# Open Source SLAM Library for Embedded Systems

Degree programme: Master of Science in Engineering | Specialisation: Industrial Technologies


Simultaneous location and mapping (SLAM) is a technology used for robot navigation and augmented reality. Today most SLAM libraries are proprietary or not ready for embedded systems. In this thesis, we write and analyse a library based on Semi-Dense Visual Odometry (SVO), which runs on embedded systems. Compared to the publicly available implementation of SVO, our version uses stereo cameras as input and it has fewer dependencies to third-party libraries.


## Introduction

Humans use different sensors to estimate the pose of their head in a room. Everyone who has ever tried to stand on one leg with their eyes closed knows that closed eyes make finding the balance harder. This experiment shows that our eyes are an import factor for us humans to balance. In this thesis we use optical information from a camera to estimate its angle and position (pose). Knowing the position of an object is important in a lot of different applications and becomes always more important towards autonomous navigation. Examples of such applications are drone navigation or augmented reality. Those applications require a flexible and ideally mobile device.

The lab for Computer Perception and Virtual Reality (CPVR) at the Bern University of Applied Science uses an algorithm called ORB SLAM for projects related to augmented reality. It tweaked the algorithm to achieve an acceptable frame rate of 10-20 frames per second(fps) on mobile phones. However, a faster frame rate would mean that no interpolation is needed or that the algorithm could run on lower end devices.

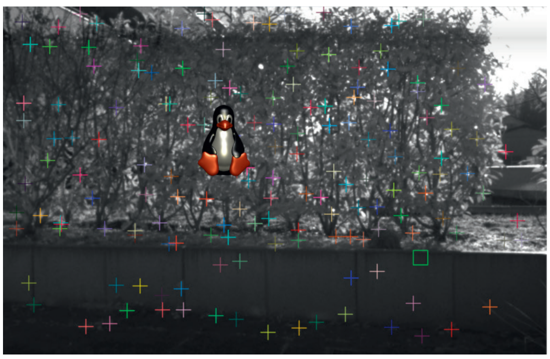
## Concept

In this thesis, we implement and analyse an algorithm called Semi-Dense Visual Odometry (SVO) which is capable to run on embedded devices. The reference implementation uses monocular cameras, while we use a stereo camera. To estimate the pose the algorithm performs four tasks. First, it maintains a 3D point cloud used to estimate the camera pose. Second, it performs a pose estimation based on the 3D cloud. We call this sparse image alignment. It uses a changed version of optical flow, which outputs a pose instead of a warp matrix. Third, to make the guess more accurate, it refines the pose by using standard optical flow, followed by a minimization of the re-projection error. Finally, in a last step, the algorithm performs a 3D point cloud update where it refines the cloud by taking the new pose into account. We implemented the algorithm as a library. To showcase it, we wrote an augmented reality application which is shown in the left figure.

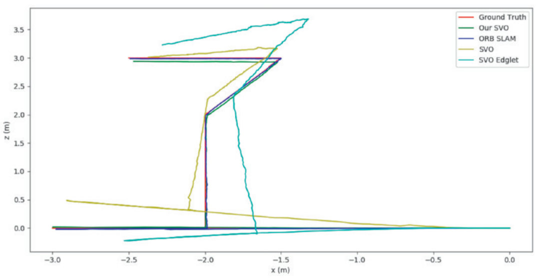
## Results

We compare our own implementation with two monocular implementations of the same algorithm and with stereo ORB SLAM. To compare the algorithms we generate a synthetic scene which we generate with the computer graphics software Blender. This method allows us to render a defined trajectory as ground truth. The figure on the right shows the trajectories of the different SLAM implementations from a top view compared to such a ground truth. We see that our SVO implementation and stereo ORB SLAM perform similar. However, our implementation runs almost twice as fast. The monocular SLAMs are, as expected, less accurate because they need to do a complex depth estimation. Based on the results, we conclude, that stereo SVO shows great potential as a SLAM algorithm for embedded devices.


Stefan Eichenberger
eichest@gmail.com

Comparison of Different SLAM Implementations




Augmented Reality Application