Where am i

Abstract:

This project demonstrates the usage of ros packages to map and localize a robot for a given and created environment, using different mapping techniques.

1-Introduction

Earlier during the localization project the robot was able to localize itself given a map of the environment but what if it wasn't given a map, here SLAM (simultaneous localization and mapping) comes in play where a robot is able to map its environment using sensor measurements to localize itself and then create a map of the environment.

2-Background

Sensor measurements that are read may contain noise which in turn makes the values inaccurate, putting that into consideration here is how each algorithm helps solve this problem and make mapping and localizing the robot an achievable task.

2.1 Slam:

Slam comes in 2 forms:

Online : estimating pose and features at time t-1.

Full slam : estimating trajectory and features at time t.

2.2 Occupancy grid:

a mapping algorithm that labels the cells in a grid free , occupied , unknown , its purpose is to build a map of the environment and localize the robot with respect to the map.

2.3 Grid Based Fast Slam:

Combines the monte carlo localization algorithm with occupancy grid ,it estimates landmarks and the robot poses thus solving the fast slam problem , using the low extended kalman filter it is able to solve features on the map.

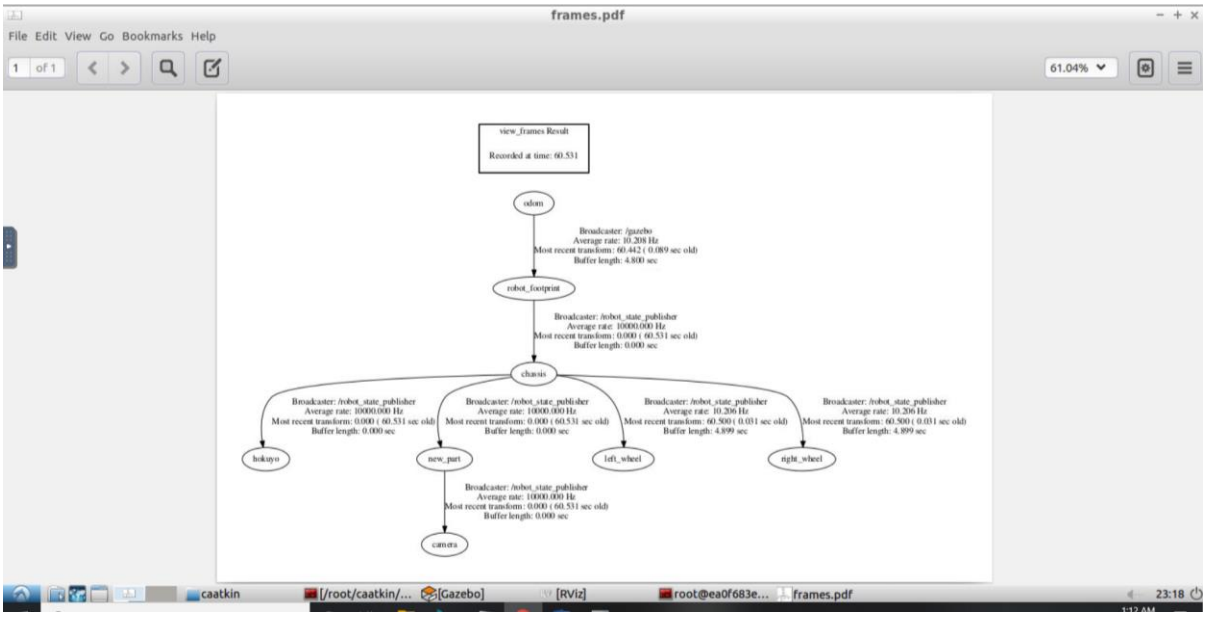
2.4 Graph Slam:

Slam problem is constructed as a graph were it tries to solve the full slam problem by creating the map and path and uses the data to check if there are relationships between poses of respective intervals (t-1) and t.

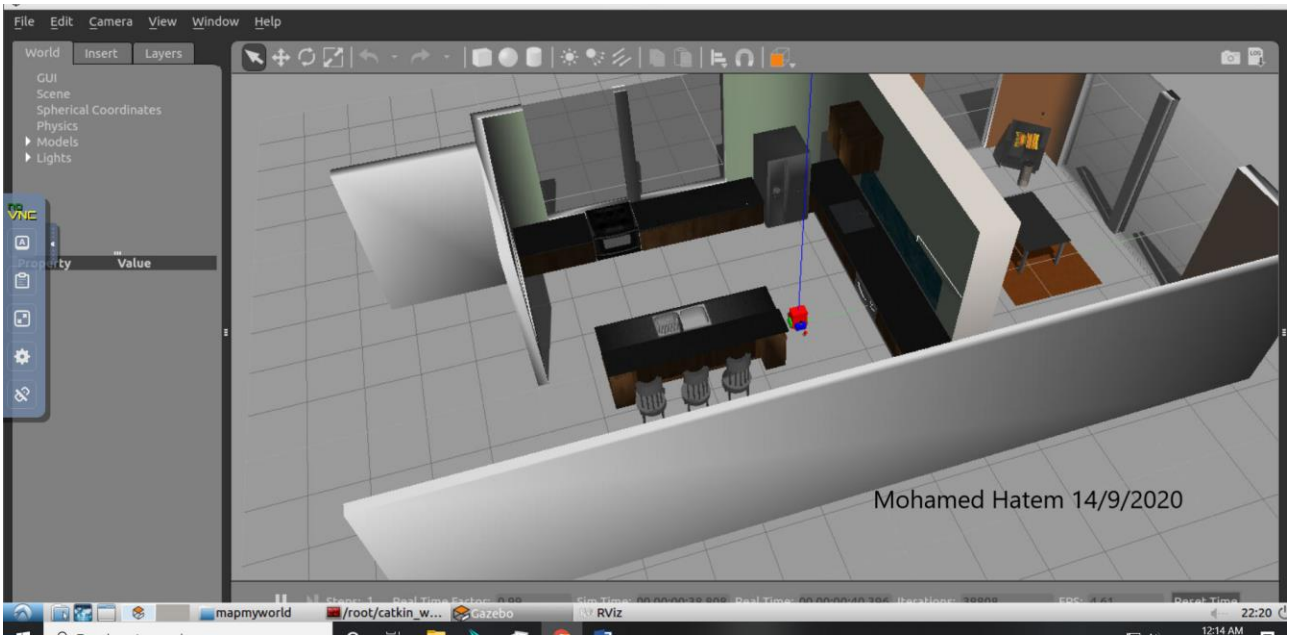
3- Configuration:

The robot model used in this project is the same robot from previous project -> where am I, with the only difference being the new rgbd camera instead of the old one.

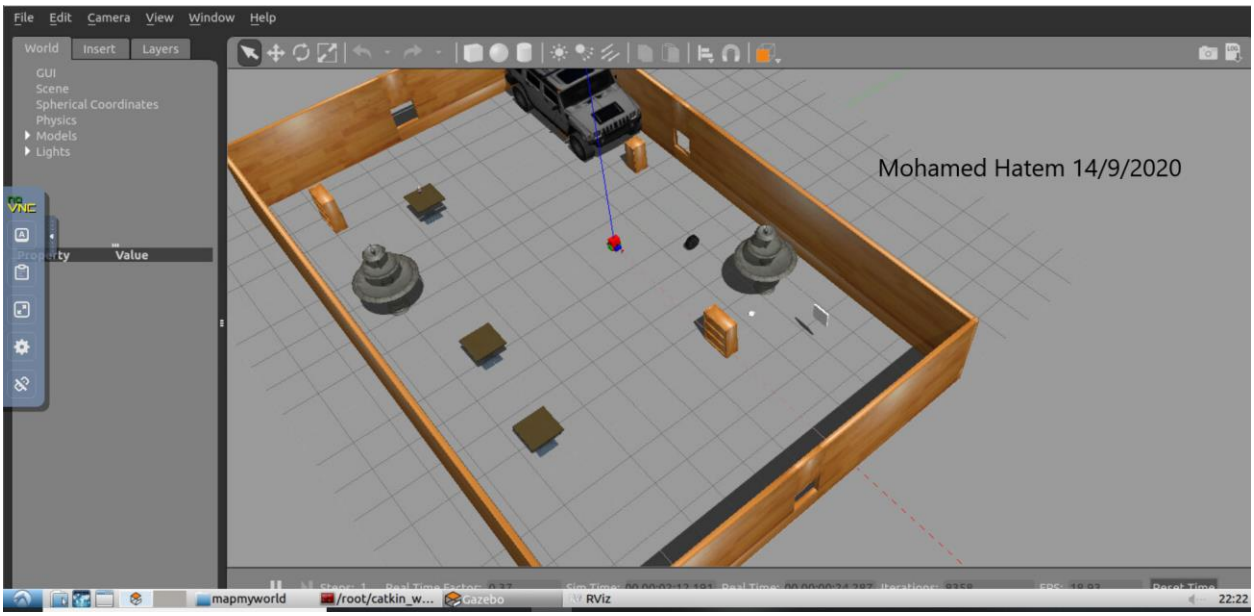
Rtab ros package is used for mapping which takes as an input both the laser scan and raw image from the bot in 2 simulated enviroments:



1- kitchen dining



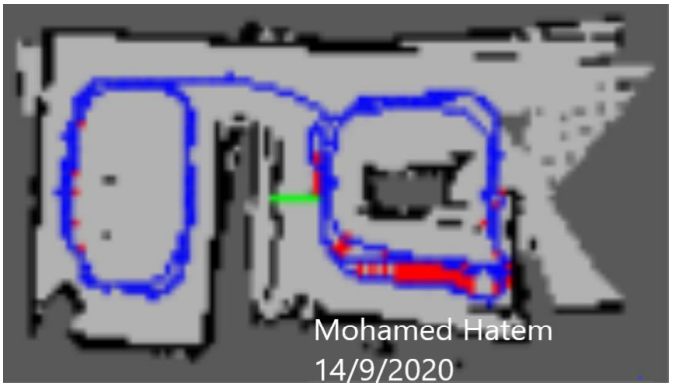
Custom created world:



4- Results

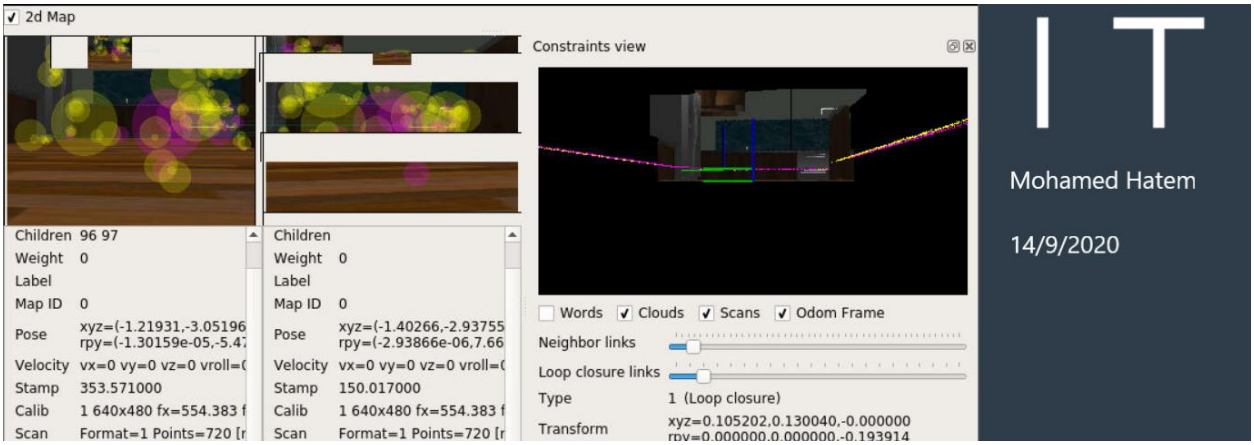
Kitchen dinning:

2dmap:



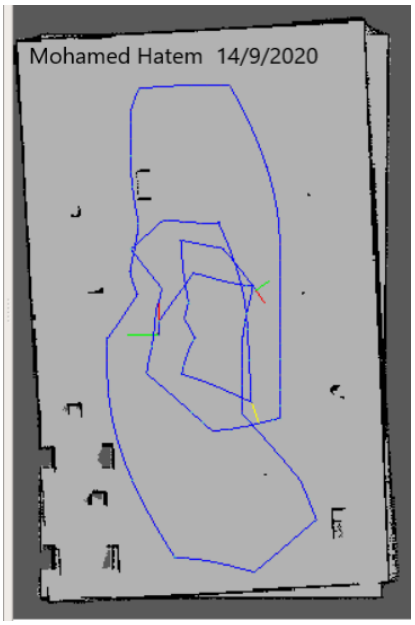
3dmap:

Loop closures:

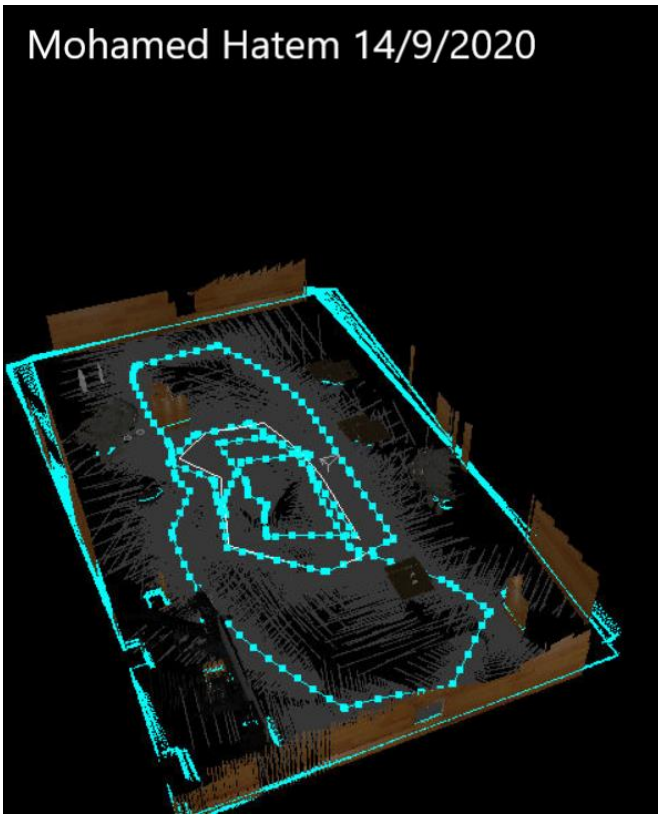


Custom created world:

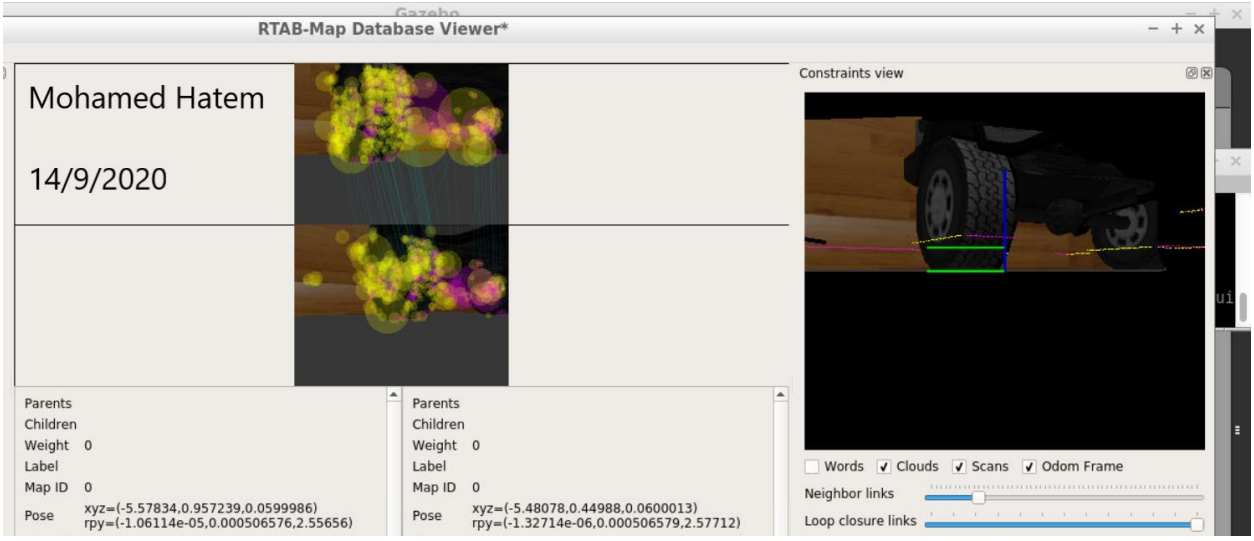
2dmap:



3dmap:



Loop closure:



5- Discussion:

the custom world had significantly less loop closures only 4 , compared to the supplied world (kitchen),

both 2d and 3d maps were generated with good results recognizing different features in both enviroments, the custom enviorment had larger space to navigate which given the same amount of time spent on both worlds gave those results. Aswell as the kitchen world being more realistic , slow performance and processing power didn’t allow for any further time for more driving of the robot.

6-Conclusion/future work:

Create a more realistic map with more features to test , create a different robot model , possibly a drone with different camera angle , this could lead to better results.