M1 Project Summary

Turtlebot localization with visual markers

Mahmoud ALI, Kevin SERRANO

Supervisor: Olivier Kermorgant - Ecole Centrale de Nantes

Full-Report Link

YouTube Video 2, trilateration-absolute Localization

YouTube Video 1, Extended kalman Filter EKF-hybrid localization

The robot localization is considered as a key element for autonomous robot that is able to map, navigate and plan. If a robot isn't able to determine where it is, it will be difficult to decide what to do next. Autonomous robot must consistently identify its position while moving in a given map. One of the fundamental problems for providing a mobile robot with autonomous capabilities is to combine different sensory information to locate the robot in its environment.

We have placed three beacons inside the robot environments, beacons that we used were three spheres with different colors and known size. **OCCAM Omni 60 camera** captures 360$\circ$ panoramic video was used to detect the beacons, then the distances from the robot to each beacon were calculated to be used for the robot localization.

Different image processing operations were done to detect the three beacons. First, the five images -output of OCCAM Omni 60- were rectified and tiled together, then a blob detector algorithm was used to detect the beacons by applying different filters (according to color, shape, and size of the beacons), Finally the distance to each beacon was calculated using the actual and apparent diameters of the beacons and the focus length of the camera.

Different frameworks are used for robot localization, relative localization "Odometry", Absolute Localization, Sequential relative and absolute localization, and Hybrid localization. Making use of the three distances from the beacons to the robot we were able to implement both absolute and hybrid localization methods for turtlebot.

Two techniques for absolute localization framework were implemented e.g Trilateration and Optimization techniques. Trilateration is the process of finding the absolute locations of points by measurements of distances. Using the trilateration formulas, The absolute robot location was calculated according to the three distances to the beacons. For the second method "optimization techniques", a c++ library **RobOptim** was used to minimize a function -which specifies the error in the three distances for a given robot pose- under certain conditions.

One way to fuse different data together and estimate the robot location is using Bayesian filter. Extended Kalman Filter **EKF** is a form of Bayesian Filters is being used when the state has a Gaussian distribution. **EKF** estimates the robot location based on some observations and a model of the evolution of the system.

In our case, The evolution model was the Turtlebot motion model which describes the robot's pose as a function of it's previous pose and control inputs "odometry formulas for (2,0) mobile robot". The observation model describes the measurement -three distance to beacons- as a function of the current pose of the robot and the beacon locations.

The project output was a c++ ROS-based framework for robot localization using visual markers. Different nodes were implemented to perform different operations:

- node to perform image Rectification.
- node to perform image Tiling.
- node to implement Blob detection and to calculate the distance for each beacon.
- node to correctly interface beacon distance to localization algorithm.
- node to implement Trilateration.
- node to implement optimization.
- node to implement EKF algorithm.
- nodes for visualization of the environment.

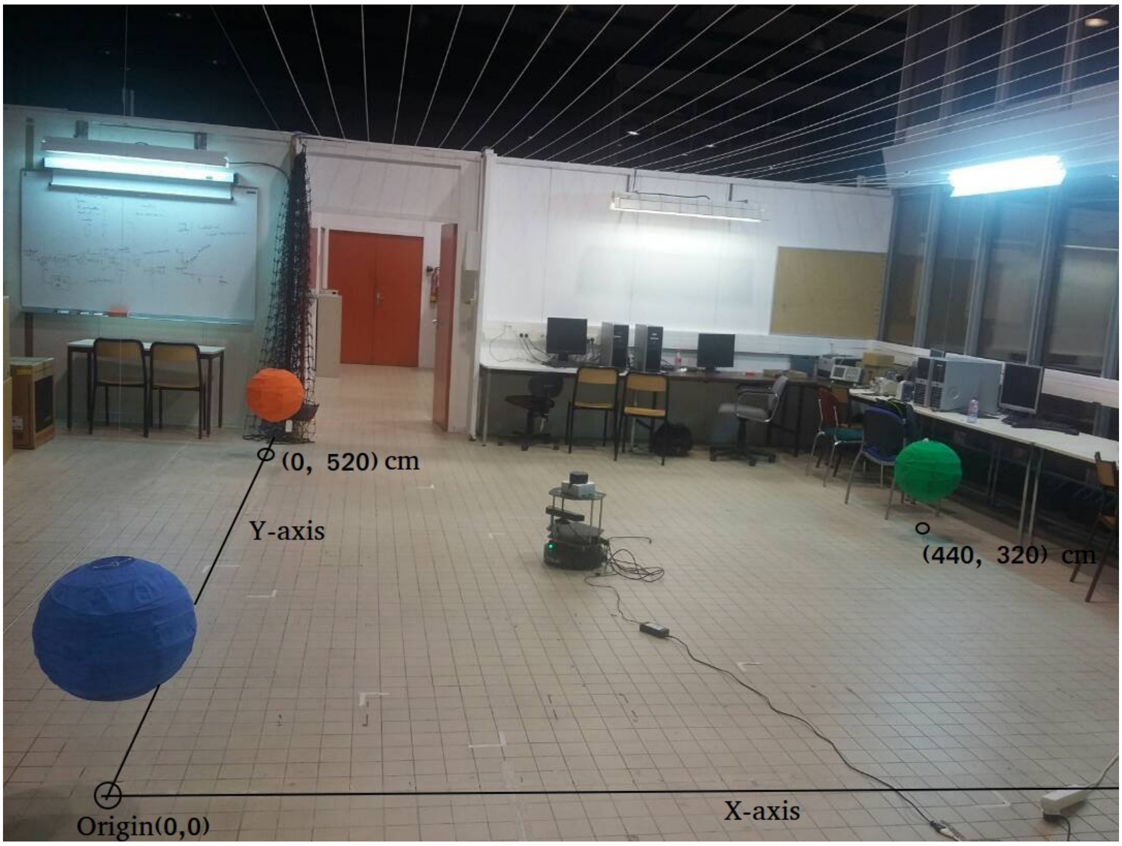

Figure 1: Robot Environment- Project Room