Registration of lidar data and analysis of a 3D point cloud

Group 9
Panteleimon Manouselis, p.manouselis@student.utwente.nl, s3084493, M-ROB
Mrinal Magar, m.b.magar@student.utwente.nl, s2689529, M-S,C
Yifan Cheng, y.cheng-2@student.utwente.nl, s3072517, M-ITech
Koen Freriks, k.freriks@student.utwente.nl, s2014637, M-ME

Registration of lidar data and analysis of a 3D point cloud

I. Introduction

In this report the registration of lidar data to construct and analyse a 3D point cloud is described. For this, an experiment was executed where a pedestrian walked a certain path in a hallway while wearing a backpack. The backpack was equipped with one IMU and three Lidars, all with different orientations.

In the following sections it will be discussed how the data is processed and transformed in a 3D point cloud. In section II the methodology is described. After, the results are presented in Section III

II. Methodology

The first step is to obtain the lidar data from the rosbags. The data is transformed from polar coordinates (r, θ, z) to carteesian coordinates (x, y). These coordinates are still in reference to the frames of the lidars $(s_1 \ and \ s_0)$. Therefore, they are transformed to the body frame using the following procedure.

$$P_b^0 = {}_{s_0}^{b}H * P_{s_0}^0$$
$$P_b^1 = {}_{s_1}^{b}H * P_{s_1}^1$$

At this point, all the lidar data is in the body frame. However, since the frame is moving the data needs to be registered w.r.t. the world frame. For this, the first scan of the IMU is chosen, denoted by f_1.

In the previous assignment a time based trajectory was obtained. Now, this trajectory can be used to place the lidar points in the world frame. The trajectory from part B is obtained and augmented with time steps using interpolation. Since the data was in 2D, $z = 1.5m$ is assumed. First the data is transformed to the first frame f_1 using the following equation.

$$P_{f_1} = {}_{b}^{f_1}H * P_b \tag{1}$$

The only thing left to be done is transforming the data points to the correct position based on the position and orientation of the pedestrian at that time step. For this, using the trajectory from part B, homogeneous matrices are constructed for every time step. This is done by comparing timestamps and calculating the platform orientation in 2D for the current position and the 1st frame position. It is good to note that as a result of the assumption of a 2D trajectory, we lose the possible rotations around the x and y-axis. What's more, the construction of the matrix is not the same as what was suggested in the instruction. The effect of the assumption and the choice of transformation matrices are commented on in Section III.

Now the data points can be transformed into their correct frames using the following procedure.

$$P_{f_N} = {}_{f_1}^{f_N}H * P_{f_1} \tag{2}$$

This will result in a 3D point cloud of the measured Lidar data.

III. Results

In this section the results are presented. Using the procedure in Section II a 3D point cloud of the relevant hallway was found. In Figure 1 a top view of the cloud can be seen, containing the combined data of scan 0 and 1. This was plotted using CloudCompare. In Figure 2 the side view of the 3D cloud is presented.

IV. Discussion

From Figure 1 it can be seen that to some extend the point cloud matches the shape of the hallway. The circular path can clearly be separated from the starting point. This indicates that transforming the points from $(s_0 \ and \ s_1)$ to the world frame f_N was successful and no major mistakes were made. However, both Figure 1 and 2 show a great number of the points that do not perfectly coincide with the hallway. The points falling outside of the hallway on the right and top-right of Figure 1 can be explained due to the fact that there are doorways present here, but this does not account for all the irregularities in the figure. In Figure 2 it can be seen that a lot of points fall outside as well, namely on the top and bottom of the floor and ceiling.

It is believed that this is to some degree caused by the loss of 3D in 2D trajectory data. It is true that the path is fully horizontal, but we can not guarantee the pedestrian walks in a straight-up manner all the time. Therefore even if

Fig. 1. Top view of the 3D point cloud (scan0 and scan1)

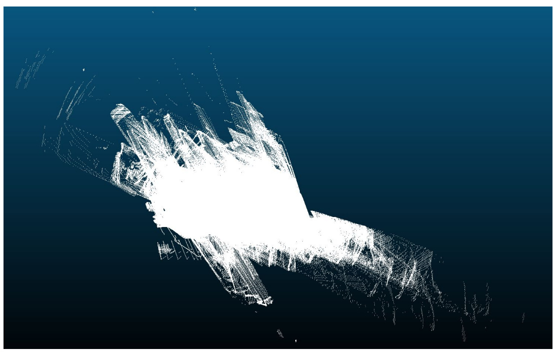

Fig. 2. Side view of the 3D point cloud(scan0 and scan1)

the z-coordinate changes minimally, the oscillations remain. However, we only generated 2D trajectory in Part B. As a result of this, the orientations around the x and y-axis are lost. This would explain the fact that the results do match the hallway to a certain level, but still contains irregularities. Therefore, it is believed that using a 3D trajectory instead of a 2D Trajectory would definitely improve the results.

Augmenting the trajectory data using the nearest neighbour interpolation method is not perfect as well. In our case, the number of scans is more than the number of trajectory data we produced in Part B. This means when doing the interpolation, some of the scans which actually collected in different positions will share the same trajectory coordinates. We further improved this by using linear interpolation, but the results were not much improved. Additionally, we also tried to construct the point clouds with fewer points, such as using only 1/10 of the points by skipping nine points each time. The shape of the point clouds has no evident change.

During the experiment, we also noticed the problem of error accumulation. It is instructed in the guide that by combining the successive transformations, the transformation for every separate scan can be found using the following equation:

$$H_{f_N}^{f_1} = H_{f_2}^{f_1} H_{f_3}^{f_2} \dots H_{f_N}^{f_{N-1}} \tag{3}$$

However, when we calculated the platform orientation in 2D from the two nearest neighbouring frame positions of the trajectory, we noticed that the change between the two points was too small for the rotation to be obvious. As shown in Figure 3, the calculated rotations are not fully aligned with the actual trajectory. Such small rotation errors in each step were accumulated and magnified during the chain of the transforms. It was hard to get a reasonable result based on this. Therefore, instead of using the two nearest neighbouring coordinates, we use the current coordinates and the 1st frame coordinates to calculate the transformation directly.

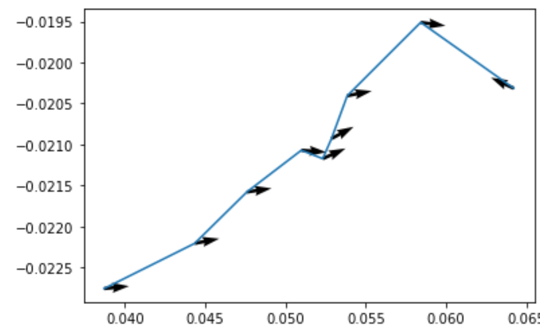

Fig. 3. the calculated orientation on each point in part of the trajectory (orientation is shown in black arrow, and the trajectory is shown in blue line)

Other improvement suggestions to solve this error accumulation problem include limiting the random orientation changes in trajectory by setting a threshold. We can also introduce keyframes to calculate the orientation shift between the keyframes to suppress the error accumulation in a window of a certain number of points.

Finally, the detection of doors is only possible in case the doors are opened. The door can be detected as a location surrounded by a continuous flat wall (so not a corner) where no wall is detected. Detection of closed doors could definitely be possible with this setup. Lidars can detect the recognizable rectangular shape of the door, but this would require better results than obtained in this case because more detail is required.