[Spring 2024] CS172 Assignment 2
Basics in computer vision

#NAME #StudentID

May 13, 2024

Acknowledgements

1. Deadline: **2024/6/11 23:59:00**. Late Policy please refer to the course slides.
2. Please submit your assignment in **Gradescope** in PDF format. Registration code: 4GNJPE
   - Giving your report in English, a report in Chinese is not accepted.
   - Handwritten homework is not accepted and we highly recommend using LaTeX. The LaTeX template has been uploaded to Blackboard.
   - Note that you **MUST** select pages for each question on Gradescope, and TA will **NEVER** select pages for you. Otherwise, you will lose **ALL** the points.
3. Please upload your code zip to the **ShanghaiTech cloud** disk.
   - `https://epan.shanghaitech.edu.cn/l/zF2Iu6`
   - All source files and *readme* should be included but remember to remove the datasets.
   - Your zip should be named as CS172_NAME_ID_hw2.zip.
4. Plagiarism or cheating is strictly prohibited.
   - DO NOT share your assignment or code!
   - NO fake solution is allowed!
   - Make sure your codes can run and are consistent with your solutions.

1 [10 points] Understand Point Cloud

A point cloud is a discrete set of data points in space. The points may represent a 3D shape or object. Each point position has its set of Cartesian coordinates (X, Y, Z). The point cloud is one of the most important and commonly used representations for object and scene shapes. In this section, you are required to visualize and plot at least three point clouds in the report. Please use the point cloud source we offered in the compressed package. You are free to use any third-party libraries like matplotlib or open3D for this section.

2 [20 points] Implement Iterative Closest Point Algorithm

Point Cloud Registration is a fundamental problem in 3D computer vision and photogrammetry. Given several sets of points in different coordinate systems, the aim of registration is to find the transformation that best aligns all of them into a common coordinate system. Point Cloud Registration plays a significant role in many vision applications such as 3D model reconstruction, cultural heritage management, landslide monitoring and solar energy analysis.

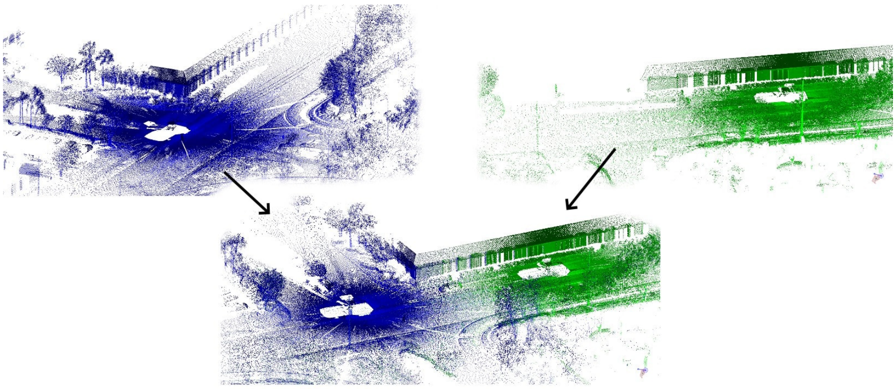

Figure 1: An example of point cloud registration

In this part, you are required to implement iterative closest point (ICP) algorithm to find a rigid transformation such that, given two finite size point clouds $\{P, Q\}$, the difference between point cloud Q and the transformed point cloud P is minimized. The ICP algorighm can be roughly breaked down into (a) finding a rigid transformation given two point clouds with **same number of points** and (b) Fixing point cloud Q and iteratively choosing a subset of point cloud P to compute rigid transformation untill convergence.

Algorithm 1 ICP algorithm

```
Input: source point cloud $p = \{p_i\}^{n_p}$, target point cloud $q = \{q_i\}^{n_q}$, threshold $\epsilon$;
Output: A rigid transformation: rotation $R$ and translation $t$;
  $\bar{p} \leftarrow \mathbf{MEAN}(P)$,   $\bar{q} \leftarrow \mathbf{MEAN}(Q)$;
  $\hat{p_i} = p_i - \bar{p}$,   $\hat{q_i} = q_i - \bar{q}$;
  $H = \sum_{i=1}^{n_p} \hat{p_i}\hat{q_i}^T$
  while True do;
      $\hat{q}^{'} = \hat{q}.nearestNeighbor(\hat{p})$          ▷ To find a match and align the pcd size.
      $H = \sum_{i=1}^{n_p} \hat{p_i}\hat{q_i}^T$
      $USV^T = SingularValueDecomposition(H)$
      $R^* = VU^T$
      $t^* = \hat{q}^{'} - R^*\hat{p}$
      $\hat{p} = R^* \cdot \hat{p} + t^*$
      if Change of $R$ and $t$ is smaller than threshold $\epsilon$ then break;
      else if The loss $\sum_{i=1}^{n_p} ||(R \cdot p_i + t) - q_i||^2$ is smaller than threshold $\epsilon$ then break;
      else if Iteraction times exceed the threshold then break;
```

You are required to implement the ICP algorithm from scratch, and third-party libraries can only be used for mathematical computation and visualization. You should visualize at least three pairs of point clouds in your report, and the source and target point cloud should have different colors, as in Figure 1.

Hint: KD tree is very useful to speed up the algorithm, try it out! (Scipy.spatial.KDTree is a good choice.) If you think the number of points in the point cloud is still too large, you may consider to use some downsample methods like farthest point sampling.

3 [20 points] ICP with RANSAC

ICP algorithm does not take noisy points into consideraction, and RANSAC is known to be effective in relieving this problem. Please use the RANSAC to reimplement the ICP. More precisely, you should use RANSAC to calculate the rigid transformation in each iteraction of ICP algorithm.

Similarly, third-party libraries can only be used for mathematical computation and visualization. You should visualize at least three pairs of point clouds in your report.