

Mapping Implementation for Multi-robot System with Glyph Localisation

Ilze Andersone¹, Aleksis Liekna², Agris Nikitenko³, ¹⁻³*Riga Technical University*

Abstract - An important prerequisite for creation of autonomous robot is the ability to create the map of the environment. Robots need to be able to construct a map of the environment and to use it for navigation. In this paper a multi-robot mapping approach is proposed, where artificial landmarks (glyphs) are used for robot localisation and only simple sensors – bumper sensors – are available for obstacle detection. The proposed mapping system creates the map of the environment and allows changing the map rapidly in case of inaccurate robot position measurements or sensor errors. The system has a central server and robots use wireless network to communicate with the server. The mapping implementation takes into account the fact that the network has limited bandwidth and only limited amount of data can be sent.

Keywords – multi-robot mapping, occupancy grid mapping, glyph localization.

I. INTRODUCTION

One of the fundamental problems in mobile robotics is environment mapping. Robots need to be able to construct a map of the environment and to use it for navigation. This paper proposes the multi-robot mapping approach where artificial landmarks (glyphs) are used for localisation and navigation and only simple short range sensors – bumper sensors – are available for obstacle detection.

The mapping approach is based on binary Bayes filter mapping [1]. The modification of algorithm supports the use of short-range sensors and the limitations of wireless network bandwidth.

For practical implementation of the multi-robot system iRobot Roomba560 vacuum cleaning robots are used that are complimentary equipped with Intel Atom CPU based embedded computing node as well as a WiFi and web camera [2].

The goal of the authors is to develop the system that delivers an effective coordination among robots in order to achieve high performance in terms of job scheduling, task allocation, trajectory planning and overlapping. For the task allocation and trajectory planning purposes the map of the environment is necessary. In this paper the mapping system is proposed which creates the map of indoor environment under the following assumptions:

- The system is a multi-robot system that consists of robots and the central server (see Figure 1 for illustration). The communication between robots and the server takes place through the wireless network.

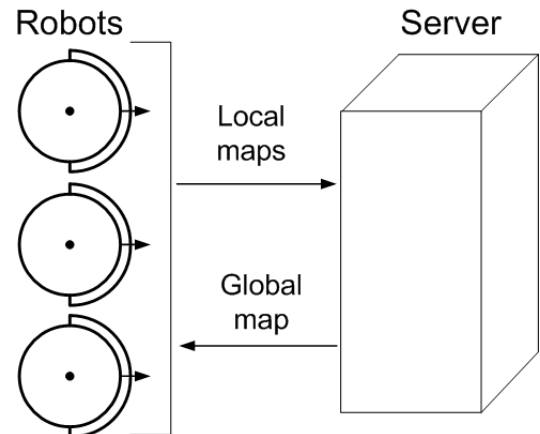


Fig. 1. The components of the multi-robot system

- The robots are able to localise themselves in the environment independent of the map by using glyph localization.
- Mapping is performed by using only short-range sensors – bumper sensors.

The system is intended for use in indoor environments with known room dimensions. Indoor environment has a high probability of change (for example, layout of the furniture). It is desirable to avoid manual redrawing of the map by a human user every time the layout of the room changes, instead, the map should be created autonomously by the robot system.

The rest of the paper is organised as follows: section II describes the related work, section III describes the glyph localisation, section IV describes the map representation in the multi-robot mapping system. Section V describes the classical binary Bayes filter occupancy grid mapping. Section VI provides a description of how the wireless network influences the mapping and how it impacts the resulting map size. Section VII explains why the classical binary Bayes filter mapping approach is not applicable to the particular multi-robot system and proposes modification for the mapping algorithm. Section VIII demonstrates the results of the proposed mapping method. Section IX provides conclusions and gives an insight into some future work.

II. RELATED WORK

Most of the robot mapping research deals with the Simultaneous Localisation and Mapping (SLAM) problem. It is entirely possible that the robot is put in an environment with no available map. In this case the robot must map the environment and localise itself in this map at the same time [3]. SLAM is recognised as a fundamental problem in robotics [4].

The robot system described in this paper, however, does not classify as a SLAM system. The glyph localisation ensures that the location of the robot is almost always available independently of the map. The robot not knowing its exact position is an exception rather than the normal situation. Therefore the approach discussed in this paper focuses solely on the mapping part of the SLAM.

The problem that is the main focus of this paper is creation of the map with primitive sensors – bumpers. Usually, sensors used for mapping are ultrasound sensors [5], cameras [6] or laser sensors [7, 8]. All these sensors are long-range sensors. In the multi-robot system discussed in this paper the camera is focused on the ceiling and used only for robot localisation in the environment. The only other sensors available are short-ranged bumper sensors, which are used for obstacle detection. With bumper sensors the only way to acquire the information about the environment is to continually compute the space that is occupied by a robot while moving and to register the collisions with obstacles.

Virtually no other authors have attempted to solve the problem of mapping the environment with short-range sensors before. In [9] one such attempt to map the environment with short-range sensors is made. The method uses odometry data from the robot's wall-following trajectory, together with readings from bumpers and wall sensors. To correct noise in odometry data, an assumption of straight walls and rectilinear turns is made. No such assumption is made in this paper.

III. GLYPH LOCALISATION

The mapping system described in this paper uses an integrated indoor robot localisation approach that is based on use of artificial landmarks – glyphs – that can be visually tracked by robots.

It is assumed that the robots will be used indoors only and the environment itself is available for necessary minor modifications in order to provide infrastructure for robots i.e. it is possible and feasible to install some special markings or beacons that provide significant information to robots about their position and translations within the environment.

To ensure that the robots can determine their precise

locations the glyphs are mounted on the ceiling and the robot uses camera to determine its position corresponding to those glyphs. With such a solution both coordinates and the heading of the robots can be acquired whenever glyphs are visible.

However, glyph data sources are not always available to the robot due to environment configurations. Therefore another important source of the position and motion information is a set of odometric sensors. In case of Roomba, each of the driving wheels is equipped with rotation encoders. When the glyphs are not visible, the robot odometry is used for position detection.

IV. THE MAP REPRESENTATION

The maps in the mapping system are represented by the occupancy grids. Occupancy grids are a very popular map representation in mobile robotics, which represents the environment as a discrete grid [10]. The occupancy values of the occupancy grid cells can acquire any value from 0 to 1 (0 corresponds to 'free' area, 1 refers to 'occupied' area). If the occupancy of the corresponding area is completely unknown, the value of the cell is 0.5. For example, the closer the cell value is to 1, the larger is the probability that this cell is occupied.

Position of each robot is represented by a triple $[x, y, heading]$, where the x is robot's X coordinate, the y is robot's Y coordinate and *heading* is the rotation of the robot in the frame of reference.

It is important to choose the appropriate representation of the map in the implementation so that it could be easily modified. The map, cell or robot size, sensor update frequency and other parameters may need to be changed therefore any modifications should be easy to implement. To provide this, the map implementation consists of the five different classes – *Map*, *MapGrid*, *Cell*, *CellStatus* and *CellContents* (see Figure 2), each of which represents a different aspect of the map:

- The *Map* class represents physical dimensions of the map.
- *MapGrid* class represents the occupancy grid. It defines the size of all cells, size of the robot (assuming it is in the shape of a circle – fitting

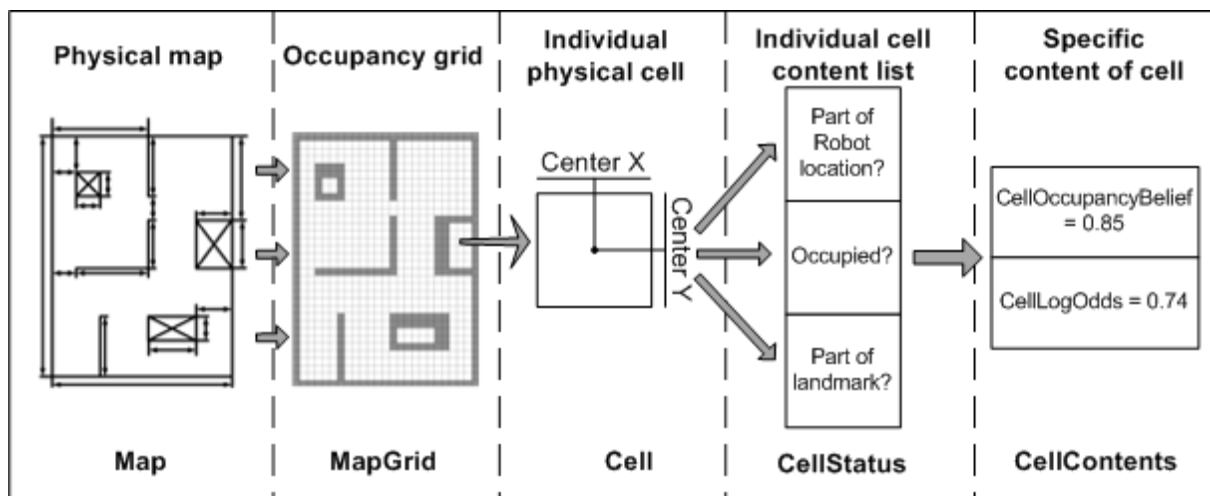


Fig. 2. The architecture of the mapping system implementation

- assumption for most vacuum cleaner robots).
- Cell** class defines physical location of an individual cell – its centre in X and Y coordinates. This class also contains two important functions for map merging that are used to detect whether the cell intersects the robot or a specific part of the robot.
- CellStatus** contains individual cell content list. Each cell may contain several different information units (for example, occupancy by static objects – walls, tables etc., occupancy by a robot, time when it was last visited etc.).
- CellContents** contains the information about the state of the particular cell. Each cell may contain several different information units (for example, occupancy by static objects – walls, tables etc., occupancy by a robot, time when it was last visited etc.).

The classes provide separation of different aspects of the map that can be easily modified when such necessity arises. Additionally, robots can create their own maps independently from the other robots and each robot can have different cell contents to support technical differences without changing the code. The only thing that must be common to all robots is the size of the map and cells.

V. BINARY BAYES FILTER MAPPING

Thanks to the glyph localisation, the X and Y coordinates of the robots are always known. Despite this, building of the map is not as simple as assigning ‘free’ or ‘occupied’ values to the cell that corresponds to a respective location. As can be seen in Figure 3, the robot simultaneously occupies several cells and all of them must be marked as ‘free’ in the particular sensor reading. Therefore, the size of the robot must be taken into account.

If a collision with an obstacle occurs during the mapping, all cells that correspond to the particular bumper sensor (left, right or both) are marked as ‘occupied’ in the current sensor reading (see Figure 4). In this case, in addition to the location the heading of the robot is also taken into account to compute the cells that border the bumper sensor.

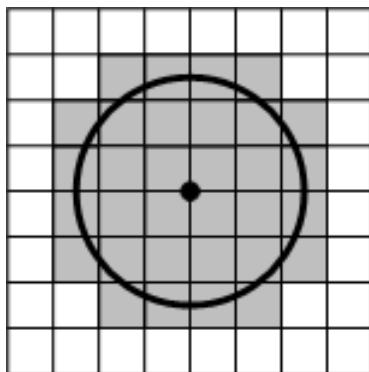


Fig. 3. Example of the free space calculation

A simple yet effective way to update the occupancy values of the map cells is the binary Bayes filter. It is applied in mapping to estimate the binary state of the cells over time. It is assumed that the state of the cells does not change over time –

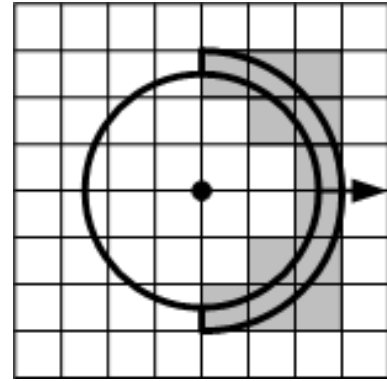


Fig. 4. Example of the occupied space calculation

that is, the environment is static. The binary Bayes filter allows maintaining the ‘memory’ of the previous sensor measurements and at the same time ensuring that the value of the cell always assumes the value between 0 and 1.

The code in Figure 5 shows unmodified cell update by using the binary Bayes filter algorithm [1].

```
Cell value update ( $l_{t-1}$ ,  $z_t$ ):
 $l_t = l_{t-1} + \log(p(x|z_t)/(1-p(x|z_t))) - \log(p(x)/(1-p(x)))$ 
 $p(x) = 1 - 1/(1 + \exp(l_t))$ 
 $l_{t-1} = l_t$ 
return  $p(x)$ 
```

Fig. 5. Binary Bayes filter mapping algorithm

The l_t depicts the log odds of the posterior belief over a binary state variable (map cell in the case of mapping) that does not change over time. The l_{t-1} is the previous log odds value of the cell. The z_t is the sensor measurement value (either *occupied* or *free*). The $p(x|z_t)$ is the probability that the value of the cell is occupied according to the current measurement. The $p(x)$ is the previous probability that the cell is occupied.

Each cell in each sensor reading is updated by using the binary Bayes filter.

VI. WIRELESS NETWORK BANDWIDTH PROBLEM

The robots in the mapping system communicate with the server that merges their maps through a wireless network. The communication network bandwidth limit is the problem that should not be neglected in multi-robot exploration [11]. Robots must also be able to communicate data among themselves and the amount of the data can be a significant load for their communication channels.

For example, if the environment is 10×10 meters large and the resolution of the map is 1 centimetre, the size of the occupancy grid map is 1000×1000 = 1 000 000 cells. Each cell may be represented as a byte that contains the occupancy value from 0 to 255 – it equals to 1 000 000 bytes ≈ 976 kilobytes ≈ 0.95 Megabytes. It means that almost a Megabyte of data would have to be transmitted each time a robot wishes to share its map even when the size of the environment is not large.

In the mapping system the maps are to be merged by one central server. To achieve the continuous updating of the global map and incorporation of the globally significant

information, the maps should be sent to the server at least once every few seconds.

The amount of data sent can quickly become too large for the wireless network to support. This is especially true, if the count of the robots increase. As it is, currently the maps of the robots are sent to the server each 50 milliseconds – that is, 20 times a second.

One of the solutions to the problem, which was adopted by our system, is the decrease of the map resolution. For the purposes of saving wireless network bandwidth and also the computational resources the resolution of the map is 10×10 centimetres. This cell size means smaller map size in terms of bytes and fewer computations required for mapping, map merging and robot path planning.

The drawback is that the map has lower resolution and a robot navigating through the environment may not be able to find some paths due to the large map cell size. However, in the real life implementation it is not a serious concern, as long as the cell size is sufficiently small compared to the robot size and is unlikely to occlude possible robot navigation paths.

VII. THE PROPOSED MAPPING APPROACH

The multi-robot system with glyph localization is not a SLAM system. The glyph localization ensures that the location of the robot is available independently of the map. Therefore the main problems that need to be solved to create the map of the environment with the existing robot platforms are the following:

- Creation of the map with short-range sensors (bumper sensors);
- The need to change some part of the map in case there are changes in the environment. The changes in the map may also be necessary in the case when glyph data source was not available to the robot at some moment due to the environment configuration. In such case an erroneous map update may occur that would require correction.
- The communication of the robot's local map with the server and the map merging of multiple robots.

A. Modified occupancy grid mapping with binary Bayes filter

Large cell size and the robot sensory equipment require modifications in the binary Bayes filter. Normally the binary Bayes filter changes the occupancy probability equally fast whether the measurement is 'occupied' or 'free'. If the initial knowledge about the cell status is 0.5 (unknown) and then two opposite measurements (one 'occupied' and one 'free') are taken, then the occupancy value of the cell after these two measurements is still 0.5 – unknown.

This situation is undesirable in the proposed mapping system because of the way the robots acquire the information about the environment – both by registering collisions with obstacles and continuously calculating space that robots occupy without collisions. Often the cells are marked as occupied by sensor readings only a few times – at the moment when the robot collides with the obstacle and the bumper sensors are activated. On the other hand, cell is marked as free

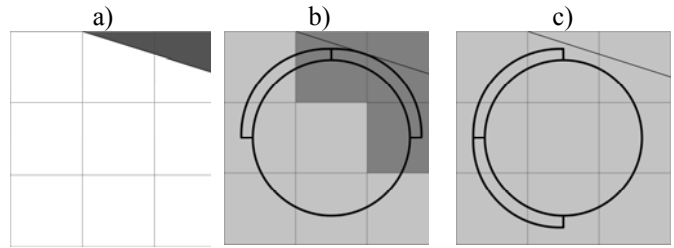


Fig. 6. Occupancy value updates in separate measurements

at every position update time when the direct contact with an obstacle is not present.

Figure 6 illustrates this situation. Let us assume that a robot with the diameter of 20 cm operates in the environment containing one obstacle in the upper right corner. The size of the environment is 30×30 cm and it is represented by 3×3 cells (Figure 6 a).

In the sensor reading when a robot bumps into an obstacle (see Figure 6 b), it marks three cells as 'occupied' and six as 'free'. The robot continues exploration and turns left (Figure 6 c). No collisions with the obstacle are present; therefore the robot marks all nine cells as 'free'. Although there is an obstacle in two cells, the robot is unable to detect it in this position. While it moves, every sensor reading (several times every second) shows all nine cells as 'free'.

If the original binary Bayes filter is left unmodified, then 'free' sensor readings quickly overwhelm any 'occupied' readings. Eventually all cells are likely to be marked as 'free' – a highly undesirable result.

To avoid this, the occupancy values of the cells are updated by binary Bayes algorithm only in cases when the sensor reading is 'occupied'. If the cell is marked as 'free' in the particular sensor reading, then the value of log odds is decreased by a constant value. The cell value update function can be seen in Figure 7.

```
Cell value update ( $l_{t-1}$ ,  $z_t$ ):
if ( $z_t$  shows 'occupied'):
     $l_t = l_{t-1} + \log(p(x|z_t)/(1-p(x|z_t)) - \log(p(x)/(1-p(x)))$ 
else if ( $z_t$  shows 'free'):
     $l_t = l_{t-1} - \text{LogOddsDecrease}$ 
 $p(x) = 1 - 1/(1 + \exp(l_t))$ 
 $l_{t-1} = l_t$ 
return  $p(x)$ 
```

Fig. 7. Modified binary Bayes filter mapping algorithm

Similarly to the classical binary Bayes algorithm, the l_t depicts the log odds, the l_{t-1} is the previous log odds value of the cell. The z_t is the sensor measurement value (either *occupied* or *free*). The $p(x|z_t)$ is the probability that the value of the cell is occupied according to the current measurement. The $p(x)$ is the previous probability that the cell is occupied.

Value of the constant *LogOddsDecrease* depends on the sensor measurement frequency and the time, in which the cell value should change from 'occupied' to 'free' (in our case it is about 3-4 seconds).

B. Merging of multiple local maps

If several robots explore the environment independently or one robot explores the same environment several times, the resulting local maps should at some time be merged into one common global map for navigation purposes. In multi-robot system with glyph localization the map merging can be easily implemented because the relative positioning of the local maps is easily acquirable. The common global glyph localization system provides all the information necessary for map merging. Additionally, the local map and cell size are equal for all the robots, which makes it easy to overlap maps for creation of one global map.

The idea of the map merging in this case is simple. At first, each robot creates a local map of its own. After some exploring time the server collects the data from robots, then creates a common global map and sends it back to the robots. The robots continue mapping with this global map. The server repeats this process regularly. Since the robots have access to their global coordinates, there is almost no risk that map merging may be incorrect. The map merging is currently under implementation.

VIII. DEMONSTRATION OF RESULTS

To demonstrate the performance of the multi-robot mapping system a mapping experiment was performed in the indoor environment that is shown in Figure 8. The environment size is 3.5×3.5 metres, which corresponds to 35×35 cells in the occupancy grid. The environment contains five obstacles – three boxes and two table legs.

The robot mapping system explored the environment and created the map seen in Figure 9. The black rectangle depicts the virtual wall that is used to constrain the robot in the experimental setting. The robot receives the signal to return to the marked region if it tries to leave. It can be seen in the Figure 9 that the robot has mapped the environment to a satisfactory extent for later navigation and all the obstacles in the environment are detected. The only exception is one of the table legs that the robot did not collide with.

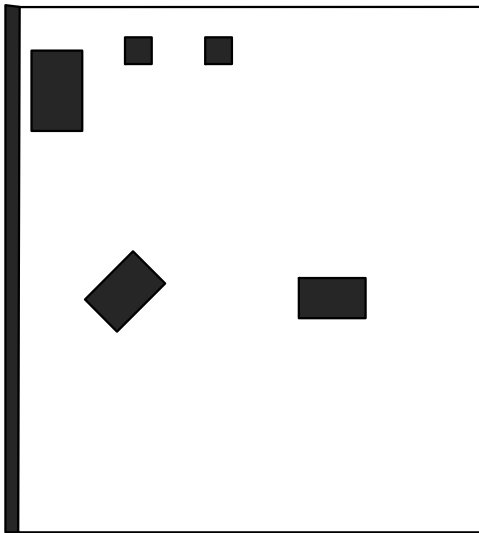


Fig. 8. Indoor environment configuration used for experiments

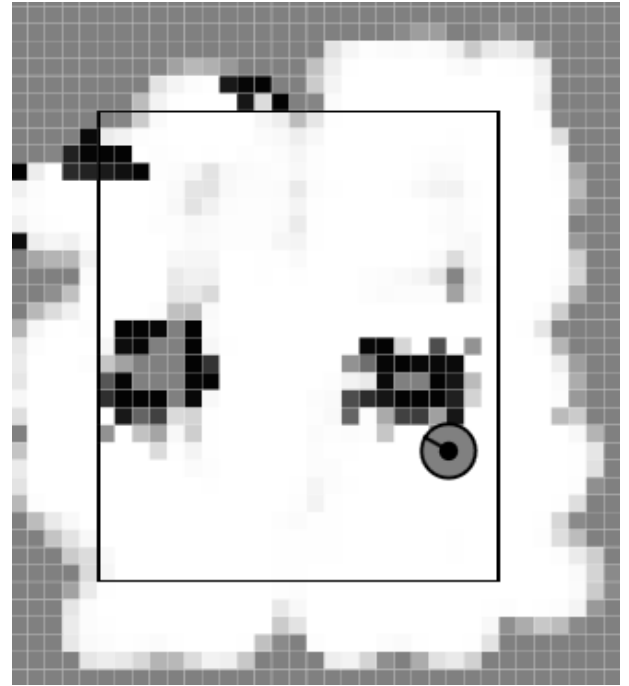


Fig. 9. The map of the environment created by using the proposed mapping approach

When the map seen in Figure 9 was acquired, one of the boxes was removed from the environment and the robot continued mapping. The new map updated by the robot can be seen in Figure 10. As shown, almost all the occupied cells that correspond to the removed box were eventually marked as free. Another change in the map is that the robot detected the other table leg and added it as an obstacle to the map.

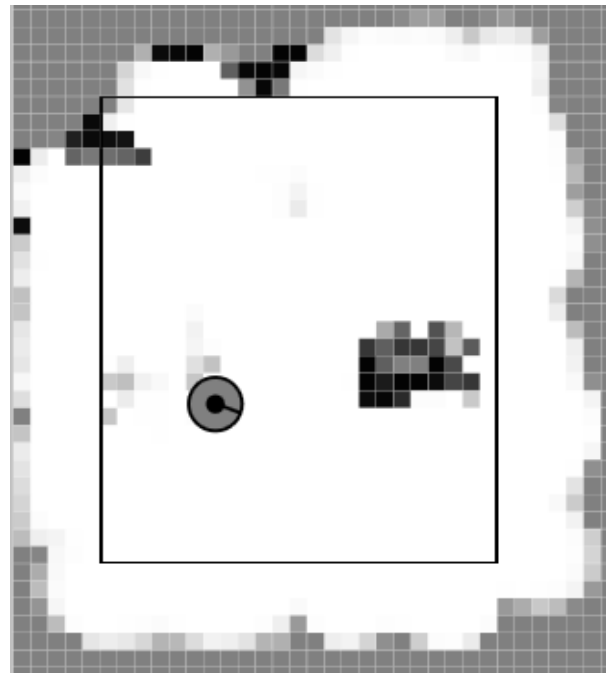


Fig. 10. The map of the environment modified by using the proposed mapping approach after removal of one obstacle

IX. CONCLUSIONS

The proposed robot mapping approach allows the mapping to be performed by the robots with the help of short-range sensors only. To achieve this, the glyph localisation system is used for detecting robot position regardless of the map.

The demonstration results show that the proposed approach allows for creation of robot maps that can be used by robots to navigate the environment and perform the tasks in specific places of the environment. The binary Bayes filter based mapping algorithm is capable of correcting the map according to the changes in cases when the configuration of the environment changes or the estimated robot position is incorrect.

Future work includes map merging implementation and further testing of the mapping in larger and more complicated environments. Another important future task is the implementation of path planning by using the acquired global map and testing its performance. Path planning is needed for the robots to be able to reach their destinations unhindered.

X. ACKNOWLEDGEMENT

This research has been supported by the European Social Fund within the project "Support for the Implementation of Doctoral Studies at Riga Technical University".

REFERENCES

- [1] S. Thrun, Probabilistic robotics. The MIT Press, 2005
- [2] <http://www.irobot.com/us/robots/home/roomba.aspx> cited: 16.08.2012.
- [3] H. Durrant-Whyte, T. Bailey, Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms, Robotics and Automation Magazine, 2006
- [4] S. Thrun. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 2003
- [5] J.-G. Kang, S.-Y. An, S. Kim, S.-Y. Oh, Sonar-Based Simultaneous Localization and Mapping Using a Neuro-Evolutionary Optimization. *Advanced Robotics*, 24(8-9), 1257-1289, 2010
- [6] Y.-T. Wang, M.-C. Lin, R.-C. Ju. Visual SLAM and Moving-object Detection for a Small-size Humanoid Robot. *International Journal of Advanced Robotic Systems*, 7(2), 2010
- [7] A. Garulli, A. Giannitrapani. Mobile robot SLAM for line-based environment representation. *Decision and Control*, 2005
- [8] A. Guez, J. Pineau, Multi-tasking SLAM. *2010 IEEE International Conference on Robotics and Automation*, 377-384, 2010
- [9] Y. Zhang, J. Liu, G. Hoffmann, Real-time indoor mapping for mobile robots with limited sensing. *Workshop on Mobile Robots*, 636-641, 2010
- [10] R. Siegwart and I.R. Nourbakhsh, Introduction to Autonomous Mobile Robots. The MIT Press, 2004
- [11] Y. Pei, M. W. Mutka, and Xi N., "Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness," *2010 IEEE International Conference on Robotics and Automation*, pp. 5460-5465, May 2010.



Ilze Andersone was born in Riga, Latvia. She received a B.Sc. degree in computer control and computer science in 2007. In 2009 she received a M.Sc. degree in computer systems. Both degrees were acquired at Riga Technical University, Riga, Latvia. She is now a PhD student at Riga Technical University.

In 2007 – 2009 she worked as a Laboratory Assistant at the Department of System Theory and Design, Riga Technical University. Since 2009 she has been working as a Researcher at the Institute of Applied Computer Systems, Riga Technical University. Her research interests include robotic mapping and robot teams.



Aleksis Liekna received his Bc.sc.ing degree in 2008 and his Mg.sc.ing. degree in 2010 from Riga Technical University. At the moment he is a PhD student at Riga Technical University. His major field of study is computer science.

He is working as a Research Assistant at Riga Technical University. His research interests include artificial intelligence and multi-agent systems.

He was awarded by the Latvian Foundation for Education for his bachelor thesis "Development and Implementation of Reinforcement Learning Model".



Agris Nikitenko has received his Dr.sc.ing. in 2006 from Riga Technical University as well as Bc.sc.ing. and Mg.sc.ing. focusing on artificial intelligence in his thesis.

Currently he is docent in the Department of Systems Theory and Design of Riga Technical University and vice dean of study affairs in the Faculty of Computer Science and Information Technology. His scientific interests cover artificial intelligence and autonomy as well as their application in robotics.

He has received award of Verner von Siemens for his doctoral thesis in 2007. At present he is member of IEEE and ACM as well as represents Latvia in NATO RTO AVT panel.