Fusion of Sensor Data in a Dynamic Representation

Maurizio Piaggio, Antonio Sgorbissa, Gianni Vercelli, Renato Zaccaria

DIST - University of Genoa, via Opera Pia 13, Genoa, Italy
e-mail: piaggio@dist.unige.it

Abstract

The task of moving a robot in a complex and totally or partially unknown environment involves the construction of a suitable representation of the robot's surroundings. This paper proposes an analogical statistical representation of the world the robot moves in. Its construction, updating and usage is described, focusing on the techniques used for the integration of sensor data and on their fusion with a priori information about the environment. An additional mechanism that allows the map to be efficiently transmitted to a remote workstation, connected using a radio link, has also been envisaged. Thus the computational load of map related processes can be distributed, taking advantage of the remote station's computational power. The experimental setup and some results obtained are also illustrated.

1. Introduction

The task of moving a robot in a complex and totally or partially unknown environment involves the construction of a suitable representation of the robot's surroundings. This paper proposes an analogical statistical *map* of the world the robot moves in based on proximity ultrasonic sensor readings. It focuses on the perceptive processes that deal with its construction, updating and usage, and, in particular, on the techniques used for the integration of sensor data and the fusion with a priori information about the environment.

The map and its related processes are integrated in a complete robotic system for a mobile robot [1], basing on a hybrid cognitive model [2]. The system is structured in two main units: the *symbolic component* and the *procedural component*. The former contains the declarative representations that constitute the "high level" knowledge of the system. The latter encompasses both the execution of reactive and partially reactive behaviors, and the processing of iconic, analogical, representations [3] in the tradition of mental imagery [4]. In particular it is this part of the robotic system that is responsible of perception and world representation. Its general structure is shown in figure 1. It is organized as a set of concurrent procedures called *experts*. In the figure, different types of experts are represented as ellipses. Ellipses labeled with A are actuators, ellipses labeled with R are recognisers. The iconic representations are managed by the AIM (Active Isomorphic Memory), which is an analogical, image based, representation of the environment.

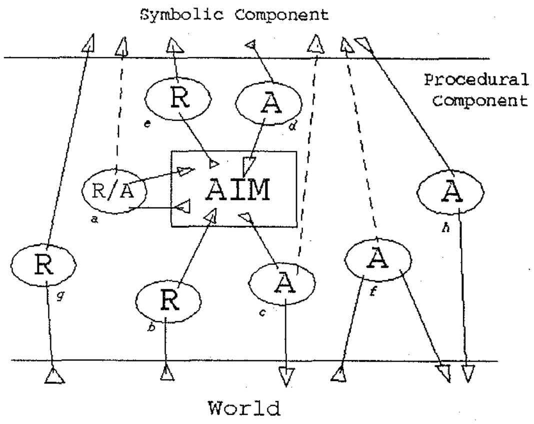

Figure 1. Procedural Component. A - Actuator R - Recogniser: a (determine dynamic evolution of AIM), b-c (connect AIM to real-world), d-e (connect AIM to symbolic component), f (purely reactive), g-h (connect sensing and acting directly to symbolic).

This paper limits its scope to the experts involved in perception and representation (see [2] for a more general description of the component and of the overall model). The next section presents the analogical world map contained in the AIM and the experts of type b that are responsible for sensor data acquisition and fusion. The third section describes the experts of type d-e that

0-8186-7695-7/96 $5.00 © 1996 IEEE
Proceedings of EUROBOT '96

respectively obtain a priori information about the world from the symbolic component and return eventual changes perceived. In the fourth section we present our experimental set-up and provide preliminary results. Conclusions follow.

2. Sensor Data Representation

2.1. A Statistical Dynamic Map

The analogical representation we propose consists in a statistical dynamic map, thus capable of storing information that is more detailed than simple on-off values to determine the presence of an obstacle. This choice is in part dictated by the usage of proximity ultrasonic sensors that are subject to different errors due to the type of reflecting surface of the obstacle and to sensor noise [5]. Consequently a single sensor reading can frequently be misleading and it is certainly unreliable to be used for autonomous navigation or exploration. Thus, the map is not directly based on the most recent readings but it statistically records the number of times an obstacle has been detected in a given position.

These values can be grouped as shown in figure 2 to identify a region of reliability (probable presence of an obstacle) and another of unreliability (probable free space). The boundary between the regions is obviously dynamically variable depending on the robot's motivations and status. If it "feels" confident or it is in a hurry the boundary would be placed closer to the baseline; on the other hand if it feels insecure about its surrounding the boundary would be moved up, reducing the region of reliability.

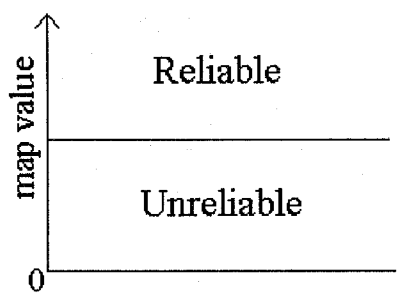

Figure 2. Significance of statistical measure

The map is a bi-dimensional array of bytes. The array is an ecocentric representation of a part of the world in which the robot moves, appropriately scaled with a fixed scaling ratio. The ratio establishes the granularity of the sensed environment and, together with the size of the array, determines the entire area that the robot can consider at a given instant.

Periodically, each proximity sensor returns the distance to the nearest obstacle with respect to the robot position (up to dynamically variable threshold distance) . It is worth noting that, because the map's description is ecocentric, the system needs to know the robot's absolute position in the world every time it has to update the representation with new sensor readings. Even though obtaining the robot's position is often a difficult task (it is here solved with DLPS©, a novel localization system based on active beacons [6]), this solution permits to integrate new sensor data with previous world descriptions without high cost map roto-translations.

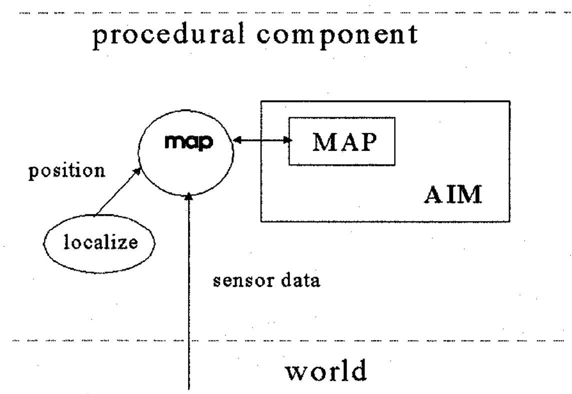

Figure 3. Procedural component and Map expert

In figure 3 the part of the procedural component that is related to the perceptive processes is shown (confront figure 1 to figure 3). There are two main experts: *localize* and *map*, both of type *b*. The localize expert is responsible of robot localization in the environment using both odometry and the active beacons system DLPS©. This information, together with proximity sensor readings, is requested by the map expert that deals with data acquisition, interpretation and fusion in the map. The map itself is contained in the AIM. It is worth noting that this map is not the only representation in the AIM. The memory is capable of handling many other additional analogical representations depending on the task to be carried out. For example, for robot navigation, an abstract potential field [7] might be used.

2.2. Fusion of Sensor Data

The proximity ultrasonic sensor readings combined with the robot's absolute position and orientation in the environment identify the coordinates in the map of the occupied space. Figure 4 illustrates this relationship: (*X-Robot*, *Y-Robot*) indicates the robot's position, *O-Robot* its orientation, (*X-Sensor*, *Y-Sensor*) the robot-relative coordinates of a proximity sensor, *O-Sensor*, the robot-relative sensor orientation and, finally, *Dist* the distance to the obstacle measured.

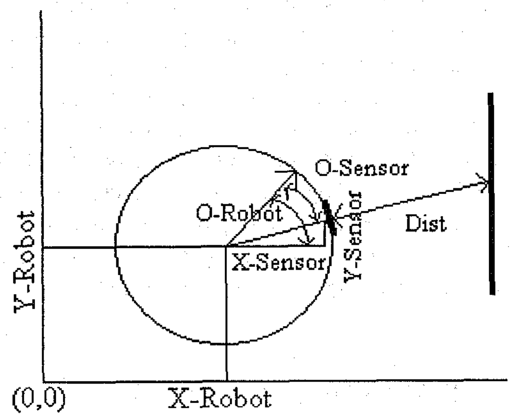

Figure 4. World-Robot-Obstacle position relationships.

An element of the map array, corresponding to a portion or *granule* of the environment , consists of a byte value. For each sensor reading indicating the presence of an obstacle, the occupied granules are calculated and their byte value is increased using the function (1). Other non-linear functions can be envisaged if different increasing ratios depending on the statistical value want to be implemented.

$$f(x) = \begin{cases} x+1 & 0 \le x < thresh \\ thresh & otherwise \end{cases} \quad (1)$$

The threshold value in function (1) can be dynamically adapted during the robot's motion. It captures the degree of confidence the robot has on its sensorial information and
on the persistency of the world. A low threshold is used in a static, reliable environment with accurate proximity sensors. A high threshold is used in a highly dynamic world or if the sensors used are not very precise. In our experiments the value does not appear to be critical, thus guaranteeing a robust robot behavior.

2.3. Handling a Dynamic World

In a dynamic world, obstacles present at a given instant might be moved, and thus disappear at a later time. Clearly, the technique described above does not handle similar situations. In addition, spurious data can accumulate over time and eventually reach the threshold established. Consequently an additional mechanism called 'map refreshment' is also implemented. In analogy to real proximity sensors, it consists in the presence of virtual sensors of "skepticism". These sensors periodically apply a decreasing function (2) to the portion of the map they are related to. The resulting effect is both the removal of obstacles that have a very low persistency in the environment, whether because they were formed from noise in sensor readings, multiple reflections or moving obstacles that are no longer present.

$$d(x) = \begin{cases} x-1 & 0 < x < thresh \\ x & otherwise \end{cases} \quad (2)$$

It must be noted that the function has an effect only on the values that are below the maximum allowed. This implies that, once detected with a certain degree of confidence, the obstacle is preserved even if it is no longer detected. If this condition were not applied, the world would vanish as soon as the robot moved away and, in case the robot returned in previous positions, the map would have to be detected from scratch

The formula is applied periodically in the region covered by the virtual sensors. Clearly, the number and placement of sensors of skepticism as well as that of proximity sensors is a relevant factor in the resulting system performance. Another important parameter is the refreshment rate. In our experiments we have obtained satisfying results when the period depends on the speed of the vehicle. Intuitively, a slow moving vehicle allows more frequent refreshments than a fast one because each obstacle is sensed a greater number of times.

The sensors of skepticism are also particularly important in the case of fusion of sensor data with a priori information. This is not an easy task because it might introduce in the map obstacles that are no longer present in the world and that may cause unpredictable robot behavior. However if we assume that a priori data is probable but not certain information we could take advantage of the probabilistic representation and insert the obstacles with a variable value (depending on how confident we are about their existence) inferior to the certainty or refreshment threshold. In this way if the obstacles are no longer present the relative granule values will periodically decrease down to the definitive object removal.

2.4. Crossing Map Boundaries

Another problem arises from the limited size of the map, related to the mobile robot host available memory size. Instead, the working environment is practically unbounded in its dimensions, especially if we the robot is to move outdoors. The problem can be partially resolved by adapting the minimum spatial resolution in order to satisfy memory requirements. However this parameter is often lower bounded in the sense that for a particular task that the robot has to carry out, a minimum resolution is normally required. Therefore, clearly, only a partial portion of the world can be represented at a given time.

If the map were centered around the robot's position, this restriction would not excessively compromise the robot's functions because it could be compared to the human being limited perceptive field. Nevertheless, as previously specified, the proposed map is ecocentric and thus an additional mechanism to avoid an undesired crossings of its boundaries with the relative loss of sensorial information has to be envisaged.

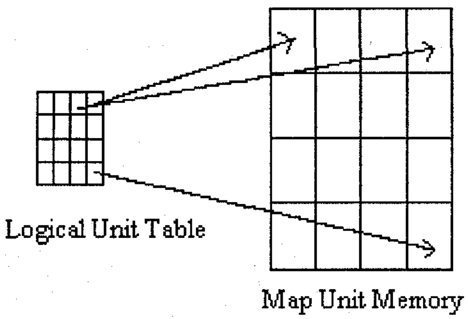

Figure 5. Map internal model. To a logical map unit corresponds a map (or memory unit). Adjacent logical units are not necessarily adjacent memory units.

The solution we propose allows the robot to keep in its host memory the regions of the world closest to its body, still maintaining an overall ecocentric description and thus avoiding inefficient map roto-translations. This is achieved by dividing the map into smaller rectangles (*map units*) each representing a physical region of the work space. To each map unit, a *logical unit* is associated; a separate *unit table* stores the relationship between the single logical units, the physical region they are related to and the map units. An example of this relationship is illustrated in figure 5. In the table, adjacent logical units correspond to adjacent places in the environment. On the contrary, physically adjacent map units in the robot's memory do not necessarily correspond to adjacent regions in the real world (or in the logical unit table).

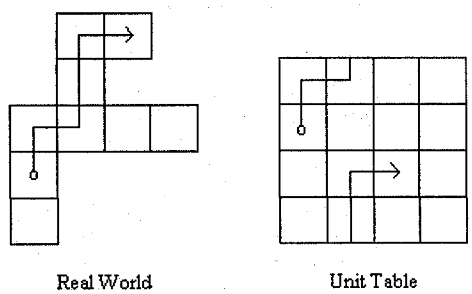

Figure 6. Example of an environment and the corresponding representation

The robot's position is associated to a logical unit and thus a map unit or physical block of memory. When, during its motion, the robot crosses the map boundaries, it 'wraps around' to the opposite logical unit in the table, which is then modified in order to represent the new region of the environment. In this way, the robot can move continuously across the map's border in exactly the same way as if it moved in the center of the map. Figure 6 illustrates this concept of out of boundary navigation.

It is worth mentioning that in the system implemented, the robot's movement is accompanied by a *zone of influence* that depends on its perceptive field. The zone indicates the physical region concerned for sensorial updating from which the corresponding logical units can be calculated. The mechanism described above is thus applied to the entire zone and not only to the single unit where the robot is placed.

2.5. Map Transmission

The map's size limitations imply that if the robot moves in a wide area a lot of sensorial data will be lost every boundary crossing. This problem may be overcome if the data acquired is transmitted to and from a remote computer that has greater storing facilities. In addition it permits to take advantage of remote high speed processing to increase the overall computational power of the mobile system. This solution is particularly advantageous using the proposed cognitive model. The symbolic component can be placed on a remote workstation whereas the procedural component can be placed on board of the mobile robot. The two sub-systems can then be connected with a communication channel such as a radio-link.

This section describes how the map can be transmitted in order to have a consistent remote copy and to limit data throughput across the channel to avoid its saturation. In this operation, the unit table described in the previous section is widely used: the robot maintains two lists: *Sending list* and *Receiving list*. These are both FIFO lists and contain elements that point to units in the unit table as shown in figure 7.

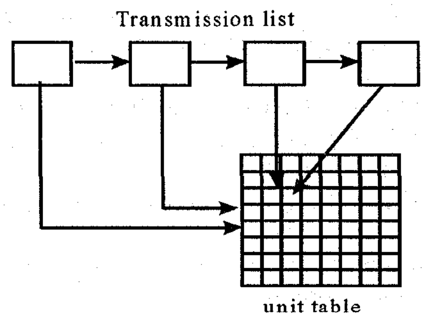

Figure 7. Transmission lists

The sending list contains the units queued that are to be sent to the remote workstation. When the robot leaves a region of map the corresponding unit is identified and is added to the sending list to inform the remote computer of the new sensorial data. If the robot has previously left the same region without the data having yet been sent, the unit would already be in the list. In this case it is left in the original position to maintain its priority. Therefore, only the units that have been updated by sensorial data are sent across the channel and thus data transmission is reduced. The receiving list contains the units that have been requested to the remote computer. This normally occurs when the robot enters in a new region for the first time and asks the symbolic component if it has a priori information that may be used. This list should normally be limited in size because if the robot has handled the situation (managed to leave the unit) without having yet received the requested data, probably this information is not really necessary and thus the corresponding unit may be removed from the list. Figure 8 presents the algorithm used.

```
<Init. Sending and Receiving List>
<when the robot leaves a unit>
      {
      if <not already present>
               <add previous unit to sending list>
      if (<unit is not present in the map> and
               <not present in the receiving list>)
               <add current unit to receiving list>
      }
```

Figure 8. Map transmission algorithm

An additional mechanism is used to reduce data throughput. Two copies of the map are kept in the robot. The former is the map that is used for sensorial information updating, the latter is instead totally consistent with the one on the remote workstation. During the robot's movement the two copies differ because of the data acquired. It can be noted that if the two copies are subtracted the amount of information to be transmitted is greatly reduced. This is the basis of the algorithm that performs in sequence the following operations:

1. the probabilistic information is binarized to determine the presence of an obstacle with a Boolean yes/no value
2. the binary units are subtracted to extract the relevant data
3. the result is sent using a Run Length Encoding compression algorithm
4. the copy of the remote map on the robot is updated with the unit sent.

Clearly, the unit table, and thus the size of the single map unit in comparison to the entire map, becomes an important parameter of the system. An excessively small value introduces memory, computation and transmission overheads. On the contrary an excessively large value reduces the effectiveness of the technique.

3. System Architecture

The general structure of the map handling expert is shown in figure 9.

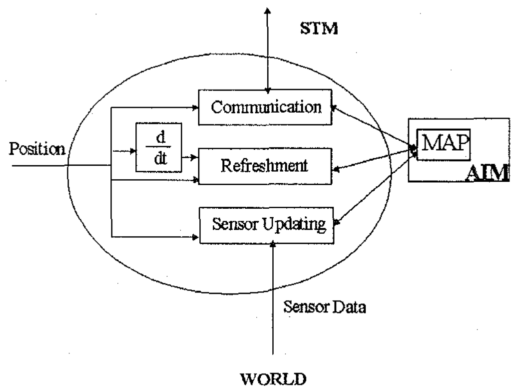

Figure 9. Map handling expert structure

The expert requires sensor data from proximity sensors and robot position from another expert responsible of robot localization. The rectangle on the bottom indicates the procedures that build and update the map with new sensorial information. As illustrated by the arrows, it requires to know the robot's absolute position in the environment. This information is also used by the communication procedures (top rectangle) and, together with the speed of the robot, the procedures that implement map refreshment (middle rectangle).

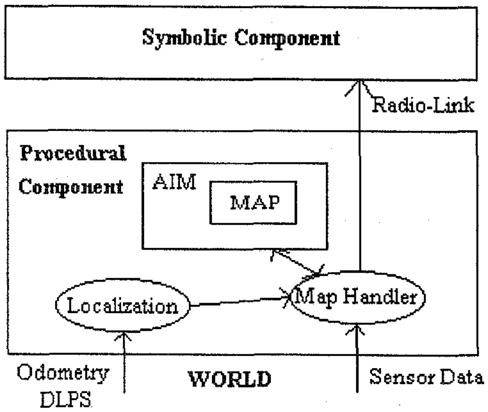

Figure 10. General Architecture

In figure 9 an arrow pointing to a Short Term Memory (STM) can be seen. This memory is part of the symbolic component where a priori world information are stored [2]. The arrow represents a special channel for map transmission: in one direction the channel allows the remote station to be informed of the environment in which the robot operates and in the other direction it permits an a priori construction of the map on the basis of symbolic short term information. The AIM clearly contains both the dynamic map and its associated unit table.

The overall architecture, shown in figure 10, allows the concurrent execution of procedural experts, asynchronously exchanging information. In particular it allows all the processes to be distributed both on the mobile robot, where the procedural component is presumably placed, and on a remote workstation where the symbolic component can take advantage of the presumably greater computational power. The transmission component of the map handling expert guarantees map consistency between the procedural and symbolic components of the system.

4. Experimental Results

The model described in the previous sections has been implemented using a TRC Labmate with an on board Intel 486DX processor and two proximity sensors belts. An analogous version which operated in a simulated environment has also been developed to facilitate the testing of new implementations. Both in simulation and in the real setup, the sensor belts contained 8 sensors each, displaced as in figure 11.

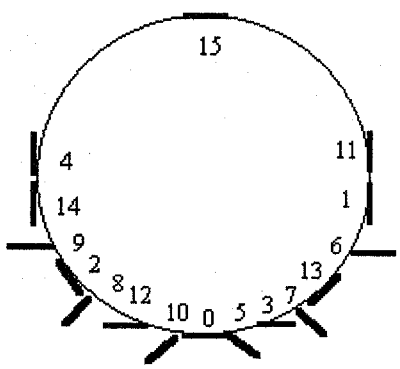

Figure 11. Proximity sensors in the mobile robot

The numbers in the figure indicate the sensor firing sequence which was calculated to maximize the sensor reading frequency. It can be noted that sensors with similar orientation and adjacent position are never fired one after the other. This avoids sensor interference when the time interval between their being fired is small. Experimentally the perceptive expert operated at a frequency of 20 Hz. Clearly, the sensors are places in order to privilege perception on the front of the robot for navigational purposes. In addition, the different sensor orientations allow a better the detection of obstacle borders.

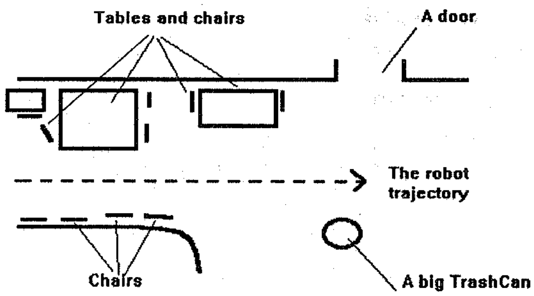

Figure 12. Real World

Figure 12 illustrates the room where the robot was situated and moved. The dotted line represents the robot's trajectory that was followed at an approximately constant medium speed. Figure 13 represents what was perceived by the moving robot.

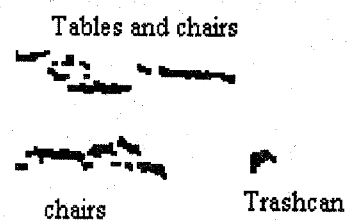

Figure 13. Sensor Acquired Map

The map has a resolution of 180 x 180 granules, covering an area of 9m x 9m (5 cm per granule). Its associated logical unit table is a 5 x 5 array. Thus, a single map unit is 36 x 36 granules (1,8 m x 1,8 m). The parameters were set to have a sensor working range similar to a unit's size. In this way map the performance of the map transmission procedures is optimized.

It must be noted that the robot's movement determines the quality of the resulting map: if the vehicle has moved following the table and chairs contours the map would have had greater details. Therefore in the case of environment exploration it would be best for the robot to move closer to the obstacles, following their borders. However, to carry out simple obstacle avoidance navigation it can be seen that the given movement allows sufficiently accurate object detection.

An example of different paths within the same room is shown in figure 14 and figure 15.

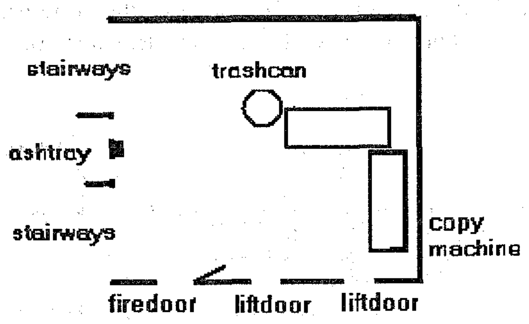

Figure 14. Real room

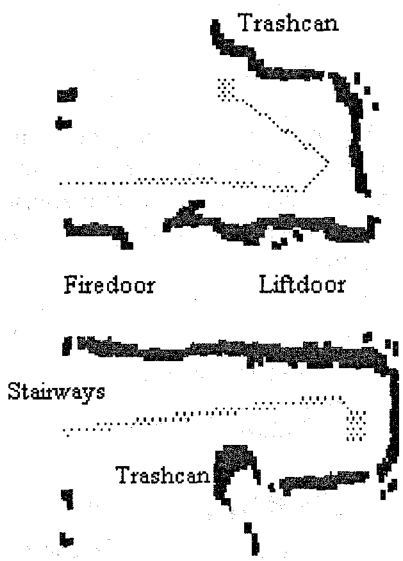

Figure 15. Different path in the real room

5. Conclusions

Although many results are available in literature about proximity sensor data acquisition [5], little efforts have been made to copy with unlimited environments, dynamic data improvement and fusion between sensing and a priori representation. In our system the dynamic map can be suitably exploited by all the cognitive components. At the higher level, recognition, classification and route planning processes [8] can be included. At a lower, reactive, level, motion planning and navigation experts can be directly connected to the local map, as in [4] [7] [8]. The current work is this field, in particular putting to work the methods described in [2]. In addition extensions of the architecture and of its related operating system, to allow the execution of completely concurrent processes and real-time processes are being developed.

References

[1] M. Piaggio et al., A hybrid robotic system for a mobile robot, Proc.

[2] M. Frixione, M. Piaggio, G. Vercelli, and R. Zaccaria, A cognitive hybrid model for autonomous navigation, Proc. 4th Congress of the Italian Association for Artificial Intelligence, AIIA 1995, Florence, Italy.

[3] L. Steels, Exploiting analogical representations, in Designing Autonomous Agents, 1990, MIT Press, Cambridge (Ma).

[4] S. Kosslyn, Image and Mind, 1980, Harvard University Press.

[5] A. Elfes, Sonar-based real-world mapping and navigation, 1986, Autonomous Robot Vehicles (233-249), Springer-Verlag.

[6] F. Giuffrida, C. Massucco, P. Morasso, G. Vercelli, and R. Zaccaria, Multi-level navigation using an active localisation system, Proc. Intl. Conf. on Intelligent Robots and Systems - IROS 1995, Pittsburg (PA), USA.

[7] O. Khatib, Real time obstacle avoidance for manipulators and mobile robots, Int. J. Robotics Res. 1986, 5 (1), 90-99.

[8] J.-C. Latombe, Robot Motion Planning, 1991, Kluwer Academic Publishers, Boston.

[9] E.Ardizzone, A.Camurri, M.Frixione, and R.Zaccaria (1993), A hybrid scheme for action representation, International Journal of Intelligent Systems, 8 (3), 371-403.

[10] R. Brooks (1990), Elephants don't play chess, in (Maes 1990).

[11] I.J. Cox and G.T. Wilfong (eds.) (1990), Autonomous Robot Vehicles, Springer, New York and Berlin.

[12] I.A. Ferguson (1992), TouringMachines: Autonomous Agents with Attitudes, Compuer Laboratory, University of Cambridge, Cambridge, UK, 1992. Technical Report 250.

[14] P. Maes (ed.) (1990), Designing Autonomous Agents, MIT Press, Cambridge (Ma).

[15] B. Nebel (1990), Reasoning and Revision in Hybrid Representation Systems. Spriger Verlag, Berlin.

[16] M.G. Slack (1993), Navigation templates: mediating qualitative guidance and quantitative control in mobile robots, IEEE Transactions on Systems, Man, and Cybernetics, 23 (2).