CS284: Simultaneous Localization and Mapping

Homework 5, Fall Semester 2021

- Contact: lkneip@shanghaitech.edu.cn
- TA: cuili@shanghaitech.edu.cn
- Submission deadline: **23:59, 30th of November 2021** (please submit a zip via email)
- For general questions not directly related to the homework material, please first use Piazza.
- **Please reread our policies on academic integrity. It is a very serious matter, and any violations will be prosecuted to the fullest extend.**

Task Description

In Homework 5 you will reconstruct a cute bunny.

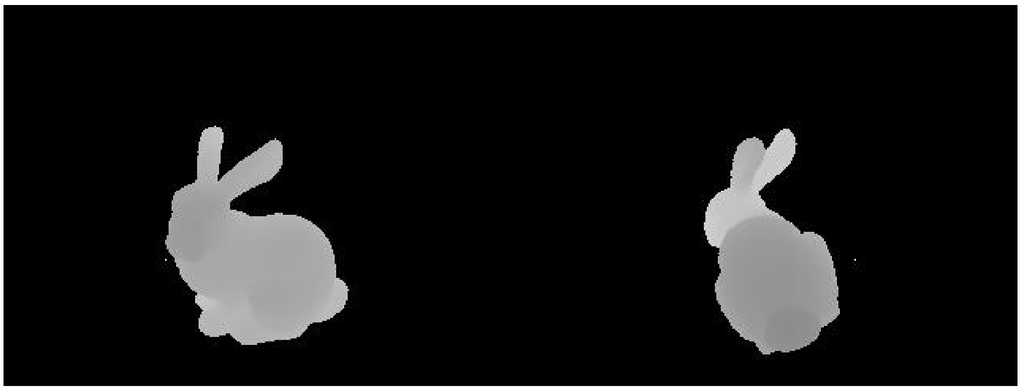

Fig 1: examples of bunny scans

- There are 14 scans in total. As Fig1 shows, each scan takes the form of a uint16 depth image with the resolution of 320*240, described in the local coordinate system of the camera(Note that the unit of depth values should be "meter", and the camera matrix is:

$$K = \begin{bmatrix} 259.2 & 0 & 160 \\ 0 & 259.2 & 120 \\ 0 & 0 & 1 \end{bmatrix}$$

  You should divide each non-zero pixel value by 1.3476×10^5 to get the true depth value.

- The folder "pose" include each camera pose in the world frame. The values are stored as 4*4 transformation matrix.

- The bunny is roughly 7.5 inches high in the real world, which means that you can put all of the bunny points into a bounding box with a side length of 0.2m. You can divide this bounding box into cubes with side length of 0.01m, which means that your tensor will have 8000 voxels in total. You are free to choose a finer sampling grid, ideally it is left as a variable in your implementation and you can generate more fine-grained results once your code is finalized. The above is only the minimum resolution.

- For each depth scan location, calculate the truncated signed distance field as taught in class (Lecture_15). Choose a suitable truncation distance, and use the camera transformation parameters to generate the distance field.

- Fuse the truncated signed distance fields as taught in class (you may choose the incremental method).

- With the final sdf values, try to reconstruct the mesh using marching cubes or any other algorithm that can transform an sdf into a mesh.(Hint: in python, you can simply use *skimage.measure.marching_cubes_lewiner* function）

Submission

Please submit a PDF report (not more than 3 pages!) along with the code to the TA email address before the deadline (put professor in CC). Your code and results should be the same with what you submitted in the email, then arrange for a time with the TA to demonstrate your code. Your report must include:

- A description of your implementation. This should cover a basic description of the algorithm, especially how do you calculate the sdf.
- It should also include the physical structure of the program (i.e. file description, dependencies), and instructions on how to compile and use it (i.e. how to interpret the result).
- Figures that show the results of your reconstruction. You should at least show reconstruction results from the **top, bottom, left, right, front** and **rear** directions. For example, you can write your reconstruction results into .ply file and visualize them using *pcread* function in *matlab*, or just open your reconstructed .ply/.obj file in the *MeshLab* software.
- Observations on the performance of your algorithm, both in terms of efficiency and in terms of accuracy. And thoughts on how to improve the results.

Please use ”SLAM HW5 – Your name (Chinese Pinyin, e.g. Wang Xiaoming)” as the email-header when sending the email. Please pack all the files into a zip file, the structure should be:

[Your name] folder
– Your name.pdf
– [code] folder