Universiteit Antwerpen

Progress Report Week 12

Michiel Aernouts
michiel.aernouts@student.uantwerpen.be

May 20, 2017

Abstract

Optimisation of wireless networks is critical for the localisation of wireless devices. For this purpose, a wave propagation model of the environment can be created. Such a model contains a map of the environment combined with RF measurements that are obtained within that map. In this paper, we compare several visual SLAM algorithms such as LSD SLAM [1] and RGB-D SLAM [2] that can be used to render an accurate 3D map of an indoor environment. In order to test these algorithms, simulation software is used to navigate a drone around a room. A camera that is mounted on the drone provides necessary data for the algorithms. After finishing a SLAM algorithm, the resulting point cloud can be implemented in an OctoMap [3] to generate a volumetric representation. An initial guess of the environment is modelled and merged with the OctoMap that resulted from the SLAM algorithm. This way, an accurate model of the environment is created.

1 Progress

1.1 Static transformation

Last week, I mentioned trajectory errors caused by a wrong transformation between `/camera_link` and `/base_link`. I solved this by adding the correct yaw, pitch and roll angles to the `transformer.launch` file in the `kinect_transformer` package. Roll rotation has to be π radians, as the Kinect is mounted upside down. Also, pitch has to be 0.40 radians, as the Kinect is tilted 22 degrees downwards. When applying these changes, we can use RViz to validate that the transformation between the camera and the Erle Copter is correct.
After correcting the static transformation, I recorded a new dataset with the Kinect camera.

1.2 RGBD SLAM

The new dataset was used to create a new OctoMap with RGBD SLAM. However, I am still facing issues regarding computational resources. RGBD SLAM crashes after a while, causing the OctoMap to be incomplete. Only rooms V315, V317 and a part of the hallway where implemented in a fine map, as seen in figure 1.

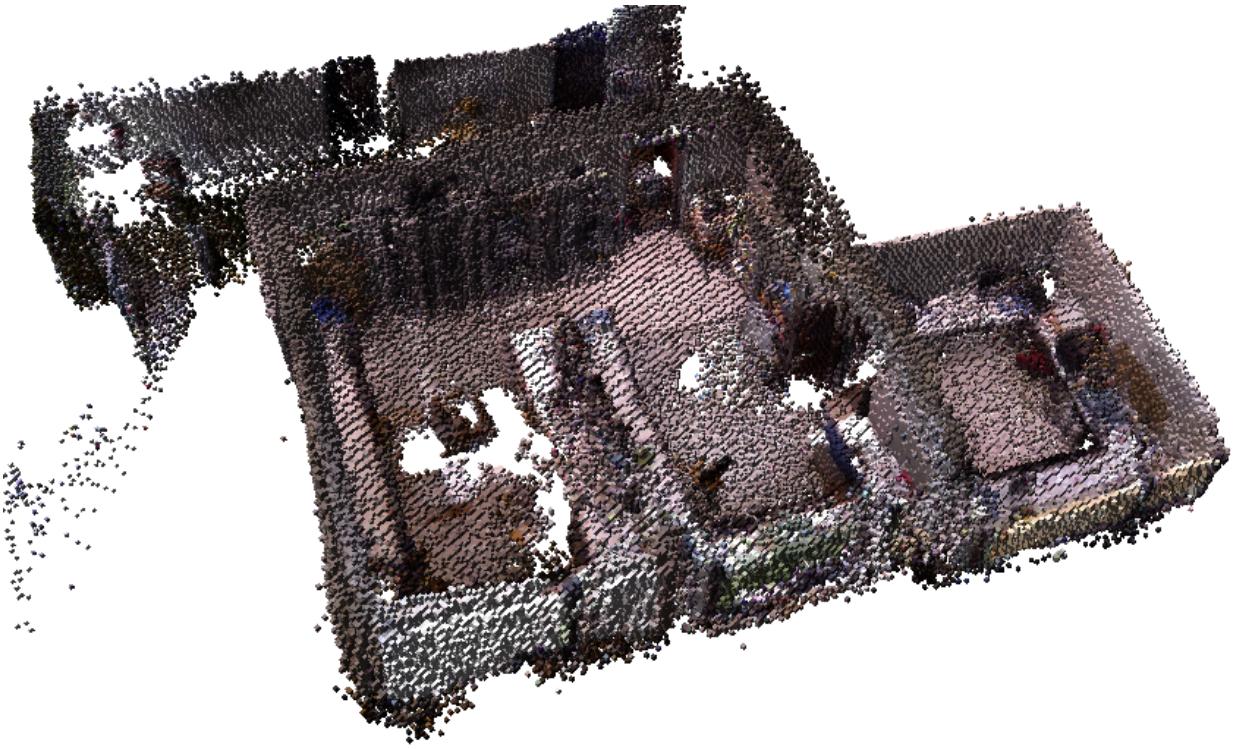

Figure 1: OctoMap of rooms V315, V317 and a part of the hallway after employing the RGBD SLAM algorithm.

1.2.1 Sidenote

An OctoMap can be converted to a schematic using `https://github.com/idryanov/2schematic`. Such a schematic can be used in a Minecraft editor, and put into a Minecraft world (figure 2)!

Figure 2: OctoMap implemented in Minecraft

1.3 OctoMap

1.3.1 Initial Guess

We will try to update an initial guess model of the environment with actual measurements that were obtained with the Kinect camera. The initial model has to be created and converted to an OctoMap. Afterwards, this OctoMap will be merged with the RGBD SLAM OctoMap. In order to model the initial guess, I utilized OpenSCAD. This is the same 3D modelling software that I used to model the propeller guards for the Erle-Copter. Firstly, the initial guess model was created based on the exact dimensions of the environment that was recorded in my Kinect dataset. Secondly, the model was exported as an `.stl` file. Thirdly, the `.stl` file was converted to a `.binvox` file. This was done with a software package found at `http://www.patrickmin.com/binvox/`. Lastly, the `.binvox` file was converted to an octree with the `binvox2bt` executable that is provided in the OctoMap library. When executing `binvox2bt`, I marked all non-occupied space as free space by adding the `--mark-free` parameter. The resulting can be seen in figure 3.

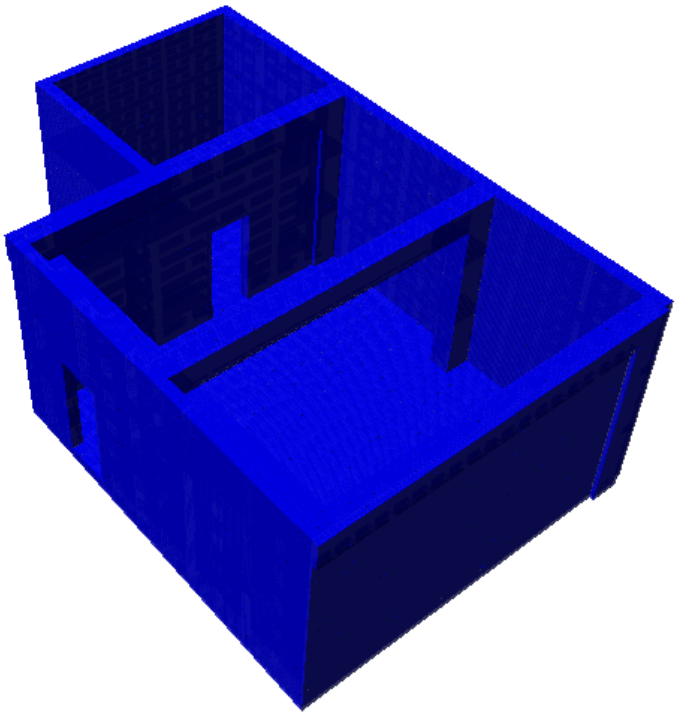

Figure 3: Initial guess OctoMap of room V315 at UA campus Groenenborger.

1.3.2 Publisher

The initial guess model has to be updated with the OctoMap from RGBD SLAM. The first step to achieve this is publishing the initial guess `.ot` or `.bt` file as a message, so that it can be seen in RViz as a ROS topic. For this purpose, I created the ROS package `octomap_merger` that includes C++ code for publishing `.bt` files to a ROS message (`octomap_publisher`).

```
1  #include "ros/ros.h"
2  #include "std_msgs/String.h"
3  #include <octomap/octomap.h>
4  #include <octomap_msgs/conversions.h>
5  #include <octomap/OcTree.h>
6  #include <sstream>
7  #include <iostream>
8
9  void printUsage(char* self){
10   std::cerr << "USAGE: rosrun octomap_merger octomap_publisher
       <InputFile.bt>\n";
```

```
11    std::cerr << "This␣tool␣publishes␣the␣given␣.bt␣file␣to␣an␣
         Octomap␣message\n";
12    exit(0);
13  }
14
15  int main(int argc, char **argv){
16    if (argc != 2)
17      printUsage(argv[0]);
18
19    ros::init(argc, argv, "octomap_publisher");
20    ros::NodeHandle n;
21    std::string inputFile = "";
22    inputFile = std::string(argv[1]);
23
24    //publisher
25    ros::Publisher octomap_pub = n.advertise<octomap_msgs::
         Octomap>("octomap_loaded", 1);
26    ros::Rate loop_rate(10);
27
28    octomap_msgs::Octomap octomap;
29    octomap::OcTree myOctomap(inputFile);
30    octomap_msgs::fullMapToMsg(myOctomap, octomap);
31
32    ROS_INFO("Publishing␣Octree␣to␣ROS␣message");
33
34    octomap.header.frame_id = "/map";
35
36    int count = 0;
37    while (ros::ok()){
38      octomap_pub.publish(octomap);
39      ros::spinOnce();
40       loop_rate.sleep();
41      ++count;
42    }
43    return 0;
44  }
```

As the `fullMaptoMsg` function only supports `.bt` files, I also included `octree_converter`, that can be used to transform `.ot` files to `.bt` files.
When executing `rosrun octomap_merger octomap_publisher initial_guess_models/v315_v317.bt`,

the `v315_v317.bt` file will be published to a ROS message. The result can be seen in RViz (figure 4).

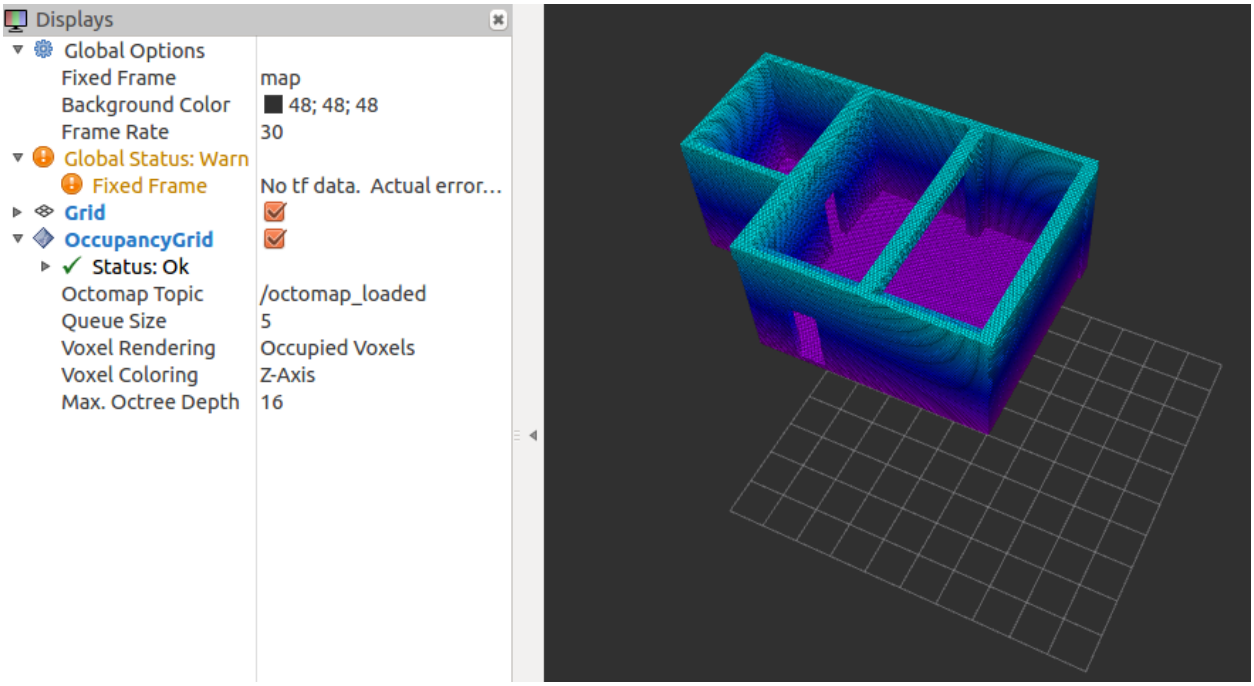

Figure 4: Initial guess model of room V315, published as a ROS message

However, the `octomap_publisher` will need some changes. For now, the OctoMap is published over and over again. This is not a feasible approach if the initial model has to be updated with the RGBD SLAM OctoMap.

1.4 LSD SLAM

Using a HD webcam that runs with the `gscam` ROS driver, I conducted brief indoor tests with Monocular LSD SLAM [1]. In short, the LSD SLAM algorithm seems to render an accurate trajectory but the resulting map contains a high amount of noise.
Engel et al recommend the use of a wide field-of-view camera (130 degrees). When such a camera is made available for our research, this will be the our next step in testing LSD SLAM.

2 Planning week 13

- Improving `octomap_publisher`

- Getting the RGBD SLAM OctoMap in RViz
- Merging OctoMaps

References

[1] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. *Computer Vision ECCV 2014*, pages 834–849, 2014.

[2] Felix Endres, Jurgen Hess, Nikolas Engelhard, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the {RGB}-D {SLAM} system. *2012 {IEEE} International Conference on Robotics and Automation*, 2012.

[3] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. {OctoMap}: an efficient probabilistic {3D} mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2 2013.