COBEM UBERLÂNDIA 2019

ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-0845
MAPPING IN DYNAMIC ENVIRONMENTS USING DEEP LEARNING-BASED HUMAN DETECTION

João Carlos Virgolino Soares
Pontifícia Universidade Católica do Rio de Janeiro, Department of Mechanical Engineering
R. Marquês de São Vicente 225, Gávea, Rio de Janeiro, RJ - Brazil - 22451-900
virgolinosoares@gmail.com

Marcelo Gattass
Pontifícia Universidade Católica do Rio de Janeiro, Department of Informatics
R. Marquês de São Vicente 225, Gávea, Rio de Janeiro, RJ - Brazil - 22451-900
mgattass@tecgraf.puc-rio.br

Marco Antonio Meggiolaro
Pontifícia Universidade Católica do Rio de Janeiro, Department of Mechanical Engineering
R. Marquês de São Vicente 225, Gávea, Rio de Janeiro, RJ - Brazil - 22451-900
meggi@puc-rio.br

Abstract. *Simultaneous Localization and Mapping (SLAM) is one of the key problems in mobile robotics. However, most of the state-of-the-art SLAM systems only work properly in static environments, limiting their applicability. Performing SLAM in scenarios with dynamic objects is still an open problem, with solutions usually relying on Optical Flow or feature-based methods. Deep learning techniques offer a high-level information of the scene that can improve the solution in environments with a priori dynamic objects, such as people. This work proposes an extension of RTAB-Map, a state-of-the-art SLAM system, to enable mapping in the presence of people in the scene using a Deep Learning-based Object Detection algorithm. The system is evaluated with an RGB-D dataset and an experimental setup.*

Keywords: *SLAM, Dynamic Environments, Deep Learning, Object Detection, ROS*

1. INTRODUCTION

The Simultaneous Localization and Mapping (SLAM) problem is a fundamental capability of autonomous mobile robots that operate in unknown and unstructured environments. It consists in creating a map of the environment and, at the same time, estimating the pose of the robot in this map. Several SLAM systems use monocular, stereo or RGB-D cameras as their primary sensor, due do their low cost and richness of information, allowing a robust place recognition. There are high quality Visual SLAM systems in the literature, with both precision and efficiency. However, their majority, for instance ORBSLAM2 (Mur-Artal and Tardós, 2017), LSD-SLAM (Engel *et al.*, 2014) and other keyframe-based approaches (Soares and Meggiolaro, 2018), assume a static environment, which imposes a limitation of its applicability.

The ability to perform Visual SLAM in a dynamic scenario is still an open problem in robotics. The approaches usually consist in filtering moving objects from the frame, treating them as noise, using feature-based techniques (Tan *et al.*, 2013) or direct methods (Alcantarilla *et al.*, 2012). However, these approaches do not recognize when an *a priori* dynamic object, as a person, is in the scene. If a person remains static for a period of time and then starts moving, these approaches would be unable to classify the person as a dynamic object in the initial frames.

Deep learning techniques, in other hand, can be a solution to this problem, offering high-level information about the environment. Several new approaches are being proposed using these techniques. For instance, Bescós *et al.* (2018) created DynaSLAM using the instance segmentation method, proposed by He *et al.* (2017), to obtain the pixel-wise information of different classes detected in the images, and to filter the keypoints contained in the *a priori* dynamic regions. Despite being accurate, instance segmentation is still computationally slow. Object detection also offers high-level information, but with an increased speed.

There are three main challenges in performing dynamic Visual SLAM: detecting dynamic objects in the scene, excluding the dynamic objects from the map and filtering the information given to the tracking system. This work proposes a solution to those three problems using a deep learning-based object detection technique combined with an activation filter, considering only *a priori* dynamic objects.

RTAB-Map (Real-Time Appearance-Based Mapping) (Labbé and Michaud, 2018) is a SLAM system with a robust place recognition methodology and a efficient memory management system. However, it does not work properly when there are moving objects in the scene. Fig. 1 shows the point cloud map generated by the RTAB-Map system using data from a sequence of an RGB-D dataset. This sequence is used to evaluate the capability of the SLAM system to deal with dynamic environments. The map is corrupted by the movement of people.

This paper proposes an extension of the RTAB-Map system to allow its use in dynamic scenarios, using a deep learning-based object detection system to filter the frames containing people, which are *a priori* dynamic objects. The TUM RGB-D SLAM dataset (Sturm *et al.*, 2012) is used to evaluate the proposed system and a mobile robot is used to perform experiments. The system is built as a Robot Operating System (ROS) package (Quingley, 2009).

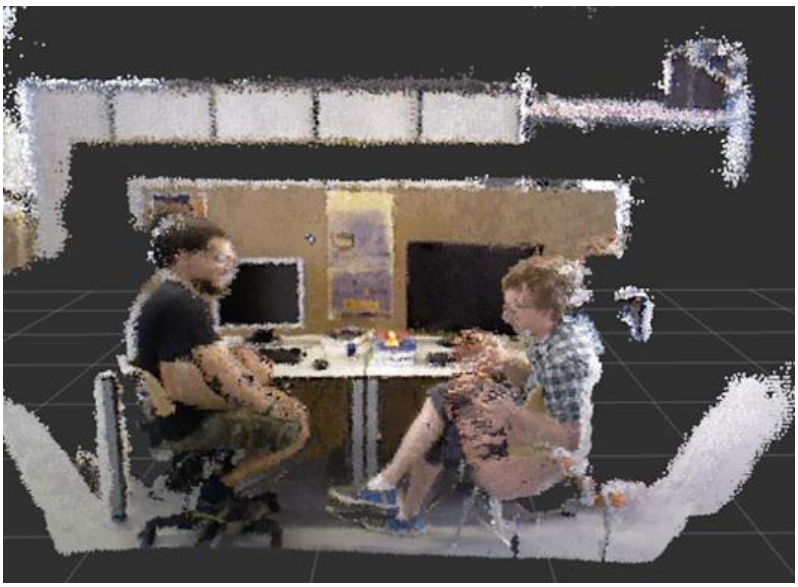

Figure 1. Point cloud Map of a dynamic scene using RTAB-Map

This paper is organized as follows. Section 2 explains the methodology used to solve the problem. Section 3 presents the evaluation of the system using an RGB-D dataset. Section 4 presents the robot used for indoor experiments and the results obtained. Section 5 presents the conclusions and suggestions for future work.

2. METHODOLOGY

The proposed methodology is composed of three main parts: SLAM, object detection and an activation filter, which are detailed in the following sections. The SLAM system receives sensor data and gives the point cloud map of the environment, along with the optimized trajectory of the robot. The object detection system and the filter are responsible to restrict the data sent to the SLAM framework. Fig. 2 shows the block diagram of the proposed methodology.

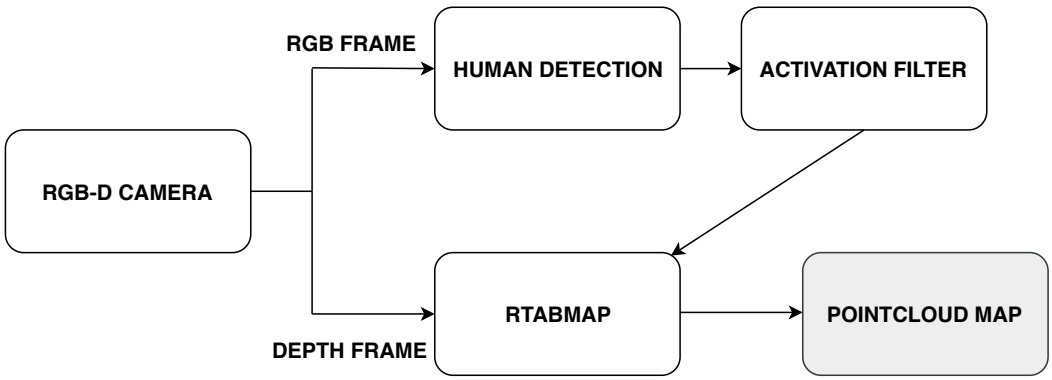

Figure 2. Diagram of the proposed methodology

2.1 SLAM

The current standard formulation of SLAM is based on graphs (Carlone *et al.*, 2016), where the poses of the robot, given by the odometry system, are represented by the nodes of the graph, and the edges are constraints between two observations, created by a place recognition system. When these constraints are created, the graph is optimized to correct the odometry drift.

The standard SLAM problem is divided into two main parts: front-end and back-end. The front-end is responsible for creating the graph, using sensor information to estimate the pose of the robot, and a place recognition system to create constraints between two poses. The back-end problem is responsible for the optimization of the graph, reducing the constraint errors, and is translated in Eqs. (1) and (2). The objective is to obtain the maximum belief over the states x given the measurements z.

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, p(x|z) \tag{1}$$

$$\hat{x} = \underset{x}{\operatorname{argmin}} \sum_{ij} e_{ij}^T \Omega_{ij} e_{ij} \tag{2}$$

where Ω is the matrix that represents the uncertainty of each measurement, and e_{ij} is the difference between the estimated state and the measured state of the robot. The back-end optimization is essential to assure a solution with minimal error. However, even with a high-quality back-end system, if the graph created by the front-end is corrupted, has mismatches or contain spurious information, the solution would be incorrect. Thus, it is highly important to filter the information given to the back-end, specially in scenarios with high levels of noise, such as dynamic environments.

RTAB-Map (Labbé and Michaud, 2018) is a Graph-based SLAM approach that can work with RGB-D or Stereo Cameras, using RGB images, depth images and odometry information to generate a point cloud map of the environment. This work uses RTAB-Map as the global SLAM solution and adds a pre-processing step to its front-end, preventing the addition of spurious nodes to the graph.

2.2 Object Detection

Object detection is the task of determining the location of objects in an image and also the class of each detected object. There are different types of deep learning-based object detection algorithms. They all provide the classes of the detected objects, the 2D bounding boxes with their corresponding positions and a confidence number that states the probability associated with each detection.

The R-CNN detectors use convolutional neural networks (CNN) to extract features from images. The first R-CNN based approach (Girshick *et al.*, 2014) was too slow and was not an end-to-end deep learning object detector. The Fast R-CNN algorithm (Girshick, 2015) improved the accuracy and speed of its predecessor, and a new version was proposed in 2015, the Faster R-CNN (Ren *et al.*, 2015), introducing a technique called Region Proposed Network.

Despite being accurate, these algorithms are not able to work real-time due to its complex pipelines. The YOLO (You Only Look Once) technique from Redmon *et al.* (2016), in other hand, can run at 45 frames per second using a GPU. The previous algorithms do not check the full image. Instead, they apply their model to multiple locations. YOLO works as a single regression problem, using a single convolutional network to predict the bounding boxes and their class probabilities. Fig. 3 shows the YOLO model. The input image is divided in a grid, in which each cell predicts one set of class probabilities and a number of bounding boxes. Each bounding box consists in 5 predictions: the x and y coordinates of the center of each box relative to the border of the cell, width, height and the confidence.

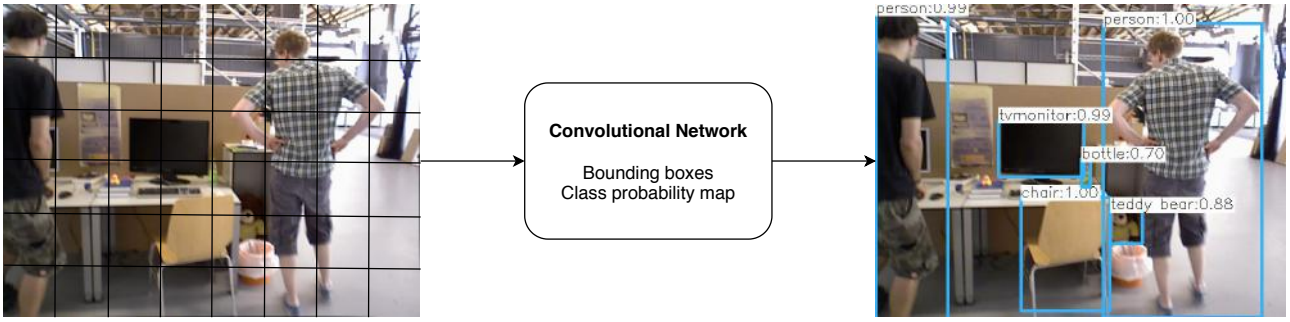

Figure 3. YOLO Model

This work uses the OpenCV implementation of YOLO, trained with the COCO dataset (Lin *et al.*, 2014). Fig. 4 shows an image with detected objects. It shows different categories being classified, such as people, and other static objects, with their respective bounding boxes and confidence numbers.

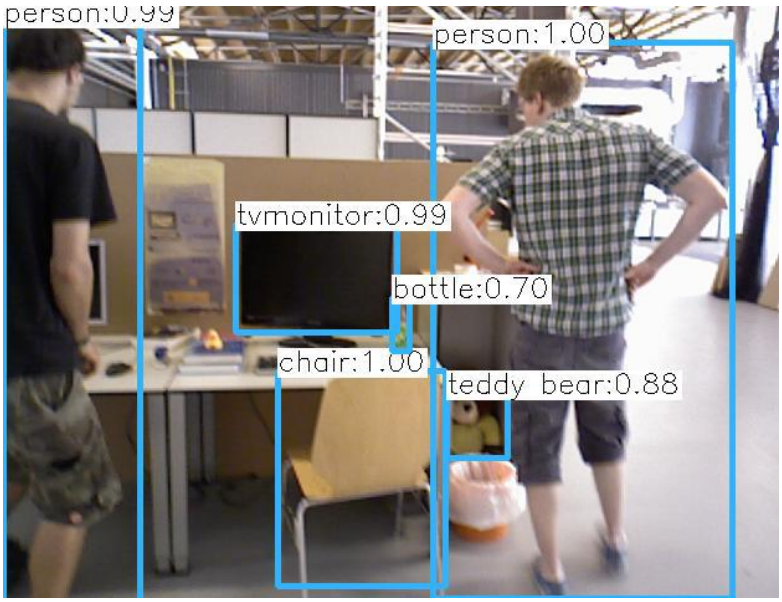

Figure 4. Final Object Detection

2.3 Filter

In the RGB-D mode, the RTAB-Map framework is only activated when receives both depth and RGB images. Instead of sending both images to the RTAB-Map, the RGB images pass through the object detector and a filter. If there are no people detected in the frame, then it is sent to RTAB-Map. In other case, the system awaits for the next frame. With this methodology, no people is added to the final map.

2.4 ROS

ROS is a framework for robotics software development, built as a large group of small programs called nodes. The nodes communicate one to another through messages, carried by topics, that can be sensor input, debug messages or control output. The proposed methodology is built as a ROS package and ROS works as middleware of the system. The filter with the object detector is built as a ROS node that receives RGB images as input and send them to the SLAM node, as shown in Fig. 5.

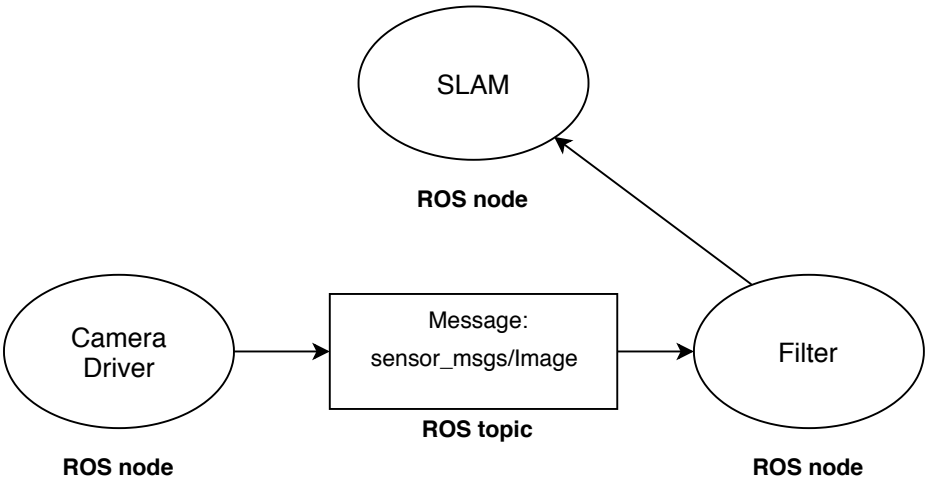

Figure 5. ROS architecture

3. DATASET EVALUATION

The RGB-D dataset from the Technical University of Munich (Sturm *et al.*, 2012) was used to evaluate the system. It provides several sequences of color and depth images obtained from a Kinect sensor. The sequence "fr3_walking_xyz" was chosen for the evaluation, with 28 seconds of duration and $5.791m$ of trajectory length. This sequence is suitable to evaluate the robustness of SLAM algorithms to people moving quickly in the scene. The camera is moved along three directions whilst two persons are moving around. Fig. 6 shows the final map using only the RTAB-Map system. Fig. 7 shows the final map using the proposed methodology.

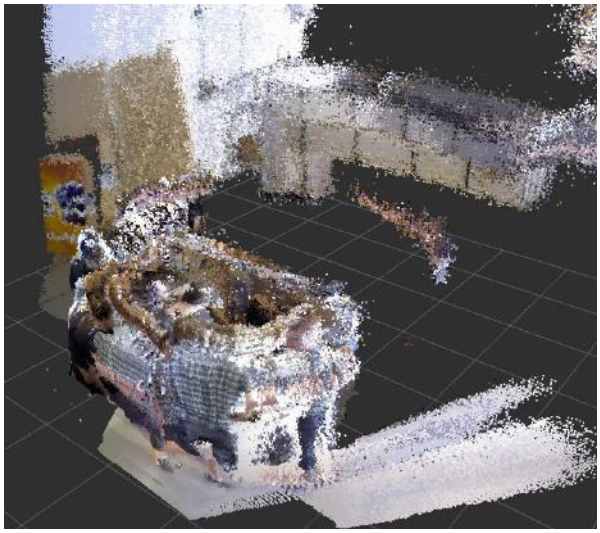

Figure 6. Mapping without object detection filter

Figure 7. Mapping with object detection filter

4. EXPERIMENTAL EVALUATION

Indoor experiments were conducted in the Robotics Laboratory from the Pontifical Catholic University of Rio de Janeiro. Fig. 8 shows the mobile robot used in the experiments, an iRobot Create, equipped with a ZED Stereo Camera and a Jetson TX2 board. The iRobot Create is a commercial differential drive platform equipped with an encoder in each wheel. The ZED Camera provides visual odometry and depth information, besides the stereo images.

Figure 8. Experimental Setup

Besides the object detector/filter developed and RTAB-Map, the following third-party ROS nodes were used in this work:

- Rviz: A visualization tool used to show the point cloud map
- ZED-ros-wrapper: The camera driver
- create_autonomy: iRobot driver

The ZED driver provides the camera RGB and depth images, sent to the object detector and filter. The robot driver receives velocity commands that are sent to the wheels and sends wheel odometry readings to RTAB-Map, which can also be used with visual odometry.

In the experiments, the robot performed SLAM with a person walking in the scene. Figs. 9 and 10 show RGB images received by the ZED camera with the object detection system. A person is detected in Fig. 10. Fig. 11 shows the point cloud map without the filter. The person corrupted the final map. Fig. 12 shows that the person was filtered from the map using the proposed approach.

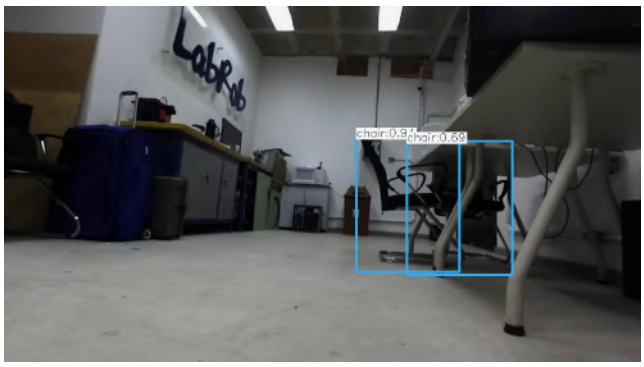

Figure 9. Object Detection

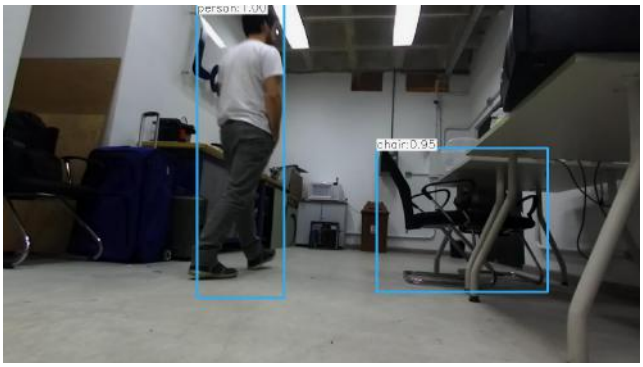

Figure 10. Person detected

Figure 11. Mapping without object detection filter

Figure 12. Mapping with object detection filter

The experiments show that the proposed system was able to prevent people from corrupting the point cloud map. One drawback of this approach is not being able to map the environment while people is in front of the camera. However, this can be an advantage if used in an active SLAM system, in terms of safety.

5. CONCLUSION

This work presented an extension to the RTAB-Map system to perform SLAM in environments with people. Dataset tests showed that proposed approach is successful. The system also worked in a mobile robot performing SLAM in a indoor environment with one person walking through the front of the robot. Future works include the use of instance segmentation to filter only the pixels corresponding to the moving object and use the rest of the image for the SLAM process.

6. ACKNOWLEDGEMENTS

The authors of this work gratefully acknowledge the support of NVIDIA Corporation with the donation of the Jetson TX2 board used for this research, and the financial support of CNPq.

7. REFERENCES

Alcantarilla, P.F., Yebes, J.J., Almazán, J. and Bergasa, L.M., 2012. "On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments". In *International Conference on Robotics and Automation (ICRA)*.

Bescós, B., Fácil, J.M., Civera, J. and Neira, J., 2018. "DynaSLAM: Tracking, mapping and inpainting in dynamic environments". *IEEE RA-L*.

Carlone, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. and Leonard, J., 2016. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age". *IEEE Transactions on robotics*, Vol. 32, pp. 1309–1332.

Engel, J., Schöps, T. and Cremers, D., 2014. "Lsd-slam: Large-scale direct monocular slam". In *European Conference on Computer Vision*. Zürich, Switzerland.

Girshick, R., 2015. "Fast r-cnn". In *Proceedings of the IEEE international conference on computer vision*.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation". In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. "Mask r-cnn". In *IEEE international conference on computer vision*.

Labbé, M. and Michaud, F., 2018. "Rtab-map as an open-source lidar and visual slam library for large-scale and long-term online operation". *Journal of Field Robotics*.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C., 2014. "Microsoft coco: common objects in context". In *European Conference on Computer Vision (ECCV)*.

Mur-Artal, R. and Tardós, J.D., 2017. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras". *IEEE Transactions on Robotics*, Vol. 33, pp. 1255–1262.

Quingley, M., 2009. "Ros: an open-source robot operating system". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. "You only look once: Unified, real-time object detection". In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. "Faster r-cnn: Towards real-time object detection with region proposal networks". *Advances in neural information processing systems*, pp. 91–99.

Soares, J. and Meggiolaro, M., 2018. "Keyframe-based rgb-d slam for mobile robots with visual odometry in indoor environments using graph optimization". In *Proceedings of the IEEE Latin American Robotics Symposium*.

Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D., 2012. "A benchmark for the evaluation of rgb-d slam systems". In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Algarve, Portugal.

Tan, W., Liu, H., Dong, Z., Zhang, G. and Bao, H., 2013. "Robust monocular slam in dynamic environments". In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.