

Artificial Landmark Navigation System

Ireneusz Hallmann Barbara Siemiątkowska

Institute of Fundamental Technological Research,
Polish Academy of Science,
21 Swietokrzyska Str. 00-049 Warsaw,
Poland e-mail: irhal@ippt.gov.pl, bsiem@ippt.gov.pl

Abstract. Mobile robot navigation can be stated into following problems: detection of obstacles finding a free path leading from the robot's position to the goal position, estimation of the robot position and realization of the path. In our experiments the mobile robot B14 is used. The vehicle is equipped with 16 sonars, 16 infrared sensors, on-board Pentium computer system and one gray-scale camera. The grid-based map of an environment is built on the base of sonars and infrared sensors. The status of each cell of the map can be *free*, *occupied* or *unknown*. The map is built up incrementally and fuzzy method of data aggregation is used. The mapping method requires knowledge of position of the robot within its environment. Therefore a self-localization module is necessary. In presented navigation system artificial visual landmarks are used in order to compute the robot's position. We show experimentally that the resulting method is able to localize efficiently a mobile robot in partially known environment.

1 Introduction

This paper addresses the problem of mobile robot navigation in partially known indoor environment. In our experiments the robot B14 equipped with 16 sonars, 16 infrared sensors, on-board a Pentium computer system and one over-head gray-scale camera is used. The map of robot's environment is represented as a grid of cells and it is built basing on sonars and infrared sensors indications. A diffusion method is used for path planning.

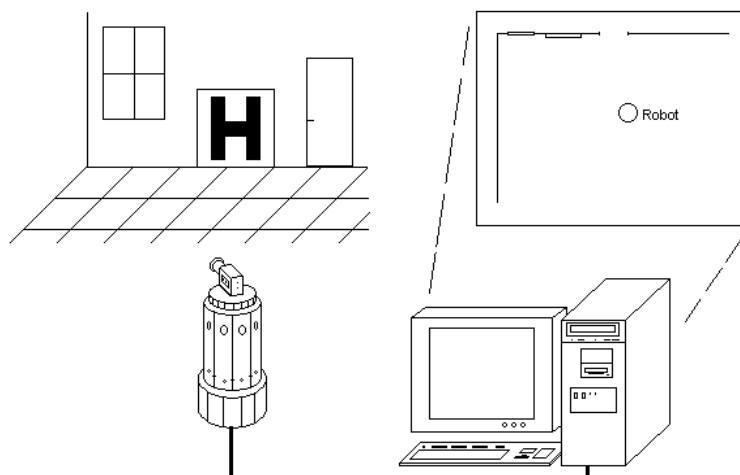


Fig. 1. *The system architecture*

A mobile robot should know where it is to navigate in indoor environment, so position estimation is the main problem in mobile robotics. Early methods of localization were developed by cartographers and navigators and employed geometric triangulations. In these methods a map of known landmarks was given and numbers of bearings to the landmarks were obtained [1]. A number of methods have been proposed for localization using some natural landmarks - corners, walls, doors [6] [3] [12]. Methods which avoid to

use maps of robot's environment have also been developed. These methods rely on Principal Component Analysis or statistic [17] [8].

The purpose of our method of localization is to update the robot's position and orientation using artificial landmarks whose sizes, shapes and positions are known. The localization is based on image analysis. The images are sent by the gray-level camera which is mounted on the top of the robot. Picture 1 presents the system architecture.

This paper is organized as follows: in section 2 map building method is reported, section 3 presents path planning method. Self-localization module is described in section 4 and followed by experimental results and conclusions.

2 Map building

The map of the environment is represented as $M \times N$ dimensional array of cells. Based on sonars and infrared sensors indications and the fuzzy method of aggregation a possibility to be occupied by the obstacles is computed for each cell [13]. The cells are labeled : *unknown* - indicates that no or little information has been obtained or obtained information is contradictory, *occupied* represents the cell occupied by the obstacles, *free* indicates that the cell is empty and here the robot can move.

For each robot's position a fuzzy set $\{Z_k\}$ is created, its membership function μ^k assigns a possibility to be occupied for each cell.

$$\mu^k : M \times N \longrightarrow [0, 1] \quad (1)$$

Another family $\{M_k\}$ of fuzzy sets is created incrementally. Membership function of set $\{M_k\}$ is defined as follows:

$$m^k : M \times N \longrightarrow [0, 1] \quad (2)$$

$$\forall_{kl \in X} m_{kl}^0 = 0.5 \quad (3)$$

$$M^{k+1} = M^k \circ Z^{k+1} \quad (4)$$

\circ - aggregation operator.

Fuzzy set M^k represents the aggregated map of the robot's environment obtained from k-different robot's positions. Aggregation function $h(a,b)$ is defined as follows:

$$h(a, b) = a + \lambda \cdot (b - 0.5) \cdot a \cdot (1 - a) \quad (5)$$

b - represents a possibility of the occupation of a cell, it is computed basing on sensors reading, a means aggregated (in k steps) possibility, $h(a,b)$ - represents a new value of possibility, λ is a parameter, which allows to define the level of influence of new information. Table 1 presents how many times the information that a cell is occupied should be confirmed to obtain the possibility value equals 0.9. It is assumed that the initial value equals 0.1.

3 Path planning

The method of value iteration is used for path planning [14][19]. The method (diffusion method) consists of three parts:

1. The initialization of an array of values

$$c_{ij}(0) = \begin{cases} R \gg 0 & \text{if the cell is a goal} \\ -1 & \text{if the cell is occupied} \\ 0 & \text{if the cell is free.} \end{cases} \quad (6)$$

Initially, all cells of the diffusion map which correspond to a region free from the obstacles, are set to 0, the cell which represents a part of an obstacle is set to -1. The cell which represents the goal position, is set to R - a very large value.

b	λ	number of iterations
0.6	1.2	36
0.6	2.0	22
0.8	1.2	13
0.8	1.8	9
0.8	2.0	8
0.9	1.2	10
0.9	1.6	8
0.9	1.8	6
0.9	2.0	5

Table 1. Number of iterations for different values of parameters λ

2. The diffusion process - update loop.

Values of all cells are updated using the following rule:

$$c_{ij} = \begin{cases} c_{ij}^{old} & \text{if } c_{ij}^{old} \leq 0 \\ \max(c_{i+j+l}^{old}, \text{dist}(c_{ij}, c_{i+kj+l})) & \text{otherwise} \end{cases} \quad (7)$$

where $\text{dist}(c_{ij}, c_{kl})$ - is a distance between cells c_{ij} and c_{kl} . The update rule is iterated until $c_{ij}^{old} = c_{ij}^{new}$ for all cells.

3. Determining of the direction of motion.

If c_{ij} is a current position of the robot, then the motion direction is indicated by the cell whose value is maximal.

Usually update loop requires number of multiplications proportional to N^4 where N - is the dimension of an array but Cellular Neural Network implementation allows to speed up diffusion process [4]. The path which is planned using diffusion method usually passes near to the obstacles. To avoid collision following modifications of diffusion loop is performed:

$$c_{ij} = \begin{cases} c_{ij}^{old} & \text{if } c_{ij}^{old} \leq 0 \\ \max(c_{i+j+l}^{old}, p_{fr}(i, j) \cdot \text{dist}(c_{ij}, c_{i+kj+l})) & \text{otherwise} \end{cases} \quad (8)$$

where $p_{fr}(i, j)$ - the possibility that the cell c_{ij} is free from the obstacles [2].

4 Localization

The intention of localization is to update the robot's position and orientation using artificial landmarks whose sizes and shapes and positions are known. The localization is considered as the calculation of the robot's position and orientation in its environment. The localization is based on image analysis. We require that the robot should be able to localize itself at every point of the scene. Therefore from every point of robot's environment at least one landmark must be visible. The localization problem is reduced to two dimensions, and therefore all calculations are simple. It is achieved by placing landmarks on the same height as the height of the robot's camera. Our method is similar to the triangulation method, but instead of three landmarks only one landmark is used. A triangle is build using special landmarks, which are composed with two parallel parts. The distance between the parts is known and two angles of view are computed based on the image taken from the camera. Figure 2 presents the difference between triangulation and our method of localization.

The localization is made in several stages. The first stage is to get the image from the camera. Then, image preprocessing is done: nonlinear distortions are corrected, and thresholding and noise filtering are performed. Then the landmark is extracted from the image, its height and center of gravity are computed and the robot's position is calculated.

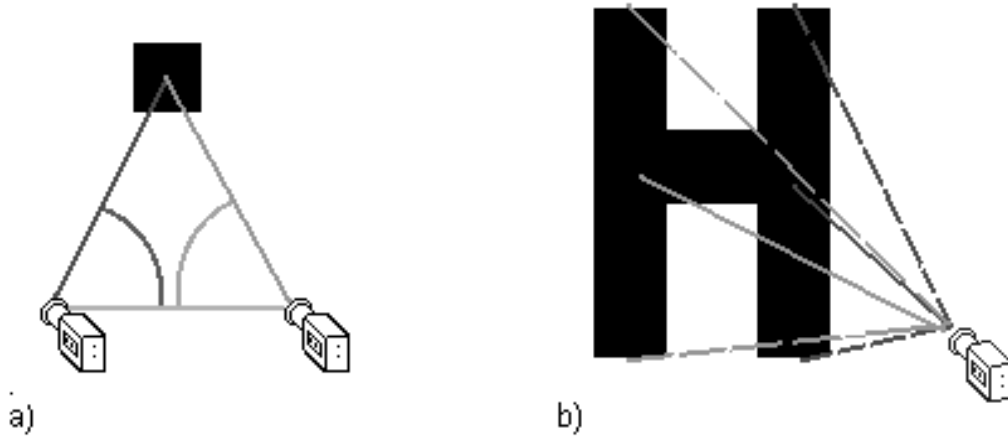


Fig. 2. Localization method: a) the triangulation method b) our method

4.1 Image preprocessing

In our experiments a mobile robot B14 with PC computer, equipped with a single camera and frame grabber is used. Images are get from the camera, which is installed on the robot's top. The camera and frame grabber work in NTSC TV standard. The image can be in any resolution up to 640x480 pixels. The number of colors can be 256 gray levels or 32K colors. For localization purpose we use gray images with maximal resolution 640x480 pixels. Processing of a gray level image is faster than processing of a 32K color image. High resolution is needed to obtain good precision of distance measurement. The size of such a picture is above 300KB. Because of fish-eye nonlinear distortions, an image taken directly from the camera cannot be used for localization. Figure 3 presents an image of flat rectangular grid taken from the camera.

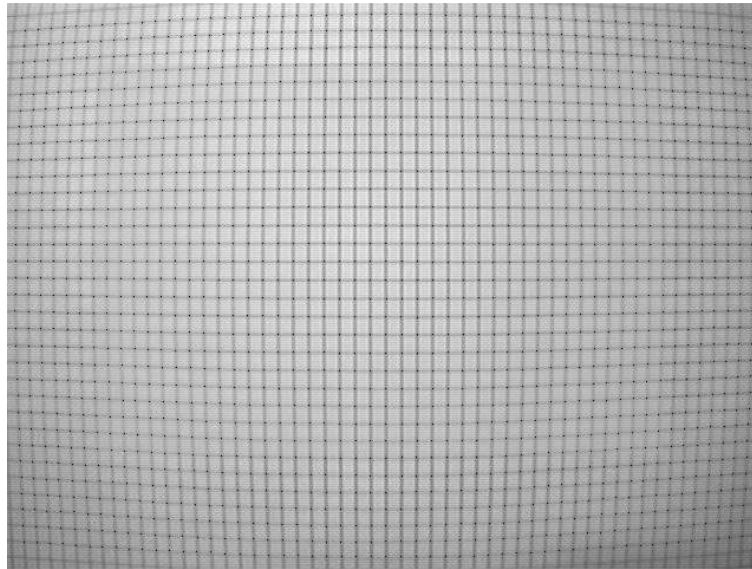


Fig. 3. Acquired image of rectangular grid

At the beginning, the correction of nonlinear distortions is performed. The method shown in figure 4 is used. Figure 4a) shows rectangular grid. Figure 4b) presents the same grid in the picture got from the camera. The grid's nodes are not at correct places but correct coordinates are known and error of displacement can be computed. The image of grid with distortions is considered as set of quadrangles, which should be

transformed into rectangles (squares). Using two-linear transformation each quadrangle is transformed into square which position and size are known. Figure 4c) presents a part of robot's environment, the image of this part is presented on figure 4d). In figure 4e) the nodes are mapped onto the image. Figure 4f) presents the result of two-linear transformation.

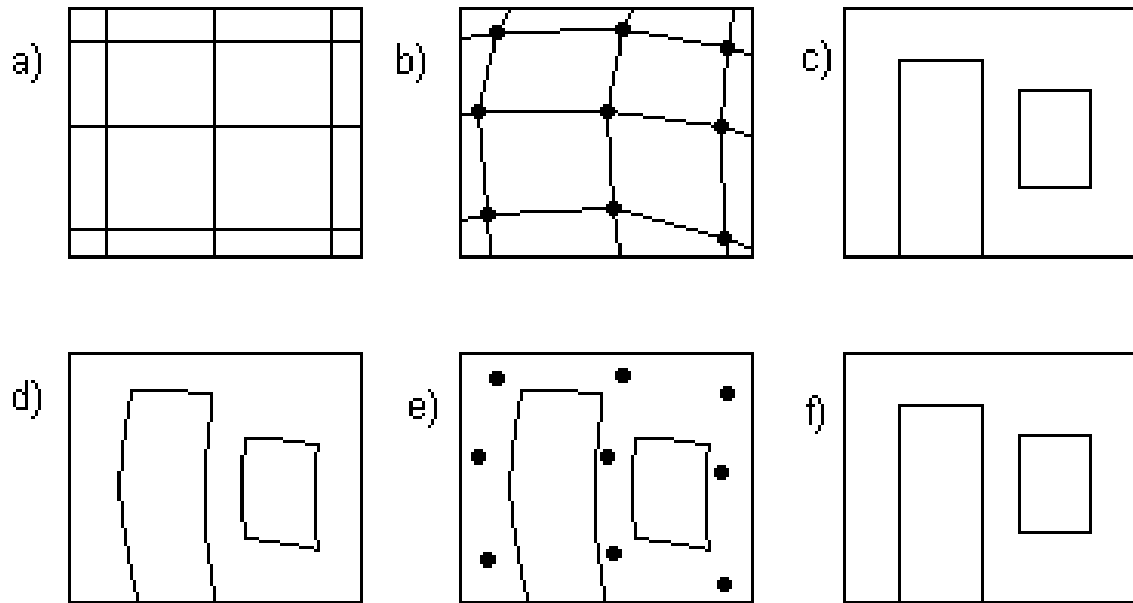


Fig. 4. *Scheme of correction of nonlinear distortions*

The image after transformation is larger than the origin, and does not contain border parts of the source image. Figure 5 shows correction of nonlinear distortions.

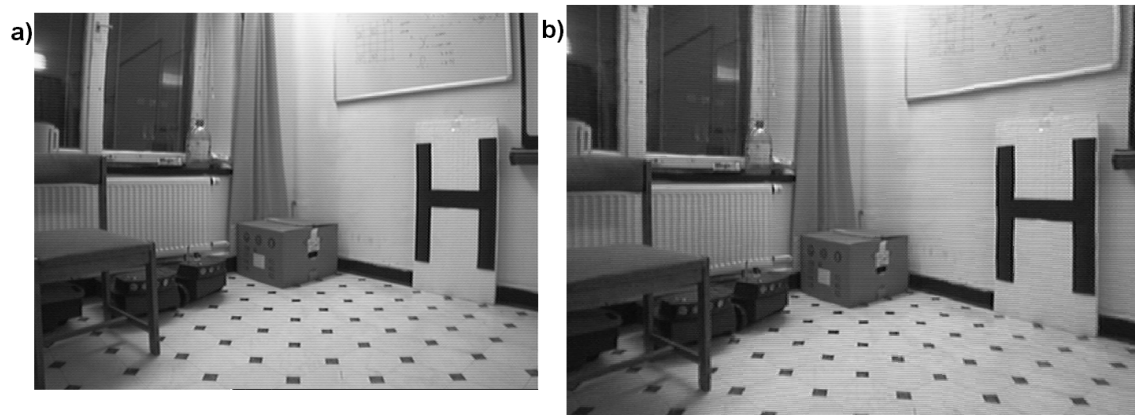


Fig. 5. *Correction of nonlinear distortions on sample image with landmark:
a) the image with fish-eye distortions; b) the same image after correction*

Next, thresholding is performed. All pixels whose value is above threshold are set as white, and other as black. In the last stage of image preprocessing median filtering is performed. The size of filter window is 3x3 pixels. Filtering is done after thresholding, because filtering of black-and-white image is much faster than filtering of gray-scaled image. During filtering of a two color image it is not necessary to sort pixels, but to

compare number of each color pixels is enough. During filtering high frequency noise and small details are removed from the picture. Figure 6 presents effect of median filtering.



Fig. 6. *Median filtering: a) the image after thresholding, with noise and small details; b) the same image after filtering*

4.2 Landmark detection

Landmarks are detected by computing moments of objects. Given a two-dimensional function $f(x, y)$ moment of order $(i+j)$ is defined by relation:

$$m_{ij} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) x^i y^j dx dy \quad (9)$$

Central moments are calculated similarly, but center of gravity is used:

$$\mu_{i,j} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) (x - \bar{x})^i (y - \bar{y})^j dx dy \quad (10)$$

where:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (11)$$

A point (\bar{x}, \bar{y}) is the center of gravity of a shape.

For a digital image, equation (9) becomes:

$$m_{ij} = \sum_X \sum_Y f(x, y) x^i y^j. \quad (12)$$

Because values of all pixels in an object are the same, equation (12) can be simplified as follows:

$$m_{ij} = \sum_X \sum_Y x^i y^j \quad (13)$$

Central moments for digital image are calculated as follows:

$$\mu_{ij} = \sum_X \sum_Y (x - \bar{x})^i (y - \bar{y})^j \quad (14)$$

Central moments are independent of object's position in the image. The normalized central moments are independent on the scale of an image and are defined as follows:

$$\eta_{ij} = \frac{\mu_{ij}}{m_{00}^\gamma} \quad (15)$$

where

$$\gamma = \frac{i + j}{2} \quad (16)$$

α	dist1	dist2	η_{00}	η_{02}	η_{11}	η_{20}
0	160	0	0.479	0.101	0.002	0.092
0	180	0	0.465	0.100	0.003	0.091
0	200	0	0.464	0.101	0.001	0.092
0	220	0	0.459	0.101	0.002	0.093
0	220	20	0.456	0.09	0.002	0.091
0	220	40	0.469	0.10	0.002	0.093
0	220	60	0.457	0.099	0.002	0.093
0	220	80	0.459	0.093	0.002	0.095
0	220	100	0.459	0.098	0.001	0.091
-12	220	40	0.462	0.092	0.002	0.097
-3	220	40	0.465	0.098	0.003	0.094
3	220	40	0.469	0.100	0.002	0.093
12	220	40	0.453	0.102	0.001	0.090
22	220	40	0.453	0.098	0.000	0.089
26	220	40	0.449	0.095	0.000	0.088
29	220	40	0.450	0.093	0.001	0.089

Table 2. Moments for H-shaped obstacles

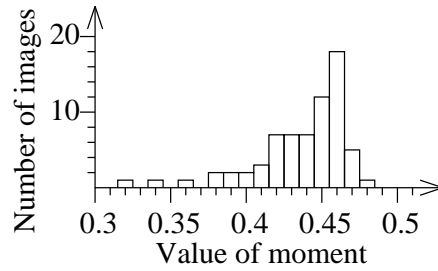


Fig. 7. Distribution of values η_{00} for H-shaped landmark

For each detected object, normalized central moments: η_{00} , η_{01} , η_{02} , η_{03} , η_{10} , η_{11} , η_{12} , η_{20} , η_{21} and η_{30} are computed. Moments of an landmark differ from moments of other objects. Values of moments of the landmark seen from different robot's position are within certain ranges of values. These ranges were found experimentally. An object is recognized as landmark, if values of all its moments fit in those ranges.

In our experiments landmarks of different shapes have been used. Table 2 presents values of normalized central moments of H-shaped landmark. Figure 7 presents distribution of moment η_{00} computed for 69 different robot's position.

When the landmark is extracted, its left and right parts are separated and their heights and centers of

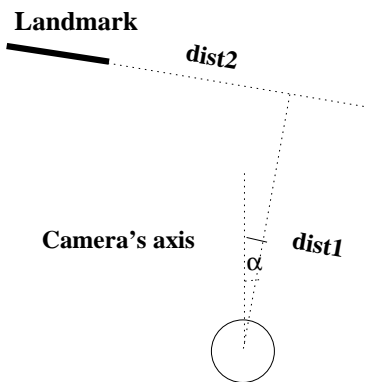


Fig. 8. The localization of the robot

gravity are calculated. Values of parameters (heights of parallel segments and their widths) are compared with information from database and the distances between the robot and each part of the landmark are computed. The position of centers of gravity allows to obtain the robot's orientation relatively to the landmark. Finally, $Dist1$, $Dist2$ and α , shown in Figure 8, are obtained.

4.3 Experiments

The main problem of robot's localization is to choose the shapes of the landmarks. That landmark should differ from natural objects, should be easy to recognize and it should give enough information to compute the position of the robot relatively to the landmark.

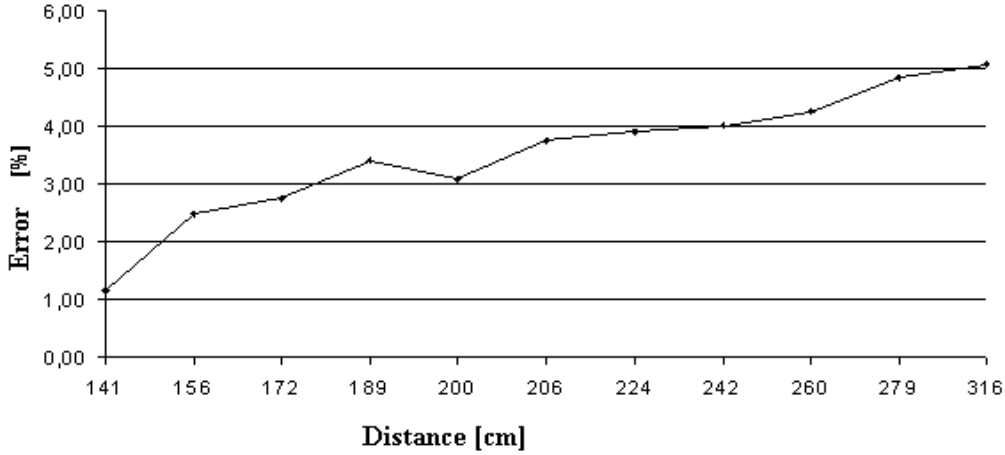


Fig. 9. Error of localization using circle-shaped landmark

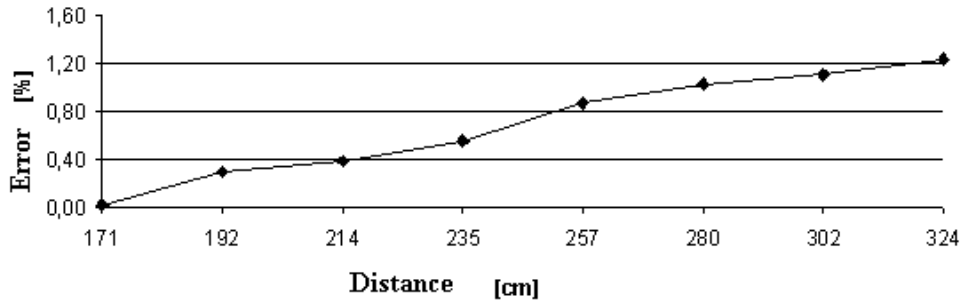


Fig. 10. Error of localization using H-shaped landmark

We make experiments using different shapes of landmarks. First landmark was circle-shaped. For such kind of object the error of localization was big. Then the landmark composed with 2 vertical parallel strips was used. Localization results were quite good, but it was too similar to natural elements of the environment. H-shaped landmark is used at present. The results of localization are the same as for two vertical strips, but the landmark differs from other standard elements of the environment.

Figure 9 presents the error of computed distance between robot and circle-shaped landmark for different positions of the robot. Figure 10 presents the error of localization for different distances, between the robot and H-shaped landmark. The error is computed as follows:

$$error = \frac{computed_distance - measured_distance}{measured_distance} \cdot 100\% \quad (17)$$

Figure 11 presents the area before the landmark, divided in regions. The regions are specified by average values of the error of the localization. The landmark is presented as two filled squares. The localization is impossible in areas A and B. If robot is in area A landmark is too large to fit into image, and in area B landmark cannot be found. Table 3 presents the error of localization in areas shown in Figure 11.

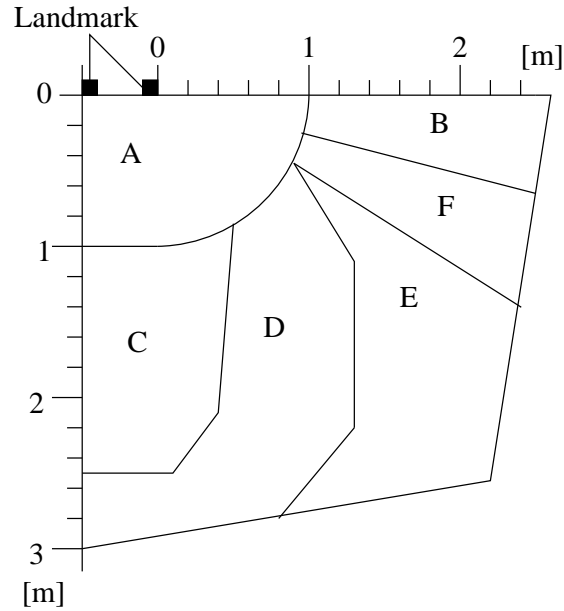


Fig. 11. *Areas of error bounds before the landmark*

Area	Error Dist1	Error Dist2	error α
A	—	—	—
B	—	—	—
C	0-10cm	0-10cm	1°
D	0-10cm	10-30cm	1°
E	10-30cm	30-40cm	2-3°
F	10-30cm	30-40cm	3-5°

Table 3. *Error of localization in regions*

5 Conclusion

The navigation system is presented in this paper. Experiments performed in real indoor environment, revealed efficiency of the discussed methods. Localization method can provide high accuracy and efficiency of on-line map building and navigation in partially known environment. Path planning method has also some advantages - it does not suffer from local minima, and it does not fail when the goal is unreachable.

References

1. Avis D., Imai H., Locating a robot with angle measurements, *J. Symbolic Computation*, no. 10, pp. 311-326, 1990
2. Bennewitz M., Burgard W., Coordinating the Motion of Multiple Robots using a Probabilistic Model, *SIRS2000*, Reading, 2000, pp. 229-238.
3. Borenstein J., Feng L. Measurement and Correction of Systematic Odometry Errors in Mobile Robots, *IEEE Transactions on Robotics and Automation*, Vol.12, No.6, December 1996.
4. Chua O., Young L. Cellular Neural Networks: Theory and Cellular Neural Networks: Applications, *IEEE Transactions on Circuits and Systems*, 35, 1257-1290, 1988.
5. Crowley J.L. Navigation for an Intelligent Mobile Robot, *IEEE Journal on Robotics and Automation*, 31-41, Vol. RA-1, No. 1, March 1995.
6. Siemiatkowska B., Dubrawski A., Neural Methods For Navigation Of Equipped With A Laser Range Finder, *A World Congress On Expert Systems, Artificial Neural Networks*, Mexico City, 1998, pp.9-16.
7. Duda R.O., Hart P.E. *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
8. Dudek G., Zhang C., "Vision-based robot localization without explicit object models", *Proc. Int. Conf. Robotics and Automation*, 1996.
9. Gonzales R.C. *Digital Image Processing*, Addison-Wesley, 1987.
10. Dubois S., Glanz F., An autogressive model approach to two-dimensional shape classification, *IEEE Trans. Pattern Anal. Mach. Intell* 8, pp. 55-56, 1986.
11. Eckstein W., Steger C., Architecture for Computer Vision, *Journal of Electronic Imaging*, pp. 244-261, 1997.
12. Wijk O., Christensen H.I., Sonar Based Pose Tracking Using Natural Landmarks, *SIRS 1999*, Portugal, 1999, pp. 245 - 253.
13. Klir G. J., Folger T. A., Fuzzy sets, Uncertainty and information, *Prentice Hall*, Englewood Cliffs, New Jersey, 1988.
14. Latombe J. C., Robot motion planning. *Kluwer Academic Publishers*, 1991.
15. Leonard J.J., Durrant-Whyte H.F. *Direct Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers, Dordrecht 1992
16. Levitt T. , Lawton D., Chelberg D. , Nelson P., Qualitative navigation, *Proc. DARPA Image Understanding Workshop*, Morgan Kaufmann, 1987, pp. 447-465.
17. Nayar S., Murase H., Nene S., Learning position and tracking visual appearance", *IEEE Robotics and Automation*, San Diego, 1994, pp. 3237-3246.
18. Siemiatkowska B., Path Planning for a Team of Robots, *SIRS2000*, Reading. 2000, pp. 343-352.
19. Steels, L., Steps towards common sense, *Proceedings ECAI-88*, 49-54, Munchen 1988.