Supervised Multi-Agent Exploration Of Unknown Environments

Dario Maio, Stefano Rizzi

DEIS - CIOC-CNR - Facoltà di Ingegneria, Università di Bologna

Abstract - Exploration is a central issue for autonomous agents which must carry out navigation tasks in environments whose detailed description is not known *a priori*. In our approach the environment is described, from a symbolic point of view, by means of a graph; clustering techniques allow for further levels of abstraction to be defined, leading to a multi-layered representation. In this work we propose an exploration algorithm in which a supervisor co-ordinates several agents. All agents are equal, and pursue a local strategy aimed at exploring a finite portion of a graph; nevertheless, the existence of multiple levels of abstraction in the environment representation allows for the agents' behaviours to differentiate. The supervisor plays the twofold role of assigning dynamically to each agent an abstraction level for exploring the environment, and of co-ordinating the agents assigned to the same level using the knowledge acquired by the agents on the higher levels. Two exploration agendas are employed: a local agenda is used by the agents to keep track of the unexplored routes, whereas a global agenda is used by the supervisor to calculate the agent demands on each abstraction level.

I. INTRODUCTION

Autonomous agents (AAs) are mobile versatile machines capable of interacting coherently with an environment and executing a variety of tasks in unpredictable conditions [9]. They can be profitably employed to explore hazardous and hostile environments, as well as to perform repetitive tasks in dynamic environments such as offices, factories, hospitals, etc.

Autonomy means capability of navigating the environment; navigation, in turn, necessarily relies on a description of the environment. It is important to draw a distinction between *knowledge* and *meta-knowledge* of the environment. By knowledge we mean a topological and metric description of the specific environment the agent is moving in, more or less detailed according to the complexity of the tasks required. With the term meta-knowledge, instead, we denote general information concerning a whole category of environments: for instance, a geometric model of the shapes of the objects which are likely to be met in the environment. Both kinds of knowledge may be either known *a priori* to the agent or not, depending on the applications considered.

In our work we consider the case where the agent is given no *a priori* knowledge, so that it must learn the environment description on-line by interpreting sensor data. As to meta-knowledge, we assume that it is available in terms of:

- Descriptions of typical sensor patterns present in the environment. Selection of patterns corresponding to distinctive or significant categories of objects

and places enables recognition of *landmarks* through a sensor-based classification algorithm [5] [10]. Examples of significant categories are computers and telephones in office environments, buildings and monuments in urban environments.
- Characterization of semantically significant *clusters* of objects or places, obtained through sensor-based recognition of passageways between different clusters [6]. Example of clusters in office environments are rooms and floors, identified by recognizing doors and stairs.

On these assumptions we have proposed, in [7], a multi-layered architecture for representing the environmental knowledge to be used by an AA for navigation. Our approach is based in the first place on the distinction between reactive motion control and high-level path planning, both necessary for autonomous navigation, and carried out by relying on sub-symbolic and symbolic knowledge, respectively [1] [4]. Meta-knowledge for aggregation allows for the symbolic representation of the environment to be distributed on different abstraction levels in order to supply a richer description of the environment and, at the same time, carry out navigation tasks more efficiently [8]. In section II we outline the main concepts underlying the architecture proposed.

The lack of *a priori* knowledge of the environment gives relevance to the problem of on-line exploration, the subject of this paper. The existence of different abstraction levels allows for the global topological and metric structure of the environment to be acquired rapidly, so as to make the agents ready as soon as possible to execute navigation tasks. Section III proposes an algorithm for supervised multi-agent exploration, where the supervisor may be either an external controller or one of the agents. The supervisor dynamically assigns each agent the task of exploring the environment at a specific abstraction level, and co-ordinates the agents assigned to the same level using knowledge acquired by the agents that explore the environment at higher abstraction levels. Two agendas are employed: a local and a global agenda. The agents share the local agenda and use it to carry out their exploration strategy. The global agenda is used by the supervisor to assign agents to the different abstraction levels.

II. LAYERED ARCHITECTURE FOR ENVIRONMENT REPRESENTATION

A *knowledge layer* is a view of the environment at a specific abstraction level; it includes only those details of the environment which are significant for a specific family of tasks or sub-tasks, and represents them in the most suitable formalism. A multi-layered representation of the environment is semantically richer than a "flat" representation. Besides, complex path planning tasks can be decomposed into a number of sequential or parallel sub-tasks, each supported by a specific layer.

During exploration, agents experience the environment through their sensors. We assume that *metric*, *visual* and *symbolic* information are acquired. By metric information we mean that the agent can compute where it is at any time, for instance by means of a compass, an odometer and an altimeter. By visual information we mean that the agent is equipped with a sensor (a sonar or a camera) which returns an "image" of the nearby surroundings. By symbolic information we mean that some landmarks are tagged with a name, obtained by "reading" a sign or something written.

Sensory knowledge cannot be directly exploited, so it must be reorganized and interpreted. This is done by abstracting from the set of sensor measures one or more layers which are more suitable for some of the agent's tasks. Each of these layers may in turn generate other layers for other tasks, through a procedure of progressive abstraction which creates a taxonomy of layers. The architecture we propose includes the following layers:

- *Sub-symbolic layer*. This is responsible for inter-landmark movement, including obstacle avoidance, and is supported by a neural network whose input consists of measures from the metric and visual sensors, and whose output is the direction to be followed [2].
- *Symbolic layer*. This is the foundation for path planning, and describes the environment as a map of landmarks and feasible inter-landmark paths (*routes*). A route is an abstraction corresponding to the straight-line connection between two landmarks, and is described by a cost expressing for instance the length of the corresponding physical path. The symbolic layer is represented through a directed graph whose vertices and arcs correspond, respectively, to landmarks and routes.
- *Clustered layers*. A hierarchy of clustered layers at different abstraction levels may be defined starting from the symbolic layer. At the first abstraction level, the environment is described as a map of clusters of landmarks and inter-cluster passageways (*bridges*); at the subsequent levels, each cluster includes clusters of the level below. Clustered layers are represented as directed graphs whose vertices and arcs correspond, respectively, to clusters and bridges.
- *Meta-layers*. These are abstracted from the symbolic layer or from any clustered layer, and describe the entities in the environment through the category they belong to.

In the following we introduce the terminology used in section III to outline the exploration algorithm.

Given a graph $\mathcal{G}$=(V,A), with V a set of vertices and A a set of arcs, we denominate with *clustering* a partition of the vertices and arcs of $\mathcal{G}$ into a set of clusters and a set of bridges. A *cluster* is a connected sub-graph of $\mathcal{G}$. The *bridge* between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is the set of arcs of $\mathcal{G}$ which connect a vertex of $\mathcal{C}_i$ to a vertex of $\mathcal{C}_j$. All clusters and bridges are disjointed.

We call *1-clusters* and *1-bridges* the clusters and bridges determined by applying a clustering to the symbolic layer. The directed graph whose vertices and arcs correspond, respectively, to the 1-clusters and the 1-bridges is called a *1-clustered layer* (Figure 1). A clustered layer may in turn be clustered; in general, we name *k-clustered layer* (k=1,..n, where n is the maximum abstraction level) the graph obtained by applying clustering k times, starting from the symbolic layer. The clusters and bridges of a k-clustered layer are called *k-clusters* and *k-bridges*, respectively. The symbol $\mathcal{C}^{(k)}$ denotes a k-cluster. For analogy, we will also call the symbolic layer a 0-clustered layer.

Again, we will assume that the n-clustered layer contains exactly one n-cluster $C^{(n)}$ (that is, the (n-1)-clustered layer is not partitioned).

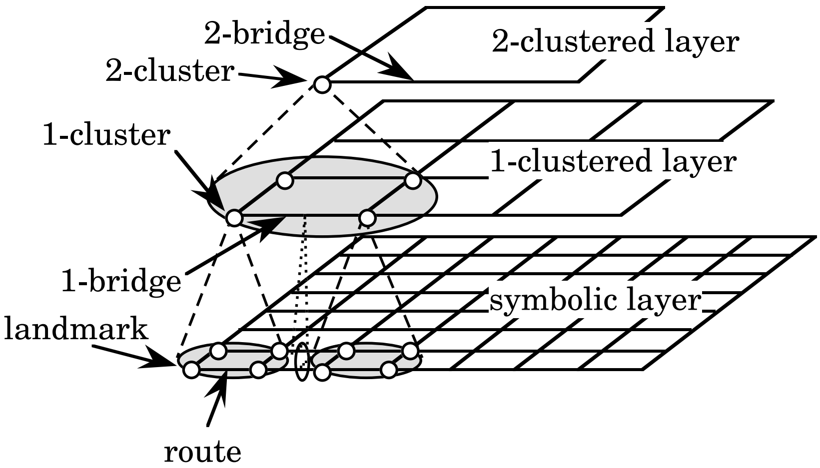

Figure 1. Hierarchy of clustered layers abstracted from the symbolic layer.

Consider for instance an office environment where landmarks correspond to telephones, computers, photocopiers, etc. Within the 1-clustered layer the environment would probably be modelled as a graph of rooms (1-clusters) connected by doors (1-bridges); similarly, 2-clusters might correspond to the different floors and 2-bridges to stairs.

III. Algorithm for supervised multi-agent exploration

In our approach, exploration is carried out at a symbolic level. The link between symbolic exploration and the sensor level is established by assuming that:

- When an agent is located in a landmark, it can make hypotheses about the directions of the possible routes departing from that landmark. If a restricted number of paths are physically possible in the environment (for instance, the streets in a city), these can be directly recognized through the visual sensors; otherwise, assuming that the agents can sense landmarks within a fixed range, we consider routes the paths leading to the neighbouring landmarks.
- The agents can recognize the routes which mark a change of cluster at any abstraction level (for instance, a route traversing a threshold).
- When an agent is located in a landmark, it can choose to explore one of the routes sensed by assigning the estimated position of the destination landmark as the current goal to the sub-symbolic layer.

We give the following definitions:

A 0-cluster (i.e., a landmark) has been visited when it has been sensed.
A k-cluster (k=1,..n-1) has been visited when at least one of the (k-1)-clusters it contains has been visited.

The k-clustered layer (k=1,..n) has been explored when all the k-clusters it contains have been visited.

In these terms, our goal can be formulated as the exploration of the graph corresponding to the symbolic layer, that is, the visit to all the landmarks (a complete exploration of the symbolic layer logically implies a complete exploration of all the clustered layers). The symbolic layer is partitioned into 1-clusters; if several agents are available, it is worth adopting a policy of subdivision of the explorative tasks, aimed at assigning each agent to a different 1-cluster. An efficient subdivision policy presupposes "awareness" of the existing 1-clusters: one or more agents should then be entrusted with the task of visiting all the 1-clusters, that is, of exploring the 1-clustered layer. Exploration of the 1-clustered layer, in turn, takes advantage from knowledge of the 2-clusters, so that the decomposition process is iterated as far as the maximum level of clustering, n. Therefore, each clustered layer (from level 0 to n-1) is assigned one or more agents who share exploration according to information reported from the agents at the higher levels.

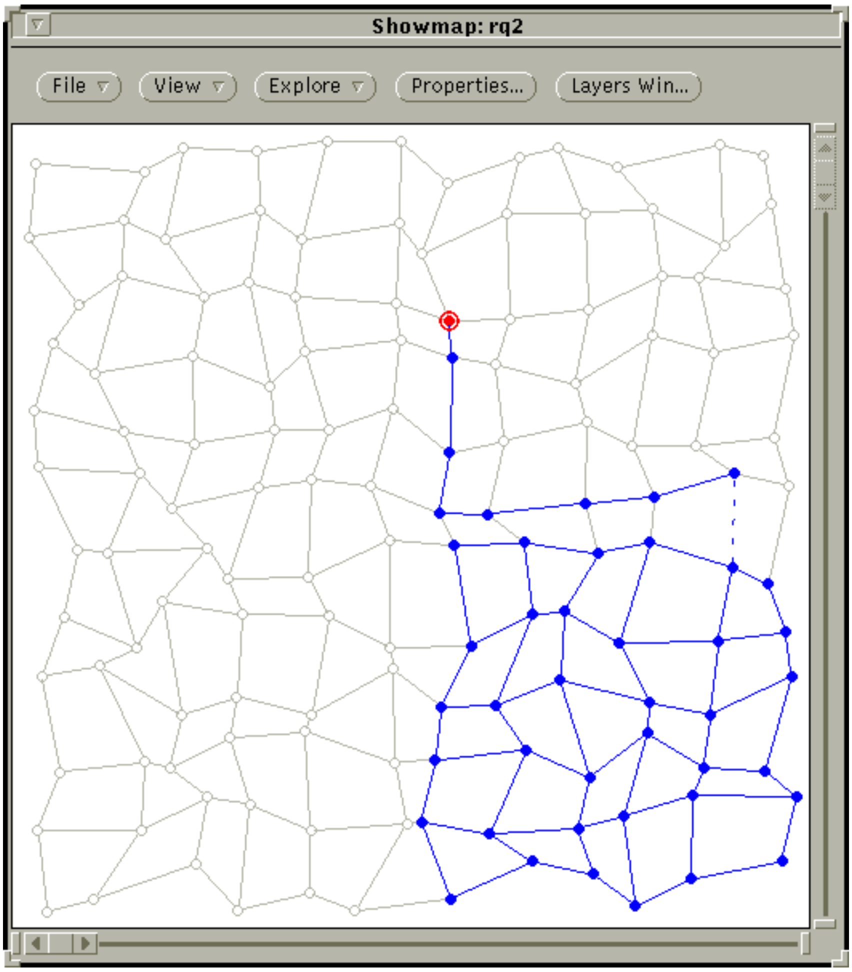

Figure 2. Partial exploration path of a 0-agent. The agent's current position is circled. Black landmarks have already been visited; black routes have already been undertaken. Dashed lines represent routes belonging to 1-bridges.

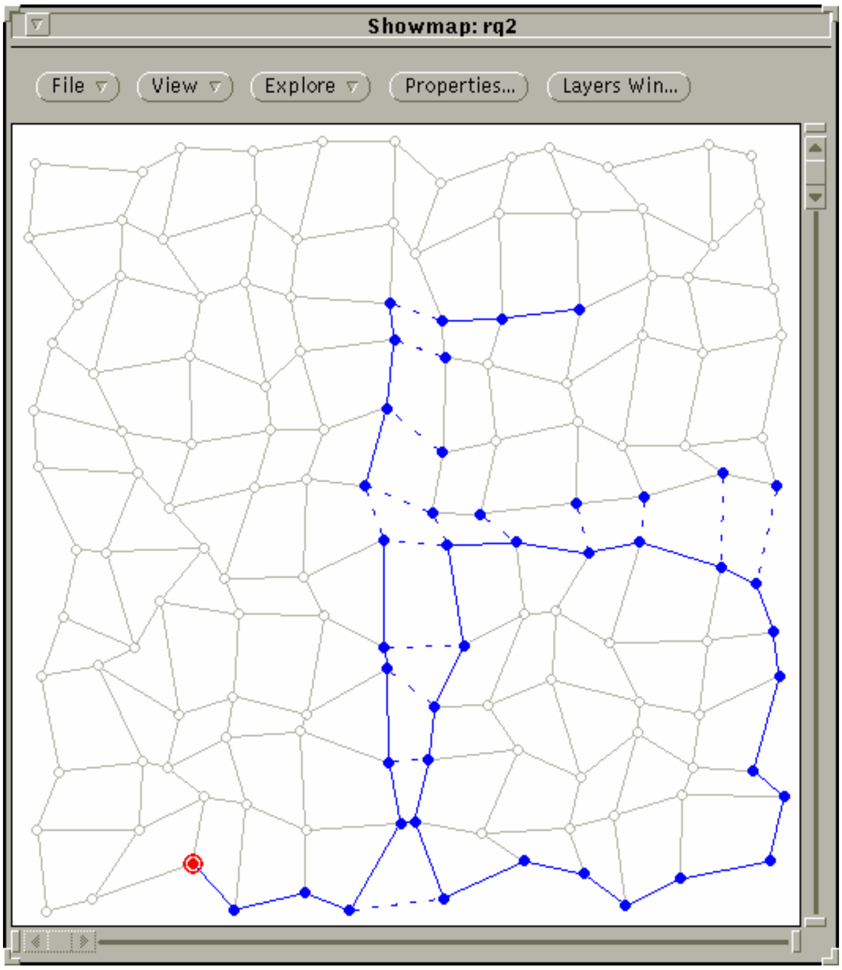

Figure 3. Partial exploration path of a 1-agent. The agent's current position is circled. Black landmarks have already been visited; black routes have already been undertaken. Dashed lines represent routes belonging to 1-bridges. The 1-agent first traverses the 1-cluster where it is positioned in order to reach a 1-bridge; then it begins to follow the edges of the 1-clusters aimed at discovering new 1-clusters and the 1-bridges between them.

A. The agents

All the agents are structurally equal, and capable of following a standard strategy aimed at exploring a finite graph based on local knowledge only. The strategy adopted is based on Tremaux's algorithm, modified so as to consider the contemporary presence of several agents and the existence of mono-directional routes (one-way streets, doors which can be opened one way only, etc.) [3]. As a matter of fact, the agents' behaviours are diversified by their assignment to increasing abstraction levels. We will call *k-agent* (k=0,..n-1) one whose task consists of exploring the graph representing the k-clustered layer. Agents' exploration strategy is supported by a local agenda which is structured in layers corresponding to the different clustered layers. The k-th layer of the local agenda reports, for each k-cluster visited, all the k-bridges that have not been undertaken yet.

Please note that, while visibility of the routes departing from a visited landmark is guaranteed by the sensor level, the same is not true for k-clusters (k>0). In fact, visiting for instance a 1-cluster only requires visiting one of its landmarks, while getting information of the departing 1-bridges entails following the whole edge of the 1-cluster. Figures 2 and 3 show the different exploration paths followed by a 0-agent and a 1-agent on the same map: the 0-agent carries out exhaustive exploration inside 1-clusters (Figure 2); the 1-agent, instead, follows the edges of the 1-clusters and undertakes the routes contained in the 1-bridges (Figure 3). From a behavioural point of view we might say that, though all agents are equally "curious" (due to their standard exploration strategy), those working on low layers are "meticulous", while those working on high layers are more "superficial".

B. The supervisor

The *supervisor* may be either an external controller or one of the agents; it is capable of communicating with all the agents and manages the knowledge base storing the environment representation. The first role played by the supervisor consists in deciding dynamically how many agents will be entrusted with environment exploration at each abstraction level. The second role is the co-ordination of the k-agents within the k-clustered layer, which is carried out by dynamically assigning to each k-agent the task of exploring a specific (k+1)-cluster (that is, a sub-graph of the k-clustered layer). Both roles are supported by a global agenda which reports, for each clustered layer, the current progress of exploration. More specifically, the scheduling of the k-agents (k=0,..n-1) is based on the list of all the (k+1)-clusters which have been visited by a w-agent (w>k) but have not been completely explored by any k-agent, yet. Figure 4 shows the data flows exchanged between the supervisor and the agents.

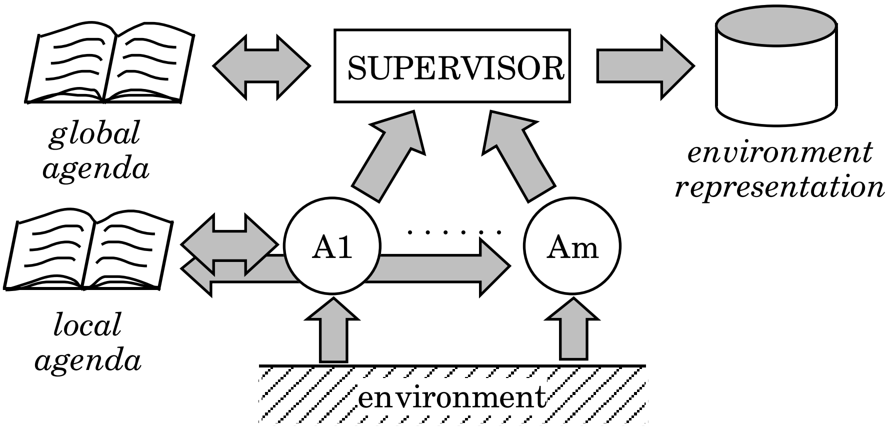

Figure 4. Data flows between the supervisor and the agents. The agents acquire data from the environment, and hand them to the supervisor which stores them in the knowledge base.

The algorithm used by the supervisor to manage the agents and the global agenda can be summarized as follows:

initialize agents' levels;
initialize agents' scopes;
initialize global agenda;
while global agenda is not empty do
{ handle events *visited* and *explored* ;
}
end exploration.

Event handling:

- when *visited*(A, $C^{(k)}$) do
  /* the k-agent A has visited a new cluster $C^{(k)}$; k>0*/
  { put $C^{(k)}$ in global agenda;
  if global agenda is unbalanced
  { decrease level of A to k-1;
  set scope of A to $C^{(k)}$;
  }
  }

- when *explored*($A^{(k)}$, $C^{(k+1)}$) do
  /* the k-agent $A^{(k)}$ has terminated exploration of cluster $C^{(k+1)}$ */
  { remove $C^{(k+1)}$ from global agenda;
  set scope of A to nearest (k+1)-cluster in global agenda;
  }

The agents' levels are initialized by assigning one agent to each level, starting from level n-1 and descending towards level 0; should the number of agents overcome the number of levels n, all the remaining agents are assigned to level n-1. The initial scope of a k-agent is the (k+1)-cluster which includes the k-cluster the agent is located in. The global agenda initially contains the current scopes of all the agents.

Every time a k-agent visits a new k-cluster, its level may be reassigned on the basis of the information contained in the global agenda. The supervisor policy in reassigning levels is aimed at keeping a balance between the discovery of new clusters and their exploration. As the level decreases, the complexity of the corresponding graph increases and exploration takes progressively more time; hence, if the number of agents assigned on each level were the same, the k-agents would visit more k-clusters than the (k-1)-agents could explore. In order to avoid this unbalance, agents are progressively moved towards the lower levels.

When a k-agent completes exploration of a (k+1)-cluster, it is assigned to the nearest unexplored (k+1)-cluster present in the global agenda; if all the (k+1)-clusters have already been explored, its level is decreased.

CONCLUSION

In this paper we have proposed an algorithm for supervised multi-agent exploration of environments. The layered representation adopted for the environment enables the agents to carry out exploration at different abstraction levels. The supervisor co-ordinates the agents by assigning them to the different knowledge layers and, inside each layer, to a restricted scope.

Some simulation results for our algorithm are shown in Figure 5. The exploration time is compared with an ideal time computed as the ratio between the mono-agent optimal exploration time and the number of agents.

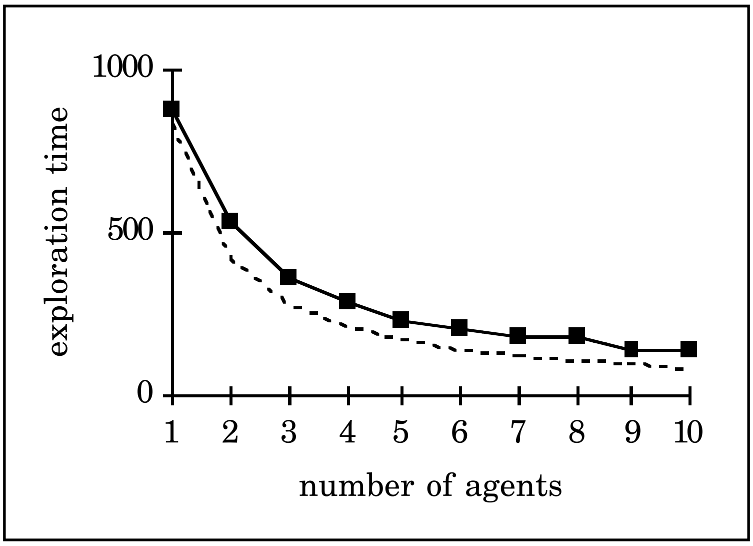

Figure 5. Exploration time on a sample of maps as a function of the number of agents employed. The dotted line shows the ideal exploration time, computed as the ratio between the mono-agent optimal exploration time and the number of agents.

REFERENCES

[1] P. Ciaccia, D. Maio and S. Rizzi, "Integrating knowledge-based systems and neural networks for navigational tasks", in *Proc. IEEE COMPEURO*, Bologna, Italy, pp. 652-656, 1991.

[2] P. Ciaccia, B. Montanari, "Reinforcement-based systems for solving navigational tasks in large domains", in *Proc. MCPA '93*, Halmstad, Sweden, 1993.

[3] J. Edmonds and E.L. Johnson, "Matching, Euler tours and the chinese postman", in Mathematical Programming, V, n. 1, pp. 88-124, 1973.

[4] E. Gat, "On the role of stored internal state in the control of autonomous mobile robots", in *AI Magazine*, 14, n.1, pp. 64-73, 1993.

[5] B.J. Kuipers and Y.T. Byun, "A robust, qualitative method for robot spatial learning", in *Proc. AAAI88*, Saint Paul, Minnesota, vol. 2, pp. 774-779, 1988.

[6] D. Maio and S. Rizzi, "Map Learning and Clustering in Autonomous Systems", to appear in *IEEE Trans. Patt. Anal. Machine Intell.*, 1993.

[7] D. Maio and S. Rizzi, "Knowledge architecture for environment representation in autonomous agents", to appear in *Proc. ISCIS VIII*, Istanbul, Turkey, 1993.

[8] D. Maio and S. Rizzi, "A hybrid approach to path planning in autonomous agents", to appear in *Proc. ICED*, Bangkok, Thailand, 1994.

[9] D. Nitzan, "Development of intelligent robots: achievements and issues", in *IEEE Journal of Robotics and Automation*, RA-1, n. 1, pp. 3-13, 1985.
[10] W.A. Phillips, P.J.B. Hancock, N.J. Willson and L.S.Smith, "On the acquisition of object concepts from sensory data", in *Neural Computers*, NATO ASI Series, vol. F41, R. Eckmiller and Ch.v.d. Malsburg Ed., Springer-Verlag, pp. 159-168, 1988.