Achmad et al. 2017. *Int. J. Vehicle Structures & Systems*, 9(3), 190-194
ISSN: 0975-3060 (Print), 0975-3540 (Online)
doi: 10.4273/ijvss.9.3.12

International Journal of Vehicle Structures & Systems
Available online at www.maftree.org/eja

Tele-Operated Mobile Robot for 3D Visual Inspection Utilizing Distributed Operating System Platform

M.S. Hendriyawan Achmad[a], Gigih Priyandoko[b,c], Rosmazi Rosli[b,d] and Mohd Razali Daud[e]

[a]*Dept. of Electrical Engg., Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia*
Corresponding Author, Email: hendriyawanachmad@uty.ac.id

[b]*Faculty of Mech. Engg., Universiti Malaysia Pahang, Pahang, Malaysia*
[c]*Email: gigih@ump.edu.my*
[d]*Email: rosmazi@ump.edu.my*

[e]*Faculty of Electrical & Electronics Engg., Universiti Malaysia Pahang, Pahang, Malaysia*
Email: mrazali@ump.edu.my

ABSTRACT:

This research examines how humans work with tele-operated unmanned mobile robots to perform interaction to do the inspection in industrial plant area to produce a 3D map for further critical evaluation. This experiment focuses on two parts, the way human-robot doing the remote interactions using robust method and the way robot perceives its surrounding environment as a 3D perspective map. Robot operating system (ROS) as a tool was utilized in the development and implementation of this research which comes up with robust data communication method in the form of messages and topics. RGBD simultaneous localization and mapping (SLAM) performs the visual odometry and mapping function simultaneously to construct 3D map using Kinect sensor. The results showed that the mobile robot-based tele-operated system utilizing RGBD SLAM for 3D inspection task under ROS environment are successful in extending human perspective in term of remote surveillance in the large area of the industrial plant.

KEYWORDS:

Mobile Robot; Human-Robot interaction; Automation; 3D Inspection; Robot operating system; Kinect; RGBD SLAM.

CITATION:

M.S.H. Achmad, G. Priyandoko, R. Rosli and M.R. Daud. 2017. Tele-Operated Mobile Robot for 3D Visual Inspection Utilizing Distributed Operating System Platform, *Int. J. Vehicle Structures & Systems*, 9(3), 190-194. doi:10.4273/ijvss.9.3.12.

1. Introduction

Inspection robots enable operators to plan outages more precisely and efficiently, thereby reduce downtime. The inspection increases the availability of facilities and makes them more profitable, as well as boosting the safety of people and the environment. Eliminating individual entry in confined spaces and performing the inspection from a safe & remote location were the key motivations of this research. It can save the life of workers and allows the operator to access the areas that could not be found by anyone before by obtaining more coverage to inspect behind baffles & obstacles. While performing the inspection, the mapping is an important step when a mobile robot wants to explore the unknown environment [1-3]. The robot should be able to move and find the pathways to avoid obstacles and achieve the goal. Mobile robot navigation technology is now increasingly sophisticated, with the technology of 2D/3D mapping [4]. Many studies have been done on the implementation of 2D/3D Simultaneous Localization And Mapping (SLAM) to determine the goal position with respect to the current location of the robot in an area that has not been recognized previously [5]. The differential mobile robots are commonly used by many researchers due to the ease of kinematic run in a narrow and confined area, as in the halls of an industrial plant for vehicle manufacture.

Robot operating system (ROS) is an open source framework which has the capability to handle all software layers from low-level up to high-level. Because of that, everyone all over the world can build and share ROS stack and package for certain purposes. High demands on ROS stack from researchers lead to exponentially increase in the number of ROS stack production year to year. Therefore, ROS provides better methods which can be chosen based on our desires and purposes [6]. This research work focuses on the way human-robot doing the remote interactions using a robust method and the way robot perceives its surrounding environment as a 3D perspective map. ROS development and implementation comes up with robust data communication method in the form of messages and topics. RGBD SLAM performs the visual odometry and mapping function simultaneously to construct 3D map using Kinect sensor.

2. Methods

2.1. Differential Mobile Robot

The differential mobile robot (DMR) as shown in Fig. 1, which commonly used by researchers in addition to the

Ackermann model was used. The DMR has two independent wheel drive system that can be controlled independently. It has a turning radius that is more varied than the Ackermann, even capable of rotating exactly on the center of gravity of the robot body. So the DMR has the required manoeuvrability to avoid obstacles better than the Ackermann model in term of narrow space. Fig. 2 shows the kinematic model of the robot. This model accommodates two inputs - linear velocity (v) and angular velocity (ω) towards the target position.

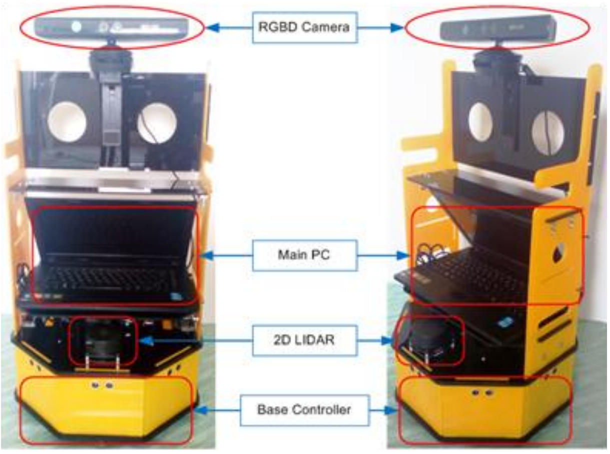

Fig. 1: Prototype of DMR

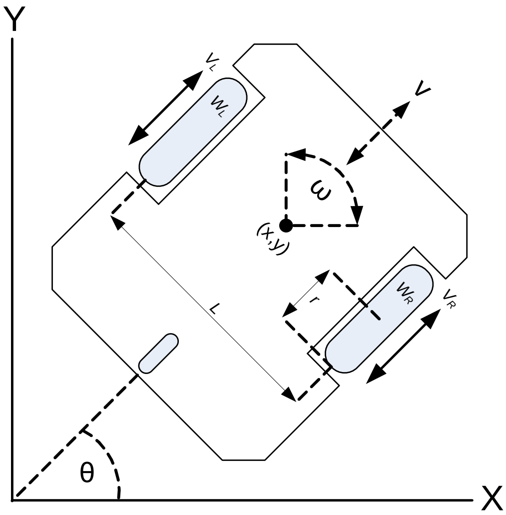

Fig. 2: Kinematic model of DMR

The DMR model adopts unicycle kinematic relations using,

$$x' = v \cos \theta \tag{1}$$

$$y' = v \sin \theta \tag{2}$$

$$\theta' = \omega \tag{3}$$

Where (x', y') and (θ') are the dynamic position and orientation respectively in discrete time. In the DMR kinematic model, v and ω are obtained from the variations in the speed of the right wheel (v_R) and the left wheel (v_L) simultaneously using,

$$x' = 0.5r(v_R + v_L)\cos \theta \tag{4}$$

$$y' = 0.5r(v_R + v_L)\sin \theta \tag{5}$$

$$\theta' = r(v_R - v_L)/L \tag{6}$$

Where r denotes the wheel radius and L denotes the distance between the right and left wheels. Balancing Eqns. (1)-(3) and (4)-(6), v_R and v_L can be derived as follows,

$$v_R = 0.5(2v + \omega L)/r \tag{7}$$

$$v_L = 0.5(2v - \omega L)/r \tag{8}$$

Based on Eqns. (1)-(8), the base controller of the robot can translate the twisted message from the primary controller containing the linear and angular velocity as set points into the speed of wheels for both of the right and left sides. Fig. 3 describes the diagram of base controller system as a part of ROS node. The base controller node subscribes v and ω in the form of geometry_msgs/Twist as ROS standard message from the other ROS node, i.e., the remote operator (human). Simultaneously, nav_msgs/Odometry is published by the base controller, which is consist of x,y,z(=0) that is obtained from the encoders. The odometry message from the core controller can be used for pose estimation together with laser and visual odometry for a better result. ROS geometry_msgs/Twist message is translated into v_R and v_L based on Eqns. (7)-(8) as set points for two PID controllers. Afterwards, the PID controllers maintain linear velocity and angular velocity of the mobile robot towards the target position as desired by the remote operator.

2.2. ROS framework

ROS is an open source software framework primarily based on UNIX platform for operating robots. It is widely conducted for robotics research in the last decade. A clear overview of ROS has been presented by [6-7]. ROS has three levels of concept: File system level, Computation Graph level and Community level.

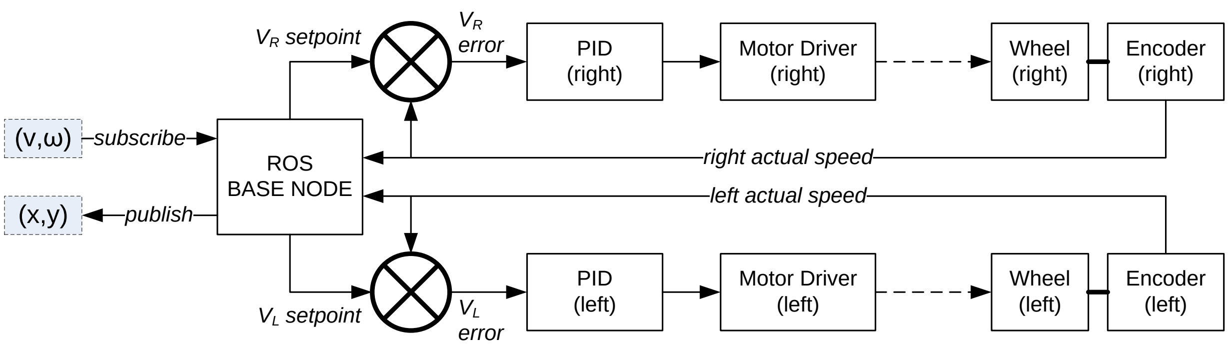

Fig. 3: DMR main controller acting as ROS node

ROS provides what the service user would expect from an operating system, including hardware layer abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between the processes, and package management. It also provides the tools and libraries for obtaining, building, writing, and running code across multiple platforms. The primary goal of ROS is to support code reuse in robotics research and development. These processes can be grouped into Stacks and Packages, which can be easily shared and distributed. ROS also supports a federated system of code Repositories that enable distributed collaboration as well.

Fig. 4 shows the basic concept of ROS with modular and distributed system capabilities. ROS Master serves as a major control for the entire distribution of messages within the network. Each node registers the identity to the ROS Master including the IP address and the name of the topic to be published or subscribed. When ROS nodes subscribe to a particular topic, ROS nodes contact the ROS master to get the IP address of other ROS nodes which publish the associated topics. The name of the topic is unique, so ROS nodes should not publish the topic with the same name in the same ROS network. Relationships between the ROS nodes were not merely publish and subscribe, but also maintain the client-server relationship. ROS node can ask the service (request) to the other nodes to do something and then get the results (response) without processing it themselves. It indicates the advantages of ROS system associated with the concept of distribution of information, so as to facilitate the users to build a robotic system which has a large-scale operation with a simple solution.

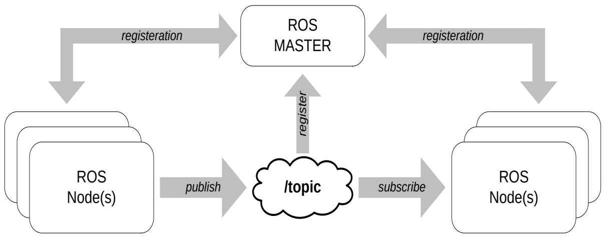

Fig. 4: Nodes communication model in the ROS environment

Fig. 5 describes the proposed ROS ecosystem where the human as a remote operator interacts with the mobile robot using joystick as a control device while doing inspection based on the 3D environment visualization presented by ROS GUI tool and RVIZ node. The base controller inside the robot subscribes the linear and angular velocity messages to drive the robot towards the operator's desired position. Kinect RGB-D sensor captures the surrounding environment in camera coverage for both RGB image and depth image. RTAB-MAP node subscribes the image messages from RGB-D camera node to perform mapping and odometry function simultaneously, and then publish 2D/3D visual information to the RVIZ node as a visual feedback to the remote operator. This continuous loop is always adhered until the accomplishment of mission unless an occurrence of certain condition that breaks the communication link, i.e., network down.

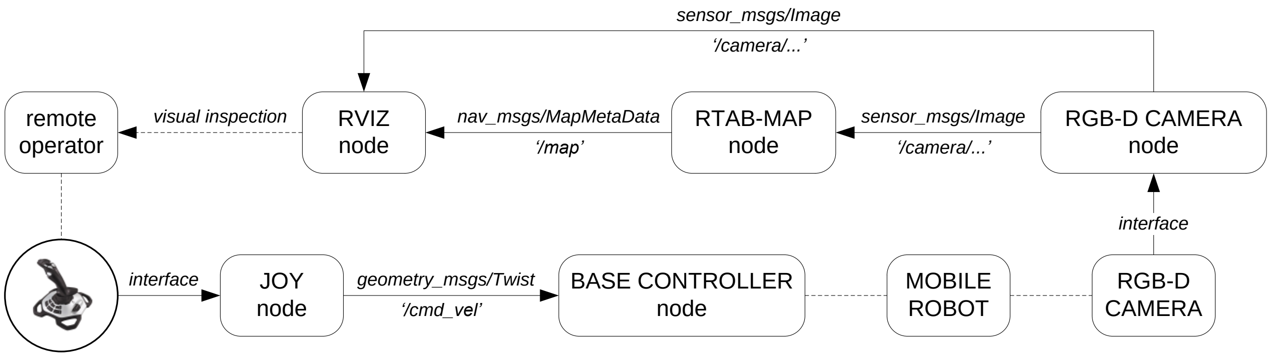

Fig. 5: Proposed ROS ecosystem for remote inspection based on 3D mapping

2.3. 3D Mapping

Real-Time Appearance-Based Mapping (RTAB-MAP) [8] is used to perform a 3D mapping with the robot simultaneously. RTAB-MAP method combines two algorithms, loop closure detection and shrinks a memory management process that limits the number of nodes [9], and graph optimization [10]. A RTAB-MAP runs on ROS environment which provides modular components so that some of them can be easily replaced. A loop closure hypothesis is evaluated by using a Bayesian filter over all previous images. A loop closure is detected when a loop closure hypothesis reach a pre-defined threshold H. SURF is performed to track visual image features and collected into the visual dictionary for likelihood calculation required by the filter. 3D pose estimation can be conducted using the calibration matrix and the depth image information given within the ROS message. The 3D scene can be constructed based on pose estimation using rigid transformation. Once a loop closure is detected, the correction can be done using the rigid transformation between the matching images based on the 3D features correspondences.

The problem of minimizing the non-linear errors is always present in robotics. The graph always expresses this error function. The graph optimization uses to find the configuration of parameters or state variables that maximally explain a set of measurements that are affected by Gaussian noise. A modified Tree-based network Optimizer (TORO) as a graph optimization approach is used by taking the root of the tree to be the latest node added to the current map graph, which is always uniquely defined across intra-session and inter-session mapping. For simultaneous mapping along with the robot, new incoming data must be processed faster than the time required to acquire them. Memory management approach is used to maintain a graph manageable online by the loop closure detection and graph optimization algorithms. The approach works as shown in Fig. 6. The memory model is composed of Short-Term Memory (STM), Working Memory (WM) and Long-Term Memory (LTM). In short, when a robot revisits particular area which was previously abandoned and forgotten, it can remember the area incrementally if at least one node of this area is still in the WM.

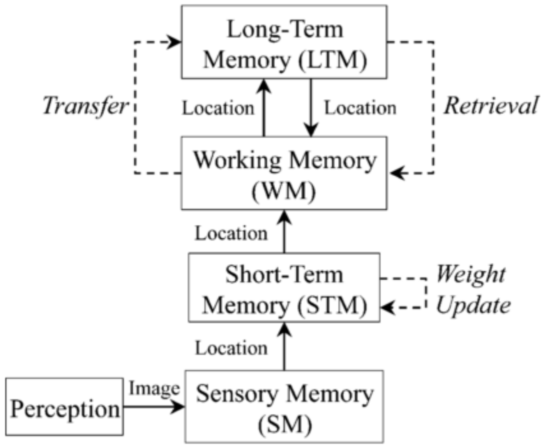

Fig. 6: RTAB-MAP memory management model

SM processes the image signature to decrease the input data dimension and uses the extracted features for loop-closure detection. Every new image signature stores in the new location in SM, before sends it to the short-term memory (STM). STM updates newly allocated locations through a process referred to as weight update. The weight update considers the similarity of the stored value between the newly allocated location and last one. If the comparison result shows the similarity, then it merges them into the new one and increases the weight of the new location. While STM observes the similarities through time between consecutive images for weight updates, the WM detects the loop closures between the scenes in 3D space. However, RTAB-MAP does not use locations in STM to prevent the loop closures on places that have just been visited due to the similarity between the last scene and the recent one in most of the time. LTM are not involved for loop-closure detection. Accordingly, it takes into account to choose the locations in LTM carefully. A simple approach utilizes first-in-first-out (FIFO) policy to prune the oldest locations from the map to deal with real-time constraints issue. Nevertheless, this sets a maximum sequence of places which can be stored while exploring an unknown environment. Thus, impossible to find a match if the processing time of pruning oldest location reaches the threshold time before the loop closures can be detected.

This study displays the visual of obtained 3D map by using Octomap. Octomap is an open-source implementation tool of the probabilistic 3D mapping technique proposed in [11]. This technique uses a tree-based representation (Octree) to provide maximum flexibility related to the mapped area and resolution as shown in Fig. 7. It performs a probabilistic occupancy estimation to guarantee repeatability and to deal with sensor noise. Furthermore, compression methods are applied to ensure the compactness of memory usage of resulting 3D map. An Octree technique is using a hierarchical data structure for spatial subdivision in 3D space where each node represents the occupied area in a cubic volume, commonly known as a voxel. Every single volume is recursively divided into eight sub-volumes until reaching the minimum voxel size. However, the minimum voxel size determines the resolution of the Octree. Sensor readings are performed using occupancy grid mapping technique [12]. Thus, the occupancy probability $P(n|z_{1:t})$ of a leaf node-n provided by the sensor measurements $z_{1:t}$ is estimated using,

$$P(n\mid z_{1:t})=\left[1+\frac{1-P(n\mid z_t)}{P(n\mid z_t)}\frac{1-P(n\mid z_{1:t-1})}{P(n\mid z_{1:t-1})}-\frac{P(n))}{1-P(n)}\right]^{-1} \quad (9)$$

The current measurement z_t, a prior probability $P(n)$, and the previous estimate $P(n|z_{1:t-1})$ determine the update of occupancy probability $P(n|z_{1:t})$. This value is subjected to the sensor that generated the value of z_t.

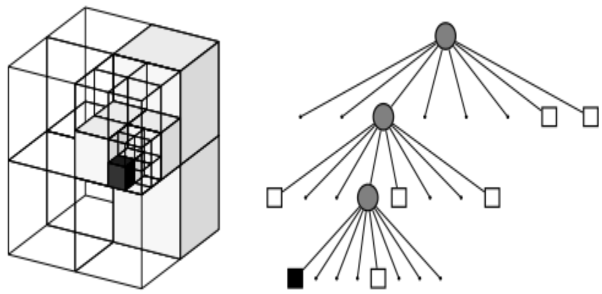

Fig. 7: Octree model: Volumetric Pixel (Left) and Corresponding octal tree (Right)

3. Results and discussions

Fig. 5 shows the design of ROS ecosystem which determines the success of the robot mission, by putting the right nodes in the right places and connecting them without conflict between them. In this work, Kinect RGB-D sensor is utilized and located on the top side of the robot as shown in Fig.1 in conjunction with 3D mapping function to provide information in the form of ROS message containing the RGB images, disparity images, camera information and transformation frame of the sensor. The odometry function is handled by RTAB-MAP internally by using visual odometry method which is relying on images features' correspondence based on SURF algorithm utilizing the Kinect sensor alone. Known map localization is not a part of the discussion in this work; since it is exploration and mapping mission only. Fig. 8 showed the mapping result and actual robot locations simultaneously using the RGB camera during inspection test. This information is used to associate a 3D inspection map with a 2D RGB image. It enables a remote operator to understand the situation around the robot either in 2D or 3D space. Based on the final map shown in Fig. 8, the Octree method has successfully visualized the area with a low density of measured points, thus decreased the memory consumption significantly during real-time mapping in an unknown area. The voxel ratio of the map depends on the number of octree level as depicted by Eqn. (9) and Fig. 7.

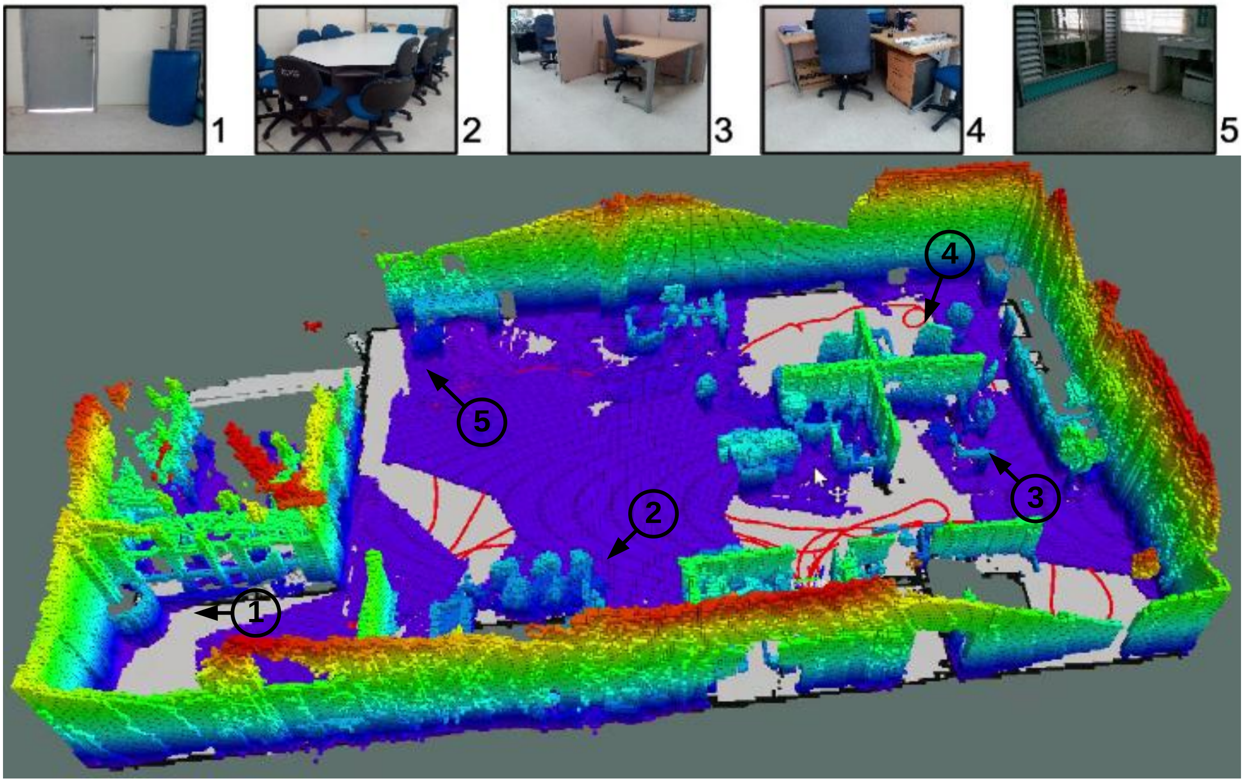

Fig. 8: Visualization of 3D occupied map used for inspection mission

4. Conclusions

In this work, the odometry function becomes a big issue in term of accuracy and precision. Visual odometry which comes up with uncertainty error depending up on image features detection and tracking is relying on how SURF algorithm deals with the texture-less object and the acceleration of the camera movement about the object. Texture-less area and high acceleration of camera movement produce a significant error in odometry function. The translation and rotational speed of the robot are the challenges to perform the detail inspection resulting in low drift 3D map. This work has successfully demonstrated the methods on how to apply mobile robot based on ROS platform to do the exploration and generate the 3D map which can be implemented in the remote inspection of the industrial plant or unknown environment.

ACKNOWLEDGEMENTS:

This project is supported by Universiti Malaysia Pahang under Research Grant Scheme (RDU140358).

REFERENCES:

[1] C. Chen and C. Yinhang. 2008. Research on map building by mobile robots, *Proc. 2nd Int. Symp. on Intelligent Information Tech. Application*, 2, 673-677. https://doi.org/ 10.1109/iita.2008.205.

[2] D.F. Wolf and G.S. Sukhatme. 2008. Semantic mapping using mobile robots, *IEEE Trans. Robot*, 24(2), 245-258. https://doi.org/10.1109/TRO.2008.917001.

[3] R.B. Rusu. 2010. Semantic 3D object maps for everyday manipulation in human living environments, *Kunstl. Intelligenz*, 24, 345-348.

[4] R. Zlot and M. Bosse. 2014. Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine, *Springer Trans. Adv. Robot*, 92(8), 479-494. https://doi.org/10.1007/978-3-642-40686-7_32.

[5] B.B. Cortes, X.C. Sole and J. Salvi. 2011. Indoor SLAM using a range-augmented omnidirectional vision, *Proc. IEEE Int. Conf. Computer Vision*, 280-287.

[6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler and A. Mg. 2009. ROS: An open-source robot operating system, *IEEE Int. Conf. Robotics and Automation*.

[7] S. Cousins. 2010. ROS on the PR2, *IEEE Robot. Autom. Mag.*, 17(9), 23-25. https://doi.org/10.1109/MRA.2010. 938502.

[8] M. Labbe and F. Michaud. 2014. Online global loop closure detection for large-scale multi-session graph-based SLAM, *IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, 2661-2666. https://doi.org/10.1109/iros.2014. 6942926.

[9] M. Labbe and F. Michaud. 2013. Appearance-based loop closure detection for online large-scale and long-term operation, *IEEE Trans. Robot.*, 29(3), 734-745. https://doi.org/10.1109/TRO.2013.2242375.

[10] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff and W. Burgard. 2007. Efficient estimation of accurate maximum likelihood maps in 3D, *IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, 3472-3478. https://doi.org/ 10.1109/iros.2007.4399030.

[11] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss and W. Burgard. 2013. OctoMap: An efficient probabilistic 3D mapping framework based on octrees, *Auto. Robot.*, 34(3), 189-206. https://doi.org/10.1007/ s10514-012-9321-0.

[12] H. Moravec and A. Elfes. 1985. High resolution map from wide angle sonar, *IEEE Int. Conf. Robot. Autom.*, 2, 116-121. https://doi.org/10.1109/robot.1985.1087316.