3D Point Cloud Filtering

Yuncheng Yuan(1778307)

January 22, 2023

1 Point cloud importing

1. The video that used to generate this point cloud was recorded from ZED Mini (a stereo camera). Recall what you learned in the lecture, could you briefly describe the process of point cloud generation using a stereo camera?

   Ans:

   (a) According to the default parameters of stereo camera, prepare to calculate the alignment of images from camera.

   (b) Detecting the corners and edge from the two images. After detecting, start to find the corresponding points.

   (c) Calculate the position of items by calculating the intersections of rays in the corresponding from camera.

   (d) Generate the point cloud by the 3D coordinate.

2. As you can see in RVIZ, this is a very noisy point cloud. Could you explain the reason why there are so many noises?

   Ans:

   (a) Poor lighting conditions: If the lighting conditions are poor, it can make it difficult to detect and match features in the images, leading to more noise in the point cloud.

   (b) Camera calibration: Poor camera calibration can lead to inaccuracies in the triangulation step, resulting in more noise in the point cloud.

   (c) Algorithm robustness: The quality of the point cloud will depend on the robustness of the algorithm used for feature detection, matching, and triangulation, if the algorithm is not robust or it is not well implemented, it will generate more noise.

   (d) Environmental factors: Reflections, shadows, and other environmental factors can make it difficult to detect and match features in the images, leading to more noise in the point cloud.

   (e) Sensor noise: The sensor itself can introduce noise, for example, the stereo cameras can have noise in the image due to sensor characteristics.

3. Before designing the filtering node, your node first needs to subscribe to the point cloud topic. Recall what you have learned from the "listener.py" example. Try to create a node that subscribes point cloud topics.

   Ans: After running the talker.py, the listener will should the message from talker.py. Every time the talker.py upload one message, the listener.py will receive one same message.

   create a node that subscribes point cloud topics

2 Design your filtering node in ROS

1. Design your filtering node and record your filtered point cloud in a .bag file. Don't forget to show images of the complete filtered point cloud in the report.

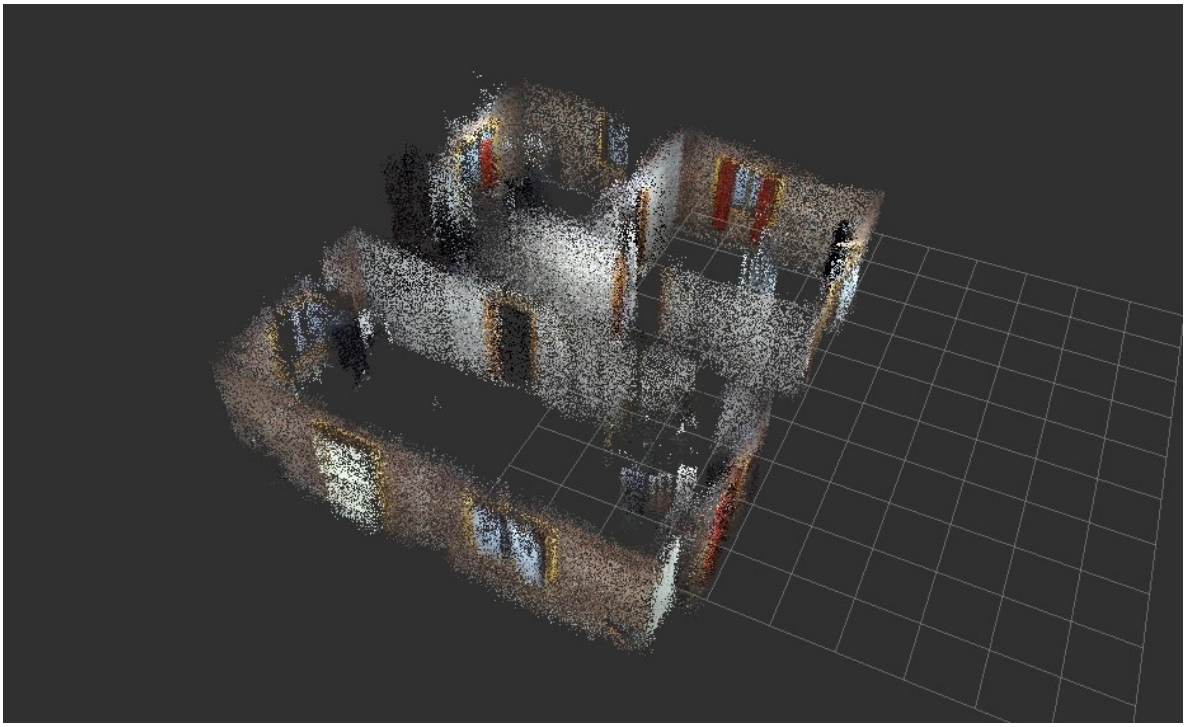

Figure 1: The Image generated by ROS

Ans:

.bag file hyperlink: https://tuenl-my.sharepoint.com/:f:/g/personal/y_yuan_student_tue_nl/Eh-p-RftOnRIsiIe3d8cLYUB5GG5_N3eb5___1Rk_lrKYg?e=Pe5BKG

2. Explain details of your filtering procedure. You may also draw a flow chart to support your text.

   Ans:

   The filter is to generate to a raw point cloud by first creating a 2D grid using a provided step size. It then applies a Gaussian filter to the grid using a provided sigma value. The function then calculates the mean of the filtered image, and sets all values in the image that are lower than the mean to 0. It then sets all non-zero values in the image to 1. The function then loops through the grid, and for each 'white square' (a square with a value of 1), it appends the corresponding points in the raw point cloud to a filtered point cloud, which is then returned.

3. Do you have any idea on how to further improve your filtering node in terms of speed and performance? Explain your idea here.

   Ans:

   (a) change the step length of Gaussian. To get the higher speed of generating, we could increase the step length. although the image will get a little blur, the speed of generating image will get faster.

   (b) The Mean value of Gaussian filter could be larger and get better image.

   (c) Implementing CPU acceleration: Point cloud processing needs a lot of computing by CPU. The result can be better by using a high performance CPU.

   (d) Use a better data structure: Use another data structure(like: octree) can be used to organize the point cloud in a hierarchical manner and accelerate the searching and processing of points.

   (e) Adapt another language. Using python will spend a lot of time on generating image. Use C++ will become faster and better result.

   (f) Using more advanced filter: Use neural networks to process point clouds and can be more efficient than traditional filter.

3 Reference

https://github.com/langestefan/5LSH0_final_assignments/tree/main/assignment_3_point_cloud