A Framework for Semantic Map Construction

Basem Ibraheem[1], Abeer Hamdy[2],Nevien Darwish[3]

[1]*Computer and Systems Department, Electronics Research Institute*
Giza, Egypt

[2]*Faculty of Informatics and Computer Science, British University in Egypt,*
& Computers and Systems Department, Electronics Research Institute
Egypt

[3]*Faculty of engineering,Cairo University*
Giza, Egypt

Abstract

The paper proposes a framework for building a semantic map for indoor environment using a mobile robot. The framework includes five main modules which are: 1) Raw and spatial data acquisition module, 2)Steering and exploration module, 3)Information detection and extraction module, 4)Semantic knowledge mapping module and 5) High level task planner interface. The framework will be implemented on the ATRV-Mini robot. A simulation and control environment has been developed for this purpose. The steering and exploration module and the tasks responsible for the spatial data acquisition using sonar sensors and constructing a sonar occupancy map have been developed. The simulation results showed that the ATRV-mini robot is capable of learning metric maps of potential navigation areas.

Keywords: *Mobile Robot, ATRV-mini, navigation, Fuzzy controller, Map Learning.*

1. Introduction

Autonomous indoor robots can be of great impact in different applications. Autonomous robot moving in unknown environment encounters more challenges than the one who recognizing and memorizing the environment. Recognizing and memorizing the environment requires a scheme to store data about the environment. This scheme produces a map that should satisfy many constraints such as ability to store all information recognized, supporting quick store and retrieve operations and allowing further processing and updates. The robot should be able to identify its current location, and its relation to a target location to allow motion plans execution. Landmarks and distinct environment features provide a great help for localization to take place. Researchers have suggested many types of maps in the context of automatic environment map learning using robots the most useful types are:1) Metric Area Based Maps, 2)Topological Maps, 3) Hybrid maps and recently, 4)Semantic Maps.

Metric Area Based Map [3], [9], [10], [11], [14], [18], [19], [20], [22], [25] is a grid based area map uses grid cells to represent the regions of the environment. Each cell (region) is distinguished by a property that represents its occupation which could be: unknown, empty, occupied. **Topological Map** [15], [24] represents the environment as a graph, with the nodes representing recognizable locations or landmarks, and the edges representing clear paths from one node to another, usually doors or corridors.

Metric maps are easy to construct but suffer from the required high computational cost. On the contrary, topological maps are difficult to build but small space is required to build a huge one. Researchers suggested ***Hybrid Maps*** [4], [5], [12], [16] to overcome the limitation of each map type. Hybrid map can be obtained by correlating recognized landmarks on certain locations over an occupancy metric map. This type of maps leads to another challenge which is the matching techniques and algorithms.

Recently, ***Semantic maps*** [1],[2], [6], [7], [8], [13], [23] have come up to capture the human point-of-view of robot environments, enabling high-level and more intelligent robot development and also human-robot interaction. Semantic map for a mobile robot is defined in [6] as *"a map that contains in addition to spatial information about the environment, assignments of mapped features to entities (e.g. objects, events, functionalities located in space) of known classes. Further knowledge about these entities, independent of the map contents, is available for reasoning in some knowledge base with an associated reasoning engine"*.

This paper proposes a generic framework for constructing semantic Maps for indoor environment. The framework is introduced in Section 2. Section 3 introduces the robot specification and explains the constructed simulation environment. Sections 4 to 7 discuss the implementation and simulation results of the steering & exploration module and the sonar occupancy map generators. Finally, section 8 concludes the paper and introduces the future work.

2. Proposed Semantic Map Construction Framework

The proposed framework for semantic map construction is composed of five main modules, as shown in figure 1, which are:

1. Steering and Exploration Engine

It is a Fuzzy Controller responsible for controlling the robot motion to safely explore an indoor environment, while doing the data acquisition tasks required to construct the semantic map.

2. Raw and Spatial Data Acquisition Layer

This layer consists of two main components: “Online occupancy map generator” and “Textual sign detector”. *Online Occupancy Map Generator* creates an online metric map by using a probability model of sonar beams with respect to robot location information. While *Textual Sign detector is responsible* for seeking for signs around the explored areas. Whenever a sign is detected, it signals the steering and exploration engine to stop the robot until the sign image is correctly zoomed and acquired.

3. Information Detection and Extraction

This module consists of two components: High resolution offline occupancy map generator and Textual sign reader. *High-Resolution offline Occupancy Map generator* eliminates sonar noise from the online one and generates a more accurate map with higher feature details. *Textual Sign Reader Module* is activated whenever a sign is detected to recognize its textual content.

4. Semantic Knowledge Mapping

This module includes three components: Sign labeling and correlation component, indoor ontology models and reasoning engine and knowledge domain updating component.

Sign labeling and correlation is responsible for overlaying the recognized signs on the occupancy map, with respect to location and orientation information of the robot and camera head to generate a hybrid map.

Knowledge domain updating component applies object recognition and correlation techniques on images perceived by the robot for improving the knowledge existing in the constructed semantic map. While, the *indoor ontology models and reasoning engine* defines the relations between entities of the indoor environment domains such as a corridor connects between halls or forks into rooms, a door rotates to allow or prevent entering a room , an elevator connects between rooms in different levels and so on. The different functions of rooms in a facility will also be identified and linked to each other as well as equipments existing by default due to a defined function such as a Shredder machine in the mail room.

5. High-Level Task Planner Interface

It’s a *Text Based Query System* that includes a human computer interface tailored to receive instructions in simple English phrases biased by realistic needs such as “send the admin laptop to maintenance” or “Fetch the manager’s mail”. A traditional task planner interface requires spatial destination information or a sequence of land mark destinations to generate and follow a path plan [34]. Semantic maps can improve task planning in many ways including extending the capabilities of the planner by reasoning about semantic information, and improving the planning performance in large specified domains [6].

A semantic task planner will abstract the task plan into objectives to be achieved by reasoning through the available knowledge and utilizing the spatial information; the robot knowledge base should allow it to break down and map the objectives into a lower level of spatial destination sequences. For this knowledge base to be deterministic and contained in a model – ontology - a certain limited domain has to be considered. For example if the robot is capable of reading a student class time-table while navigating the walls of a school, by modelling the time, space, and a relation between instructors and subjects. The robot can use the identified class room numbers from door signs and their locations on the hybrid map to reach a specific instructor in locations according to time say to deliver him a book or an assignment.

The proposed framework will be implemented on the ATRV-Mini robot owned by The Electronic Research Institute in Egypt.

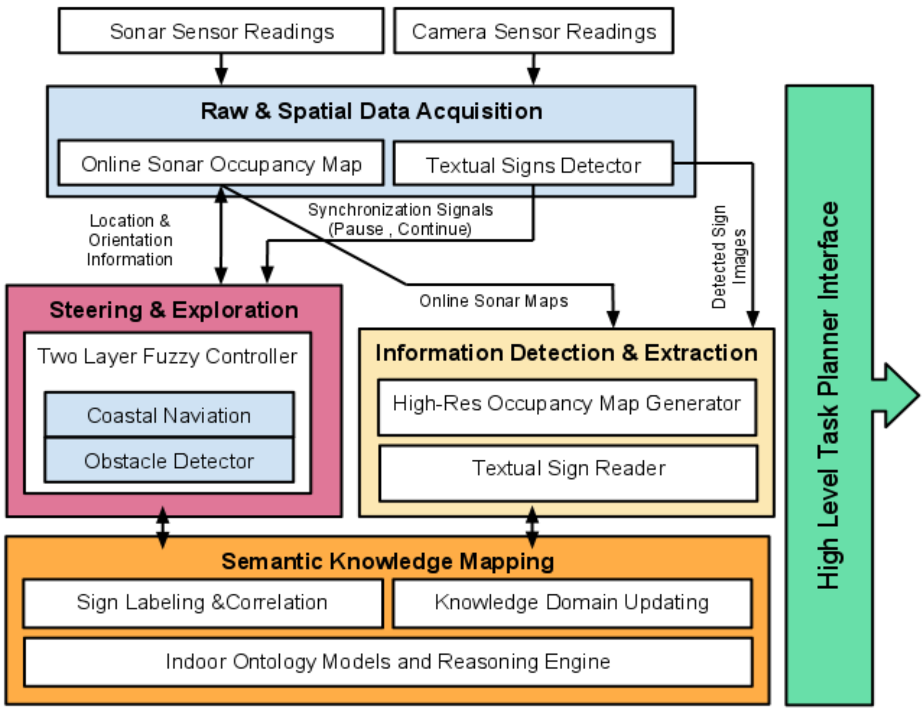


Fig. 1 High-Level Semantic Task Planner Framework

3. Development Platform

3.1 The Robot

Our research robot "ATRV-Mini" is a product of IRobot [28]. It is a skid steering robot, with 16 Polaroid™ sonar sensors array and a PTZ CCD camera as shown in fig 2. The maximum distance reported by manufacture is 5.5 meters, but during calibrations, the maximum value that we could read was almost 4 meters. Researches agreed that the sonar sensor accurate beam is bounded by a cone that has a head angle of 30 degrees. The navigation commands given to the robot are the values of the translation and rotational velocity.

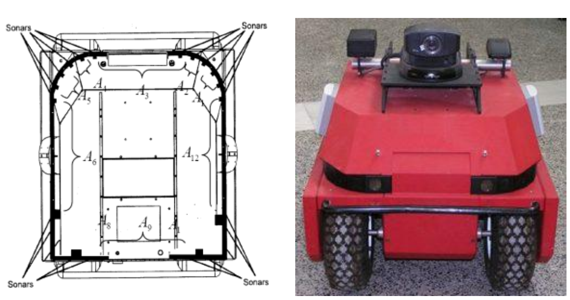


Fig. 2 ATRV-Mini robot & its sonar distribution

3.2 The Development Environment

A simulation environment is required to facilitate developing and testing the different framework modules before carrying them out on the real robot. Several open source simulators had been investigated; among them the "***player-stage***"[26] which found to be the best fit for our purpose.

*"**Player**"* is a cross-platform robot device interface and server for robot control. Player server is installed on the robot to provide the client (robot control program) access to the robot devices (sensors and actuators) over TCP network. The client can be written using any programming language that supports TCP sockets, and can run on any computer with a network communication to the robot. Player supports concurrent connections to the robot devices which allows for concurrent clients.

"***Stage***" is a multi-agent visual simulation environment that can used to simulate mobile robots moving in two dimensional bitmapped environments. It is capable of simulating many sensors such as sonar, laser and cameras, also simulating actuators such as robot motors and gripper arms. The simulated robot devices customized by "stage" are replaced by the device drivers of the ATRV-mini devices to make the simulated robot in compatible with the real one.

Stage presents a standard Player interface so few or no changes are required to move between simulation and hardware. Stage receives the same control code and executes it on the visual interface as if it is a robot with player server and device drivers installed.

To achieve our purpose "player" and "stage" have been interfaced with an open source C++ data flow oriented development environment called "***FlowDesigner***"[27] with some reusable artificial intelligence and image processing libraries. *FlowDesigner* features a RAD GUI with a visual debugger and can be used to build complex applications by combining small, reusable building blocks. It is used to facilitate building the required fuzzy and neural network engines and the computer vision image processing modules of the framework. *FlowDesigner* is written in C++ and features a plugin mechanism that allows plugins/toolboxes to be easily added. A toolbox was developed and added to the flow designer to facilitate communication with the player server or stage. Figure 3 shows the main building blocks of the development environment including the player-stage and *FlowDesigner* integrated above the ATRV network platform. Additional free open source software under GNU has been used to assist in the construction of the framework which are: 1)Image manipulating program "***GIMP***" [29] to manipulate maps and photos captured by robot camera. 2)Computer Aided Design for drawing maps "***QCAD***" [30]

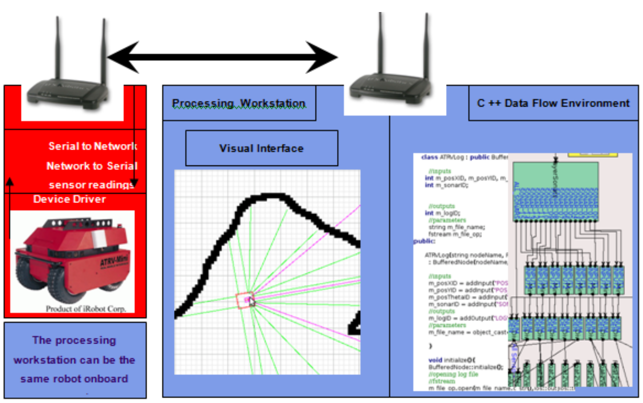


Fig. 3 Development environment

4. Steering and Exploration Module

The environment exploration strategy suggested in this paper is the perimeter following strategy. Perimeter following is achieved by keeping the robot going on a safe velocity near and parallel to the walls. This strategy is suggested due to the nature of the design plans of indoor environments (like hospitals and schools),each floor has a connected path of corridors that when traversed, can provide full coverage for environment scanning during mapping. Moreover, Moving parallel to the wall is beneficial for textual sign reader module as it makes the robot look at textual signs hanged on walls being followed from angles near to an orthogonal view, which expected to boost the recognition of the sign content (out of the scope of this paper). A two-layer fuzzy logic controller is suggested to achieve this objective. The first layer is for collision prediction and the second for perimeter following.

The idea of our fuzzy controller is based on a fuzzy controller has been introduced by Doitsidis and valavanis [17] for point-to-point navigation. Their controller consists of two layers; a layer for estimating collision probability in different directions around the robot based on sensor readings, and another layer for minimizing heading angle to the target point based on current and target X,Y location values while keeping the robot moving with a safe velocity. Their controller is implemented on a similar research robot ATRV-Mini, but with 24 sonar sensors. In our controller, we redesigned the two knowledge bases of the two layers to maintain the goal of our exploration module and taking into consideration the sensors distribution of our robot. Figure 4 shows the architecture of the suggested two layer fuzzy logic controller. The knowledge base of each layer is designed based on human experience.

Layer 1 knowledge base

The input to this layer is sonar sensor readings and the output is the collision probabilities. Each two sensors have been grouped with the common grouping operator among researchers *"Minimum operator"* as shown in fig 5a.

This layer consists of two separate fuzzy controllers: front controller and side controller. Front controller takes 3 sensor-groups input (Fr, FrLt, FrRt) and its output is the front collision probability (FrontCP). The side controller takes two sensor-groups input (Rt, RtFr) and its output is side collision probability (RightCP).

Each sensor-group reading has been fuzzified into four trapezoidal fuzzy sets representing the minimum distance between the group sensors and the obstacle causing this reading. These sets are "VERYCLOSE", "CLOSE", "FAR" and "VERYFAR" as shown in fig 5b. While each collision probability output (ranges from 0 to 1) are fuzzified into 4 fuzzy sets , "NO", "ALMOST", "YES" and "DANGER" as shown in fig 6. The proposed side controller consists of 16 fuzzy rule while the front controller contains 64 rules (rules are not listed because of space limitation).

Layer 2 knowledge base

The second layer consists of a single controller that keeps moving the robot forward while keeping the robot parallel and close to the walls. The *inputs* to this layer are the collision probabilities of the first layer (FrontCP, RightCP) along with the difference between the two sensors facing wall in the right (ΔRt=Rt1-Rt2), assuming that the robot will follow the walls with its right side. This difference ΔRt should be almost zero when robot is almost parallel to a wall. Fig 7 shows the input fuzzy sets ΔRt.

The output of this layer is the translation and rotation velocities of the robot. As reaching the maximum values of translation or rotation is not safe during navigation in indoor environments; in addition it is required to have a considerably low speed of motion for the sonar sensors to correctly scan the open space and produce a smooth plot; 70% of the maximum translation and 40% of the maximum rotation were considered. Fig 8 & Fig 9 show the output fuzzy sets of this layer. Note that, in this controller we considered only right and front collision probabilities as inputs and ignored both of left and back collision probabilities to reduce the number of rules to 48 rules.

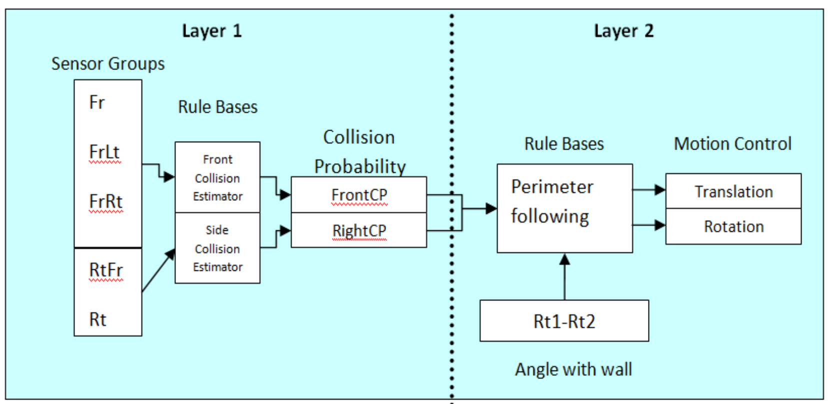
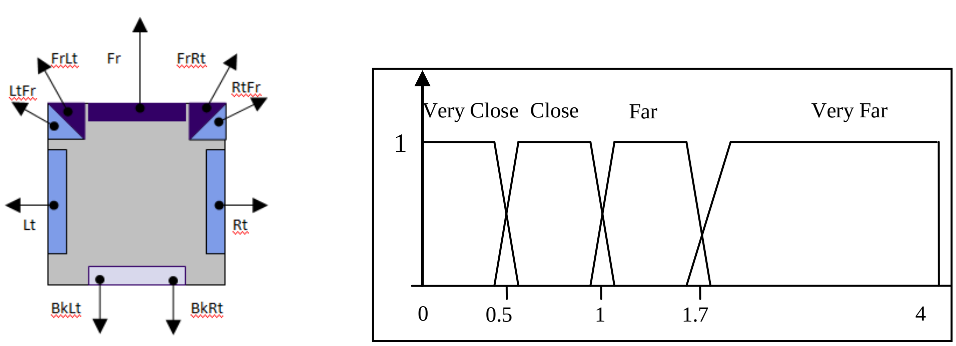


Fig 4 Two layer fuzzy controller for exploration



(a) (b)

Fig. 5 (a) Sensor groups, (b) Distance fuzzy sets.

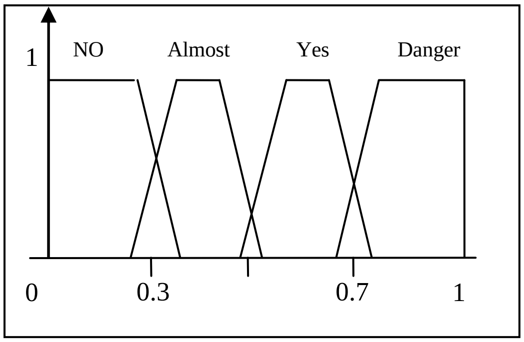


Fig. 6 Collision Probability Fuzzy Sets

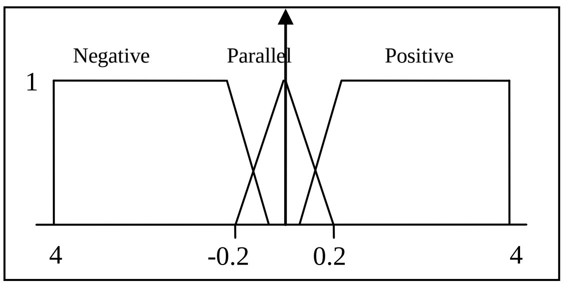


Fig. 7 Fuzzy sets of ΔRt

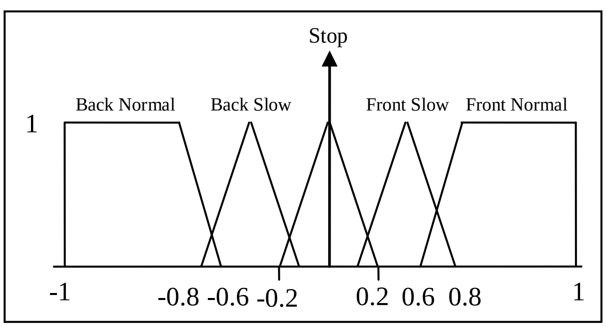


Fig. 8 Fuzzy sets of translation velocity

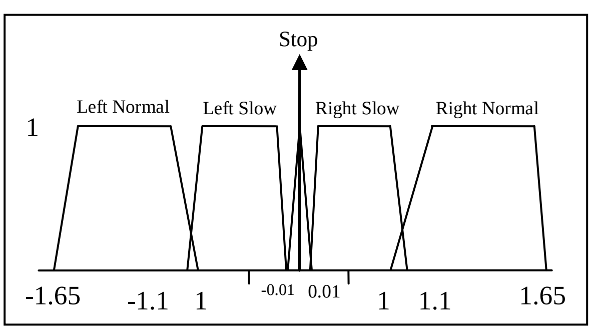


Fig. 9 Fuzzy sets of rotation velocity

5. Online Occupancy Map Generator

This module generates a low resolution grid metric map that contains information about places occupied with walls and furniture, or free such as corridors and halls. The map is created progressively while the robot is exploring the environment. After the robot finishes exploration (i.e. the perimeter is fully traversed) , a more accurate map with higher feature details is generated by the high resolution offline map generator.
Resolution is decreased for the online map to have a faster updating algorithm. The online map is stored as a gray scale bitmap (0~255) where each pixel in the bitmap represents a specified area of the environment and each pixel's gray scale level identifies a belief of occupancy. The resolution decrease was done by decreasing the available gray scale levels or degrees of occupancy certainty rather than by enlarging the area represented by each pixel. Only 4 gray levels were used as shown by table 1.

Table 1Gray Levels Of The Online Map

Bitmap Pixel Value	Unoccupancy Probability	Occurrence to a pixel when
00	0%	Initialized, unexplored or never updated
01	33%	First sonar reading affecting the pixel
10	66%	Second sonar reading affecting the pixel
11	100%	Third sonar reading affecting the pixel

Two online maps are being calculated, a local map which is used in updating a global one. The following subsections visualize and discuss the sonar and robot models that are used in creating the local map, and then the algorithm of updating the global map is introduced.

5.1 Single Sonar Beam Probability Model

The sonar beam model is considered a uniform 30 degree triangle with initial height of 80 pixels each pixel represents 5 cm, 80 pixel represents 4 meters. The grey scale level of the beam is one third of the maximum intensity (01) as shown in Fig 10. Each beam is stored as a square image matrix, clipping of modeling beams to match sonar reading value was done by clipping the square matrix.

5.2 Robot Model

The robot body is considered 50 cm * 50 cm and represented by 10*10 pixels image. The probability model of the robot is 100% free space, as it is always occupied by the robot itself (Fig 11). Preprocessing also creates 359 rotated robots, to be used later in representing robot at different orientations on the global map.


Fig. 10 Sonar beam model

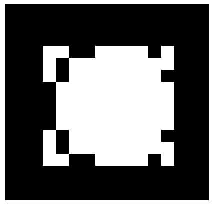


Fig. 11 Robot model

5.3 Local Map

Figure 12 is a bitmap shows the robot body after attaching 16 beams models to their pivot points. It can be seen that overlapping readings has a superposition effect. The local map in this bitmap has all sensor readings set to the maximum value (4 meters).

5.4 Global Map

When the robot starts moving or rotating, this effect is traced into a global map, by adding different local maps that are generated during motion as shown in Fig 13.

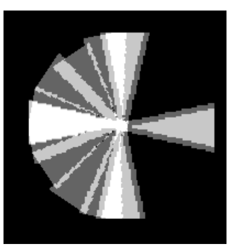


Fig. 12 Local map


Fig. 13 Accumulated local maps

The global map update algorithm proceeds as follows:
The system is started with a global map GM initialized as unknown map (all pixels are black). The robot placement is assumed to be the origin of the map. The exploration module is responsible for moving the robot in order to discover the map. For each step moved, the local map LM of the robot is accumulated over the global map using the following algorithm:

1. Robot moves to a new Position (X,Y) and Orientation (θ), determined by exploration module,
2. Read all sonar sensor values into array Sonars(16),
3. For each new step (i) received Do :
    a. Initialize LM(i) ,as unknown, in black (00) pixels
    b. Initialize a 16 beam models array each with length equivalent to the maximum *unclipped* range of the sonar sensors , store them into beams array Beams(16)
    c. Apply a robot model with neutral orientation = Robot($\theta = 0$) over the local Map LM(i)
    d. Clip the beams array Beams(16) by the current step sonar readings Sonars(16)
        i. SonarModels(16)=Clip (Beams(16),Sonars(16))
    e. Pivot each beam in SonarModels(16) on robot model at position and orientation according to the sensor position on the hardware to complete generating the local map LM(i)
4. Orient LM
    a. Rotate LM against the GM with actual robot orientation(θ)
    b. Shift LM against GM origin by actual robot position (x,y)
5. Update Global Map
    a. For Each Pixel in LM(i) do
       If Pixel value is 00 Then Bypass
       Else Add pixel to global map area of concern
       (Clipping is automatic set to 11)
6. This algorithm is terminated with the exploration module relieves localization information from the online map that it has already reached the original location (circulated the perimeter)
7. The resulting map is processed by image processing techniques for smoothing and boarder tracing to generate quantifiable representation of walls and free spaces.

6. High Resolution Offline Map Generator

The offline map generation process runs after the robot finishes exploring the environment and the online map is generated. The resulting map is a more accurate bitmap; it has three advantages over the online map which are:1) Higher feature detail (perimeter details). 2) Smooth and hence more consistent levels and changes of levels of occupancy probability. 3) Unexplored areas are identified from boundary areas; this happens by darkening the termination of a beam when it collides with an obstacle rather than just clipping it. The offline map algorithm used is similar to feature prediction algorithm suggested by Shane O. Sullivan[19].

7. Experiments and Results

7.1 Steering and Exploration Module

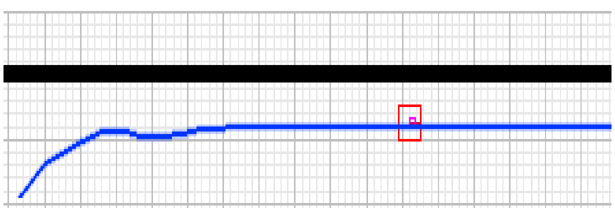
The two-layer fuzzy logic controller has been implemented on the robot simulator to test the behavior of the robot traversing different perimeters with different complexities.

Testcase1: Straight perimeter following

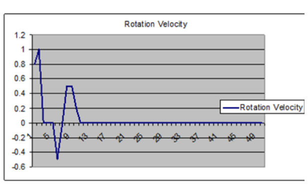
Figure 14 shows a straight forward pilot test to test fuzzy engine stability of the original open source libraries used. Figures 14a, 14b show the robot translation and rotation velocities values with time during the straight perimeter following. It has been found that the controller is stable without any unexplained performance issues.

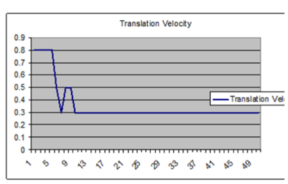
Test Case2: Complex Perimeter following

Figure 15a shows the original environment, while figure 15b shows the constructed map by the robot while following the environment's perimeter. The green dots specify locations visited by the robot along the perimeter which shows a good robot performance for reaching the environment sections except for the narrow corridors. The dashed red arrow on the CAD map shows the missed corridor. The problem of narrow corridors could be overcome by taking the readings of left sonar sensors into consideration while building the knowledge base of the fuzzy controller.



(a) The path of the robot starting from the left.

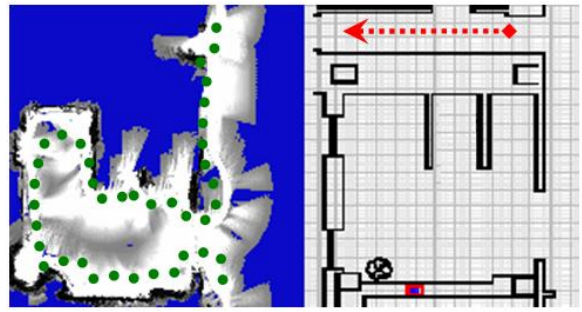




(b) Rotation velocity values during the robot path

(c) Translation velocity values during the robot path

Fig. 14 Straight perimeter following Test case1



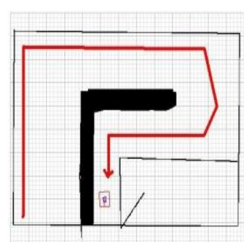
(b) (a)

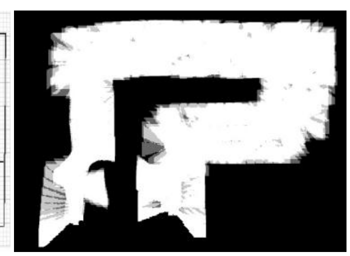
Fig. 15 Complex Perimeter following

7.2 Simulations of Online Map generator

Figure 15b shows the first test case for the online map generator module and it shows the capability of the robot on constructing an online map for the potential navigation regions while exploring the environment.

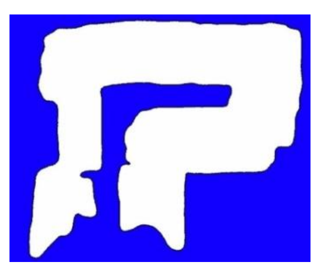
In advance of passing the online map to the offline map generator; the online map is processed (using image processing techniques (Edge detection, Boarder tracing) to produce a more accurate one with well defined borders. Fig 16b, 16c show the generated online map and the processed one to the CAD environment shown in Fig 16a.





(a) Original map and robot path.

(b) Generated online map.



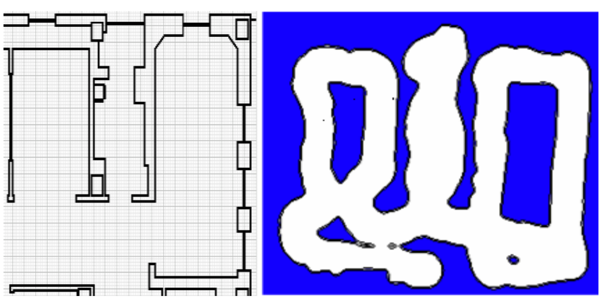
(c) Processed Online Map.

Fig. 16 Online map generation test case2

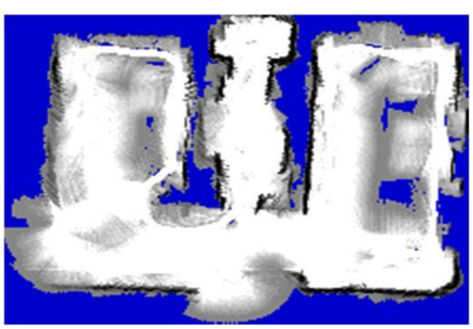
Figure 17 b shows the processed online map to the environment in fig 17a. It should be noted that online map generator algorithm doesn't differentiate between boarders resulting from clipping a sonar beam due to collision with obstacles or sonar beams which were left unclipped at maximum distance, in this way sonar left unclipped will have a boarder, which doesn't represent actual boarders in the environment (black areas in the middle of each room which doesn't exist in the original environment). This issue is treated by the offline map generation algorithm.

7.3 Simulations of Offline Map Generator

Figure 17 shows the original CAD map and the learnt online map and the offline map generated using Sullivan algorithm for feature prediction [19]. A major difference between the online and offline maps rather than wall details; is that unexplored areas are left unbounded in the offline map, because the boundary in offline map is a result of object feature prediction, not just a separation between known and unknown areas as in the simplified online map. This difference helps in later exploration (exploring unvisited areas) for full environment coverage.



(a) Original CAD map (b) Processed online map



(c) Offline Map

Fig. 17 Offline map generation test case

8. Conclusion and Future Work

The aim of our research is to develop a framework for constructing a semantic map for indoor environment using the ATRV-Mini mobile robot. The paper presented the different modules of the proposed framework. A simulation and robot control environment was constructed to facilitate implementing and testing the framework. The modules responsible of robot exploration and sonar information gathering for map construction are implemented and tested. The results showed that the robot is capable of learning the potential navigational areas. In a later phase, Textual signs will be detected, identified and allocated over the generated map. Textual signs can be used as landmarks and to represent two important aspects of the environment which are the function of each place and its relation to the high level tasks as well as the people functioning inside the facility. Thus, it should be possible to request a function from the robot such as deliver a laptop for maintenance or put some envelops in the outgoing mail box.

Acknowledgment

We acknowledge Prof. Ashraf Abedl Wahab professor of machine learning at Electronics Research Institute for his support to the work presented in this paper.

References

[1] A. Ranganathan, F. Dellaert, "Semantic modeling of places using objects", in Robotics Science and Systems Conf., 2007.

[2] A. J. B. Trevor, J. G. Rogers III, C. Nieto-Granda, and H.I. Christensen, "Applying domain knowledge to SLAM using virtual measurements", International Conference on Robotics and Automation, 2010.

[3] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots", In Proc. of the IEEE: Special Issue on Multi-Robot Systems, October 2006.

[4] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon and F. Savelli, "Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy", In Proc. of Int. Conference on Robotics and Automation, New Orleans, USA, 2004.

[5] B. Lorenz, H. Ohlbach and E. Stoffel, "A Hybrid Spatial Model for Representing Indoor Environments ", Web and Wireless Geographical Information Systems , SpringerLink, Volume 4295, 2006.

[6] C. Galindo, J.A. Fernández-Madrigal, J. González, A. Saffiotti "Robot Task Planning using Semantic Maps" , Robotics and Autonomous Systems 56, 2008.

[7] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fern´andez-Madrigal, J. Gonz´alez, "Multi-hierarchical semantic maps for mobile robotics In Proc. of the IEEE/RSJ Int. Conf on Intelligent Robots and Systems (IROS), Edmonton, CA, 2005.

[8] D. Meger, P. Forssen, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. Little, D. Lowe, B. Dow, "Curious George: An attentive semantic robot", in IROS Workshop: From sensors to human spatial concepts, pp. 390–404,2007.

[9] D. H¨ahnel, W. Burgard, D. Fox, "An efficient Fast SLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements", In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, Oct. 2003.

[10] G. Grisetti, C. Stachniss, W. Burgard, "Improving Grid Based SLAM with Rao Blackwellized Particle Filters by Adaptive Proposals and Selective Resembling", In Proc. of Int. Conference on Robotics and Automation, Barcelona, Spain , 2005.

[11] H. Zender, P. Jensfelt, O. Martinez-Mozos, G.-J. Kruijff, W. Burgard, "An integrated robotic system for spatial understanding and situated interaction in indoor environments", in: AAAI-07, Integrated Intell. Track, pp. 1584– 1589, 2007.

[12] J. Carlos, P. Broch , M. Teresa and E. Monferrer , "Cognitive Maps for Mobile Robot Navigation: A Hybrid representation Using Reference Systems", In Michael Hofbaur, Bernhard Rinner, and Franz Wotawa, editors, Proceedings of the 19th International Workshop on Qualitative Reasoning.2005.

[13] J. G. Rogers III, A. J. B. Trevor, C. Nieto-Granda and H. I. Christensen, "Simultaneous Localization and Mapping with Learned Object Recognition and Semantic Data Association", In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems September 25-30, 2011. San Francisco, CA, USA

[14] K. M. Chow, A. B. Rad and Y. L. Ip , "Enhancement of Probabilistic Grid-based Map for Mobile Robot Application", Journal of Intelligent and Robotic Systems ,Volume 34, Issue 2, Kluwer Academic Publishers ,USA, June 2002 .

[15] K. R. Beevers and W. H. Huang, "Loop closing in topological maps", In Proc. of IEEE Int. Conference on Robotics and Automation (ICRA 2005), pp. 4378-4383, Barcelona, Spain, April 2005.

[16] L. Romero, E. Morales and E. Sucar, "A Robust Exploration and Navigation Approach for Indoor Mobile Robots Merging Local and Global Strategies" , Advances in Artificial Intelligence, SpringerLink, Volume 1952/2000, pp.389-398,2000.

[17] L. Doitsidis, K. P. Valavanis, N. C. Tsourveloudis, "Fuzzy Logic Based Autonomous Skid Steering Vehicle Navigation", In Proceedings of ICRA'2002,2002, pp.2171-2177.

[18] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, B. Stankiewicz, "Integrating Multiple Representations of Spatial Knowledge for Mapping, Navigation, and Communication, Interaction Challenges for Intelligent Assistants", AAAI Spring Symposium Series, 2007.

[19] L. Jung-Suk, N. S. Yep, C. W. Kyun, "Robust RBPF-SLAM for Indoor Mobile Robots Using Sonar Sensors in Non-Static Environments", Advanced Robotics, Vol. 25, 2011.

[20] L. Kleeman, "Advanced sonar and odometry error modeling for simultaneous localization and map building" , Intelligent Robots and Systems, 2003. (IROS 2003).

[21] S. O'Sullivan, J.J. Collins, M. Mansfield, D. Haskett, and M. Eaton ,"Linear Feature Prediction for Confidence Estimation of Sonar Readings in Map Building", In Proc. of 9th International Symposium on Artificial Life and Robotics (AROB), 2004.

[22] S. Thrun, Y. Liu, D. Koller, et al., "Simultaneous localization and mapping with sparse extended information", International Journal of Robotics Research, 2004.

[23] S. Vasudevan, S. Gachter, V. Nguyen, R. Siegwart, "Cognitive maps for mobile robots and object based approach", Robotics and Autonomous Systems 55, 2007.

[24] W. H. Huang and K. R. Beevers, "Topological mapping with sensing-limited robots", In M. Erdmann et al., editors, Algorithmic Foundations of Robotics VI, pp. 235-250, Zeist, Springer, 2005.

[25] X. Li, X. Huang, And M. Wang, "Robot Map Building From Sonar Sensors And Dsmt", Information & Security. An International Journal, Vol. 20, 2006.

[26] Player Stage project http://www.sf.net/playerstage.

[27] FlowDesigner project http://www.sf.net/flowdesigner.

[28] iRobot, iRobot Corporation: Home Robots , Official Website: http://www.irobot.com.

[29] GIMP: http://www.gimp.org/

[30] Qcad: www.qcad.org