Learning Spatial Models for Navigation

Susan L. Epstein[1], Anoop Aroor[1], Matthew Evanusa[1], Elizabeth I. Sklar[2], and Simon Parsons[2]

[1] Hunter College and The Graduate Center of The City University of New York, New York NY 10065, USA
susan.epstein@hunter.cuny.edu, aaroor@gc.cuny.edu, matthew.evanusa@gmail.com
[2] University of Liverpool, Liverpool, UK
e.i.sklar,s.d.parsons@liverpool.ac.uk

Abstract. Typically, autonomous robot navigation relies on a detailed, accurate map. The associated representations, however, do not readily support human-friendly interaction. The approach reported here offers an alternative: navigation with a spatial model and commonsense qualitative spatial reasoning. Both are based on research about how people experience and represent space. The spatial model quickly develops as the result of incremental learning during travel. In extensive empirical testing, qualitative spatial reasoning principles that reference this model support increasingly effective navigation in a variety of built spaces.

Keywords: navigation, learning, spatial model, heuristics, qualitative spatial reasoning

1 Introduction

A person who travels without a map to multiple locations (*targets*) relies on local perception to build a mental model that supports her goals. That model is replete with *spatial affordances*, abstractions that remove perceived but irrelevant details [11] and support spatial reasoning. The thesis of our work is that, despite sensor noise and actuator uncertainty, an autonomous robot can quickly learn to travel effectively when it too relies on commonsense qualitative spatial reasoning and models spatial affordances. This paper reports on *SemaFORR*, a hierarchical architecture for autonomous robot navigation. SemaFORR both makes navigation decisions and identifies spatial affordances. The principal results reported here are SemaFORR's ability to learn a serviceable spatial mental model from spatial affordances quickly, and to navigate with that model to a sequence of targets with increasing effectiveness.

The robot begins from a corner of its environment and then tries to reach a set of targets in a prespecified order as quickly as possible. Instead of a map, the robot has limited local perception, that is, it senses obstructions in a few directions and only in its immediate vicinity. Each time the robot finishes with a target, SemaFORR analyzes the perceptual history from that trip to learn and refine a spatial mental model of the environment. The revised model then serves as input for navigation to the next target.

Navigation is viewed here as a sequence of actions selected one at a time. To select the robot's next action, SemaFORR pragmatically capitalizes on the synergy among commonsense spatial rationales. Each rationale is a reactive procedure whose input is the robot's current percepts and spatial model. Some rationales (e.g., "move to the target you perceive directly before you") are applicable to any environment. Other rationales (e.g., "move through this exit") exploit affordances present in the model. The resultant system is transparent, human-friendly, and could advance human-robot collaboration.

The environments investigated here are three small, built spaces with different topologies, and a real-world indoor space of considerably greater complexity. By construction, SemaFORR can operate either physical robots in our laboratory or simulated ones on our screens. The thorough and extensive empirical work reported here, however, would have dramatically taxed our robot hardware. It also would have required considerably more elapsed time to recharge and recalibrate each robot periodically. The results reported here, therefore, are in simulation, but with realistic actuators that may shift the robot somewhat more or less than intended, as they do the physical robots.

The next section of this paper summarizes related work in intelligent architectures and robot navigation. Subsequent sections describe how SemaFORR decides and learns, and provide the experimental design and results. The paper closes with a discussion that considers the ramifications of our system decisions, and outlines our current work.

2 Related Work

FORR (FOr the Right Reasons) is a general architecture for learning and problem solving [7]. A FORR-based program is built to learn quickly, adapt rapidly, and restructure its own decision process. These properties provide robustness in complex, unpredictable situations. FORR was confirmed as cognitively plausible on human game players [29], and has since learned successfully in a variety of application domains, including game playing, constraint satisfaction, and human-human dialogue.

Ariadne was an early FORR-based application for a simulated robot in a grid [8, 9]. Ariadne's task was idealized, however. The robot operated alone in a static environment. Its sensors had no range limit and its actuators were noise-free. The robot moved perfectly and only orthogonally. It had no physical footprint; instead it occupied an entire grid cell. Moreover, what Ariadne learned, while intuitively appealing, was not based on what we now know about how people represent and experience space. As a result, Ariadne fared best in random environments without organizational principles or in environments with extensive, centralized open space. Environments built for people (e.g., a set of offices) proved considerably more difficult for Ariadne.

In contrast, SemaFORR is intended for dynamic, partially observable environments including complex office buildings, warehouses, and search and rescue. In such environments, maps may be unreliable for path planning, and landmarks may be obscured or obliterated. Communication may be sporadic and slow, sensors and actuators noisy, and barriers or passageways unanticipated. Moreover, a SemaFORR robot, while autonomous, is intended to work with others.

SemaFORR is part of the *HRTeam* (Human-Robot Team) project, where a person collaborates with a set of heterogeneous, autonomous, low-end physical robots. HRTeam's long-term goal is to support the person as the team investigates environments presumed unsafe for people. The person contributes to decision making but does not control it. This motivates our approach, in which each robot uses commonsense qualitative spatial reasoning and a spatial mental model to determine its actions. The HRTeam framework includes software to assign targets, a central server for communication, a shared knowledge store for the components of the spatial model, and a controller for each robot. Because HRTeam's software framework is built on Player/Stage [13], physical and simulated robots use the same decision maker (which selects actions in the controller), and the same driver (which sends commands from the controller to the robot's motors) (See [31] for further details on HRTeam.)

SemaFORR draws from research on how people experience, remember, and move through space (e.g., [16]). Its spatial model is inspired by what researchers now know about human spatial perception and navigation. Instead of an image-like metric map, people rely on what appears to be a gradually acquired collage of different kinds of knowledge [36]. Because metrically or topologically impossible environments do not deter people [41], neurophysiologists have suggested that human mental models remove perceived but irrelevant details [11]. For clarity, additional related cognitive work is cited in the next section.

SemaFORR's cognitive underpinnings have led to key differences from traditional approaches to robot navigation. The state-of-the-art approach to navigation in mobile robotics is to construct a detailed metric map using probabilistic *SLAM* (simultaneous localization and mapping) [2, 6]. Once a map has been constructed, the robot can *localize* (find its position there) and plan a path between any two points on the map. While the robot constructs the map, it also localizes within the map segment it has constructed (hence "simultaneous") and can plan paths within that segment. Plans can also be constructed to points outside the map segment, although unknown features may interfere with their execution. This can be somewhat mitigated if path planning and execution are combined with obstacle avoidance. SemaFORR has no map.

A purely reactive robot navigation architecture can support modular software design and flexibility in an environment not specifically structured for it. Such an architecture uses "if $\langle sensor$ - $value\rangle$ then $\langle action\rangle$" rules to select actions. To cover a variety of reactive behaviors, early approaches relied, for example, on subsumption architectures [5] or potential fields [1]. Subsumption architectures, however, require careful engineering to order all their applicable rules, and neither subsumption nor potential fields learns spatial features in the environment. Instead, SemaFORR's decision process integrates obvious correct reactions (e.g., "don't move into a wall") with commonsense qualitative spatial reasoning principles (e.g., "move in the direction of the target"). Many of these responses reference SemaFORR's mental model. Layered robot architectures typically partition control based on functionality (e.g., with layers for reactive feedback control, planning, and low-level action selection [12]). In contrast, SemaFORR makes only low-level action decisions.

Traditional robot navigation architectures can afford to be reactive because they rely on a *plan*, a sequence of *waypoints* the robot should go to on its way to its target (e.g., [24]). The A* algorithm [18] produces optimal paths, but it explores many alternatives and assumes full knowledge of a static environment. Despite a reliable map, however, a realistic environment may include noisy actuators, dynamic map changes, and other moving agents, all of which may necessitate plan repair or replanning. Rather than cache all pairs of shortest paths [4] or plan from previous searches [22], SemaFORR reacts to its local perceptions and its spatial model. In other work, HRTeam relies on a skeletal version of SemaFORR with an A* path planner that uses a global (i.e., full) map of the environment [10]. Here, however, we test the bounds of what a single robot can achieve alone, without a map and without a human or robot partner.

Finally, *semantic mapping* seeks to abstract spatial representations constructed for robots so that they can also support communication with people [23]. Most semantic mapping commits first to SLAM and then tries to explain its results in more human-friendly terms, often by augmentation of metric maps with objects (e.g., desks) or labels (e.g., "office"). Some work in semantic mapping deliberately steers the robot (e.g., [27, 38]); SemaFORR's robot is autonomous. Other work in semantic mapping is restricted to extremely simple environments with labeled training examples (e.g., [40]); SemaFORR's environments can, as in Section 5, be quite complex. In summary, semantic mapping performs inference on metric maps derived from sensor data, while SemaFORR derives affordances directly from sensor data. Thus, instead of recording obstructions, SemaFORR learns ways to facilitate navigation. SLAM addresses "where am I?" while SemaFORR addresses "why should I chose this action?" This approach supports more transparent reasoning and more natural communication with people.

3 SemaFORR

A SemaFORR robot's task is to *visit* (come within ϵ of) each of a pre-sequenced set of targets. To support this goal, SemaFORR learns a spatial model of its environment that emerges as it explores and reasons about space. This section explains SemaFORR's decision context and how it learns a spatial model. Then it describes how SemaFORR chooses an action and explains the individual components of that reasoning mechanism with a unifying example.

3.1 The Decision Context

At a *decision point*, SemaFORR selects the robot's next action. The robot's *action repertoire* is its set of possible actions: forward linear moves (henceforth, simply *moves*), clockwise and counterclockwise rotations (*turns*), and a pause (a no-op). Although the robot could theoretically make a move or turn of any size, SemaFORR restricts that choice to a discrete set of possibilities. The *intensity* of an action is a qualitative representation of how far the robot is intended to travel or turn. A move has intensity only between 1 and 5; a turn has intensity between 1 and 4, either clockwise or counterclockwise. Intensities are ordinal labels calibrated to correspond to a particular physical robot and its environment. Thus there are 14 possible actions in the robot's action repertoire.

The outcome of an action, however, is realistically non-deterministic. This paper focuses on the Surveyor SRV-1 Blackfin, a small platform in our laboratory with a webcam and 802.11g wireless. We have extensively observed and measured the actuator noise on a set of Blackfins there, and model it probabilistically here. As a result, when SemaFORR decides to act with intensity i, the robot acts with intensity $i \pm \delta$, where δ is an increasing function of i.

For localization, the robot relies on a system of overhead cameras, simulated here. The *position* of a robot is its location coordinates (x,y) and its orientation θ on the true map (henceforth, the *world*). The location of a target is specified with respect to the same coordinate system.

The robot's knowledge store is a set of descriptives that capture its experience, goals, and behavior. A *descriptive* is a shared data object with functions that determine how and when to update it. HRTeam's *DM* (Descriptive Manager) provides a shared knowledge store of descriptives for all team members. The DM receives messages from HRTeam's central server, extracts relevant data from them, and provides the current value of any descriptive to the robot on demand. Basic descriptives include the robot's position, its *agenda* (list of target locations to visit), its current target, and the history of its decisions made thus far on its way to that target.

The robot's percepts are represented as a descriptive called the *wall register*. It simulates a set of limited-range measurements for the distance from the robot to the nearest wall in 10 directions. From the robot's heading of 0°, these measurements are taken on either side at 8.87°, 17.5°, 37.2°, 74.5°, and 195°. In the example in Figure 1(a), not every ray touches an obstacle; some halt at their maximum range. Note that wall-register values are egocentric, while the positions of the robot and its targets are allocentric. (The walls also have a buffer that thickens them slightly for these measurements, to prevent unintended collisions from noisy actuators.) SemaFORR builds and refines its spatial model incrementally, from a history of its percepts and positions.

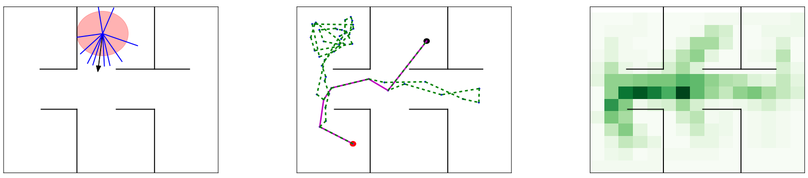
3.2 The Spatial Model

The components of SemaFORR's spatial model are spatial affordances, ways the environment provides opportunities to address a goal [14]. SemaFORR's affordances support its reasoning and its explicit representation about two-dimensional space (in the spirit, but without the finer granularity, of [20]). An affordance is calculated from sensor data and the robot's decision points; it describes spatial knowledge that supports effective navigation.

Instead of a map or a formal logic, SemaFORR's affordances summarize what it has experienced in its environmentas locations, lines, and areas. Each category of affordance is represented as a separate descriptive. SemaFORR has three kinds of affordances: trails with markers, regions with exits, and conveyors. Trails and conveyors are learned only from *successful travel* (i.e., immediately after the robot reaches its target); regions are learned whether or not travel succeeds.

A *trail* affords the ability to travel along a familiar, ordered sequence of locations. As the robot travels, the DM records its *path* as a sequence of *decision states*: the robot's position and the wall register values there. The trail-learning algorithm is analogous to the way people compute return paths [17], but with locations rather than landmarks or viewpoints. When the robot reaches its target, to derive the trail the algorithm processes a copy of that path backwards. At each decision state, the algorithm uses the wall register to identify a better (i.e., more direct) choice. The result is a (typically shorter) sequence of decision points that reduces the computational and physical effort required to travel between the target and the robot's starting point. An example appears in Figure 1(b).

The algorithm begins with a trail that is merely a copy $s_1, s_2, \ldots, s_k$ of the decision states that formed the path. Then, from s_k, the algorithm looks for the smallest i where the wall register at s_i perceived s_k. If it can find such an s_i it reduces the trail to $s_1, s_2, \ldots, s_i, m_k$, where the *trail marker* m_k is the location of s_k and the wall register values at s_k. This process repeats for each decision state along the trail, moving from the target backwards to the starting point. In the worst case, learning time is quadratic in the path length. The resultant trail is an ordered set of trail markers with line segments that connect them. Although a trail is likely to be suboptimal, it is more direct and has fewer digressions than the path from which it originates.



(a) (b) (c)

Fig. 1. (a) The robot's position produces values for its wall register, which provides its local view. The arrow represents the robot's heading; the subtended circle is the detected region. (b) A (dotted) path begun in the lower left room and the (solid) trail derived from it, with dots at the trail markers. (c) A conveyor grid; darker cells are more often crossed by trails.

A *region* is an obstruction-free local area. Wall register vectors are limited only by obstacles (e.g., walls) and the sensors' range. The robot only senses (i.e., produces wall register values) from a position where it is about to make a decision. At each such position, it computes a region as a circle whose center is its current location and whose radius is the length of its shortest wall register value, as in Figure 1(a). Regions are reminiscent of some human mental models [19, 30] and of areas in online mapping [35], but do not require that the robot map all walls first. As the robot travels, regions may gradually grow or shrink, but they never overlap. (This was a pragmatic design decision; there is a tradeoff between the number of regions and the cost to maintain and use them.) An *exit* from a region is a point on its circumference that intersects with a path.

Finally, a *conveyor* affords visitation to a small area that has often facilitated travel (similar to [25]). The *conveyor grid* covers the footprint of the world with cells about 1.5 times the size of the robot's footprint. The conveyor-learning algorithm tallies the frequency with which all trails pass through its cells. High-count cells in that grid are conveyors; they appear darker in the example in Figure 1(c). The descriptives form a knowledge store over which SemaFORR reasons.

3.3 The Reasoning Mechanism

SemaFORR selects one action at a time, that is, it does not plan. To select an action, it executes a *decision cycle*. At the beginning of a decision cycle, SemaFORR retrieves the current descriptive values from the DM and caches them on the robot. These include its position and the current target and spatial model. Then SemaFORR reasons about which action to choose from its 14-action repertoire. The output of a decision cycle is the selected action.

In SemaFORR, a *rationale* is a plausible reason to select an action. An *Advisor* is a boundedly rational (resource-limited) procedure that applies a rationale to evaluate actions. SemaFORR's use of multiple rationales is consistent with the recent result that multiple wayfinding strategies best predict human route selection [33]. The input to an Advisor is a set of possible actions and the descriptives' values. The output of an Advisor is a (possibly empty) set of *comments*, each of which expresses an opinion about the appropriateness of a single action from the perspective of the Advisor's rationale.

SemaFORR partitions its Advisors into tiers that correspond to Montello's distinction between locomotion (*tier 1*) and wayfinding (*tier 3*)[26]. As shown in Figure 2, a decision cycle first invokes the tier-1 Advisors in a predetermined order. One at a time, they have the opportunity to comment. In tier 1, Advisors' rationales are quick to compute and assumed to be correct. Each tier-1 rationale gives rise to a single Advisor that mandates or vetoes obvious reactions. If any tier-1 Advisor mandates an action, that becomes the decision and no further Advisors are consulted. If a tier-1 Advisor vetoes an action, it is eliminated from the set of possible actions passed to the next tier-1 Advisor.

Despite possible vetoes, tier-1 processing always retains some action, so that decision making always returns a value. If at any point in tier 1 only pause

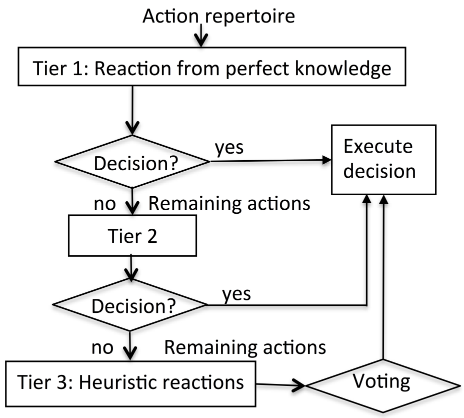


Fig. 2. A schematic for SemaFORR's control structure with sample input.

remains, it becomes the decision, and no other Advisors are consulted, that is, the robot does nothing until its next decision cycle. Otherwise, the unvetoed actions are forwarded to tier 3, which chooses among them. (Tier 2, which supports the selection of action sequences, is the focus of current development, and not used in the work reported here.)

In tier 3, SemaFORR alternately chooses a pause or a move on one decision cycle, and then a pause or a turn on the next. Thus, if in Figure 2 the previous decision cycle were for turns, only the unvetoed moves and pause would be forwarded to tier 3. Note that pause does not halt movement or sensing; it merely defers a decision to the next cycle, and thereby permits longer consecutive moves. The turns of intensity one serve the same purpose as the pause, that is, they permit longer turns in the same direction.

In tier 3, all Advisors comment before any decision is made. SemaFORR's tier-3 Advisors have deliberately disparate spatial rationales. To resolve their differences of opinion and to capitalize on the synergy among them, *voting* tallies the comment strengths from all tier-3 Advisors. When Advisor i comments on action j with strength s_{ij}, voting returns an action with maximum total strength:

$$argmax_j \sum_i s_{ij} \tag{1}$$

Ties in voting are broken at random. The remainder of this section explains the commonsense rationales and how they rely on local perception and the spatial model, along with a unifying example of a SemaFORR decision.

3.4 Tier-1 Advisors

Table 1 lists SemaFORR's 13 Advisor rationales by tier. There are three tier-1 Advisors. The first, VICTORY, comments if it does not sense an obstruction in

Table 1. Rationales that underlie SemaFORR 22 Advisors. There is one Advisor for each tier-1 rationale. Each tier-3 rationale is implemented by a move Advisor and a turn Advisor (with the exception of †, applicable to turns only).

Tier 1, in order	
VICTORY	Go to an unobstructed target.
AVOIDWALLS	Do not go within ϵ of a wall.
NOTOPPOSITE	Do not return to last heading.
Tier 3 heuristics vote	
	Based on commonsense reasoning
BIGSTEP	Take a long step or turn in the direction of a long step.
ELBOWROOM	Go where there is room to move.
EXPLORER	Go to unfamiliar locations.
GOAROUND†	Turn to avoid obstacles directly before you.
GREEDY	Go closer to the target.
	Based on learned spatial affordances
CONVEY	Go to frequent, distant conveyors.
ENTER	Go into a region via an exit.
EXIT	Leave a region via an exit.
TRAILER	Pursue a useful trail segment.
UNLIKELY	Do not enter a leaf region unless it contains the target.

its “line of sight” to the target. It mandates the move that will bring the robot closest to the target or the turn that will head the robot most directly toward it. The second, AVOIDWALLS opposes actions likely to bring the robot too close to a wall and thereby risk collision due to its noisy actuators. (“Likely” is within δ of a wall, given the wall register values.) Finally, NOTOPPOSITE vetoes turns that would simply restore the robot’s immediately preceding heading.

Thus, by design, tier 1 selects an action toward to an unobstructed target within range, and otherwise forwards to tier 3 only actions that avoid collisions with walls and do not oscillate in place. Recall that tier-1 Advisors are expected to be correct. Given the uncertainty inherent the robot’s actuators and its partial view of the world, little else can be safely asserted. The remainder of SemaFORR’s reasoning is necessarily heuristic.

3.5 Tier-3 Advisors

SemaFORR’s tier-3 Advisors have heuristic rationales. With the exception of GOAROUND, each tier-3 rationale gives rise to two Advisors: one for moves (name ends in M) and one for turns (name ends in T). A turn Advisor considers how its associated move Advisor would comment after each possible turn. A turn decision is not a classical robotics plan, however, because it makes no commitment to a subsequent move; it only anticipates one. For example, GREEDYM comments on moves with strengths that are inversely proportional to the distance they are expected to place the robot from the target. In tandem, GREEDYT calculates its comment strengths from how close the robot could come to the target if it were to turn and then make GREEDYM’s most preferred move.

Each tier-3 Advisor has a metric that assigns a real value to each possible action. To comment on a given set of n actions, an Advisor applies its metric, ranks the actions in descending order by their metric values, and then assigns corresponding *comment strengths* from n down to 1. The larger the strength, the more the Advisor prefers the action.

Four more tier-3 rationales also represent commonsense and rely only on local perception. BIGSTEP supports large actions, with comment strengths proportional to the action's size. ELBOWROOM supports actions that keep the robot further from walls. When the robot is facing a wall, GOAROUNDT supports turns that veer away from it, and prefers larger turns more strongly when the wall is closer. Finally, EXPLORER advocates exploration to reduce uncertainty, confirmed behavior for people in noisy, dynamic environments [32]. It supports actions toward locations that are relatively novel with respect to the current target (i.e., minimize the total Euclidean distance to previous decision points).

The remaining tier-3 rationales exploit learned spatial affordances. CONVEY supports actions to high-count conveyors, with preference for those further from the robot. When high-count conveyors are near one another, CONVEY thereby advocates travel through those locations rather than merely to them.

TRAILER is a case-based reasoning mechanism for trails. A trail is *accessible* if and only if a ray from the robot's current wall register intersects some line segment of the trail between two consecutive trail markers. Unless it has already done so, TRAILER identifies an accessible trail that has a marker m within sensory range of the target (as indicated by the wall register at m). If there is such a trail, TRAILER's comments greedily support actions toward trail markers further along the trail segment that leads to the target. TRAILERM's comment strengths reflect the ability of each move to get the robot farther along that trail on its way to the vicinity of the target. There is no plan-like commitment to a trail, however, because the other Advisors may draw the robot elsewhere. (Indeed, once the robot arrives in the immediate vicinity of the target, VICTORY will take control.) Furthermore, if TRAILER cannot sense any marker on its selected trail for four consecutive decision cycles, it is "lost," does not comment, and looks for a new trail on the next decision cycle.

Three Advisor rationales reference regions. A *leaf region* is defined as one whose exits all lie within an arc of no more than 90°. (With perfect knowledge, a leaf region would be a dead-end.) When the target lies in region T, the robot in region R, and R is adjacent to T, ENTER supports actions into T, UNLIKELY opposes actions into a leaf region other than T, and EXIT supports actions toward any exit from R if the target is not in R, in the spirit of [3].

Figure 3 provides a unifying example. It superimposed on the true map the robot, its current target, and the current spatial model, with leaf regions drawn lighter. At this point in the decision cycle, AVOIDWALLS has already vetoed the move of intensity 5 because actuator error could drive the robot into a wall. Because SemaFORR turned on its previous decision, only pause and moves with intensity 1, 2, 3, and 4 along the robot's current heading have been forwarded to the tier-3 Advisors.

Comments: <Advisor, move, strength>

BigStepM, pause, 1	TrailerM, pause ,1	ElbowRoomM, pause, 5
BigStepM, 1, 2	TrailerM, 1 ,2	ElbowRoomM, 1 ,4
BigStepM, 2, 3	TrailerM, 2, 4	ElbowRoomM, 2, 3
BigStepM, 3, 4	TrailerM, 3, 5	ElbowRoomM, 3, 2
BigStepM, 4, 5	TrailerM, 4, 3	ElbowRoomM, 4, 1
GreedyM, pause, 1	ConveyM, pause, 0	ExplorerM, pause, 0
GreedyM, 1, 2	ConveyM, 1, 1	ExplorerM, 1, 1
GreedyM, 2, 3	ConveyM, 2, 1	ExplorerM, 2, 2
GreedyM, 3, 4	ConveyM, 3, 2	ExplorerM, 3, 3
GreedyM, 4, 5	ConveyM, 4, 0	ExplorerM, 4, 4

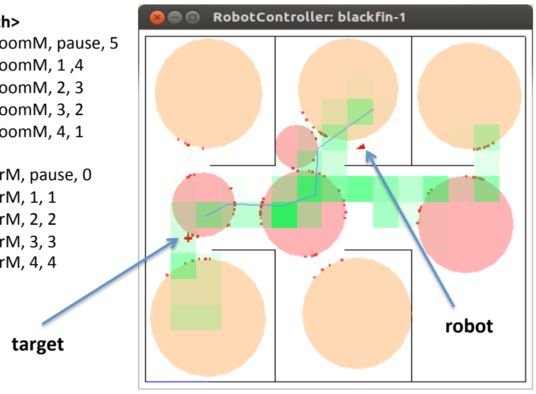


Fig. 3. A spatial model and the comments generated by the tier-3 Advisors when the robot is at the indicated decision point on its way to a target. For clarity, the walls (unknown to the robot) and only the trail selected by the Advisors are shown.

The comments in Figure 3 clearly reflect the rationales and the current decision point. GoAroundT will not comment, because no turns are currently available. BigStepM's comment strengths reflect the size of each move. Since the target is far from the robot's location, GreedyM comments on 4 with strength 5, and on the other moves with decreasing strengths. For clarity, Figure 3 shows only the selected trail that *Trailer* wants to pursue. ConveyM's comment strengths direct the robot to the darker grid cell that lies in front of it. ElbowRoomM's comments encourage moves to locations further from walls. By this time the robot had visited much of the room it was in, so ExplorerM's comment strengths seek to drive it out into the hallway, to less familiar territory. EnterM and UnlikelyM do not comment because the target is not within any region, and ExitM does not comment because the robot is not in any region. Because the vote in Figure 3 is tied among moves 2, 3, and 4, one of them will be selected at random.

4 Empirical Design

To examine how SemaFORR's knowledge and skill evolve in world W, the robot is given an agenda, a sequence of targets T to visit. In a *run*, the robot begins at the same position (in the lower left of W) and attempts to visit each target in the fixed but randomized order of T. Once the robot reaches a target, it addresses the next one on its agenda from its current position. If the robot does not reach a target after 250 decisions, it is deemed to have *failed* on that target, and addresses the next target in T from its current position. The evaluation criteria for a run are the *success rate* (percentage of reached targets in T), the total elapsed wall-clock time in seconds, and the total distance traveled in centimeters.

The pair $\langle W, T \rangle$ is called a *setting*. There are two sources of non-determinism in a run: actuator variance and random tie-breaking during voting. Thus, to gauge performance consistency within a given setting, results are averaged over 5 runs. To gauge performance consistency within a given world W, there are 5 randomly generated, 40-target sets T for each world W, and results are averaged over all settings. Thus our data describes navigation performance on 1000 targets (25 runs of 40 targets) in each world. To gauge performance consistency across different navigation challenges, our experiments investigate three worlds built for people but with different connectivities [28]. World A simulates an office space, world B a rotunda, and world C a warehouse or library stacks.

The experiments reported here compare SemaFORR with *SemaFORR-A**, a gold standard for robot planning. From a map of the world, SemaFORR-A* plans a shortest (A*) path to each new target t. This path is represented as a sequence of waypoints from the robot's initial location to t and avoids walls on the map. To make a decision, SemaFORR-A* selects the action intended to bring the robot closest to its next waypoint in the plan. Such navigation would be perfect were it not for actuator errors, which may move the robot to a position where a waypoint is obstructed or too far away. In that case, SemaFORR-A* must replan. To reduce the impact of actuator error and thereby help adhere to the plan, SemaFORR-A* selects only small (intensity-1) moves and turns. Comparison to SemaFORR-A* evaluates the impact of reactivity and a local, rather than a global, view.

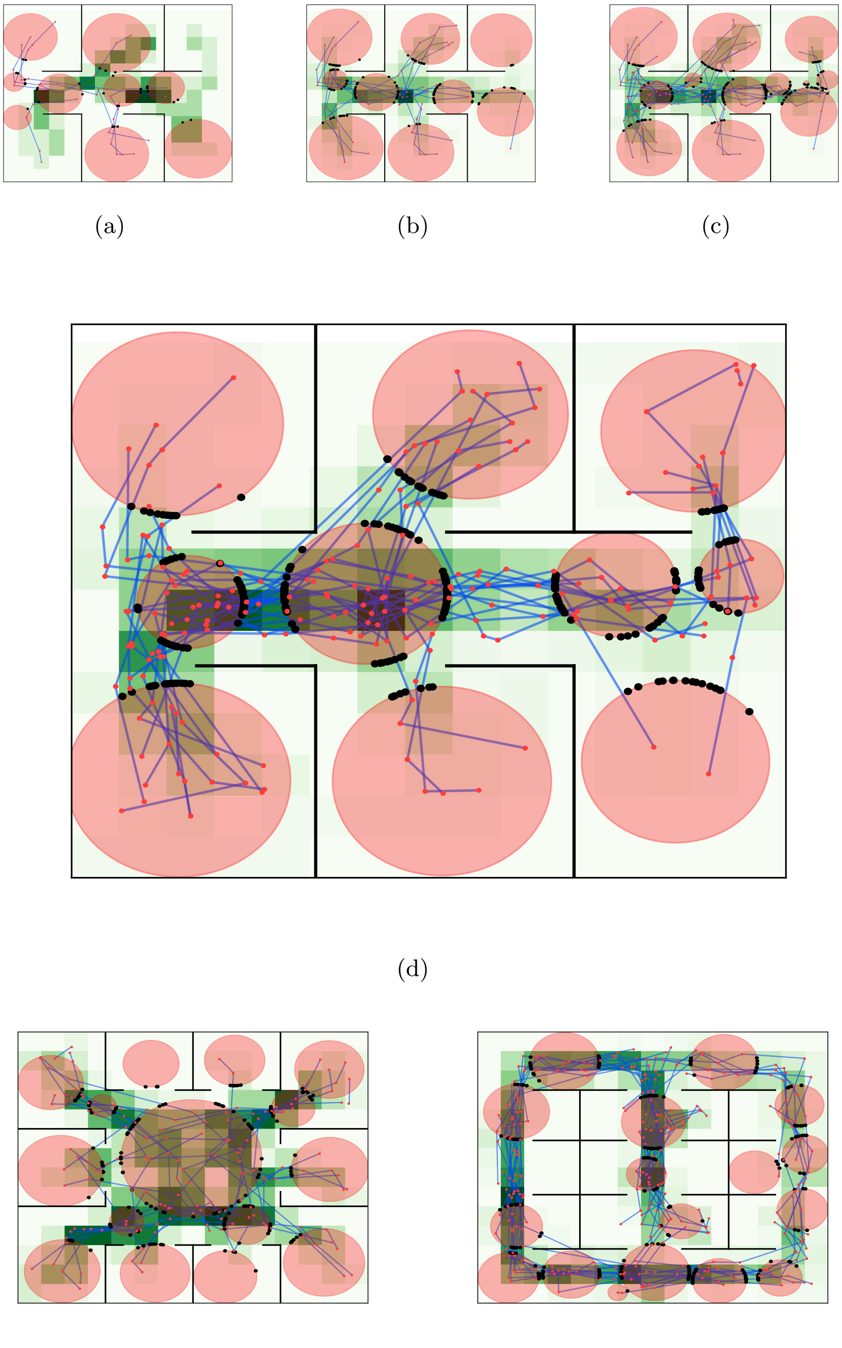
To tease apart SemaFORR's navigation skills, we also test five ablated versions of SemaFORR. *SemaFORR-B* navigates with only commonsense qualitative spatial reasoning, as represented by tier 1 plus four qualitative commonsense rationales in tier 3: BigStep, ElbowRoom, GoAround, and Greedy. SemaFORR-B has no spatial affordances; comparison to it evaluates the impact of commonsense spatial reasoning. To gauge the impact of exploration, *SemaFORR-E* augments SeamFORR-B with Explorer. To evaluate the impact of the individual spatial affordances, *SemaFORR-C, SemaFORR-R,* and *SemaFORR-T*, each add a single spatial affordance (conveyors, regions, or trails, respectively) to SemaFORR-E, along with their associated Advisors.

5 Results

5.1 Learned spatial models

A qualitative evaluation metric is the appropriateness of SemaFORR's learned spatial models. Figures 4(a)- 4(d) show how a spatial model for world A evolved during a single run. Note how they develop quickly, with few changes after the first 10 targets. Videos of the robot's travel show how a model evolves after each target despite actuator error: `http://youtu.be/3C_675H6-xk`, `https://youtu.be/4WF8unQlSm8`.

Figures 4(d)- 4(f) show, for each world, the spatial model learned after a single run, overlaid on the walls of its true map. Inspection indicates that these



(a) (b) (c) (d) (e) (f)

Fig. 4. Spatial models in SemaFORR, overlaid on the corresponding true map, which the robot does not have. Conveyors are shown as grid cells (darker is more salient), and regions as circles with dotted exits. Trails are line segments with dots at the trail markers. The spatial model for world A as it evolves after (a) 10 (b) 20 and (c) 30 targets. After 40 targets, SemaFORR's learned spatial models of (d) world A, (e) world B, and (f) world C.

Table 2. Navigation performance of SemaFORR-A* (an ideal planner), SemaFORR, and five ablated versions of it, measured by time in seconds, distance, and success (percentage of reached targets per run).

	World A			World B			World C		
Navigator	Time	Dist.	Succ.	Time	Dist.	Succ.	Time	Dist.	Succ.
SemaFORR-A*	1035.9	400.1	100.0	884.6	335.1	100.0	1119.9	437.9	100.0
SemaFORR-B	1823.7	974.5	94.1	1124.6	565.0	97.8	2473.5	887.5	82.2
. . . SemaFORR-E	1415.1	1018.8	99.4	1090.2	730.9	99.2	1497.7	983.2	98.5
SemaFORR-C	1323.8	977.4	99.8	1009.8	698.7	99.2	1303.2	933.2	99.7
SemaFORR-R	1280.1	892.2	99.6	941.5	612.6	99.6	1524.7	919.7	98.2
SemaFORR-T	1163.8	813.0	99.5	867.2	553.3	99.8	1278.0	775.7	99.6
SemaFORR	1221.2	854.5	99.5	835.3	554.9	99.7	1275.7	798.2	99.8

final models varied little over the five runs for one setting. They also varied little from one set of targets T to another in the same world. Observe how the regions capture the "rooms" in worlds A and B, but only two of the cubicles in world C; targets in that particular world-C setting appeared in cubicles less often. Note, too, how the conveyors develop a "highway" for the hallway in world A, diagonal "highways" for world B, and perimeter and central "highways" for world C.

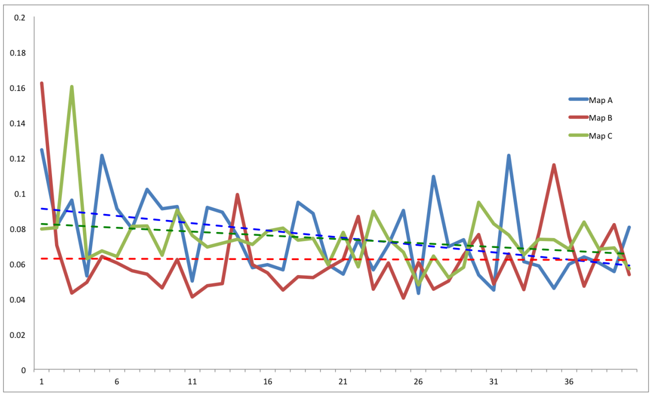
5.2 Performance

Performance results appear in Table 2. For all our navigators, the target sets in world B are clearly the easiest, and those in world C the most difficult. In the following discussion, high variance caused both by actuator error and randomized target sets T makes some apparent differences inconclusive; differences cited here are at $p < 0.05$.

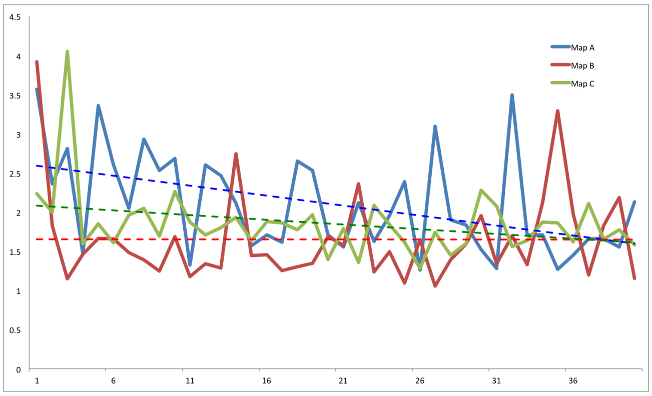
Without a map and given its penchant for exploration, SemaFORR should not be expected to match SemaFORR-A*'s distance along its optimal paths in a complete map. Nonetheless, in world B SemaFORR reaches its targets just as fast as SemaFORR-A*. In worlds A and C, SemaFORR travels further but is only slightly slower than SemaFORR-A* (18% and 14% slower, respectively).

Both SemaFORR-A* and SemaFORR spend most of their time in travel rather than decision making. SemaFORR-A* devotes about 19% of its time to decisions in all 3 worlds. SemaFORR devotes 17% of its time to decisions in worlds A and B, and 18% in C. Moreover, in every world, SemaFORR's learning requires less than 0.01% of the elapsed time.

Compared to SemaFORR-B, SemaFORR improves navigation: time is substantially reduced, distance decreases in worlds A and C, and reliability (as measured by success rate) rises. SemaFORR-E demonstrates the improvement exploration brings to the commonsense reasoning of SemaFORR-B, and the price the robot pays for it. Travel time is reduced by 22%, 3%, and 39%, in worlds A, B, and C, respectively, and the success rate rises, particularly in world C. Exploration also increases travel distance, however, by 5%, 29%, and 11%, respectively. Moreover, as one would expect, SemaFORR-E does not demonstrate



(a)



(b)

Fig. 5. For each world, plotted across 40 targets and averaged over all runs, the ratio of SemaFORR's (a) time and (b) distance to the A* distance to each target. Dashed regression lines with negative slopes indicate improved performance across time in worlds A and C.

any improvement across targets; it fails as late as on the 32nd target in world A, the 36th in world B, and the 29th in world C. Compared to exploration plus commonsense, each spatial affordance alone reduces travel time (except for regions in world C) and distance, with improvements in every success rate for all but SemaFORR-C in world B. The trails (SemaFORR-T) are particularly effective; they can reach the targets as quickly in world B as SemaFORR-A* does, presumably because they allow longer, highly effective steps.

SemaFORR's superiority to SemaFORR-B demonstrates that learning spatial affordances is an important component in SemaFORR's performance. There are two ways to gauge if that learning is online, that is, if SemaFORR's performance improves across a sequence of targets. First, with more experience, SemaFORR should fail on targets less often. While SemaFORR-B has runs in which it fails as late as on the 40th (final) target, SemaFORR's last failure on any setting and run in map A was on the 19th and in map C on the 23rd. (There was one failure in world B on the 35th, probably brought on by actuator error because higher-intensity moves are more frequently possible in B.) Second, SemaFORR should reach its targets faster. Because random generation makes some targets intrinsically more difficult to reach, we normalize performance by how hard it is to reach a target, estimated here by the distance in an A* plan. (This ignores the need to turn correctly and differs from SemaFORR-A*'s distances, which are subject to actuator error.) Figure 5 shows that the regression trend lines for this ratio descend across 40 tasks for worlds A and C for both time and distance. In other words, with experience SemaFORR learns to reach its targets more quickly and in a shorter distance.

6 Discussion

SemaFORR is not envisioned as a replacement for SLAM, but as a companion to it, one that facilitates robot-human interaction. A cognitively plausible mental model like SemaFORR's can be shared with a person at a level of abstraction that is both meaningful and parsimonious. "I'm at the center of the biggest region" are considerably more informative for people than "My pose is $< 10, 20, 39° >$."

When, as in HRTeam, a person collaborates with an agent, the ability to explain the agent's reasoning in a human-friendly manner is a first step toward transparent, more natural communication and the establishment of trust. Instead of a metric map, SemaFORR relies on commonsense reasons that support low-level actions. As a result, a SemaFORR robot can explain any of its reasons. A comment from Explorer, for example, can be paraphrased as "Let's go this way because we haven't explored it much yet," and one from Enter as "I want to go into this dead-end [leaf region] because it contains the target."

For a decision made in tier 3, SemaFORR has a mechanism to translate it into natural language that reflects both the rationales of the Advisors whose comments supported it and the degree of their preference (i.e., comment strength). For example, a vote where the strength of ElbowRoom's comment is considerably higher than that of any other comment becomes "I really want to move into open space." In contrast, a vote in which the comments from BigStep, Explorer, and Greedy dominate but with somewhat lower strengths than in the previous example becomes "It seems reasonable to move as far as possible, into a new area, and toward our target." Furthermore, because of its modularity, SemaFORR can readily incorporate new explanations as a spatial model becomes more elaborate. Recent work on the construction of a depiction from a verbal description (e.g., [37]) and the negotiation of a route between a robot and person (e.g., [39]) could also be applicable here. The remainder of this section examines the components that drive SemaFORR, its real-world applicability, and our current work.

6.1 What Makes SemaFORR Work

SemaFORR uses its sensors' values to compute and store a simplistic model of its spatial environment. This model is dynamic; it consists of affordances (regions, trails, and conveyors) that proliferate and change as the robot experiences new parts of the environment. Because the model is based on heuristic learning algorithms that analyze only the robot's local views and actual travel, it is necessarily a set of approximations. The interdependence among affordances (e.g., the presence of some high-scoring conveyors along a trail) is deliberate; it both mediates disagreement between possibly disparate approximations and enriches their usefulness. SemaFORR's spatial representation provides flexibility and efficiency, and provides a human-like basis for strategy formation [34].

SemaFORR's architecture exploits the synergy among naïve commonsense reasons. For example, in a mix of heuristics that argue with various intensities for or against a set of moves, GREEDY is just one among many reasons to move down a hallway toward the right room. Qualitative reasoning also allows the robot to correct for, and even anticipate, the inaccuracies of its actuators. For example, when a large move incurs a large actuator error that draws the robot off its intended trail and into an irrelevant room in world A, EXPLORER and EXIT soon pull it back out again.

SemaFORR's model of the robot is realistic. As the floor surface changes or as batteries drain, it is reasonable to assume that a real-world robot's position after it executes a decision may not be precisely what it anticipated. This motivated the replication of laboratory-observed actuator error during simulation. Advisors' comment strengths also deliberately smooth data already recognized as approximate. SemaFORR discretizes movement in continuous space into a sequence of frequent decisions. It chooses an action at least once per second and as often as 3 times per second, depending upon the length of the intervening moves. The result appears To the human observer as smooth navigation.

SemaFORR's learning is pragmatic; it only infers conveyors, trails, and exits from regions when it manages to reach its target. (Regions, however, are simple local observations, learned either when the robot arrives at its current target, or it reaches its decision limit.) Moreover, SemaFORR's learning algorithms are heuristically honed for fast computation and retrieval. Thus, the resultant spatial model is necessarily an approximation and not a map. The robot represents only what it experiences. If a setting does not take the robot to an area where it can capture a local view, the model will not include that area. In the sensor placement described here, there is also a bias toward the robot's heading, which collects more information about what is in front of the robot. Nonetheless, learning during navigation supports flexibility and gradual improvement. Reinforcement learning that relies on an abstract map [21] is somewhat similar, but SemaFORR extracts and labels its own training examples heuristically, from its experience.

SemaFORR is similar to robotics work both in subsumption and in potential fields. Tier 1 is analogous to a subsumption architecture, where rules are carefully engineered and ordered. (The robot vacuum cleaner Roomba™, for example, has a subsumption architecture.) Tier 1, however, has only 3 Advisors, and they make only about 24.13%, 37.09%, and 22.75% of all decisions in worlds A, B, and C, respectively; the heuristic Advisors in tier 3 make all the others. SemaFORR's tier 3 is analogous to a potential field, where forces attract or repel the robot with vectors analogous to Advisors' comment strengths. Potential fields, however, are vulnerable to local minima. SemaFORR avoids local minima through two mechanisms: exploration and randomization. Exploration draws it to locations less recently visited; voting ties broken at random provide enough non-determinism to extricate it from repetitive behavior.

Robotics has traditionally relied on precision planning from a global view; if the robot has no map it immediately tries to construct one. People in a complex environment, however, lack the working memory to construct an A* plan. Instead, they satisfice with a spatial model, commonsense qualitative reasoning, and the ability to learn. SemaFORR tests the extent to which such behavior supports navigation. People can, however, concatenate previously successful routes to construct a new one. While SemaFORR could similarly piece together trail segments, computation over extensive stored experience would soon become costly. Instead, SemaFORR's tier-3 Advisors foreshadow some of the approaches anticipated for tier 2, which is currently under development. A planner could, for example, support a turn decision followed immediately by the move that made it attractive, or it could follow a trail more closely.

6.2 Transition to the Real World

Although the results reported here are for simulation, Player/Stage simplifies the transition to real-world execution. SemaFORR-A* controls real-world Blackfins in a laboratory whose walls are replicated in world A. Indeed, the values used here for how close the robot can come to a wall and how close it must come to the target were gleaned from the metrics already used in the lab. (The same values were also applied, unchanged, to worlds B and C.) An earlier study demonstrated that some performance metrics gathered in HRTeam's simulation are good predictors of behavior in the physical world [31]. Current work includes on-the-floor experiments to confirm that transfer from simulation to the physical world remains effective. Meanwhile, we continue to hone SemaFORR in simulation, where we can run online experiments quickly.

Both the descriptives and the Advisors were developed in world A, but are sufficiently generic to apply to worlds B and C as well. To test the extent of that generality, and to see how SemaFORR's approach scales, we have reproduced for simulation a considerably more elaborate environment, a wing from the floor of a large building that includes one of our labs. This is a realistic built space, about 40 times the area of world A, with 2 hallways and about 7 times as many rooms. It is also considerably more complex; there are rooms with multiple doors and rooms that are accessible only through a sequence of other rooms (i.e., not directly off a hallway). The complexity of this world necessitates more targets and a larger decision limit before failure. Here we used 50 randomly-chosen targets and a 400 decision-cycle limit; all other settings and algorithms remained the same.

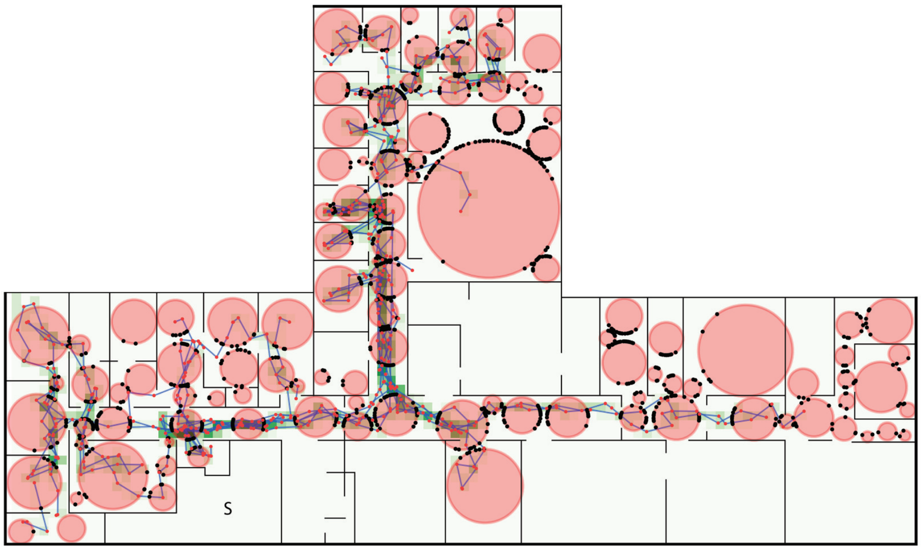


Fig. 6. What SemaFORR has learned after 50 targets in a challenging real-world environment. The robot enters at the stairwell, marked with an S.

The ease with which SemaFORR scales is visible in Figure 6, shown after a single run. Figure 6 makes clear that the model only includes areas in which the robot travels; more points would likely be necessary to cover this world and to refine the regions that appear to cross a wall. (Regions typically reduce with more travel through them.) It also shows some opportunities for further improvements in SemaFORR. It is clear that for less square rooms it may be worthwhile to merge regions in some way. Moreover, as the complexity of the space increases, the likelihood that any trail matches a new target declines significantly. One way to address this issue would be to combine subsequences of trails at runtime.

We expect SemaFORR's reactive approach to support a variety of other behaviors, some of which go unaddressed by modern, plan-based navigation. In particular, an agent with an agenda need not consider it in a prescribed order. A brief detour to address a target on the agenda but not the current focus of attention is an obvious extension to SemaFORR. Moreover, it should be possible to transfer knowledge between similar environments, such as floors in some office buildings. In addition, when an individual robot needs repair or recharging, a clone given the acquired knowledge is a near-seamless replacement (subject to its idiosyncratic actuator and sensor noise). Finally, how often a robot must expend energy to sense is an open question. Because SemaFORR senses only between actions, a variety of tested intensities could provide a preliminary answer.

Current work capitalizes on SemaFORR's modularity to support its gradual development. One current research avenue is the use of a team of robots that addresses a setting simultaneously, with targets assigned to particular robots. Each robot remains autonomous, with its own copy of SemaFORR, but all robots share in the construction and use of the same spatial model, stored on the DM. We are now testing rationales (analogous to AvoidWalls, ElbowRoom, and BigStep) to avoid robot-robot collisions and crowding.

SemaFORR's modularity includes the ability to support robots on different platforms (e.g., a Blackfin and a Turtlebot^{TM}) with different maneuverability and different footprints. Features that may appear platform-specific here are actually modular and readily replaced without hand tuning. For example, the number of discrete commands was intended for the Blackfin, but a slightly larger set of intensities should pose no difficulty for the architecture. (The maximum-intensity command actually reflects the furthest one would want the robot to move or turn without sensing again. That is learnable as a function of the world and the robot platform.) How close the robot must come to the target to be successful is a function of the robot's footprint . Different sensors (e.g., a Kinect or 20 equally-spaced infrared units along the robot's perimeter) could be readily accommodated. Indeed, better sensors should provide considerably more accurate local views that could further improve performance. Thus, we believe that a SemaFORR-supported heterogeneous multi-robot team is a tractable next step.

Meanwhile, SemaFORR quickly learns features of an environment that facilitate effective autonomous navigation without costly mapping or planning. That knowledge transfers from one task to another. When a SemaFORR robot travels, it moves around obstacles and toward its target, with big steps where its world permits. It also anticipates access within regions, uses markers from old trails, turns around obstacles, explores new locations, and recovers from its own errors. Remarkably, that suffices to reach targets in these environments, and quickly builds a simple spatial model of the world that facilitates explainable, human-friendly, effective navigation.

Acknowledgments. This work work was supported in part by the National Science Foundation under IIS 11-17000, IIS 11-16843 and CNS 08-51901, and by The City University of New York, Collaborative Incentive Research Grant 1642. We thank Tuna Ozgelen, Smiljana Petrovic, Eric Schneider, and our anonymous reviewers for their insights and guidance. Tereza Shterenberg produced the map in Figure 6.

References

1. Arkin, R.C.: Integrating Behavioral, Perceptual , and World Knowledge in Reactive Navigation. Robotics and Autonomous Systems 6 105-122 (1990)
2. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping: part II. IEEE Robotics & Automation Magazine 13 (3) 108-117 (2006)
3. Björnsson, Y., Halldórsson, K.: Improved Heuristics for Optimal Path-finding on Game Maps. In: Laird, J., Schaeffer, J. (eds.) AIIDE 2006, pp.9-14. AAAI (2006)
4. Botea, A., Harabor, D.: Path Planning with Compressed All-Pairs Shortest Paths Data. In: Borrajo, D., Fratini, S., Kambhampati, S., Oddi, A. (eds.) ICAPS-2013, pp. 288-292. AAAI (2013)

5. Brooks, R.A.: Intelligence without Representation. Artificial Intelligence 47 (1-3) 139-160 (1991)
6. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine 13 (2) 99-110 (2006)
7. Epstein, S.L.: For the Right Reasons: The FORR Architecture for Learning in a Skill Domain. Cognitive Science 18 (3), 479-511 (1994)
8. Epstein, S.L. Spatial Representation for Pragmatic Navigation. In: Hirtle, S.C., Frank, A.U. (eds.) COSIT '97, pp.373-388. Springer Verlag (1997)
9. Epstein, S.L.: Pragmatic Navigation: Reactivity, Heuristics, and Search. Artificial Intelligence 100 (1-2) 275-322 (1998)
10. Epstein, S., Schneider, E., Ozgelen, A.T., Munoz, J.P., Costantino, M., Sklar, E.I., Parsons, S.: Applying FORR to human/multi-robot teams. In: Human-Agent-Robot Teamwork Workshop at 7th ACM/IEEE International Conference on Human-Robot Interaction (2012)
11. Frommberger, L., Wolter, D.: Spatial Abstraction: Aspectualization, Coarsening, and Conceptual Classification. In: Freksa, C., Newcombe, N. S., Gårdenfors, P., Wölfl, S. (eds) Spatial Cognition VI, LNAI 5248, 311-327 (2008)
12. Gat, E.: On Three-Layer Architectures. In: Kortenkamp, D., Bonnasso, R. P., Murphy, R. (eds.) Artificial Intelligence and Mobile Robots. AAAI Press (1998)
13. Gerkey, B., Vaughan, R., Howard, A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: The International Conference on Advanced Robotics pp. 317-323 (2003)
14. Gibson, J.J.: The theory of affordances. In: Shaw, R., Bransford, J. (eds.) Perceiving, Acting, and Knowing: Toward an Ecological Psychology, pp. 6782. Lawrence Erlbaum, Mahwah (1977)
15. Golledge, R. G.: Path Selection and Route Preference in Human Navigation: A Progress Report. In: Frank, A.U., Kuhn, W. (eds.) COSIT 95, 207-222. Springer, Berlin (1995)
16. Golledge, R, G.: Human Wayfinding and Cognitive Maps. In: Golledge, R, G. (ed.) Wayfinding Behavior. pp.5-45. Hopkins University Press, Baltimore (1999)
17. Hamburger, K., Dienelt, L. E., Strickrodt, M., Röser, F. Spatial cognition: the return path. In: Knauff, M., Pauen, M., Sebanz, N., Wachsmuth, I. (eds.) Cognitive Science 2013, pp. 537-542 (2013)
18. Hart, P. E., Nilsson, N. J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2), 100-107 (1968)
19. Hölscher, C., Tenbrink, T., Wiener, J. M.: Would you follow your own route description? Cognitive strategies in urban route planning. Cognition 121, 228-247 (2011)
20. Jonietz, D., Timpf, S.: 2013. An Affordance-Based Simulation Framework for Assessing Spatial Suitability. In: Tenbrink, T., Stell, J., Galton, A., Wood, Z. (eds.) COSIT 2013. LNCS, vol. 8116, pp. 180-184 Heidelberg, Springer (2013)
21. Joshi, S., Schermerhorn, P., Khardon, R., Scheutz, M.: Abstract planning for reactive robots. In: Parker, L.E, (ed.) ICRA 2012. IEEE pp. 4379-4384 (2012)
22. Koenig, S., Likhachev, M., Furcy, D.: Lifelong planning A*. Artificial Intelligence 155 (1), 93-146 (2004)
23. Kostavelis, I., Gasteratos, A.: Semantic mapping for mobile robotics tasks: A survey. Robotics and Autonomous Systems 66: 86-103 (2015)
24. Mataric, M.: Integration of representation into goal-driven behavior-based robots. IEEE Transactions on Robotics and Automation 8 (3), 304-312 (1992)

25. Meilinger, T.: The Network of Reference Frames Theory: A Synthesis of Graphs and Cognitive Maps. In: Freksa, C., Newcombe, N. S., Gårdenfors, P., Wölfl, S. (eds) Spatial Cognition VI, LNAI 5248, 344-360 (2008)
26. Montello, D.R., Sas, C.: Human Factors of Wayfinding in Navigation. In: Karwowski, W. (ed.) International Encyclopedia of Ergonomics and Human Factors, 2nd edition, pp.2003-2008 CRC Press. (2006)
27. Mozos, Ó. M., Triebel, R., Jensfelt, P., Rottmann, A., Burgard, W.: Supervised semantic labeling of places using information extracted from sensor data. Robotics and Autonomous Systems 55: 391-402 (2007)
28. Ozgelen, A. T., Sklar, E. I.: Toward a Human-Centric Task Complexity Model for Interaction with Multi-Robot Teams. In: Workshop on Human-Agent Interaction Design and Models (HAIDM) at AAMAS-2014 (2014)
29. Ratterman, M.J., Epstein, S.L.: Skilled like a Person: A Comparison of Human and Computer Game Playing. In: Moore, J.D., Lehman, J.F. (eds.) Cognitive Science 95. Lawrence Erlbaum Associates, pp. 709-714 (1995)
30. Reineking, T., Kohlhagen, C., Zetsche, C.: Efficient Wayfinding in Hierarchically Regionalized Spatial Environments. In: Freksa, C., Newcombe, N. S., Grdenfors, P., Wölfl, S. (eds.) Spatial Cognition VI. Freiburg, Germany: 56-70 (2008)
31. Sklar, E. I., Ozgelen, A. T., Munoz, J. P., Gonzalea, J., Manashirov, M., Epstein, S.L., Parsons, S.: Designing the HRTeam Framework: Lessons Learned from a Rough-and-Ready Human/Multi-Robot Team. In: Autonomous Robots and Multirobot Systems workshop (2011)
32. Speekenbrink, M., Konstantinidis, E.: Uncertainty and exploration in a restless bandit task. In: Bello, P., Gaurini, M., McShane, M., Scassellati, B. (eds.) Cognitive Science 2014, pp. 1491 - 1496 (2014)
33. Takemiya, M., Ishikawa, T.: Strategy-Based Dynamic Real-Time Route Prediction. In: Tenbrink, T., Stell, J., Galton, A., Wood, Z. (eds.) COSIT 2013. LNCS, vol. 8116, pp. 149-168 Heidelberg, Springer (2013)
34. Tenbrink, T., Bergmann, E., Konieczny L.: Wayfinding and description strategies in an unfamiliar complex building. In: 33rd Annual Conference of the Cognitive Science Society, pp. 1262-1267 (2011)
35. Thrun, S., Bücken, A., Burgard, W., Fox, D., Fröhlinghaus, T., Hennig, D., Hofman, T., Krell, M., Schmidt, T.: Map Learning and High-Speed Navigation in RHINO. In: Kortenkamp, D., Bonasso, R.P., Murphy, R. (eds.) AI-based Mobile Robots: Case Studies of Successful Robot Systems, pp. 21-52. MIT Press. (1998)
36. Tversky, B.: Cognitive Maps, Cognitive Collages, and Spatial Mental Models. In: Frank, A.U., Campari, I. (eds.) Spatial information theory: A theoretical basis for GIS. LNCS, vol. 716, pp.14-24. Springer-Verlag, Berlin (1993)
37. Vasardani, M., Timpf, S. , Winter, S., Tomko M.: From Descriptions to Depictions: A Conceptual Framework. In: Tenbrink, T., Stell, J., Galton, A., Wood, Z. (eds.) COSIT 2013. LNCS, vol. 8116, pp. 299-319 Heidelberg, Springer (2013)
38. Vasudevan, S., Gächter, S., Nguyen, V., Siegwart, R.: Cognitive maps for mobile robots —an object based approach. Robotics and Autonomous Systems 55(5): 359-371 (2007)
39. Weiser, P., Frank, A. U.: Cognitive Transactions - A Communication Model. In: Tenbrink, T., Stell, J., Galton, A., Wood, Z. (eds.) COSIT 2013. LNCS, vol. 8116, pp. 129-148 Heidelberg, Springer (2013)
40. Wolf, D. F., Sukhatme, G. S.: Semantic Mapping Using Mobile Robots. IEEE Transactions on Robotics 24(2): 245-258 (2008)
41. Zetzsche, C., Galbraith, C., Wolter, J., Schill, K.: Representation of Space: Image-like or Sensorimotor? Spatial Vision. 22 (5) 409-424 (2009)