3D Object Localisation With 2D CNN Object Detector And 2D Odometry*

Rafal Staszak[0000−0002−5235−4201] and Dominik Belter[0000−0003−3002−9747]

Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poznan, Poland
rafal.p.staszak@doctorate.put.poznan.pl

Abstract. In this paper, we deal with the problem of objects detection and 3D position estimation by a mobile-manipulating robot equipped with an RGB-D camera and 2D laser scanner. Instead of estimating 3D position from a single image using CNN, we propose an application of CNN-based 2D object detection and gradient-based optimization that allows estimating 3D object poses from a sequence of images and robot poses obtained from an on-board 2D localization system.

Keywords: Object detection · pose estimation · mapping.

1 Introduction

An autonomous mobile robot equipped with vision, mapping, and odometry systems provides data that can be used to extract object features in an environment. In robotics applications, it is crucial to be able to determine the class and location of objects in the three-dimensional space. This might aid further execution of different robotic tasks, in which approximate object location in 3D space are needed.

In this paper, we deal with the problem of estimating the 3D position of the objects with respect to the 3D map created by the robot moving in this environment. We assume that the robot is equipped with an RGB-D camera used to register a 3D model of the environment and detect objects. Also, the robot localizes itself using onboard sensors like 2D laser scanners.

The work has the following key contribution:

- determining 3D object positions from a set of 2D detections,
- GPU implementation of a gradient-based optimisation method for estimating 3D object position.

2 Related Work

The majority of modern learning-based methods make use of deep neural networks, which both detect and predict object poses in a single-shot fashion [1–3].

* This work was parialy supported by the National Centre for Research and Development (NCBR) through project LIDER/33/0176/L-8/16/NCBR/2017 and PUT project SBAD

Such approaches excel at determining the poses in terms of translation and rotation. Moreover, they only need to be fed with a single set of image data, which speeds up the process of detecting and inferring object locations. However, the training phase of these neural networks requires training dataset containing detailed information about every object instance, its pose in space, and the bounding box in the image.

Another methods employ online optimization or matching in order to track the objects [4, 5]. These methods compute metrics similarity between the object model and the intensity of pixels in registered images. These solutions might prove to be efficient, although they require storing exact representations of either CAD models or image patterns with assigned poses. In contrast to methods that learn the 3D pose of the object and classical methods that are based on optimization only. We propose a method that utilises 2D CNN and image-based object detector [6], the estimated pose of the robot, and the efficient optimization on the GPU to estimate the pose of the detected objects in the 3D space.

3 Object Position Estimation System

During the environment scanning procedure, the object detector [6] continuously processes registered images and the information about bounding boxes, object classes, odometry measurements. After that, the detected bounding boxes are used to calculate the object middle points in every registered image. Given the odometry measurements, a set of lines $\vec{n}_i^c$ is drawn crossing the camera position m_i^c and the 3D projection of the determined middle points. In such a way, every detected object instance has a corresponding set of lines that represent the directions of those objects being observed from various viewpoints.

In a best-case scenario, the lines should intersect in the same point in space. However, the inaccuracy of the object detector parameters leads to a problem of minimising a metric error relative to the line set.

The goal of the optimization process is to estimate the position of an object of c class which is denoted as o^c. It is worth noting that the point o^c is optimized against a whole set of lines inferred from object detections for consecutively registered frames in the scanning procedure. The number of detected lines for a given class c depends on the number of input images and successful detections within a determined IoU threshold.

$$\min_o d\left(o^c\right) = \sum_{i=1}^{n} \frac{\left\|\left(o^c - m_i^c\right) \times \vec{n}_i^c\right\|}{\left\|\vec{n}_i^c\right\|}. \tag{1}$$

The minimized value is an accumulated distance between object point o^c and the whole set of corresponding lines l_i^c. This can be calculated given the anchor points m_i^c and directions of lines $\vec{n_i^c}$:

In theory, the cost function displays convex characteristics and converges quickly towards a sub-optimal minimum. If an object is observed from various viewpoints, then the resulting lines may form a conical structure. The calculated

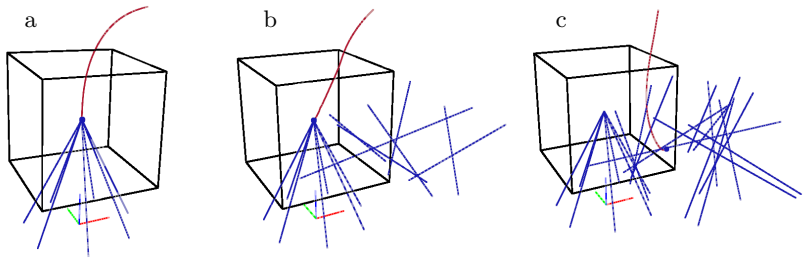

Fig. 1. Experimental estimations with various disturbance level. The intersection coordinates of 7 lines are $(0, 0, 1)$ and the unit cube showcases the scale. The estimation results in (a) and (b), which contain 0 and 7 disturbance lines, respectively, are close to the intersection point ($d_a = 0.003$) and ($d_b = 0.004$). The example with a significant disturbance (20 randomly generated lines) shows a relevant disparity ($d_c = 0.842$).

gradient is attracted towards an approximated apex of the cone. A large number of false-positive detections disturb the final outcome. However, single outliers can be outweighed by the correct detection if the number is significantly higher.

3.1 Parallel Computing

The gradient-based optimization method was formulated through matrix equations, where line parameters have been included in different components. Therefore, it was possible to make use of enormous performance in processing matrix operations and gradient calculations included within PyTorch.

4 Experiments

4.1 Datasets

Firstly, it was essential to train the object detector in order to recognise objects and extract their corresponding bounding boxes from the recorded frames. We have selected a range of objects: ring, sponge, dispenser button, probe, button, and plastic stands of 2 types. In total, 1347 sample images have been registered using Kinect for Xbox One depth sensor. Each training sample with contains annotated ground truth bounding boxes and object classes.

4.2 Results

During the experimental verification of the proposed method, the robot was continuously registering the current information about its kinematic structure, odometry, and RGBD data from the depth sensor in the given environment. The robot was controlled manually during this procedure and localized using the Slamtec Mapper M1M1 scanner. The obtained model of the environment and 3D position of the detected objects are presented in Fig. 2.

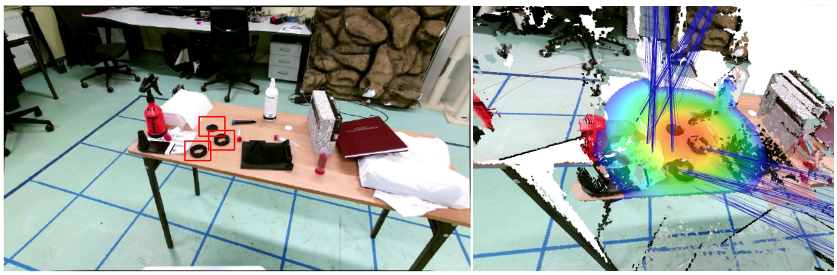

Fig. 2. Estimated positions of the objects (black rings) on the map obtained from the RGB-D camera (right). Blue lines represent measurements defined by the robot pose and the 2D detection results on the RGB images. The heat map represents proximity to the point that is found in an optimization process.

5 Conclusions And Future Work

In this paper, we present a system that estimates the positions of objects in 3D space using a 2D CNN-based detector and poses of the robot estimated by 2D lidar-based SLAM. The problem is formulated as an optimization problem that is efficiently solved using parallelized GPU implementation. In the feature, we are going to extend the problem to simultaneously estimation of objects in 3D space and robot poses like in object-based SLAM [7].

References

1. W. Kehl, F. Manhardt, F. Tombariand S. Ilic, and N. Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538. IEEE, 2017.
2. Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems*, 2018.
3. C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *Computer Vision and Pattern Recognition*, 2019.
4. T. Hodan, X. Zabulis, M. Lourakis, S. Obdrzalek, and J. Matas. Detection and Fine 3D Pose Estimation of Textureless Objects in RGB-D Images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4421–4428. IEEE, 2015.
5. Y. Konishi, K. Hattori, and M. Hashimoto. Real-Time 6D Object Pose Estimation on CPU. In *arXiv*, 2018.
6. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C Berg. SSD: Single shot multibox detector, European Conference on Computer Vision. In Leibe B. et al., editor, *LNCS vol. 9905*, pages 21–37. Springer, 2016.
7. Kyel Ok, Katherine Liu, Kris Frey, Jonathan P. How, and Nicholas Roy. Robust object-based slam for high-speed autonomous navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 669–675, 2019.