

Integrating Multiple Representations of Spatial Knowledge for Mapping, Navigation, and Communication

Patrick Beeson, Matt MacMahon, Joseph Modayil, Aniket Murarka,
Benjamin Kuipers, and Brian Stankiewicz

Department of Computer Sciences, Department of Psychology
The University of Texas at Austin

{pbeeson, adastra, modayil, aniket, kuipers}@cs.utexas.edu, bstankie@psy.utexas.edu

Abstract

A robotic *chauffeur* should reason about spatial information with a variety of scales, dimensions, and ontologies. Rich representations of both the quantitative and qualitative characteristics of space not only enable robust navigation behavior, but also permit natural communication with a human passenger. We apply a hierarchical framework of spatial knowledge inspired by human cognitive abilities, the Hybrid Spatial Semantic Hierarchy, to common navigation tasks: safe motion, localization, map-building, and route planning. We also discuss the straightforward mapping between the variety of ways in which people communicate with a chauffeur and the framework’s heterogeneous concepts of spatial knowledge. We present pilot experiments with a virtual chauffeur.

Introduction

How to represent knowledge is one of the most important questions for any research domain. The way an agent represents its world influences how it learns, how knowledge is transferred across tasks, and how knowledge is communicated—either between agents or to a human. In this paper, we present a hierarchical framework of spatial knowledge, the *Hybrid Spatial Semantic Hierarchy (HSSH)*, and discuss how it provides a rich, natural, and effective interface for human interaction with a mobile robot.

Specifically, we focus on a *chauffeur* robot, e.g. a smart wheelchair (Simpson 2005), that transports a human passenger. The chauffeur’s spatial knowledge must be rich and accessible, so the passenger can direct the robot through natural language route instruction, can trust the robot to move autonomously in routine situations, can answer requests for additional information made by the robot, and can query the robot about its behavior.

We believe a monolithic representation for spatial knowledge is inadequate for such challenging domains. For example, the most prevalent representations of spatial knowledge for mobile robots are global metrical maps of environments, e.g. occupancy grids or landmark maps (Thrun 2002). Though these large metrical maps are useful for autonomous mapping and navigation using range sensors, they are insufficient for human-robot interaction (Bos, Klein, & Oka 2003).

Humans represent space across a variety of complex abstractions (Siegel & White 1975; Golledge 1999; Kuipers

2000), yet communication is successful because people share similar categories and concepts about the world. To communicate, a chauffeur robot needs multiple abstractions of space, with concepts that correspond to the ontologies the passenger uses in thought and communication.

The HSSH allows a robot to describe the world using qualitatively different representations, each with its own ontology. The hierarchy of connected representations is useful for the many tasks of navigation: safe motion, localization, map-building, and route planning. Equally important, since the multiple representations are motivated by human cognitive abilities, they provide a “natural” way for a chauffeur robot to interact with a passenger. We show natural language, joystick, and GUI interactions with the chauffeur, using HSSH representations.

HSSH Overview

The Spatial Semantic Hierarchy (SSH) provides a hierarchy of abstractions for reasoning about *large-scale space* (Kuipers 2000). Large-scale spaces are environments that cannot be seen from any one location. The large-scale environments an SSH agent inhabits must have a certain structure. Specifically, the environments must be describable as sets of places connected by paths—the SSH is not applicable in a trackless desert.

The Hybrid Spatial Semantic Hierarchy is a more specific framework of spatial knowledge that differs from the SSH in several key ways. Most importantly, the HSSH incorporates knowledge of *small-scale space*. A small-scale space is the area visible within the agent’s sensory horizon. The HSSH factors spatial reasoning about the environment into reasoning at four levels: *local metrical*, *local symbolic*, *global symbolic*, and *global metrical* (see Figure 1).

The next four sections detail these heterogeneous representations of the HSSH framework, focusing on the concepts needed for a mobile robot to have spatial reasoning for effective human-robot interaction. A working map-building implementation of the HSSH provides concrete examples. We then discuss how the current HSSH implementation handles specific human-robot interactions.

Local Metrical Reasoning

Humans have reliable metrical models of their local surround (Golledge 1999). People can navigate through com-

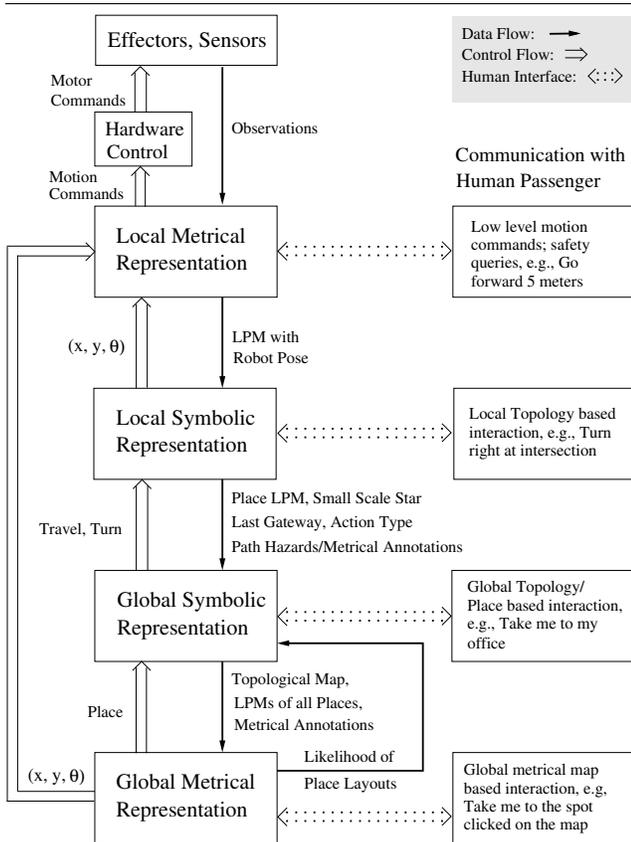


Figure 1: **HSSH description.** The HSSH is an integrated framework of multiple, disparate representations of spatial knowledge. Each level of abstraction uses its own ontology with concepts motivated by human communication about spatial navigation.

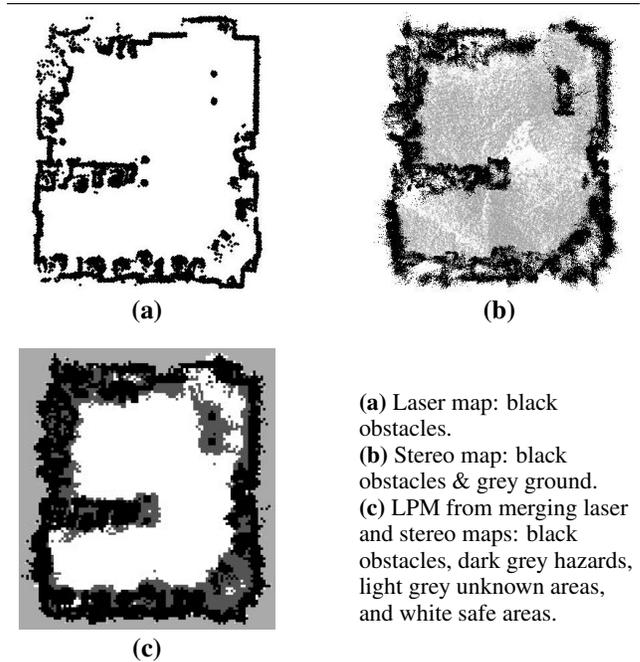
plex small-scale spaces and refer to intricate spatial relationships both verbally and graphically (Skubic *et al.* 2004).

Similarly, an agent uses the HSSH local metrical level to reason about the geometry of its immediate perceptual surround. At this level, several important navigation issues exist: recognizing safely navigable space, mapping the locations of fixed obstacles and hazards, identifying “non-static” obstacles (e.g. doors and pedestrians), planning local motion, and avoiding obstacles.

Local Perceptual Map (LPM)

We represent the geometric structure of an agent’s surroundings using a bounded 2D, scrolling metrical grid that remains centered on the robot. Regions on the grid are annotated as being obstacles, hazards, areas of caution, safe areas, and unknown areas. Such a grid, called the local perceptual map (LPM), captures the structural information required for safe navigation. The LPM is small enough that the agent can always localize itself within it, but large enough to model nearby obstacles and structure (e.g. 10×10 m).

We build the LPM using both lasers and vision (Murarka, Modayil, & Kuipers 2006). We use laser range-finders for incremental localization and mapping of obstacles in the 2D laser plane (occupancy-grid SLAM). Vision detects hazards



(a) Laser map: black obstacles.
 (b) Stereo map: black obstacles & grey ground.
 (c) LPM from merging laser and stereo maps: black obstacles, dark grey hazards, light grey unknown areas, and white safe areas.

Figure 2: **Local metrical modeling.**

and other obstacles in the surrounding 3D space of the robot. Dense stereo vision and visual feature-based methods build a 3D point cloud model of the surrounding space, used to identify safe and hazardous regions. The laser metrical map is fused with a projection of the 3D point cloud into the robot’s 2D travel plane, creating an LPM that models the hazards visible in both modalities. Figure 2 demonstrates building the LPM of a room.

The ability to identify “non-static” obstacles (Modayil & Kuipers 2004) makes it possible for the robot to identify environmental elements, such as doors, that can potentially allow navigation, even when they appear as obstacles in a vanilla occupancy grid implementation.

Control

Local metrical control is fairly low-level, consisting of planning safe motion in the LPM. Most commands from the passenger or from the local symbolic level are converted to pose coordinates in the LPM’s reference frame and a safe motion path is computed. Our implementation does A* search to find time-optimal motion paths. The motion path is constantly recomputed to respond to changing circumstances and to ensure safe progress towards the goal.

The computed motion path is broken into a sequence of open loop controls and sent to a lower-level control module. The control module rejects unsafe commands by quick analysis of the LPM, e.g. driving over a drop-off or moving forward into an obstacle. User instructions to slow down or to halt directly affect these control commands.

Interaction with a Human Passenger

At the local metrical level, the passenger may communicate to the robot through a variety of instructions/queries. Shared control is important at this level—for instance, the

robot should fine-tune coarse user commands to move safely through a door or stay on a sidewalk.

Some natural language instructions at this level are low-level motion commands, e.g. “rotate right,” “go forward five meters,” “halt.” With a labeled map, other instructions direct the chauffeur to named locations, to objects, to regions identified by local spatial relations, e.g. “between the chairs.”

The passenger may also indicate where they want to go by clicking or drawing on the LPM display. Displaying the LPM also allows the robot to improve its state of knowledge by asking the passenger about ambiguous or low information situations. For example, the robot can ask the passenger to provide safety information about uncertain regions or to verify the hazard annotations on the LPM.

Most often, we expect local metrical control to receive commands from passengers using a joystick. Joystick commands, translational and rotational velocity, are compared against the LPM for safety before sending modified velocity commands to the physical device.

Local Symbolic Reasoning

Humans also model their local surround symbolically by recognizing the navigational affordances of a place: the entrances and exits. People often refer to intersections by their shape (e.g. “T” and “corner”), or by how many paths pass through (e.g. “three-way intersection” and “dead-end”) (Geldof 2003; MacMahon, Stankiewicz, & Kuipers 2006).

A robot chauffeur should communicate about local space in the same terms. The chauffeur must recognize the ways in and out of the local space, both to detect when entering a place and to leave by the indicated exits. It must also identify the local topology for commands like “Take a left.”

Gateways

In the HSSH, a gateway is a boundary between the qualitatively different regions near the robot and away from the robot. Each gateway has two directions, inward and outward, termed *directed gateways*. Our current implementation uses the gateway finding algorithm defined by Beeson, Jong, & Kuipers (2005). The algorithm finds *constrictions* using the *skeleton* of free space in the LPM to identify the gateways as shown in Figures 3(a,b).

Path Fragments

Once gateways are found, the robot determines the local *path fragments*: portions of large-scale topological paths in the LPM. Each gateway is associated with exactly one path fragment, while each path fragment is associated with either one or two gateways. A path fragment that terminates in the LPM has one gateway; a path fragment that continues through the LPM has two gateways (Figure 3(c)).

The robot must determine whether a path fragment continues through the local area or terminates at the place. In our implementation, a path continues if each of two gateways is the clear unique default travel continuation of the other. This is implemented by checking if a ray normal to one gateway intersects only the other and vice versa.

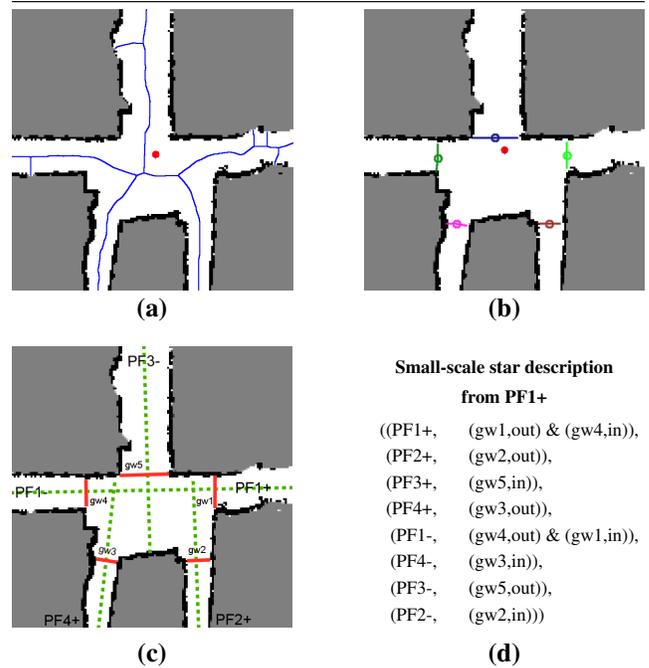


Figure 3: **Identifying Gateways & Local Topology in an LPM.** The robot is shown as a circle in the center of the LPM. (a) To find gateways, the algorithm computes the *skeleton of LPM free space*. The skeleton is examined to find *constrictions*. (b) The final *gateways* are represented as line segments connecting the skeletal constrictions (circles) with the closest obstacles. (c) Given gateways, *path fragments* are identified. (d) The *local topology* is described by the *small-scale star* for the place: an enumeration of local path fragments in a circular order.

Detecting Places

Using gateways and path fragments, we can formulate a robust criterion for detecting topological places. By the *path continuity criterion*, a robot moving along a path is surrounded by exactly two gateways, one in front and one behind, that define a single path fragment. If the number of gateways or path fragments changes, the robot has entered a topological place (Beeson, Jong, & Kuipers 2005). All intersections have at least two path fragments (even “L” intersections), while a dead-end has only a single gateway.

Because places are grounded in the LPM, the number and extent of places in an environment depends on the size of the underlying LPM.

Upon a confirmation of place detection, a snapshot of the LPM is stored. A pose in the stored snapshot is selected as the origin of that place’s frame of reference. Upon arriving at a place, the robot also stores its metrical estimation about the location of the current place in the frame of reference of the previous place. This metrical information is an annotation on the global symbolic topological map.

Describing the Local Topology

The *local topology* of a place is a model of the topological relations between the path fragments at that place. The *small-scale star* describes the local topology as the circular ordering of directed gateways and directed path frag-

ments (Figure 3(d)).

The small-scale star is constructed as follows. (1) Create a set of tuples $\langle PF, GW \rangle$. PF is a path fragment and a direction (+ or -). GW is the directed gateway (or pair of gateways) on PF facing the same direction. (2) Initialize the circular order with those tuples containing outward-facing gateways, in their clockwise sequence around the place. Both directed gateways of path fragments that pass through the place neighborhood, now appear in the order. (3) Each remaining tuple, containing an inward-facing gateway and a path fragment terminating at the place, is inserted into the order where the path would exit the place, as determined by the path continuity criterion.

Control

At the local symbolic level of the HSSH framework, navigation is abstracted to travels and turns. Turn and travel actions are abstractions of continuous motion through the LPM between directed gateways. To travel along a path, the robot decides which of the two gateways faces the desired direction and sends the coordinates of the gateway to the local metrical control. During a travel action, the gateways are constantly being recomputed, with new goal coordinates sent to the lower control. To turn at a place, the robot decides which outward-facing gateway is the correct exit, then sends its coordinates to the local metrical controller.

Interaction with a Human Passenger

A chauffeur needs to be able to categorize a place symbolically to handle several common spatial instructions. Some instructions rely on recognizing the topological position of gateways at places, for instance, “Take the second left.” Others rely on the entire local topology: “At the four-way intersection, turn right.” Finally, more complex local topologies require understanding the relationships in the small-scale star, such as the distinction between “veer right” and “take the sharp right” in a five-way intersection.

Global Symbolic Reasoning

Humans think about their global environment symbolically (Siegel & White 1975; Golledge 1999). For instance, in complex environments, people plan, carry out, and describe travel using symbolic places and paths (Geldof 2003). A chauffeur should recognize paths as coherent collections of ordered places and identify a familiar place upon returning from another direction.

By keeping a history of travel and turn actions, a robot can replicate routes it has traveled in the past. However, if the robot can recognize the same place seen at different times when traveling different routes, it can simplify its model of the world, creating more efficient routes between places. This problem in robotics is termed *loop closing*. *Perceptual aliasing* can create multiple loop closing hypotheses, causing *structural ambiguity*.

Topological Map-Building

Within the HSSH, the topological map-builder maintains a tree whose nodes are pairs $\langle M, x \rangle$, where M is a topological

After performing action a , arriving at directed gateway gw , and observing the resulting view $v = \langle \text{LPM}, \psi, gw \rangle$, for each map M and “you-are-here” pointer x on the fringe of the tree:

If M includes $\langle x, a, x' \rangle$ and $\text{view}(x', v')$,

- if $\langle \psi', gw' \rangle = \langle \psi, gw \rangle$, $\langle M, x' \rangle$ is the successor to $\langle M, x \rangle$;
- if $\langle \psi', gw' \rangle \neq \langle \psi, gw \rangle$, mark $\langle M, x \rangle$ as inconsistent.

Otherwise

- M does not include $\langle x, a, x' \rangle$. Let M' be M extended with a new distinctive state symbol x' and the assertions $\text{view}(x', v)$ and $\langle x, a, x' \rangle$
- If a is a travel action, consider the $k \geq 0$ dstates x_j in M such that the local topology of $\text{view}(x_j)$ matches the local topology of $\text{view}(x')$. $\langle M, x \rangle$ has $k + 1$ successors:
 - $\langle M' \cup \{x' = x_j\}, x' \rangle$ for $1 \leq j \leq k$, plus
 - $\langle M' \cup \{\forall j x' \neq x_j\}, x' \rangle$.
- If any one of the new successor maps violates any topological assumptions, mark it inconsistent.

Figure 4: Building the tree of topological maps. Each node in the tree $\langle M, x \rangle$ is a hypothesis of a topological map M and the current pose x (place location and orientation along a path). ψ represents the small-scale star model of the local topology for x .

map and x is a *distinctive state* (*dstate*) within M representing the robot’s current location. Each dstate corresponds to a particular place, path, and direction on that path; dstates in a place are connected by turn actions.

A *view* of a dstate is represented by the structure $v = \langle \text{LPM}, \psi, gw \rangle$, where ψ is the small-scale star description of its local topology and gw is the directed gateway at which the last action terminated. Views can be matched by testing whether the local topologies are isomorphic. Since view matching considers the entire local topology, ambiguity can only arise after travel actions.

The topological map-building algorithm, shown in Figure 4, creates a tree of map hypotheses. For non-trivial environments, the tree of maps grows too quickly to maintain in real time, so we build this tree in a best-first fashion.

Several algorithms can facilitate this best-first approach: apply a prioritized circumscription policy to sort symbolic map hypotheses (Remolina & Kuipers 2004); apply a constraint to remove non-planar topologies (Savelli & Kuipers 2004); compare the stored LPMs along with the local topologies; sort topological maps based on their maximum likelihood global layout (see below). We are currently evaluating the strengths and weaknesses of these approaches.

Control

Control at the global symbolic level consists of topological route planning and execution. The robot searches the preferred topological map to find a route between places. Executing a route consists of providing a sequence of travel and turn commands to the local symbolic level and replanning if the robot moves off the route or loses certainty over its current location in the topological map.

Routes can be optimized based on the situation or the user’s preferences. Metrical annotations of the distances between neighboring places can be used to find the shortest path. Hazards seen recently in the LPM when traveling a

path segment can also be considered during symbolic planning. In addition to using the best map hypothesis, the robot can account for structural ambiguity by planning routes that are applicable in several of the most probable maps. These routes can be chosen to avoid confusing areas or to resolve the ambiguity.

Interaction with a Human Passenger

The passenger can command the chauffeur at the global symbolic level in several ways. First, the passenger may ask the chauffeur to go to a place and rely on the topological route planning to determine how to get there. Or the passenger may want to specify a particular plan, such as “Take the quiet route to my office” where this route is stored from some previous time. Satisfying locative conditions in some commands, such as “At the corner, go left” and “Take the second right,” can be implemented at the global topological level, by planning to reach the precondition location, although they can be done at the local symbolic level with reactive control (MacMahon, Stankiewicz, & Kuipers 2006).

Distance queries are usually answered by using metrical annotations on the topological map rather than a global metrical map: “How far is the goal?” is interpreted as “How long is the route?” instead of “How far as the crow flies?”

Global Metrical Reasoning

Though humans have a difficult time drawing accurate maps of large-scale environments (Siegel & White 1975), people use external global metrical maps effectively to plan navigation, to gauge distances between places, and to communicate place locations and routes. The robot can create a usable global, metrical layout using a topological map of an environment, the stored snapshots of the LPM at each topological place, and the metrical annotations regarding neighboring places. Modayil, Beeson, & Kuipers (2004) present a general theory of building a global metrical map from a metrically annotated topological skeleton. See Figure 5 for an overview and example of the process.

Global Metrical Map-Building

The implementation used to make Figure 5(d) performed Metropolis-Hastings sampling over the global layout of places before integrating the data along the individual path segments. This ran offline and took some time to determine the maximum likelihood layout of the places.

If we assume that the uncertainty between neighboring places is Gaussian, finding the maximum-likelihood global layout essentially becomes an Extended Kalman Filter mapping problem (Smith, Self, & Cheeseman 1990); however, a place map contains significantly fewer states than a map with a state for every observation in the robot’s history, making global map-building much easier when the topological structure has been determined.

Control

At the global symbolic level of the HSSH framework, instructions are given by the user in global coordinates or stored names. Planning a path in the global metrical map

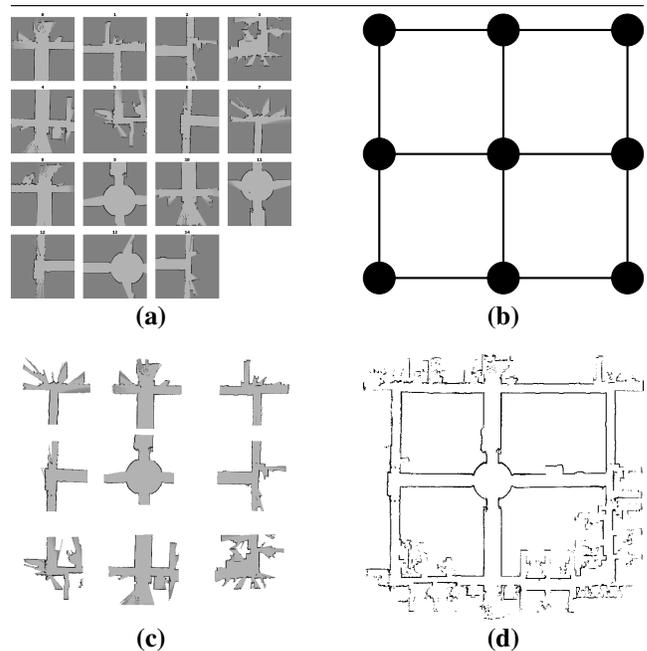


Figure 5: **Building the global metrical map.**
 (a) Store a snapshot of the LPM at every place detection event.
 (b) Find the best topological map consistent with the exploration.
 (c) Layout the stored LPMs on the topological map.
 (d) Find the layout of places, scale the distance along each path segment, and plot the observations seen traveling between LPMs.

is inefficient. Instead, this level determines the topological place closest to the goal state. This place is sent to the global symbolic controller, which in turn sends travels and turns to the local symbolic level, which in turn sends local coordinates to the local metrical level. Once the robot arrives at the place, it aligns the LPM’s frame of reference with the global map’s frame of reference and sends the goal coordinates directly to the local metrical control.

Interaction with a Human Passenger

The passenger may use the global metrical map as a dynamic graphical “you-are-here” map and instruct the robot to move to a location. The chauffeur may be commanded by clicking a location on a graphical display. Additionally, given annotations, relative positioning commands may be applied at the global level, such as “Go to the other side of this building.”

Human-Robot Interaction

Human route instructions include a variety of actions and descriptions. Instructions can steer low-level motion, guide larger-grained actions, or provide a goal for high-level navigation. Environmental descriptions can provide context or verification by specifying features of a location. Descriptions may also indirectly guide the robots toward the goal by providing necessary information in a declarative form. Additionally, the robot may ask for assistance with perceiving the environment. Finally, dialogue should include queries and feedback about the chauffeur’s knowledge, understanding, and capabilities.

To allow the passenger to communicate using the disparate representations presented here, a chauffeur robot must be able to communicate across a variety of modes. In the Simpson (2005) survey of smart wheelchairs, three communicating methods are common: (1) natural language interfaces (NLIs), (2) graphical user interfaces (GUIs), and (3) shared control using a joystick.

Following verbal instructions

We have implemented a natural language understanding and route-following system. While it has not yet been integrated with the full HSSH implementation discussed above, it has been tested using simulated robots and a large corpus of natural language route instructions (MacMahon, Stankiewicz, & Kuipers 2006). The instruction-following architecture has been tested using HSSH representations, such as the small-scale star, and navigates using local symbolic control: turn to the next gateway and travel to the next place. The route modeling and execution systems use knowledge of both language and space to infer the intended meaning of an instruction set. This is done by filling in implied actions and trusting extracted local topologies over mistake-prone turn directions and path distances.

Inferring topological maps from route instructions

Route instructions often do not specify the complete route, leaving navigational ambiguity. For instance, a turn direction might be unspecified, leaving topological ambiguity. Interestingly, these linguistic and perceptual ambiguities while following route instructions are analogous to ambiguity from perceptual aliasing in an exploration trace. Therefore, we can apply the same consistency filtering and map ordering algorithms from the HSSH global symbolic level to reason about topological maps from route instructions.

The HSSH can handle the ambiguous maps derived from under-specified or linguistically ambiguous route instructions using the same global symbolic reasoning that handles the spatial ambiguity in topological map-building. The current route instruction modeler forms an imperative model of instructions as plans consisting of turn and travel actions to be taken under the described conditions. Since turns and travels link gateways and places, inferring an under-specified topological route map from an instruction text is possible. The partial, ambiguous map of the environment derived from language understanding, and the partial, ambiguous map learned from exploration, can be represented and reasoned about in the same way by the same algorithms.

We believe that the same hierarchy of spatial representations will be useful in the future to incorporate sketch maps as a communication medium (Skubic *et al.* 2004).

Joystick and GUI control: Preliminary Data

In a pilot experiment, we were able to demonstrate a behavioral benefit for using both the local metrical and local symbolic levels of the HSSH implementation when assisting users with (simulated) degraded vision—low-vision people are a potential community of users for smart wheelchairs. Local metrical control was evaluated using a joystick interface, while local symbolic control was evaluated using a

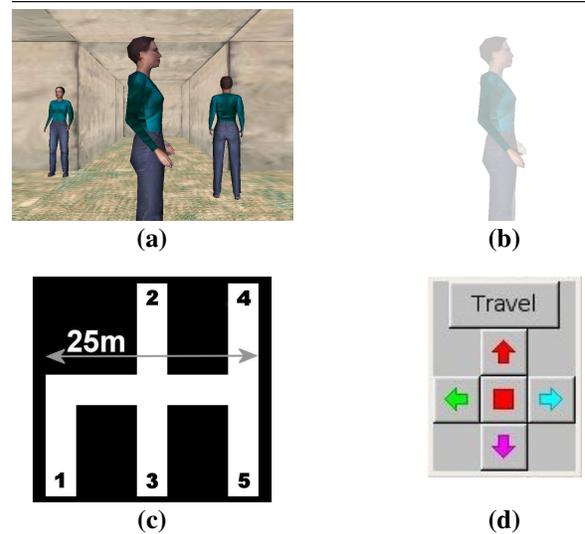


Figure 6: **Stimuli, Map, and Interface for Pilot Study.**

- (a) Subject's environment view in the Normal Vision condition.
- (b) Subject's limited view in Degraded Vision (fog) condition.
- (c) Environment layout used. Numbers correspond to goal states.
- (d) The local symbolic control GUI. Individual buttons light up when applicable for the current environmental situation.

GUI.

We tested the HSSH implementation in an virtual, indoor environment. The environment consisted of seven hallways with ten static human avatars placed randomly throughout the environment (Figure 6). We ran three subjects each in four conditions, defined by whether the participant's vision was obscured from fog (*Normal Vision* and *Degraded Vision*) and by the method of navigation (*Manual*, *Safety* and *Command*). The four conditions were *Normal-Vision: Manual*, *Degraded-Vision: Manual*, *Degraded-Vision: Safety* and *Degraded-Vision: Command*.

In the *Manual* conditions, participants moved through the environment using a joystick interface. The output of the joystick directly controlled the behavior of a virtual wheelchair.

The *Safety* condition tested the HSSH *local metrical* level. In this condition, participants navigated using a joystick, but the HSSH local metrical control modified the behavior of the virtual wheelchair relative to the joystick input. Specifically, local metrical control uses the LPM to enforce safe motion, by slowing down when close to obstacles and by refusing to perform unsafe actions (e.g., running into a pedestrian or wall).

The *Command* condition tested the HSSH *local symbolic* level. In the *Command* condition, an image showing the local topology of the current intersection was displayed on the screen. The user selected a command from four options (forward, left, right, backwards; see Figure 6(d)). The virtual wheelchair would then leave along the selected path segment and travel to the next intersection. As discussed earlier, local symbolic control passes through local metrical control, so the chauffeur also avoided obstacles in this mode.

We degraded the visual input for the participants by adding fog to the environment (see Figures 6(a,b)), such

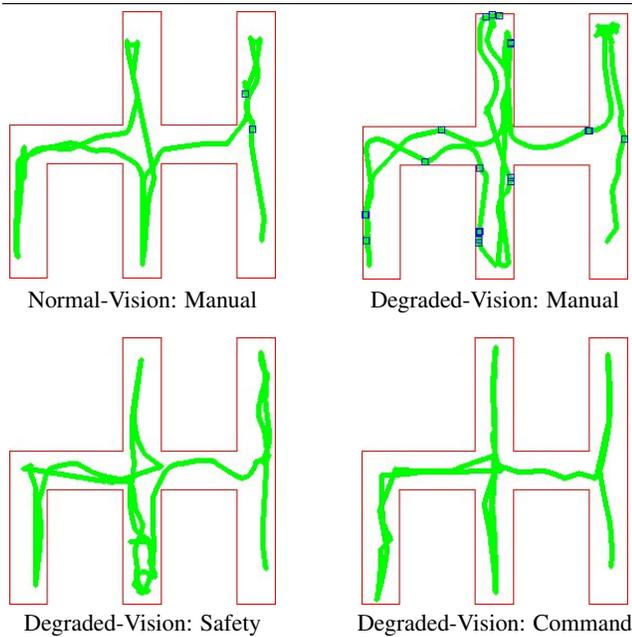


Figure 7: **Motion Trajectories.** Sample traces of the paths from a participant in the four conditions. Small squares indicate collisions.

that objects further than three meters from the viewpoint were not visible. The virtual wheelchair perceived the environment via (simulated) laser range-finders that viewed the world at about the height of the avatars' shins. Fog did not affect the distances returned by the simulated lasers.

We ran the participants in five trials per condition. A trial consisted of five goal locations presented in a random order. Each trial started by placing a participant at the same starting position (Position 3 in the map shown in Figure 6(c)). Each circuit of five places was run in each of the four conditions to control for the expected distance traveled and for time to complete a circuit. These twenty trials were randomly ordered for each subject in order to reduce the influence of task learning during the experiment.

All participants knew the layout of the environment, but not the locations of the obstacles (avatars), which were randomly distributed for each trial. At the beginning of a trial, a participant was told by the computer to go to a particular location (e.g., "Position 3. Go to position 5"). The participant used the joystick (or the GUI for the *Command* condition) to travel to the specified goal. When the participant reached the specified destination, the computer announced the goal name (e.g., "Position 5"), then gave the participant another goal location ("Go to Position 2").

Figure 7 shows sample recorded traces from one of the subjects in each of the four conditions. Figure 8 shows the mean distance traveled and the mean number of collisions (with both avatars and walls) on a circuit.

The Pilot Study addresses three primary questions:

Effect of Degraded Vision Does reducing the visual information by adding fog make the task more difficult?

Benefit of Assisted Joystick Control Is performance better with local metrical control (collision avoidance)?

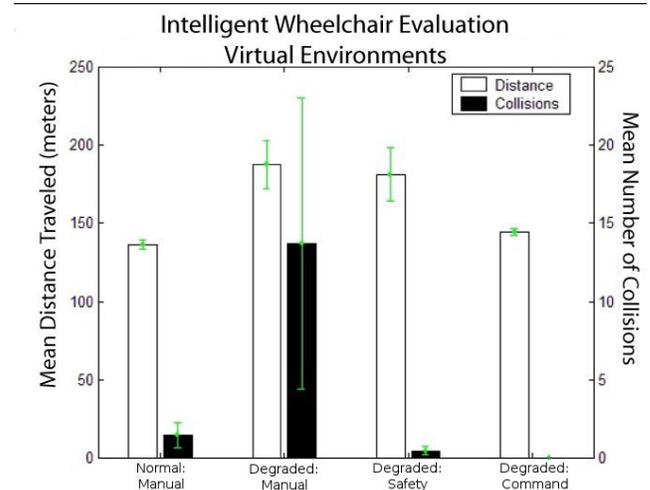


Figure 8: **Summary data for the Pilot Study's four conditions.** White bars show mean distance traveled for a circuit. Black bars are mean number of collisions. Error bars represent Standard Error.

Benefit of Local Symbolic Navigation Does the system perform better using local symbolic knowledge (obstacle avoidance and path planning) in the chauffeur?

Preliminary Results To answer these three questions, we measured the distance for subjects to reach all five goal states in a circuit and the number of collisions with either an avatar or a wall.

Driving a physical wheelchair with low vision can be difficult. Our first evaluation is whether degrading vision (by adding fog) had any effect on the subjects' performance using the virtual wheelchair. We can see the effect of degrading the visual input by comparing the performances in *Normal-Vision: Manual* versus *Degraded-Vision: Manual* in Figure 8. There was a 37% increase in the mean distance traveled (Normal = 136.5 meters; Degraded = 187.6 meters) and a 936% increase in the mean number of collisions (Normal = 1.5; Degraded = 13.7). This suggests that the degradation of vision made the task significantly more difficult.

To evaluate the benefit of adding local metrical control, we compared performances for *Degraded-Vision: Manual* versus *Degraded-Vision: Safety*. Adding safety showed no benefit for the distance traveled (Manual = 181.3 meters; Safety = 187.6 meters). However, collisions decreased by 97% (Manual = 13.7; Safety = 0.5).

Note all remaining collisions in *Safety* mode were with the hands of the avatars, which stuck out from the body (see Figure 6(a)) above the height of the wheelchair's range scan; thus, the hands were invisible to the chauffeur agent. This is an inherent problem with planar sensing devices, like laser range-finders, which motivates using stereo vision to improve LPMs in real-world scenarios (see Figure 2).

To evaluate the benefit of adding local symbolic control to the chauffeur, we compared performances for *Degraded-Vision: Manual* versus *Degraded-Vision: Command*. In this case, adding the local topology navigation aid reduced the distance traveled by 23% (Command = 144.3 meters; Man-

ual = 187.6 meters). Furthermore, collisions were avoided entirely (Command = 0.0; Manual = 13.7).

Comparing the performance for *Degraded-Vision: Command* versus *Normal-Vision: Manual* contrasts the autonomous chauffeur system to a vehicle driven by a passenger with full visual abilities. We find virtually no difference in the distance traveled (*Degraded-Vision: Command* = 144.3 meters; *Normal-Vision: Manual* = 136.5 meters), with a slight benefit with the HSSH local symbolic control in terms of the number of collisions (*Degraded-Vision: Command* = 0.0; *Normal-Vision: Manual* = 1.5). In this preliminary experiment, the autonomous system drove as well as the human with full vision.

Conclusion

The Hybrid Spatial Semantic Hierarchy (HSSH) is a framework that integrates multiple representations for spatial reasoning. Such a framework is necessary for a chauffeur robot, which must reason and communicate about the world across a variety of abstractions.

We have demonstrated an implementation of this framework for map-building. The local metrical level provides safe local path planning and metrical estimates of local space. The local symbolic level binds the local metrical map to symbolic entities such as entrances, exits, places, and paths. The global symbolic level tackles the problem of structural ambiguity in order to create a consistent topological map of a large-scale environment. The global metrical level builds a global metrical map on top of the topological skeleton by using the metrical annotations provided from the local metrical level.

The concepts used at each level of abstraction are motivated by concepts from studies on human communication and behavior during navigation tasks. We discuss an implementation of natural language understanding of human route instructions that uses the concepts in the local symbolic abstraction to control a simulated robot.

A particularly important remaining task is to fully integrate and rigorously evaluate the system presented here. We have shown through preliminary experiments that autonomous, hierarchical control can provide benefits over manual control, even for fully cognizant and physically abled subjects. Future experiments should compare local symbolic control and global symbolic control.

Assistive technologies, such as a chauffeur agent on a smart wheelchair, provide a compelling application for intelligent systems. We believe that the complexity of the task requires the kind of modular knowledge representation and modular architecture that we have demonstrated here. We believe that the same structure will be essential to achieving the robustness of human common-sense knowledge in a wide range of other applications.

Acknowledgements

This work is supported in part by the National Science Foundation (IIS-0413257 and IIS-0538927), by the National Institutes of Health (EY016089), and by an IBM Faculty Research Award.

References

- Beeson, P.; Jong, N. K.; and Kuipers, B. 2005. Towards autonomous topological place detection using the Extended Voronoi Graph. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*.
- Bos, J.; Klein, E.; and Oka, T. 2003. Meaningful conversation with a mobile robot. In *Proc. of the Conf. of the Eur. Chap. of the Assoc. of Computational Linguistics (EACL)*.
- Geldof, S. 2003. Corpus analysis for NLG. In Reiter, E.; Horacek, H.; and van Deemter, K., eds., *Proc. of the Eur. Work. on Natural Language Generation (ENLG)*.
- Golledge, R. G., ed. 1999. *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. Baltimore, MD: The Johns Hopkins Univ. Press.
- Kuipers, B. 2000. The Spatial Semantic Hierarchy. *Artificial Intelligence* 119:191–233.
- MacMahon, M.; Stankiewicz, B.; and Kuipers, B. J. 2006. Walk the talk: Connecting language, knowledge, action in route instructions. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*.
- Modayil, J., and Kuipers, B. 2004. Bootstrap learning for object discovery. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.
- Modayil, J.; Beeson, P.; and Kuipers, B. 2004. Using the topological skeleton for scalable global metrical map-building. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.
- Murarka, A.; Modayil, J.; and Kuipers, B. 2006. Local safety maps for a wheelchair robot using vision and lasers. In *3rd Canadian Conf. on Computer and Robot Vision (CRV)*.
- Remolina, E., and Kuipers, B. 2004. Towards a general theory of topological maps. *Artificial Intelligence* 152:47–104.
- Savelli, F., and Kuipers, B. 2004. Loop-closing and planarity in topological map-building. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.
- Siegel, A. W., and White, S. H. 1975. The development of spatial representations of large-scale environments. In Reese, H. W., ed., *Advances in Child Development and Behavior*, volume 10. New York: Academic Press. 9–55.
- Simpson, R. C. 2005. Smart wheelchairs: A literature review. *Rehabilitation Research and Development* 42(4):423–436.
- Skubic, M.; Blisard, S.; Bailey, C.; Adams, J.; and Matsakis, P. 2004. Qualitative analysis of sketched route maps: Translating a sketch into linguistic descriptions. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics* 34(2):1275–1282.
- Smith, R.; Self, M.; and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*. New York: Springer-Verlag. 167–193.
- Thrun, S. 2002. Robotic mapping: A survey. In Lake-meyer, G., and Nebel, B., eds., *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann.