Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–14

Mobile Robot Navigation With Use of Semantic Map Constructed From 3D Laser Range Scans

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

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

Abstract

In the article we describe a system which allows a mobile robot equipped with a 3D laser range finder to navigate in indoor and outdoor environment. A global map of the environment is constructed, the particle filter algorithm is used in order to accurately determine the position of the robot. Based only on data from the laser, the robot is able to recognize certain classes of objects like a floor, a door, a washbasin, or a wastebasket. For complex objects, the recognition process is based on Haar features 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 use of semantic labels, e.g. "go to the nearest *wastebasket*". The obstacle-free path is generated using a Cellular Neural Network which takes into account travel costs like distance or the ground quality. This path planning method is fast and in comparison to potential field method, it does not suffer from local minima problem. We present some results of experiments performed in a real indoor environment.

1 Introduction

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

Knowledge about robot environment is usually encoded in a form of a map. The mapping problem is one of the most active areas in robotics. Most of methods focus on 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 most popular geometric map representation is the occupancy grid. This kind of representation allows fast generation of a collision-free path, however, if very precise map of environment is required a huge amount of memory is necessary. Feature-based representations are attractive because of their compactness and are very useful during process of localization but path-planning based on this kind of map is time consuming.

- topological maps (Latombe, 1992; Remolina and Kuipers, 2004) represent relations between distinctive parts in the environment, they can be used to solve abstract tasks.

This two representations can be combined into a hybrid map (Pfingsthorn *et al.*, 2007; Thrun *et al.*, 2005) which contains both metric and topological information. Recently researchers have been focused on semantic maps that contain data about meaning of the detected objects, functionalities or even events (Rusu *et al.*, 2008; Mozos *et al.*, 2007; Siemiatkowska *et al.*, 2009).

The robot needs to know its position in the environment. The most widely used is 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 travelled 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 simultaneously estimate the robot position (Olson, 2000; Grewal and Andrews, 2001; Weingarten and Siegwart, 2005). In this method, the encoder readings are used as an input 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 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, $w \in [0, 1]$. This kind of localization is used in our approach.

The aim of path planning is to find 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 a global or local. Global methods (Latombe, 1992; Bennewitz *et al.*, 2000) require 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 information about nearby obstacles is taken into account. Although the method is fast, it can be trapped in local minima.

The problem how to find optimal collision-free path is strictly connected with 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 travelled by the robot is less important than in the case of outdoor navigation. When the robot moves outside 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 along a road than along a grass.

In this article the system which allows the robot to navigate in outdoor and indoor environment is presented. Data obtained from a 3D laser range scanner is analyzed and semantic labels are attached to the detected objects. The map is represented as a grid of cells and a list of labels which are 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.

2 Map building

Mapping process is very important element of the navigation system. A map can represent geometric or semantic features of the environment.

When geometric representation is used it is assumed that the robot operates in a 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.

If the robot is assumed to cooperate with people, the workspace W has to be transformed into information space I. In our method each object is described by the following parameters: (s, u_s, r_s, f_s), where s - represents the semantic label, $u : S \to R^+$, u represents traversability level of the object, $r \in R$ is the radius of the influence of the object, $f_s : R \to R$ is the function which represents the influence of the object. Parameters u, r, f_s are used during path planning and are described in the next section of the article. Process of map building consists of the following parts: perceptions, data analysis, localization and data aggregation.

2.1 Perception

Perception of the environment (Siemiatkowska *et al.*, 2009) is one of the crucial problems in mobile robotics. During path-planning process a mobile robot must be able to detect certain classes of objects and landmarks. This makes possible to estimate 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-based cameras. Images from such cameras can be used do detect objects based, e.g., on appearance (Krose *et al.*, 2007) as well as to estimate robot position and object size when using stereoscopic vision. There are, however, some limits connected with this approach. For example, lightening conditions vary significantly which makes the detection process difficult and often unreliable Geometric information calculated from stereo-vision is limited due to large errors.

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

The experiments described here have been performed on a mobile robot "Elektron" which has been built at 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 which enables to make 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°. The scanning laser enables to measure the distance to obstacles within $-90° \leq \phi \leq 90°$ with resolution of 0.5°, see Fig. 1. The data is subsequently transmitted to the control unit by means of an RS 422 bus.

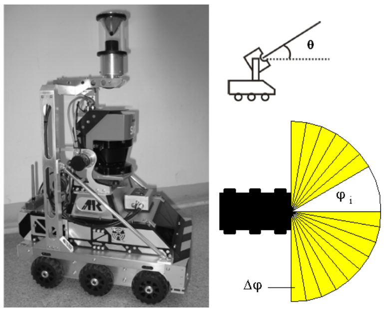


FIGURE 1: The „Electron" robot with explanation of the ϕ, θ angles.

The laser scanner 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, r_i is the measured distance. It is common that the 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 in 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.

The most straightforward way to transform data from the laser scanner is to use (ϕ, θ) as pixel coordinates and assign the pixel color according to the measured distance. However, since this approach does not lead to a satisfactory method of representing 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. For each pixel (i, j) using simple trigonometry we obtain its position $\mathbf{p}$ in 3D Cartesian coordinates with the robot in its center (a standard procedure for point cloud methods). Then four neighbouring points $\mathbf{p}_{1\ldots4}$ with $(i \pm 1, j-1), (i \pm 1, j+1)$ are considered (alternatively we take into account more points, however it does not give any improvement to the overall procedure). If a point $\mathbf{p}_n$ is too far or to close to $\mathbf{p}$, including it in calculation might lead to spurious errors. Therefore all $\mathbf{p}_n$ not fulfilling the inequality

$$\epsilon_0 \leq \|\mathbf{p} - \mathbf{p_n}\| \leq \epsilon_1$$

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

$$\mathbf{n} = \sum_{1 \leq i \leq 3} \sum_{i < j \leq 4} \mathbf{p_i} \times \mathbf{p_j}$$

where $\times$ is the cross product, $\mathbf{n}$ is normalized afterwards. 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. Note that for example ceiling or floor will have red and green component equal to 128, while blue will be larger than 128 for floor (making it blueish) and smaller than 128 for ceiling (making it look more yellowish). On the other hand all of the planes perpendicular to the laser scanner will have the blue component equal to 128. Moreover, walls which are placed along the line of sight of the robot will be pink in its left-hand side and cyan on the other side. A sample image for an outdoor scene is depicted in Fig. 2.

2.2 Segmentation

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

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

2.2.1 Rule-based identification of uniform areas

The first step is to perform a fast segmentation of the gathered data into areas, each one representing a uniform polygon in the real scene. Along the most important areas are, of course, ceiling, floor, walls, doors, etc. Besides the list of polygons, some numbers characterizing physical properties of a polygon can be extracted as well. These can be later used for better object classification.

In the first step a flood fill algorithm (we use the OpenCV library), is run on an image representing distance with pixel (0,0) as the seed point. The threshold for the algorithm is constant and it corresponds to about 5 cm (difference between neighbor pixels is considered when flooding). If the resulting area is large enough, i.e., has total size greater than 30 pixels is marked with number 1 and then the same procedure is applied for a next pixel which has not been assigned to the area. In this way we obtain area number 2, and the procedure is repeated until all the pixels are assigned to one of the n areas. All areas which are too small to be classified are marked with number 0 and are not considered in any later stage of the process.

After the first step is finished, one has list of areas which represent „continuous structures" of the environment. For example, a chair standing in front of a wall will be assigned to different area than wall since there is a large change of distance between the chairs edge and the wall. On the other hand, walls, ceiling, and floor will be classified as a single area since change of the distance in all the corners is assumed to be small.

Having the set of areas together with information about x, y, z coordinates in the real environment for each pixel, one can define simple rules for identifying objects of certain class. Then the following characteristics are used to identify certain objects (see also, e.g., Vosselman *et al.* (2004)):

- size – usually an object which is supposed to be detected is characterized by some reasonable geometrical size
- orientation – for example: walls or doors are always vertical, ceiling, floor are always horizontal, floor has z component equal to the ground level
- topology – relations between surfaces is important
- measure of „roughness” – value(s) used to describe to what extend an area is flat

Based on the above characteristics a simple rule-based classifier is applied. For example, in such system an object is labeled as *door* if it is a single, vertical surface with width in range 1-2m, height 1.5-2.5m, it connects directly to a *floor* at its bottom and to a *wall* from all other sides.

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

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

where w is width of window for which the averaged dot product of 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 with width w'. For each A' its averaged „roughness”, $\bar{R}$ is calculated. For example, for outdoor environment, $w = 2, w' = 5$, an area for sidewalk has $\bar{R} \approx 1$, whereas region covered by grass has $\bar{R} \approx 0.95$. The value $\bar{R}$ gives information about traveling cost through A' for the mobile robot. This cost is later used in the path planning algorithm. Results of classification for a sample outdoor scene is shown in Fig. 2.

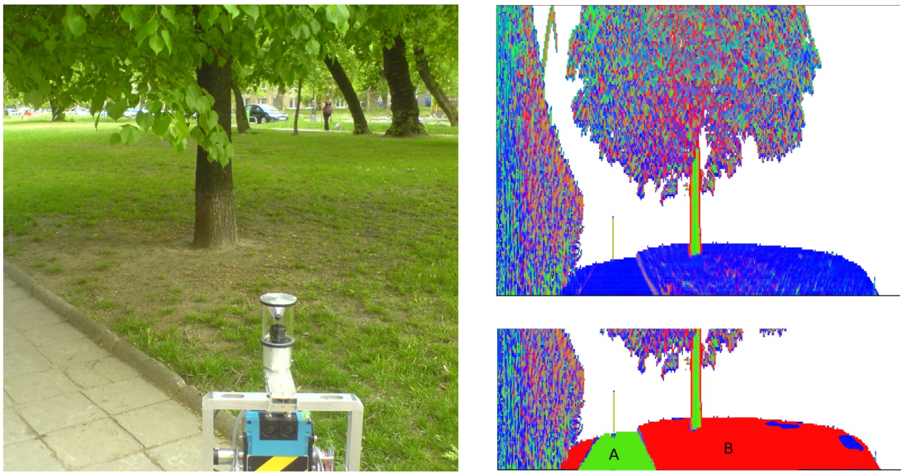


Figure 2: *Left:* Photo of an outdoor scene. *Right:* top: an RGB image representing normal vectors constructed from data from laser range finder, bottom: result of a rule-based classification, A-sidewalk (a flat terrain with low travel cost), B-grass (a terrain with average „roughness” coeeficient $\bar{R} \approx 0.95$; travel cost is higher than for the sidewalk).

2.2.2 Object identification with Haar features

Treating laser scanner data as an image makes possible to directly apply well known methods for object detection and pattern recognition from image analysis field. Here we show how to enrich our classification system by using scheme based on a boosted cascade of simple features to detect objects. Algorithms which are applied here are available in OpenCV library and they implement methods proposed by Viola and Jones (2001, 2004) for basic set of Haar-like features and by Lienhart and Maydt (2002) for rotated features which enhance the basic set. After the system is trained for recognition of certain objects, new images can be analysed very fast while maintaining good hit rate and reasonably low false alarm rate. This makes the method interesting and practical for our purposes. Moreover, having direct geometrical information about a detected object we can often reject false classifications just by analysing its real size.

Images generated from laser data in the way proposed above have, of course, different properties than usual visual images gathered by cameras. For example, illumination and any lighting conditions are not of our interest here. Either in bright light with many shadows or in completely dark room one gets the same image. On the other hand, when the robot equipped with the scanner moves on a floor, red and green components of the image change so there is some dependency upon its position. This, however, does not have to lead to problems with object detection by image analysis, since many methods used for that purpose operate on gray-scale images.

In the first stage of our classifier it is necessary to train it for objects of interest, like stairs or office equipment. Here we show an example of detection of a washbasin visible from perspective of the robot. In order to perform training a Haar-like classifier needs a large set of positive and negative example images. All of the images should be scaled to the same size, we use 20x20. As the set of negative examples we use large number of arbitrary images representing a scene without any object of our interest, i.e., without any image of a washbasin in this case.

In order to get set of positive examples we take several snapshots with the laser scanner of a scene with the object of interest. Then, after constructing images representing the scene, we crop a sub-image with the object only, making background translucent. In the next step all the sub-images are rotated about x, y, z axis by random angles, intensity is randomly modified and washbasin are placed onto a random background image. Finally we have 1000 different images of a sink with random transformations placed onto different backgrounds. Large images containing the object serve as testing images after training.

After the training process is completed, the classifier can be applied to any region of an image giving *true* if the region is likely to contain pattern similar to one of those from the positive samples set, *false* otherwise. Analysis is very fast so one can try many different regions with varied sizes from all parts of the image. By doing this in a loop one can search entire image for object of interest. Figure fig:haar shows a result of such analysis when searching with classifier trained for detection of objects similar to a sink and stairs respectively.

Areas corresponding to objects detected with the discussed method can be later processed in our classification scheme when building semantic map. Naturally, each object which is going to be recognized has to have its own specifically trained Haar-features classifier. In more sophisticated approach it is possible to combine object recognition from both: visual images and images constructed from laser scans.

FIGURE 3: Sample scenes with result of the detection (black rectangles) of a washbasin-like objects. On the left side a small subset of the training set is presented.

2.3 Localization

The global map of the environment is represented as a grid of cells. A global coordinate system is introduced in W. Each cell corresponds to the square of size $5cm \times 5cm \in W$, and a list of semantic labels s is attached to each cell of the grid. The method of transformation from W to I is described in the previous section.

The position of the robot q is described in configuration space C. 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 in form of the map. When the robot moves to the next position q_{i+1}, it gathers data from the lase scanner, which it transformed into another point cloud which has to be analyzed and transformed from local into global coordinate system. To performe this task the robot has to know its position q_i, so it has to compute following values: (Δx,Δy,$\Delta \theta$), where:

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

The most widely used method of localization is odometry, but when this method is applied the errors of determining position of the robot increase if the vehicle travels for a long time. In our approach the particle filter algorithm (Rekleitis, 2004; Fox, 2003; Fox *et al.*, 1999; Olson, 2000) is used to simultaneously estimate the robot position. In this method possible locations of the robot are represented as a set of pairs $(\mathbf{q}, \mathbf{w})$, where $\mathbf{q}$ 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 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 localization. In our approach walls are used as landmarks. This kind of localization is typically used in structured environment (Gutmann *et al.*, 1998).

The number of particles depends on the uncertainty of odometry. In the case of the mobile robot Elektron1, 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, the information about the main directions is taken into account during the propagation of the particles (Siemiątkowska and Dubrawski, 1999). When information about main directions of the environment is used the error in determining the orientation of the vehicle does not surpass 3^o. This approach allows us a reduction in the number of the particles.

Fig. 4 presents the map of the environment which was built based on laser range indications. Each pixel corresponds to $5cm \times 5cm$ square area. Cells representing objects labelled as *stairs*, *wastebasket*, *washbasin*, *door* are distinguished. White pixels represent the floor area and unknown area. Black pixels represent walls and other unclassified obstacles. The ceiling area is recognized by our algorithm but it is not presented in the picture.

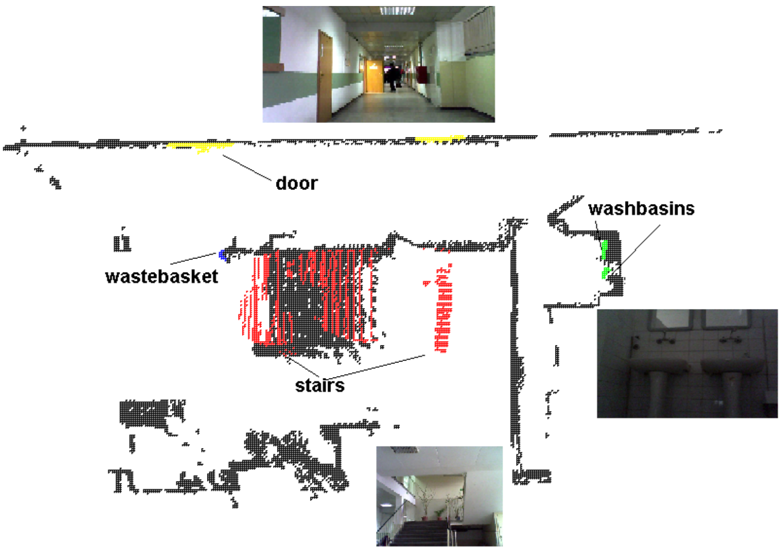


FIGURE 4: The map of the environment

3 Path planning

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

Before the process of path planning starts the traversability level (cost function) $u \in [0, L]$ is attached to each object. This value depends on the object and on the type of the robot. For example stairs is traversable for the walking robot but is not traversable for wheel vehicles, if s represents the floor then corresponding value $u = 0$, when the area is not traversable, for instance s presents a wall then $u = L$.

We assumed that the goal is given using semantic labels and motion planning occurs in the configuration space C which is the set of all possible configuration of the robot. The symbol R(q) is a set of points occupied by the robot at configuration q.

In the set of objects O the set of obstacles $O_{cc} \in O$ is distinguished. Free space $W_{\text{free}} = W - O_{cc}$. An obstacle C_{occ} in the configuration space corresponds to the set of configuration that intersects the obstacles in W.

$$C_{cc} = \{q | R(q) \cap O_{cc} \neq \emptyset\} \tag{2}$$

$Q_{\text{free}} = Q - O_{cc}$

The goal configuration Q_{goal} is defined a set of points which intersects the object O_{goal}.

A path is a continuous curve in the configuration space which is represented by a function:

$$p : [0, T] \to Q, \tag{3}$$

where $p(0) = q_R$, $p(T) = q_{\text{goal}}$ and $q_{\text{goal}} \in Q_{\text{goal}}$

It is assumed that:

$$\forall t \in [0, T] \quad p(t) \in Q_{\text{free}} \tag{4}$$

The minimum cost planning problem is defined as follows:
find the path p(t) that minimizes the travel cost from starting configuration to the destination configuration.

We are looking for a path $p_o(t)$ that minimizes the following function:

$$K(p_0) = \min_p \int_{t=0}^{T} k(p(t)) \tag{5}$$

where $k : Q \to R^+$ - is the cost function.

In the discrete case $K(p)$ is described as follows:

$$K(p_0) = \min_p \sum_{t_i=0}^{T} k(p(t_i)) \tag{6}$$

$$k(p(t_i)) = \mathrm{dist}(p(t_i), p(t_{i+1})) + u(p(t_i)) \tag{7}$$

where $\mathrm{dist}(p(t_i), p(t_{i+1}))$ is a distance between $p(t_i)$ and $p(t_{i+1})$ and $u(p(t_i))$ is the cost function being in the state $p(t_i)$. If $p(t_i) \in C_{occ}$ then $u(p(t_i)) = \infty$.

The problem is solved applying Bellman(Meyn, 2007; Bellman, 1957) approach implemented using Cellular Neural Network (CNN) (Chua and Young, 1988; Chua and Roska, 1993).

The CNN which is used for path planning consists of three layers each of them consists of $N \times M$ neurons. Each neuron corresponds to some cell of 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 of C-space belongs to Q_{goal} and $g_{ij} = 0$ in other cases.

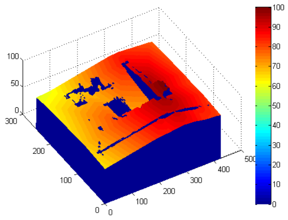
The second layer is called traversability layer, the value u_{ij} represents the cost when the robot is placed in the cell ij.

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

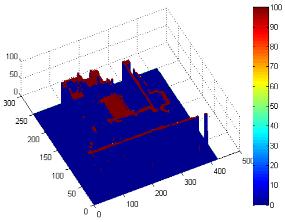
- Initialization
  Weights of connection between corresponding cells are computed using the following formulae:
  $$a_{ij}^{kl} = \mathrm{dist}(p_{ij}, p_{kl}) \tag{8}$$
  where $\mathrm{dist}(p_{ij}, p_{kl})$ is the distance between centres of gravity of areas p_{ij} and p_{kl} which are represented by cells ij and kl. Initial values of CNN's cells equal:
  $$x_{ij}(0) = g_{ij} - u_{ij} \tag{9}$$
- Diffusion process
  $$x_{ij}(t+1) = \max(g_{ij} - u_{ij}, \max_{kl \in N_{ij}}(x_{kl}(t) - a_{ij}^{kl} - u_{kl}(t)) \tag{10}$$
  where N_{ij} is the neighbourhood of the cell c_{ij}, a_{ij}^{kl} is the weight between c_{ij} and c_{kl}. The values of input signals u_{ij} represent the influence of the obstacles, $u_{ij} = L$, where L is a very large value if the area which corresponds to the cell c_{ij} is occupied by an obstacle and $u_{ij} \in [0, L]$ in other cases. The process is continued until stability:
  $$\forall ij \quad x_{ij}(t) = x_{ij}(t+1) \tag{11}$$

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 fulfilled the following requirements:

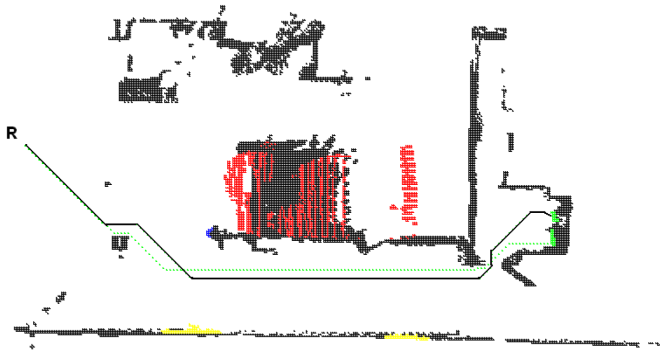
$$x_{nm} = \max_{c_{ij} \in N^{kl}} \{x_{ij}\} \tag{12}$$



a)



b)



c)

FIGURE 5: The collision free paths: a) diffusion map (the shortest path), b) values of u (the shortest path), c) planned paths: dotted line - the short path, solid - line - the safe path

Fig. 5c presents the result of path planning to a washbasin. If is assumed that the path has to be a shortest one then $r = 5$cm, $f_s(r_1) = L$ if $r_1 < r$ and $f_s(r_1) = 0$ if $r_1 \geq r$. It is assumed that s is the label attached to an obstacle. When the path has to go far from the obstacles then $r = 30$cm and f_s is defined as follows:

$$f_s(r_1) = \begin{cases} 0 \text{ if } r_1 \leq r \\ 1 \text{ if } r_1 \in [\frac{r}{2}, r] \\ L \text{ if } r_1 \leq \frac{r}{2} \end{cases} \tag{13}$$

R represents the robot position, Fig. 5a presents the diffusion map, fig. 5b presents values of u_{ij} in the case when the shortest path is planned. When the shortest path is planned the robot goes to the washbasin which is next to the door. The safe path ends up near the second washbasin.

4 Conclusions

The main purpose of the work presented in this article was to build a system for mobile robot navigation. It is assumed that the goal is given using semantic labels. Experimental results validated the proposed approach and showed the benefits of a dual representation of an environment, as well as CNN for path planning. The proposed path planning method does not suffer from local minima problem.

5 Acknowledgment

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

References

K. Azarm and G. Schmidt (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)*, pp. 1667–1674.

B. Barraquand, J.C. Langois, and J. C. Latombe (1992), Numerical potential field techniques for robot path planning, *IEEE Transactions on Robotics and Automation, Man and Cybernetics*, 22:224–241.

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

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

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

S.J. Buckley (1989), Fast motion planning for multiple moving robots, *In Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1419–1424.

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

H. Chu and H. A. Eimaraghy (1992), Real-time multi-robot path planner based on a heuristic approach, *In Proc. of the IEEE International Conference on Robotics & Automation (ICRA).*

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

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

A. Elfes (1987), Sonar-based real-world mapping and navigation, *IEEE Journal of Robotics and Automation*, 3:249–265.

D. Fox (2003), Adapting the sample size in particle filters through KLD-sampling, *International Journal of Robotics Research*, 22(12):985–1003.

D. Fox, W. Burgard, F. Dealert, and S. Thrun (1999), Monte Carlo Localization efficient position estimation for mobile robots, *AAAI/IAAI*, pp. 343–349.

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

J. S. Gutmann, W. Burgard, D. Fox, and K. Konolige (1998), Experimental comparison of localization methods, in *Proceedings of IROS.*

B. Krose, O. Booij, and Z. Zivkovic (2007), A geometrically constrained images similarity measure for visual mapping, localization and navigation, in W. Burgard and H. M. Gross, editors, *Proceedings of the 34d ECMR*, pp. 168–173.

J. C. Latombe (1992), *Robot Motion Planning. Kluwer Academic Publishers*, MA Boston, ISBN 0-7923-9206-X.

Rainer Lienhart and Jochen Maydt (2002), An Extended Set of Haar-like Features for Rapid Object Detection, *IEEE ICIP 2002*, 1:900–903.

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

H. MORAVEC and A. ELFES (1985), High resolution maps from wide angle sonar, *Proc. IEEE International Conference on Robotics and Automation*, pp. 116–121.

O. M. MOZOS, R. TRIEBEL, P. JENSFELT, A. ROTTMAN, and W. BURGARD (2007), Supervised semantic labeling of places using information extracted from sensor data, *Robotics and Autonomous Systems*, 5:392–402.

A. NÜCHTER, K. LINGEMANN, J. HERTZBERG, and H. SURMANN (2005), Accurate Object Localization in 3D Laser Range Scans, in *In Proceedings of the 12th IEEE International Conference on Advanced Robotics (ICAR '05), pages 665 – 672.*

A. NÜCHTER, H. SURMANN, and J. HERTZBERG (2004), Automatic Classification of Objects in 3D Laser Range Scans, in *In Proc. 8th Conf. on Intelligent Autonomous Systems, pages 963 – 970*, pp. 963–970, IOS Press.

C. OLSON (2000), Probabilistic selflocalization for mobile robots, *IEEE Transaction on Robotics and Automation*, 16(1):55–66.

M. PFINGSTHORN, B. SLAMET, and A. VISSER (2007), A scalable hybrid multi-robot slam method for highly detailed maps, in *Proceedings of the 11th RoboCup International Symposium.*

I.M. REKLEITIS (2004), A particle filter tutorial for mobile robot localization, Technical report, Raport, Univerity Montreal.

E. REMOLINA and B. KUIPERS (2004), Towards a general theory of topological maps, *Artificial Intelligence*, 152(1):47 – 104.

R. B. RUSU, Z. C. MARTONAND, N.BLODOW, M. DOLHA, and M. BEETZ (2008), Towords 3d point cloud based object maps for household environment, *Journal of Robotics and Autonomous Systems*, 56:927–941.

B. SIEMIĄTKOWSKA and A. DUBRAWSKI (1999), Global Map Building and Path Planning for Mobile Robots, *Springer LNCS*, pp. 68–74.

B. SIEMIATKOWSKA, J. SZKLARSKI, M. GNATOWSKI, and A. BORKOWSKI (2009), Towards Semantic Navigation System, in M. KLOPOTEK, A. PRZEPIORKOWSKI, S. WIERZCHON, and K. TROJANOWSKI, editors, *Recent Advances in Intelligent Information Systems*, pp. 711–720, Exit.

S. THRUN, W. BURGARD, and D. FOX (2005), *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press.

P. VIOLA and M. J. JONES (2001), Rapid Object Detection using a Boosted Cascade of Simple Features, *IEEE CVPR.*

P. VIOLA and M. J. JONES (2004), Robust real-time face detection, *International Journal of Computer Vision*, 57:137–154.

G. VOSSELMAN, B. GORTE, G. SITHOLE, and T. RABBANI (2004), Recognizing structure in laser scanner point clouds, in *Proceedings of Conference on Laser scanners for Forest and Landscape assessment and instruments.*

J. WEINGARTEN and R. SIEGWART (2005), EKF-based 3D SLAM for structured environment reconstruction, in *Proceedings of IROS.*