"Map Format for Mobile Robot Map-based Autonomous Navigation"

Andreu Corominas Murtra; Josep M. Mirats Tur

IRI-DT 2007/01 *Febrer 2007*

INSTITUT DE ROBÒTICA I INFORMÀTICA INDUSTRIAL
CSIC-UPC
Barcelona

Map Format for Mobile Robot Map-based Autonomous Navigation

Andreu Corominas Murtra, Josep M. Mirats Tur

Institut de Robòtica i Informàtica Industrial (CSIC-UPC). C/Llorens i Artigas 4-6 2nd Floor. 08028, Barcelona. Spain. E-mail: acorominas@iri.upc.edu, jmirats@iri.upc.es

Abstract

This technical report defines the spatial representation and the map file format used in a mobile robot map-based autonomous navigation system designed to be deployed in urban areas. After a discussion about common requeriments of spatial representations for map-based mobile robot autonomous navigation, a proposed environment model that fulfills previously discussed requeriments is formally presented. An example of a map representing an outdoor area of an university campus of about $10000m^2$ is given to better illustrate the map format. Finally, the report shows simulation results on global localization and path planning using the proposed map.

1 Introduction

Mobile robot map-based autonomous navigation can be defined as the ability of a mobile robot to solve without human aid three tasks. First, "Global Localization", given sensor readings and the map. Second, "Path Planning", given current position of the robot and a goal point located on the map. Third, "Path Execution", which involves "Position Tracking" and "Obstacle Avoidance".

Map-based navigation implies that the robot has a prior representation of the environment called *map*. This map is *the data model of the environment* used by the robot to autonomously navigate. Different environment representations have been studied and proposed by researchers and the widely accepted taxonomy divides spatial representations between metric and topological ones [5]. However, most of the existing spatial representations have been called *hybrid* since they use metric and topologic information to model the environment [14, 15, 8].

In this work a representation based on geometric entities on the 2D space has been choosen because of its potential scalability to larger environments and its potential localization accuracy since space is not discretized. Two dimensional information is enhanced with 'pseudo 3D' data since the proposed representation keeps height of these geometric entities and also models stairs and ramps usually found in outdoor urban environments. Moreover, semantic information is added to geometric entities to improve the spatial representation getting closer to human models.

This report is organized as follows: section 2 discuss requeriments of an environment model for mobile robot navigation, section 3 presents the description of the proposed environment model, section 4 shows some results on global localization and path planning using this map model and section 5 summarizes the conclusions and points out the future work. The report also includes an appendix describing the used map file format.

2 Requeriments of a Spatial Representation

We have identified six key requeriments for spatial representations in mobile robot map-based autonomous navigation:

- Scalability: stands for the property of a given representation to scale up to large environments keeping in reasonable bounds both memory and computational resources. In terms of memory resources, scalability is closely related with the compactness of the representation. Some authors have measured a compactness figure as a ratio of $bits/m^2$ [2]. Compactness also gains importance in a network robot context where whole or parts of the map are sent through a communications network, therefore reducing data size is of great interest. From computational efforts point of view, scalability to large environments have to be assured in order to deal with real-time requeriments of some navigation tasks.

- Accuracy: although an error coming from the mapping method has to be accepted, the spatial representation should not restrict the accuracy of localization algorithms beyond the sensor technological limitations.

- Flexibility: map model flexibilty can be defined as the capability of a given representation to be used by different navigation methods, sensors, tasks and robots.

- Three dimensional extending: performing mobile robot navigation tasks in outdoor urban areas requires 3D data coming from both perception and map data. Even if three dimensional data is not exhaustive, spatial representations have to incorporate useful 3D data in order to match three dimensional perceptions with the map model.

- Automatic conversion from a mapping source: There exist different mapping sources like robot mapping [17], Geographical Information Systems (GIS) [12] or an architect CAD. A desirable aspect is that the robot map can be built automatically either directly from robot mapping or by means of an automatic conversion from other mapping formats.

- Human compatible: representations close to human maps offer advantages, especially in systems where human-robot interactions are expected. Therefore, a map model accepting semantic information will be more suitable to interface with humans. A Geographical Information System (GIS) can be used as a powerful interface to solve this issue, especially in outdoor environments like urban areas [12]. However, we have to assure automatic translation from GIS maps to robot maps.

3 Map Description

This section presents how spatial information is arranged to form the environment data model, also called the spatial representation or the map. The representation is in the 2D plane, based on geometric entities and inspired from the 'GIS vector' format [12]. However, height information of geometric entities is added to give to the map a pseudo 3D information. Stairs and ramps are also modelled since they are key 3D obstacles in outdoor areas for navigation purposes.

The map $\mathcal{M}$ is defined with four coordinates limiting its borders and with a list of NB obstacles. (mx_1, my_1) is the left-up corner point and (mx_4, my_4) is the right-down corner point.

$$\mathcal{M} = \{mx_1, my_1, mx_4, my_4, o^1, ..., o^{NB}\} \tag{1}$$

The $k-th$ obstacle of the map, o^k, is defined with a list of NS^k segments, an integer id^k assigned to identify the obstacle, an integer ST^k describing the type of the shape representing the obstacle and semantic information related to it.

$$o^k = \{s_1^k, ..., s_{NS^k}^k, id^k, ST^k, semanticI^k\} \quad k = 1..NB \tag{2}$$

where $ST = 1$ when obstacle is represented with a closed polygon, $ST = 2$ for an opened polygon, $ST = 3$ for a closed curved shape, $ST = 4$ for an opened curved shape, $ST = 5$ for stairs and $ST = 6$ for ramps. Semantic information is a character string labelling some features of the obstacle as if it is a building, a column, a flowerpot, a trash and so on.

The $l-th$ segment of the $k-th$ obstacle , s_l^k, is defined from the a_l^k point to b_l^k point (currently, only straight segments are implemented). Height h_l^k, an indoor/outdoor boolean and semantic information also accompanies the segment.

$$s_l^k = \{ax_l^k, ay_l^k, bx_l^k, by_l^k, h_l^k, inOut_l^k, semanticI_l^k\} \quad k = 1..NB, \; l = 1..NS^k \tag{3}$$

inOut boolean takes 0 for an indoor segment and 1 for an outdoor segment and again, semantic information is a string describing some features of the segment as it represents a wall, a door, the material which is built with, the color and so on. All segments are oriented, so they are defined from left to right viewed from the free space.

3.1 Stairs and Ramps

Stairs (steps) and ramps usually present in outdoor urban areas poses to mobile robot community a challenge since 3D information has to be taken into account (from sensors and the map) in order to deal with navigation tasks in a robust manner. The proposed map represents these two obstacles identyfing them with the label ST ($ST = 5$ for stairs and $ST = 6$ for ramps).

Stairs are modelled as a list of segments, like the other obstacles of the map . From the downstairs, the first segment is orieneted from left to right, with a height equal to step height (p.e 0.2 m), just as a 'short' wall. The second step will be a segment oriented like the first one, just separated the step width and with height two times of step height (p.e 0.4 m). Other steps are built iteratively. Finally a segment inversely oriented with height = 0 ends the stairs obstacle. This representation is well suited for the ray tracing function computed for navigation purpsoes, since the robot needs to generate synthetic (simulated) range data from the map (see section 4). This stairs model is also useful for path planning because step segments define an obstacle area to be avoided for the planner (if wheeled robot!). Figure 1 shows the stairs obstacle model.

Ramps can also be modelled as obstacles with null height. Ramp borders are decribed with a closed polygon formed by the ramp projection to the 2D plane. Ramp orientation is parametrized with the normal vector to the surface. With this information ray tracing function can compute synthetic range data from the map and match them with sensor data. For path planning, ramp obstacles can be completely ignored or used to force routes to pass through them. Figure 1 draws the ramp model. Ramps are not yet implemented in the current version of the map representation.

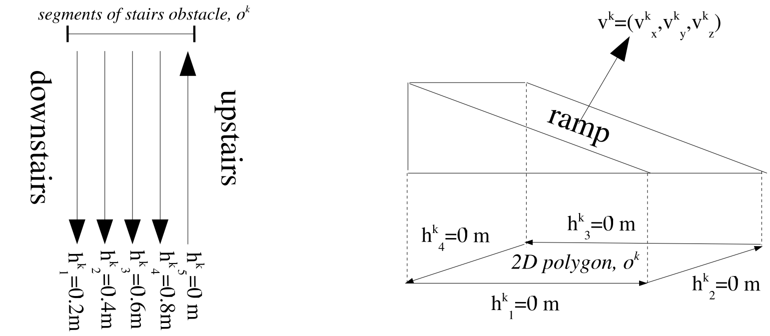

Figure 1: Stairs and Ramp model

4 Experiments

This section shows an example of using this map model to deal with two navigation tasks: global localization and path planning. The map used describes the surroundings of the 'FIB Square' at the Campus Nord of the Universitat Politècnica de Catalunya (UPC), representing an outdoor environment of about $10000m^2$. This area is expected to be the demostration field for the URUS european project [13]. Figure 2 shows the geometric part of the map of this environment, the origin of the metric coordinates and two illustrative selected points. Blue arrows in the figure mark places where pictures of the figure 3 were taken in order to familarize the reader with the environment. Preliminary results on global localization and path planning using this environment model are given in the following subsections.

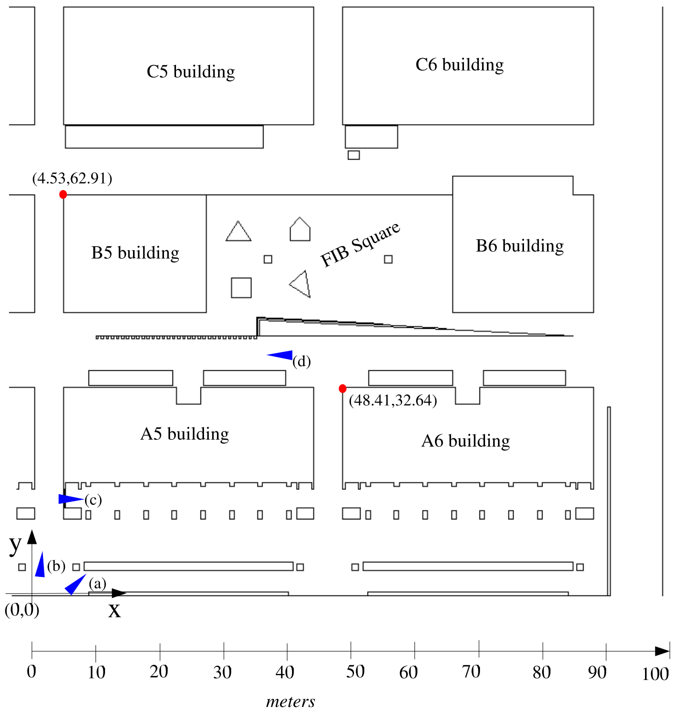

Figure 2: Geometric part of the map of the surroundings of the 'FIB Square' at the Campus Nord of the Universitat Politècnica de Catalunya (UPC)

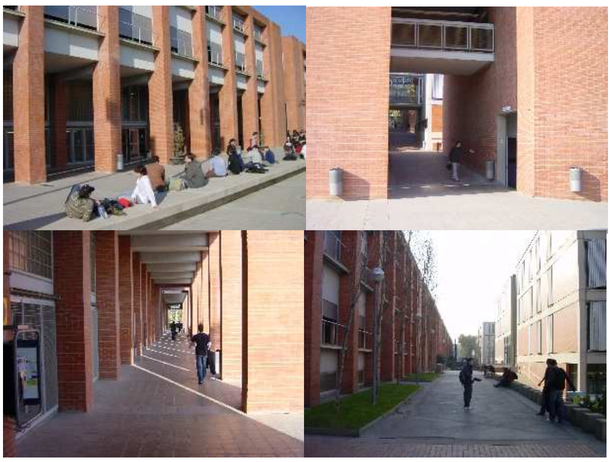

Figure 3: Pictures taken from a,b,c,d points on figure 2

4.1 Global Localization

In this subsection we present a simulated experiment using a particle filter method to global localize the robot [16]. Figure 4 shows an overview of the implemented algorithm. The simulated robot is equipped with a compass, a two dimensional laser scanner and the encoders of robot wheel motors. In the experiment the orientation of the robot is assumed to be initially known with an uncertainty of 3 degrees simulating an initial well calibrated compass reading. In large environments like a whole campus or a city quarter, initial readings provided by devices such as a compass, a GPS or even a GSM device can help to solve the global localization problem drastically reducing the search space (x, y, θ). The interest of solving global localization in an urban area of about $100m \times 100m$ resides in the fact that the current state of the art in GSM urban positionning gives an average error of one hundred meters [4].

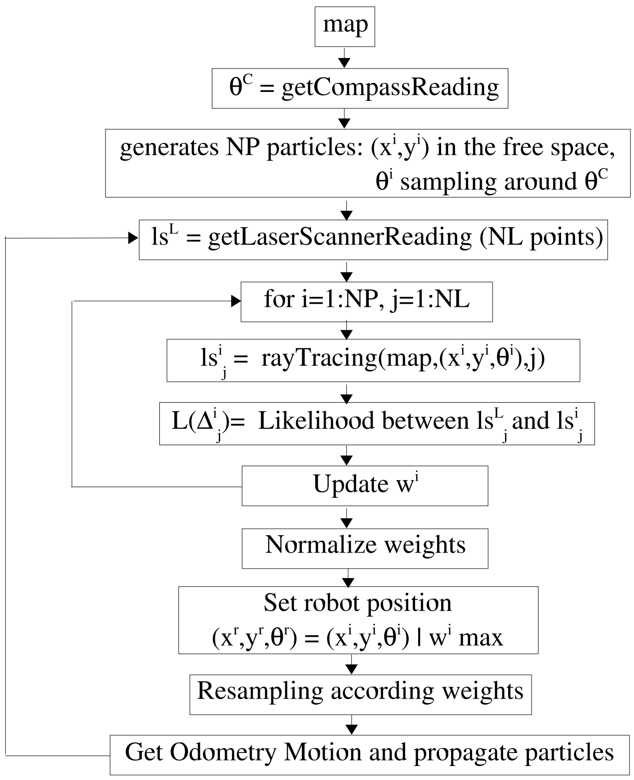

Figure 4: Overview of particle filter algorithm implemented for global localization

The likelihood for the particle $i-th$ is calculated between a simulated laser scanner ls^L with sensor noise of σ_L, generated from the actual robot position (x^r, y^r, θ^r), and the simulated synthetic laser scanner ls^i generated from the $i-th$ particle position (x^i, y^i, θ^i) on the map. For the $j-th$ synthetic ray of the $i-th$ particle, say ls^i_j, we compute the difference with the $j-th$ perceived ray ls^L_j:

$$\Delta^i_j = ls^L_j - ls^i_j \quad i = 1..NP,\ j = 1..NL \tag{4}$$

The sensor model is implicitly encoded in the likelihood function $L(\Delta^i_j)$ which is modelled as:

$$L(\Delta^i_j) = \alpha\mathcal{N}(0, \sigma_m) + \beta\mathcal{U}(-RMAX, 0) \tag{5}$$

where $\mathcal{N}(0, \sigma_m)$ is a normal bell-shaped curve centered on zero, with variance σ_m. $\mathcal{U}(-RMAX, 0)$ is an uniform function from $-RMAX$ to 0. The normal curve models sensor innacuracies and tolerancy to range measurements since the particle set is a sampling of the infinite set of all possible positions for the robot. Therefore, σ_m models both sensor noise and the acceptance of the algorithm to give good 'score' to particles close to the robot position. The second uniform function, inspired on the model proposed in [16], models the possibility that actual sensor readings could be smaller than the computed synthetic rays for a given location due to the environment dynamics when unmodelled objects (people, cars, bikes, ...) occlude modelled regions.

The update of the $i-th$ particle weight of the current iteration, $w^i(t)$, is made using all the individual likelihoods, $L(\Delta^i_j)$, between each perceived and synthetic rays and using the previous weight, $w^i(t-1)$ of the $i-th$ particle:

$$w^i(t) = w^i(t-1) \cdot \frac{1}{\sum_{j=1}^{NL} \frac{1}{L(\Delta^i_j)}} \tag{6}$$

Figure 5 shows the first four iterations of this particle filter implementation using the presented map model. These simulation results are calculated with $NP = 6000$ particles, $NL = 131$ laser scan points, scanner range limitted to $RMAX = 15m$, laser scanner noise $\sigma_L = 5cm$, $\alpha = 1$, $\beta = 0.1$ and $\sigma_m = 0.5m$. For this exposed simulation results, both compass and laser scanner readings are not actual readings from sensors, but sensor data is simulated.

The ray tracing function that computes the synthetic ray ls^i_j traced on the map from the $i-th$ particle position for the $j-th$ scan point, takes into

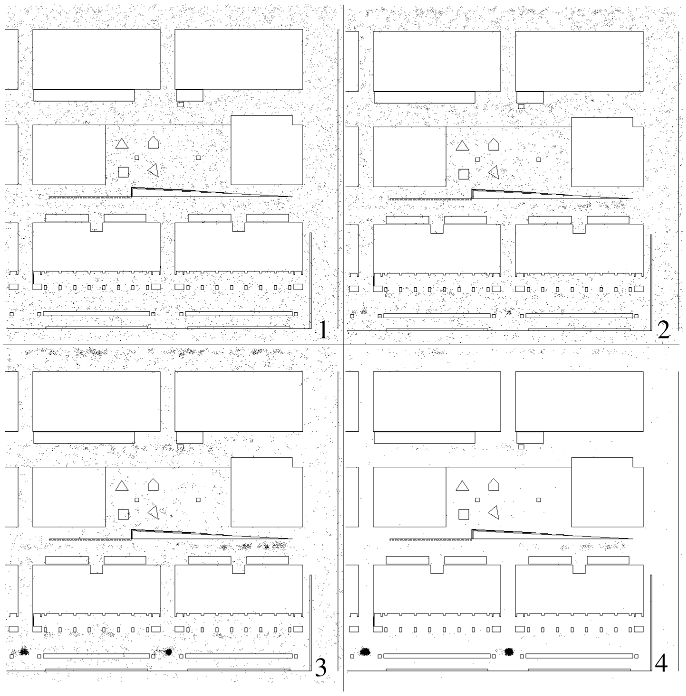

Figure 5: Four iterations of particle filter for global localization

account the laser mounting height in the robot, h^L, and the map segment height h_l^k to decide if ray stops or 'crosses' the $l-th$ segment of the $k-th$ obstacle of the map. Special care has also to be taken with stairs obstacles when ray tracing is performed. If the intersection of a synthetic ray with a given stairs obstacle is with the step segment of $h_l^k = 0$, it says that the robot is upstairs and, therefore, the ray will cross completely the obstacle. Otherwise, ray will stop at the first step with height greater than laser device mounting height. Figure 6 shows this situation.

This algorithm has a complexity of $\mathcal{O}(NP \cdot NL \cdot NS)$ where NP is the number of particles, NL the number of scan points and NS is the number of segments of the map. The dependence in NL of the complexity could limit scalability of the map to larger environments. Therefore, in order to overcome this drawback, we have adopted a trick borrowed from computational geometry to avoid a global search for each segment of the map and each synthetic ray. When the robot loads the map, it computes for each obstacle o^k the minimum bounding circle [7], so a center c^k and a radius R^k is assigned to each obstacle o^k of the map. Before computing

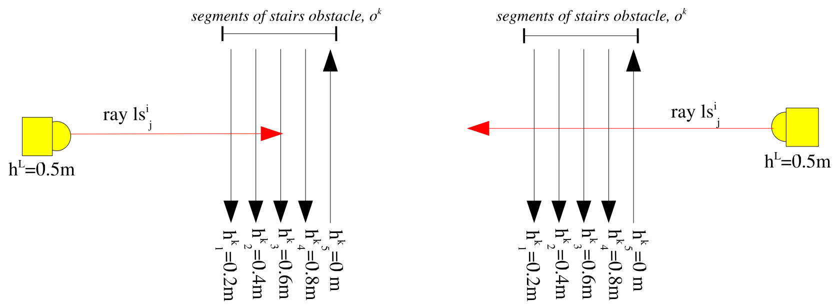

Figure 6: Ray tracing behaviour with stairs when robot is downstairs (left) or upstairs (right).

the intersection between ray and the segments of the given obstacle o^k, a single intersection test is done between semicircle of the synthetic scanner ls^i (positioned on (x^i, y^i, θ^i) with radius equal $RMAX$) and the minimum bounding circle (c^k, R^k). If intersection fails, the obstacle o^k is enterely ignored to compute ray tracing. In order to carry out real time requirements when position tracking is performed, we are studying adaptative techniques to iteratively reduce the number of particles (NP) and the number of used scan points (NL) such as it has been proposed on [9, 3].

No other global localization techniques such as [1, 6] have been implemented but from the map model point of view, automatic conversion can provide the geometric feature map required in [1] or the occupancy grid map used by [6]. However, converting the map model to other environment models used by these localization methods can deal to lost of some of the discussed properties of the original map.

4.2 Path Planning

Path planning between two points located on the map, $point0$ and $pointGoal$, has been implemented using Rapidly-Exploring Random Trees (RRT) [11].

The idea of the RRT's is to randomly and iteratively build a tree in the free space attempting to connect the start point with the goal point. In each iteration, a new randomly generated point tries to directly connect to the tree. Algorithm iterates until the goal point can also directly connect to the tree. Figure 7 presents the implemented algorithm used to generate an RRT.

Algorithm description makes use of a set of geometric primitive functions such as point interference or segment interference with the map obstacles.

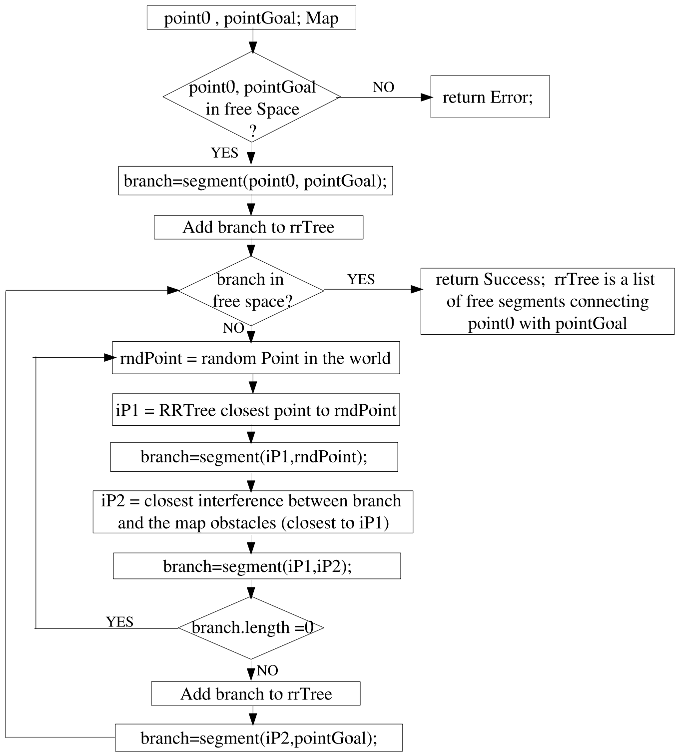

Figure 7: Implemented algorithm generating an RRT

This functions takes into account all obstacles o^k with $h^k > 0$, therefore ramps will be treated as free space assuming that wheeled robots can move across them. Figure 8 shows a generated RRTree and the resultant smoothed path on the map.

Like in the global localization problem, we have not implemented other methods for path planning but we are confident that other planning methods can be implemented using the geometric basis of the proposed environment representation (see [10, 11]).

For environments with narrow passages like the presented one (see corridors between buildings at figure 2), computational efficiency of RRT's can be improved implementing a balanced RRT. In balanced RRT's two trees are alternatively generated, one starting at $point0$ and the other one starting at $pointGoal$. The fact of building two trees speeds up the algorithm since connection between $point0$ and $pointGoal$ is reached faster. RRT algorithm also easily accept to force paths passing through ramps.

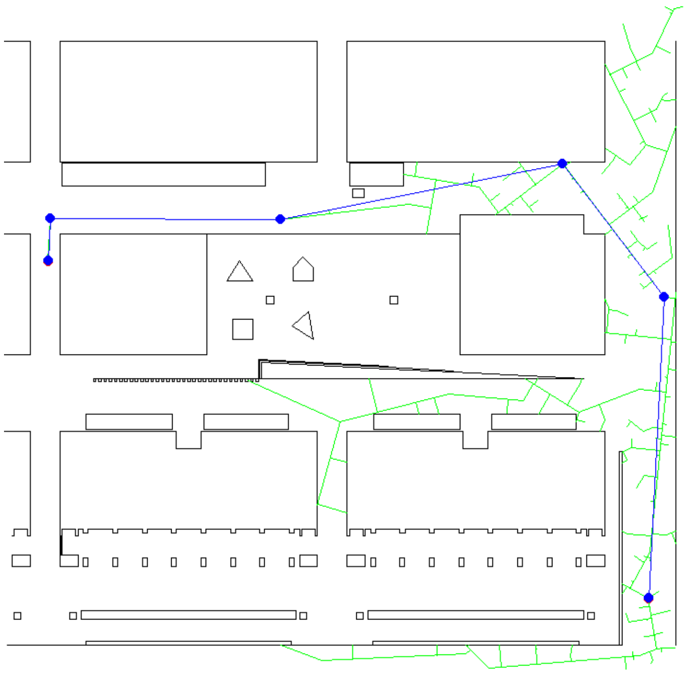

Figure 8: Path planning using Rapidly-Exploring Random Trees (RRT)

5 Conclusions and Future Work

In this technical report we have identified six key requeriments of a spatial representation for mobile robot map-based navigation: scalability, accuracy, flexibility, 3D extending, automatic conversion from an existing mapping source and human compatible representation.

The proposed map model fulfills the six requeriments exposed on section 2. The representation is scales up to large environments (such as the modelled one) thanks to the implicit 'metric/topologic grouping' when using geometric entities. In the exposed example (see section 4), an area of about $10000m^2$ is described on a file of $30KB$. Without any data compression or encoding, the compactness figure is about $3Bytes/m^2$. In terms of accuracy, the error on robot position comes from the mapping method and the sensor limitations but not from the environment model arrangement, so the proposed representation can provide robot localization up to mapping errors and sensor inaccuracies. Discussion about flexibilty can be more subjective but we have experienced global localization, path planning and position tracking using the exposed map format, proving that the same representation can be used by multiple tasks. Flexibility in terms of using this representation in robots equipped with different sensors is assured for sensors providing metric data but is not assured if sensors only provides appearance data. The map file can be edited with a GIS editor and then automatically converted to the map file format required for the robot. This implies that the robot and a GIS server can efficiently comunicate sending parts of the map and georeferenced data. However, automatic map building with a robot of such a map remains an open question. The easy communication with a GIS server offers us the possibility to use friendly and well proved human-machine interfaces to deal with geographic data.

Two simulated experiments on global localization and path planning are presented using an area of about $10000m^2$ represented with the proposed map format. The results of these experiments encourages us to use this environment model in map-based autonomous navigation for large environments like an university campus or an urban quarters.

As future works, we are working implementing ramp model to complete the map in order to perform field experiments. These field experiments will evaluate the actual possibilty of using the proposed map for map-based autonomous navigation in urban areas, or if an exhaustive extension to three dimensional data is imperative for navigation purposes in outdoor urban environments. In the perception side, we are working on using a laser scanner mounted in a tilt unit in order to beneficiate of 3D perception data to be matched with the 'pseudo' 3D data of the proposed environment model. Matching other 3D sensors like stereo cameras with these map data is also an open question.

We are also planning how to incorporate visual information to the representation entities such as shapes and lines to enhance flexibility of the map format since appearance data could help navigation tasks. However, incorporation of appearance data should keep the compactness requeriment.

References

[1] Arras, K.O., Castellanos, J.A., Schilt, M., Siegwart, R. 'Feature-based multi-hypothesis localization and tracking using geometric constraints'. Journal of Robotics and Autonomous Systems, Vol. 44, p. 41-53. 2003.

[2] Arras, K.O., Philippsen, R., Tomatis, N., de Battista, M., Schilt, M., Siegwart, R. 'A Navigation Framework for Multiple Mobile Robots and its Application at the Expo.02 Exhibition'. Proc. of the 2003 IEEE International Conference on Robotics and Automation - ICRA. Taipei, Taiwan. September 2003.

[3] Beeson, P., Muraka, A., Kuipers, B. 'Adapting Proposal Distributions for Accurate, Efficient Mobile Robot Localization'. Proc. of the 2006 IEEE International Conference on Robotics and Automation - ICRA. Orlando, Florida, USA. May 2006.

[4] Chen, M., Sohn, T., Chmelev, D., Haehnel, D., Hightower, J., Hughes, J., LaMarca, A., Potter, F., Smith, I., Varshavsky, A. 'Practical Metropolitan-Scale Positioning for GSM Phones'. Ubicomp 2006, September 2006.

[5] Filliat, D., Meyer, J-A., 'Map-based Navigation in Mobile Robots. I. A review of localization strategies', Cognitive Systems Research, Vol 4, Issue 4, December 2003.

[6] Fox, D., Burgard, W., Thrun, S. 'Markov Localization for Mobile Robots in Dynamic Environments', Journal of Artificial Intelligence Research, 11. pages 391-427. 1999.

[7] Gärtner, B. 'Fast and robust smallest enclosing balls'. Proc. 7th annu. European Symposium on Algorithms (ESA99). 1999.

[8] Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., Savelli, F. 'Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy'. Proc. of the 2004 IEEE International Conference on Robotics and Automation - ICRA. New Orleans, LA, USA. 2004.

[9] Kwok, C., Fox, D., Meila, M. 'Adaptive Real-Time Particle Filters for Robot Localization'. Proc. of the 2003 IEEE International Conference on Robotics and Automation - ICRA. Taipei, Taiwan. September 2003.

[10] Latombe, J.C. 'Robot Motion Planning'. Kluwer Academic Pub., 1991.

[11] LaValle, S.M. 'Planning Algorithms'. Cambridge University Press, 2006.

[12] Maantay, J., Ziegler, J. 'GIS for the urban environment'. Environmental Systems Research inst. 2005.

[13] Sanfeliu, A. and Andrade-Cetto, J. 'Ubiquitous networking robotics in urban settings'. In Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems, pages 14-18, Beijing, October 2006.

[14] Simmons, R. and Koenig, S. 'Probabilistic Robot Navigation in Partially Observable Environments'. In proceedings of IJCAI-95. 1995.

[15] Thrun, S., Gutmann, J., Fox, D., Burgard, W. and Kuipers, B. 'Integrating topological and metric maps for mobile robot navigation: A statistical approach' Proc. of the National Conference on Artificial Intelligence (AAAI), 1998.

[16] Thrun, S., Fox, D., Burgard, W., Dellaert, F. 'Robust Monte Carlo localization for mobile robots', Artificial Intelligence, 128. 2001.

[17] Thrun, S. 'Robotic Mapping: A survey'. In G. Lakemeyer and B.Nebel editors. Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, February 2002.

A Map File Format

The map is encoded in a .txt file as follows, and some parts of the file describing the presented environment are listed in the next page:

$border;$
$mx_1; my_1;$
$mx_4; my_4;$
#
$obstacle;$
$id^1; ST^1; NS^1; semanticInfo^1;$
#
$ax_1^1; ay_1^1;$
$bx_1^1; by_1^1;$
$h_1^1; inOut_1^1; semanticInfo_1^1;$
#
.
.
#
$ax_{NS^1}^1; ay_{NS^1}^1;$
$bx_{NS^1}^1; by_{NS^1}^1;$
$h_{NS^1}^1; inOut_{NS^1}^1; semanticInfo_{NS^1}^1;$
#
.
.
#
$obstacle;$
$id^{NB}; ST^{NB}; NS^{NB}; semanticInfo^{NB};$
#
$ax_1^{NB}; ay_1^{NB};$
$bx_1^{NB}; by_1^{NB};$
$h_1^{NB}; inOut_1^{NB}; semanticInfo_1^{NB};$
#
.
.
#
$ax_{NS^{NB}}^{NB}; ayNB_{NS^{NB}};$
$bx_{NS^{NB}}^{NB}; by_{NS^{NB}}^{NB};$
$h_{NS^{NB}}^{NB}; inOut_{NS^{NB}}^{NB}; semanticInfo_{NS^{NB}}^{NB};$
#

```
border;
-4;58;
58;-4;
#
obstacle;
1;1;4;column,A5;
#
4.53;12.10;
7.29;12.10;
5;1;wall,orange,brick;
#
7.29;12.10;
7.29;13.86;
5;1;wall,orange,brick;
#
7.29;13.86;
4.53;13.86;
5;1;wall,orange,brick;
#
4.53;13.86;
4.53;12.10;
5;1;wall,orange,brick;
#
obstacle;
2;1;4;column,A5;
#
8.12;12.10;
8.86;12.10;
5;1;wall,orange,brick;
#
8.86;12.10;
8.86;13.33;
5;1;wall,orange,brick;
#
8.86;13.33;
8.12;13.33;
5;1;wall,orange,brick;
#
8.12;13.33;
8.12;12.10;
5;1;wall,orange,brick;
#
.
.
.
obstacle;
13;1;4;bench,A5;
#
7.73;4;
40.66;4;
0.5;1;wall,grey,cement;
#
40.66;4;
40.66;5.30;
0.5;1;wall,grey,cement;
#
40.66;5.30;
7.73;5.30;
0.5;1;wall,grey,cement;
#
7.73;5.30;
7.73;4;
0.5;1;wall,grey,cement;
#
.
.
.
obstacle;
26;5;4;step,FIBsquare;
#
34.65;40.72;
57.5;40.72;
0;1;wall,grey,cement;
#
57.5;41.02;
34.65;41.02;
2;1;wall,grey,cement;
#
57.5;41.32;
34.65;41.32;
1.8;1;wall,grey,cement;
#
57.5;41.62;
34.65;41.62;
1.6;1;wall,grey,cement;
#
.
.
.
```