Control and Cybernetics

vol. **40** (2011) No. 2

Mobile robot navigation with the use of semantic map constructed from 3D laser range scans*

by

Barbara Siemiątkowska[1], Jacek Szklarski[2] and Michał Gnatowski[2]

[1]Warsaw University of Technology, Warsaw, Poland

[2]Institute of Fundamental Technological Research, Polish Academy of Sciences Warsaw, Poland

Abstract: We describe a system allowing a mobile robot equipped with a 3D laser range finder to navigate in the indoor and outdoor environment. A global map of the environment is constructed, and the particle filter algorithm is used in order to accurately determine the position of the robot. Based on data from the laser only, the robot is able to recognize certain classes of objects like a floor, a door, a washbasin, or a wastebasket, and places like corridors or rooms. For complex objects, the recognition process is based on the Haar feature identification. When an object is detected and identified, its position is associated with the appropriate place in the global map, making it possible to give orders to the robot with the use of semantic labels, e.g., "go to the nearest *wastebasket*". The obstacle-free path is generated using a Cellular Neural Network, accounting for travel costs with distance or ground quality. This path planning method is fast and in comparison with the potential field method it does not suffer from the local minima problem. We present some results of experiments performed in a real indoor environment.

Keywords: artificial intelligence, robotics, mapping.

1. Introduction

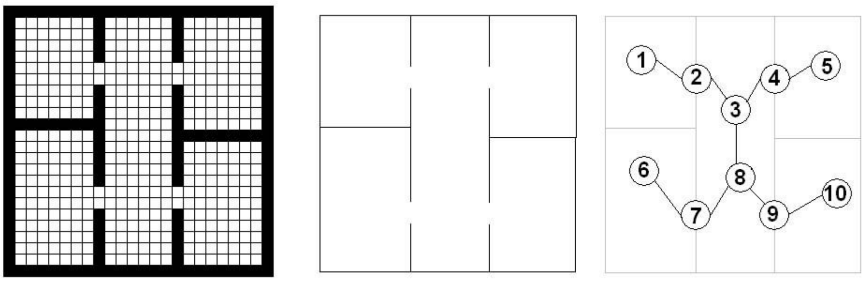
The ability to navigate is the most fundamental competence for a mobile robot. This task is defined as a combination of three fundamental elements: map building, localization and path planning.

Knowledge about the robot environment is usually encoded in a form of a map. Most methods focus on the following two categories:

- Metric maps, Thrun et al. (2005); Elfes (1987); Moravec and Elfes (1985), which represent some geometric features of the environment. One of the

*Submitted: October 2010; Accepted: June 2011

most popular geometric representation is the occupancy grid. The environment is represented as a grid of cells. Each cell is either occupied (a part of an obstacle) or free (a part of the free from obstacles space). This kind of representation allows us to combine information from various sensors in different positions of the robot. It also allows fast generation of a collision-free path. However, the method suffers from two main disadvantages: the size of the map grows with the size of the environment and the accuracy of the map largely depends on the size of a cell (consequently, if a very precise map of environment is required, then a huge amount of memory is necessary). For these reasons an alternative method, called the feature-based map, has been proposed. An example of this representation is shown in Fig. 1b. All obstacles are represented as a list of segments (Latombe, 1992). This kind of maps is attractive because of their compactness and they are very useful during the process of localization. However, the path-planning based on this kind of representation is time-consuming. Metric maps represent the location of obstacles without referring to non-geometrical features such as texture, color etc.
- Topological maps (Latombe, 1992; Remolina and Kuipers, 2004) represent relations between distinctive parts in the environment. Formally, it has a form of a graph - nodes are used to denote some areas or places in the environment, and arcs denote adjacency. Fig. 1c shows the topological map of the environment.



a) b) c)

Figure 1. Different maps of the environment: a) grid-based map, b) feature-based map, c) topological map

These two representations can be combined into a hybrid map, which contains both metric and topological information (Pfingsthorn et al., 2007).

The robot has to possess the semantic knowledge about the environment in order to improve its capabilities. Recently, researchers have been focused on the semantic maps that contain data not only about the geometry and relation between parts of the environment but also about the meaning of the detected objects (Rusu et al., 2008; Mozos et al., 2007; Siemiątkowska et al., 2009). Without this kind of information, the human-robot interaction is very difficult.

An environment can be divided into places: corridors, rooms, kitchens (inside a building), or grass, roads, etc. (outside of a building). The semantic labels attached to the places give not only information about the names, but also about functionalities: the doorway indicates the transition between different rooms, meal can be prepared in the kitchen etc. We assume that the robot has to recognize not only type of room in which it is located, but also objects of certain classes.

The mapping problem is strictly related to the localization. The robot needs to know its position in the environment in order to build a map and to perform a given task. The most widely used method is the odometry. It is inexpensive and provides a good short time accuracy, but errors in determining the position of the robot increase proportionally with the distance traveled by the vehicle. If the robot travels for a prolonged period of time additional localization methods should be applied. Usually the Kalman filter method is used to continuously estimate the robot position (Olson, 2000; Grewal and Andrews, 2001; Weingarten and Siegwart, 2005). In this method, the encoder readings are used as inputs and sensors measurements as observations. Determining the displacement of the robot in relation to the landmarks allows us to update the position of the robot in the environment. An alternative and efficient way of localization are the particle filters (Rekleitis, 2004; Fox, 2003). The key idea of the method is to represent the possible robot locations as a set of N samples (particles). Each sample consists of a pair (q, w), where q is a state vector - coordinates of a possible position of the robot, and w is a weighting factor, $w \in [0, 1]$. In the case of a human-robot interaction the robot has additionally to possess the semantic information about the places. For example, when the robot is asked to go to the kitchen and bring back a cup, it has to know its position, where the kitchen is and it has to the possess abilities to distinguish a cup from other objects.

The next step is to plan a collision-free path to the goal and to execute the task. The aim of the path planning is to find the optimum collision-free path between the starting position of the robot and the target location. Various methods are proposed to solve the problem (Latombe, 1992; Chu and Eimaraghy, 1992; Buckley, 1989). They can be classified as global or local. Global methods (Latombe, 1992; Bennewitz et al., 2000) require to have the map of the whole environment and are time consuming. When the local path planning algorithm (Buckley, 1989; Bennewitz et al., 2000; Barraquand et al., 1992; Azarm and Schmidt, 1996) is used, then only the information about obstacles in the robot vicinity is taken into account. Although the method is fast, it can converge to a local minimum and will not provide the correct solution.

The problem of finding the optimal collision-free path is strictly related to the type of the environment. In the case of indoor navigation the optimum path is the safest path, the robot has to move far from the obstacles. The distance traveled by the robot is less important than in the case of the outdoor navigation. When the robot moves outside of the building, the cost of traveling depends on the type of the ground. It is less expensive, in terms of time and energy, to move on a road than on grass.

In this article, the system which allows the robot to navigate in the outdoor and indoor environment is presented.

Data obtained from a 3D laser range scanner are analyzed and semantic labels are attached to the detected objects and places (Fig. 2 shows an example of such a map). The environment is represented as a grid of cells and a list of semantic labels is attached to each cell. The robot finds the optimal collision-free path based on the geometric and semantic information stored in the map. The goal for the robot is described using semantic labels. It is possible to ask the robot to move towards the door or to the washbasin. In the case when the same label is attached to many objects, the least expensive path is found automatically.

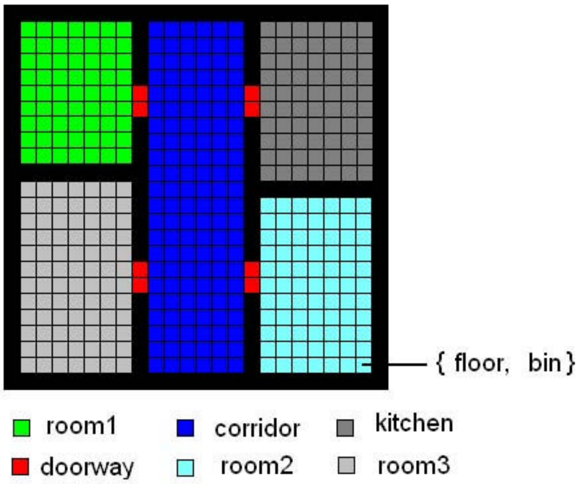


Figure 2. The semantic grid-based map of the environment

2. Perception and object classification

Perception of the environment (Siemiątkowska et al., 2009) is one of the crucial problems in mobile robotics. During the path-planning process, a mobile robot must be able to detect certain classes of objects and landmarks. This makes possible to estimate the correct position of a robot, as well as to identify its goals.

One of the most common ways to perceive the environments is to use visual CCD cameras. Images from such cameras can be used to detect objects on the basis of, e.g., appearance (Kröse et al., 2007) and to estimate robot position and object size when using the stereoscopic vision. There are, however, some limits related to this approach. For example, light conditions vary significantly, making detection process difficult and often unreliable geometric information calculated from stereo-vision is limited due to large errors.

Recently, due to cost decrease, the laser range finder scanners gain more and more popularity in the field of mobile robotics. Such scanners can give 2D information about the distance in a single plane or full 3D data. It is known that data from 3D scanners can be used to detect and classify wide range of objects (Chen and Medioni, 1991). One of the most common methods is to use the well-known ICP algorithm (Besl and McKay, 1992). In this paper we describe another approach, which is essentially based on the image analysis field. Similar work has been considered with the use of images representing distance and reflectance (Nüchter et al., 2004, 2005). One of the main advantages when compared to classic methods is its speed and low memory consumption.

The experiments described here have been performed with the help a mobile robot "Elektron", built at the Institute of Automatic Control and Robotics of Warsaw University of Technology. The basic sensor is the Sick LMS 200 indoor laser mounted on a rotating support enabling a 3-dimensional representations of the environment. The head can rotate the scanner around the horizontal axis within the angular range θ from -15° to +90° (collecting a single 3D scan in such setup can take as long as 60s). The scanning laser enables us to measure the distance to obstacles within $-90° \leq \phi \leq 90°$ with the resolution of 0.5°. The device provides measurements as a set of 3-tuples $\{\phi_i, \theta_i, r_i\}$ where ϕ and θ represent the horizontal angle of the laser ray and vertical inclination angle of the laser base, respectively, and r_i is the measured distance. The usual next step of data analysis is to transform these values into a point cloud, which is a set of 3D points in the Cartesian coordinate system with the robot at its center. However, we propose here a rather different and novel approach in which we convert the measurements into a 2D image and then apply fast and well-known algorithms used in image analysis.

2.1. Image construction and object recognition

The most straightforward way to transform data from the laser scanner is to use (ϕ, θ) as pixel coordinates and assign pixel color according to the measured distance. However, since this approach does not lead to a satisfactory method of representing the geometrical properties of the environment, we propose to map three coordinates associated with normal vector for each pixel to RGB values of an ordinary color image. Using simple trigonometry for each pixel (i, j), we obtain its position p in 3D Cartesian coordinates with the robot at its center (a standard procedure for point cloud methods). Then four neighboring points $p_{1,\ldots,4}$ with $(i \pm 1, j-1), (i \pm 1, j+1)$ are considered (alternatively we take into account more points, though it does not give any improvement to the overall procedure). Let $\mathbf{p}_{1,\ldots,4}$ be vectors pointing from p to $p_{1,\ldots,4}$, accordingly. If a point p_n is too far or too close to p, including it in the calculation might lead to spurious errors. Therefore all vectors $\mathbf{p}_n$ whose length is not fulfilling the inequality

$$\epsilon_0 \leq |\mathbf{p_n}| \leq \epsilon_1,$$

are rejected; the thresholds ϵ_0 and ϵ_1 are, for our laser, 0.5 cm and 30 cm respectively. The normal vector $\mathbf{n}'$ is calculated as

$$\mathbf{n}' = \mathbf{p_1} \times \mathbf{p_2} + \mathbf{p_2} \times \mathbf{p_3} + \mathbf{p_3} \times \mathbf{p_4} + \mathbf{p_4} \times \mathbf{p_1},$$

where $\times$ is the cross product, $\mathbf{n}'$ is normalized afterwards, $\mathbf{n} = \mathbf{n}'/|\mathbf{n}'|$. A color RGB image is constructed by assigning values of the coordinates $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$ as colors red, green and blue accordingly. Such images are later used in the object detection process.

In order to detect objects of interest on an RGB image, and place them onto a global, semantic map we distinguish two procedures:

- Rule-based identification of areas: after the simple segmentation, a rule based classifier is applied in order to detect objects like the floor, doors or grass (outdoor)
- Object identification with the Haar features: for more complex objects, we use a classifier based on the Haar features. Each single classifier is trained for the detection of one class of objects.

Details concerning the above methods will not be presented here, and can be found in our other papers, Borkowski et al. (2010) or Gnatowski et al. (2010). A sample point cloud and the corresponding RGB image for an indoor scene is depicted in Fig. 3.

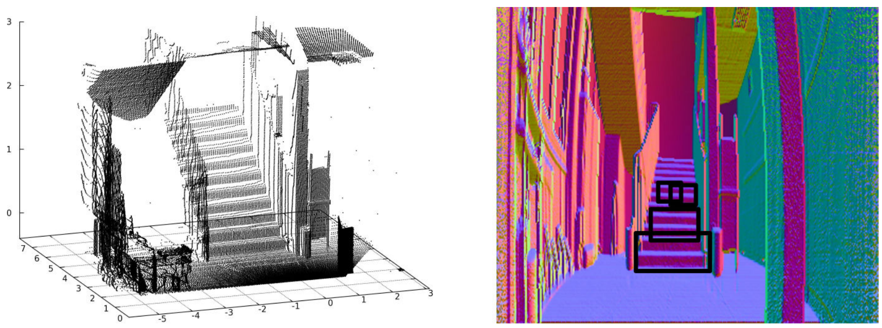


Figure 3. *Left:* A point cloud representing a single scene in the indoor environment. The robot is located at position $(0, 0)$ and faces towards the stairs. *Right:* The RGB image constructed from the corresponding point cloud. Black rectangles denote regions classified as "stairs" by the Haar features classifier

3. Map building and place classification

In our method a map represents both geometric and semantic features of the environment.

If the geometric representation is used, it is assumed that the robot operates in an R^2 or R^3 space W which is called workspace. This space contains objects $O_i \in W$, O_i is the set of points occupied by the object i. In the proposed in this paper algorithm i-th object is described by the following parameters: (s_i, u_i^o, r_i^o), where s_i is the semantic label, e.g. *chair, table, wall* etc., $u_i^o \in R$ represents the traversability level of the object, $r_i^o \in R$ is the radius of the influence of the object. Parameters u_i^o, r_i^o are used during the path planning and are described in Section 5.

3.1. Indoor environment

In our approach we suggest to use the Hough transform (Ballard, 1981; Duda and Hart, 1972) in order to distinguish the corridor from other kinds of places. For places that are not as easy to define as corridor, the classification is essentially based on the semantic labeling of objects which are recognized. For example we can find a washbasin in a bathroom, a computer in a laboratory and doors near the doorway.

The Hough transform is a technique for identifying the locations of certain types of features (usually segments) in a digital image. In the image space, the straight line can be described as:

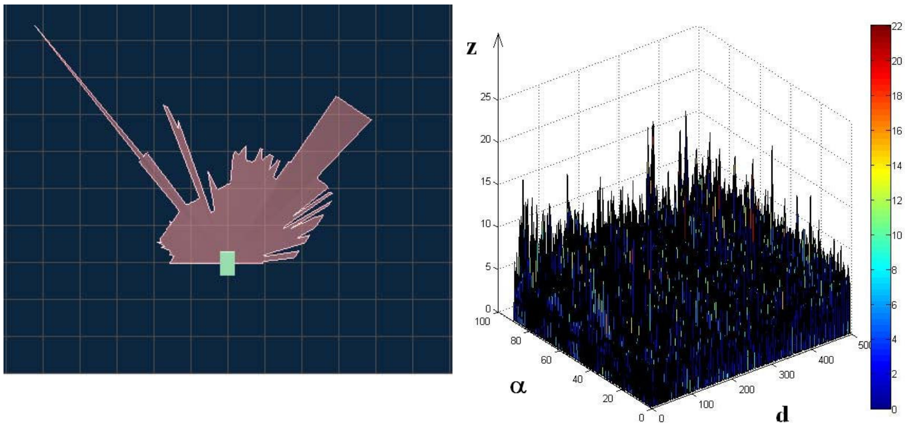
$$x \cos\alpha + y \sin\alpha - d = 0, \tag{1}$$

where the parameter α is an angle between a segment perpendicular to the line and OX axis, and parameter d is the shortest distance between the origin and the line.

The transform is implemented by quantizing the parameter space (α, d) into finite intervals (accumulator cells). As the algorithm runs, each point (x, y) "votes" for the family of lines it belongs to and the corresponding accumulator cells are incremented. Resulting peaks represent the longest segments in the image. An important advantage of the Hough transform is its tolerance towards noise and holes in the boundary line.

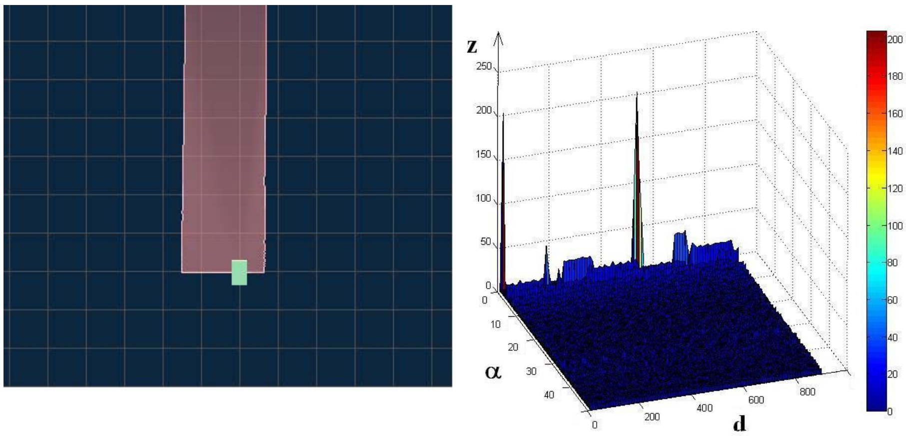
The data obtained from 2D laser range finder can be presented as an image. Fig. 4a presents the laser reading obtained in a room and Fig. 5a data taken in the corridor. If the Hough transform is applied to data obtained in a corridor, there are two evident peaks which represent walls. In the case of rooms there are many peaks. Fig. 4b presents the Hough transform of 4a and Fig. 5b the Hough transform when the data is taken in the corridor(Fig. 5a).

If the area is not classified as a *corridor* then the semantic label is attached based on classes of objects, which are observed. For example *laboratory* is the area which contains chairs, desks etc. There are stairs in the hall, and hand



a) b)

Figure 4. Hough transform of the data in a room: a) Data taken from 2D laser range finder, b) Hough transform, **Z** represents the number of votes for line described by parameters (d, α)



a) b)

Figure 5. Hough transform of the data in a corridor: a) Data taken from 2D laser range finder, b) Hough transform, **Z** represents the number of votes for line described by parameters (d, α)

basins in a toilet. Before we start classifying places in the indoor environment, it is necessary to classify objects (see Section 2.1).

3.2. Outdoor environment

Generally, in our approach we distinguish two classes of the ground area: a side walk and a grass. The classification is performed based on the "roughness" of a terrain.

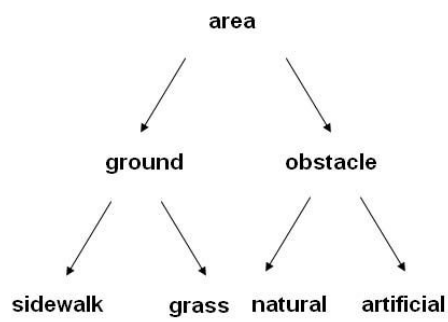


Figure 6. Places in an outdoor environment

As the measure of the "roughness" R, for each point (i, j) we use:

$$R_{i,j} = (w^2 - 1)^{-1} \sum_{-w \le k \ne 0 \le w} \sum_{-w \le l \ne 0 \le w} \mathbf{n}_{i,j} \cdot \mathbf{n}_{k,l}, \qquad w > 1,$$

where w is the width of a window for which the averaged dot product of the normal vector at (i, j) and its neighbors is calculated. Then, for each area A, one can further divide it into smaller areas A', squares having the width of w'. For each A', its averaged „roughness", $\bar{R}$ is calculated. For example, for the outdoor environment, $w = 2, w' = 5$, an area for the sidewalk has $L\bar{R} \approx 1$, whereas the region covered by grass has $\bar{R} \approx 0.95$. The value of $\bar{R}$ gives information about the traveling cost through A' for the mobile robot. This cost is later used in the path planning algorithm. Results of the classification for a sample outdoor scene are shown in Fig. 7.

Similarly, natural and man-made elements of the environment can be distinguished with this approach (e.g., buildings with flat walls and trees).

4. Localization

We assumed that the robot is placed initially at the point q_i, $i = 0$, $q_0 = (x_0, y_0, \theta_0)$, where $(x_0, y_0) \in W$ and θ_0 describes the orientation of the robot in the global coordinate system. The data taken at q_0 are analyzed and represented

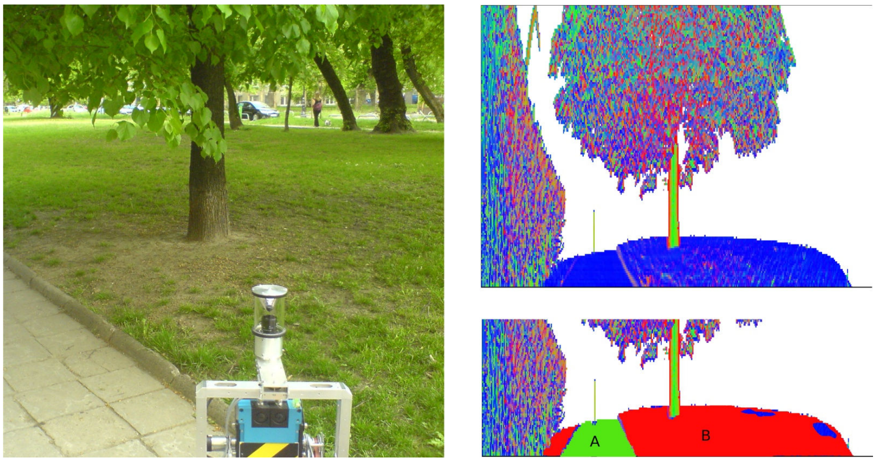


Figure 7. *Left:* Photo of an outdoor scene. *Right:* top: an RGB image representing normal vectors constructed with the use of data from laser range finder, bottom: result of a rule-based classification, A-sidewalk (a flat terrain with low travel cost), B-grass (a terrain with average „roughness" coefficient $\bar{R} \approx 0.95$; travel cost is higher than for the sidewalk)

in the form of the map. When the robot moves to the next position q_{i+1}, it gathers data from the laser scanner, which are then transformed into a point cloud which, in turn, is analyzed and transformed from the local into the global coordinate system. To perform this task the robot has to know its position q_i, so it has to find the following values: (Δx, Δy, $\Delta\theta$), where:

$$\begin{aligned} \Delta x &= x_{i+1} - x_i, \\ \Delta y &= y_{i+1} - y_i, \\ \Delta\theta &= \theta_{i+1} - \theta_i. \end{aligned} \tag{2}$$

A sample transformation for two consecutive scans is depicted in Fig. 8.

In our approach, the particle filter algorithm (Rekleitis, 2004; Fox, 2003; Fox et al., 19999; Olson, 2000) is used to simultaneously estimate the robot position. In this method, the possible locations of the robot are represented as a set of pairs (q, w), where q is a state vector (position and orientation of the robot in the global coordinate system) and $w \in [0, 1]$ is the weighting factor which describes the confidence level that the robot is in q. The algorithm consists of the following steps:

- Initial set Q^i of particles is generated.
- In the next step the new set Q^{i+1} is computed. The particles are iteratively propagated using the control input (motion model). On the basis of the

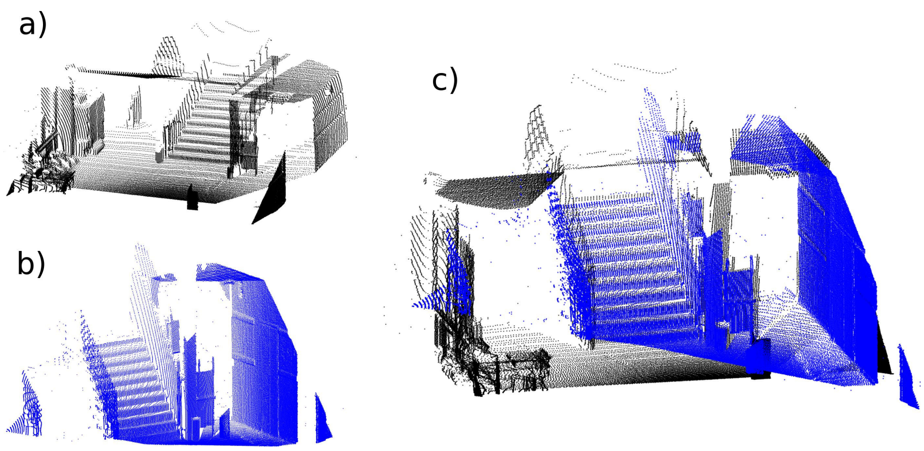


Figure 8. *a)/b)* Point cloud from scan A/B which correspond to scan no. 36 and 37 from Fig. 9. *c)* Joined point cloud. A has been transformed using $(\Delta x\ \Delta y\ \Delta\theta) = (-0.15\text{m}, 1.192\text{m}, 44°)$

measurement model, the weight w_{i+1} is attached to each particle.

- The particles, which have the maximum values of w_{i+1} are multiplied and particles with the value of w_{i+1} below some threshold are reduced.

The main part of the algorithm is to detect and to match characteristic features of the environment. The semantic information is very useful during the localization. In our approach, walls are used as landmarks. This kind of localization is typically used in the structured environment (Gutmann et al., 1998).

The number of particles depends on the uncertainty of odometry. In the case of the mobile robot Elektron 1, the error of determining the orientation of the robot surpasses 30^o so a large number of particles has to be used during the localization process. In order to improve the odometry, information about the main directions is taken into account during the propagation of the particles (Siemiątkowska and Dubrawski, 1999). When the information about main directions of the environment is used, the error in determining the orientation of the vehicle does not surpass 3^o and the number of particles can be reduced.

4.1. A sample global map

Using localization and place identification (based on object identification and Hough transforms) it is possible to aggregate data collected by the robot into a global map. Fig. 9 depicts such a map constructed from 44 3D scans taken at different robot position. Each scan has been classified into the following categories:

- an office room (if an office chair has been detected on the scan),
- a hall (a scan containing recognized stairs),
- a corridor (a specific Hough transform),
- a bathroom (a scan containing a recognized washbasin),
- unclassified.

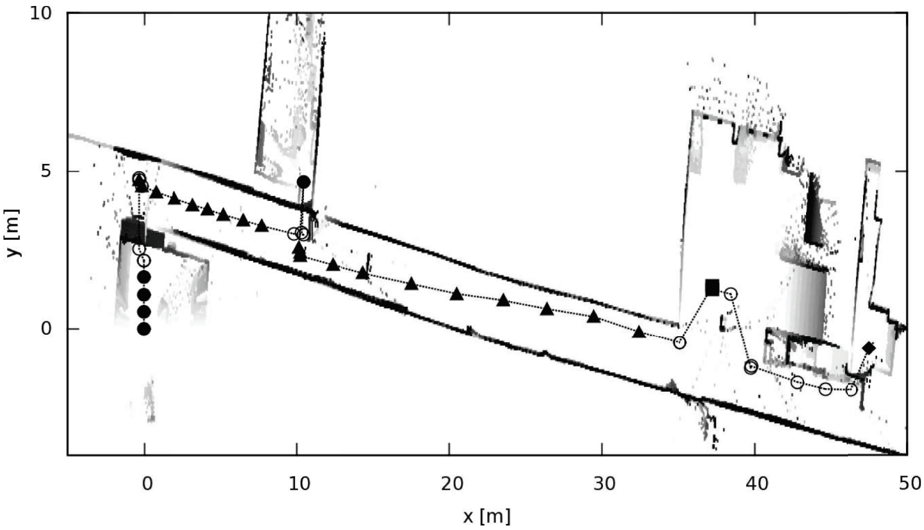


Figure 9. A global map representing a sample indoor environment. Gray-scale contours represent obstacle height (ceiling and floor have been removed). Circles, triangles and squares represent positions at which 3D scan has been taken by the robot and classified, its initial position was $(0, 0)$: ● – an office room, ■ – a hall, ▲ – a corridor, ⧫ – a bathroom, ○ – unclassified

5. Path planning

In our approach the Cellular Neural Network (Chua and Roska, 1993; Chua and Young, 1988) is used for the collision free path planning. The Cellular Neural Network (*CNN*) is a single-layer network defined on regular lattices. The neurons are usually arranged in a rectangular network. It is assumed that *CNN* consists of cells that interact locally. This type of CNN can be viewed as a generalization of cellular automata. The neurons can be modeled as locally connected finite states machines. The state x_{ij} of a cell c_{ij} depends on the states of the neighboring cells, values of input signals u_{ij}, and values of interconnection weights, a_{kl}^{ij} and b_{kl}^{ij}; a_{kl}^{ij} is a weight between cells c_{kl} and c_{ij}, b_{kl}^{ij} is a weight between u_{kl} and cell c_{ij}.

The grid-based representation of the environment is used for collision free path planning. To each cell of a map the list of objects, which are found in the corresponding area of the environment is attached and the cost funcion $u(p_{ij})$ of being in the cell p_{ij} is computed based on values u^o_k, where object k belongs to the cell p_{ij}, $u(p_{ij}) = L$, (where $L \in R$ and it is a very large number) if the cell p_{ij} is occupied by an obstacles, an $u(p_{ij}) \in [0, L)$ in other cases.

The set of cells of the map defines the set of possible positions of the robot and the minimum cost planning problem is defined as follows:
find the sequance of cells $p^0 = \{p^0_0, ..., p^0_N\}$ *that minimizes the travel cost* $K(p)$ *from the starting configuration (initial robot position* p^0_0*) to the destination configuration (goal position* p^0_N*).*

$K(p)$ is described as follows:

$$K(p) = \sum_{i=1}^{N} k(p_i) \tag{3}$$

$$k(p_i) = \mathrm{dist}(p_{i-1}, p_i) + u(p_i) \tag{4}$$

where $dist(p_{i-1}, p_i)$ is a distance between p_{i-1} and p_i and $u(p_i)$ is the cost function for being in the cell p_i. The problem is solved applying the Bellman Meyn (2007); Bellman (1957) approach, implemented using the Cellular Neural Network (CNN) (Chua and Young, 1988; Chua and Roska, 1993).

The CNN, which is used for the path planning, consists of three layers each of them composed of $N \times M$ neurons. Each neuron corresponds to a cell of the grid-based map of the environment.

The first layer is the goal layer, the symbol g_{ij} describes the state of neuron ij in the goal layer, $g_{ij} = L$ if the corresponding area belongs to set of goals Q_{goal} and $g_{ij} = 0$ in other cases.

The second layer is called the traversability layer, the value u_{ij} represents the cost when the robot is placed in the cell ij.

The third layer is called the diffusion layer. Symbol x_{ij} represents the state of the cell ij. The process of the path planning consists of the following steps:

- Initialization
  Weights of connection between corresponding cells are computed using the following formula:

$$a^{kl}_{ij} = dist(p_{ij}, p_{kl}) \tag{5}$$

  where $\mathrm{dist}(p_{ij}, p_{kl})$ is the distance between centers of gravity of areas p_{ij} and p_{kl}, represented by cells ij and kl. Initial values of CNN's cells are:

$$x_{ij}(0) = max(0, g_{ij} - f_{ij}), \tag{6}$$

  $f_{ij} = f(u_{i-r,j-r}, ..., u_{i+r,j+r})$, is a function of u_{kl} values. If r=0 then $f_{ij} = u_{ij}$, when $f_{ij} = max(u_{i-r,j-r}, ..., u_{i+r,j+r})$ then the dimension of the robot can be taken into acount during collision free path planning.

- Diffusion process

$$x_{ij}(t+1) = max(0, g_{ij} - u_{ij}, max_{kl \in N_{ij}}(x_{kl}(t) - a_{ij}^{kl} - f_{ij}(t)) \quad (7)$$

where N_{ij} is the neighborhood of the cell c_{ij}, a_{ij}^{kl} is the weight between c_{ij} and c_{kl}.
The process is continued until:

$$\forall ij \quad x_{ij}(t) = x_{ij}(t+1). \quad (8)$$

The collision-free path is represented as a list of cells. When the cell c_{kl} indicates the current position, the next position is indicated by the cell c_{nm} which fulfills the following requirements:

$$x_{nm} = \max_{c_{ij} \in N^{kl}}\{x_{ij}\} \quad (9)$$

Fig. 10 represents the results of the collision-free path planning method. The robot is asked to go to a chair. For two different robot positions, paths to the nearest chair are generated automatically, without any additional rule-based system.

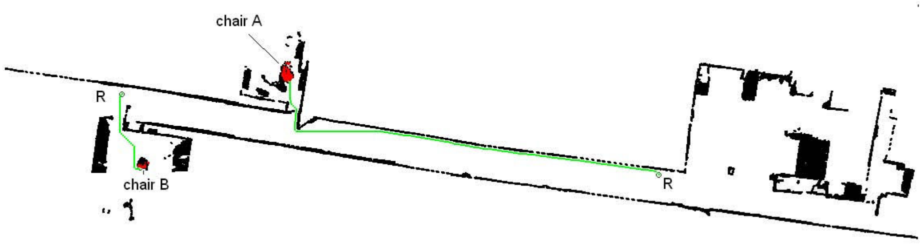


Figure 10. The planned paths for two different position of the robot

6. Conclusions

The main purpose of the work presented in this paper was to build a system for the mobile robot navigation. Experimental results validated the proposed approach and showed the benefits of a dual representation of an environment, as well as CNN for the path planning. The proposed path planning method does not suffer from the local minima problem and a situation when robot or the goal is surrounded by obstacles is easily recognized. The method allows us to describe the goal using the semantic labels and to take into account different criteria during path plannig. It is possible to plan the shortest path, the safest path or force the robot to avoid certain places.

Acknowledgment

This work has been partially supported by the Polish Ministry of Science and Higher Education (grant: 4311/B/T02/2007/33).

References

Azarm, K. and Schmidt, G. (1996) A Decentralized Approach for the Conflict-free Motion of Multiple Mobile Robots. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 1667–1674.

Ballard, D.H. (1981) Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, **2**(13), 111–122.

Barraquand, B., Langois, J.C. and Latombe, J.C. (1992) Numerical Potential Field Techniques for Robot Path Planning, *IEEE Transactions on Robotics and Automation, Man and Cybernetics*, **22**, 224–241.

Bellman, R.E. (1957) *Dynamic Programming.* Princeton University Press.

Bennewitz, M., Burgard, W. and Thrun, S. (2000) Optimizing schedules for prioritized path planning of multi-robot systems. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA).* IEEE, **1**, 271–276.

Besl, P. and McKay, N. (1992) A method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 239–256.

Borkowski, A., Siemiątkowska, B. and Szklarski, J. (2010) Towards Semantic Navigation in Mobile Robotics. In: G. Engels, C. Lewerentz, W. Schäfer, A. Schürr, and B. Westfechtel, eds., *Graph Transformations and Model-Driven Engineering. Essays Dedicated to Manfred Nagl.* **LNCS 5765**, Springer, 730–759.

Buckley, S.J. (1989) Fast Motion Planning for Multiple Moving Robots. In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 1419–1424.

Chen, Y. and Medioni, G. (1991) Object Modeling by Registration of Multiple Range Images. In: *Proc. IEEE Conf. on Robotics and Automation.* IEEE, 616–625.

Chu, H. and Eimaraghy, A. (1992) Real-time Multi-robot Path Planner Based on a Heuristic Approach. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA).* IEEE, **1**, 475–478.

Chua, L. and Roska, T. (1993) The CNN paradigm. *IEEE Transactions on Circuit Systems*, **40**, 147–156.

Chua, L. and Young, L. (1988) Cellular Neural Network. *IEEE Transactions on Circuit Systems*, **35**, 1271–1290.

Duda, R.O. and Hart, P.E. (1972) Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, **15**(1), 11–15.

Elfes, A. (1987) Sonar-based Real-world Mapping and Navigation. *IEEE Journal of Robotics and Automation*, **3**, 249–265.

Fox, D. (2003) Adapting the Sample Size in Particle Filters through KLD-sampling. *International Journal of Robotics Research*, **22**(12), 985–1003.

Fox, D., Burgard, W., Deallert, F. and Thrun, S. (1999) Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI-99).* AAAI, 343–349.

Gnatowski, M., Siemiątkowska, B. and Szklarski, J. (2010) Extraction of semantic information from the 3D laser range finder. In: W. Schiehlen and V. Parenti-Castelli, eds., *ROMANSY 18. Robot Design, Dynamics and Control.* **CISM 524**, Springer, 383–390.

Grewal, M.S. and Andrews, A.P. (2001) *Kalman Filtering: Theory and Practice Using MATLAB.* John Wiley and Sons.

Gutmann, J.S., Burgard, W., Fox, D. and Konolige, K. (1998) Experimental Comparison of Localization Methods. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98).* IEEE.

Kröse, B., Booij, O. and Zivkovic, Z. (2007) A Geometrically Constrained Image Similarity Measure for Visual Mapping, Localization and Navigation. In: W. Burgard and H.M. Gross, eds., *Proc. of the 3rd European Conference on Mobile Robots, EMCR 2007.* http://ecmr07.informatik.uni-freiburg.de/proceedings/ECMR07_0072.pdf.

Latombe, J.C. (1992) *Robot Motion Planning.* Kluwer Academic Publishers, MA Boston.

Meyn, S.P. (2007) *Control Techniques for Complex Networks.* Cambridge University Press.

Moravec, H. and Elfes, A. (1985) High Resolution Maps from Wide Angle Sonar. *Proc. IEEE International Conference on Robotics and Automation.* IEEE, 116–121.

Mozos, O.M., Triebel, R., Jensfelt, P., Rottman, A. and Burgard, W. (2007) Supervised Semantic Labeling of Places Using Information Extracted From Sensor Data. *Robotics and Autonomous Systems*, **5**, 392–402.

Nüchter, A., Lingemann, K., Hertzberg, J. and Surmann, H. (2005) Accurate Object Localization in 3D Laser Range Scans. In: *In Proceedings of the 12th IEEE International Conference on Advanced Robotics (ICAR '05).* IEEE, 665–672.

Nüchter, A., Surmann, H. and Hertzberg, J. (2004) Automatic Classification of Objects in 3D Laser Range Scans. In: *Proc. 8th Conf. on Intelligent Autonomous Systems.* IOS Press, 963–970.

Olson, C. (2000) Probabilistic Selflocalization for Mobile Robots. *IEEE Transaction on Robotics and Automation*, **16**(1), 55–66.

Pfingsthorn, M., Slamet, B. and Visser, A. (2007) A Scalable Hybrid Multi-robot Slam Method for Highly Detailed Maps. In: *Proceedings of the 11th RoboCup International Symposium*. IEEE, 457–464.

Rekleitis, I.M. (2004) A Particle Filter Tutorial for Mobile Robot Localization. Technical report, University of Montreal.

Remolina, E. and Kuipers, B. (2004) Towards a General Theory of Topological Maps. *Artificial Intelligence*, **152**(1), 47 – 104.

Rusu, R.B., Martonand, Z.C., Blodow, N., Dolha, M. and Beetz, M. (2008) Towards 3d Point Cloud Based Object Maps for Household Environment. *Journal of Robotics and Autonomous Systems*, **56**, 927–941.

Siemiątkowska, B. and Dubrawski, A. (1999) Global Map Building and Path Planning for Mobile Robots. **LNCS 1424**, Springer, 68–74.

Siemiątkowska, B., Szklarski, J., Gnatowski, M. and Borkowski, A. (2009) Towards Semantic Navigation System. In: M. Klopotek, A. Przepiorkowski, S. Wierzchon and K. Trojanowski, eds., *Recent Advances in Intelligent Information Systems*. Warsaw, Exit, 711–720.

Thrun, S., Burgard, W. and Fox, D. (2005) *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Weingarten, J. and Siegwart, R. (2005) EKF-based 3D SLAM for Structured Environment Reconstruction. In: *Proc. of IROS 2005*. IEEE, 3834–3839.