Adaptive Automata for Mapping Unknown Environments by Mobile Robots

Miguel Angelo de Abreu de Sousa, André Riyuiti Hirakawa, and João José Neto

Escola Politécnica da Universidade de São Paulo,
Departamento de Engenharia de Computação e Sistemas Digitais,
Av. Prof. Luciano Gualberto, travessa 3, no.158, São Paulo - SP - Brasil - CEP 05508-900
{miguel.sousa, andre.hirakawa, joao.jose}@poli.usp.br

Abstract. Robotic mapping is one of the most important requirements for a truly autonomous mobile robot. Mobile robots should be able to building abstract representation of the physical environment, in order to navigate and work in such environment. This paper presents an adaptive way to make such representation. The proposed system allows the robot to explore all the environment and acquire the information incoming from the sensors (presence or absence of obstacles) while it travels. The robot may start the mapping process at any point of the space to be mapped. Due to the adaptability of the chosen method, the process has the capability of dynamically increase the memory requirements according to the already mapped area, even without any a priori knowledge of the environment.

Keywords: Adaptive Automata, Robotics, Navigation and Robotic Mapping.

1 Introduction

Early approaches for allowing mobile robots to move around used to employ a preliminary map of the environment stored in its memory. Those approaches do not provide an adequate solution since storing a complete geometrical map of the environment, searching the database for localization and the path planning process significantly increases the computational complexity of the system, making the approaches prohibitive for actual implementations [2].

Another problem related to those approaches refers to the repetitive nature of non-automatic mapping processes. Each unstructured environment in which the robot is intend to work, such as, buildings, offices, industries and agricultural fields, has to be mapped first and the resulting map must be manually registered in its memory.

There is also a question related to the possibility of robots to work in hazardous and unknown environments. Dangerous tasks like mining, undersea operations, working in disaster areas, space and planetary exploration are examples of situations in which robots may have to map the field before being able to work properly.

C. Lemaître, C.A. Reyes, and J.A. Gonzalez (Eds.): IBERAMIA 2004, LNAI 3315, pp. 562–571, 2004.

Stimulated by such reasons, robotic mapping has been a strongly researched topic in robotics and artificial intelligence for two decades, and still presenting challenging research subjects, e. g. mapping dynamic or large areas [9].

The present work proposes an adaptive mechanism to steer a robot to cover all its unknown environment and, during this exploration, to collect information from its sensors and to organize them as a map. This map is built in such a way that robot's navigation become easier.

2 Related Work

Since the early 80's robotic mapping research area has been split between metric and topological approaches. Metric maps represent the environment by using its geometric properties [4] [8]. Topological maps describe environments as a set of important places, which are connected by arcs [2] [6]. These arcs have attached information on how to navigate through such places. Nevertheless, the exact frontier between these approaches has always been fuzzy, since topological maps rely on geometric information about the world [9].

Adaptive devices change their structure and behavior according to their external stimulus. Such feature represents an intuitive and trustful way for modeling physical environments and to conduct the robot, despite the complexity of the environment.

2.1 Adaptive Automata

Adaptive automata, first proposed in [7], extends the concept of finite automata by incorporating the feature of performing dynamic self-reconfiguration in response to externally collected information. Such behavior provides adaptive automata with learning capability, which makes them suitable for representing knowledge.

It has been shown that adaptive automata are Turing-powerful devices [7] and they have also been applied on several applications, such as pattern recognition [3] and systems description [1].

Adaptive automata may be viewed as self-modifying state machines whose structure includes a set of states and a set of transitions interconnecting such states. States may be classified in: initial state; a set of final states; and a set of intermediate states. Incoming stimuli change the internal state of the machine.

The self-modifying feature of adaptive automata is due to the capability it has of changing its own set of transition rules. Adaptive actions may be attached to the transitions which are able to either add new states and transitions or remove already existents ones, consequently, achieving a new structure. Hence, incoming stimuli may change the set of internal states and modify the general configuration of the automaton. See [7] for details on concepts and notation.

Transition rules in adaptive automata are represented as:

$$(g , e , a) : B \rightarrow (g' , e' , a') : A$$

g: push-down store contents before the transition;
g': push-down store contents after the transition;

e: current state before the transition;
e': current state after the transition;
a: input stimulus before the transition;
a': input stimulus after the transition;
B: adaptive action before applying the transition;
A: adaptive action after applying the transition.

Adaptive actions A and B are both optional. Three different elementary adaptive actions are allowed: inspection – search the current state set for a given transition; deletion – erase a given transition from the current state set; and insertion – add a given transition to the current set of states. Such actions are denoted by preceding the desired transition by the signs ?, – and +, respectively. Figure 1 shows a graphic representation of the transition.

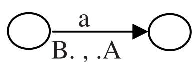

Fig. 1. Simplified transition (e , a) : B → e' : A when g, g' and a' are omitted

2.2 The Mapping Automaton

The initial work on representing physical environments by using adaptive automata has been proposed in [5]. The present paper extends this work.

The adaptive mapping automaton starts from a square lattice (figure 2a) consisting of nine states connected by special transitions, all of them denoting areas to be mapped. The central state is the initial state of the automaton, and represents the starting point of the exploring path.

In order to allow a clear presentation of the method, a graphic representation of the adaptive automaton will be used (The dot-marked state corresponds to the actual position of the robot and single lines represent areas not yet mapped.).

Initial automaton also presents special tags (X), marking corner states, and special transitions, which are provided for supporting expansions in the lattice, as shown in figure 2b.

The automaton properly replaces the four adjacent non-filled transitions according to the data information collected by the robot's sensors while it performs the exploring moves. The information collected by the sensors contains indications on the direction – north, south, east or west – and the condition of the place – free or busy.

Figure 2c shows one possible example of the four-data information collected by sensors and stored by the automaton. Double arrows indicate non-obstructed areas and bold lines denote obstructed ways.

In order to exemplify a complete map, figure 3a illustrates the representation of the information acquired by the automaton after exploring a simple room. The dot-marked state shows that the robot completed the exploration at the rightmost upper space of the room.

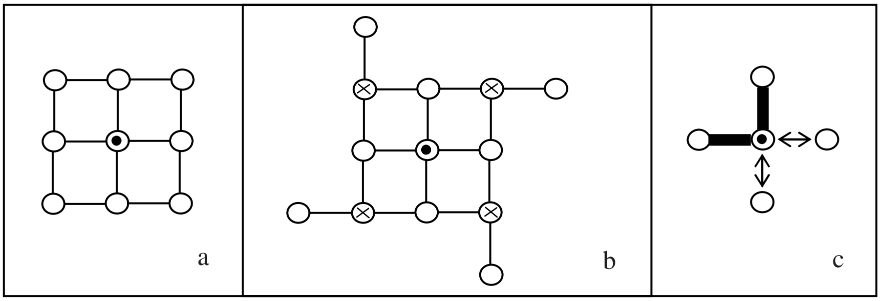

Fig. 2. (a) Initial lattice in an adaptive automaton. (b) Special tags and transitions in initial automaton. (c) Example of information coming from the sensors: two directions obstructed and two free directions

To keep the relation between the part already built of the map and the real environment, the initial state of the automaton is adopted as representing the origin of the map. This state corresponds to the initial mapping place. So, any point in the map is associated to any point in the physical environment through the association of each transition in the map's representation to some corresponding displacement performed by the robot in the real world.

Note that in actual applications the automaton is represented in the algebraic way (as described in section 2.1).

This mapping process allows dynamic memory occupation usage according to the amount of already mapped area. This feature contrasts with classic approaches, such as those described in [4], [6], [10].

3 The Model

The proposed model to perform robotic mapping and exploring motion by using adaptive automata is depicted in figure 4.

In this proposal, an information management system supplies the exploring-motion automaton with data collected from the sensors, and the current neighborhood information previously modeled in the mapping automaton. Data collected by the sensors contain information on the direction (north, south, east or west) and condition (occupied or free). Data from the map contain information on the two adjacent states in the four directions and their condition (free, occupied, not mapped or not created state). Figure 3b illustrates an example of information from the map representing the two south states in an occupied condition and all other states in a free condition.

The output information generated by this automaton indicates in which direction the robot is going to move. According to the configuration of the environment, the exploring-motion automaton also presents in its output a landmark information for future assistance to the navigation.

The management system supplies the mapping automaton with information collected from sensors followed by the direction information and the landmark – in case of occurrence. The mapping subsystem is responsible to store all the sensor information on the presence or absence of obstacles close to the robot and the navigation auxiliary landmarks. The motion decision is also transferred to the motion system that controls the robot's motors.

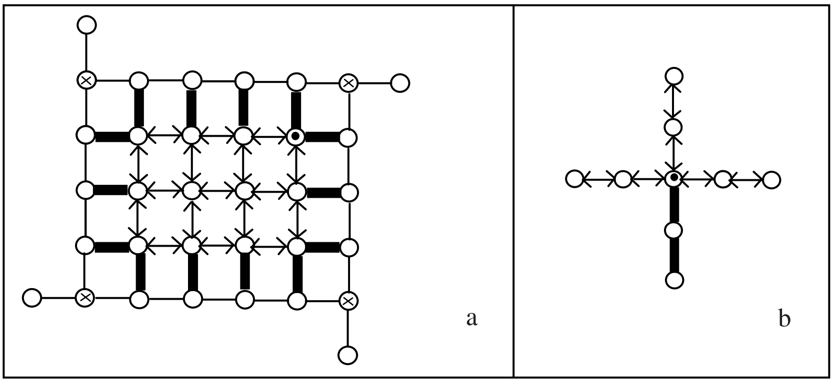

Fig. 3. (a) Example of a simple room mapped. (b) Information extracted from the map

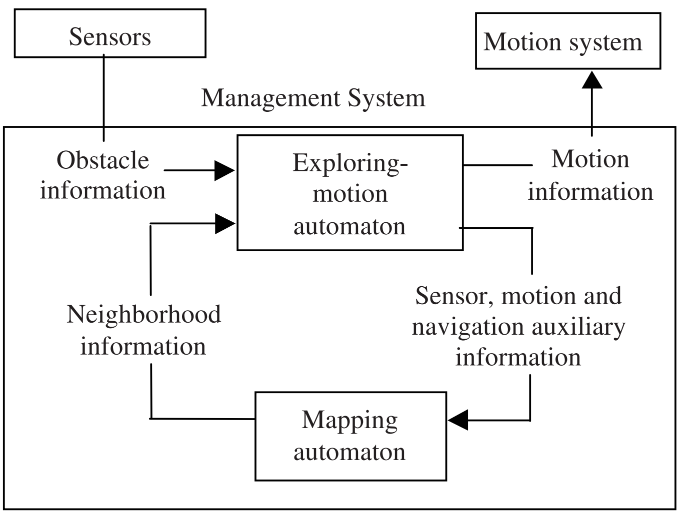

Fig. 4. System model

Note that the environment, sensors and motions have been simulated in order to validate the proposed map building mechanism.

4 The Exploring-Motion Automaton

As described in section 4, an adaptive automaton is used for determining the robot's next move. For this purpose, it is supplied with information collected by the sensors and with neighborhood information previously registered in the map. Its operation allows the robot to cover the whole environment by describing a zigzag path. This section illustrates some details of this procedure through the following example:

a. Starting at any point of the environment, the automaton leads the robot to north direction until it finds an obstacle (Figure 6a).
b. The exploring-motion automaton conducts the robot from north to south until it finds an obstacle. Then, it turns around and comes back in a parallel path, describing a zig-zag, which grows up to east (Figure 6b).
c. After each step towards east and before the robot comes back in the parallel path, the automaton searches for a free space "behind" the robot, filling this space (in case of occurrence), and then returning to its zig-zag path (Figure 6c).
d. When this east-growing zig-zag path is exhausted, the robot returns backwards by the same trajectory searching for a sequence of "free – not mapped" adjacent states at east or west (for this purpose the automaton uses information extracted from the map as shown in figure 3b). Such sequence means that there is a non-mapped space in that this direction and the way is free to reach it. Note that such space may be a simple place or another environment as complex as the already mapped one (Figure 6d).
e. When such new space is located at the eastern side of the robot, it explores this space by performing the zig-zag path, as described in step 'b' (Figure 6e).
f. When such new space is located at the western side of the robot, it explores this space by performing the zig-zag path, but grows to the west only after assuring that the sequence of "free – not mapped" adjacent states still exist at this direction. A sequence of "free – free" adjacent states means that this space has already been mapped and the automaton leads the robot backwards, as described in step 'd' (Figure 6f).
g. When the robot returns to its initial point of exploration, the automaton leads it to the same sequence described above, changing 'east' to 'west', i. e., the zig-zag path grows to west until exhausted, and displacements to the east are conditioned to occurrence of a sequence of "free – non-mapped" adjacent states at this direction (Figure 6g).
h. When the robot returns again to its initial point of exploration, the automaton signs that the environment is entirely explored and full (Figure 6h).

The zig-zag path has been chosen by its generalist features and because the environment is completely unknown. Some approaches divide the environment in rectangles during the exploration. In fact, a zig-zag path may be interpreted as a rectangle with unitary side and whose inside is completely known, which is an important feature in the mapping process.

When a new space is found during the exploration it is entirely explored before proceeding. Such a depth search has been chosen because breadth search implies increasing the actual motion of the robot, then, exploring an entire branch before moving to another one is usually cheaper.

While the environment is explored, the exploring-motion automaton may sign to the mapping automaton some special states, or landmarks, which are properly marked on the map. During the navigation process such landmarks are helpful for plan a trajectory from some initial position to a target position. The system calculates the path between such landmarks and, during navigation, it must find which landmarks are nearest to the initial and to the target positions [12], [13]. Then, those landmarks may be viewed as sub-goals in the navigation process. In order to implement such sub-goals, the exploring-motion automaton searches obstacles to the east or to the west during the zig-zag. If an obstacle is detected (a wall for instance) the proposed automaton marks the central state on the free space before and/or after the wall and, during the return trail, it signs to the mapping automaton that this state is a sub-goal (Figure 6i).

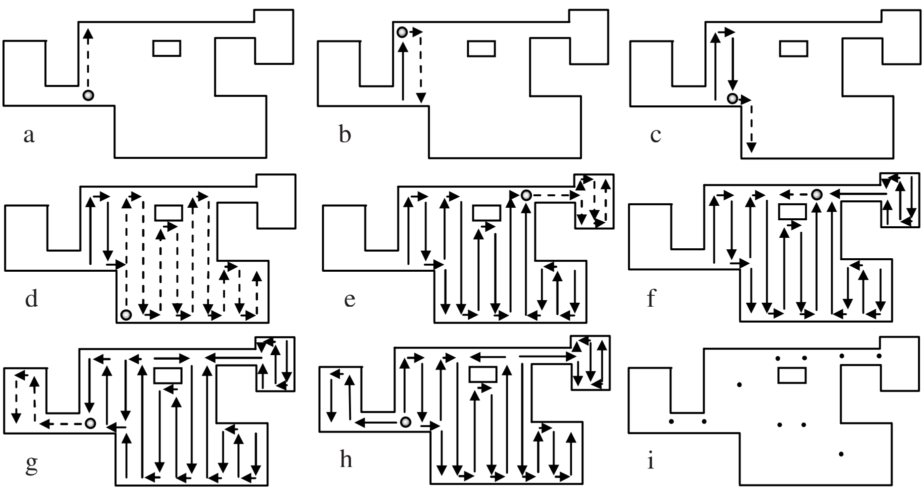

Fig. 6. (a) Initial north move. (b) East-growing zig-zag path. (c) Space filling before complete zig-zag path. (d) Exhausted east-growing zig-zag path. (e) East new space found during the return move. (f) West new space found during the return move. (g) Exhausted west-growing zig-zag path. (h) Environment entire explored. (i) Landmarks defined for the environment-example

The present proposal allows the robot to cover all environment despite to its complexity and to start at any point of the space to be mapped. These features are advantages if compared to other approaches (such as presented in [11]).

4.1 Description of the Automaton

For short, the description of the motion of the exploring automaton is represented in table 2, which expresses the relation between the system's situations of exploration (directions) and the incoming sensor information and the information encoded in the map. Table 1 shows the encoding of table 2.

Table 1. Encoding of table 2

Action	Move to:	Direction	Action	Move to:	Direction
A	North	1.	J	north	6
B	south	2	K	west	7
C	east	3	L	south	8
E	final of exploration	-	M	west	9
F	movement of return	10	N	south	3
G	movement of return	11	O	north	5
H	north	4	Q	north	7
I	east	5	R	south	9

Direction 1: Initial north Direction 2: South east-growing Direction 3: South complementary east-growing Direction 4: North east-growing Direction 5: North complementary east-growing Direction 6: North west-growing Direction 7: North complementary west-growing Direction 8: South west-growing Direction 9: South complementary west-growing Direction 10: Return from east Direction 11: Return from west	N: north S: south E: east W: west Ac: action ↑: free ∅: occupied

Table 2. Situations of exploration versus the incoming information from sensors and map

Direc-tions	**Enlargements allowed to**																			
	east and to west					**east**					**west**					**none**				
	N	**S**	**E**	**W**	**Ac**	**N**	**S**	**E**	**W**	**Ac**	**N**	**S**	**E**	**W**	**Ac**	**N**	**S**	**E**	**W**	**Ac**
1	↑				**A**	↑				**A**	↑				**A**	↑				**A**
	∅	↑			**B**	∅	↑			**B**	∅	↑			**B**	∅	↑			**B**
	∅	∅	↑		**C**	∅	∅	↑		**C**	∅	∅	↑		**C**	∅	∅	↑		**C**
	∅	∅	∅	↑	**K**	∅	∅	∅	↑	**K**	∅	∅	∅	↑	**K**	∅	∅	∅	↑	**K**
	∅	∅	∅	∅	**E**	∅	∅	∅	∅	**E**	∅	∅	∅	∅	**E**	∅	∅	∅	∅	**E**
2		↑			**B**		↑			**B**		↑			**B**		↑			**B**
		∅	↑		**C**		∅	↑		**C**		∅		↑	**M**		∅			**F**
		∅	∅	↑	**M**		∅	∅		**F**		∅		∅	**F**					
		∅	∅	∅	**F**															

Table 2 (***continuation***)

Directions	Enlargements allowed to: east and to west					east					west					none				
	N	**S**	**E**	**W**	**Ac**	**N**	**S**	**E**	**W**	**Ac**	**N**	**S**	**E**	**W**	**Ac**	**N**	**S**	**E**	**W**	**Ac**
3		↑			**N**		↑			**N**		↑			**N**		↑			**N**
	↑	∅			**H**	↑	∅			**H**	↑	∅			**H**	↑	∅			**H**
	∅	∅	↑		**C**	∅	∅	↑		**C**	∅	∅			**F**	∅	∅			**F**
	∅	∅	∅		**F**	∅	∅	∅		**F**										
4	↑				**H**	↑				**H**	↑				**H**	↑				**H**
	∅		↑		**I**	∅		↑		**I**	∅			↑	**K**	∅				**F**
	∅		∅	↑	**K**	∅		∅		**F**	∅			∅	**F**					
	∅		∅	∅	**F**															
5	↑				**O**	↑				**O**	↑				**O**	↑				**O**
	∅	↑			**B**	∅	↑			**B**	∅	↑			**B**	∅	↑			**B**
	∅	∅	↑		**I**	∅	∅	↑		**I**	∅	∅			**F**	∅	∅			**F**
	∅	∅	∅		**F**	∅	∅	∅		**F**										
6	↑				**J**	↑				**J**	↑				**J**	↑				**J**
	∅			↑	**K**	∅		↑		**I**	∅			↑	**K**	∅				**G**
	∅		↑	∅	**I**	∅		∅		**G**	∅			∅	**G**					
	∅		∅	∅	**G**															
7	↑				**Q**	↑				**Q**	↑				**Q**	↑				**O**
	∅	↑			**L**	∅	↑			**L**	∅	↑			**L**	∅	↑			**B**
	∅	∅		↑	**K**	∅	∅			**F**	∅	∅		↑	**K**	∅	∅			**G**
	∅	∅		∅	**G**						∅	∅		∅	**G**					
8		↑			**L**		↑			**L**		↑			**L**		↑			**L**
		∅		↑	**M**		∅	↑		**C**		∅		↑	**M**		∅			**G**
		∅	↑	∅	**C**		∅	∅		**G**		∅		∅	**G**					
		∅	∅	∅	**G**															
9		↑			**R**		↑			**R**		↑			**R**		↑			**R**
	↑	∅			**J**	↑	∅			**J**	↑	∅			**J**	↑	∅			**J**
	∅	∅		↑	**M**	∅	∅			**G**	∅	∅		↑	**M**	∅	∅			**G**
	∅	∅		∅	**G**						∅	∅		∅	**G**					
10			↑		**C**			↑		**C**				↑	**K**					**F**
			∅	↑	**K**			∅		**F**				∅	**F**					
			∅	∅	**F**															
11				↑	**C**			↑		**C**				↑	**K**					**G**
			↑	∅	**K**			∅		**G**				∅	**G**					
			∅	∅	**G**															

5 Conclusion and Future Work

Robotic mapping is an essential feature to allow robots to complete certain tasks in unstructured and unknown environments. This work has shown an alternative to the classic mapping approaches: adaptive algorithms provide a new way to build maps and conduct the robot for unknown environments, covering all space. During the exploration, sensors attached to the robot scan the environment for the presence or absence of close obstacles, and such information is collected into the model by enabling the automaton to perform appropriate self-modifications. The exploring-motion automaton also provides special landmarks to the map, which may be used as subgoals on further navigation. The present proposal has the advantage of allowing the robot to explore complex environments without a priori knowledge of the place and the advantage of memory space usage increasing with the actually mapped area.

Future works should approach the navigation problem by using the landmarks and the map created.

References

1. Almeida Jr., J. R.; Neto, J. J. "Using Adaptive Models for Systems Description, Applied Modelling and Simulation", International Conference on Applied Modelling and Simulation, Cairns, Australia, 1999.
2. Brooks, R. A. "Visual Map Making for a Mobile Robot", Proceedings IEEE Int. Conference Robotics and Automation., St. Louis, MO, 1985, pp. 824–829.
3. Costa, E. R.; Hirakawa, A. R.; Neto, J. J. "An Adaptive Alternative for Syntactic Pattern Recognition", Proceeding of 3rd Int. Symposium on Robotics and Automation, ISRA 2002, Toluca, Mexico, 2002, pp. 409-413.
4. Donald, B. R.; Howell J. "Practical Mobile Robot Self-Localization", Proceedings International Conference on Robotics and Automation, San Francisco, CA, 2000.
5. Hirakawa, A. R; Júnior, J. R. A.; Neto, J. J. "Adaptive Automata for Independent Autonomous Navigation in Unknown Environment", International Conference on Applied Simulation and Modelling, Banff, Alberta, 2000.
6. Jennings, J.; Kirkwood-Watts, C.; Tanis, C. "Distributed Map-Making and Navigation in Dynamic Environments", In Proceedings of the International Conference on Intelligent Robots and Systems IROS'98, Victoria, Canada, 1998.
7. Neto, J. J. "Adaptive Automata for Context-Dependent Languages", ACM SIGPLAN Notices, 1994, v.29, n.9, pp.115-24.
8. Sukhatme, G. S.; Wolf, D. F. "Towards Mapping Dynamic Environments", Proceedings of the International Conference on Advanced Robotics, 2003, pp. 594-600.
9. Thrun, S. "Robotic Mapping: A Survey", Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, 2002.
10. Zimmer, U. R. "Embedding local map patches in a globally consistent topological map", Proceedings of the International Symposium on Underwater Technology, Tokyo, Japan, 2000, pp.301-305.
11. Yang, S.; Luo, C.; Meng, M. "Entire Region Filling in Indoor Environments using Neural Networks", Proceedings of the 4° World Congress on Intelligent Control and Automation, Shangai, China, 2002, pp.2039-2044.
12. Ma, C.; Liu, L.; Li, W. "Mobile Robot Motion by Integration of Low-Level Behavior Control and High-Level Global Planning", Proc. of the IEEE Int. Conference on System, Man and Cybernetics, China, 1996, pp.310-315.
13. Innocenti, C. et al. "Trajectory planning and real-time control of an autonomous mobile robot equipped with vision and ultrasonic sensors", Proc. of the Int. Conference on Intelligent Robots and Systems IROS'94, Munich, September 12-16, 1994, pp.1861-1866.