Building Three Dimensional Maps based on Cellular Neural Networks

MICHAL GNATOWSKI
BARBARA SIEMIATKOWSKA
RAFAŁ CHOJECKI
Institute of Fundamental Technological Research
Polish Academy of Sciences
Swietokrzyska 21, 00-049 Warsaw
Institute of Automatic Control and Robotics
Warsaw University of Technology
Boboli 8, 02-525 Warsaw
POLAND
mignat@ippt.gov.pl http://www.ippt.gov.pl/~mignat/
bsiem@ippt.gov.pl r.chojecki@mchtr.pw.edu.pl

Abstract: - One of the main problem in robotics is map building. In the article 3D map building based on 3D laser data is presented. The data is obtained from a SICK laser mounted on a rotating support what gives a 3D representation of the scene. The map is divided into cells and each cell represents a certain area of the scene and keeps a list of objects. This is a real-time system, which consumes little computer memory and works properly in indoor environment.

Key-Words: - Robotics, mapping, navigation, cellular neural networks, pattern recognition, data analysis.

1 Introduction

Over the last years robotics applications in the industry as well as in everyday life have increased rapidly. A robot is to perform a major task assigned by a human, like a transportation task, cleaning or painting a given area, etc., but in order to complete it, in the background, the robot must perform a number of other additional tasks. The ability to navigate in an unknown environment is the most important competence for a mobile robot. The navigation system usually consists of the following modules[16,17]:

- map building,
- localization,
- path planning,
- determination of the optimal controls (linear and angular velocities).

Mapping techniques for mobile robots can be classified according to different map representations. One of the more popular map representations is the occupancy grid [20]. In this map the environment is described as a set of cells, where each cell corresponds to some rectangular area of the environment. A value is attached to each cell. The value equals 1 if the cell corresponds to the area occupied by the obstacles or equals 0 if the area is free. In some systems the values show a possibility that the area is occupied or not. This kind of map allows fast generation of a collision-free path for a mobile robot. However, the accuracy of the map depends on the grid's resolution. If a very precise map of the environment is required, then the method is computationally expensive and a huge amount of memory is necessary.

In many navigation systems a geometrical map (feature-based map) is used. In this kind of mapping techniques all obstacles are described by mathematical formulae.

Geometrical representations[17] are attractive because of their compactness and they are also very useful during the process of localization, but path-planning based on this kind of map is time consuming. Mobile robots work in a three dimensional real world, but usually they use two dimensional maps which are error-prone and have several limitations. 3D maps enable to move on a hilly surface and to recognize obstacles which are on different heights. Additionally, it is impossible to search for certain objects in 2D maps. In robotics stereovision systems are used [1] but they consume a lot of computer resources and need stable light conditions.

In our algorithm we propose to build a 3D map based on 3D laser data. On account of large numbers of data the algorithm must be effective. Aggregation and filtration time must be short. Scene representation must be so accurate that a robot must be able to move safely and to localize itself. In the literature, there are many ways of scene representation [2, 5, 4, 3,14], for example, creating 3D grid-based maps [6]. In each cell a value from 0 to 1 is kept and the value represents a probability that there is an obstacle in the cell. Another way of scene representation is to cover obstacles with a polygon network. Usually triangles are used because all vertexes lie on a plane. Sometimes other polygons are used, but first one must check if all vertexes lie on a plane. A scene can also be represented by calculating mathematical equations of obstacles.

All methods shown above consume a lot of computer memory and take a lot of time. Their main disadvantage is that they cannot be used directly in path planning.

Sometimes 2.5D representation is used [7,8], but it is not sufficient in localization process[9].

In the method presented in the article a dual raster-object representation is used. A map is represented as a two dimensional cells table but it differs from classical methods in the way that it keeps references to objects in each cell. The main idea of the system is using cellular neural networks which allow to do calculations parallelly. The idea of the method is shown in figure 1.

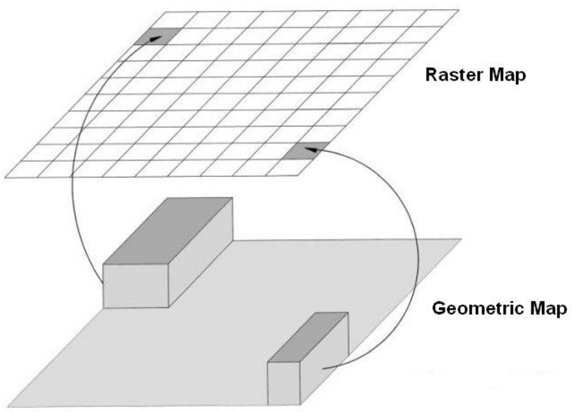


Figure 1. Model of the environment

2 Cellular Neural Networks

The idea of Cellular Neural Networks (CNN) was introduced by Leon O. Chua and L. Young [10] in 1988. CNN is a single-layer network defined on regular lattices.

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{k,l\in N_{ij}} a_{kl}^{ij} f(x_{kl}(t)) + \sum_{k,l\in N_{ij}} b_{kl}^{ij} f(u_{kl}(t)) + I$$

where x_{ij} denotes the state of a cell c_{ij} and $ij \in NxM$, where:

N_{ij} denotes the neighbourhood of a cell c_{ij} , a_{kl}^{ij} is an interconnection weight between cells a_{kl} and a_{ij} , b_{kl}^{ij} is the feedforward template parameter, u_{ij} is an input signal, and I is bias term, f is a linear saturation function. A cell c_{kl} belongs to the neighbourhood of the cell c_{ij} if the condition if fulfilled:

$$\max(|i-k|,|j-l|) \le r, \qquad (1)$$

Where r is a neighbourhood parameter.

Figure 2 presents a scheme of a single neuron in CNN, where x_{ij} denotes the state of a cell c_{ij} and $ij \in NxM$ and N_{ij} is the neighbourhood of a cell c_{ij}. a_{kl}^{ij} is the interconnection weight between the cells a_{kl} and a_{ij} and b_{ij} is the feedforeward template parameter, u_{ij} is an input signal, I is a bias term, f_1 is a linear function.

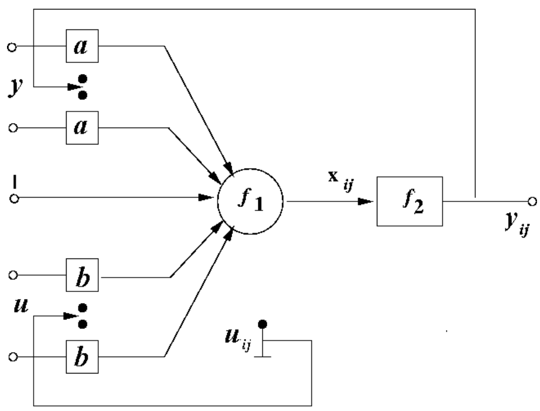
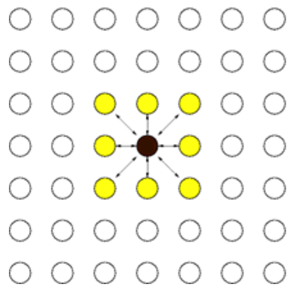
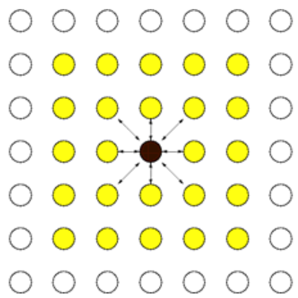


Figure 2 A single neuron scheme

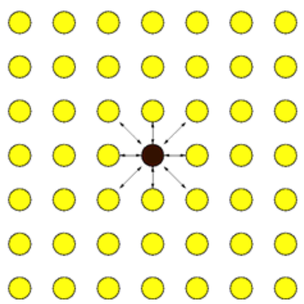
The neighbourhood of each cell can be defined dependently on the distance between the cells. Yellow color in figure 3 presents three different kinds of regular neighbourhood.



(a) r = 1,



(b) r = 2,



(c) r = 3,

Figure 3 Cell regular neighborhood

Cell neighbourhood may be defined irregularly. An example of such irregularity is presented in figure 4.

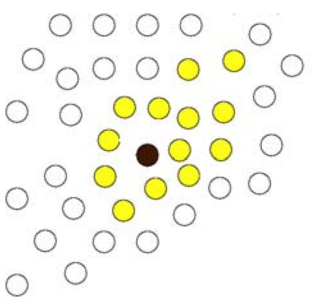


Figure 4 Irregular neighbourhood

The dynamics of discrete-time CNN is described by equation:

$$x_{ij}(t+1) = \sum_{c_{kl} \in N_r^{ij}} a_{ij}^{kl} y_{kl}(t) + \sum_{c_{kl} \in N_r^{ij}} b_{ij}^{kl} u_{kl}(t) + I \qquad (2)$$
$$y_{ij}(t+1) = f_N x_{ij}(t+1)$$

Where f_N is output activation function.

Chua extended the definition of CNN in 1997 [11]. It is assumed that it consists of cells that interact locally. This type of CNN can be viewed as a generalization of cellular automata. The neurons can be modelled as locally connected finite states machines. This type of CNN can be viewed as a generalization of cellular automata[15]. CNNs are widely used for image processing and patterns recognition[21] but it can be also used for path planning[18,19].

2 Hardware

The experiments have been done on a mobile robot “Elektron” which has been designed and built at Institute of Automatic Control and Robotics of Warsaw University of Technology.

The basic sensor is a head module comprising a 3-dimentional scanning laser rangefinder used for navigation and creating 3-dimensional representations of the environment. The module consists of a SICK LMS 200 scanning laser rangefinder installed on a rotating head. The head can rotate the scanner around the horizontal axis within the angular range from -15° to +90°.

The scanning laser enables to measure the distance from to the obstacle within 180° with resolution of 0.5°. The data is subsequently transmitted to the control unit by means of an RS 422 bus.

The module is powered by a DC planetary gear motor. The power is then transferred by means of a toothed belt transmission. Two rotational encoders measure the scanning velocity and angle. The first encoder installed on the motor shaft is used for regulating the position whereas the other is responsible for measuring the rotation angle directly on the rotation axis of the scanner. The two measuring systems allow precise steering and positioning of the sensor. The unit controlling the head enables both continuous as well as step-by-step modes of the head. PID control algorithms were used for positioning and controlling the drive unit. Communication with the main control unit is achieved by means of an RS 422 bus.

The robot is presented in figure 5.

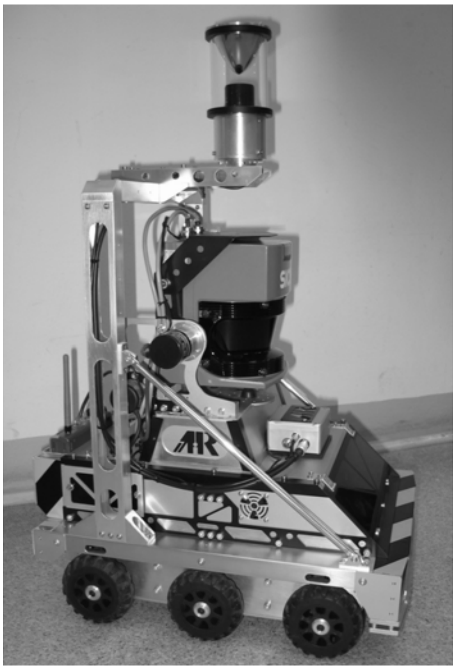
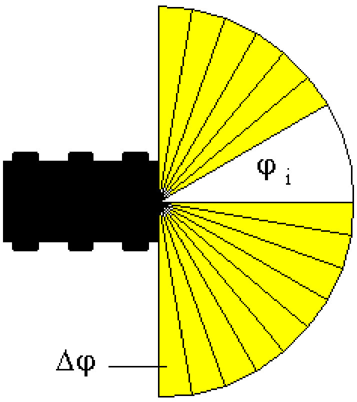


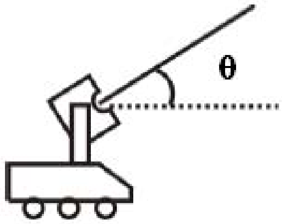
Figure 5 Robot "Elektron" with a 3D laser on a rotating support.

The laser scans a scene and gives 2D data. The rotating support rotates the laser vertically, which allows to make 3D scans. A laser generates data in polar coordinate system (r_i, ϕ_i, θ_i), where θ_i is vertical angle of rotating support, ϕ_i is a horizontal scan angle and r_i is the distance to a given obstacle. If we assume a robot moves on a flat surface and its position is in the beginning of a Cartesian coordinate system along the OX axis, the position of an obstacle is as follows:

$$\begin{aligned} x_i &= r_i \sin\theta_i \sin\varphi_i \\ y_i &= r_i \sin\varphi_i \\ z_i &= r_i \cos\theta_i \sin\varphi_i \end{aligned} \qquad (3)$$



a)

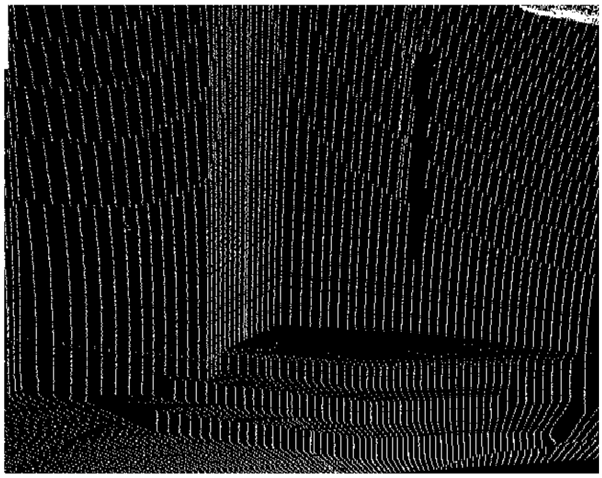


b)

Figure 6 The values of parameters: φ and Θ

Figure 7 shows a photo of the environment and the data given from 3D laser.

(a)



(b)

Figure 7 A place where the data was taken (a) and the data from the 3D laser (b).

3 2.5D map

A typical two dimensional map of an environment is a grid-based map. In each cell a value 0 or 1 is kept which means there is no or there is an obstacle in the cell respectively. More complex kind of a map is a 2.5D map. The 2.5D map is an example of a simplified three dimensional map. The idea behind the method rests on keeping in each cell the value of the maximum obstacle height.

The algorithm of building 2.5D map of the environment consists of the following modules:

- *Initialisation*
  A value *L* is attached to each cell of the map. The value L is smaller then the minimum value of the height of the obstacles. In an office environment *L=0*.
- *Computing the coordinate of the obstacles*
  For the given values of parameters: (r_i, ϕ_i, θ_i), The coordinate (x_i,y_i,z_i) are computed using formulae 3.
- *Computing the coordinate of the cell*
  The coordinate of the cell *kl* of the grid-based map are computed as follows:

$$k = \frac{x_i}{dx} \qquad l = \frac{y_i}{dy} \tag{4}$$

  where *dx* X *dy* is the grid resolution.
- *Uppdating*
  Value z_i is the input value to the cell *kl.*
  The new state of the cell *kl* is computed as follows:

$$x_{kl}(t+1) = \begin{cases} x_{kl}(t) & if\ x_{kl}(t) > z_i \\ z_i & f\ x_{kl}(t) \le z_i \end{cases} \tag{5}$$

This kind of map consumes a little more computer memory than a classical grid-based representation but gives much more information about the environment. One of the advantages is a possibility to go up or down stairs if robots mechanics allows it.
A typical example of 2.5D maps is presented in figure 8. One can see stairs and walls on the figure. If robot is available to walk up the stairs this kind of map allows to find a proper path.

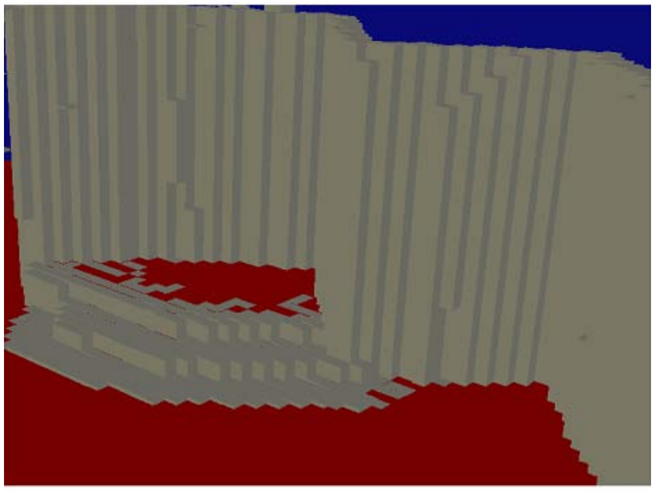


Figure 8 An example of 2.5 D map

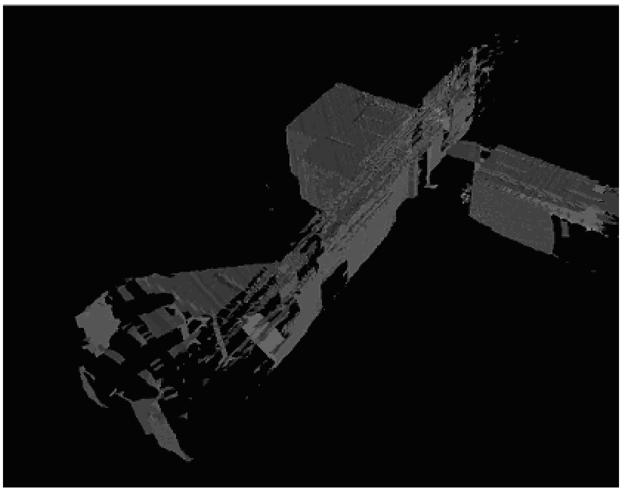
4 Extended 2.5D

A typical 2.5D map has one main disadvantage. If there are more obstacles than one in a cell the map memorizes only the highest one. For example, if there is a small obstacle on the floor which could be easily passed by the robot and above it there is another obstacle, for example, a lamp under a ceiling, then the two obstacles are remembered as one and for the system the cell in occupied. To solve this problem extended 2.5D maps are used. In such a map in each cell a list of minimum and a maximum height of each obstacle is kept.

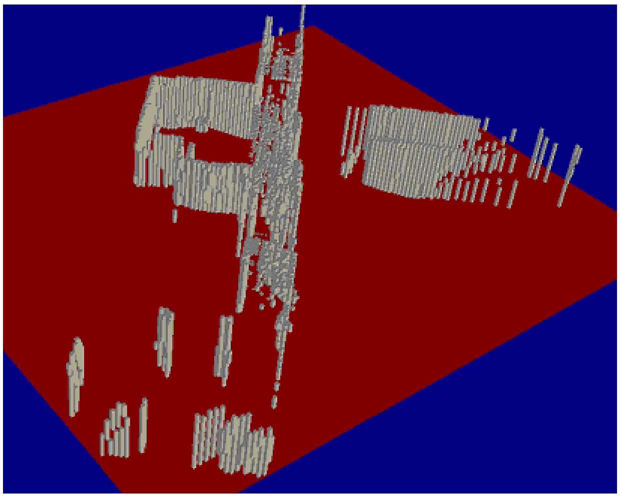
Figure 9 presents an example of such 2.5D map. There is a corridor and rooms on the pictures. In the corridor there are pipes under the ceiling. Due to using an extended 2.5D map the cells in corridor under the pipes are available.

The algorithm of building extended 2.5D map of the environment consists of the following modules

- *Initialisation*
  A value *n=0* is attached to each cell of the map. The value n is the number of objects which have been detected in the cell.
- *Computing the coordinate of the obstacles*
  For the given values of parameters: (r_i, ϕ_i, θ_i), The coordinate (x_i,y_i,z_i) are computed using formulae 3.
- *Computing the coordinate of the cell*
  The coordinate of the cell *kl* of the grid based map are computed using formulae 4.



(a)



(b)

Figure 9 a) and b) Examples of an extended 2.5D map.

- Updating

  Value z_i is the input value to the cell *kl*.
  If the value n=0 then the new state of the cell *kl* is computed as follows:
  1. new record is created,
  2. $record^0.min=record^0.max= z_i$
  3. n=1

  If the value of n>0 then records from 0 to *n-1*are checked.
  If for some k<n:
  1. $record^k.min \le z_i \le record^0.max$ then the process is stopped
  2. $record^k.min > z_i$ and $| record^k.min- z_i|<\varepsilon$ then $record^k.min =z_i$
  3. $record^k.max < z_i$ and $| record^k.max- z_i|<\varepsilon$ then $record^k.max =z_i$
  4. in other cases the number n is increased by one, a new record is created, $record^0.min=record^0.max= z_i$

5 System Architecture

A classical multilayer CNN was used, shown in figure 3. Neurons are positioned on a regular network. Input value u_{ij} to a cell c_{ij} is the value indicated by the laser properly to the horizontal angle $i\Delta\varphi$ and vertical angle θ_j. Based on u_{ij}, horizontal angle φ_i, and vertical angle θ_j it is possible to figure out the position (x,y,z) according to (3). For every point *p*=(x,y,z) a normal vector to the surface is calculated. In our algorithm the normal vector is a product of vectors *p1* and *p2*, where *p1* and *p2* are figured out from neighborhood points to the analyzed point. Data obtained from a laser are noised and if the distance between points is small (less than a laser measurement error) then the vector product error may be high.

In the algorithm *p1* and *p2* are figured out not from the closest points to the explored point *p*, but from points (x_i, y_i, z_i), where the following inequality is fulfilled:

$$\varepsilon_1 \ge \sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2} \ge \varepsilon_2, \quad (6)$$

where ε_1 and ε_2 are fixed thresholds. The normal vectors are output signal of one layer and input signals to the next one.

The dimension of layers are the same. An analysis of normal vectors neighbourhood allows to do a surface classification. The idea behind the method is presented in figures 10 and 11.

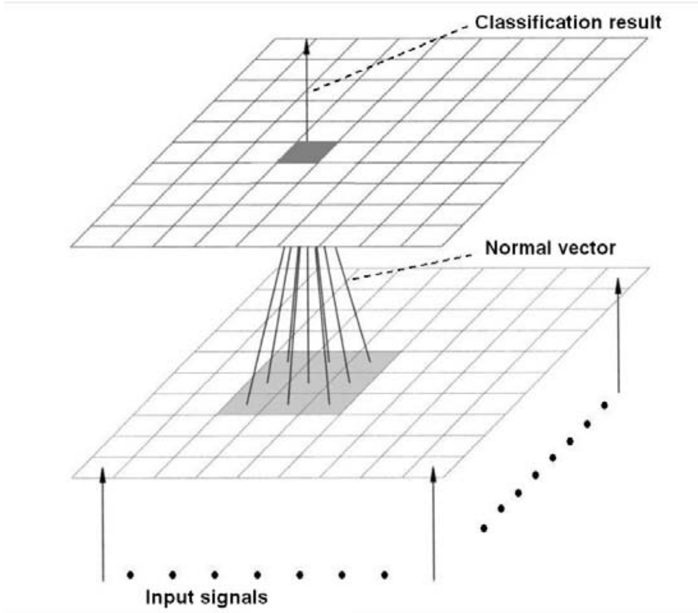


Figure 10. Network structure.

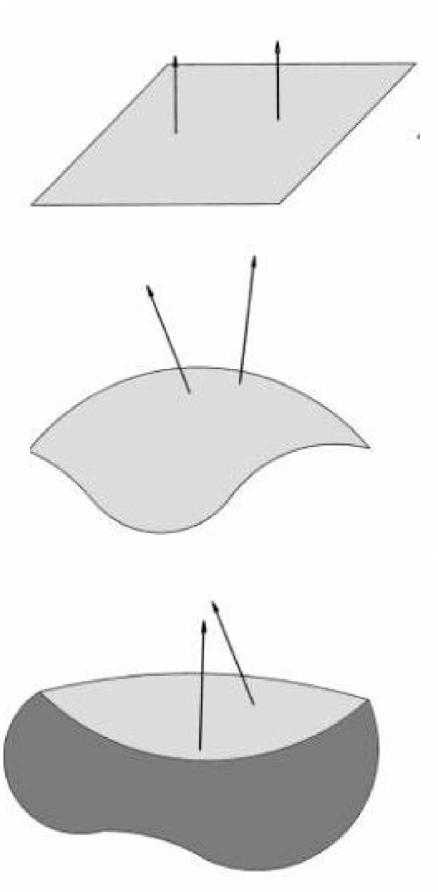


Figure 11. Normal vectors to different surfaces.

If a point belongs to a surface, then the normal vectors of the neighbouring cells are parallel. Figure 12 presents results of data classification where normal vector is figured out based on respectively the closest and adaptive neighbourhood. In Figure 12 a different colors at the bottom represent negative influence of error in taking measurements to classification (here shown as different grey scale). This is a good example where the closest neighbourhood does not give proper results. Due to noised data the closest neighbourhood does not provide a possibility to distinguish a flat surface. Taking into consideration wider neighbourhood, noised data do not influence planes recognition, which is shown on Figure 12 b.
Figure 13 shows a dependency in normal computing for different distances between the points. The Y axis represents the angle given in degrees.

If a set points $\{x_i, y_i, z_i\}_{i=1..N}$ lies on a surface:
$n_x x + n_y y + n_z z + 1 = 0$,

then using the regression method, we are looking for such n_x, n_y, n_z parameters, that the function S has a minimum.

$$S(n_x, n_y, n_z) = \sum_{i=0}^{N} (n_x x_i + n_y y_i + n_z z_i)^2 \qquad (7)$$

Values n_x, n_y, n_z fulfil the equation:

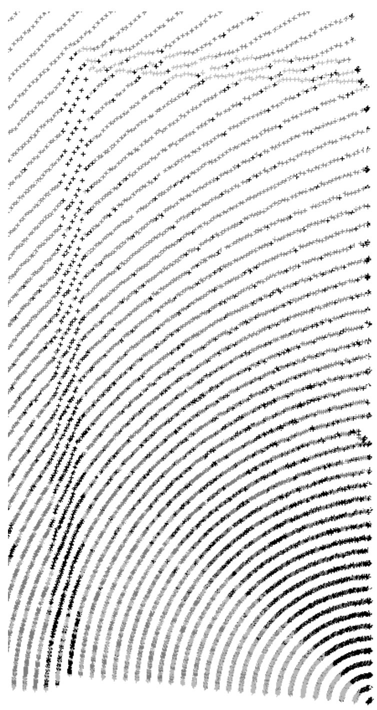
$$\begin{aligned} \frac{\partial S}{\partial n_x} &= 2\sum_{i=1}^{N}(n_x x_i + n_y y_i + n_z z_i + 1)x_i = 0 \\ \frac{\partial S}{\partial n_y} &= 2\sum_{i=1}^{N}(n_x x_i + n_y y_i + n_z z_i + 1)y_i = 0 \\ \frac{\partial S}{\partial n_z} &= 2\sum_{i=1}^{N}(n_x x_i + n_y y_i + n_z z_i + 1)z_i = 0 \end{aligned} \qquad (8)$$

Equitation (7) may be written as (9).

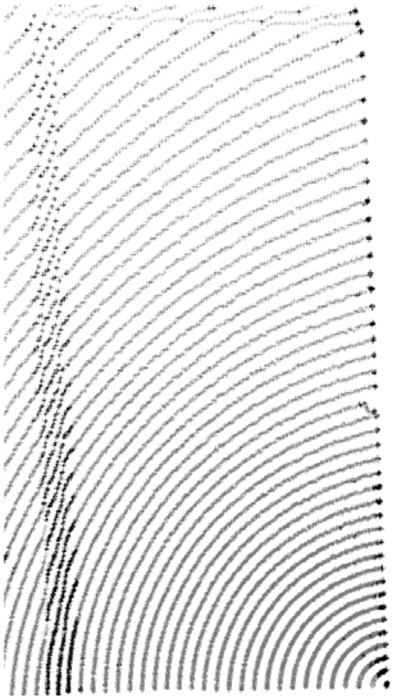
$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i z_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} -\sum x_i \\ -\sum y_i \\ -\sum z_i \end{bmatrix} \qquad (9)$$

Uncertainty of the surface position defines the equitation (10):

$$\sigma = \frac{1}{N}\sum_{i=1}^{N} \frac{|n_x x_i + n_y y_i + n_z z_i + 1|}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \qquad (10)$$



(a)



(b)

Figure 12. Points classification a) neighbourhood=1, b) adaptive neighbourhood.

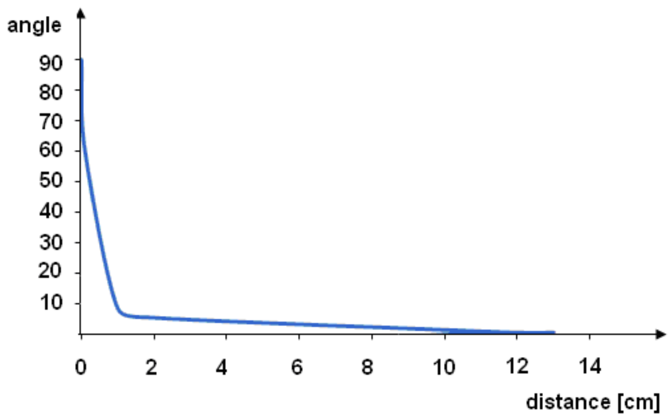
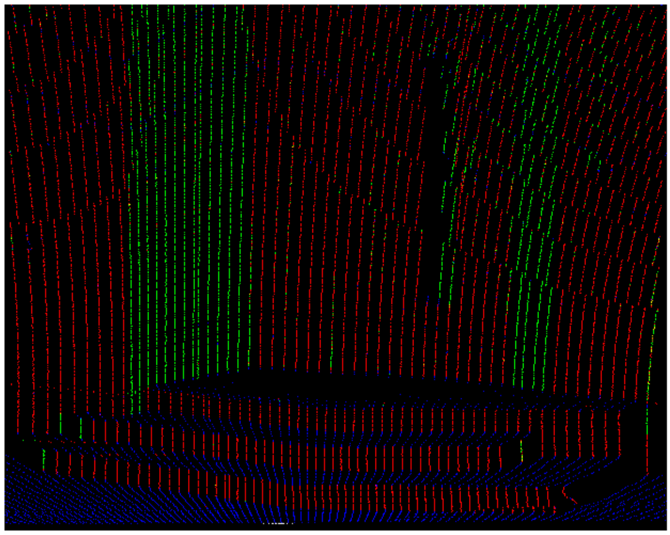


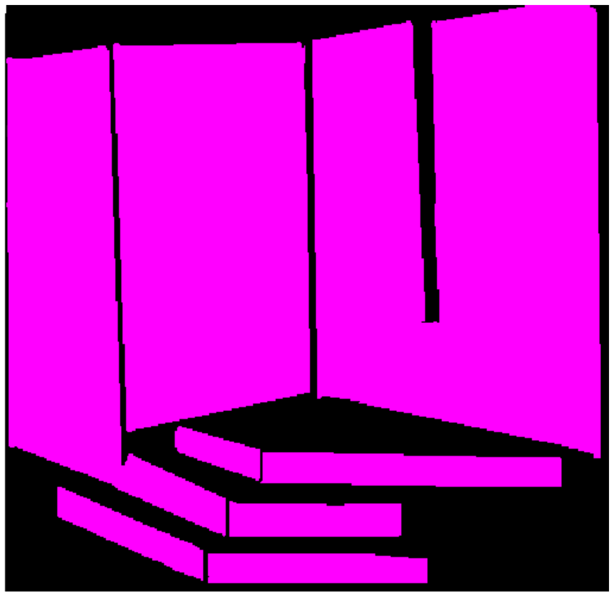
Figure 13 Normal computing error disposition

5 Experiments

In the experiments data from figure 7 was taken in order to detect planes from the data. Figure 14 presents the results of the experiment. Colors red, green and blue represent different directions: parallel to the viewer, perpendicular and horizontal respectively. Figure 14a) shows walls marked in different colours, which proves that the surface has been classified correctly. Figure 14b) shows that three different planes detected as well as some steps. One can see that small lift door window was not detected at all.



(a)



(b)

Figure 14. Planes detection. a) Points classification, b)Detected planes.

Similar to planes detection, other kinds of detection may be performed, for example edge detection. Figure 15 presents the results of the vertical edges detection. The same mechanism can be used to detect concave and convex surfaces.

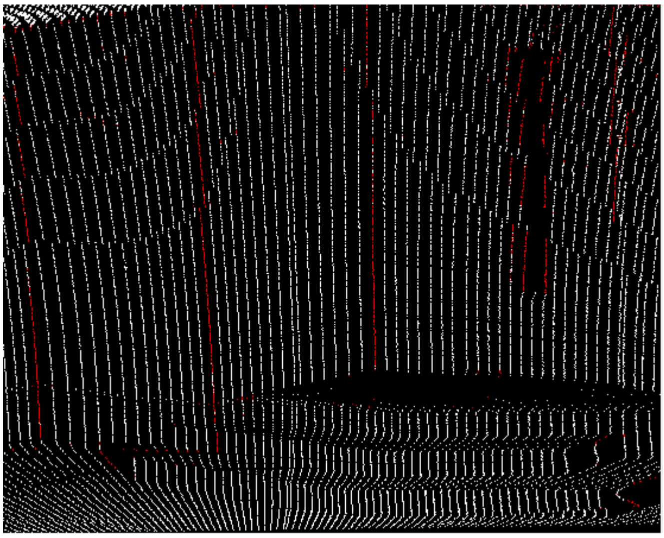
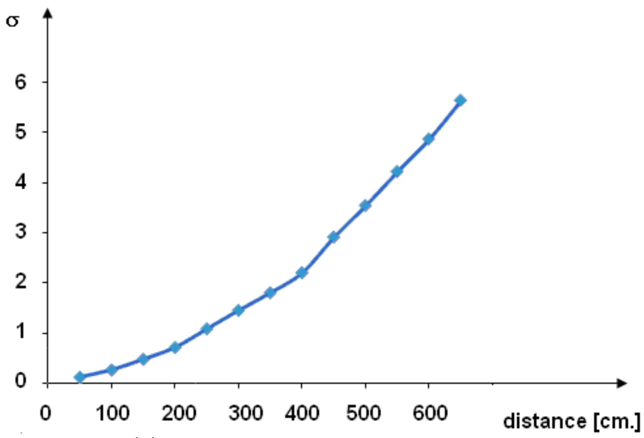


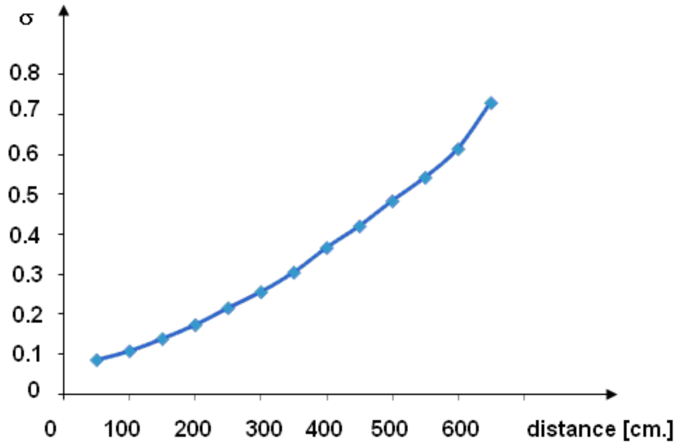
Figure 15 Vertical edge detection (red lines)

Other experiments were performed in order to fix influence noise of data to detected objects accuracy. Fig. 16 shows the difference of height detected planes to the parameter σ. The height of obstacles were 20 cm and 200cm in fig .16 a) and b) respectively. The accuracy of detected objects depends on the number of collected points.

When computations are performed sequentially time of analyzing the cloud of 186000 points takes a few seconds. The main advantage of the proposed solution is a possibility to start the computing before the whole data is obtained.



(a)



(b)

Figure 16 Parameter σ in different height of the obstacle

4 Conclusion

In the article a method of using Cellular Neural Networks (CNN) in segmentation data from a 3D laser was presented. Our research is a base of a bigger topological system to describe robots environment. Such description will allow a robot to understand human commands given in a natural way, e.g. go the door.

4 Acknowledgment

This work has been supported by Ministry of Scientific Research and Information Technology, State Committee for Scientific Research, grant no. 4311/B/To2/2007/33.

References:

[1] R. Floryczyk, *Robot Vision*. Wiley-Vch, 2005

[2] R. Triebel and B. Frank and J. Meyer and W. Burgard. *First Steps Towards a Robotic System for Flexible Volumetric Mapping of Indoor Environments*, CDROM. IAV04, 2004

[3] O. Chum and J. Matas. *Randomized Ransac with t(d,d) Test*. British Machine Vision Conference. p. 448–457, 2002

[4] W. Schroeder and J. Zarge and W. Lorensen. *Decimation of Triangle Meshes*. Computer Graphics. p. 65–70, 1992

[5] K. Niewiarowski *3D Map Building and Visualization by Mobile Robots* (in Polish). Master thesis. Warsaw University of Technology 2006

[6] G. Sakas and J. Hartig. *Interactive Visualization of Large Scalar Voxel Fields*. Visualization, Boston, USA, s. 29–36, 1992
[7] R. Sawwa and B. Siemiatkowska and J. Racz. *2.5D Map Building Based on LRF Readouts*. III-rd Int. Symp. on Methods and Models in Automation and Robotics. p. 13–15. 1997.
[8] R. Sawwa and B. Siemiatkowska and J. Racz. *A Laser Range Finder for Mobile Robot Navigation*. 28-th International Symposium on Robotics. Detroit, MI USA. s. 13–15, 1997
[9] Andersen C.S. Jones S. and Crowley J.L. *Appearance Based Processes for Visual Navigation*. Proceedings of Symposium on Intelligent Robotics Systems. s. 227–236 1999
[10] L. Chua and L. Young. *Cellular Neural Network*. IEEE Transaction on Circuit System. Vol. 36, p. 1271–1290, 1988
[11] L. Chua and T. Roska. *The CNN paradigm*. IEEE Transaction on Circuit Systems. Vol. 40, s. 147–156, 1993
[12] Weingarten, J. and Siegwart, R. *EKF-based 3D SLAM for Structured Environment Reconstruction*. IROS 2005
[13] D. Heric, D. Zazula. Reconstruction of Object Contours Using
[14] Directional Wavelet Transform. WSEAS Transactions on Computers, vol. 4, iss. 10, p. 1305-1312, 2005.
[15]P. Povalej, M. Lenič, P. Kokol. Combining multiple specialists' opinions with cellular automata to improve medical decision making. WSEAS Transactions on Computers, vol. 3, iss. 6, p. 2089-2093, 2004.
[16] F. Cupertino, V. Giordano, D. Naso, L. Salvatore, B. Turchiano Experimenting fuzzy control strategies for mobile robots on a rapid prototyping system WSEAS Transactions on Systems, 3(2), 2004
[17] I.M. Rekleitis. A particle filter tutorial for mobile robot localization. Raport, Univerity Montreal, 2004.
[18] B. Siemiatkowska and A. Dubrawski. Cellular neural network for navigation of a mobile robot. In Proceedings of RSTC'98, pages 201–211, 1998.
[19] B. Siemiatkowska and A. Dubrawski. Global map building and path planning for mobile robots. Control and Decision Making, pages 68–74, 1999.
[20] S. Thrun, D. Fox, and W. Burgard. Probabilistic robotics. MIT Press, 2005.
[21] B. Siemiątkowska, R. Kosiński, Application of Neural Networks for Safety Control, WSEAS Transaction on Computers, vol. 3 , July 2004