

Progress Report

1st Mykhailo Dmytrenko

2nd Antonio Sgorbissa

I. INTRODUCTION

In this brief overview the current progress of the project will be discussed, highlighting different points of interest, as well as plans for the future work. In the context of human SLAM for environmental disasters, there are currently several subjects of the main focus : **Karto - SLAM, Ontologies, Voice recognition and usage of extra sensors - Thingy91.**

II. KARTO - SLAM

Following the work of the Master thesis student who worked on this project, Karto-SLAM was chosen as a framework for the graph-based SLAM. The above-mentioned was improved significantly performance-wise and was able to successfully generate a graph based map real-time with a size of 120 by 55.6 meters with 2706 nodes in total with 1 forced loop closure and 7 automatic closures.

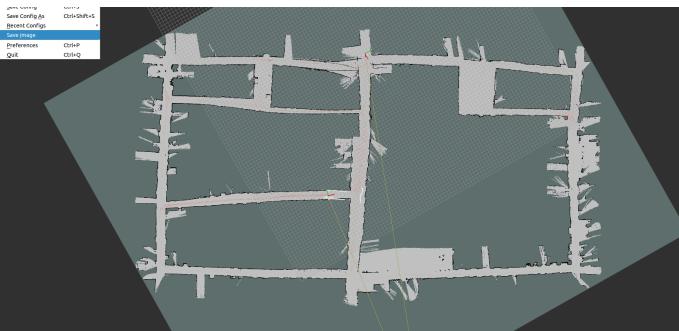


Fig. 1. Optimized Map built in real-time

III. ONTOLOGY

A major introduction to the loop closure algorithm are ontologies. The words were chosen according to the typical landmarks which are typically seen in the urban environment - in this case the historical centre of Genova. Example photo can be seen in Fig.2. Corresponding ontology tree is depicted on the Fig.3.

Several Python scripts automatically read and generate lists of words compatible with the speech recognition engine as well as with graph-based SLAM.

IV. VOICE RECOGNITION

Finding a reliable offline recognition engine presented a major challenge. Initially, CMU Sphinx recognizer [1] has been chosen. However, it failed to prove to be usable enough for the needed application. The main issues were unrecognizable words, false negatives and inability to recognize all



Fig. 2. Example photo of the streetview



Fig. 3. Ontology generated

words in the ontology at once. A different engine was chosen called Picovoice [2]. It is an efficient offline solution which uses compressed neural networks and can work on resource-restrained microcontrollers. It was successfully adapted to the current application in tandem with the ontology and it is able to recognize all 94 words of current ontology at once.

V. EXTRA SENSORS

In the context of environmental disasters there is also a need to gather extra information about the environment - temperature, vibrations, humidity etc. Thingy91 from Nordic Semiconductor [3] does precisely this. The kit is packed with a multitude of sensors for motion, impact, air quality and much more. In addition to this, Nordic Semiconductors offers

a cloud which gets all the data from the kit through the cellular data connection which than can be retrieved from anywhere. Currently, there is a work in progress on retrieving the accelerometer data from the device.



Fig. 4. Thingy91

VI. SOFTWARE OVERVIEW

The general overview of the ROS nodes can be seen on Fig.5. For the audio input **Mapper** is listening for the word index which is then added to the graph node as a custom object. Then, in order to make use of semantic distances these indices are converted back into words where the ontologies information can be used again through a ROS service **dist_service**.

VII. DETERMINISTIC LOOP CLOSURE

Currently, deterministic loop closure algorithm uses 3 parameters to decide how to close the loop: laser scan response, physical distance and semantic distance. In case several criteria are met, the edge strength is determined proportionally to the above-mentioned parameters. E.g. the covariance of the link created is smaller the closer the nodes are to each other.

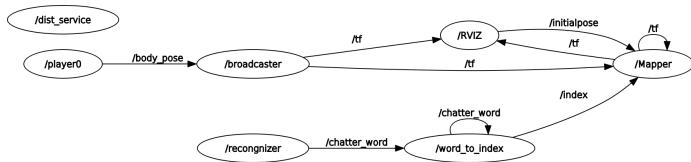


Fig. 5. Overview of ROS structure

REFERENCES

- [1] Sphinx home page. [Online]. Available: <https://cmusphinx.github.io/>
- [2] Picovoice homepage. [Online]. Available: <https://picovoice.ai/>
- [3] Thingy91 web-page. [Online]. Available: nordicsemi.com/Software-and-tools/Prototyping-platforms/Nordic-Thingy-91