15th Triennial World Congress, Barcelona, Spain

TASK-DRIVEN, MULTIPLE ABSTRACTION FOR MODELING MOBILE ROBOT LARGE-SCALE SPACE*

Juan A. Fernández, Javier González

Dpto. Ingeniería de Sistemas y Automática, Universidad de Málaga
Campus Teatinos - E.T.S.I. Informática
29071 Málaga (Spain)
e-mail: {jafma,jgonzalez}@ctima.uma.es

Abstract: A mobile robot needs a model of its environment to perform planned tasks. Abstraction-based (hierarchical) arrangements of information allow the robot to plan its tasks efficiently. In this paper, a model with multiple hierarchies of abstraction is applied to mobile robots. It provides better adaptability to a wider range of environments, tasks, and agents than single-abstraction models. A system is also presented which automatically constructs such a model. It is inspired on the idea that the best model for an agent is the one that allows it to operate most efficiently. Some example applications to mobile robots illustrate these ideas. *Copyright (c) 2002 IFAC.*

Keywords: Mobile Robots, Hierarchical Structures, Graph Theoretic Models, Learning Systems, Robot Navigation.

1. INTRODUCTION

Many works can be found in the literature relating to modeling the world where mobile robots explore, navigate, and plan complex tasks (Kuipers, 2000; Fennema *et al.,* 1990; Dario and Rizzi, 1996; Fernández and González, 1997). Since mobile robots need to explore large-scale spaces that are not visible from a single vantage point, techniques are required to create "global maps" by integrating a number of "local maps" acquired during exploration. Usually, free space, obstacles, and landmarks existing in the environment are modeled as 2D maps in a variety of formats. These models range from pure geometrical (González *et al.,* 1994; Elfes, 1989) to qualitative or hybrid (qualitative plus quantitative) representations (Levitt *et al.,* 1987; Kuipers, 2000). The main problem of pure geometrical models is the error accumulation in geometrical measurements (Engelson and McDermott, 1992). Qualitative models avoid this problem, but often do not provide methods for optimizing the amount of information retrieved for each operation, that is, for discarding details that are not required.

This problem can be solved by using hierarchical models (Remolina *et al.,* 1999; Fernández and González, 1998a; Fennema *et al.,* 1990; Dario and Rizzi, 1996), which can be seen as particular types of qualitative or hybrid models. A classic hierarchical model arranges information at different levels of detail, called hierarchical levels, in a single hierarchy of abstraction, possibly including other types of information (geometrical, procedural, etc.) associated with its elements. This kind of representations provides interesting advantages: it allows the robot to select the amount of detail that is more appropriate for each operation; when information is abstracted, concepts that are barely sensitive to changes in the dynamic real world are created; the operations on the model can yield partial results before completion; etc.

In this paper, we contribute with two extensions of classic single-hierarchy graph-like models for mobile robots: a) *multiple hierarchies of abstraction*, and b) *task-driven automatic construction of multiple hierarchies of abstraction.* Both ideas are inspired by the way human beings arrange information for operating in the world

* This work has been supported by the Spanish Government under the research project CICYT-TAP99-0948.

efficiently. The utility of multiple hierarchies of abstraction has been demonstrated recently in (Fernández and González, 2002) regarding the operation of hierarchical path search (using hierarchical information for efficiently searching routes that connect elements in the model).

In the rest of the paper, multiple abstraction and task-driven construction of abstractions are presented. Section 2 overviews a model of multiple abstraction called Multi-AH-graph. Section 3 describes an implementation of the task-driven paradigm called CLAUDIA. Section 4 presents experiments developed with both CLAUDIA and the Multi-AH-graph model for mobile robots. Finally, some conclusions and future work are outlined.

2. A MODEL OF MULTIPLE ABSTRACTION

This section describes a graph-like model of multiple abstraction, called Multi-AH-graph, which is based on simpler representations such as plain graphs or single-hierarchy graphs. Complete formalizations of all these models can be found in (Fernández, 2000; Fernández and González, 2002).

2.1 Single-Hierarchy Graph Model (AH-graph)

Informally, an AH-graph is a sequence of **hierarchical levels**. Fig. 1 shows two examples of AH-graphs. Each **hierarchical level** contains a *plain directed multigraph,* that is, a plain directed graph which can hold more than one arc between a given pair of nodes. The weights of these arcs belong to a set of **cost intervals**, as is shown later on. In addition, each hierarchical level (except the higher one) defines two **abstraction functions** that permit to abstract the nodes and arcs of the level, respectively, to nodes and arcs of the next higher level (called **supernodes** and **superarcs** of the lower level nodes and arcs). Using the inverse of these functions, we can obtain the set of nodes that are abstracted to a given supernode. That set is called a **cluster.** As we will see further on, **clustering** a plain graph is the basic operation for automatically constructing a hierarchy or a multi-hierarchy.

The lowest level of the hierarchy is also called **ground hierarchical level** of the AH-graph. It represents the data with the maximum amount of detail that is available. The highest level is also called **universal hierarchical level** of the AH-graph, and it represents the same data with the minimum amount of detail (usually as a single node). The way this reduction of data is set up is defined by the abstraction functions for nodes and arcs. The only existing connections between nodes of different hierarchical levels in an AH-graph are given by these functions.

The weights of the arcs of an AH-graph are defined as **cost intervals** (numerical intervals), in order to deal with uncertainty in the quantities they represent (a useful feature for robotic applications).

Non-structural information is represented in an AH-graph by **annotations** both in nodes and arcs. The number of annotations a node or arc can store is not restricted. An **annotation** is defined as a pair $(\mathbb{N},\delta)$, where the first value is an unique identifier for the annotation and the second value is a set (block) of data which depends on the type of information stored.

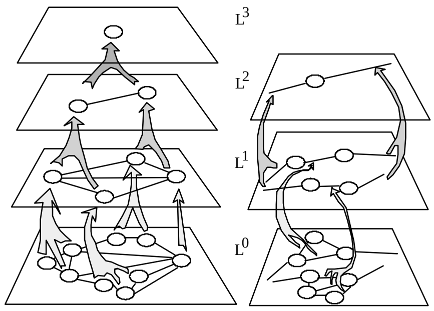

Fig. 1. Two examples of AH-graphs of four and three hierarchical levels, respectively. The AH-graph on the left side illustrates the behavior of the abstraction function for nodes from the lower to the higher hierarchical levels. The AH-graph on the right illustrates the behavior of the abstraction function for arcs.

2.2 Multi-Hierarchy Graph Model (Multi-AH-graph)

The proposed multi-hierarchical model is called Multi-AH-graph. It is based on the AH-graph model, that has been enhanced in order to deal with multiple hierarchies of abstraction.

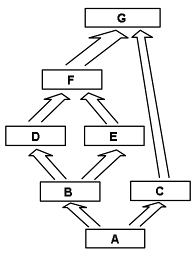

Fig. 2. An example of multi-hierarchy. In this directed acyclic graph, hierarchical levels are represented by rectangular boxes. Thick arrows represent abstraction links. Level **A** is the only ground hierarchical level of this multi-hierarchy, and level **G** the only universal level.

Informally, a Multi-AH-graph is a set of **hierarchies.** Each **hierarchy** is an AH-graph. In a Multi-AH-graph, any hierarchical level can be shared by any number of hierarchies. The number of shared hierarchical levels depends upon the power of detecting equivalence between hierarchical levels of the multi-hierarchy. A polinomial-time algorithm to detect hierarchical level equivalence (called **h-isomorphism**) has been presented elsewhere (Fernández, 2000).

In general, the hierarchical levels of a Multi-AH-graph can be considered nodes, and the abstraction links, arcs of a DAG[1]. The DAG induced by a given Multi-AH-graph M is denoted $\varphi(M)$ (see fig. 2).

The arcs of $\varphi(M)$ are abstraction links. An abstraction link between two hierarchical levels represents all the values of the abstraction functions for nodes and arcs connecting both levels. There is a different abstraction link for each hierarchy of M to which both hierarchical levels belong.

3. TASK-DRIVEN AUTOMATIC CONSTRUCTION OF MULTIPLE ABSTRACTIONS

For constructing automatically a "good" multi-hierarchy of abstraction, first a definition of this "goodness" must be given. This paper deals with a definition of "goodness" inspired by some psychological theories (Shepard, 1987; Harnad, 1987), which claim that abstraction provides an effective way of reducing the data gathered by the sensory apparatus of an agent. This claim is the core of our task-driven approach: a hierarchy of concepts is "good" *as long as it is good for performing efficiently the operations of an agent in a given environment.* Thus, the so constructed multi-hierarchies are used for solving tasks and are optimized depending on the efficiency achieved at those tasks. This approach satisfies two important objectives: first, it provides a flexible enough scheme that gathers hierarchical information from a wide range of sources, and second, it is a simple system that can adapt to very different types of agents, tasks, and environments, with a relatively small set of components.

Task-driven construction of multi-hierarchies has been implemented in a system called CLAUDIA[2] that improves a multi-hierarchy of abstraction for a given agent, tasks to perform, and environment. The system is divided into two main parts: the ground information interface (GI^2), and the optimizer (ACO) (see fig. 3). The GI^2 provides ground concepts and relations to the ACO. When CLAUDIA is used in mobile robots, the GI^2 is the interface between the information acquired by the sensory system and the multi-hierarchical model: it yields a plain graph whose nodes represents elements detected by the sensors (for example: range data, regions in an image, or line segments from laser 2D maps), and whose arcs represent relations existing between nodes (for example: similar regions, line paralellism, etc.)[3]. The ACO subsystem constructs multi-hierarchies upon the ground plain graph provided by the GI^2 by using several clustering methods, and optimizes the result through task-driven techniques.

The internal composition of the ACO (shown in fig. 3) permits us to plug in different procedures for acquiring hierarchical information. This is done through the *clustering* and *single-hierarchy constructor* modules. The former is in charge of defining groups of concepts (nodes) in a given plain graph, whereas the latter constructs the hierarchical levels of a given hierarchy by using the clustering module recursively, and detects h-isomorphism for reducing storage requirements. The clustering module plays the main role in the construction of multi-hierarchies.

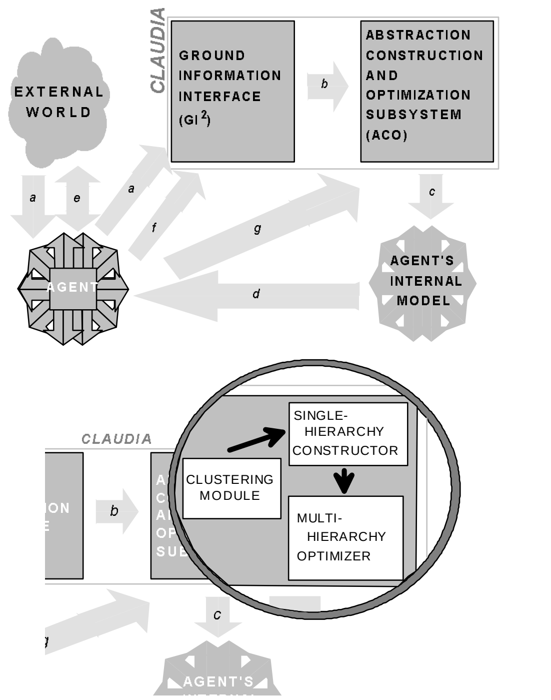

Fig. 3. *Top:* CLAUDIA as a part of an agent tha operates in a given environment. a) Externa world data is acquired by the GI^2 througl agent's physical perceptors, and provided to the optimizer as a plain graph. b) The ACC constructs an initial Multi-AH-graph anc optimizes it using feedback information fron the results of operating in the real world. c) The Multi-AH-graph may be part of a more sophisticated internal model of the agent and its environment. d) The physical agent operates ir the real world by consulting its internal model. e) This produce consequences (f) and (g) that are also a source of information for the GI^2, and the ACO. *Bottom:* Components of the task-driver automatic abstraction subsystem of CLAUDIA (ACO).

The multi-hierarchy optimizer module of the ACO subsystem is in charge of improving the goodness of a multi-hierarchy. It searches automatically for better multi-hierarchies of abstraction using a classic hill-climbing algorithm (other optimization procedures are possible but appreciably more complex (Holland, 1992; Hopfield and Tank, 1985)). Broadly, the implemented procedure consists of making local variations in the shape of the current multi-hierarchy to produce new multi-hierarchies (its *neighbors)*, and then selecting the best neighbor for continuing the search. The variations of the current multi-hierarchy are

[1] Directed Acyclic multiGraph.
[2] C.L.AU.D.I.A.: Concept Learning, AUtonomous Device that Improves Abstraction.
[3] Both nodes and arcs can store other types of information as annotations (range measurements, distance traveled, etc.).

generated by creating randomly new hierarchical levels (through new clusterings).

4. EXPERIMENTS

CLAUDIA and the Multi-AH-graph model have been implemented for large-scale space modeling in mobile robots. In the first set of experiments (section 4.1), the ACO subsystem of CLAUDIA is evaluated by providing it with some particular types of plain graphs. The definition of goodness has been based on the task of hierarchical path search (Fernández and González, 2002). In the second set of experiments (section 4.2), CLAUDIA is included as part of the software architecture of the RAM-2 mobile robot in order to plan paths for navigation.

Based on this particular kind of task, the goodness of a multi-hierarchy is defined to account for the following issues: the **percentage of path planning tasks that cannot be solved** with the multi-hierarchy (f_1), the **average computational cost** of solving path planning tasks in the multi-hierarchy f_2), the **average optimality** of the solution paths f_3), and the **storage needs** for the multi-hierarchy f_4). These factors are expressed as a weighted linear combination:

$$Goodness(M) = w_1.f_1 + w_2.f_2 + w_3.f_3 + w_4.f_4 \quad \textbf{[1]}$$

4.1 Evaluation of the ACO Subsystem

CLAUDIA has provided good results taking into account the exponential complexity of the Multi-AH-graph model. Three different plain graphs have been used as ground levels: a *grid* graph consisting of a rectangular grid of 100 nodes with relations among neighbor nodes; a *random* graph of 100 nodes with a 25% density of relations; and a 100 nodes *structured* graph formed by 20 groups of 5 nodes that have a high density of internal relations (70%), and a lower density of relations among nodes of different groups (1 or 2 arcs). A graph-partitioning algorithm based on a growth procedure similar to the STAR algorithm (Mkadmi, 1993) has been used for clustering.

Different evaluation tests have been performed by defining different weights for each of the factors of expression **[1]**. Table 1 shows the average percentage of improvement obtained in the goodness of the final multi-hierarchies with respect to the initial ones, for a set of five different tests and for each of the three plain graphs. All the tests have been done on a Pentium III computer running at 450Mhz. Although the optimization time has been limited to 4 hours, and then the algorithm stopped, that time seems to be acceptable since the improvement in goodness decreases exponentially at the end.

It can be noticed from table 1 that in the first three tests (General test (G.), Computational Cost test (C.C.), and Path Optimality test (P.O.)), the random graph yields better results than the other graphs. This is explained by the implicit structures existing in the grid and structured graphs, which imply a smaller number of alternatives when clustering the ground graph, and therefore, a smaller improvement in goodness than in the random graph.

Some other conclusions can be drawn from these results: first, the improvement in the goodness of the initial multi-hierarchy (constructed at random) are quite good: about 30% in the grid and structured graphs, and about 60% in the random graph (in the optimization of Solving Capability (S.C.) and Storage Requirements (S.R.) tests, the results are more similar to each other since those goodness definitions are not so dependent on the clustering of nodes); second, in general, the larger the number of hierarchies of the initial multi-hierarchy, the better the absolute value of goodness, because a multi-hierarchy with more than one hierarchy adapts better to a set of different problems (this effect has been studied in (Fernández, 2000)).

	G(M)	Random	Grid	Structured
G.	$0.5.f_2 + 0.5.f_3$	57.89 %	26.93 %	32.21 %
C.C.	f_2	59.05 %	20.51 %	35.72 %
P.O.	f_3	52.08 %	35.58 %	41.37 %
S.C.	f_1	0.17 %	0 %	0 %
S.R.	f_4	45.84 %	50 %	50 %

Table 1. Results of the evaluation tests performed by CLAUDIA during 4 hours of computation. Each row shows the average percentages of improvement in the goodness of the initial multi-hierarchy. Notice that in the solving capability test (S.C.) no improvement is achieved, since the initial multi-hierarchies are optimal yet.

4.2 Implementation of CLAUDIA in the RAM-2 Mobile Robot

CLAUDIA has been plugged into the software architecture of the RAM-2 mobile robot (fig. 4). RAM-2 has been designed and constructed in the System Engineering and Automation department of the University of Malaga. It has a variety of sensors such as two laser scanners, a sonar ring, and a camera, and a manipulator arm placed in front of the platform. In the RAM-2 architecture a 2D geometrical map builder module obtains line segment maps of the areas where the robot navigates using the laser rangefinder (González *et al.,* 1992), a position estimator module calculates the position and orientation of the robot by matching the data acquired by the radial laser rangefinder against the maps previously built by the map builder (González *et al.,* 1995), a local navigator module moves the robot to a desired relative location avoiding collisions with unexpected obstacles (Muñoz *et al.,* 1998), and a command executor module takes navigation commands (for example: "go to the node labeled B in the Multi-AH-graph") and moves the robot along a geometrical trajectory derived from path searching in the Multi-AH-graph previously generated by CLAUDIA. This command module also implements the GI^2 as a human-machine interactive process.

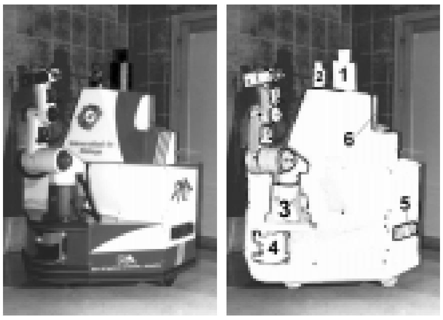

Fig. 4. The RAM-2 mobile robot, on which an application that uses CLAUDIA has been implemented. 1) Radial laser scanner. 2) Camera 3) Robot arm. 4) Frontal laser scanner. 5) Sonars. 6) On-board display.

At a first stage of the experiment, the ground hierarchical level of the Multi-AH-graph is built through the interactive GI^2: the operator provides the structural information (distinctive places) that supports the perceptual information gathered by the robot sensors (maps and reference systems from the map builder and the position estimator modules, respectively). The resulting annotated plain graph is provided to the ACO subsystem of CLAUDIA, which optimizes an initial multi-hierarchy generated at random through performing path planning tasks.

At a second stage, when the final Multi-AH-graph is completely built, the command executor module can receive a navigation command for moving the robot to another place represented in the Multi-AH-graph. For this purpose, the executor firstly issues a path-finding request to the Multi-AH-graph for searching a path between both nodes. Then, it follows the solution path moving the robot from the location represented by each node to the next (Fernández and González, 1998b).

The floor plan of the real environment where the robot navigates is shown in fig. 5. A simplified floor plan of the environment and the distinctive places marked by the human operator by manual guidance of the robot are also shown.

For this experiment, the factors that define the goodness of the multi-hierarchies are defined as follows:

$$Goodness(M) = 0.25f_1 + 0.25f_2 + 0.25f_3 + 0.25f_4 \qquad \textbf{[2]}$$

CLAUDIA begins by generating an initial multi-hierarchy with only one hierarchy (since we want to reduce the computational cost of calculating goodness) with three hierarchical levels. The lowest one corresponds to the ground graph depicted in fig. 5, and the highest one consists of only one node. The clustering of nodes of the ground level of this hierarchy is not very good, since it defines clusters of very different sizes[4]. The goodness of that multi-hierarchy is 0.618208.

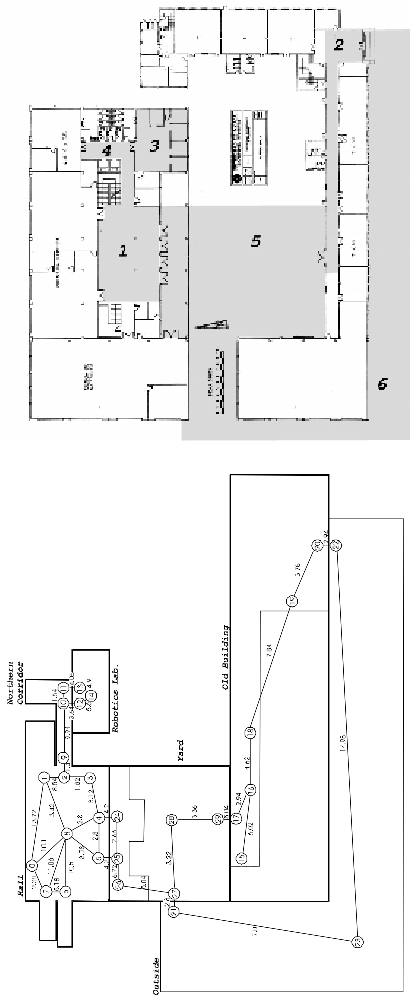

Fig. 5. Floor plan of the real environment where RAM-2 navigates and topology extracted by the GI^2. 1- Hall. 2- Old building. 3- Robotics Lab. 4 Northern corridor. 5- Yard. 6- Outside.

The final multi-hierarchy (after 4 hours of optimization) has again one hierarchy (due to the storage requirement factor of expression **[2]**) with five hierarchical levels. The first level is shown on the floor plan in fig. 6. The clusterings of this multi-hierarchy are more uniform in size, which improves its value of goodness: 0.903202. The final multi-hierarchy is 46,1% better than the initial multi-hierarchy for performing path planning operations, if the goodness is defined as shown in expression **[2]**.

[4] As explained in (Fernández and González, 2002), the computational cost of solving search problems is low when clusters are small, while the optimality of the solutions is better when clusters are large. If a clustering defines clusters of very different size, the worst cases for computational cost and solution optimality are found, and therefore the goodness of the multi-hierarchy as defined in expression **[2]** tends to decrease. Multi-hierarchies with uniform size of clusters tends to avoid bad results in computational costs and path optimality.

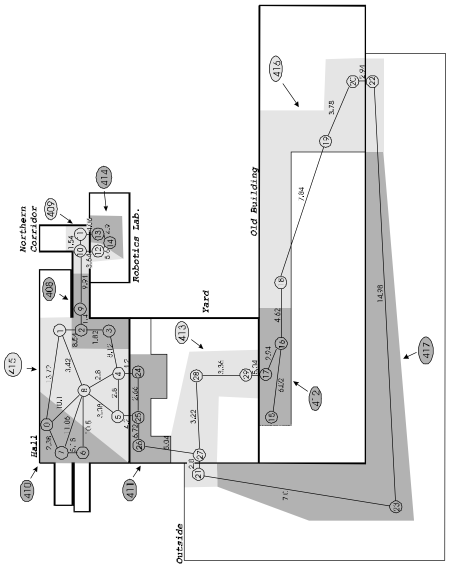

Fig. 6. First level of the resulting clusterization of the space in the final multi-hierarchy. Notice that arcs with greater costs (large distances) tend to be external (connecting different areas).

5. CONCLUSIONS AND FUTURE WORK

This paper contributes with two important extensions of hierarchical models of large-scale space: we have introduced a multiple abstraction model of the environment and presented an automatic system for constructing such a model from environmental information. The model and the automatic construction system have been successfully implemented in the RAM-2 mobile robot. Several experiments demonstrated the suitability of both approaches to increase both the adaptability of the robot to its environment and its efficiency in performing operations.

The work presented in this paper opens a wide area of research, since an automatic abstraction system such as the one described here can be adapted to many applications (stereo computer vision, geographical information systems, network routing, cognitive science, etc.), and the system itself can be extended and improved. Some lines of future research are: more general and powerful definitions for the goodness of a multi-hierarchy, mechanisms capable of coping with dynamic variations in the lower hierarchical levels, and the improvement of the Multi-AH-graph model for dealing with uncertainty in the existence of relations and concepts.

Currently, we are also working on connecting the Spatial Semantic Hierarchy (Kuipers, 2000) as the ground information interface (GI^2) of CLAUDIA (Remolina *et al.,* 1999).

REFERENCES

Dario M. and Rizzi S. (1996) Dynamic Clustering of Maps in Autonomous Agents. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 18, No. 11, pp. 1080-1091.

Elfes A. (1989), Using Occupancy Grids for Mobile Robot Perception and Navigation. In *IEEE Computer.*

Engelson S.P. and McDermott D.V. (1992), Error Correction in Mobile Robot Map Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'92).*

Fennema C., Hanson A., Riseman E., Beveridge J.R., and Kumar R. (1990), Model-Directed Mobile Robot Navigation. In *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 20, No. 6, pp. 1352-1369.

Fernández J.A. (2000), Modeling and Generation of Multiple Abstractions for Representing Large-Scale Space. An Application to Mobile Robotics. *PhD thesis,* System Engineering and Automation Department, University of Málaga (Spain). ISBN 84-607-0598-6.

Fernández J.A. and González J. (1997), General World Representation for Mobile Robot Operations. In *Proceedings of the 7th Conference of the Spanish Association for the Artificial Intelligence (CAEPIA'97),* pp. 35-44.

Fernández J.A. and González J. (1998a), Hierarchical Path Search for Mobile Robot Path Planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98).*

Fernández J.A. and González J. (1998b), Mobile Robot Path Planning and Navigation Using Hierarchical Graphs. In *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98).*

Fernández J.A. and González J. (2002), Multi-Hierarchical Path Search. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 1.

González J., Ollero A., and Reina A. (1994), Map Building for a Mobile Robot Equipped with a Laser Range Scanner. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'94).*

González J., Stentz A., and Ollero A. (1995), A Mobile Robot Iconic Position Estimator using a Radial Laser Scanner. In *Journal of Intelligent&Robotic Systems,* Vol. 13, pp. 161-179.

Harnad S. (1987), Category Induction and Representation. In *Categorical Perception: the Groundwork of Cognition,* Harnard S. (ed.), New York, Cambridge University Press, Chapter 8.

Holland J.H. (1992), Genetic Algorithms. In *Scientific American,* Vol. 267, No. 1, pp. 66.

Hopfield J.J. and Tank D.W. (1985), Neural Computation of Decisions in Optimization Problems. In *Biological Cybernetics,* Vol. 52, pp. 141-152.

Kuipers B.J. (2000), The Spatial Semantic Hierarchy. In *Artificial Intelligence* , 119, pp 191-233.

Levitt T.S., Lawton D.T., Chelberg D.M., and Nelson P.C. (1987), Qualitative Navigation. In *Proceedings of the DARPA Image Understanding Workshop,* pp. 447-465.

Mkadmi T. (1993), Speeding Up State-Space Search by Automatic Abstraction. Master's thesis, University of Ottawa.

Muñoz V.F., Cruz A., García-Cerezo A. (1998), Speed Planning and Generation Approach Based on the Path-Time Space for Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98).*

Remolina E., Fernández J.A., Kuipers B.J., and González J. (1999), Formalizing Regions in the Spatial Semantic Hierarchy: an AH-graphs Implementation Approach. In *Lecture Notes in Computer Science 1661,* pp. 109-124. Freksa C. and Mark D.M. (eds.). Berling, Springer. ISBN 3-540-66365-7.

Shepard R.N. (1987), Toward a Universal Law of Generalization for Psychological Science. In *Science,* Vol. 237, pp. 1317-1323.