing metrical information in separate local frames of reference into a single global frame of reference, guided by the topological map.

This paper has four main sections. We begin by surveying research on metrical, topological, and hybrid mapping techniques. Next, we review the SSH: the representation framework, the logical formalization, and an implementation. We then present local perceptual models and their interface with the SSH. We conclude by summarizing the benefits of this extension and discussing future work.

2 Related Work in Mobile Robot Mapping

2.1 Metrical Maps

Researchers have worked extensively on making metrical methods, such as occupancy grids, computationally feasible for building large maps with a single, global coordinate system [Gutmann and Schlegel, 1996]. Additionally, methods to create metrical maps using a set of landmarks have been well explored [Smith *et al.*, 1990; Dissanayake *et al.*, 2001; Montemerlo *et al.*, 2002].

Metrical maps excel in handling some of the initial, low-level problems roboticists encounter. Because these meth-

*Research of the Intelligent Robotics Lab is supported in part by the Texas Higher Education Coordinating Board, Advanced Technology Program (Grant 003658-0656-2001), and by an IBM Faculty Research Award.

†Supported by the Naval Research Laboratory.

‡ALCOR Group, Dipartimento di Informatica e Sistemistica "Antonio Ruberti", Università di Roma "La Sapienza"

ods are often used with high-precision sensors which reduce *perceptual aliasing* (when observations at multiple locations are similar), localizing within local environments is efficient. This reduces the effect of odometry error on pose estimation. Furthermore, recent probabilistic methods make metrical mapping robust and fast [Thrun, 2000].

Current metrical map implementations have several disadvantages. Metrical maps reduce position error in local space, but errors propagate over large spaces. Similarly, metrical maps cannot easily handle cyclical environments once position estimates have drifted sufficiently. Mapping and planning in very large metrical maps can be time consuming and algorithms are often run offline. Metrical maps also suffer from the lack of a good interface for higher-level, symbolic problem solvers. They are insufficient for a robot to reason about the layout of its environment or to communicate route directions to another robot that lacks the same map.

2.2 Topological Maps

A topological map represents an environment as a graph. There have been several distinct topological mapping implementations [Kuipers and Byun, 1991; Mataric, 1992; Shatkay and Kaelbling, 1997; Duckett and Nehmzow, 1999] which differ in the semantics for the graphs. Some topological mapping implementations build topological maps autonomously, some are given topological maps *a priori*, and some explore autonomously while the researcher provides place names to overcome *perceptual aliasing* (multiple locations are similar in appearance).

Topological maps are more compact representations than global metrical maps. They allow high-level symbolic reasoning for map-building, navigation, planning, and communication. Since the environment is discretized into a graph, movement errors that accumulate between graph nodes do not necessarily accumulate across a global frame of reference.

To date, topological mapping implementations have not created maps of extremely large environments. Often implementations are brittle; they assume the environment is well-structured, static, and simple. However, cognitive map research supports the creation of topological maps of large, complex environments [Siegel and White, 1975; Yeap, 1988; Chown *et al.*, 1995; Kuipers, 2000].

2.3 Hybrid Maps

Recently, researchers have begun to look at hybrid topological/metrical maps. The SSH, often thought of as a framework for creating only topological maps, has always allowed for a global metrical map to be created *after* the topological map. Kuipers and Byun [1991] created a "patchwork metrical map" using the topological map as a base for integrating data gathered locally at places and along paths. Local frames of reference at place neighborhoods and along path segments are relaxed into a single global frame of reference, minimizing the "strain" at their joints.

Thrun *et al.* [1998] also create a topological map to guide the generation of a global metrical map. Thrun [1998] tried the opposite direction, creating a topological map from a global metrical map, which unfortunately entails the aforementioned metrical map scaling problems. Related work connects the compact representations of rooms into both global metrical and topological maps [Yeap and Jefferies, 1999].

Most work on hybrid maps has dealt with generating topological and metrical maps as disjoint, sequential processes. Some recent research integrates metrical and topological mapping by matching local metrical models to eliminate topological place aliasing [Duckett and Saffiotti, 2000]. Once a correct topological map is built, the local metrical models are pieced together to make a global metrical map.

Our extension to the SSH uses local perceptual models (currently, occupancy grids) to extract local topology descriptions of places. Local topologies constrain the global topological map search by eliminating many inconsistent maps. Unlike previous methods, we do not compare actual LPMs but assume that the local topology description of a place can be created deterministically for comparison.[1] Finally, LPMs can be reused to create a patchwork global metrical map.

3 The Spatial Semantic Hierarchy (SSH)

3.1 SSH Framework

The SSH describes knowledge of large-scale space in terms of four distinct representations for spatial knowledge with different ontologies [Kuipers, 2000]. Large-scale space is defined as space whose structure is beyond the sensory horizon of the robot. When an environment is described as a large-scale space, places are represented as zero-dimensional objects, connected by one-dimensional paths, and perhaps contained in two-dimensional regions.

At the *control level*, knowledge consists of hill-climbing control laws that define isolated *distinctive states (dstates)*, and trajectory-following control laws that take the robot from one dstate to the neighborhood of the next. The *causal level* consists of *causal schemas* $\langle ds, a, ds' \rangle$ where the states ds and ds' correspond to distinctive states, and the deterministic *action* a represents the sequence of control laws for moving from ds to ds'. Each dstate has a single *view*, a description of the sensory input vector obtained at a distinctive state. The *topological level* consists of *places*, *paths*, and *regions* related by connectivity, order, and containment. A topological map is constructed by an abduction process to explain a sequence of observations. The *metrical level* is a global metrical map constructed from a patchwork of local metrical maps which use the topological map as a skeleton.

By definition, the details of the robot's sensory experience are below the level of abstraction of large-scale space: they belong to *small-scale space*. A major contribution of the current work is to show how the same environment can be described as both small-scale and large-scale representations, how the two representations are related, and how they help each other build a more powerful and robust spatial model.

3.2 SSH Logical Formalization

Remolina and Kuipers [2001; In Press] present a formalization of the SSH framework as a non-monotonic logical the-

[1] Transient objects, such as pedestrians, can be eliminated from the LPM; however, we leave as future work the generalization of local topology extraction to cover quasi-static entities such as doors, which can be open in one LPM and closed in another.

ory. The theory contains axioms describing the properties and relationships of actions, views, distinctive states, causal schemas, places, paths, and regions.

This theory provides a clear and precise specification of the possible logical models (topological maps) given the sequence of actions and views observed while exploring. A *prioritized circumscription policy* [McCarthy, 1980; Lifschitz, 1994] specifies how the simplest of these logical models is identified. This policy drives the *abduction* of the topological map from experience. In particular, a *nested abnormality theory* [Lifschitz, 1995] manages the complex, block-structured set of axioms and their circumscription.

The SSH logical theory consists of several sets of axioms. One set of axioms describes the robot's sensorimotor experience by asserting causal schemas. Another set of axioms enforces the SSH topological properties. A third set of axioms incorporates local metrical information, such as bounds on the path distances between places and the local radial angles between paths at a place.

3.3 SSH Implementation

The current implementation of the SSH causal and topological map builder follows the logical formalization. It takes as input an alternating sequence of views and actions $v_0, a_1, v_1, a_2, v_2, \ldots, a_n, v_n$. An *observation* consists of an action and the resulting view $\langle a_i, v_i \rangle$. The algorithm conducts a best-first search in the space of maps to find the simplest map that is consistent with the axioms and explains all the observations. This corresponds to the unshaded portion of Figure 1. This implementation allows online execution, providing the robot with the current preferred topological map at any time while exploring the environment.

A map consists of a set of constant symbols referring to actions, views, dstates, causal schemas, places, paths, and regions. A map also contains ground instances of both spatial relations (such as left-of relations) and equality relations (specifying which symbols refer to the same object). A new observation $\langle a_i, v_i \rangle$ may require that the current map be extended. Typically several new map extensions are consistent with the axioms, producing a search tree of potential maps.

For example, suppose the current map M explains the observation sequence through view v_{i-1} and the current dstate is ds_{i-1}, which means that $view(ds_{i-1}, v_{i-1})$. Now the robot experiences the observation $\langle a_i, v_i \rangle$. If M already contains a causal schema $\langle ds_h, a_i, ds_k \rangle$ where $ds_h \equiv ds_{i-1}$ and $view(ds_k, v_i)$, then M explains the current observation ($ds_i \equiv ds_k$), so no map extension is necessary. If $\neg view(ds_k, v_i)$, then M is discarded because of an incorrect equivalence between two dstates at some previous time step.

If the observation is not explained by M, the algorithm extends the map by creating the new dstate symbol ds_i, adding the causal schema $\langle ds_{i-1}, a_i, ds_i \rangle$, and asserting $view(ds_i, v_i)$. It extends the tree of maps by branching on the assertion $ds_i \equiv ds_{i-l}$ for *every* previously-known dstate where $view(ds_{i-l}, v_i)$. This is in addition to the map extension where ds_i is a completely new dstate. These maps are checked for consistency with the axioms in the SSH causal and topological theory. The simplest consistent map (minimal by the prioritized circumscription policy) M' is selected from the entire map search tree and expanded with the next observation it not yet seen. M' may remain the simplest consistent map, may be discarded due to an inconsistent observation, or may create more map extensions to be added to the search tree.

The best-first search is guaranteed to find a consistent map of the observation sequence if one exists; however, best-first search can overlook the globally simplest map.[2] The local topology representations presented here reduce the size of the search tree, which may permit exhaustive breadth-first search to find the globally simplest map. The local metrical axioms for abduction are not currently implemented but can refine the map preference order to improve best-first search.

4 Integrating Small-Scale Space into the SSH

The SSH was originally designed to make minimal assumptions about a robot's sensorimotor system. Here, we extend the SSH to include local perceptual models of small-scale space. We show how the structure of these models can provide local topology descriptions that greatly improve the search for the correct topological map. This extension corresponds to the shaded portion of Figure 1.

First, we show how a bounded local perceptual model (LPM) describes the neighborhood of a topological place. (It can also be used to describe the ephemeral surroundings of the robot as it executes a travel action.) Second, we show how the LPM of a place neighborhood can be analyzed to identify descriptions of *gateways* and *path fragments*. Third, we show how to derive a complete *local topology* description of the place neighborhood from the set of gateways and path fragments in the LPM. Fourth, we explain the benefits of using the local topology during construction of the global topological map.

4.1 Local Perceptual Models

The SSH topological map describes a place according to its role in *large-scale* space: as a point location on one or more one-dimensional paths. The direction ($+$ or $-$) on a path denotes the direction up or down the order of places along the path. Each place has a dstate facing each direction along each of its paths. After arriving at one dstate at a place, a turn action takes the robot to another dstate, ready to travel along a path to a different place.

The *local perceptual model* (LPM) unpacks this simple abstraction of a place into an extended two-dimensional region, describing the *small-scale* space within the sensory scope of the robot. In our current implementation, the LPM is a bounded occupancy grid, but the paradigm is intended to extend to other sensor models. The LPM provides all the usual benefits of metrical maps: serving as a virtual sensor (or "observer"), supporting local path planning, and improving obstacle avoidance.

[2] If the observation sequence is too short, or if the environment is highly symmetrical, the resulting map could be *simpler* than the actual environment. With real environments, rich sensors, and extended exploration this is usually not a problem. Only rare pathological environments are inconsistent with the SSH logical theory.

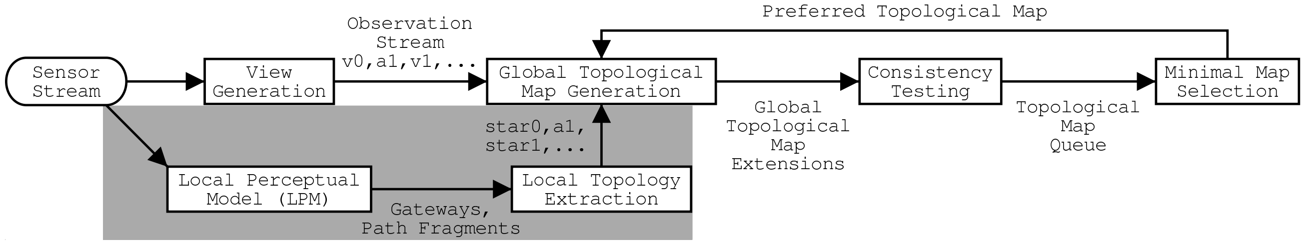


Figure 1: **Topological Mapping Implementation.** The unshaded region is the classic SSH implementation (Section 3.3) which relies on hill-climbing to gather views. The shaded region is the extended SSH (Section 4) that extracts the local topology from a local perceptual model. This extension substantially reduces map search allowing the optimal map to be found faster.

The abstraction relation between place (a large-scale-space object) and LPM (a small-scale-space object) is the key issue of this paper. Arriving at a dstate corresponds to arriving at a *gateway* associated with a *path fragment* in the LPM, facing inward. (Gateways and path fragments are detailed in Section 4.2). A turn action from this dstate to another corresponds to moving within the LPM to reach another gateway, facing outward.[3] Travel away from the place along a path starts with the robot facing outward at a gateway with an applicable trajectory-following control law.

The use of the LPM imposes a responsibility on the robot: on arrival at a place, it must explore enough to eliminate uncertainty about which accessible positions are free or obstructed within the bounded scope of the local place neighborhood. In return, the robot obtains two substantial benefits in terms of its spatial knowledge. First, it can localize unambiguously at any pose within the LPM, rather than relying on the minimalist SSH strategy of hill-climbing to an unambiguous pose. Second, it constructs a *complete* representation of the set of gateways and path fragments, and hence of dstates and possible turn actions, at that state. This complete local topology description will be quite helpful to the construction of the global topological map.

4.2 Identifying Gateways and Path Fragments

A *gateway* is a boundary between qualitatively different regions of the environment: specifically a boundary of the local place neighborhood (in the classic SSH, the boundary between trajectory- following and hill-climbing applicability). Each gateway has two directions, *inward* and *outward*. Gateways can be identified using several different criteria that appear to be functionally identical. We plan to investigate these alternatives in more detail in future work.

- Positions within an LPM can be tagged with the control law applicability conditions they satisfy, allowing gateways to be defined as boundaries between regions of different control law applicability. This criterion most closely matches the definition of gateways given above.
- In the PLAN framework, Chown *et al.* [1995] define gateways as the locations of major changes in visibility.

> "In buildings, these [gateways] are typically doorways.... Therefore, a gateway occurs where there is at least a partial visual separation between two neighboring areas and the gateway itself is a visual opening to a previously obscured area. At such a place, one has the option of entering the new area or staying in the previous area." (page 32)

- A geometric criterion for identifying gateways *in corridors* is the medial axis of free space in the LPM. A gateway corresponds to a "constriction" (or "critical line" [Thrun, 1998]) along a medial axis edge, where the distance between the edge and obstacles is a local minimum near a larger maximum. Due to its computational simplicity and our current experimental environments, this is the criterion used in our current implementation (Figure 2).

The local perceptual model of a place neighborhood will also include *path fragments*: portions of large-scale paths that are grounded in small-scale space. Each path fragment is associated with at least one gateway, while each gateway is associated with exactly one path fragment. Path fragments associated with a single gateway are portions of paths that terminate at the topological place. In large-scale space, many paths may continue through each place. For each such path, the LPM will contain a path fragment associated with two gateways (Figure 2(c) has two gateways that lie on the same, horizontal path fragment). Path fragments are never associated with more than two gateways.

We assume that the robot has a procedure to decide whether or not a given gateway shares its path fragment with another gateway (i.e. the path fragment continues through the place neighborhood). The details of this procedure are below the level of our theory and depend on the robot configuration and the environment.

An essential requirement of this step is that the *complete* set of path fragments and gateways in the LPM are identified. This extracts the local topology and abstracts topological path relations directly from gateway information instead of creating causal schemas from turn actions.

4.3 Describing the Local Topology

Whether a robot is reasoning about large-scale or small-scale space, when a robot arrives at a place on a path, it must select

[3] Even following one travel action with another along the same path, which requires no turn action in large-scale space, results in physically moving between gateways in the LPM.

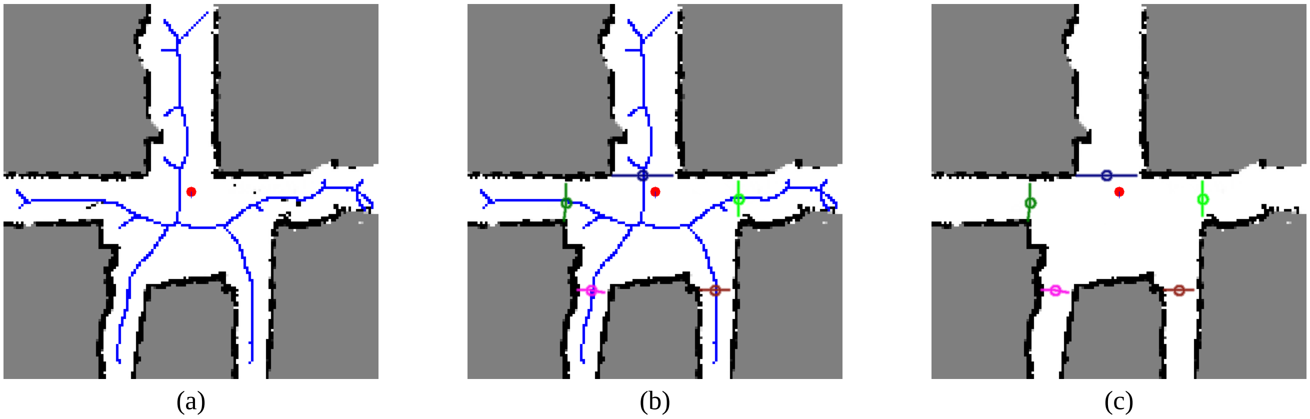


Figure 2: **Finding Gateways in an LPM.** Our current implementation of a local perceptual model (LPM) is a bounded occupancy grid. The robot is shown as a circle in the center of the LPM. **(a)** To find gateways in corridor environments, the algorithm computes the medial axis of the occupancy grid free space. **(b)** The maximum of the medial axis graph is found (where the distance of obstacles from the graph is maximal) and each edge is traversed, looking for "constrictions" (where the distance between the graph edge and obstacles is a local minimum). **(c)** The final gateways are drawn as lines connecting the graph edge minima (circle) with the closest obstacles.

a path on which to depart. From the complete set of path fragments and gateways identified in the LPM, the local topological structure of the place neighborhood defines the options for this choice.

Building the local topology enforces a circular order on directed path fragments and gateways. This process is shown for two LPMS in Figure 3.

- Create a set of tuples $\langle PF, GW, CL\rangle$, where:
  - PF is a path fragment and a direction ($+$ or $-$) along it;
  - GW is the gateway (or pair of gateways) on PF facing the same direction;
  - CL is the control law for a travel action away from the place along the path fragment in that direction. In our current implementation, these are: Midline, LeftWall, RightWall, DeadEnd, and None. (For terminating path fragments, DeadEnd means that further travel is blocked, and None means that no control law is applicable.)
- Initialize the circular order with the tuples including outward-facing gateways, in their clockwise sequence around the place. For path fragments that both enter and leave the place neighborhood, both directed path fragments will now appear in the order.
- Each tuple with an inward-facing gateway and a path fragment that terminates at the place is now inserted into the order. Its position in the order between outward-facing gateways is determined by the same procedure that decides whether a path fragment continues through the neighborhood.[4]

[4] In the current implementation, an inward-facing gateway is projected across the LPM until it intersects a gateway on the other side, an obstacle, or is too far from any obstacle to support a control law.

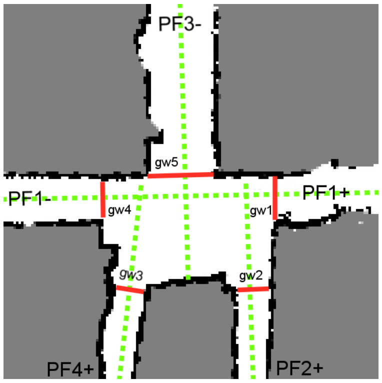
This produces a structure called the *small-scale star* description of the local topology (Figures 3(b) and 3(e)), since the elements of the description (path fragments and gateways) belong to the small-scale space ontology of the LPM. When this place is incorporated into a topological map, each path fragment is bound to a corresponding path, and a distinctive state is defined for each (path, direction) combination. This produces a corresponding circularly ordered structure called a *large-scale star* that describes the local topology in the large-scale space ontology (Figures 3(c) and 3(f)).

The local topology description provides a purely qualitative account of "left" and "right". Starting from the robot's current path fragment and direction, the subset of path fragments (or dstates in the large-scale star) encountered by moving down the circular order until observing the same path fragment in the opposite direction yields the possible destinations of a "Turn right" action. This avoids the need to define "right" and "left" in terms of a threshold on some angular variable.

Since the set of path fragments and gateways in the LPM is complete, the description of the dstates and directed paths at the place in the circular order of the large-scale star is also complete. Causal schemas for turn actions $\langle ds_i, turn, ds_j\rangle$ are implicitly defined between every pair of dstates ds_i and ds_j at the place. This simplifies construction of the global topological map.

4.4 Building the Topological Map

The global topological map is built using the search algorithm described in Section 3.3. The existing implementation processes views and actions. To recapitulate, for a given observation $\langle a_i, v_i\rangle$, starting at a dstate ds_{i-1}, if the map does not contain an equivalent of the causal schema $\langle ds_{i-1}, a_i, ds_i\rangle$, then the search tree is extended with maps in which the successor state ds_i matches each existing dstate with view v_i.



(a)

Small-scale star description from PF1+

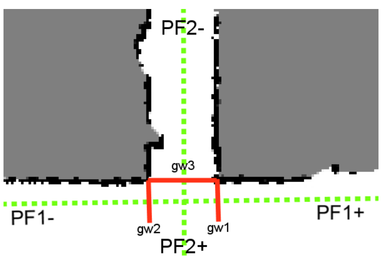
((PF1+,	(gw1,out) & (gw4,in),	Midline),
(PF2+,	(gw2,out),	Midline),
(PF3+,	(gw5,in),	DeadEnd),
(PF4+,	(gw3,out),	Midline),
(PF1-,	(gw4,out) & (gw1,in),	Midline),
(PF4-,	(gw3,in),	DeadEnd),
(PF3-,	(gw5,out),	Midline),
(PF2-,	(gw2,in),	DeadEnd)

(b)

An example large-scale star abstraction

((ds1,	Pa1+),
(ds2,	Pa2+),
(ds3,	(Pa3+)),
(ds4,	Pa4+),
(ds5,	Pa1-),
(ds6,	(Pa4-)),
(ds7,	Pa3-),
(ds8,	(Pa2-))

(c)



(d)

Small-scale star description from PF1+

((PF1+,	(gw1,out) & (gw2,in),	LeftWall),
(PF2+,	(gw2,in),	None),
(PF1-,	(gw2,out) & (gw1,in),	RightWall),
(PF2-,	(gw2,out),	Midline)

(e)

An example large-scale star abstraction

((ds1,	Pa1+),
(ds2,	(Pa2+)),
(ds3,	Pa1-),
(ds4,	Pa2-)

(f)

Figure 3: **Local Topological Extraction of the Star Description.** **(a)** Given the gateways and path fragments of a place the robot must extract the local topology. **(b)** The local topology extraction forms a small-scale star description for the place. The star enumerates all the path fragments encountered in clockwise order. **(c)** Once a local topology is extracted, the map builder creates dstates and paths for the place (a large-scale star) in order to create maps. This environment has five gateways, four paths, and eight dstates. Path directions in parentheses denote termination at the place. **(d-f)**The same process in another environment that has three gateways, two paths, and four dstates. Here, there is a path fragment (PF2+) that does not intersect an obstacle and does not have an applicable control law.

This is in addition to the map extension where ds_i is a new dstate. The proposed new maps are checked against the axioms of the SSH logical theory, and inconsistent maps are discarded.

The local topology description substantially improves the computational complexity by constraining this search. The new algorithm processes places and actions. After a travel action, when the robot arrives at a place neighborhood, it builds an LPM and a star description of the place. This provides both a complete set of dstates at the place and a set of schemas describing all possible turn actions among them, removing the need for views.

For turn actions, the correct causal schemas $\langle ds_i, turn_k, ds_j \rangle$ have already been created, so no new map extensions are necessary. For a travel action, the tree of maps must be extended whenever the causal schema $\langle ds_i, travel_k, ds_j \rangle$ does not already appear in the map, but the branching is constrained. Rather than adding a map to the tree for each dstate ds_j with view v_j, the algorithm will create a new map only for each place with the same local topology as the current place. Perceptual aliasing can occur only when the local topology descriptions of two places match, not simply when two dstates have the same view.

In both the previous and new algorithms, the tree of maps branches whenever there is no causal schema $\langle ds_i, a_k, ds_j \rangle$ in the map to describe the current observation. Therefore, the worst-case complexity for breadth-first search in both cases is exponential, $O(b^d)$, where $b-1$ is the number of known states matching the current observation and d is the number of actions that can cause branching.

In the previous algorithm (unshaded portion of Figure 1), branching can occur for any turn or travel action, so $d \le p+q$ where p is the number of turn actions executed and q is the number of travel actions. The branching factor is the number of previously discovered dstates with views matching the observed view plus one (the hypothesis where the robot is at a new dstate). In the worst case, all dstates have the same view, thus we can bound $b \le p+q+1$. Consequently, the complexity of the previous implementation is $O((p+q+1)^{(p+q)})$.

In the new algorithm (shaded portion of Figure 1), local topology reduces both the branching factor b and the branching frequency d. Since the local topology describes every possible turn at the current place, branching occurs only after travel actions, never after turns, so $d \leq q$. After a novel travel action, the map branching factor is the number of *places* in the map that match the observed *local topology*, plus one (hypothesis where the robot is at a new place). This clearly constrains search more than view matching, but in the worst case is still the number of travel actions, so $b \leq q + 1$. Therefore, the worst-case complexity of the new algorithm is $O((q + 1)^q)$.

These worst-case bounds only occur in environments which are completely perceptually aliased; however, even with partial perceptual aliasing, using LPMs greatly reduces search. The next following illustrates the performance of each in an interesting, small real-world environment.

A real world example

We illustrate our extended version of the SSH by building a topological map of a real office environment. The environment contains multiple nested loops and significant perceptual aliasing—in both dstates (same view) and places (same local topology). The environment and the paths explored are shown in Figure 4(a). The true environment has 9 places, 36 dstates (four at each place), 6 paths, and twelve path segments which connect the nine places. The robot visits all nine places in 14 travels actions, which is the optimal number of travel actions to cover all twelve path segments. A total of 15 LPMs are made during exploration.

For the previous SSH implementation (Section 3.3), the exploration routine turns to every dstate during each visit to a place *before* turning to the dstate it leaves on. The turns ensure every dstate in the environment is visited while maintaining the optimal number of travel actions. The turns also create multiple maps during each visit to a place.

Within this algorithm, exhaustive breadth-first search creates over 5,000 consistent candidate maps before the eighth travel action. This high number of models is due to the large amount of perceptual aliasing among dstates (i.e. the small number of views) in this environment. This search becomes intolerably slow after eight travel actions, so we stopped the process before exploration completed.

The best-first search is able to finish the entire route with around 1000 consistent candidate maps (the number varies because of a non-deterministic selection between equally simple maps). The topological map that is minimal according to the circumscription policy is an oversimplification of the true environment due to the large amount of perceptual aliasing which often unifies dstates with the same view (regardless of the local topology of their respective places).

The new algorithm generates no new maps after turn actions. Hence, new maps are only added to the search tree when perceptual aliasing of entire places occurs after a travel action. This algorithm creates only 35 consistent candidate maps for the total exploration when using breadth-first search. Moreover, the structurally correct topological map of the environment (Figure 4(b)) is recognized by the algorithm as minimal according to the circumscription policy when compared to the other candidate maps.[5]

5 Future Work and Concluding Remarks

This paper extends the Spatial Semantic Hierarchy to exploit local perceptual models. In this integrated system, local perceptual models resolve uncertainties encountered during robot motion and generate local topological description at topological places while the SSH axioms manage global topological ambiguities. Local topologies extracted from LPMs reduce the number of maps generated, which makes the map search more efficient. This hybrid cognitive-mapping approach, utilizing the reciprocal strengths of both topological and metrical maps, facilitates efficient, accurate mapping of complex large-scale environments.

This extension to the Spatial Semantic Hierarchy creates a number of new, interesting research issues.

- How can the SSH logical theory be extended to formalize our extended approach?
- How can using omnidirectional cameras instead of metrical maps reinforce the discovery and use of gateways?
- How can a robot explore and map dynamic environments (e.g. where doors open and close across LPMs)?
- Can this extension scale to hierarchical definitions of places (e.g. rooms, buildings, neighborhoods, cities)?

References

[Chown *et al.*, 1995] E. Chown, S. Kaplan, and D. Kortenkamp. Prototypes, location, and associative networks (PLAN): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–51, 1995.

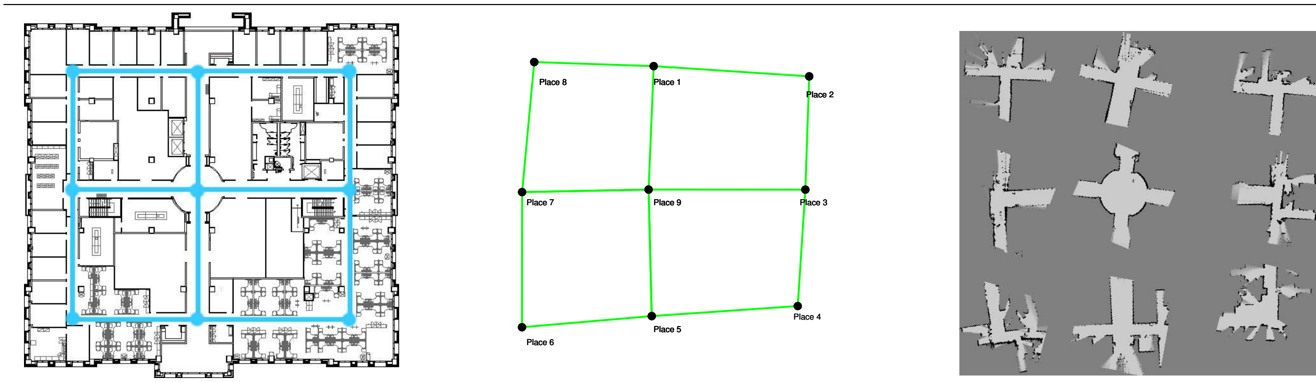
[Dissanayake *et al.*, 2001] G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Trans. on Robotics and Automation*, 17(3):229–241, 2001.

[Duckett and Nehmzow, 1999] T. Duckett and U. Nehmzow. Exploration of unknown environments using a compass, topological map and neural network. In *Proc. of the 1999 IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA'99)*, Monterey, CA, 1999.

[Duckett and Saffiotti, 2000] T. Duckett and A. Saffiotti. Building globally consistent gridmaps from topologies. In *Proc. of the 6th Int. IFAC Symp. on Robot Control (SYRORO-00)*, Vienna, Austria, September 2000. Elsevier.

[Gutmann and Schlegel, 1996] J.-S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. of the 1st Euromicro Work. on Advanced Mobile Robots*, 1996.

[5] At the time this is written, the new algorithm described here has been fully implemented through Section 4.2. The algorithms in Sections 4.3 and 4.4 have been hand-simulated using the gateways that were automatically identified in LPMs collected during physical robot exploration.



(a) (b) (c)

Figure 4: **Map-Building Example. (a)** A CAD drawing of the third floor of the ACES building with a skeleton of the robot's exploration. Only the hallways were explored by the robot in this routine: doors and cubicles were ignored. **(b)** A visualization of the topological map. The image is obtained by a very preliminary implementation of relaxation of local distance and orientation displacements between each place. This was run on a small amount of data which makes the layout sensitive to errors. **(c)** A partial patchwork metrical map with the LPMs of places shown on top of the relaxed map. Once the data gathered along paths are added, the metrical map can be relaxed to maximize scan correlations. This should improve the relaxation of (b) and (c) even further.

[Kuipers and Byun, 1991] B. J. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journ. of Robotics and Autonomous Systems*, 8:47–63, 1991.

[Kuipers, 2000] B. J. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[Lifschitz, 1994] V. Lifschitz. Circumscription. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in AI and Logic Programming*, volume 3, pages 298–352. Oxford Univ. Press, 1994.

[Lifschitz, 1995] V. Lifschitz. Nested abnormality theories. *Artificial Intelligence*, 74:351–365, 1995.

[Lynch, 1960] K. Lynch. *The Image of the City*. MIT Press, Cambridge, MA, 1960.

[Mataric, 1992] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Automation*, 8(3):304–312, 1992.

[McCarthy, 1980] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1,2):27–39, 171–172, 1980.

[Montemerlo *et al.*, 2002] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. of the AAAI National Conf. on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[Remolina and Kuipers, 2001] E. Remolina and B. Kuipers. A logical account of causal and topological maps. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-01)*, pages 5–11, Menlo Park, CA, 2001.

[Remolina and Kuipers, In Press] E. Remolina and B. Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, In Press. http://www.cs.utexas.edu/users/qr/papers/Remolina-aij-03.html.

[Shatkay and Kaelbling, 1997] H. Shatkay and L. P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI-97)*, pages 920–927, San Mateo, CA, 1997. Morgan Kaufmann.

[Siegel and White, 1975] A. W. Siegel and S. H. White. The development of spatial representations of large-scale environments. In H. W. Reese, editor, *Advances in Child Development and Behavior*, volume 10. Academic Press, New York, 1975.

[Smith *et al.*, 1990] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167–193, 1990.

[Thrun *et al.*, 1998] S. Thrun, S. Gutmann, D. Fox, W. Burgard, and B. J. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proc. of the 15th National Conf. on Artificial Intelligence (AAAI-98)*, pages 989–995. AAAI/MIT Press, 1998.

[Thrun, 1998] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[Thrun, 2000] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.

[Yeap and Jefferies, 1999] W. K. Yeap and M. E. Jefferies. Computing a representation of the local environment. *Artificial Intelligence*, 107(2):265–301, 1999.

[Yeap, 1988] W. K. Yeap. Towards a computational theory of cognitive maps. *Artificial Intelligence*, 34:297–360, 1988.