In *Parallel Problem Solving from Nature*, Hans-Paul Schwefel and Reinhard Maenner, eds., Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1991, 381-38 .

Parallel, Decentralized Spatial Mapping for Robot Navigation and Path Planning

Maja J Mataric

maja@ai.mit.edu

MIT Artificial Intelligence Lab
545 Technology Square #721
Cambridge, MA 02139, USA

Abstract

We present a distributed model for mobile robot spatial mapping and path planning. The method was implemented and tested on a sonar and compass-based physical mobile robot controlled by three competence layers:

- Low-level navigation: a collection of reflex-like rules whose combination results in emergent collision-free edge-following.
- Landmark detection: using the underlying reflexive navigation, the robot dynamically extracts procedurally-defined landmarks from the environment.
- Spatial Mapping and Path Planning: the landmarks are used to construct a distributed topological map of the environment. The locations in the map are individual, independently acting processes. This implementation allows for localization in constant time. *Spreading of activation* is used to compute both topological and physical shortest paths in linear time.

1 Introduction

Navigation and spatial modeling in biological systems have been studied extensively by biologists and psychologists. Even insects and simple animals have been shown to employ internal spatial representations for navigation tasks [**Waterman 89**] [**Gallistel 89**] [**Gould 82**]. The nature of those representations still remains a puzzle, and issues, such as topological versus metric models, and centralized versus distributed maps, continue to be debated. Although it is known that the spatial representations are distributed on the neuronal level, the mapping between physical and neuronal space is not known.

In the field of Artificial Intelligence and Mobile Robotics, the issues of spatial mapping and path planning lie at the heart of the representation problem. Although the field has seen an increase in the use of parallel computational hardware in the past several years, the algorithms employed so far have all utilized centralized representations ([**Kuipers and Byun 88**],[**Chatila and Laumond 85**], [**Moravec and Elfes 85**], etc.).

Inspired by the success of parallel algorithms in nature, we have implemented a mobile robot which uses a parallel, decentralized spatial model for environment mapping and path planning.

2 The Implementation

In nature, many examples of specialized hardware and computation are found manifesting simple and elegant solutions to what appear to be complex problems [**Wehner 87**]. We have drawn from those examples in designing the simplest yet fully functional solution to the problem of spatial mapping and path planning. Our algorithm was designed to be general and independent of a specific implementation. However, its validity and success were tested on a specific physical robot in a variety of unstructured environments.

The robot, Toto, consists of an omnidirectional circular base supplied with a ring of twelve ultrasonic ranging sensors and a compass. It is fully autonomous with on-board power and processing. Toto was programmed in the Behavior Language, based on the subsumption architecture [**Brooks 90**]. The Behavior Language is a real-time parallel robot programming language [**Brooks 87**]. It provides abstractions for programming collections of behaviors executed in parallel. In fact, the software of the robot consists entirely of short, simple rules which receive inputs from the sensors on the robot, and send outputs to other rules or to the robot's actuators. This results in a tight coupling between sensing and action allowing the robot to act in real-time. Additionally, the fact that all of the rules are active in parallel and without centralized reasoning engine forces the control strategy to be fully decentralized.

The robot's goal is to wander around its environment, explore it, and map its topological structure. While exploring, it continually updates its representation, and uses it to plan and execute paths to any previously visited landmarks. Toto's low-level navigation consists of a collection of simple reflex-like behaviors. The behaviors consist of rules which, when acting in parallel, result in an emergent collision-free edge-following [**Mataric 90b**]. This approach is an alternative to the classical, sequential control structures usually employed in mobile robots. It is simple, tractable, and has been shown to perform reliably in unstructured office environments [**Mataric and Brooks 90**].

The process of landmark detection is executed continuously, in parallel with the rest of system's activities. Landmarks are chosen to be large, permanent structures the organization of which represents the topology of the space to be mapped. Additionally, they are selected for their permanence and reliability of detection with the given sensors and the underlying navigation algorithm. In the office environment they include walls, corridors, and irregular boundaries.

Analogous to the exploration method, landmark detection is also designed as a simple behavior. The robot monitors its own motion, and deduces, from its proprioceptors as well as the external sensors (range and compass), if it is following a landmark (a wall or a corridor) or a lack of one (an irregular boundary). The landmarks are defined and detected procedurally, in parallel with, and aided by the navigation behavior. Consequently, there is no need for a central controller making decisions as to which actions the robot should take and where it should direct its attention.

3 Decentralized Spatial Mapping

The detected landmarks are dynamically assembled into a topological representation of the explored space through the construction of a landmark network. Each landmark represents a location and is allocated a node in the network. Adjacency in physical space is indicated by nearest-neighbor links in the network. The structure of the resulting network is thus isomorphic to the topological organization of the detected landmarks in physical space.

Instead of implementing the network as a centralized data structure, we distributed it over many processes executing asynchronously, in parallel. To simplify the mapping we allocated a unique behavior (or process) to each landmark in the network. Like biological neurons, the processes are concurrently acting behaviors: all receive sensor and landmark inputs from the navigation and landmark detection behaviors. In addition to those broadcast messages, the network behaviors communicate by sending and receiving messages from their nearest neighbors via the topological neighbor links which serve as message wires.

The robot's location in the world corresponds to a landmark in the network. The node representing that landmark is *active*, and sends lateral inhibition to its neighbors which propagate it throughout the network, thus insuring uniqueness of localization. Making sure that only a single location is active requires occasional disambiguation between similar landmarks. This is done by spreading *expectation* to the neighbor of the active node in the robot's direction of travel [**Mataric 90a**]. Expectation preserves context, thus distinguishing different sequences of locations.

By utilizing the decentralized, parallel implementation, we distribute the representation of the map in space rather than time. This is particularly intuitive since we can take advantage of the isomorphism between the network and physical space. However, the method generalizes to other levels of abstraction.

One of the clear advantages of using a parallel approach is in fast localization. The robot finds out where it is by comparing the detected landmark to all known locations in parallel. Since expectation is propagated dynamically, it is a part of the location descriptor at the time of landmark matching. Consequently, the robot localizes in constant time $O(C)$ regardless of the size of the network. This is an improvement over the $O(n)$ running time of a serial implementation. Since localization must be performed often, this improvement is significant. Additionally, it scales well to large maps, both in physical size and the number of mapped landmarks.

4 Decentralized Path Planning

The purpose of the map is to allow the robot to return to arbitrary known locations. Given the fully distributed nature of the map, the path planning algorithm must rely on a local rather than global strategy. We use an adaptation of *spreading of activation* [**Quillian 69**] which is equivalent to parallel breadth first graph search. The goal location is selected by the user as any previously visited landmark. The network location corresponding to the goal starts the search by passing a *calling* message to all of its immediate neighbors. The message is propagated from neighbor to neighbor throughout the graph. A simple rule eliminates searching through cycles [**Mataric 90a**]. The search is terminated when the message reaches the active landmark, i.e. the one corresponding to the robot's position in physical space.

The control of the robot is stateless in that it performs its reflexive low-level navigation behavior regardless of whether it is exploring or heading toward a goal. In the latter case, as it encounters landmarks, it also receives a call message, which tells it which direction to take if at a decision point. Consequently, wherever in the network the robot is, it knows what to do: it takes the direction of the calling message. An entirely local decision takes it on the shortest path to the goal.

An additional advantage of the locality of the approach is the minimization of replanning. If the robot becomes lost or is unexpectedly moved to another location, it continues to navigate reflexively. As soon as it localizes it is able to continue pursuing the shortest path to the goal from its new location, without the need to replan.

Finding the shortest topological path within the network is performed in worst case linear time $O(n)$ in the number of landmarks in the network. This is an improvement over the $O(n^2)$ running time of serial shortest path search-based algorithms. In order to optimize the generated paths based on physical rather than topological distance, we added a length estimate to each landmark. As the parallel search is executed, the traversed distance is accumulated by adding the length of each traversed landmark. These landmark lengths serve as weights in the search. The resulting algorithm is a greedy gradient descent on landmark length. By taking the locally shortest path at each point, the robot follows the physically shortest path withing the known path space [**Mataric 90a**].

5 Conclusions

Our algorithm is geared toward fine-grained parallelism and would function best implemented on a collection of simple processors. Toto was programmed in a real-time parallel programming language executed on a single processor for the purposes of economy and hardware simplicity. The algorithm is fully portable to parallel hardware.

The parallel, decentralized implementation suggests the use of powerful local techniques which minimize computation overhead. The distribution of the control structure over concurrently acting processes results in more than just the increased speed of execution. It effectively spreads out the control architecture by completely removing a centralized reasoner. The resulting system is much more robust as it does not depend too strongly on any particular part of the behavior network. These results point to the importance of exploiting parallelism and decentralization in the choice of representation for real-time reasoning systems. Biological evidence supports these claims [**Arbib and Lieblich 70**][**Mataric 90c**].

The algorithm was successfully tested in a number of trials in unaltered office spaces with static and dynamic obstacles. The resulting data is reported on in [**Mataric 90a**] and [**Mataric and Brooks 90**]. The results demonstrate the feasibility and the advantages of employing parallelism on different levels of robot control.

Acknowledgements

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the research was provided in part by the Artificial Intelligence Center of Hughes Research Labs, and in part by the University Research Initiative under Office of Naval Research contract N0014-86-K-0685. Rodney Brooks originally proposed the idea of a distributed representation, and provided helpful suggestions and discussions.

References

[**Arbib and Lieblich 70**] "Motivational Learning of Spatial Behavior", Michael A. Arbib and Israel Lieblich, *Systems Neuroscience*, 1970.

[**Brooks 90**] "The Behavior Language; User's Guide", Rodney A. Brooks, *MIT Artificial Intelligence Lab Memo # 1227*, April 1990.

[**Brooks 87**] "A Hardware Retargetable Distributed Layered Architecture for Mobile Robot Control", Rodney A. Brooks, *Proceedings IEEE Robotics and Automation*, Raleigh, NC, April 1987, 106-110.

[**Chatila and Laumond 85**] "Position Referencing and Consistent World Modeling for Mobile Robots", R. Chatila and J. Laumond, *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, March 1985.

[**Gallistel 89**] "Animal Cognition: The Representation of Space, Time and Number", C. R. Gallistel, *Annual Review of Psychology 1989*, Vol 40, Palo Alto, California, 155-189.

[**Gould 82**] "Ethology: The Mechanisms and Evolution of Behavior", James L. Gould, *W. W. Norton and Company, New York*, 1982.

[**Kuipers and Byun 88**] "A Robust, Qualitative Approach to a Spatial Learning Mobile Robot", Benjamin J. Kuipers and Yung-Tai Byun, *SPIE Advances in Intelligent Robotics*

Systems Proceedings, November 1988.

[Mataric 90a] "A Distributed Model for Mobile Robot Environment-Learning and Navigation", Maja J Mataric, *MIT Artificial Intelligence Lab Technical Report #1228*, June 1990.

[Mataric 90b] "Qualitative Sonar Based Environment Learning for Mobile Robots", Maja J Mataric, *SPIE 1989 Mobile Robots IV Proceedings*, November, 1989.

[Mataric 90c] "Navigating With a Rat Brain", Maja J Mataric, *Proceedings, Simulation of Adaptive Behavior: From Animals to Animats*, MIT Press Bradford Books, 1990.

[Mataric and Brooks 90] "Learning a Distributed Map Representation Based on Navigation Behaviors", Maja J Mataric and Rodney A. Brooks, *Proceedings of 1990 USA-Japan Symposium on Flexible Automation*, June 1990.

[Moravec and Elfes 85] "High Resolution Maps From Wide Angle Sonar", Proceedings of the 1985 IEEE International Conference on Robotics and Automation, *Hans P. Moravec and Alberto Elfes*, St. Louis, MO, March 1985.

[Quillian 69] "Semantic Memory", M. Ross Quillian, *Semantic Information Processing, Marvin Minsky, ed.*, MIT Press, Cambridge, MA, 1969, 227-270.

[Waterman 89] "Animal Navigation", Talbot H. Waterman, *Scientific American Library*, New York, 1989.

[Wehner 87] "'Matched Filters' – Neural Models of the External World", Rüdiger Wehner, *J. comp. Physiol. A* 161, 511–531.