Formalizing Regions in the Spatial Semantic Hierarchy: an AH-Graphs implementation approach

Emilio Remolina[1], Juan A. Fernandez[2], Benjamin Kuipers[1], and Javier Gonzalez[2]

[1] Department of Computer Science⋆⋆⋆
University of Texas at Austin Austin, TX 78712, USA
{eremolin,kuipers}@cs.utexas.edu
[2] Departamento de Ingenieria de Sistemas y Automatica (ISA) †
E.T.S.I. Informatica, Universidad de Malaga
Campus Teatinos - 29080 Malaga, Spain
{jafma,jgonzalez}@ctima.uma.es

Abstract. We are interested in the problem of how an agent organizes its sensorimotor experiences in order to create a spatial representation. Our approach to solve this problem is the Spatial Semantic Hierarchy (SSH), an ontological hierarchy of representations for knowledge of large-scale space. At the SSH topological level, space is represented by *places* and connectivity relationships among them. Places are arranged into *paths* so that the topological representation looks like the street network of a city. Grouping places into *regions* allows an agent to reason efficiently about its spatial knowledge. Regions can be organized in a hierarchical structure suitable for hierarchical planning and human-level interface. In this paper we show how a hierarchy of regions can be automatically created by an agent. We extend the SSH axiomatic theory to include regions as first order objects at the SSH topological level. Based on this formalization, an implementation using Annotated Hierarchical graphs (AH-graphs) is proposed. The AH-graph model is chosen for its efficiency to perform basic operations like path planning, its facility to integrate information needed by different agent's tasks, and because it provides a large indexed database of knowledge about the world with a friendly flow of information from and to human operators.

Keywords: Cognitive map, Spatial reasoning, Regions, Hierarchical representations of space.

⋆⋆⋆ This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-9504138 and CDA 9617327, by NASA grant NAG 9-898, and by the Texas Advanced Research Program under grants no. 003658-242 and 003658-347.

† Part of this work was carried out during a stay of the second author at the Department of Computer Sciences of the University of Texas at Austin, under grant of the Spanish Government. The work on AH-graphs and NEXUS has been supported by the Spanish Government under research contract CICYT-TAP96-0763.

1 Introduction

The basic problem we are interested in solving is how an agent creates its spatial representation from its sensorimotor experiences. Our approach to solve this problem is the **Spatial Semantic Hierarchy (SSH)** [24, 25, 23]. The SSH is a computational theory of the cognitive map [35, 22]. It is an ontological hierarchy, where each level of the hierarchy has its own *ontology* abstracting the ontology of the levels below it. It comprises four levels: control, causal, topological and metrical. Two fundamental ontological distinctions are embedded in the SSH. First, the continuous world of the control level is abstracted to a discrete symbolic representation at the causal and topological levels, to which the metrical level adds continuous properties. Second, the egocentric world of the control and causal levels is abstracted to the world-centered ontologies of the topological and metrical levels.

In this paper we are primarily concerned with the SSH topological level. At this level, space is represented by *places* and connectivity relationships among them. *Places* are arranged into *paths* so that the topological map[1] looks like the street network of a city. Places can be grouped into *regions*, which in turn can be organized into a hierarchy. This hierarchy of regions is useful for planning, navigation and human-level interface of autonomous robots. Hierarchical planning methods as described in [22, 32, 4, 11] are supported by this representation.

Research on hierarchical representations of space has become an important topic for different disciplines: GIS [19], graph theory [17], planning [37], robotics [25, 9, 11], etc.. Several hierarchical representations of the topology of the environment have been proposed. A complete model designed for the study of the computational efficiency and suitability of different robotics operations is the Annotated Hierarchical Graph (AH-graph) model [10, 11]. The AH-graph model represents different types of relationships between elements of the world, including containment relationships which lead to a hierarchical representation. It can also support the inclusion of other types of information: geometrical, physical, procedural, etc., as annotations on the topology. When it is used to model large-scale space, the result is a representation of the environment with several levels of detail that reduces the computational cost of basic operations like path search [11], and provides a friendly flow of information from and to human operators [10].

In this paper we extend the SSH axiomatic theory to include regions as first order objects at the SSH topological level. The resulting representation resembles the one described in the SSH's predecesor, the TOUR model [22], and maps well into the AH-graph model. Accordingly, we present an implementation of the SSH topological model based on AH-graphs. Finally, we explore different

[1] We use the term *topological map* to refer to the SSH topological level.

automatic methods to define regions.

The paper is organized as follows. Sections 2 and 3 describe the SSH and formally define the hierarchy of regions. Section 4 defines the AH-graph model. Section 5 shows how the AH-graph model is used to implement the SSH topological and metrical levels. Section 6 analyzes the problem of automatically constructing hierarchies from a plain representation of space. Finally, we present the conclusions of this work.

2 The Spatial Semantic Hierarchy

The Spatial Semantic Hierarchy (SSH) [24, 25, 23] is an *ontological hierarchy* of representations for knowledge of large-scale space.[2] An ontological hierarchy shows how multiple representations for the same kind of knowledge can coexist. Each level of the hierarchy has its own *ontology* (the set of objects and relations it uses for describing the world) and its own set of inference and problem-solving methods. The objects, relations, and assumptions required by each level are provided by those below it. The SSH abstracts the structure of an agent's spatial knowledge in a way that is relatively independent of its sensorimotor apparatus and the environment within which it moves. Next we present the SSH's levels:

- At the *control level* of the hierarchy, the ontology is an egocentric sensorimotor one, without knowledge of fixed objects or places in an external environment. A *distinctive state* is defined as the local maximum found by a hill-climbing control strategy, climbing the gradient of a selected feature, or *distinctiveness measure*. Trajectory-following control laws [26] take the robot from one distinctive state to the neighborhood of the next.
- The ontology at the SSH *causal level* consists of views, distinctive states, actions and schemas. A *view* is a description of the sensory input obtained at a locally distinctive state. An *action* denotes a sequence of one or more control laws. A *schema* is a tuple $\langle (V, dp), A, (V', dq) \rangle$ representing the (temporally extended) event in which the robot takes a particular action A, starting with view V at the distinctive state dp, and terminating with view V' at distinctive state dq.
- At the *topological level* of the hierarchy, the ontology consists of *places, paths*, and *regions*, with connectivity and containment relations. At the topological level, the spatial representation posits the minimal set of paths and places consistent with the set of schemas.[3] A place corresponds to a set of distinctive states linked by turn actions. A path is a structure that includes an ordered sequence of places connected by travel actions without turns. Paths

[2] In large-scale space the structure of the environment is revealed by integrating local observations over time, rather than being perceived from a single vantage point.

[3] In order to formally state these minimality conditions, the causal and topological levels are formalized as circumscriptive theories [36, 29].

are used in the cognitive map to describe linear geographical structures such as streets. Places and paths define a topological network which can be used to guide exploration of new environments and to solve new route-finding problems.[4] Using the network representation, navigation among distinctive states is not dependent on the accuracy, or even the existence, of metrical knowledge of the environment.
- At the *metrical level* of the hierarchy, the ontology for places, paths, and sensory features is extended to include metrical properties such as distance, direction, shape, etc.. Geometrical features are extracted from sensory input, and represented as annotations on the places and paths of the topological network.

In this paper we formally state the basic properties of regions. We will then propose an implementation using AH-graphs. Next we define some of the predicates we use to represent the relationships among objects in the SSH.

Places on a path are arranged into a linear order. Distance between places is explicitly defined only for places on a same path. Relative orientation between places is derived from the angle between paths at a common place. We use the following predicates to represent this information:

1. $at(ds, p)$: distinctive state ds is at the topological place p.[5]
2. $path\_order(pa, p, q)$: place p is before place q in path pa. The order of places on a path is a total linear order.
3. $path\_distance(pa, p, q, d)$: d is the distance between place p and place q according to path pa.[6]
4. $path\_angle(p, pa, pa', a)$: a is the angle from path pa to path pa' at place p.

Regions are sets of places. The containment relations permit a particular place to have a partially ordered set of containing regions, rather than simply a nested sequence. A downward mapping, going from more abstract to more specific descriptions of places is necessary in order to allow information to be stated at one level of abstraction and used at another.

3 Formalizing Regions at the SSH topological level

Regions are sets of places. *Regions* themselves can be grouped to form new regions. We extend the SSH topological level relations by adding the following predicates (and abbreviations):

[4] Notice that although the topological map has a graph like structure, a path in the graph theory sense is not necessarily a SSH topological path.

[5] See the SSH's control and causal levels.

[6] Currently, we represent the uncertainty associated with distance between places by closed intervals of real numbers. Other representations for uncertainty are possible, for example, probability distributions.

1. $in\_region(p, r)$: place p is in place (region) r.
2. $is\_region(r)$ stands for the formula $\exists p\, in\_region(p, r)$.

By $default$ a place is not a region. Two regions are equal whenever they represent the same set of places. Two places are equal if they represent the same set of distinctive states.Accordingly, equality between regions must satisfy the following axioms: [7]

$$\neg is\_region(p) \wedge \neg is\_region(q) \rightarrow p = q \equiv \{ds : \ at(ds, p)\} = \{ds : \ at(ds, q)\} \tag{1}$$

$$is\_region(p) \vee is\_region(q) \rightarrow \tag{2}$$
$$p = q \equiv \{s : \ \neg is\_region(s), in\_region^{*}(s, p)\} = \{s : \ \neg is\_region(s), in\_region^{*}(s, q)\}$$

Defining paths among regions. Once places have been arranged into regions, we must define paths among regions. We do so by *lifting* the order relation among places in a path to their corresponding regions. We introduce the following relation between paths:

1. $lifted\_to(pa, pa1)$: path $pa1$ is created by $lifting$ path pa.

The next axiom defines the relationship between lifted paths.

$$lifted\_to(pa, pa1) \wedge path\_order(pa, p, q) \wedge in\_region(p, rp) \wedge \tag{3}$$
$$in\_region(q, rq) \wedge rp \neq rq \rightarrow path\_order(pa1, rp, rq)$$

We require regions to be “path-convex” sets of places, that is,

$$path\_order(pa, p, q) \wedge in\_region(p, r) \wedge in\_region(q, r) \wedge \tag{4}$$
$$path\_order(pa, p, s) \wedge path\_order(pa, s, q) \rightarrow in\_region(s, r)$$

The next example illustrates how Axiom 3 works and why we require regions to be path-convex.

Example 1.

Consider the path pa depicted in Figure 1a. Suppose we have the regions $A = \{a, b\}$, $C = \{c\}$ and $D = \{d, e, f\}$. Let’s consider how to lift path pa to path $pa1$. Since place a is before place c in path pa (i.e. $path\_order(pa, a, c)$ is true) then, region A is before region C in path $pa1$ (i.e. $path\_order(pa1, A, C)$ is true).

Notice that since a and b belong to the same region A, it is not the case that $path\_order(pa1, A, A)$, although it is the case that $path\_order(pa, a, b)$.

Consider a similar scenario as above. Suppose we have two regions, $A = \{a, b, d, e, f\}$ and $C = \{c\}$ (see Figure 1b). Suppose we lift path pa to path $pa1$.

[7] in_region^{*} denotes the transitive closure of in_region.

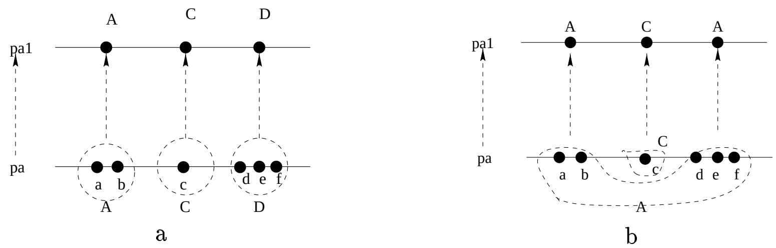

Fig. 1. (a) Lifting paths. Path pa is lifted to path $pa1$. The order of places in path pa defined the order of their corresponding regions in path $pa1$. (b) Regions have to be "path-convex" in order for our path lifting method to work (see text).

Since place a is before place c in path pa, region A is before region C in path $pa1$. Similarly, since place c is before place d in path pa, region C is before place A in path $pa1$. Since the order among places in a path is not symmetrical, we will have a contradiction. { end of example }

By requiring that regions be connected sets of places, the definitions above guaranty that any topological path among regions can be translated to actual behaviors at the SSH control level. This in turn implies that we can use hierarchical planning techniques for robot navigation.

Defining distance between regions. At the SSH metrical level we must define the distance and orientation between regions. These could be defined in different ways. For example, for each region a place is chosen as its representative, and orientation among regions is then defined as orientation among representatives. In the same vein, it is possible to define "the center of mass" of a region, and then define the orientation among regions as the orientation among their centers of mass.

In the current implementation of the SSH we represent distances by closed intervals of real positive numbers. The distance between two regions is the minimum interval containing all distances between any two places in the regions. Formally,[8]

$$\begin{aligned}
&path_order(pa1, r1, r2) \wedge \\
&a = min\{a' : \exists pa, p, q \; [lifted_to(pa, pa1), in_region(p, r1), in_region(q, r2), \\
&\qquad\qquad path_distance(pa, p, q, [a', b'])]\} \wedge \\
&b = min\{b' : \exists pa, p, q \; [lifted_to(pa, pa1), in_region(p, r1), in_region(q, r2), \\
&\qquad\qquad path_distance(pa, p, q, [a', b'])]\} \\
&\rightarrow path_distance(pa1, r1, r2, [a, b])
\end{aligned} \tag{5}$$

[8] Notation: We assume all our formulas universally quantified. Whenever a formula ϕ is a conjunction, $\phi = C_1 \wedge \ldots \wedge C_n$ we will replace $\wedge$ by , and write $C_1, \ldots, C_n$.

Defining orientation between regions. In order to define the angle between paths at a region, at the SSH metrical level we associate a *frame of reference* with a region. Each place in the region gets a location with respect to this frame of reference.[9] In addition, for each path pa and place p in pa we associate a *heading* (i.e. angle) indicating the direction of pa at p.[10] We use the following predicates at the SSH metrical level:

1. $frame\_of\_reference(r, f)$: f is a frame of reference for region r.
2. $path\_heading(pa, p, f, v)$: v is the heading of path pa at place p w.r.t. the frame of reference f.

At the SSH topological level, we define the places at which the path *enters* or *leaves* a region. These places define the "boundary" of the region and are important when creating a hierarchical plan. We use the following relations:

1. $place\_enter(pa, r, p)$: p is a place at which the path pa enters region r.
2. $path\_leaves(pa, r, p)$: p is a place at which the path pa leaves region r.

We explicitly define the predicates above as follows:

$$place\_enter(pa, r, p) \stackrel{def}{\equiv} \\ in\_region(p, r) \wedge \forall q \, \{path\_order(pa, q, p) \rightarrow \neg in\_region(q, r)\} \tag{6}$$

$$place\_leaves(pa, r, p) \stackrel{def}{\equiv} \\ in\_region(p, r) \wedge \forall q \, \{path\_order(pa, p, q) \rightarrow \neg in\_region(q, r)\} \tag{7}$$

Finally, the angle between paths at a region is defined as the directed angle between the corresponding headings associated with the paths at the places where they leave the region. Formally,

$$\begin{aligned} & frame\_of\_reference(r, f) \wedge on\_path(r, pa) \wedge on\_path(r, pa') \wedge \\ & A = \{a : \exists pa1, pa1', p, q, h_p, h_q \\ & \qquad [lifted\_to(pa1, pa), lifted\_to(pa1', pa'), \\ & \qquad place\_leaves(pa1, r, p), place\_leaves(pa1', r, q), \\ & \qquad path\_heading(pa1, p, f, h_p), path\_heading(pa1', q, f, h_q), \\ & \qquad directed\_angle(f, h_p, h_q, a)] \\ & \quad \} \wedge \\ & ang = min\_int\_cover(A) \\ & \rightarrow path\_angle(r, pa, pa', ang) \end{aligned} \tag{8}$$

[9] Given a global frame of reference associated with a region, locations for places are assigned such that they preserve the estimated distance and relative orientation between consecutive places in a path. Information from multiple paths is combined to further constrain the assignment of locations (a constraint propagation algorithm is used for this purpose).

[10] As with distances, a heading is represented by an interval. However, in this paper we assume that headings are represented by real values.

where $min\_int\_cover(A)$ denotes the minimum angle interval covering all the angles in the set A. Example 2 shows how all the above concepts work in a T-like environment.

The rest of the paper describes how to implement the previous axiomatization using AH-graphs.

4 The AH-Graph Model

The AH-graph model has been designed as a hierarchical database for storing the information that a mobile robot gathers from the real world. Since it is general enough to represent large amounts of knowledge in a highly indexed way, this model can also be useful in other applications (GIS, networks, large databases, etc.). The information stored in an AH-graph can be processed by a human operator in a friendly manner, discarding unnecessary details when elements are accessed (dealing with different levels of abstraction). Its hierarchical nature also leads to important reductions in the computational cost of searching paths between its elements [11].

Basic Definitions. An Annotated Hierarchical graph (AH-graph) represents a portion of the world as an ordered list of plain graphs, called *hierarchical levels*: $\{L^0, L^1, \dots L^{k-1}\}$. A hierarchical level L^i is a quadruple $\langle N^i, A^i, s_n^i, s_a^i \rangle$. Figure 2 illustrates the basic idea behind an AH-graph.

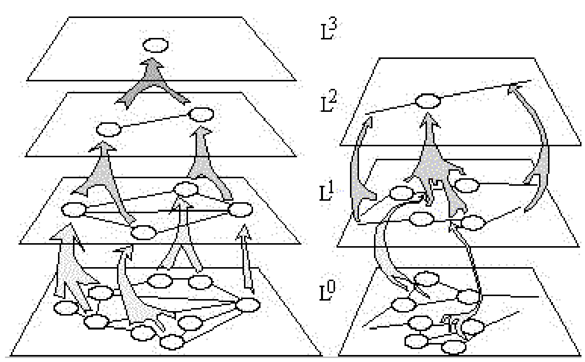

Fig. 2. Representation of two portions of the world with different levels of detail, by two AH-graphs of four and three hierarchical levels, respectively. The AH-graph on the left illustrates the summarization of nodes from the lower to the higher hierarchical levels. The AH-graph on the right illustrates the summarization of arcs.

The nodes of the AH-graph are the sets $\{N^0, N^1, \ldots, N^{k-1}\}$. They represent elements of the world: objects, regions of space, groups of elements, parts of elements, etc.. The arcs of the AH-graph are the sets $\{A^0, A^1, \ldots, A^{k-1}\}$. An arc $a(n_i, n_j) \in A^r$, going from $n_i \in N^r$ to $n_j \in N^r$, represents the fact that the node n_i is related to the node n_j.[11]

Level L^0 is the *lowest hierarchical level* of the AH-graph. It represents the world with the maximum amount of detail that is available. L^{k-1} is the *highest hierarchical level.* It represents the world with the minimum amount of detail, usually as a single node. A set of nodes in N^i $(i < k-1)$ is represented by a node in N^{i+1}, called its *supernode.* Similarly, a set of arcs in A^i $(i < k-1)$ is represented by an arc in A^{i+1}, called its *superarc.* These connections among multiple views of the world (hierarchical levels) are formally established using the summarization function for nodes s_n^i and the summarization function for arcs s_a^i.

s_n^i is a function from N^i into N^{i+1} (if $i = k-1$, the image set contains a virtual node that represents all the nodes of L^{k-1}). s_n^i yields the supernode of N^{i+1} which represents a given node of N^i. Similarly, the function s_a^i yields the superarc of A^{i+1} representing a given arc in A^i. The domain of s_a^i is the set of arcs $a(n_s, n_t) \in A^i$ that satisfy $s_n^i(n_s) \neq s_n^i(n_t)$. Its image is a set of arcs in A^{i+1}. The function is not defined in L^{k-1}.

The inverses of the functions above, $[s_n^i]^{-1}$ and $[s_a^i]^{-1}$, are interesting as well (for example, in order to refine a path), since they provide more detailed information about a given node or arc.

Costs The costs (weights) assigned to the arcs of each hierarchical level are defined as numeric intervals, called *cost intervals*:[12]

$$\mathbf{i} = [i^-, i^+], \text{ where } i^+ \geq i^- \geq 0,\ i^+ \neq \infty$$

The sum and propagation operators on cost intervals are defined as follows:

$$\mathbf{i} + \mathbf{j} = [i^- + j^-, i^+ + j^+]$$

$$\overset{*}{\bigwedge}\{\mathbf{i_1}, \mathbf{i_2}, \ldots, \mathbf{i_r}\} = \mathbf{i_1} \wedge \mathbf{i_2} \wedge \ldots \wedge \mathbf{i_r}$$
$$\text{where } \ \mathbf{i} \wedge \mathbf{j} = [min(i^-, j^-), max(i^+, j^+)]$$

The cost of an arc $a(n_i, n_j)$ is given by the function $arc\_cost(a(n_i, n_j))$. This value is calculated by *propagating* the cost of subarcs of $a(n_i, n_j)$ (if $a(n_i, n_j) \in$

[11] Notice that arcs only exist between nodes in the same level of the hierarchy.

[12] Cost intervals are denoted in bold.

A^0 then its cost is assumed to be given). Given an arc $a(n_i, n_j) \in A^r$ with a non-empty set of subarcs $[s_a^{r-1}]^{-1}(a(n_i, n_j))$,

$$arc\_cost(a(n_i, n_j)) = \overset{*}{\bigwedge}\{arc\_cost(a(n'_i, n'_j)) : a(n'_i, n'_j) \in [s_a^{r-1}]^{-1}(a(n_i, n_j))\} \quad (9)$$

Hierarchical path search. In [11] the issue of hierarchical path search is studied in-depth based on the AH-graph model. In order to find a path from node a to node b, the method is to construct the sequences of containing supernodes above the two nodes, and find the smallest common containing supernode. Then, proceeding downward in the two sequences, look for the solutions to the problem indexed under pairs of disjoint containing supernodes. It is proved that there exists a Sufficient Condition for Optimality (SCO) in the structure of an AH-graph under which the former algorithm always finds optimal paths.

Annotations. In addition to the hierarchical and topological structure of the AH-graph, both arcs and nodes can hold an undetermined number of *annotations*: blocks of data that contain any type of information. This provides a simple way of attaching procedural, physical or geometrical information to the basic topological skeleton. An annotation is a pair *(identifier, data)*. Every annotation has a unique identifier (the key). The data indexed by an identifier id under a node $n_j \in N^i$ is denoted by $data\_node(n_j, id)$. Similarly, data annotated in arcs is denoted by $data\_arc(a(n_i, n_j), id)$.

5 Using AH-Graphs to implement the SSH topological level

The SSH hierarchy of regions can be easily implemented using the AH-graph model. SSH's *regions* are represented by *nodes* in an AH-graph. The containment relation among regions is represented by the summarizing function s_n^i. Similarly, the lifting of a path is represented by the summarizing function s_a^i. Finally, SSH's metrical information is represented by *annotations*. Using AH-graphs to implement the SSH's regions provides the following advantages:

- There are several efficient algorithms to perform basic operations on an AH-graph: hierarchical path searching, automatic hierarchy construction, importing data from other graph especification format, etc..
- We have implemented a friendly graphical interface that allows the user to maintain an AH-graph, providing complete editing, storing, and import/export operations.
- The AH-graph model has been implemented as a module of a robot architecture that uses a new software called NEXUS [12, 13]. This software allows programmers to integrate a number of modules that perform different operations: navigation, object manipulation, perception, etc..

Next we describe the main issues of the SSH's regions implementation.

1. The cost associated with an arc represents the distance between regions. The cost of arcs between nodes in L^0 is derived from the actual travelling of the agent in the environment. The cost of any superarc is calculated using equation (9), which turns out to be equivalent to our specification in axiom (5). The AH-graph implementation ensures that any change in an arc's cost is propagated to its superarcs.
2. Every node $n \in N^i$ holds an annotation

$$("frame", frame\_data)$$

where $frame\_data$ contains information about the frame of reference associated with node n, and the location of the different subnodes of n w.r.t. this frame. These locations are assigned as describe in Footnote 9. The AH-graph provides an automatic procedure that updates $frame\_data$ whenever metrical information about subnodes of n change.[13]
3. Every node n_i holds an annotation

$$("relative\_angles", \{(arc_0, arc_1, angle_{01}), (arc_1, arc_2, angle_{12}), \ldots, \})$$

where the initial node of arc_i is n_i. A triple $(arc_s, arc_t, angle_{st})$ defines the angle from arc_s to arc_t measured from the point of view of the node n_i.

4. Every node n_i holds an annotation

$$("absolute\_angles", \{(arc_0, absangle_0), \ldots\})$$

where the pair $(arc_j, absangle_j)$ indicates the absolute angle of the arc arc_j with respect to the frame of reference of n_i.

The implementation of the SSH hierarchy of regions is completed by creating in the AH-graph nodes and arcs equivalent to the SSH 's places (regions) and paths. The predicates defined in Section 3 have an AH-graph counterpart satisfying the SSH's axiomatic specification.

6 Building Spatial Hierarchies

There is not a unique criterion for grouping places into regions. Regions are often defined in terms of legislative boundaries, visual texture, typical activities, ethnic composition, and other characteristics that are not strictly aspects of spatial cognition. Having said that, next we present some algorithms used in our robot experiments to build spatial hierarchies.

[13] Notice that, if the angles between arcs and the costs of the arcs are not contradictory inside a given supernode, they lead to the definition of a local position for every subnode of that supernode. In the real world, however, an agent has to deal with errors in these measurements that lead to contradictions and therefore, errors in the specification of such frames of reference. In this paper we assume that these contradicitons are eliminated inside any supernode.

1. Optimality Edge Elimination Test. This algorithm provides a tool for constructing incrementally a hierarchy that satisfies the Sufficient Condition for Optimality (SCO). A hierarchical search algorithm that takes advantage of this condition can obtain reductions in the computational cost of about 89% w.r.t. a plain graph search algorithm using the lowest hierarchical level.
2. Optimal Search Hierarchy Approach. This algorithm constructs a hierarchy that is an estimate of a hierarchy which satisfies the SCO. It works by selecting the arcs of the plain graph with the greatest costs. These arcs are considered to be "external" arcs which connect different regions. The nodes of such arcs define new regions at the next higher hierarchical level.
3. Neighborhoods Creation. This method creates regions by growing up initial seeds, until a given size of the regions is reached. A regions r is a set of connected places such that if a place p is in r and the distance from p to place q is less than a given threshold, then place q is in r. Formally,

$$\begin{aligned} & in\_region(p,r) \wedge on\_path(p,pa) \wedge \\ & \{path\_order(pa,p,q) \vee path\_order(pa,q,p)\} \wedge \\ & path\_distance(pa,p,q,d) \wedge d \leq \epsilon \rightarrow in\_region(q,r) \end{aligned} \tag{10}$$

4. Hybrid User-Machine Hierarchy Creation. Whenever the agent enters a region of the environment that a person recognizes as a different region from the one that the agent has recently visited, that information is provided to a regions-database. This database is checked out for inconsistencies. If there are no contradictions in the data, that information can be used to create regions.
5. Topological Examination. This algorithm finds in the plain graph certain structures that can be represented as supernodes. Typical structures can be: nodes connected by a path of width 1, cycles, sets of nodes that are connected to the rest of the graph just by one arc, etc..

In order to overcome the limitations of each of the methods mentioned above, a hierarchical extraction procedure should be the result of integrating these individual procedures.

Example 2.

Consider the T-environment depicted in Figure 3a. At a the agent starts following the corridor until it reaches b, where the right wall disappears. Suppose that the agent decides to follow the left wall up to c, where the agent notices that the right wall reappears. e is a place at which the agent losses track of the right corner close to b. Suppose that at b the agent turns w.r.t. the right corner until it reaches g. f is a place at which the agent losses track of the left wall. The set $\{a,b,c,d,e,f,g,h\}$ will be the set of distinctive states associated with this environment. Each distinctive state has associated a topological place, which we will denote by the corresponding capital letter.[14] Two paths exist in

[14] In the example it is the case that only one distinctive state is associated with a topological place. In general this is not the case. Should the robot rotate at b, there will be another distinctive state b' in which the robot faces in the direction of a.

this environment, $\langle A, B, E, C, D\rangle$ and $\langle B, F, G, H\rangle$.

Notice that the places B, C, E, F and G are very close to each other. We can automatically set a threshold such that these places become a region, RB. By virtue of axiom 10 (and the choice of our threshold) place A gets associated to a region RA whose only place is A. Similarly, places D and H get associated with regions $RD = \{D\}$ and $RH = \{H\}$, respectively. Once these regions have been created, the path $\langle A, B, E, C, D\rangle$ can be lifted to the path $\langle RA, RB, RD\rangle$, and the path $\langle B, F, G, H\rangle$ can be lifted to the path $\langle RB, RH\rangle$. The resulting topological map has two paths and four places, as illustrated in Figure 3b.

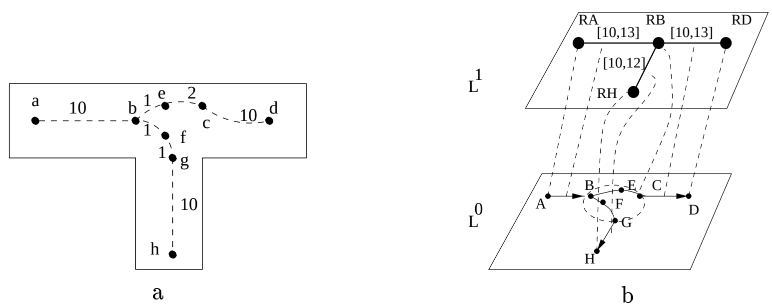

Fig. 3. Places $\{b, c, e, f, g\}$ are grouped into the region RB. Distance among places is annotated in the corresponding edges. The angle between paths is the one suggested by the figure. (b) shows the resulting AH-graph associated with (a).

Notice that the path $\langle RA, RB, RD\rangle$ enters region RB at place B and leaves it at place C. As for path $\langle RB, RH\rangle$ it enters RB at place B and leaves it at place G. Let's suppose that the heading of path $\langle A, B, E, C, D\rangle$ at place C is 0 degrees. A frame of coordinates for region B will indicate that the heading of path $\langle B, F, G, H\rangle$ at place G is about -90 degrees. Consequently, we can deduce that the angle between paths $\langle RA, RB, RD\rangle$ and $\langle RB, RH\rangle$ at place B is about -90 degrees. The resulting map is indicated in Figure 3b. { end of example }

Our methods to create hierarchical maps have proven to be adequate for most office-like environments. The important fact when working with a region hierarchy is to establish what properties are preserved when moving up and down the hierarchy. These properties will allow us to establish the soundness of reasoning mechanisms based on regions as well as characterize the typical errors people might make when using regions to infer properties of the places in them.

7 Related work

The SSH is a computational theory of the cognitive map There are many other proposals in the same spirit of the SSH [4, 31, 33, 6, 28, 16, 34, 8, 38]. Most of these proposals agree on the key characteristics of the cognitive map: the use of multiple frames of reference, qualitative sense of metrical information, and connectivity relations among landmarks. Ideas stemming from these theories have been used in robotics for map building and robot navigation (see [18, 5]).

The idea of using hierarchies of abstraction for reducing the computational cost of certain operations is old and common to many disciplines: GIS [19], graph theory [17], planning [37, 21, 1, 15, 1], robotics [25, 9, 11], etc.. It is inspired by the way humans solve problems. The AH-graph is an implementation of a hierarchy of abstraction differing from those in that it includes a representation of uncertainty in some values (the costs assigned to the relations between elements) and it models directly the environment of an agent, not a state space. It is a simplification of a more complex scheme, the Multi-AH-graph model, which allows to model more than one hierarchy of abstraction on the same set of ground data.

One of the unsolved problems on abstraction is how to build automatically the abstraction levels. There exist a few approaches to this problem in the planning literature [1, 15, 21, 17] that allow one to build "good" hierarchies automatically. The "good" concept refers to the reduction in computational cost of some operation, usually path finding. In the particular case that the information is modeled using a graph-like representation, building automatically hierarchies from a plain graph is directly related to clustering the plain graph. This often leads to NP-complete problems [14] unless the desired objectives or the structure of the plain graph are constrained [21, 3, 2, 7] or some knowledge on the domain of the problem is used [27, 20, 30].

8 Conclusions

This paper extends a computer model of the human cognitive map, the Spatial Semantic Hierarchy (SSH). The SSH is enhanced with the formalization of a hierarchical representation of space. This formalization is compatible with the topological level of the cognitive map represented by the SSH. We presented an implementation of this axiomatic theory using the AH-graphs model.

A hierarchy of space allows an agent to perform operations more efficiently than using a plain map. In particular, spatial reasoning is improved by reducing the number of elements involved in a plan. The implementation of the SSH's regions using AH-graphs provides mechanisms for efficiently finding chains of relations between elements of the map, as well as friendly access to the SSH's topological map.

We also described some methods that allow us to design several algorithms to automatically build a hierarchy of space. This is a complex problem that seems not to be solvable by a simple procedure. We are exploring hybrid approaches to combine the different methods proposed in this paper.

References

[1] F. Bacchus and Q. Yang. Downward refinement and the efficiency of hierarchical problem solving. *Artificial Intelligence*, 71:43–100, 1994.

[2] F.J. Brandenburg. Graph clustering I: cycles of cliques. In *Fifth symposium on graph drawing*, 1997.

[3] A. Brandstädt, V.D. Chepoi, and F.F. Dragan. Clique r-domination and clique r-packing problems on dually chordal graphs. *SIAM journal on Discrete Mathematics*, 10(1):109–1127, 1997.

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

[5] Kortenkamp D., Bonasso R.P., and Murphy R. *Artificial Intelligence and Mobile Robots.* AAAI press, 1998.

[6] E. Davis. The Mercator representation of spatial knowledge. In *AAAI-93*, 1993.

[7] S. Dutt. New faster kernighan-lin-type graph-partitioning algorithms. In *Twenty-eighth annual ACM symposium on theory of computing*, pages 603–611, 1996.

[8] Sean P. Engelson and Drew V. McDermott. Maps considered as adaptive planning resources. In *AAAI Fall Symposium on Applications of AI to Real-World Autonomous Mobile Robots, Working Notes*, Cambridge, MA, October 1992.

[9] C. Fennema, A. Hanson, E. Riseman, J.R. Beveridge, and R. Kumar. Model-directed mobile robot navigation. *IEEE transactions on Systems, Mans and Cybernetics*, 20(6), 1990.

[10] J. Fernandez and J. Gonzalez. A general world representation for mobile robot operations. In *Seventh conference of the Spanish association for artificial intelligence (CAEPIA-97)*, 1997.

[11] J. Fernandez and J. Gonzalez. Hierarchical graph search for mobile robot path planning. In *IEEE international conference on Robotics and Automation (ICRA'98)*, 1998.

[12] J. Fernandez and J. Gonzalez. NEXUS: A flexible, efficient and robust framework for integrating the software components of a robotic system. In *IEEE International conference on Robotics and Automation (ICRA'98)*, 1998.

[13] J. Fernandez and J. Gonzalez. The NEXUS open system for integrating robotic software. *Robotics and Computer Integrated Manufacturing*, 1999. In Press.

[14] M.R. Garey and D.S. Johnson. *Computers and Intractability: A guide to the theory of NP-Completeness.* W.H. Freeman and Co., New York, 1979.

[15] F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial Intelligence*, 57(2-3):323–389, 1992.

[16] S. Gopal, R. L. Klatzky, and T. R. Smith. Navigator: a psychologically based model of environmental learning through navigation. *Journal of Environmental Psychology*, 9:309–331, 1989.

[17] R.C. Holte, T. Mkadmi, R.M. Zimmer, and A.J. MacDonald. Speeding up problem solving by abstraction: a graph oriented approach. *Artificial Intelligence*, (85):321–361, 1996.

[18] Borestein J., Everett H.R., and Feng L. *Navigating mobile robots: systems and techniques*. A. K. Peters, Wesllesley, Massachusetts, 1996.

[19] N. Jing, Y. W. Huang, and E.A. Rundensteiner. Hierarchical optimization of optimal path finding for transportation applications. In *Fifth international conference on information and knowledge management (CIKM'96)*, pages 261–268, 1996.

[20] N. Jing, Y.W. Huang, and E.A. Rundensteiner. Effective graph clustering for path queries in digital map databases. In *Fifth international conference on information and knowledge management (CIKM'96)*, pages 215–222, 1996.

[21] C.A. Knoblock. *Automatically Generating Abstractions for Problem Solving*. PhD thesis, Computer Sciences, Carnegie Mellon University, 1991. Technical Report CMU-CS-91-120.

[22] B. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978.

[23] B. Kuipers. A hierarchy of qualitative representations for space. In *Working papers of the Tenth International Workshop on Qualitative Reasoning about Physical Systems (QR-96)*. AAAI Press, 1996.

[24] B. Kuipers and Y. T. Byun. A robust qualitative method for spatial learning in unknown environments. In Morgan Kaufmann, editor, *AAAI-88*, 1988.

[25] B. Kuipers, R. Froom, W. Y. Lee, and D. Pierce. The semantic hierarchy in robot learning. In J. Connell and S. Mahadevan, editors, *Robot Learning*, pages 141–170. Kulwer Academic Publishers, 1993.

[26] B.C. Kuo. *Automatic Control Systems*. Prentice-Hall, Inc., fifth edition, 1987.

[27] M.T. Kuo and C.K. Cheng. A new network flow approach for hierarchical tree partitioning. In *ACM/IEEE design automation conference*, pages 512–517, 1997.

[28] David Leiser and Avishai Zilbershatz. THE TRAVELLER: a computational model of spatial network learning. *Environment and Behavior*, 21(4):435–463, 1989.

[29] V. Lifschitz. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 297–352. Oxford University Press, 1994.

[30] C.L. McCreary, J.J. Thompson, D.H. Gill, T.J. Smith, and Y. Zhu. Partitioning and scheduling using graph decomposition. In *Twenty-eighth annual ACM symposium on theory of computing*, 1996.

[31] D. McDermott. A theory of metric spatial inference. In *AAAI-80*, pages 246–248, 1980.

[32] D. McDermott. Spatial reasoning. In Shapiro S., editor, *Encyclopedia of artificial intelligence*, volume 2, pages 863–870. John Wiley, 1987.

[33] D. V. McDermott and E. Davis. Planning routes through uncertain territory. *Artificial Intelligence*, 22:107–156, 1984.

[34] Michael O'Neill. A biologically based model of spatial cognition and wayfinding. *Journal of Environmental Psychology*, 11:299–320, 1991.

[35] J. Piaget and B. Inhelder. *The child's conception of space*. New York, Norton, 1967.

[36] E. Remolina and B. Kuipers. Towards a formalization of the spatial semantic hierarchy. In *Fourth Symposium on Logical Formalizations of Commonsense Reasoning, London*, January 1998.

[37] E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974.

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