# Geographical Information Systems for Mobile Robotics Map Based Navigation in Urban Environments

Josep M. Mirats Tur [a,*], Claudio Zinggerling [b],
Andreu Corominas Murtra [a]

[a] *Institut de Robòtica i Informàtica Industrial, IRI (CSIC-UPC).*
*C/Llorens i Artigas 4-6, 08028, Barcelona, Spain.*

[b] *Centre Internacional de Mètodes Numèrics en l'Enginyeria, CIMNE.*
*Edif. C-1 Campus Nord UPC, C/Gran Capità s/n, 08034, Barcelona, Spain.*

---

## Abstract

In order to solve most of the existing mobile robotics applications, the robot needs some information about its spatial environment encoded in what it has been commonly called a *map*. The knowledge contained in such a map, whichever the approach used to obtain it, will mainly be used by the robot to have the ability to navigate in a given environment, that is, to reach any goal from any start point while avoiding static and dynamic obstacles. We are describing in this paper a method that allows a robot or team of robots to navigate in large urban areas for which an existing map in a standard human understandable fashion is available. As detailed maps of most urban areas already exist, it will be assumed that a map of the zone where the robot is supposed to work into is given which has not been constructed using the robot's own sensors. We propose in this paper the use of an existing Geographical Information System based map of an urban zone so that a robot or a team of robots can connect to this map and use it to navigation purposes. Details of the implemented system architecture as well as a position tracking experiment in a real outdoor environment, a University Campus, are provided.

*Key words:* map-based navigation, mobile robotics, Geographical Information Systems (GIS)

---

* Corresponding author.
*Email addresses:* `jmirats@iri.upc.edu` (Josep M. Mirats Tur), `czingger@cimne.upc.edu` (Claudio Zinggerling), `acorominas@iri.upc.edu` (Andreu Corominas Murtra).

## 1 Introduction

Mobile robotics has become a vast research area in the last years due to the great interest and expectation created around mobile robots. They are expected to provide capacities of working in remote [6] or harmful environments [11], where humans can not enter because of the extreme environmental conditions, as well as substitute or assist people in daily and tedious tasks such as cleaning [34] or merchandise transportation [13].

Whichever the application sought, with the exception of those being accomplished with random movements as well as those designed for exploration purposes, and in order to success with the assigned task, the robot needs some information about its spatial environment, often encoded in what it has been called a *map*. It is clear that the robot needs the ability to navigate, defined in this context as the ability to reach any goal from any start point while avoiding static and dynamic obstacles. In other words, the robot must have some kind of knowledge about the relative position of other objects in the space they share, in order to, for instance, be able to interact with these objects (think for instance in manipulation or transportation applications) or to calculate a path to a desired position.

In order for the robot to have such spatial environment knowledge we mainly find two different and complementary approaches. The first one is based on the assumption that the robot does not know anything about the environment in which it will perform its tasks, so the robot must learn it before being capable of navigation. This is by far the most popular approach within the robotics community, trying to solve the problem of *map learning*, i.e., obtaining a suitable representation for the data acquired by the robot sensors during an exploration phase. The problem of map learning is very closely related to that of localization leading to a question of the kind: who was the first, the egg or the chicken?. A map is needed to perform the robot localization, but, on the other hand, it is necessary to know the robot's position in order to build the map. This challenge has been often referred to as the *SLAM* (Simultaneous Localization and Mapping) problem, existing a vast literature about it [18,1,30,9,19,20].

The second approach is based on the assumption that the robot will perform its tasks in an already known and hence somewhat mapped environment. Therefore, the robot is provided with an initial, surely partial and incomplete, knowledge about the environment which is actually contained in a map which has not been constructed using its own sensors. This kind of map based navigation is natural for humans: when visiting an unknown city, we just need an existing map in standard format in order to be able to navigate as defined above.

Two kinds of map representations have been commonly used during years, topological and metric maps; although the trend is to combine both approaches leading to a vast class of hybrid maps [4], see [8] for some discussion and [12] for a review. In topological maps the environment is stored by linked nodes. They contain distinctive places the robot can reach [16,17] and the connexions between them, without metric information [33] in its basic form. Metric maps store unambiguous location of objects, usually in a 2D reference frame, which allow to precisely positioning them. These maps are easy to read and reuse although require an important amount of memory to be stored and make path-planning computationally expensive. In a metric map, objects may be stored from different points of view: they can be considered as punctual [10], as different points recorded from a surface [31], corners or points with an associated orientation [5,14] or lines defining polygonal boundaries [3,15]. Conversely, they can contain a free space representation, i.e., the portion of the environment that is accessible to the robot, instead of representing the objects in the map. This is the main idea behind occupancy grids [24,27].

Representations close to human maps offer advantages, specially in systems where human-robot interactions are expected or needed. Although the use of a map by a person requires high-level cognitive functions (mainly interpret the map, and establishing correspondence with the real world) partially solved for robots, a map model organized within a hierarchy and accepting semantic information will be more suitable to interface with humans. Spatial representations such as Geographical Information Systems based maps can use already developed interfaces as a powerful human-robot interaction tool, specially in environments like urban areas [21]. This paper adresses the challenge of using such standard format and human understandable maps for mobile robot navigation in outdoor scenarios, specifically in urban zones.

As detailed maps of most urban areas already exist, it will be assumed that a map of the zone where the robot is supposed to work into is given. We propose in this paper the use of an existing Geographical Information System based map of an urban zone so that a robot or a team of robots can connect to this map and use it to navigation purposes.

The organization of this paper is as follows. Section 2 includes a brief review on Geographical Information Systems. Next, section 3 explains the robot map format used. Then, the way in how the existing GIS map is automatically converted to a robot understandable format is explained in section 4. The designed architecture as well as some implementation details are given in section 5. Then, results obtained with current implementation of the presented method are shown by means of an experiment run in section 6. Finally, some conclusions and future work are outlined in section 7.

## 2 Geographical Information Systems

A Geographical Information System (GIS, from now on) can be defined as a geographic information handling technology that allows to manage a series of spatial data (geographic data) and to make complex analyses following the criteria imposed by the scientific personnel [35,26]. In recent years the importance of GIS have grown considerably since getting reliable geographic data is becoming day by day more essential and, because to obtain this data represents a large investment in economic and time terms.

Due to the nature of the information managed by GIS (geographic information plus semantic data), they are being used in most of the information management systems that depend or lean on the location of objects on predetermined surroundings [21]. Particularly, in robotics, a GIS may be helpful when it is desired to give autonomy and human-like resolution capabilities to a robot, or a team of robots, in non-controlled surroundings, i.e, with the main purpose that the robot or robots know the terrain where they will be moving and hence could anticipate or take decisions in advance [29]. Also, a GIS system would help to maintain control of the robot or robots indicating on a map their position accurately.

### *2.1 GIS main characteristics*

GIS evolve from the combination of classical paper maps and modern CAD systems. They have been developed with the intention of offering digital cartography integrated to alphanumeric information [25]. They are made up of layers of data represented in graphical form, where each layer presents or displays information related to a specific group of data (for example, parcels, blocks or streets).

There are diverse forms to model the connection between geographic and topological objects. Depending on the form in which this connection is carried out, two types of geographic information systems within the frame of two principal groups are defined: Vectorial GIS and Raster GIS [25,23]. The first group uses vectors defined by pairs of coordinates related to some cartographic system to compose each layer of the map. The second group displays and stores drawings or images like a dot matrix (cells).

The range of applications of GIS is widely open. It includes applications in management of environment, logistic, urbanism and transport or networks of distribution (water, energy and telephony) among others [21,26]. Because of the large number of GIS uses, the range of applications can be divided mainly into three classes:

- As a tool for the production of maps.
- Support for spatial analysis.
- Geographic data base, with functions of storage and space information retrieval.

According to existing literature, the cause that makes a system to be denominated like a GIS is the capacity to perform whether consultations or spatial analysis [26]. This means that any data or object within a GIS can geographically be located by means of its coordinates and/or by their descriptive attributes [25]. These descriptive attributes are a powerful tool, since they can relate each existing geographic object to an alphanumeric data set, representing this another of the fundamental characteristics of the GIS [25,23].

### *2.2 GIS Implementations*

Nowadays there are several types of available GIS, in the form of commercial software as well as open source codes. Esri's ArcGIS [36] and Intergraph GeoMedia [37] are two of the up to date most used commercial applications. Within the Open Source applications we can find MapServer [38] and Grass [39] which are widely used by scientists and the general user. As a matter of curiosity we outline the following:

- ArcGIS is an integrated collection of GIS software products for building a complete GIS [36].
- GeoMedia provides a full suite of powerful analysis tools, including attribute and spatial query, buffer zones, spatial overlays, and thematics [37].
- MapServer is an Open Source development environment for building spatially-enabled internet applications; is not a full-featured GIS system, nor it does aspire to be. Instead, MapServer excels at rendering spatial data (maps, images, and vector data) for the web [38].
- GRASS is a GIS used for geospatial data management and analysis, image processing, graphics/maps production, spatial modeling, and visualization [39].

## 3 Robot map model

Our main objective is to be able to use, for robot navigation, human compatible maps as those provided by GIS. The application, such as the one in the URUS project [28], is envisaged for multirobot teams providing different services in a given area of a urban environment such as a university campus or a city quarter.

So we are interested in maintaining scalability up to large environments keeping in reasonable bounds both, memory and computational resources, since the application itself is designed to be used in potentially large scenarios. Compactness also gains importance when whole or parts of the map have to be sent through a communication network, as is the case of the current GIS based application. Hence, we want the robot map to fulfill the requeriments exposed on [7], that is: scalability, accuracy, flexibility, three dimensional description, automatic conversion from a GIS source and human compatible.

So we will describe now how spatial information is arranged to form the environment data model used by the robot to navigate, also called the spatial representation or the map. The representation is basically on the 2D plane, based on geometric entities and inspired from the 'GIS vector' format [21]. 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.

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

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 related semantic information, $semanticI^k$.

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

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.

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

*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.

Stairs (steps) and ramps, usually present in outdoor urban areas, pose to the mobile robot community a challenge since 3D information has to be taken into account (from sensors and the map) in order to deal with the 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 in the current implementation).

*Stairs* are modelled as a list of segments, like the other obstacles of the map. From downstairs, the first segment is oriented 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, representing the presence of an obstacle but with null height. Figure 1 shows the stairs obstacle model.

*Ramps* can also be modelled as obstacles with null height. Ramp borders are described with a closed polygon formed by the ramp projection to the 2D plane. Ramp orientation is parametrized with the normal vector to the surface. Figure 1 draws the ramp model.

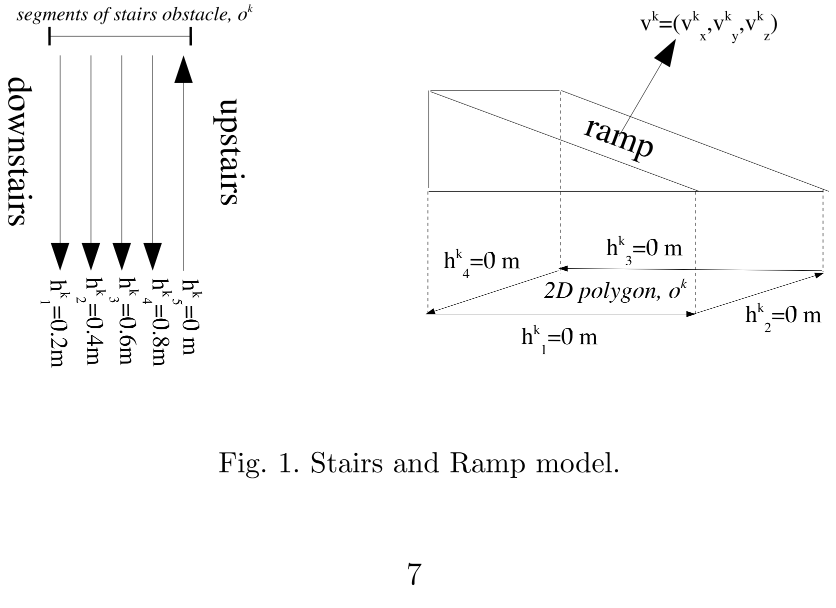


Fig. 1. Stairs and Ramp model.

Finally just note that the designed representation scales up to large environments (such as the modelled one) thanks to the implicit 'metric/topologic grouping' when using geometric entities. In the run experiment (see section 6), an area of about $10000m^2$ has been described on a file of $30KB$. Without any data compression or encoding, the compactness figure, which can be measured as a ratio of $bits/m^2$ [2], is about $3Bytes/m^2$.

## 4 From GIS to robot map

In the whole system, a computer acting as a GIS server contains a map of the whole urban area where the robot, robots, or different teams of robots are supposed to develop their tasks. Hence, given a GIS map from a determined urban zone, the first stage is to translate it to a robot understandable format.

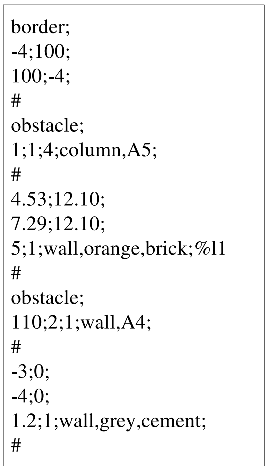
The GIS software is installed in the structure's server. One of its main functions is to manage, control and attend each one of the robots that have been contacted by the server. The server system has an agent which, in real time, directs each robot in the assigned tasks, such as provide maps of the nearby zone of each robot, program a route to follow, specify a destination point to itself, make an on screen localization (in real time) of the position of the robot, control possible anomalies or annex new elements to the cartography among others.

### *4.1 Map Generation*

One of the most important tasks that the GIS server performs is to provide a robot map of a certain area centered on the actual position of the robot from a general GIS map of the area where the robot or team of robots are performing their duties. This robot obtained map is obtained converting a standard GIS (vectorial) format to a text file where the geometry is translated to an understandable format by the robot as shown in figure 2.

Since GIS is a multilayer system, each of the layers loaded in the general map must be checked to determine which elements are of interest to generate the partial map that will be sent to the robot (only layers of area and lines will be used). Thanks to one of the main characteristics of GIS, each object being associated to alphanumeric data, the information related to each element is used to group the geometry.

The robot map is provided at the request of a given robot under two determined circumstances:



```
border;
-4;100;
100;-4;
#
obstacle;
1;1;4;column,A5;
#
4.53;12.10;
7.29;12.10;
5;1;wall,orange,brick;%11
#
obstacle;
110;2;1;wall,A4;
#
-3;0;
-4;0;
1.2;1;wall,grey,cement;
#
```

Fig. 2. Example of the generated text file containing the map in a robot understandable fashion.

- When the robot has to auto locate itself for the first time and sends a command Ready to the server (Refer to table 2 for a description on the currently implemented robots requests). In this case it receives the whole map.
- When the robot requests a map directly by means of the GetMap command, (see table 2).

Once the map is obtained by coding the GIS map, it is sent towards the robot by means of the command SetMap(M), where M is the file coded in binary format.

### *4.2 Construction process of the partial map*

The map sent to the robot is automatically generated in the GIS server. The procedure is as follows:

- Delimitate the zone to be sent to the robot creating an area of H x W around the actual position of the robot.
- Check each layer of interest of the geometry that is within the noticeable area.
- Translate the geometry of each one of the objects within each layer to the text format.
- Add the alphanumeric information related to each object to be included in the map.
- Group the geometry by the type of obstacles.
- Build the map in text format and encapsulate it in binary format to be sent to the robot.

## 5 System definition and implementation

A server-client scheme has been designed, where the computer acting as GIS server provides map services to any robot client that needs it. Communication between the robots and the GIS server is done via point to point wireless ethernet connections. Communications are supposed to exist during the whole operation of the robots, it is beyond the scope of this paper the study of robot formations so as to maintain stable communications with the central GIS server.

Once the point to point wireless connection is set up between the robot and the GIS server, we have to provide means for information exchange between them. This has been accomplished by means of a Data Base Management System (DBMS). In our implementation we have used SQL (Structured Query Language), which is a computer language aimed to store, manipulate, and retrieve data stored in relational databases [22], so it is perfectly suited to implement a bridge between the GIS software containing the map and the robot algorithms used to navigate. So a DBMS using SQL is installed in the GIS server computer. Each of the robots is provided with a user and password identification in order to stablish the connection to the database. Employing user and password identification, appart from a security issue, gives the possibility to more than one robot to access the same GIS map.

From the robot's point of view, DBMS managament is provided by using a public C++ library called MySQL++ which may run under Linux or windows operating systems and can be easily integrated into the onboard robot's algorithms. The main required parameters for a robot to stablish a connection to a GIS server are: the name of the database, we may have more than one database in the same computer server, the host name or IP address of the DBMS server, the server address in private wireless network, the user name to log in under, and the password to use when logging in.

In order to maintain the system scalable, robust and flexible a *plug and play* procedure to add new robots has been designed. Suppose a new robot should be added to an already multi-robot team in order to aid for some task. The robot should notify its existence to the server in order to be able to use the GIS map, and hence sharing the spatial knowledge together with the rest of the robots. When the robot has checked for all its sensors and has started all the required internal algorithms, it sends a *Ready* signal to the server, together with a random generated number, and then waits for server acknowledge (which should be the same robot random generated number) and assigned identity tag. The robot will then use this assigned *id* to communicate with the server until it finishes its tasks. Also, in order to mantain an active status of a robot into the server system, each robot sends periodically an *alive* message.

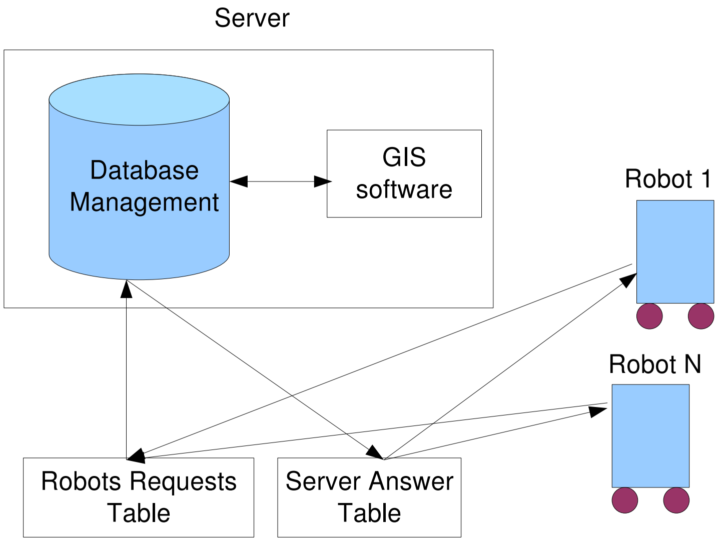


Fig. 3. Simplified description for the designed communication architecture between robots and GIS server.

Now, a connection has logically been stablished between a robot and the GIS server and information exchange between them may proceed. This is accomplished by the use of two tables, one for robots requests and another one for server answers. Different procedures are designed for operation with both tables. For the robots requests table ($RR$) only robots are allowed to write, while the server may perform read and delete operations. Each $RR$ table row will contain a unique robot request. The procedure is as follows, a robot requests a specific operation, see table 2, so a new entry in the $RR$ server table is generated. The server, then reads the new entry and whether processes or stores it in secondary tables not accessible from the robots for internal computations (think for instance in having implemented a global path planning algorithm in the GIS server). Then, the server delete this entry from the $RR$ table and may proceed to process the next robot request.

Table 1
Permitted operations for robots and server.

| | **Robots Requests Table** | **Server Answers table** |
|---|---|---|
| **Read** | Robot | Server |
| **Write** | Server | Robot |
| **Delete** | Server | Robot |

On the other hand, the server answer table, $SA$, can only be written by the server while robots can only perform read and delete operation on it. Every row of the $SA$ table will contain a unique answer which will be delete by the corresponding robot once read. Table 1 summarises the permitted actions for robots and server with both tables while tables 2 and 3 summarise the currently implemented functions.

Table 2
Currently implemented functions for robots requests.

| **Operation code** | **Parameters** | **Description** |
|---|---|---|
| Ready | Magic number | Indicates the robot is ready to start tasks and listen to the $SA$ table |
| Sttoped | | Robot is stopped |
| Alive | | Robot is alive, periodically sent |
| OpAck | | Robot acknowledges for the received operation from the server |
| SendPos | x,y,$\theta$ and robot velocity | Robot sends position estimation to server. To be sent periodically |
| SendGPS | Lat, Long | Robot sends GPS Latitude and Longitude information |
| SendImage | I - binary image, x,y and ID | Robot sends a binary encoded image (I) grabbed at position (x,y) associated to an object map (ID) |
| GetMap | x,y, A | Robot asks for a new map centered on its actual (x,y) position and with area A |
| GetPath | | Robot asks for a new path to complete the actual task. The server knows actual position and destination |
| GetPosIni | | Robot asks for an initial position of a track |
| GetPosEnd | | Robot asks for an end position of a track |
| SetObject | x,y, ID | Robot has observed an object identified as ID in position (x,y) that has to be included within the map |

Just as an example of how these commands may work, consider the case of controlling and tracking the robot's position. The robot indicates to the GIS server its position at every moment through the communication protocol by means of the command SendPos. When the appropriate agent on the server detects an entrance of this command, draws the position in the GIS map. The system draws and stores the trajectory of the robot from the origin to the actual location to verify and to control the tasks made by each robot at every moment. In the case that one robot has left its original trajectory or if it is desired to reassign a new one, there is the possibility of aborting the task assigned to the robot by means of the command Abort. To indicate a new destination it will be done by the command SetPosEnd (X,Y).

Table 3
Currently implemented functions for server answers.

| **Operation code** | **Parameters** | **Description** |
|---|---|---|
| Start | | Server is ready to listen to the $RR$ table |
| OpAck | | Server acknowledges for the received request from the robot |
| Abort | | Server indicates the robot that must abort the current task. User supplied for supervisory purposes |
| SetMap | M | Server has sent the robot the requested map. M is the binary coded file sent |
| SetPath | $(x_i, y_i)$ | Server sends the robot a new path to complete the actual task |
| SetPosIni | $(x_{ini}, y_{ini})$ | Server gives the robot an initial position of a track to be run |
| SetPosEnd | $(x_{end}, y_{end})$ | Server gives the robot an end position of a track to be run |

## 6 Experiments

This section shows an example of using this map-based navigation framework to deal with a navigation task as it was previously defined in the introduction section. The used map describes the surroundings of the Computer Science School at the Campus Nord of the Universitat Politcnica de Catalunya (UPC), representing an outdoor environment of about $10000m^2$ which will be the test bench scenario for the European URUS project [28]. Figure 4 shows the geometric part of the map for this environment, the origin of the metric coordinates and as well as the coordinates for two illustrative selected points. Blue arrows in the figure mark places where pictures shown in figure 5 were taken in order to familarize the reader with the environment.

Our interest with this experiment is twofold, first to evaluate whether or not the robot is able to navigate in urban environments using a human-like map which has not been constructed using its own sensors, second to test the system implementation as defined in the previous sections. For this aim a position tracking experiment has been designed and run at the Campus Nord of the UPC. The experiment consisted in performing a close loop around building A5 (see figure 4), accounting for a path lenght of approximately 150 meters. The employed tracking method was a particle filter inspired on the work reported in [32]. The odometry of the robot was used to propagate the particles while a 2D Leuze Rotoscan Laser scanner was used to compare the information stored in the map with the observations the robot took from the environment; this comparison was used for each particle in order to update the probability of the considered position hypothesis. Results of this experiment are shown in figures 6 and 7 where some snapshots representing the position of the robot seen respectively by the GIS server and the robot itself are shown.

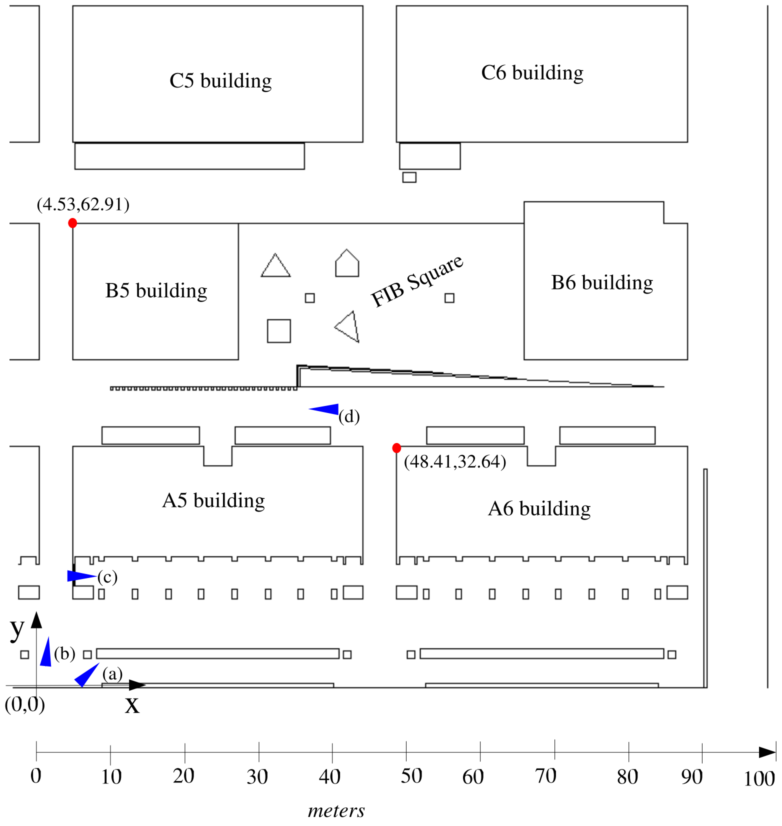


Fig. 4. Geometric part of the map of the surroundings of the 'FIB Square' at the Campus Nord of the Universitat Politècnica de Catalunya (UPC)

Fig. 5. Pictures taken from a,b,c,d points on figure 4

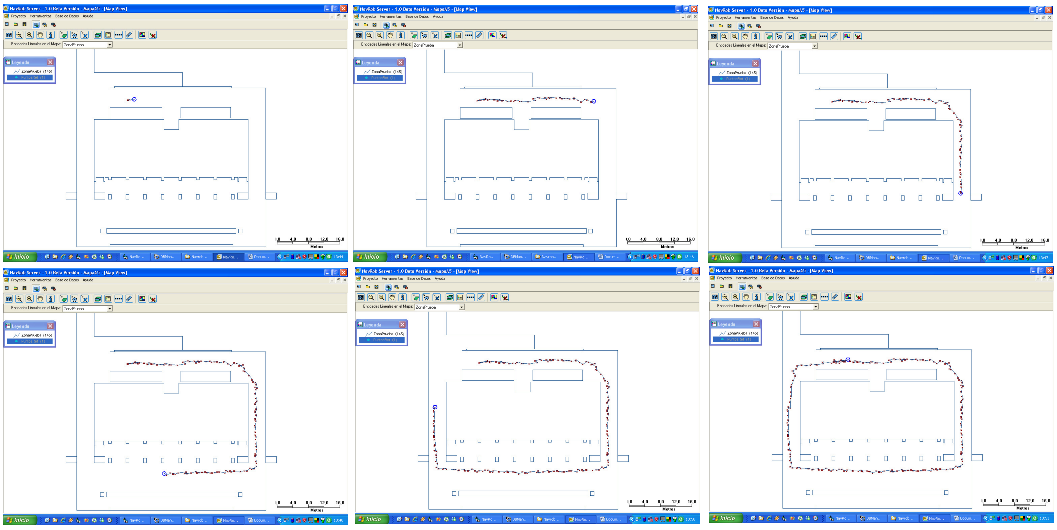


Fig. 6. Robot reported position within the GIS server

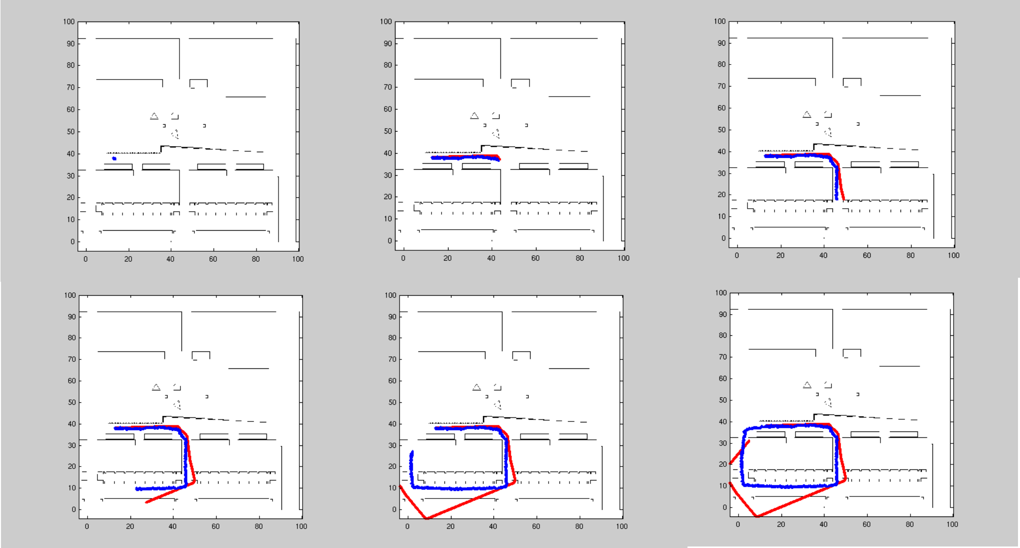


Fig. 7. Robot position seen locally by the robot. Blue line is for the output of the tracking filter while red line is pure odometry

## 7 Conclusions

More and more Society is demanding new robotics applications in which robot performance is closer and closer for humans to understand. If we think in the use of mobile robots in common human environments, as urban settings are, for common tasks, as may be for instance surveillance, merchandise delivering or garbage collection, we will see the need for a robot or team of robots solving a given task to use or re-use an existing map of a big urban-like area. Yet, better if this existing map can be in a human-understandable fashion, since we will surely need to have human-robot interaction in order to specify tasks and places to go, for instance in a big city quarter.

This is the issue we have mainly tackled in the present paper. A server-client architecture has been designed in which the server contains a GIS based map of the area, commonly large, where a team of robots is supposed to work, and the clients, actually a robot or team of robots, can connect to the general data base in order to retreive a part or the whole of the map needed to perform their tasks. Clients can, at the same time, provide up to date information in order to maintain updated the general data base. So, given a large area to be covered by a team of robots for which a human-like map format exists, we have provided a method that allow this team of robots to safely and robustly navigate in this large area. A real experiment in a common outdoor environment, a University Campus, has been performed with a single robot in order to validate the GIS based navigation system.

## Acknowledgments

This work has partly been supported by Cicyt Navrob project DPI2004-05414 of the Spanish Science and Education Ministry and European STREP project URUS with reference IST-045062.

## References

[1] Arleo A., J. Millan, D. Floreano, 'Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation', IEEE Transactions on Robotics and Automation, 15(6), 1999, pp. 990-1000.

[2] Arras K.O., R. Philippsen, N. Tomatis, M. de Battista, M. Schilt, R. Siegwart, '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] Austin D.J., B.J. McCarragher, 'Geometric constraint identification and mapping for mobile robots', Robotics and Autonomous Systems, 2001, Vol. 35, No. 2, pp. 59-76.

[4] Bailey T., E. Nebot, 'Localisation in large-scale environments', Robotics and Autonomous Systems, vol. 37, n. 4, pp. 261-281, 2001.

[5] Borghi G., D. Brugali, 'Autonomous map-learning for a multi-sensor mobile robot using diktiometric representation and negotiation mechanism', In Proceedings of the international Conference on advanced robotics (ICAR-95), pp.521-528.

[6] Clark C.M., S.M. Rock, and J.C. Latombe, 'Dynamic networks for motion planning in multi-robot space systems', Int. Symp. Artificial Intelligence, Robotics and Automation in Space, Nara, Japan, May 2003.

[7] Corominas Murtra A., J.M. Mirats Tur, 'Map Format for Mobile Robot Map-based Autonomous Navigation', Technical Report IRI-DT 2007/01. February 2007.

[8] Corominas Murtra A., J.M. Mirats Tur, A. sanfeliu, 'Spatial Representations in Mobile Robotics: from Requeriments to an Ontology-based Framework', submitted to ECMR07.

[9] Davison A.J., Y.G. Cid, and N. Kita, 'Real-time 3d slam with wide-angle vision' in Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, July 2004.

[10] Feder H., J. Leonard, C. Smith, 'Adaptive mobile robot navigation and mapping', International Journal of Robotics Research, 1999, 18(7), pp. 650-668.

[11] Ferguson D., A. Morris, D. Hähnel, C. Baker, Z. Omohundro, C. Reverte, S. Thayer, W. Whittaker, W. Burgard, and S. Thrun, 'An autonomous robotic system for mapping abandoned mines', In S. Thrun, L. Saul, and B. Schölkopf eds., NIPS. MIT Press, 2003.

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

[13] Gro$\beta$ R., E. Tuci, M. Dorigo, M. Bonani, and F. Mondada. 'Object transport by modular robots that self-assemble', In Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation, pages 2558-2564, IEEE Computer Society Press, Los Alamitos, CA, 2006.

[14] Grosmann A., R. Poli, 'Robust mobile robot localisation from sparse and noisy proximity readings using Hough transform and probability grids', Robotics and Autonomous Systems, 2001, Vol. 37, No. 1, pp. 1-18.

[15] Iocchi L., D. Nardi, 'Hough Localization for mobile robots in polygonal environments', Robotics and Autonomous Systems, 2002, Vol. 40, No. 1, pp. 43-58.

[16] Kuipers B., 'The Spatial Semantic Hierarchy', Artificial Intelligence, No. 119, pages 191-233, 2000.

[17] Kuipers B., J. Modayil, P. Beeson, M. MacMahon, F. Savelli, '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.

[18] Kunz C., T. Willeke, I. Nourbakhsh, 'Automatic mapping of dynamic office environments', In Proceedings of the IEEE international conference on robotics and automation, 1997, Vol. 2. pp. 1681-1687.

[19] Lemaire T., S. Lacroix, and J. Solà, 'A practical 3d bearing only slam algorithm', in IEEE International Conference on Intelligent Robots and Systems, August 2005.

[20] Levinson J., M. Montemerlo and S. Thrun, 'Map-Based Precision Vehicle Localization in Urban Environments', Proceedings of Robotics: Science and Systems, June 2007, Atlanta, GA, USA.

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

[22] Melton J., 'Understanding the New SQL: A Complete Guide', 1993, Morgan Kaufmann publishers, ISBN 1558602453.

[23] Moldes F.J., 'Tecnología de los Sistemas de Información Geográfica', Ra-Ma, 1995.

[24] Moravec H., A. Elfes, 'High resolution maps from wide angular sensors', In Proceedings of the IEEE international conference on robotics and automation (ICRA-85), IEEE Press, pp. 116-121.

[25] Moreno A., et al., 'Sistemas y análisis de la información geográfica. Manual de autoaprendizaje con ArcGIS', Ra-Ma, 2005.

[26] Pamuk A., 'Mapping global cities: GIS methods in urban analysis', Esri Pr., 2006.

[27] Ribo M., A. Pinz, 'A comparison of three uncertainty calculi for building sonar-based occupancy grids', Robotics and Autonomous Systems, 2001, Vol. 35, No. 3-4, pp. 201-209.

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

[29] Tao C. Vicent, J. Li, 'Advances in mobile mapping technology', Taylor and Francis, 2007.

[30] Thrun S., 'Learning metric-topological maps for indoor mobile robot navigation', Artificial Intelligence, 99(1), pp. 21-71, 1999.

[31] Thrun S., W. Burgard, D. Fox, 'A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping', In Proceedings of the IEEE international conference on robotics and automation, 2000, pp.321-328.

[32] Thrun S., D. Fox, W. Burgard, F. Dellaert, 'Robust Monte Carlo localization for mobile robots', Artificial Intelligence, Vol. 128, n. 1, pp. 99-141, 2001.

[33] Tylor C.J., D.J. Kriegman, 'Vision based motion planning and exploration algorithms for mobile robots', IEEE Transactions on robotics and Automation, 1998, 14(3), pp. 417-427.

[34] Yong-Joo Oh, Y. Watanabe, 'Development of small robot for home floor cleaning', Proc. of the SICE Annual Conf., Vol. 5, pp. 3222-3223, 2002.

[35] Fahui Wang, 'Quantitative methods and applications in GIS', CRC Pr I Llc. 2006.

[36] http://www.esri.com

[37] http://www.intergraph.com

[38] http://mapserver.gis.umn.edu/

[39] http://grass.itc.it/