Smartphone based ORB-SLAM embedded system for a mobile robot

Ernest Bardadyn

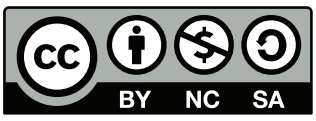
Department of electronics, Wroclaw University of Technology

Supervisor: Ph. D. Witold Paluszyski

Janurary 21, 2018

Abstract

The main goal of this project was to evaluate the usability of a monocular camera based SLAM system running on an Android smartphone for mapping and localization of a mobile robot. The secondary goal was to allow an Android app to send control commands to a robot through a wireless connection.



1.Introduction

Regular SLAM algorithms require expensive sensors based either on laser distance measuring, sonars or multiple cameras. A system based on a common and cheap Android smartphone could allow to easily implement such functionality in most types of mobile robots.

1.1 Goals

1. Create a new or utilize an existing application executing one of the monocular SLAM algorithms
2. Mount a smartphone on a mobile robot
3. Establish a wireless connection between robot and a smartphone to send control commands to allow a robot to travel to a selected place

1.2 Tools

1. Android Studio 3.0.1 - to edit and compile the android app
2. Android Native Development Kit r11 - to compile the native c++ code into libraries executable on a smartphone
3. Python with RosAria package and RosAriaDriver class for remote controlling of a Pioneer 3-DX mobile robot from a PC computer
4. CloudCompare v.2.6.2 - a tool for visualizing point clouds

1.3 Hardware

1. LG K8 smartphone
2. Pioneer 3-DX mobile robot

3. PC computer

2.ORB-SLAM application

The application used in this project is available to download from a GitHub repository at the address https://github.com/FangGet/ORB_SLAM2_Android[2] . It is distrubuted under GNU General Public license. The mapping and localization is based on an ORB-SLAM2 alghorithm developed by Ral Mur-Artal, and Juan D. Tards.[1] It allows to create a point cloud of feature points acquired using ORB alghorithm available in an OpenCV library. The algorithm resolves spatial positions of points relative to the agent by comparing a series of consequent mono images taken by a forward or backward moving camera.

The repository was imported as an Android Studio project, a necessary additional functionality of saving acquired points to a csv file was written and finally the native C++ code was compiled using ndk-build for armeabi-v7a architecture and the app was installed to a smartphone. The OpenCV manager application had to be additionally installed on the device in order to run the app.

3.Hardware

1. Pioneer 3D-X - a differential drive mobile robot with an on-board industrial computer. It is equipped with sonar sensors and a wireless ethernet card.

2. LG K8 android smartphone - equipped with 1,30 GHz 4-core processor, 1.5GB of RAM memory and running an Android 6.0 operating system.

4.Robot control

A PC computer available in the L1.5 room was used to control the robot using local wireless ethernet connection. The installed RosAriaDriver_package[3] allows to write simple python applications for controlling the robot. A basic wander procedure with collision avoidance based on the sonar data was created.

5.ORB-SLAM handheld testing

First tests of the application were performed from hand. Careful movements allowed to capture a series of points mapping a room, an object (coffee mug) and a house interior. The main issues proved to be:

1. Low FPS rate of the camera due to the high cpu load of the smartphone due to extensive computation of the SLAM algorithm
2. Losing the continuity of tracked points during turns

Despite those problems the application was able to recognize the current position of the camera provided that the exact location has been already mapped. The point clouds of each test were saved and imported to CloudCompare which allows for visualization of the data. The results are presented below:

1. Mapping of a desk with a focus on a coffee mug

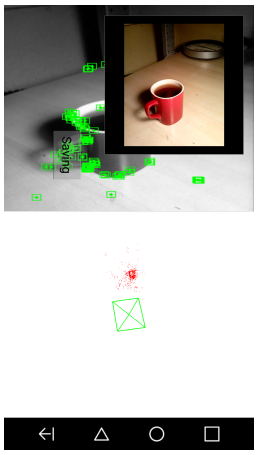


Figure 1: Phone screen during mapping

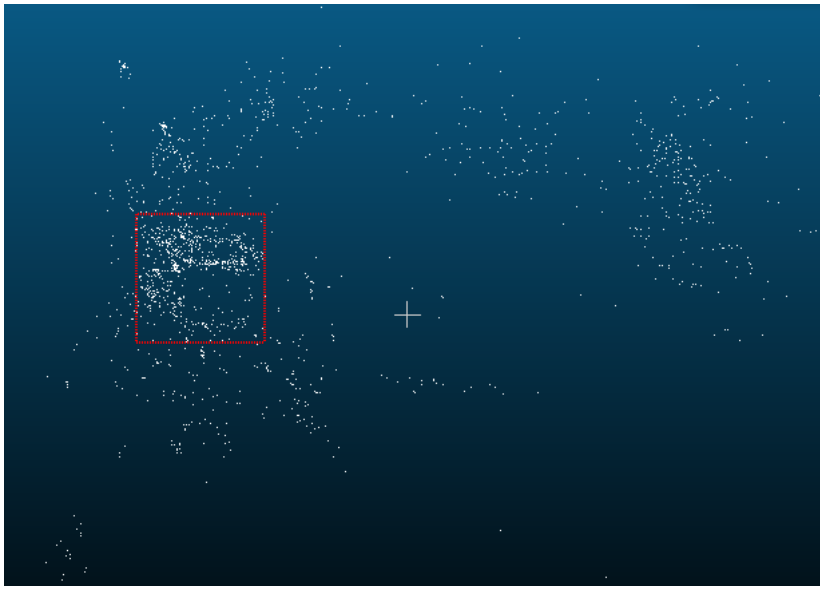


Figure 2: Resulting point cloud

2. A map of a house

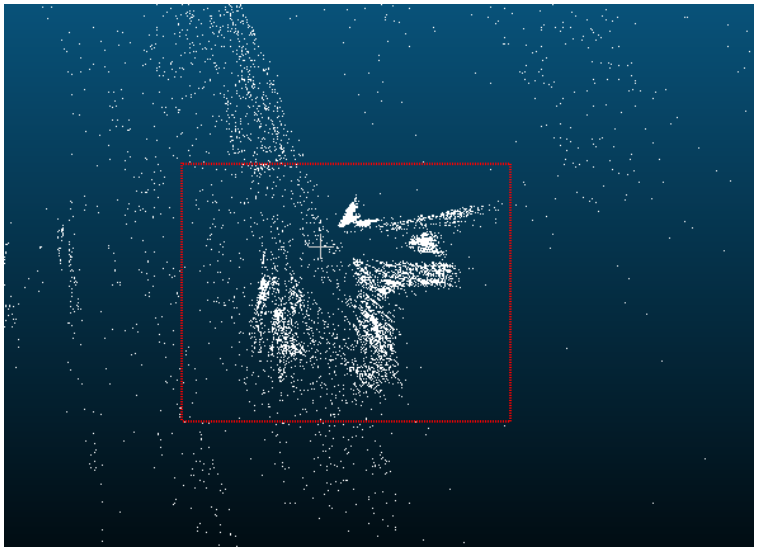


Figure 3: Resulting point cloud

The resulting data is very noisy and disjointed due to the loops not having been closed.

5.On robot testing

The smartphone was mounted on the top of the robot using a metal mounting piece and a rubber band. The application was initialized and the wander program was ran on the computer with a wireless connection to a robot. The system failed to keep the continuity of the map points during turns and therefore the acquired points were taken only for the initial part of the robot path, from the start up to the first turn at an obstacle.

6.Results and conclusions

The algorithm was evaluated for its applicability in a simple system. Due to the used smartphone having very low computing power the system works slowly and in order to keep the continuity of the map required for the localization the movements of the camera have to be slow. The main issue with using a simple robot movement strategy is at the turns. The ORB-SLAM running on a slow device can not acquire new map points when the camera is not moving forwards or backwards, because the feature points disappear from the field of view before they can be registered as map points, therefore it can not keep the continuity when the robot is turning around its own axis. A different robot control program or a different kind of a mobile robot could give good results using the same SLAM application. ORB-SLAM gives good real time results when applied to datasets recorded from moving cars since the turns are taken with a bigger radius and the map points can be easily acquired even during them. The smartphone - robot communication and control part has not been developed during this project.

References

(1) *ORB-SLAM: a Versatile and Accurate Monocular SLAM System* Mur-Artal, Raúl, Montiel, J. M. M. and Tardós, Juan D.

(2) ORB-SLAM2 android app repository - https://github.com/FangGet/ORB_SLAM2_Android

(3) RosAriaDriver Python class by Damian Baranski - http://panamint.ict.pwr.wroc.pl/ dbaransk/rosaria_drive/dox/html/index.html