LITERATURE SURVEY:

COGNITIVE APPROACH TO ROBOT SPATIAL MAPPING

Ghislain Fouodji Tasse

Supervisor: Dr. K.L. Bradshaw

Department of Computer Science, Rhodes University

June 26, 2009

Abstract

One of the major issues in Robotics or more precisely in Intelligent Systems in today's research and innovation is self awareness. The world of innovation in Robotics has been developing autonomous robots with the abilities of reasoning, learning and accomplishing basic tasks. Self mapping is one of these abilities and it is very crucial to any physical agent which claims to be aware of its environment. To make this claim true, many approaches are being used whereby the most successful ones are cognitive approaches. This survey discusses these approaches together with their applications in robot mapping.

1 Introduction

Several issues should be considered while building a map in an indoor environment. One of these issues is the concurrent mapping and localization problem, which is a pivotal problem in mobile robotics. If the position of the robot is known, building a map becomes straight forward as shown by Elfes in [5]. To handle these issues, different algorithms use different approaches which are based on different hypotheses. This paper will describe these differences by presenting and describing various cognitive approaches for robot spatial mapping. To introduce these concepts nicely, the first section of the survey is dedicated to robot spatial perception. The way the robot perceives its environment determines which mapping approach should be used and how it should be applied. In the second section, we describe some of these mapping approaches. Before implementing these approaches, we think that it was important to explain the Simultaneous Localization and Mapping problem which is an important factor in mapping algorithms. More details on this problem are given in the third section of this paper. Now, with the knowledge gathered in the previous sections, we appropriately describe the subsequent mapping algorithms and their characteristics. Further in this survey, we present a sample robot on which these mapping algorithms can be tested and show how their efficiency also depends on the physical capabilities of the robot.

2 Robot Spatial Perception

2.1 Overview

As artificial sensors and organic sensors are different, the robot's perception and human's perception of the same environment are also different. Regardless to this difference, a robot by definition is expected to interact with the world in the same way that a human does. This is only true if a robot spatial representation carries the same information as a human spatial representation. To satisfy this criterion, two notions of space representation are defined and considered: the notion of a representation for the local space i.e. the small area of the environment the individual is currently in, versus a global representation in which the individual total experience of its spatial environment can be represented using a single coordinate system. Related to this contrast, is the contrast of a metric representation, where properties such as distance, size and location are explicitly or implicitly represented, versus a topological representation where relationships such as connectivity between individual elements are represented [11, p.2]. More details on these representations are given in the following subsections.

2.2 Metric Representation

A metric map is the capture of the geometry properties of the environment [17, 18]. By geometric properties, we refer to the geometric relations between the objects and a fixed frame of reference defined in the map. This is perhaps the most explicit map in robotics since it explicitly represents the occupancy of space by storing the exact position of objects in a global frame. For example, a tourist's scale of a map is a metric map [4, p.213]. So, a metric map only reproduces the spatial state of an environment, which carries no functional information (see Fig. 1).

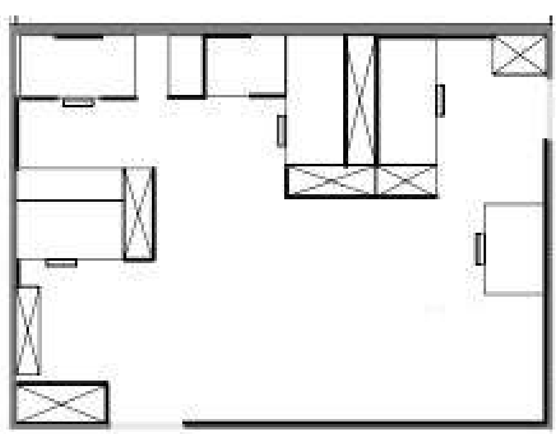

Figure 1: An office of the institute and the lines representing it in the local metric map [18].

It is commonly used to represent the environment as a two dimensional space in which it places the objects. With such a map, the object's information available to a robot is the object's precise coordinates [x,y,theta], where the pair (x,y) gives the object's position and theta its orientation [8, p43]. Hence, the precision of the information given by a metric map depends widely on the quality of sensors. As sensors are often subject to noise, this dependance is a weakness for the metric map. Additionnally, this weakness becomes more relevant as the map increases in size.

2.3 Topological Representation

A topological map naturally captures qualitative and relational information from the environment. For example, a subway map is a topological map [4, p.213]. It represents the environment as a list of significant places connected via arcs. The latter usually carry information on how a robot can travel from one place to another. Unlike the metric map which is an absolute representation, a topological map represents objects relative to each other. Hence, it is can be simulated by a graph whereby the places correspond to nodes and arcs correspond to paths which connect two nodes if they are adjacent in the real environment [13, p.597]. Figure 2 shows an example of a topological framework that only considers places and the relations between them.

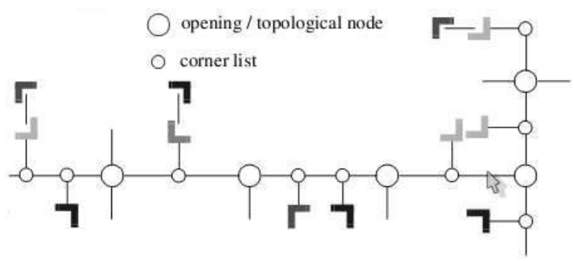

Figure 2: The topological map is represented by a graph. It contains nodes connected to each other with the list of corner features lying between them [18].

However, using topological representations, one cannot easily determine a previously visited part of the environment if it is approached from a different side [11, p.1]. Effectively, each place is only identified by the arc that leads to it. So if the arc or path used by the robot is not shown on the map, the place will be marked as unknown.

2.4 Hybrid Representation

From the explanation giving above, metric and topological maps are different representations of the same environment. Since each focuses on different aspects of the environment, they can be combined to give a *robust* map which contains both qualitative and quantitative information of that environment [19]. The idea is to enrich the topological map with metric information. This is achieved by representing each node (place) in a topological framework with a local metric map as shown in Fig. 3.

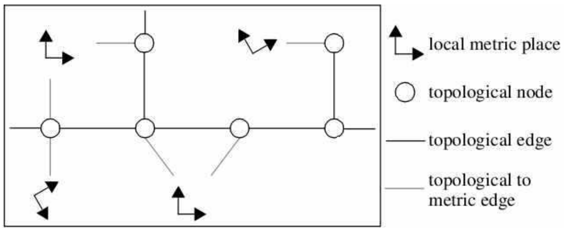

Figure 3: The environment is represented by places given by their metric maps and nodes representing topological locations.

So to move from one place to another, a robot moves metrically in that place and then outside the place, it moves topologically till it reaches the goal place where it switches back to the metric map[18]. Since metric maps only represent local spaces, this method contributes to the reduction of the impact of noise on the map. This is the description of a hybrid representation.

2.5 Mapping Problems

While aiming at representing the environment in the best way, mapping approaches have some additional issues that should seriously be considered. These issues are difficult to handle since they depend on the nature of an environment which appears to be unknown.

2.5.1 Odometry Errors

Robot sensors are naturally limited in what they can perceive. These limitations are either induced by the sensor itself or by the nature of the environment. The range and resolution of a sensor is subject to physical limitations. For example, the resolution of a camera image is limited. Sensors are also subject to noise, which perturbs sensor measurements in unpredictable ways and hence limits the information that can be extracted. *Noise* is a global term which is used to describe wheel slippage or surface imperfections.

Additionally, odometry errors can be caused by robotic software. Models are just abstractions of the real world [2]. Uncertainty in the map data can be created through algorithmic approximations. Furthermore, robots are real-time systems and can only carry out limited computational operations. Because of this, many popular algorithms are approximate, and achieve timely response by sacrificing accuracy.

2.5.2 Dynamic Environments

Most mapping algorithms assume that the environment is static. This assumption reduces the computational complexity of the algorithm by eliminating some variables that should have been considered. But these algorithms are most likely to fail in the real word which has a continuously changing environment. To overcome this problem, most mapping algorithms consider and handle places with high dynamics as noise [6].

However, the resulting map is still static and the dynamics of the environment is ignored. As a better solution to the problem, Nikos and Costas in [12] propose a filtering algorithm: the Temporal Occupancy Grid Algorithm. This algorithm classified objects into three categories: static objects (e.g. walls, bed), objects with low dynamics (e.g. chairs, doors) and objects with high dynamics (e.g. humans). In so doing, the algorithm can model the dynamics of the environment.

3 Simultaneous Localization and Mapping (SLAM)

When a robot is placed in an unknown environment, it discovers its environment whilst navigating there in. So the map is continuously updated whilst the robot gradually moves through the environment. Figure 4 is a flow chart demonstration of the SLAM process.

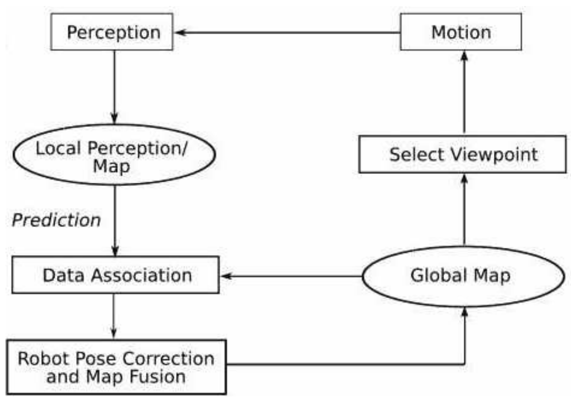

Figure 4: Flowchart of the SLAM process [8].

But, SLAM posed a serious difficulty in mobile robots. The SLAM problem asks if it is possible for an autonomous robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously using this map to compute its location [3]. Effectively, it is practically impossible to localize an object without a map or raw knowledge of the environment. It is like looking for a point in a graph without the graph itself. Conversely, it is difficult for a robot to build a map if it has no knowledge of its position in the environment, which is similar to building a graph without an origin.

Therefore, both localization and mapping should be implemented simultaneously. For mapping, it is sufficient to localize the initial position of the robot which will be the "origin or starting point" of the map. Since we are in an unknown environment, various cognitive mapping approaches define their starting point in different ways. But generally, to solve the SLAM problem, landmark based approaches are used in building the map.

4 Cognitive Mapping

4.1 Overview

Human and animals brains have very intelligent methods for building their own knowledge or representation of their environment. Using these methods, they end up with a relatively robust map which they can confidently use during navigation. We define a “robust map” as an “intelligent” map in which the capabilities of reasoning, self awareness and adaptation to dynamic environments have been implemented. A common expression for referring to a “robust map” is: cognitive map. What is of importance here is not the resulting maps but the processes used in their construction. This section focuses on explaining diverse and successful mapping methodologies as proposed by researchers.

4.2 Vision based Mapping

Many researchers are currently working on the robot spatial mapping problem using this approach. A vision based map provides the robot with a wide range of knowledge of its environment. In this case, the resulting map is not a representation or interpretation of the environment; it is a snapshot of the environment. This approach maps the space as it is, using stereo cameras as range sensors. Vision-based approaches combined with stable natural landmarks in unmodified environments are highly desirable for a wide range of applications [14]. In this approach, three factors are taken into consideration: the 3D position of the robot, its camera position, and the view of the environment relative to these positions (see Fig. 5). So, the picture of an object represents the actual object and the positions of the robot and camera are used as the coordinates of the object.

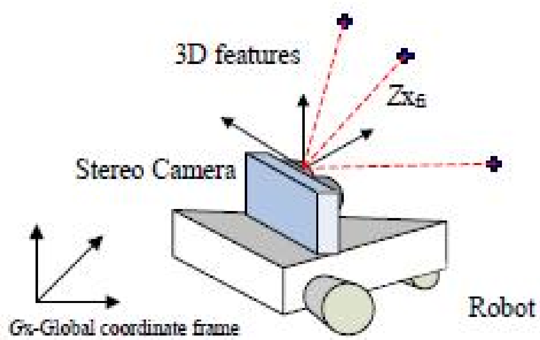

Figure 5: Coordinate system for a Vision Based Mapping.

During navigation, the robot uses these data to reconstruct the 3D environment [15]. So the robot is not only aware of obstacles,but it also knows the nature of the obstacles. The efficiency of a vision based approach relies on the filter it uses for identifying objects. A good vision-based filtering method is the Bayesian filtering method which uses a sampling-based density representation [14]. From a visual map, it localizes objects using a scalar brightness measurement. The visual map can be defined as a set of robot poses which are images captured by the robot in specific positions. A

practical description of this definition is given in [7].
Nevertheless, this approach faces a wide range of difficulties. Images require more space; take more time due to rendering which increases the computational complexity of the approach. Also, camera sensors are more subject to noise (lighting problems) than common range sensors.

4.3 Shape based Mapping

4.3.1 Overview

This approach studies the geometrical configuration of the environment to be mapped. We believe that if one can get enough information about the size and relative positions of each object in the environment, one can build a robust map. Accordingly, the shape based approach aims at representing objects in space by defining mathematical relations between them. At the end, we have a graph (the map) with a set of points (objects) and mathematical functions. The pioneer of this approach is a researcher called D. Wolter who was the first to propose a novel geometric model for robot mapping [21]. He affirmes that this approach is an improvement to bridge the gap between metric information and topological information. Figure 6 illustrates his architecture for the shape based approach.

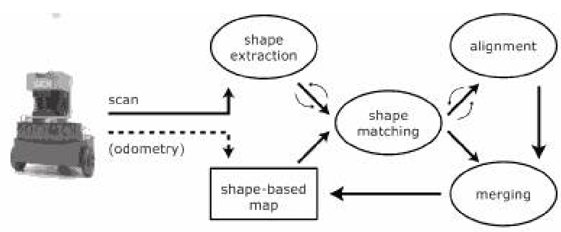

Figure 6: Shape based Architecture.

4.3.2 Methodology

In his book [20], Wolter proposes a shaped based approach which combines a boundary based approach together with a structure based approach. The structural approach represents shapes as a colour graph representing metric data alongside configurational information [21]. But this approach appears to face difficulties in identifying shapes lacking structural information. This could be due to the error in range sensors data. So he decided to consider a boundary based approach which focuses more on the boundaries of obstacles than their overall structure. Shape is represented as a structure of boundaries in which the boundaries are defined using polygonal lines (polylines). So a polygonal map is a set of polylines (which describe obstacles) and vectors of polylines (which establish the relations between two obstacles). Using this method, let us describe how information (obstacle' s shape) can be extracted from an unknown environment.
In the process of retrieving this information, D. Wloter proposes a simple heuristic: Traversing the reflection points in a (cyclic) order as measured by the LRF (Large Range Finder), an object transition is said to be present wherever two consecutive points are further apart than a given distance threshold. But this heuristic does not remove noise in the sensor readings. So we introduce a

technique called Discrete Curve Evolution (DCE) proposed by Latecki & Lakamper to first make the data more compact without losing valuable shape information and next, to cancel out noise. DCE is a process which proceeds iteratively: Irrelevant vertices get removed until no irrelevant ones remain (see Fig. 7). Though the process is context-sensitive, it depends on a vertex v and its two neighbour vertices u and w according to the following formula:

$$K(u, v, w) = |d(u, v) + d(v, w) - d(u, w)| \qquad [21]$$

If K(u,v,w) is less than the given threshold, then vertex v is removed.

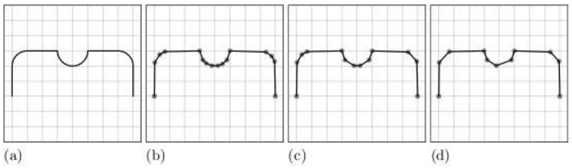

Figure 7: Illustration of the DCE technique.

With the computational processing power of actual robots, this approach seems to be very successfull.

5 Cognitive Robot Mapping

5.1 Probabilistic Algorithms

Every algorithm use in robot mapping should be able to handle uncertainty. As discussed in Section 2.5.1, sensor readings are not exact and can badly influence the robot' s perception of the environment. This problem is solved or partially solved at two levels: the physical level, which means improving the sensor accuracy, or the software level, which leads to probabilistic algorithms. Most mapping algorithms use the calculus of probability theory to improve the reliability of their resulting map [2]. Instead of relying on a single "best guess" value from sensor data, the probabilistic algorithm computes a probability distrubution over a set of sensor values [16]. So the sensor values used in the algorithm computation is one that is likely to match the reality. In the algorithm flow chart, this computation is refered to as data integrity or data validation. Before contructing the map (the final output), the algorithm sanitizes the inputs giving by the robot range sensors. As a result, a probabilistic robot can gracefully recover from errors, handle ambiguities, and integrate sensor data in a consistent way.

5.2 Grid Algorithms

5.2.1 Overview

Grid algorithms are a very simple but efficient method for spatial mapping. The grid-occupancy representation treats the world an unstructured array, with composed cells that are independently either occupied or unoccupied [8, p.137]. Depending on the algorithm, the cell can be any polygonal (see Fig. 8). This approach eases the construction of maps and spatial reasoning based on those maps. Eventually, It handles sensor data noise by estimating the size of the cells using probabilistic sensor models [5]. Occupancy grid representation gives the robot sufficient information for its navigation and path planning. A number of robotic tasks can be accomplished through operations performed on such representation. So the basic idea of an occupancy grid algorithm is to partition the space into cells where each cell is qualified with probabilistic estimates of its state (occupied or empty).

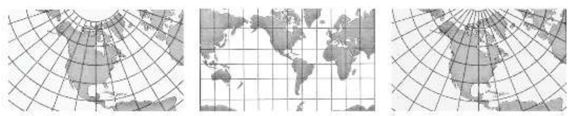

Figure 8: Different projections maintain different kinds of spatial knowledge, which leads to different forms of grid cells [9].

5.2.2 Methodology

An occupancy field is a function O(x) where x belongs to a set of continuous spatial coordinates, x=(x1, x2, ..., x3) [5]. Let us define a cell C as C=(i,j) and its occupancy state as occ(i,j) where i,j are the coordinates of the cell in the unstructured array. So, the function O takes x and maps it into the corresponding cell (i,j coordinates). We also define a probability function P which given a cell, returns the probability of its occupancy state P(occ(i,j)). occ(i,j) is a discrete random variable with two states, occupied and empty, denoted by OCC and EMP respectively. Since the cell states are exhaustive and exclusive, P[occ(i,j)=OCC] + P[occ(i,j)=EMP] = 1. A good probability function is given by the Bayes's theorem which takes as variables two sensor readings of the same cell.
As a summary, grid occupancy mapping requires two steps: the first step is to represent the space as an array of cells; the second step is to determine their occupancy state. Using a probabilistic approach to estimate the cell's state addresses the problem of generating maps from noisy and uncertain sensor measurement data is overcome. But the weakness of grid algorithms is that they are useful only when robot poses are well known.

5.3 Local Space Representation

5.3.1 Overview

The environment in which a robot is currently navigating is referred to as a local environment. In the case of an indoor environment, a room will be an example of local environment. Representing a local environment is the first step in cognitive mapping. The robot should map its current environment as it experiences it. In the contrary to the grid mapping where the position of the robot is assumed to be known, a local space representation should define its own starting point. Yeap in [42] argued that an important basis for computing a cognitive map is the ability to compute and recognise local environments [22]. So a mapping algorithm must first recognize the local environment and define a coordinate system (the starting point) for it. But the difficulty is to determine where one local environment ends and another begins [22]. Of course, the algorithm should know which space to map before starting to map it. The general problem is to find suitable connections between surfaces to form a boundary surrounding the viewer. Finding boundaries of a local environment is exclusive to finding exits in that environment.

5.3.2 Methodology

Ref. [22] proposes a cognitive approach for identifying exits which is defined as follows: "Whenever one surface is viewed as occluded by another surface, a gap exists which we label as an occluded edge. The occluded edge and the exit are thus virtual surfaces. An exit is the shortest edge covering the occluded edge." Firstly the surfaces in the current view are divided about the occluded edge FG so that F is in group I (FD, DC, BA) and G in group II (KJ, IG) (see Fig. 9). Then the exit is found by taking the occluding vertex F and connecting it to the nearest point (to F) on a surface in the group opposite to it, i.e. group II. Candidate points are H and J, but J is the closest point to F so the exit is JF. Coincidentally, both points J and F are occluding.

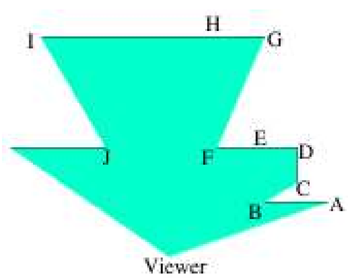

figure9: A view of the environment (in 2D) [22].

5.4 Global Space Representation

Global Space representation is the case where we not only want to represent a room, but the whole house. One can think of defining a global or single coordinate system whereby the map of each room will be computed with respect to it. But this method increases noise in the inputs. As a matter of fact, the robot has to navigate for a longer time and in a bigger space before generating a usable map. If one data is false, the whole map may be false too. This is the reason why instead of using this approach, researchers propose that a global map can be generated by building a network of local environments. In others words, each room is mapped independently and then grouped after all, to give the map of the house. In this case, our local space representation defined above is referred to as an Absolute Space Representation (ASR). ASRs are used to identify and describe local environments which have been visited by the viewer [23].

Moreover, the different local spaces which are computed can be connected together in the way they are experienced to form a topological network (see Fig. 10). So the robot perception of the environment can be defined in a topological representation, as a collection of local space representations, each with its own coordinate system, and connections between them which will allow the robot to travel from one to the other. Since this method maps the environment as the agent is navigating through it, the connections between ASRs result in how one passes from one ASR to another. In other words, the algorithm should figure out which rooms' exits match, since practically, the exit of one room is the entry of another room. This whole idea was developed by Yeap and Jefferies in [11] and is referred to as a topological network of metric local space representations.

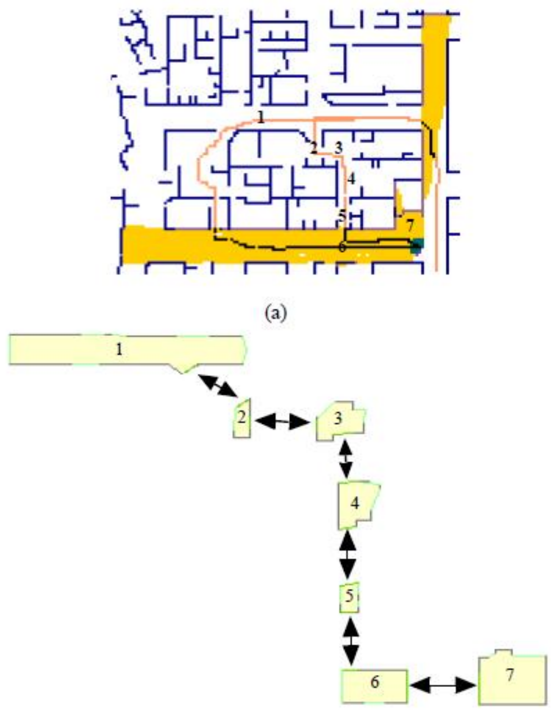

Figure 10: The topological network of ASRs computed as a simulated viewer follows the path in (a). The ASRs are numbered in the order in which they are experienced. The numbered areas in the environment in (a) correspond to the same numbered ASRs in the topological network [11].

However, this approach is incomplete when it comes to navigating using the computed map. Effectively, connections between the ASRs are defined as the robot leaves one ASR to enter another. So not all the possible connections are defined, but only those experienced by the robot. Kuipers and Byun [10] propose a solution to this by using topological matching [11]. They use the information of a current ASR to find all possible connections to it. Then the robot follows each found path to validate the connection and come back to its initial position. In so doing, the network represented in Figure 10 can be extended to the one in Figure 11.

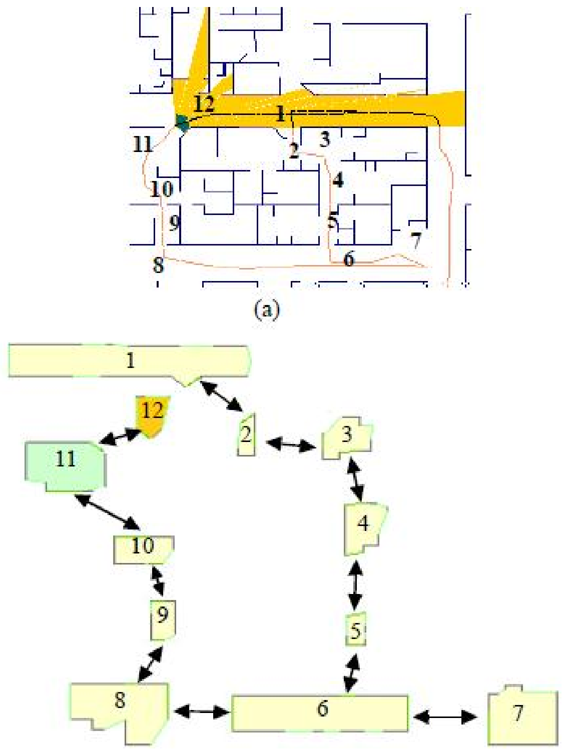

Figure 11: An extended topological network of ASRs.

6 Application: Lego Mindstroms NXT

Lego Mindstroms NXT is a mobile robot designed by Mindstroms which has the basic and sufficient properties to perform autonomous spatial mapping. It is equipped with a NXT brick which acts as a CPU in the architecture robot. These are the technical specifications of the NXT brick which were taken from the Mindstroms website [1]:

- 32-bit ARM7 microcontroller
- 256 Kbytes FLASH, 64 Kbytes RAM
- 8-bit AVR microcontroller
- 4 Kbytes FLASH, 512 Byte RAM
- Bluetooth wireless communication (Bluetooth Class II V2.0 compliant)
- USB full speed port (12 Mbit/s)
- 4 input ports (4 sensors) and 3 output ports (3 motors)
- 100 x 64 pixel LCD graphical display
- 6 AA rechargeable lithium batteries.

Two motors are used to drive the robot's wheels and the third one is an extra motor. Four types of sensors can be connected to the NXT brick: ultrasonic, touch, sound, light sensor. Our focus here is on the ultrasonic sensor, which the key sensor used by the Lego NXT for mapping.

Ultrasonic Sensor

The Ultrasonic Sensor helps the robot to judge distances and ”see” where objects are. Using the NXT Brick, the Ultrasonic Sensor is able to detect an object and measure its proximity in inches or centimetres.

7 Conclusion

In this literature review, we have provided sufficient evidence to show that if a robot is placed in an unknown environment, it will be able experience navigation without collisions by learning the spatial properties of its surroundings. Cognitive algorithms give this learning capability to the robot which makes it really autonomous. In our discussion, we sequentially improve the robot perception of its environment, a cognitive mapping approach that suits this perception and how the approach can be implemented through an algorithm. These three steps of cognitive mapping are correlated and moving one step to the other required the understanding and consideration of several issues. These issues have also been more or less presented and corresponding solutions have been appropriately discussed. In this way, we have fully covered the knowledge needed to implement a mobile robot which can learn its environment and navigate through it autonomously.

References

[1] http://mindstorms.lego.com/overview/the_nxt.aspx.

[2] http://mitpress.mit.edu/books/chapters/.

[3] Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-whyte, H. F., and Csorba, M. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation 17* (2001), 229–241.

[4] Dudek, G., and Jenkin, M. *Computational principles of mobile robotics.* Cambridge University Press, New York, USA, 2000.

[5] Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer 22*, 6 (1989), 46–57.

[6] Fox, D., Burgard, W., Thrun, S., and Cremers, A. B. Position estimation for mobile robots in dynamic environments. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence* (Menlo Park, CA, USA, 1998), American Association for Artificial Intelligence, pp. 983–988.

[7] Herath, D., Kodagoda, S., and Dissanayake, G. Simultaneous localisation and mapping: A stereo vision based approach. In *Intelligent Robots and Systems, IEEE/RSJ International Conference* (2006), pp. 922–927.

[8] Jefferies, M. E., and Yeap, W.-K. *Robotics and Cognitive Approaches to Spatial Mapping.* Springer, 2008.

[9] Klippel, A., and Kulik, L. Using grids in maps. In *Theory and Application of Diagrams. First International Conference* (2000).

[10] Kuipers, B., and tai Byun, Y. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems 8* (1991), 47–63.

[11] Margaret E. Jefferies, W. K. Y. *The Utility of Global Representations in a Cognitive Map.* Springer, London, UK, 2001.

[12] Mitsou, N. C., and Tzafestas, C. S. Temporal occupancy grid for mobile robot dynamic environment mapping. In *Control & Automation, 2007. MED '07. Mediterranean Conference* (2007), pp. 1–8.

[13] Pagello, E., Groen, F., Arai, T., Dillmann, R., and Stentz, A. *Intelligent autonomous systems 6.* IOS Press, 2000.

[14] Se, S., Lowe, D., and Little, J. Vision-based mobile robot localization and mapping using scale-invariant features. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA* (2001), pp. 2051–2058.

[15] Templeton, T., Shim, D. H., Geyer, C., and Sastry, S. S. Autonomous vision-based landing and terrain mapping using an mpc-controlled unmanned rotorcraft. In *IEEE International Conference on Robotics and Automation for consideration* (2006), pp. 1349–1356.

[16] Thrun, S. Probabilistic algorithms in robotics. *AI Magazine 21* (2000), 93–109.

[17] Thrun, S. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium* (2002), Morgan Kaufmann Publishers Inc., pp. 1–35.

[18] Tomatis, N., Nourbakhsh, I., and Siegwart, R. Simultaneous localization and map building: A global topological model with local metric maps. In *Intelligent Robots and Systems, IEEE/RSJ International Conference* (Maui, HI, USA, 2001), vol. 1, pp. 421–426.

[19] Verri, A., and Uras, C. Metric-topological approach to shape representation and recognition. *Image and vision computing 14* (1996), 189–207.

[20] Wolter, D. *Spatial Representation and Reasoning for Robot Mapping: A Shape-Based Approach.* Springer, 2008.

[21] Wolter, D., Latecki, L. J., Lakĺamper, R., and Sun, X. Shape-based robot mapping. In *Advances in Artificial Intelligence* (2004), vol. 3238, Springer, pp. 439–452.

[22] Yeap, W. K., and Jefferies, M. E. Computing a representation of the local environment. *Artificial Intelligence 107*, 2 (1999), 265–301.

[23] Yeap, W. K., Jefferies, M. E., and Naylor, P. An mfis for computing a raw cognitive map. In *IJCAI* (1991), pp. 373–380.