Sistemas Autónomos / Autonomous Systems

Mapping using a Laser Range Finder,

David Teles , Leonor Silva, Filipe Silva, Pedro Soares

contacto@davidteles.com, leonor.oliveira.e.silva@tecnico.ulisboa.pt, filipe.fialho.silva@tecnico.ulisboa.pt, pedro.a.s.soares@tecnico.ulisboa.pt

MEEC/MEIC

Instituto Superior Técnico, Universidade de Lisboa

What we wanted to do!

The main purpose of the project was to create a software that would use a moving platform "robot" with a LRF "Laser Range Finder" that could create a map of the surroundings in real time, using ROS as the communication basis.

Pionner 3DX

Hokuyo URG-04LX-UG01

For this, we started the development using as the main platform the Pionner 3DX and the Hokuyo URG-04LX-UG01.

The assumptions that we made!

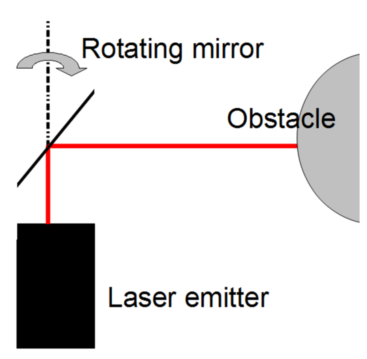
For this project we made some early assumptions, that are really important for the understanding of the following decisions that we made.

- The position of the robot is known and precise enough to do mapping
- All of the TF are published and correct
- We are using a 2D LRF

The main problems to be solve!

1. Getting the laser data to a easy to process state

Since the LRF works by shooting a laser into a rotating mirror, all the readings have an associated range and angle that need to be transformed into a 2D point cloud (array of x and y coordinates)

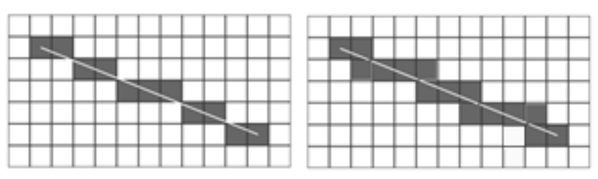


The point cloud is in the laser frame, so we needed to get it to the map frame, to do this we used the the TF topic to get the Transformation and Rotation between this two frames.

2. Detecting which cells were crossed by the laser beam

To determine which cells were in the path of the laser beam an algorithm similar to Bresenham's line rasterization was used. An alteration was made to include every cell crossed.

3. Updating the map



After detecting which cells were crossed by the laser beam these are classified according to the following probabilistic model

$$p(m_i|z_t, x_t) = \begin{cases} 0{,}3 \; if \;\; considered\; free \\ 0{,}6 \;\; if \; considered\; occupied \\ 0.5 \; if \; out\; of\; range \end{cases}$$

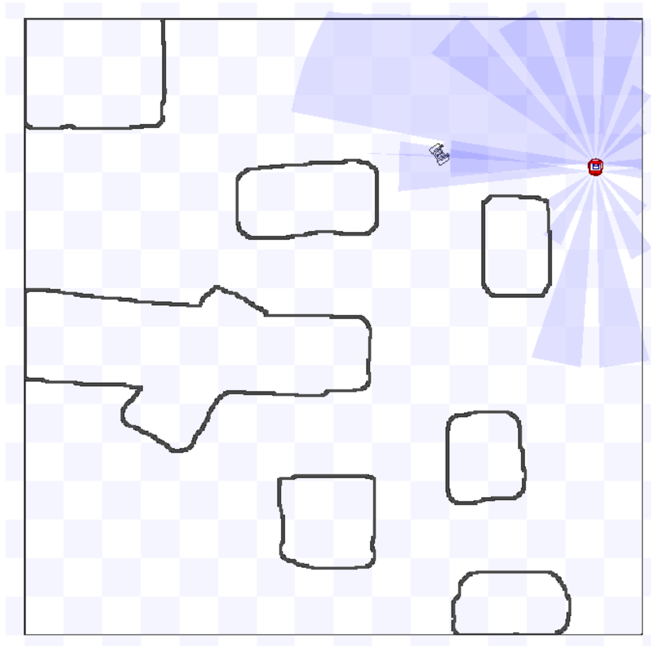
Then basically this process is repeated cyclically after each measurement and all the present and posterior probabilities assigned to each cell are taken into account by using the following model

$$p(m_i|z_{1:t}, x_{1:t}) = \left[1 + \frac{1 - p(m_i|z_t, x_t)}{p(m_i|z_{:t}, x_t)} \frac{1 - m_i|z_{1:t-1}, x_{1:t-1})}{p(m_i|z_{1:t-1}, x_{1:t-1})} \frac{p(m_i)}{1 - p(m_i)}\right]^{-1}$$

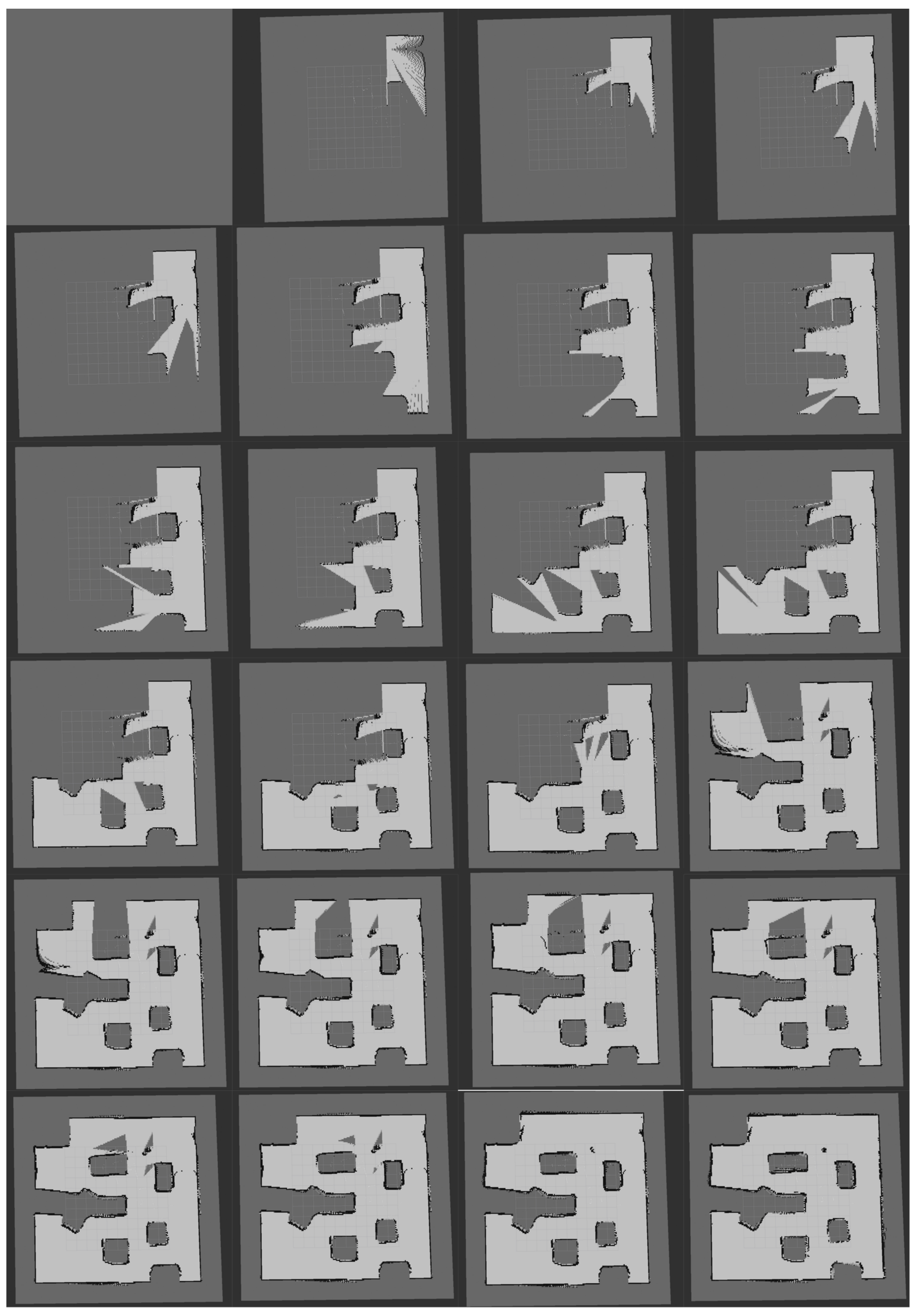
Results

After finishing the software it was time to do some testing. To have a good comparison we run the software in two different instances.

Using a ROS Stage simulating different hardware than the one that we use during the software development.



The stage used to test the software



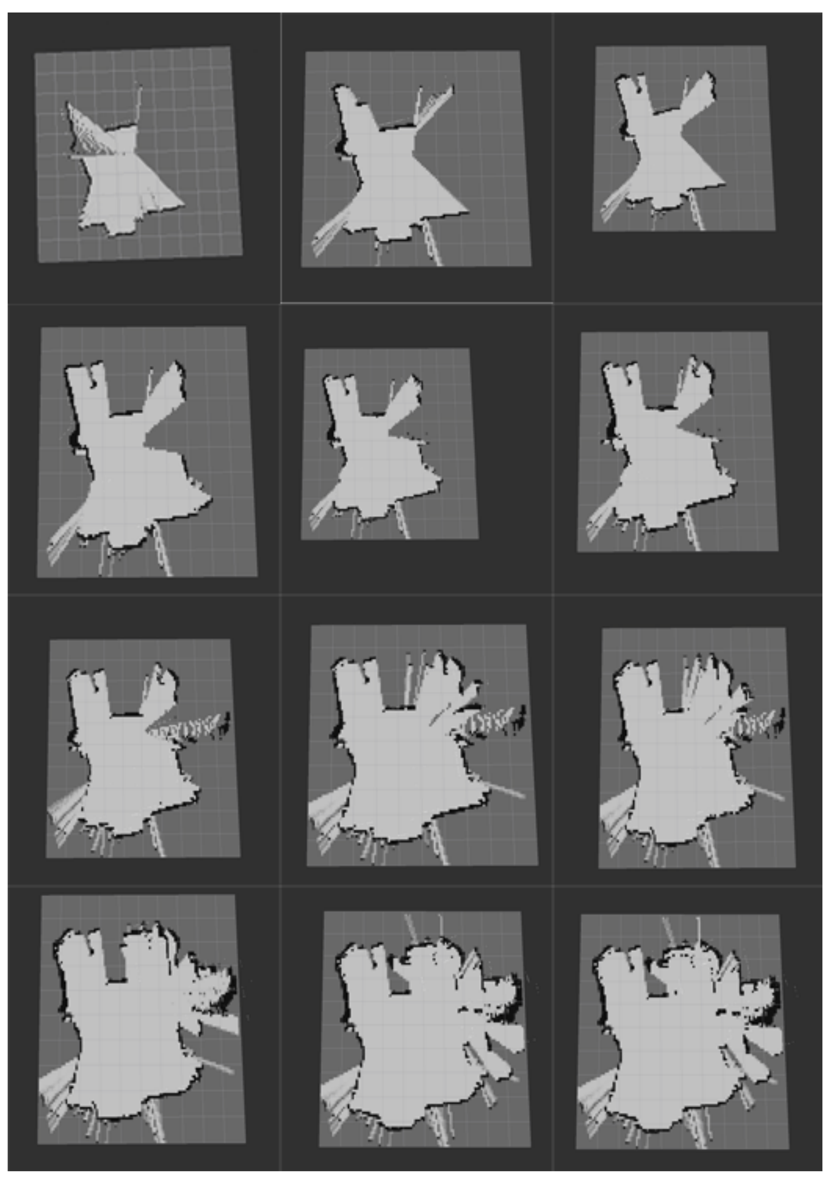
The evolution of the map, when testing the software using the ROS Stage

When comparing the two images we can see that the fidelity between the map and the stage is good, we used a resolution of 10 cm per square to help in the processing time.

Because of the nature of the probabilistic map small errors, like the ones that can be spotted in the 5 row 1 column, are corrected with time giving us a better output result.

If we increase the resolution by a factor of 2x the processing grows to 4x, which means that we cannot move the robot too fast or we will have an incomplete map.

We also tested the software using a ROSBag acquired in the 8th Floor of the North Tower of Técnico Lisboa, using a Motion capture system, that uses a array of cameras, that triangulate a marker, placed in the robot giving us a very precise position and heading, which is one of the main predicates for mapping with this software.



The evolution of the map, using a ROSBag

With this we got the following result, that for proving the accuracy of the mapping isn't the best option since we don't have anything to compare with, but shows that we can run the software with real data, received in real time, still obtaining acceptable results using a 10cm per square resolution.

In this result we can also see some errors created by some reflective surfaces that can fool the laser range finder, since it uses an IR Laser to make the measurements, and creates some errors, that are slowly corrected with the robot moving around.

The output map can be used for localization and/or task planning on future operations where the space is already known providing that the obstacles disposition didn't change (static world).

https://github.com/davidteles/SAut-Mapping