International Conference on Applied Internet and Information Technologies, 2016 DOI:10.20544/AIIT2016.22

Autonomous Devices to Map Rooms and Other Indoor Spaces and Storing Maps in the Cloud

Henrik Lábadi [1], Anita Sabo [1], Bojan Kuljić [2] and Goran Arslanović [3]

[1,2,3] Subotica Technical-College of Applied Sciences, MarkaOreskovića 16, Subotica, Serbia

12214193@vts.su.ac.rs, saboanita@gmail.com, bojan.kuljic@gmail.com, arslanovic.gorka@gmail.com

Abstract. The publication presents a three wheeled robot that has been designed to map rooms, halls and other indoor areas. The device uses an ultrasonic sensor for measuring distance, which is later used for both navigation and obstacle detection. Data were used later to compose a matrix – the schematic map of the room. This map could be uploaded to the cloud for later use by other 3rd party devices so they do not have to redo the mapping process again.

Keywords: Autonomous Robots, Indoor Mapping, Ultrasound, Distance Measurement, Cloud.

1. Expectations

When the robot was designed, the expectations were clear: a self-navigating robot should be built that could map indoor areas all by itself. This mapping process does not have to be very deep in terms of navigation, the limit was set to one clockwise turn from the main path as demonstrated in the Figure 1.

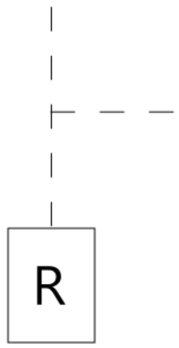

Fig.1. The robot's main path and the single allowed turn

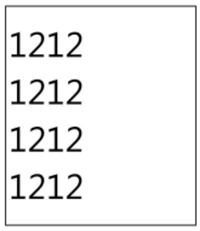

Fig. 2. Example represantiotion of the result matrix. Each number is a distance from the nearest obstacle, starting from ahead going in clockwise direction

The device is capable of finding a way back to the main path should it navigate away. Besides of exploring the room, the robot was able to do 360 degree turns in order to measure distance from obstacles in each direction.

The results were processed as an n-by-m matrix, where values of each cell were the result of the measurements as shown in the Figure 2.

2. Basic properties

The robot used in the experiments was a specifically build three-wheeled vehicle. The back wheels were driven; the single front wheel had no specific function. The robot was able to move forward and backward and turn in 360 degrees without steering. Instead, when there was a need for turning, the wheels started spinning at different speeds [1][2][3]. Because the faster wheel travels further, it rotates the vehicle in the opposite direction. The downside of this method was that because there is no on-board gyroscope on the device, the angle could not be determined precisely and the robot had to be calibrated for different surfaces[4][5][6].

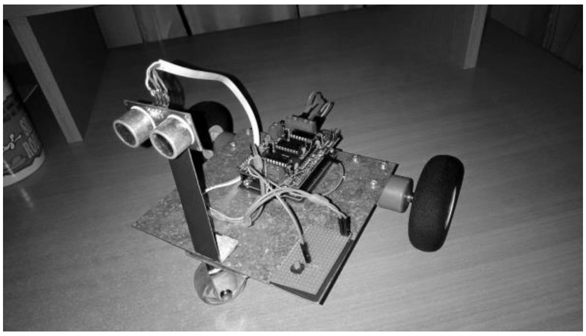

Fig. 3. The device

The device was designed to be small, so it could get under furniture and between furniture as shown in the Figure 3. However, due to the ultrasound sensor's field of view these scenarios were ruled out [7][8]. The vehicle consists of the following components:

- One Arduino board
- One Arduino shield
- Two servos
- One ultrasound sensor
- Housing
- Two wheels
- Cables

The Arduino board constitutes the main control unit. It runs the software and is wired to the shield board, the power button and the sensor. The shield board is connected to the servos, which have the wheels attached under the vehicle itself. The sensor was mounted higher above the main plain of the device.

3. Methods for measuring distance, and navigation

Distance was measured using the sensor on the front of the device. The robot starts from a random starting position. At first, it does not know anything about the world around itself. The first step is to do a full circle and measure the distances from the obstacles as in the Figure 4 and the Figure 5. The computed starting point would be determined by the software. It tries to get to the nearest corner and use that point as a new starting point [9][10][11]. This method has been chosen to simplify the mapping process by not having to worry about fragments. By this way on the other hand the vehicle could travel farther at first. When the secondary starting point has been determined, the robot tries to navigate there. In a special case, when all the distances are equal, the forward right corner is chosen as secondary starting point. Clearly, in this special case the original placement of the robot plays an important role.

1000 [T]
1652 [R]
112 [B]
1555 [L]

Fig. 4. Sample initial measurement

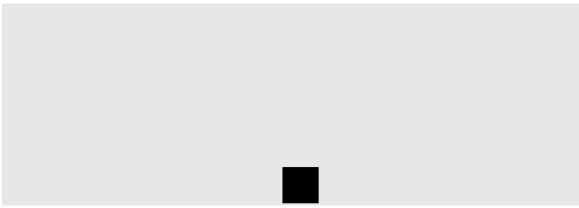

Fig. 5. Starting position of the robot in the room if the measured data is from Figure 4

Because there is no gyroscope on the robot, the operation is completely dead reckoning based. For the most cases, it is sufficient, as the robot travels mostly in a straight line. Straight runs can be measured using the onboard distance sensor. When the distance from the nearest obstacle changes, that measured value becomes the travel distance for that period. The cases where dead reckoning falls short were the turns. Without a compass or a gyroscope, the only way to execute them was by blindly powering one side of the wheels while holding the other one or spinning it backwards [12][13][14]. The problem with this method was the need for long calibration and that even with extensive calibration, the angle could not be trusted completely. When setting up the robot, different surfaces provided different grip for the wheels. It also means that if the system is set up with the parameters for carpet, it will not work properly on wooden floor or concrete.

The navigation is rather simple as shown in the Figure 6. When the secondary start point is found, the robot moves one unit from the border of the world. This is parallel to the side border of the world, towards the other side of the room. The units are flexible and can be set up during the calibration process in the code. By default, this value is 100 millimetres. After the first forward move, the robot checks its surroundings again by doing a 360 degree turn in four stops, 90 degrees each. These readings are saved and the process starts again. It runs in an infinite loop until the end of the room is found. In that position, the robot does a 90 degree turn, moves one unit forward, takes another 90 degree turn and the straight running part starts over. The robot's movements can be described as a cycle:

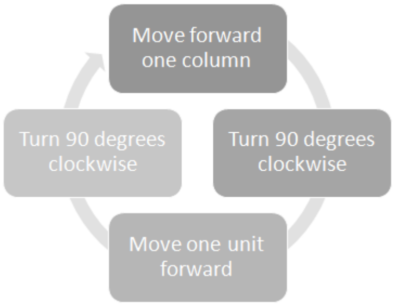

Fig. 6. The robot's movement cycle

Likewise, the straight run forward consists of two elements, "Move one unit forward" and "Turn four times 90 degrees, measure distance". In practice, this method means that these components could be defined as separate functions in the code.

4. Measuring distance and composing matrices

The distance from the closest obstacle is measured after every step. The result is a 2x2 matrix containing the values in millimetres in the following order (Table I):

TABLE I.
STRUCTURE OF THE MEASUREMENT MATRIX

Top	Right
Bottom	Left

Since the device knows its direction (up-down or down-up) from the number of the columns it scanned already [15]...[30]. If the current column number is odd, the direction is up-down. If it's even, it is down-up. In case of up-down direction, the "forward" measurement becomes "bottom", the "back" "top", the "left" and the "right" readings are swapped. In case of down-up direction, the "forward" data mean "top", the "back" "down", "left" and "right" are kept as they are. At the end of the columns, the distance readings are not stored, they only signal to the software whether the end of the process was reached. The data are saved in the following form:

```
start
[0, 0, 1000, 1000]
[30, 0, 970, 1000]
…
```

```
[1000, 0, 0, 1000]
eoc
[1000, 30, 0, 970]
[970, 30, 30, 970]
…
eoc
…
eof
```

After the room has been processed, the results were uploaded to the computer, where an application converted those data to a matrix, which represented the schematic map of the room.

The Figure 8 depicts the experiment which was performed in the room 3x3 m^2. Objects in the room were the table, the square box and the round box. It was concluded that the errors occurred in instances where the objects were close to the walls, but that had no effect because those distances were smaller than the width of the robot.

5. The cloud service

Probably the most important part of the project was the potential unified cloud based storage of the mapped rooms. A proof of concept version of this service has been implemented with some basic functionality [31].

Matrices can be uploaded from computers, tablets and mobile phones, in the future automatic upload from the devices themselves will be possible. This would enable seamless working and uploading, maybe even without human interaction. It could lead to scenarios, when the robot would map a whole factory and when the battery gets depleted it would find a charging station, continue the work and when each room is finished, the work would be uploaded to the internet.

Currently one can upload new maps and browse existing ones, search by several criteria and download the required maps. Users can also view the matrices in real time using the online interface.

robomapstore

Map name	Uploaded by	Tags
livingroom	hlabadi	home, living room, house
factoryhall	hlabadi	industry, factory, hall
schoolX	hlabadi	industry, education, school, university
schoolX2	hlabadi	industry, education, school, university

Fig. 7. The interface of the map store

As it can be seen on the figure, the interface is really simple and it was designed for functionality. The interface has three columns:

- The name of the map
- The uploader
- The tags for easy finding

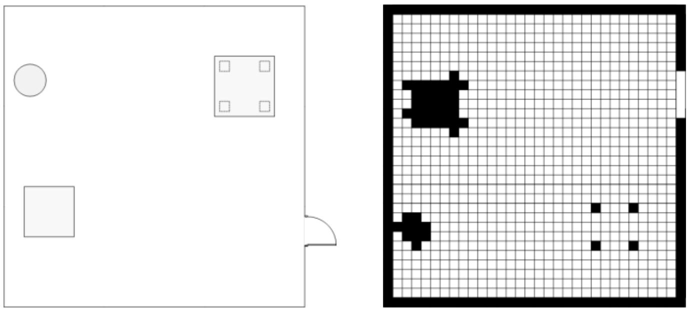

Fig. 8. The representation of the actual room and coresponding map matrix

None of these fields has to be unique, maps with the same name are allowed. On the other hand, in case of some industrial designs, it might be required for the map to be private. This is not implemented in the current version, but requires only a simple "where" clause to be added to the current SQL (Structured Query Language) query.

When one clicks on the map name a selection of options is shown. The user can either download the map in a .robomap file, they can open it in the online viewer or they can copy the link. In future versions, push-to-device support will be added, which would enable automatic upload to compatible devices.

Because the above mentioned cloud based storage engine makes portability and the delivery of the map files really easy, a decision has been made to write a specification for devices that are supported by this cloud delivery system. Most importantly, Azure IoT Hub was to be used. Any device that supports Azure IoT Hub is mostly compatible with the delivery method. Another solution was to poll the server each thirty seconds.

Because Microsoft supports a wide variety of devices, in most cases no 3rd party implementation is required when implementing the communication module of the solution.

Processing the .robomap files can be difficult, too. The robot developed for the experiment uses an Arduino board, where the sensors' readings were used to form the matrices. That means, if the inverse is performed, then the robot could be driven through the matrix by itself. While Arduino is popular, the main target for the map consuming feature was Raspberry Pi. It can be programmed using C# in .net, many libraries are available for the platform. Raspberry Pi also runs Windows 10 IoT Core.

6. Shortcomings and future improvements

After the experiments were done, some serious shortcomings were found in the solution, which were not foreseen during the design process. The most important one was the lack of compass or other sensor to measure the exact angle of turning. This would make the lengthy calibration process obsolete and improve the efficiency of the measurements, because the columns would be perfectly straight. This addition could also help eliminate the even-odd column number based direction detection. This is the most important problem, and as such, it has the highest priority to fix.

Another improvement area would be the addition of steerable front wheel or wheels. It would increase the turning radius, which could affect the minimum width of the columns, but would reduce the wheel drag of the unpowered (front) wheel(s) and fix the issue with the sometimes uncontrollable turning front wheel. This addition requires deeper modifications to the way the robot is controlled.

Regarding the sensors, other ones will be added to overcome the limitations of the ultrasound sensor, namely the blind spots occurring at angles around 45 degrees. It could also solve the too wide angle of the sensor and because of this, the vehicle could enter tighter spaces between furniture.

7. Conclusion

During the development and the process of experimenting with the robot, it has been found that even though the device is capable of mapping rooms, halls and other indoor areas, it currently has numerous limitations. The current stage is considered as phase 1 of the overall development, with features to be added to improve the precision and the efficiency.

The main goals however were met. A robot has been developed, which can create a map of the room, process it, and format it as a matrix. That matrix can be uploaded to the cloud to be consumed by other similar smart devices.

8. References

1. Ernesztina Zuban, Henrik Labadi, Istvan Balogh, Kornel Kovacs, Zlatko Covic: Digital Radar Aided Cane, 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY), 2012.

2. Ernesztina Zuban, Henrik Labadi, Kornel Kovacs: Detecting and processing objects using radio waves and opto-electrical methods, 3rd International Conference on Cognitive Infocommunications (CogInfoCom), 2012.

3. Borenstein, J. (Johann); Feng, L. (Liqiang): UMBmark: a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots, 1994.

4. Familiar, B.: Microservices, IoT, and Azure.Apress, 2015.

5. Bräunl, T.: Embedded robotics: mobile robot design and applications with embedded systems, Springer Science & Business Media, 2008.

6. Kleeman, L.: Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning, In Robotics and Automation, 1992 IEEE International Conference, pp. 2582-2587.

7. Juang, H. S., & Lum, K. Y.: Design and control of a two-wheel self-balancing robot using the arduino microcontroller board, In Control and Automation (ICCA), 2013 10th IEEE International Conference, pp. 634-639, 2013.

8. Borenstein, J., & Koren, Y.: Obstacle avoidance with ultrasonic sensors. Robotics and Automation, IEEE Journal of, 4(2), pp. 213-218, 1988.

9. L.Jiang, W.Yun-Peng, C.Bai-gen, W.Jian, S.Wei, Multi-sensor based vehicle autonomous navigation for vehicle infrastructure integration: concept and simulation analysis, Proceedings of the 2011 International Conference on Transportation, Mechanical and Electrical Engineering (TMEE), 2011, pp. 698–702,

10. E.Thurman, J.Riordan, D.Toal, Real-time adaptive control of multiple collocated acoustic sensors for an unmanned underwater vehicle, oceanic engineering, IEEE Journal of Oceanic Engineering 38(3) (2013) pp. 419–432

11. T.Luettel, M.Himmelsbach, H.-J.Wuensche, Autonomous ground vehicles — concepts and a path to the future, Proceedings of the IEEE100 (Special Centennial Issue) (2012) pp.1831–1839

12. M.Atia, S.Liu,H.Nematallah, T.Karamat, A.Noureldin, Integrated indoor navigation system for ground vehicles with automatic 3d alignment and position initialization, vehicular technology, Proceedings of the IEEE (Volume 100, Special Centennial Issue) 64(4) (2015) pp. 1279–1292.

13. F.BoninFont, A.Ortiz,G.Oliver, Visual navigation for mobile robots: a survey, J. Intell. RoboticSyst. 53(3)(2008) pp 263–296.

14. G.Desouza, A.Kak, Vision for mobile robot navigation: a survey, pattern analysis and machine intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2)(2002) pp. 237–267.

15. D.Gonzalez-Arjona, A.Sanchez, F.López-Colino, A.deCastro, J.Garrido, Simplified occupancy grid indoor mapping optimized for low-cost robots, ISPRSInt. J.Geo-Inf.2(4)(2013) pp. 959–977.

16. A.Ward, A.Jones, A.Hopper, A new location technique for the active office, personal communications, IEEE Personal Communications 4(5)(1997) pp. 42–47

17. A.Sanchez, A.deCastro, S.Elvira, G.G.deRivera, J.Garrido, Autonomous indoor ultrasonic positioning system based on a low-cost conditioning circuit, Measurement 45(3)(2012) pp. 276–283.

18. S.Elvira, A.deCastro, J.Garrido, ALO:an ultrasound system for localization and orientation based on angles, Microelectron.J.10 (2013) pp 959–967.

19. V. Gungor, G. Hancke, Industrial wireless sensor networks: challenges, design principles, and technical approaches, Industrial Electronics, IEEE Transactions on 56 (10) (2009) pp. 4258–4265.

20. J. Sladek, P.M. Blaszczyk, M. Kupiec, R. Sitnik, The hybrid contact-optical coordinate measuring system, Measurement 44 (3) (2011) pp 503–510.

21. D. Xu, L. Han, M. Tan, Y.F. Li, Ceiling-based visual positioning for an indoor mobile robot with monocular vision, Industrial Electronics, IEEE Transactions on 56 (5) (2009) pp. 1617–1628.

22. S. Saab, S. Nakad, A standalone RFID indoor positioning system using passive tags, Industrial Electronics, IEEE Transactions on 58 (5) (2011) pp. 1961–1970.

23. M. Rahman, L. Kleeman, Paired measurement localization: a robust approach for wireless localization, Mobile Computing, IEEE, Transactions on 8 (8) (2009) pp. 1087–1102.

A. Mirahmadi, A. Mansourzadeh, A novel method for construction of a point coordinate measuring instrument using ultrasonic waves, Measurement 44 (3) (2011) pp. 539–548.

A. Ward, A. Jones, A. Hopper, A new location technique for the active office, Personal Communications IEEE 4 (5) (1997) pp. 42–47.

24. N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location support system, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ser. MobiCom ’00, ACM, New York, NY, USA, 2000, pp. 32–43.

25. N.B. Priyantha, A.K. Miu, H. Balakrishnan, S. Teller, The cricket compass for context-aware mobile applications, in: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, ser. MobiCom ’01, ACM, New York, NY, USA, 2001, pp. 1–14.

26. Randell, H.L. Muller, Low cost indoor positioning system, in: Proceedings of the 3rd International Conference on Ubiquitous Computing, ser. UbiComp ’01, Springer-Verlag, London, UK, 2001, pp. 42–48.

27. M.R. McCarthy, H.L. Muller, RF free ultrasonic positioning, in: Wearable Computers, IEEE International Symposium, vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, 2003, pp. 79-83.

28. M. McCarthy, P. Duff, H. Muller, C. Randell, Accessible ultrasonic positioning, Pervasive Computing, IEEE 5 (4) (2006) pp. 86–93.

29. E. Dijk, C. van Berkel, R. Aarts, E. van Loenen, 3-D indoor positioning method using a single compact base station, in: Pervasive Computing and Communications, 2004. PerCom 2004, Proceedings of the Second IEEE Annual Conference on, March 2004, pp. 101–110.

30. E. Foxlin, M. Harrington, G. Pfeifer, Constellation: a wide-range wireless motion-tracking system for augmented reality and virtual set applications, in: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH ’98, ACM, New York, NY, USA, 1998, pp. 371–378.

31. J. Rodriguez-Andina, M. Moure, M. Valdes, Features, design tools, and application domains of FPGAs, Industrial Electronics, IEEE Transactions on 54 (4) (2007) pp. 1810–1823.