

Environment detection system for localization and mapping purposes

Neduchal P. * Bureš L. * Železný M. *

* Faculty of Applied Sciences, New Technologies for the Information Society

University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic, {neduchal, lbures, zelezny}@kky.zcu.cz

Abstract: The goal of this paper is to present a concept and implementation of an Environment Detection System. The system is supposed to collect data from sensors attached to the mobile robot and then enhance the map of the environment by this information. Moreover, the data can be processed to get valuable information about the environment or its change. The open-source implementation of the system is written for Robot Operating System. Thus, the system can handle data from different sensors using a unified way. It is possible by employing the messaging mechanism implemented in the Robot Operating System. Another contribution of this paper are records of our testing runs with a 6WD mobile robot equipped by multiple sensors, which can be used as a dataset for SLAM and Environment Detection System implementations.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Mobile Robots, Multisensor Integration, Simultaneous Localization And Mapping, Robotics, Sensors, Cameras, Encoders, Embedded systems

1. INTRODUCTION

Autonomous mobile robotics is a rapidly growing research field. The significant part of this field is represented by solving of localization and mapping problem. Nowadays, algorithms can create sufficiently accurate maps even in real-time – i.e. Simultaneous Localization And Mapping (SLAM) Durrant-Whyte and Bailey (2006). On the other hand, the map contains only information about the geometry of the environment.

In this paper, we present a concept and the implementation of the Environment Detection System (EDS). The system is supposed to collect additional data about the environment during the movement of the mobile robot. This additional data can be utilized for multiple tasks in mobile robotics. Our main goal is to propose a system which can be used for detection and classification of the environment properties – e.g. whether the robot is indoor or outdoor. Knowledge of these properties will be used in our future work for improving our multi-environment localization and mapping systems.

A camera usually acquires the additional information. However, there are more types of relevant data – e.g. temperature, barometric pressure, ultraviolet light, humidity or ceiling height – that can be observed by the robot.

All data are stored together with a map created by SLAM software. It means that the individual values are stored on the positions in the map where the sensors collected them. It can be used in many situations – e.g. for application where the robot is looking for local abnormalities in the environment such as radiation.

The contribution of the paper is the presentation of the proposed concept and its open-source implementation.

The third contribution is a recorded dataset suitable for testing environment detection approaches.

The paper is composed as follows. In Section 2, the related work to this particular paper is introduced. The Section 3 contains information of the proposed concept. In Section 4, the implementation of the system is briefly described. The implementation tests are discussed in Section 5. Finally, the paper is summarized in Section 6.

2. RELATED WORK

The fast-growing research field of localization and mapping (see Grisetti et al. (2007), Kohlbrecher et al. (2013), Hess et al. (2016), Cadena et al. (2016)) is focusing on multiple areas of research which aim to solve different problems. One of a common task is to create a map of an environment, then multiple approaches and problem-solving techniques can be applied.

One example can be the problem of place categorization and semantic mapping on a robot without environment-specific training Sünderhauf et al. (2015).

Next, a robust data association is a core problem of visual odometry, where image-to-image correspondences provide constraints for camera pose and map estimation. The paper Lianos et al. (2018) introduces visual semantic odometry framework to enable medium-term continuous tracking.

Semantic segmentation and semantic SLAM is a very important topic nowadays. It can be used for solving multiple tasks: the closing-loop semantic segmentation and monocular depth estimation tasks Zhang et al. (2018), combining monocular visual Simultaneous Localization And Mapping (vSLAM) and deep-learning-based semantic

segmentation Kaneko et al. (2018), generating meaningful maps for object-oriented semantic mapping Sünderhauf et al. (2016), and robust visual localization under a wide range of viewing conditions based on a joint 3D geometric and semantic understanding of the world Schnberger et al. (2018).

Semantic Localization is a popular strategy for semantic localization which focus on features found on informative structures (see Kobyshev et al. (2014), Mousavian et al. (2015)) and to re-weight or discard ambiguous features Knopp et al. (2010). Individual features Kobyshev et al. (2014) or Bag-of- Words representations Arandjelovic and Zisserman (2014), Singh and Koeck (2016) can be enhanced. An alternative strategy to semantic localization is to use high-level features such as lane markings Schreiber et al. (2013), object detections (see Ardeshir et al. (2014), Atanasov et al. (2016), Salas-Moreno et al. (2013), Toft et al. (2017)), discriminative buildings structures Wolff et al. (2016), or the camera trajectory of a car Brubaker et al. (2016). These approaches need object databases or maps containing the same types of objects, which either requires careful manual annotation Schreiber et al. (2013) or prescanning of objects Salas-Moreno et al. (2013). In addition, the feature extraction and matching process is often a complex and hand-crafted solution tailored to specific objects Ardeshir et al. (2014), Atanasov et al. (2016).

In the SLAM research field many of approaches exist. A few of recent approaches can be briefly mentioned: fusion++: volumetric object-level SLAM McCormac et al. (2018), CodeSLAM learning a compact - optimisable representation for dense visual SLAM Bloesch et al. (2018), and QuadricSLAM Nicholson et al. (2018).

3. SYSTEM OVERVIEW

The proposed system is composed of three parts. Data storage module/s, data acquisition module, data processing module. All parts are shown in Fig. 1. The whole system has inputs from sensors, mapping system and parameters of the processing module. The output of the system should be a multilayered map suitable for further processing and visualization.

3.1 Data Acquisition Module

The module/s is supposed to record data from various sensors to the EDS. Data acquisition module has three parts. The first one is communication with an attached sensor and reading sensor data. The second one is data preprocessing – when necessary. The last part is communication with the data storage module.

3.2 Data Storage Module

This module is the core of the EDS system. It stores all information from the sensors in the multilayered grid map structure. Therefore, it is necessary to provide a map – partial or full – of the environment and a current position of the mobile robot. Data from sensors are then saved to individual layers. Moreover, the position of the robot is used to determine the place where the data was recorded.

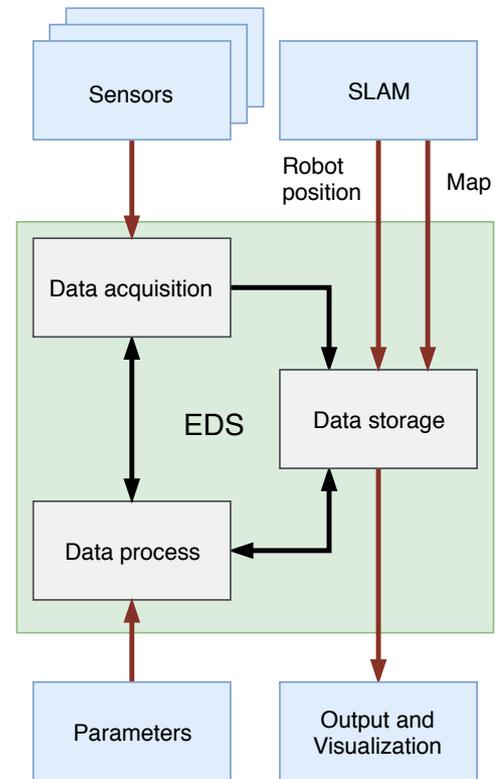


Fig. 1. Structure of EDS system concept.

3.3 Data Processing Module

This module is supposed to process data both, directly from the data acquisition module and the data storage module. The goal is to process acquired data w.r.t. Sensor type, properties and physics. Each sensor usually has different properties – e.g. the range of sensing, inertia or range of values.

4. IMPLEMENTATION OF THE SYSTEM

The concept of the system is implemented in C++ and Python. It is developed as an open-source software based on the Robot Operating System (ROS) Quigley et al. (2009) and it is available online on GitHub¹.

4.1 Robot Operating System

ROS is a framework suitable for developing modular robotic systems. Systems are usually composed of nodes. Node is an atomic software part that can run inside the ROS environment. Multiple nodes communicate through ROS message mechanism – shown in Fig. 2. Messages are published by node into so-called Topic – i.e. named gateway for the particular message type. Another node can then receive the message by subscribing the same Topic. This mechanism also works over the network between multiple computers. Thus, it is possible to monitor and control the system using a remote computer.

Another power of the framework is based on the applications and algorithms that are already implemented. It comprises drivers for various sensors, visualization tools or SLAM algorithms.

¹ <https://github.com/neduchal/env.detection>

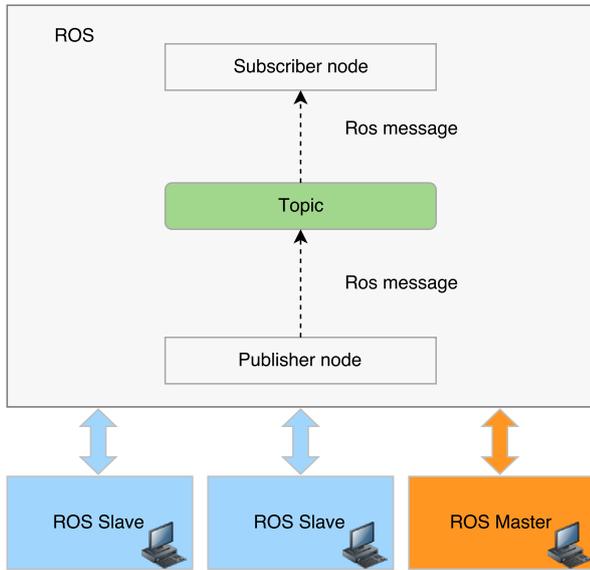


Fig. 2. Structure of ROS (Neduchal and Flidr (2016))

4.2 Data Storage Node

This storage module is built on the top of GridMap library Fankhauser and Hutter (2016). It is an open-source package for ROS available on GitHub². Thus, GridMap package is the core of the module. GridMap itself is a data structure which added multiple layers on the classic grid map representation. Each layer has its name. Thus, it is possible to store additional information in individual layers.

The overall structure of the node is shown in Fig. 3. It contains three main inputs, one main output, one optional input and two optimal outputs. Inputs called `map` and `pose` has to be provided by SLAM and odometry/pose estimation nodes respectively. Input `sensor msg` is the main input from the data acquisition nodes. Sensor message consists of two parts – the name of the layer and the value from the sensor. The value is stored in the position of the robot in the defined layer. The output of the module is the GridMap msg – i.e. message of the whole GridMap structure.

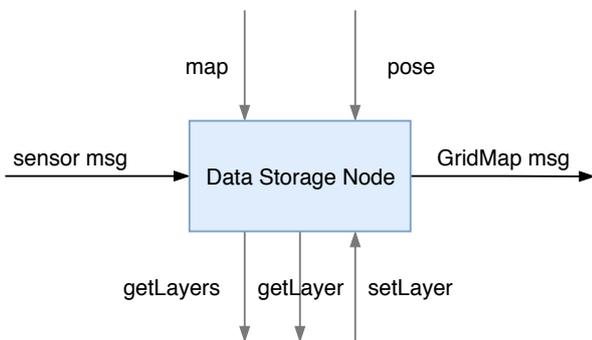


Fig. 3. Data storage node structure.

The data in the node can be affected by the optional inputs and outputs. By the `getLayers` service – i.e. function available between two nodes in ROS – which provides the

² https://github.com/ANYbotics/grid_map

list of all names of all existing layers can be acquired. On the other hand, the service `getLayer` is supposed to provide data of a particular layer in the GridMap structure to data processing nodes. Same nodes can return edited data to the data storage node by the `setLayer` service.

4.3 Data Acquisition Node

Data acquisition node is supposed to communicate – i.e. read data and control properties – with sensors and send data to the data storage node. The structure of the node is simple. The input of the node is data values from equipped sensors. The output is the sensor message which goes directly into the data storage node. Based on the layer names, it is possible to connect more data acquisition nodes to a single data storage node. The structure of the node is shown in Fig. 4.



Fig. 4. Data acquisition node structure.

4.4 Data Processing Node

Data processing module reads data from both types of previous nodes. It can process data based on the task or the properties of the data. Thus, it can be a complicated node with multiple inputs and outputs. The fundamental structure of this type of node – communicating with the data storage node – is shown in Fig. 5. It is composed of services which were described in the data storage node subsection. Of course, inputs and outputs are opposite in the case of this node. Moreover, multiple data processing nodes can be used in one system.

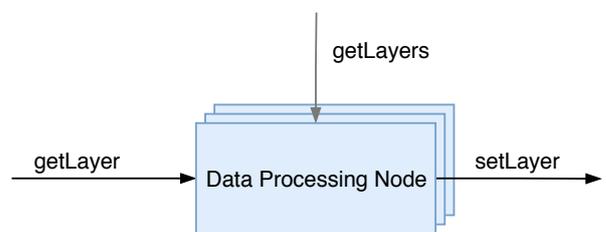


Fig. 5. Data processing node scheme.

4.5 Summary

In detail, the EDS ROS package is composed of three parts. The first one is the core which contains data storage node. The second one is the message part which contains definitions of all messages and services used in the system. The last part is the sensors part, which contains an example of data acquisition and data processing nodes – the part used during the testing runs.

5. SYSTEM TESTING

In this section, the results of testing runs of the system are shown and described. The goal of testing was to verify the

right functionality of the implementation of the proposed system concept. For system testing, the following mobile robot configuration was used.

5.1 Robot Configuration

The robot used for system testing is shown in Fig. 6. It is a 6WD chassis called Wild Thumper equipped with onboard computer NVidia Jetson and multiple sensors. In the testing, the following sensors were used. LiDAR sensor for SLAM, temperature sensor, ultrasound distance sensor for measuring of the ceiling height, humidity sensor and rotary encoders on the middle wheels for odometry information. The temperature, humidity and distance sensors are controlled using Arduino UNO 3 embedded board. It is connected to the onboard computer using USB and communication is held by the ROS message mechanism. Another embedded board called T-Rex is used for the control of the robot wheels and for diagnostic proposes. The robot is based on Ubuntu Linux 16.04 with ROS Kinetic version.



Fig. 6. Testing robot configuration

5.2 Recorded Dataset

Data was recorded using RosBag package available in ROS. Recorded rosbag file contains all acquired data and information about the time of recording. Thus, it can be rerun with different parameters to test various data acquisition and data processing nodes. The recorded dataset contains one short run – the office – and various long – indoor – runs in our building. The examples of recorded places are shown in Fig. 7

Data were processed on the computer with different configuration then a robot – Ubuntu 18.04 with ROS Melodic. The gMapping system held the creation of the map. The odometry was computed from the raw values acquired by rotary encoders on the robot wheels. In the next sections, the results of the system are shown.

5.3 Short Run Results

The EDS system was firstly tested in the short run. The robot moves from the center of the office to the corridor. The length of recorded data is 16 seconds and contains data from all equipped sensors. The results are shown in



Fig. 7. Recorded part of the building

figures below using a visualization plugin for RVIZ – ROS visualization software – available in GridMap package.

In Fig. 8, there is a visualization of a temperature sensor data. It is visible that values of the temperature are growing – color changes from magenta to green, yellow and red – during the run. Range of colors is relative – computed based on the range in the layer data. Thus, the magenta color values 26.6 and the red value is 26.8 degree of Celsius.

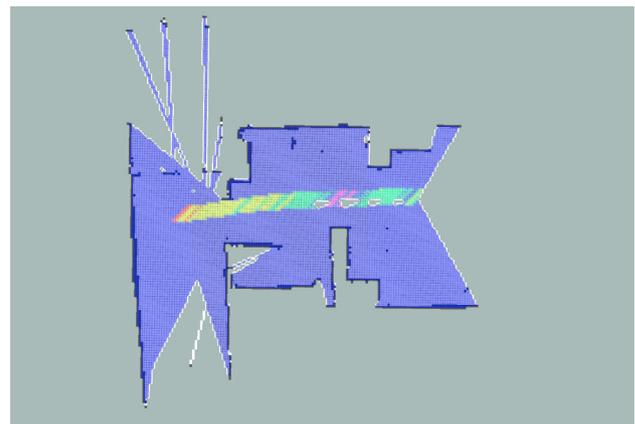


Fig. 8. Short run temperature example

The second example of the system result during the short run is shown in Fig. 9. It contains data of ultrasound distance sensors measuring ceiling height. When the robot moves through the office, the ceiling height remains almost identical (235 – 240 cm). Then the robot goes through the doors to the corridor, and the distance value goes lower to 160 cm for a short time. In the corridor, the distance value grows up to the new stable value of 220 cm.

5.4 Long Run Results

Similarly, we tested the system in the long run. The robot goes around the part of the one floor in the building. The data, in this case, has a duration of 172 seconds and creates a significantly bigger map than the previous case.

In Fig. 10, there is a visualization of temperature data during the long run. There is visible one crucial property.

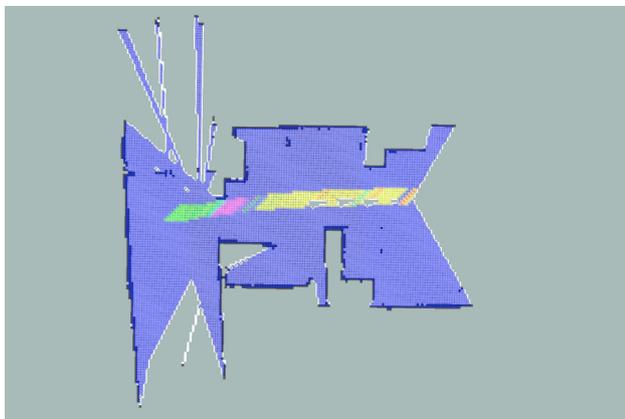


Fig. 9. Short run ceiling height example

The temperature sensor has a significant delay in the recorded data. The value of the sensor changed slowly in contrast to the movement of the robot. Thus, the same part of the corridor has two different values of the temperature. On the other hand, it can be determined where the temperature is higher/lower in the building.



Fig. 10. Long run temperature example

The second example is again ceiling height recording in Fig. 11. During the long run, the visualization is not as clear as in the case of the short run. It is caused by using the cheap and less precise sensor. It sometimes measures bad value which changes the colour range of the visualized data. On the other hand, there are still visible places with doors – Especially the office doors from the short run. They are at the top part of the map, where the office contour is visible.

The last example of the system results is shown in Fig. 12. It contains recorded humidity data during the long run. The result is slightly similar to the temperature data in Fig. 10.

6. CONCLUSION

The concept and the open-source implementation of the environment detection system were presented in this paper. The proposed system can acquire and store sensor data in the map created by SLAM solving algorithms. In Section 4, details of the implementation was described. The implementation was successfully tested and verified on the recorded dataset.

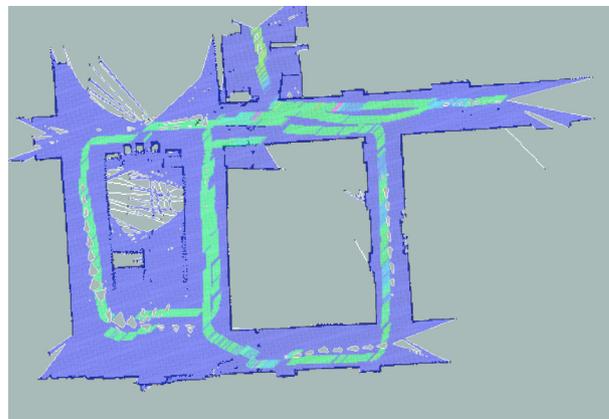


Fig. 11. Long run ceiling height data example

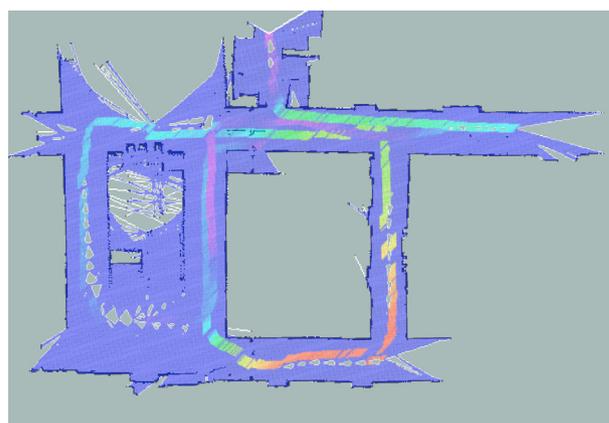


Fig. 12. Long run humidity example

In future work, we plan to use the system for environment detection as new information for SLAM algorithms. Thus, we want to recognize the properties of the environment and cause a reaction of the mobile robot – i.e. change of the parameters, behaviour or whole SLAM algorithm. Moreover, we plan to extend the proposed system by its visualization for RVIZ software.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education of the Czech Republic, project No. LTARF18017 and Ministry of Education, Youth and Sports of the Czech Republic project No. LO1506 and The work has been supported by the grant of the University of West Bohemia, project No. SGS-2019-027.

REFERENCES

- Relja Arandjelovic and Andrew Zisserman. Visual vocabulary with a semantic twist. In *ACCV*, 2014.
- Shervin Ardeshtir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah. Gis-assisted object detection and geospatial localization. In *ECCV*, 2014.
- Nikolay Atanasov, Menglong Zhu, Kostas Daniilidis, and George J. Pappas. Localization from semantic observations via the matrix permanent. *The International Journal of Robotics Research*, 35(1-3):73–99, 2016. doi: 10.1177/0278364915596589.

- Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam learning a compact, optimisable representation for dense visual slam. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- M. A. Brubaker, A. Geiger, and R. Urtasun. Map-based probabilistic visual self-localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):652–665, April 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2453975.
- Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian D Reid, and John J Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *arXiv preprint arXiv:1606.05830*, 2016.
- H Durrant-Whyte and T Bailey. Simultaneous localization and mapping (slam): Part 2. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.
- Péter Fankhauser and Marco Hutter. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. In Anis Koubaa, editor, *Robot Operating System (ROS) The Complete Reference (Volume 1)*, chapter 5. Springer, 2016. ISBN 978-3-319-26052-5.
- Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.
- Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1271–1278. IEEE, 2016.
- Masaya Kaneko, Kazuya Iwami, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
- Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In *Proceedings of the 11th European Conference on Computer Vision: Part I, ECCV'10*, pages 748–761, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15548-0, 978-3-642-15548-2.
- N. Kobyshev, H. Riemenschneider, and L. V. Gool. Matching features correctly through semantic understanding. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 472–479, Dec 2014. doi: 10.1109/3DV.2014.15.
- Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen, Uwe Klingauf, and Oskar von Stryk. Hector open source modules for autonomous mapping and navigation with rescue robots. In *Robot Soccer World Cup*, pages 624–631. Springer, 2013.
- Konstantinos-Nektarios Lianos, Johannes L. Schonberger, Marc Pollefeys, and Torsten Sattler. Vso: Visual semantic odometry. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. *CoRR*, abs/1808.08378, 2018.
- A. Mousavian, J. Koeck, and Jyh-Ming Lien. Semantically guided location recognition for outdoors scenes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4882–4889, May 2015. doi: 10.1109/ICRA.2015.7139877.
- Petr Neduchal and Miroslav Flídr. Development of a laboratory framework for testing simultaneous localization and mapping approaches. *IFAC-PapersOnLine*, 49(25): 493–498, 2016.
- Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Constrained dual quadrics from object detections as landmarks in semantic SLAM. *CoRR*, abs/1804.04011, 2018.
- Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.
- R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, June 2013. doi: 10.1109/CVPR.2013.178.
- Renato Salas-Moreno, Richard Newcombe, Hauke Strasdat, Paul Kelly, and Andrew Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.
- M. Schreiber, C. Knoppel, and U. Franke. Laneloc: Lane marking based localization using highly accurate maps. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 449–454, June 2013. doi: 10.1109/IVS.2013.6629509.
- Johannes L. Schnberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- Gautam Singh and Jana Koeck. *Semantically Guided Geolocation and Modeling in Urban Environments*, pages 101–120. I. J. Robotics, 07 2016. ISBN 978-3-319-25779-2. doi: 10.1007/978-3-319-25781-5_6.
- Niko Sünderhauf, Feras Dayoub, Sean McMahan, Ben Talbot, Ruth Schulz, Peter I. Corke, Gordon Wyeth, Ben Uproft, and Michael Milford. Place categorization and semantic mapping on a mobile robot. *CoRR*, abs/1507.02428, 2015.
- Niko Sünderhauf, Trung-Thanh Pham, Yasir Latif, Michael Milford, and Ian D. Reid. Meaningful maps - object-oriented semantic mapping. *CoRR*, abs/1609.07849, 2016.
- C. Toft, C. Olsson, and F. Kahl. Long-term 3d localization and pose from semantic labellings. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 650–659, Oct 2017. doi: 10.1109/ICCVW.2017.83.
- Maximilian Wolff, Raymond J. Collins, and Yanxi Liu. Regularity-driven facade matching between aerial and street views. In *CVPR 2016*, 2016.
- Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.