Teodora DIMITROVA-GREKOW, Valery SALAUYOU, Karol KOWALSKI
BIALYSTOK UNIVERSITY OF TECHNOLOGY
45A Wiejska St., 15-351 Białystok, Poland

Indoor Mapping Using Sonar Sensor and Otsu Method

Abstract

In this paper we present an indoor mapping algorithm based on sonar sensor. The overall object detection and mapping experiment is based on small scale local spatial information which has been accomplished in a 2D geometrical map. Considering all drawbacks and pluses of ultrasonic sensors, we present an innovative mapping approach, applying the Otsu's method and Hit-or-Miss for sonar-data processing. The collected data are treated as a gray-scale picture. For its binarization, we applied the well-known for vision-based systems threshold calculation. Then also the morphology effect, what rises additionally the mapping accuracy, as is shown at the end of the paper. The robot is based on the education construction set LEGO Mindstorms EV3 intelligent brick on ev3dev - a Debian Linux-based operating system and Python 2.0 have been used for programming. The results are evaluated and compared with the real space.

Keywords: mapping, sonar, Otsu's method, education robot.

1. Introduction

Robot mapping is process of description robot's environment and starts from raw data gathering. Then follow data pre-processing, processing, evaluation, temporary map saving and after certain amount of repeating all or some of the previous steps, a final map is completed. An autonomous robot constructs and follows a model of its environment, appropriately to the collected and processed spatial information. An ever question is the hardware-software solutions for optimal results. The wide range of robot platforms and the steadily increasing integration of science methods is a continual source of inspiration: how to overcome the everlasting problems in this field of navigation - uncertainty and inaccuracy in the real world picturing, applying an optimal effort. New mapping approaches emerge is an ever growing and interesting mission for the robotics.

The keystones for all robotized mapping systems are the well-known sensor-actor-processor configuration on one hand and the implemented algorithms on the other. And although, the first of them is to a large extent crucial for the second one (hence for the total effectiveness of the mapping system), there are still many perspectives for innovatively solutions, even with a modest hardware equipment, as an education construction, scanning the environment with a sonar sensor.

Although there are many methods realizing mapping, part of these implementation use complicate sensor systems as laser scanner or camera and respective video-data retrieval [1- 5], other have a set of sonar sensors [6, 7] or a complex calculations [1, 6].

To optimize the information-calculation task we propose a monocular robot construction built on the LEGO Mindstorms EV3 intelligent brick with ev3dev – a Debian Linux-based operating system, programmed in Python. Data gathering is realized by one sonar sensor, mounted mobile on the top of the robot corpus. Considering all drawbacks and pluses of ultrasonic sensors, we present an innovative mapping approach, applying the Otsu's method for sonar-data processing. The collected during a round-scan data are treated as a gray-scale picture. For its binarization, we applied the well-known for vision-based systems method threshold calculation. We also tested some additional computer graphics techniques as image morphology, what improving the results further, as was expected.

This paper is structured as follows. Section II introduces the main idea of our approach. There are some details about the visual techniques. Then in section III the environmental setup for the experiment have been presented. The presentation of the experimental results are discussed in Section IV.

The stability of results and the high degree of details for the environment we got by this approach are encouraging us for further surveying in this direction.

2. Map Construction Algorithm

The mapping process was realized as two independent threads. First of them is the robot motion control, whilst the second one leads the mapping construction starting from the data gathering till to the final, ready for further use or estimation map of the explored environment. As this work explores implementation of visualization methods in indoor mapping, we assume the following simplification about the robot motion into the explored space: the robot identifies a wall after its start and points perpendicularly outside it and the condition for the end of the mapping is reaching an opposite wall.

At the launch of the program, a setting of default values for the odometry is done. Hence the robot can find the absolute reference points for its correct movement during further work. Accuracy is extremely important for all distance measurements performed by the robot and for the end result of mapping. Odometry remains a relevant source of information, especially for short distances. The next step performed by the system is to start scanning the environment by using a turret with a sonar sensor. Every collected measurement is related to the engine's angle position. This necessary information is saved as a text file, where each line contains information for a certain direction.

After successful conversion into the Cartesian system, values are saved to files, and divided into individual sets of separate readings. The next step is to check whether there is any obstacle in the direction of the next move. If there is enough free space for the next step, the robot moves forward and continues gathering data. When there is an obstacle in the sensor's range, the robot decides to end its current course and proceed to the last stage, which is to save maps in native format. The program also calculates the robot's localization - its position and orientation, as it follows in regular SLAM approaches. Thus, we have at any time the robot into Cartesian space, considering the angle of the tower, the current angle indicated by the front of the robot and the robot's position in Cartesian coordinates [X, Y] based on readings from the odometry system. The final step of the first layer algorithm is to record the collected readings in the proper form, calculating the probability for each point to be free or busy. Maps are stored as files or data sets, additionally containing the robot's current or end position.

The second implementation layer serves to construct the complete, final map of the tested environment. The end-data set from the first layer comes here through digital graphic processing. After all information retrievals are complete, the end map is visualized, but above all it is examined for its accuracy. First, we applied binarization to obtain a simple and easy interpretation of the map. After testing several different threshold methods, we obtained optimal results using Otsu's technique. Then we tested several graphics morphology effects, which showed significant improvement in the previous results.

Fig. 1 presents both versions of the algorithm: the first, basic sequence and the second extended one. The first step is data pre-processing: all environment scans contained in the cache memory are linked together. In this step, the construction and its movement of the robot, along with robot localization, are considered. The end result is a set of points, organized in M columns and N rows, referring to the map. Their values are natural numbers expressing the probability of having an obstacle at that point or not.

To ease further processing of the map, we assumed that a pixel refers to a real space of square measuring 1 cm × 1 cm. It is an optimal resolution for our tests, because it enables the created maps to contain clear information in minimum space. Additionally, this do not slow the calculations. The values of each pixel are determined from raw measurements made by the sonar. If data values exceed 255, the whole space might be additionally normalized. Such a necessity could occur if enough readings per one sonar directing were done or some of the probabilistic techniques were applied. This would particularly concern remote objects and corners.

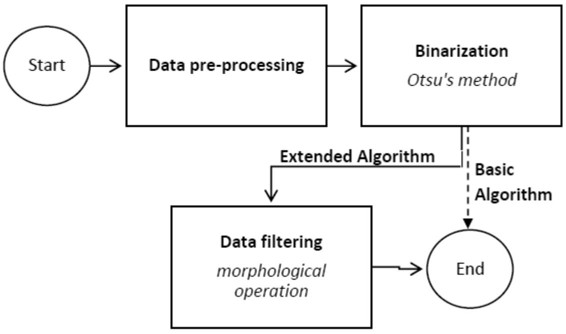

Fig. 1. Mapping algorithm using graphics methods

The next step in the mapping process is to binarize so far obtained image. Testing different threshold determination techniques we selected the Otsu's method as the leading one. During initial tests, it was compared to a number of other methods, but this approach Otsu allowed to get the best end result.

The exceeded version of our algorithm requires two additional very significant elements that allow an efficient performance of the morphological operations on any image: choice of the morphology method itself and design of a single structural element of a given size and shape. The last step is to save the constructed map in most proper for further evaluating form.

3. System Configuration

The system uses a Lego EV3 robot equipped with a sonar sensor. A rover chassis platform with a movable tower equipped with an ultrasonic rangefinder was the base mobile system.

The Lego EV3 platform-based robot is equipped with the *ev3dev* operating system from the Linux family, which enables containerization of specific tasks using virtualization service *Docker*. All communication between the equipment and the programming takes place using basic system commands. This enables using the system shell to execute command sets, which operate many times faster than normal operations in a high level language. The implementation language was Python 2.0.

For conducting our experiments we assumed next general points:

- the robot only examined closed spaces such as hallways or rooms during its work,
- to do comparison tests of the created maps, original maps of the spaces were created and only used for comparisons after reconstruction was completed,
- the robot's surrounding environment was static without any dynamically changing elements,
- the walls of the examined spaces were always perpendicular to each other and were treated as unchanging surrounding elements,
- at the start of environment examination, the robot was not familiar with its surrounding environment nor did it have additional information on obstacle location,
- obstacles were also placed inside the examined space within the robot's field of view at the appropriate height,
- obstacles located within the robot's activity area measured: 26 cm × 26 cm x 14 cm,
- the test spaces were surrounded by walls that were at least 15 cm tall.

During experiments, the robot examined two different spaces that differed in the obstacles placed inside the space. The space the robot examined measured 80 cm × 100 cm and contained all obstacles at a height accessible to the robot.

The first presented space is an expansion of the baseline space with the addition of one obstacle. This element measured 26 cm × 14 cm × 16cm and was located centrally along one of the longer walls. As a result of such placement, the examined area was not symmetrical, which made matching the generated environment to the actual environment easier. The second space was characterized by the presence of two boxes placed symmetrically to the center of the space. The boxes were placed centrally along the longer walls. In this case, the main aim was to test if the robot could correctly detect the shorter walls. Figure 2 presents photos of both environments.

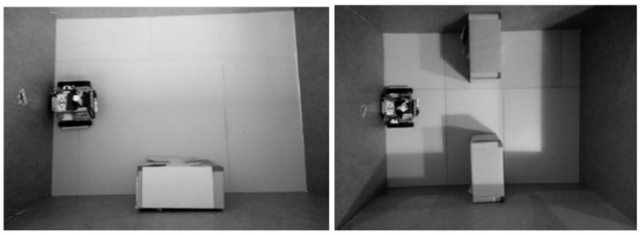
a) b)

Fig. 2. The work-spaces: a) one axe symmetry; b) two axes symmetry

4. Experimental results

To assess the robot's mapping accuracy, the map was compared to the original map of the examined space. Comparison of both images required adjusting their sizes. Despite the fact that there were not many differences, and those that existed did not exceed 3-5 pixels, adjusting the number of rows and columns in the compared images was necessary.

This problem was solved using map stretching. We used the PIL library for this, which offers image transformation methods. A larger map size is described, and then the map undergoes a transformation process. The smaller map is stretched to the size of the larger one, which translates the existing pixels or adds new ones in places where the map was stretched. The last step was to calculate accuracy error values.

After activation, the robot, by moving forward, conducted several dozen scans of the environment, wherein it performed 10 readings for each direction of the tower with the ultrasonic sensor. To describe the implemented visualization techniques, we present several temporal data sets. Data visualized from the most important mapping phases for both spaces is very interesting.

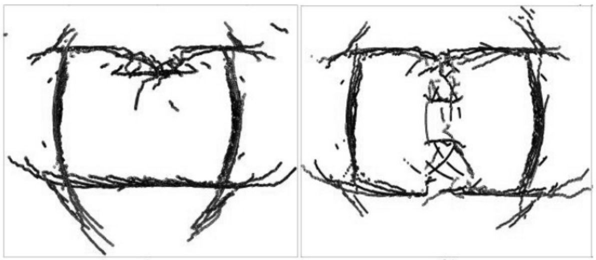
a) b)

Fig. 3. Raw-data: a) work-space A; b) work-space B

Fig. 3 presents data sets using Cartesian coordinates, which were created after combining all the map prototype readings. Despite considerable background noise, the obstacles are visible. Noise was also observed in the corners of the space as well as individual groups of points caused by sensor reading errors.

Fig. 4 presents the resulting maps created by both versions of the algorithm. Figures 4b and 4d present an improved map, which is the result of applying an additional image morphology. On the basis of data from the final maps, we calculated error values for mapping accuracy for each algorithm. Additionally, we estimated wrongly assessed areas as a fictitious obstacle, or in a more dangerous version, as a fictitious space.

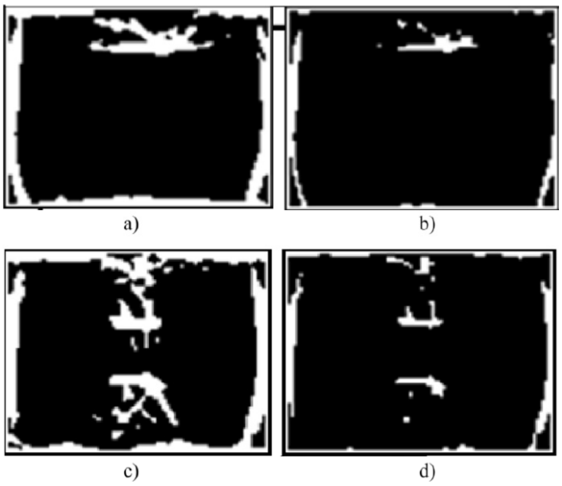

Fig. 4. Final maps: a) for work-space A only after binarization; b) for work-space A after morphology; c) for work-space B only after binarization; d) for work-space B after morphology

The results are presented in Table 1. Detection values for a fictitious obstacle FO are many times greater than for a fictitious free space FS. These are very positive results for safety reasons. Note that a greater detection error occurred for the algorithm based only on binarization. The error in this case was a maximum of 20.65% of the examined space; while the algorithm with an additional morphology was more effective, and its maximum error value was 9.825%, which is an over two-fold better result.

The experiments were repeated several times and the results are actual, with the closest to average values.

Tab. 1. Mapping accuracy results

Work-space	Algorithm version	Accuracy, %	Error, %	
			FS	FO
A	Base	84.1125	15.625	0.2625
	Extended	92.75	6.7625	0.4875
B	Base	79.35	19.650	1.0000
	Extended	90.175	8.3375	1.4875

5. Conclusions

This paper presents a novel approach to map creation. On the basis of the presented experiments, we can notice that mapping using image binarization on the basis of Otsu's method is characterized by very good results. When the experiments were additionally enriched by Hit-or-Miss morphology, the obtained results improved by 8-10%, reaching a maximum of 92.75%. This is due to the fact that morphology carried out after binarization allows for background noise reduction in the collected readings. We can also notice that algorithm effectiveness drops slightly when the number of obstacles grows. The difference is only 2.5% and this is the result of not only algorithm processes but also data reading methods as well as sensor reading errors.

The obtained results allow to be optimistic, and further plans assume examining a more complex space containing nonstandard objects. Our team's aim is to include artificial intelligence in the mapping process, which would assist in effective mapping and orientation in nonstandard spaces.

6. References

[1] Qiu H. Lu, Zhang W., Bai Y. and Fu Q.: Vision Based Mapping and Localization In Unknown Environment for Intelligent Mobile Robot. In: IEEE Intern Conference Ubiquitos Intelligence and Computing, pp. 701-706, 2014.

[2] Netramai C., Roth H., Sachenko A.: Real-Time 3D Path and Map Estimation Using a Multi-Camera System and a FastSLAM Algorithm. In: Sensors & Transducers Journal, Vol. 24, pp. 58-66, 2013.

[3] Medina Lee J. F. and Buitrago J. A.: Map generation and localization for a LEGO NXT robot. In: IEEE 2nd Colombian Conference on Automatic Control (CCAC), Manizales, pp. 1-5, 2015.

[4] Kallasi F., Rizzini. D. L., Caselli S.: Fast Keypoint Features From Laser Scanner for Robot Localization and Mapping. In: IEEE Robotics and Automation Letters, Vol. 1, pp.176-184, 2016.

[5] Patil D. A., Agiwal S. V.: Design and Implementation of Mapping Robot Using Digital Magnetic Compass and Ultrasonic Sensor. In: International Journal of Engineering Research & Technology, Vol. 4. pp. 618-621, 2015.

[6] Beinscob P. and Reinke C.: Graph SLAM based Mapping for AGV Localization in Large Scale Warehouses. In: IEEE International Conference on Automation Science and Engineering (CASE), pp. 274-279, 2016.

[7] Chong K. S., Kleeman L.: Feature-based Mapping in Real Large Environment Using an Ultrasonic Array. In: International Journal of Robotics Research, Vol 18, pp. 3-19, 1999.

Received: 15.03.2017 *Paper reviewed* *Accepted: 03.05.2017*

Teodora DIMITROVA-GREKOW, PhD, eng.

Graduated Faculty of Electronics Sofia University of Technology, Bulgaria in 1991. Postgraduate studies and doctoral thesis - Electronics and Automation at Technical University in Vienna in 1997. She is an assistant professor at the Department of Computer Science at the Technical University of Bialystok. Her research interests include synthesis of programmable systems, robotics, mechatronics, analysis and signal processing.

e-mail: t.grekow@pb.edu.pl

Prof Valery SALAUYOU, PhD, eng.

Graduated Faculty of Computer Science at Minsk University of Technology in 1982. Postgraduate studies and doctoral thesis in 1987. Habilitation in 2003 and since 2016 possesses a Professor title. He is an assistant professor at the Department of Computer Science at the Technical University of Bialystok. His research interests include synthesis of programmable systems, finite state machines design and implementation and signal processing.

e-mail: valsol@mail.ru

Karol KOWALSKI, MSc, eng.

Graduated Faculty of Computer science at Bialystok University of Technology, in 2016. Since 2016 he works as technical analyst. His research interests include computer graphics, robotics, computer networks and data warehouse.

e-mail: dam3r1@gmail.com