Computer Vision AI – Assignment 1
Iterative Closest Point - ICP

April 4, 2017

Students should work on the assignments in groups of two. Students are supposed to work on this assignment for three weeks. Some minor additions and changes might be done during these three weeks. Students will be informed for these changes via blackboard. The analysis and the conclusions must be included in a report. A final report and source code should be zipped and sent to dropbox account `https://www.dropbox.com/request/AAFdqvtd1keMBABi9Q4D` 24-04-2017, 23:59 (Amsterdam time). The submitted zip file should be named with authors' surname.

Any questions regarding to the assignment content can be discussed on Piazza `https://piazza.com/university_of_amsterdam/spring2017/52042cov6y`. Please use this access code (52042COV6Y) during registration.

README file attached to this assignment contains the information about the provided data.

1 Iterative Closest Point - ICP

ICP algorithm is first introduced by Besl and McKay (See [2] for details). Considering two point-clouds A_1(base) and A_2(target), ICP tries to find a spatial transformation that minimizes the distance (e.g. Root Mean Square (RMS)) between A_1 and A_2 which can be defined as

$$RMS(A_1, A_2, \psi) = \sqrt{\frac{1}{n} \sum_{a \in A_1} \|a - \psi(a)\|^2} \quad , \tag{1}$$

$$\min_{\psi: A_1 \to A_2, t \in \Re^d} \sum_{a \in A_1} \|Ra - t - \psi(a)\|^2 \quad . \tag{2}$$

where R and t are the rotation matrix and the translation vector in d dimensions, respectively. ψ is one-to-one matching function that creates correspondences between the elements of A_1 and A_2. R and t that minimize above equation are used to define camera movement between A_1 and A_2.

1.1 Implementation

In this assignment, students will implement the ICP algorithm and some of its variants to estimate the camera poses for given point-clouds (#####.*pcd*). First, the basic ICP algorithm will be implemented:

1. Initialize $R = I$ (identity matrix), $t = 0$.
2. Find the closest points for each point in the base point set (A_1) from the target point set (A_2) using brute-force approach.
3. Refine R and t using Singular Value Decomposition (See [1], Chapter 15, for details).
4. Go to step 2 unless RMS is unchanged.

There are different ways to improve the efficiency and the accuracy of ICP. Some of these techniques are discussed in [3]. In this assignment, students should also analyze various aspects such as accuracy, speed, stability and tolerance to noise by changing the point selection technique. Using all the points, uniform sub-sampling, random sub-sampling in each iteration and sub-sampling more from informative regions can be used as the point selection technique. Students are expected to implement these variants and report their findings in the final report.

Note: You can use the provided data (source.mat and target.mat) to test the correctness of the estimated transformation matrix.

2 Merging Scenes

2.1

Estimate the camera poses using each two consecutive frames of given data (ICP implemented in Section 1.1). Using the estimated camera poses, merge the point-clouds of all the scenes into one point-cloud and visualize the result. Does the merging produce sufficient result? Discuss why.

Now, estimate the camera pose and merge the results using every 2^{nd}, 4^{th}, and 10^{th} frames. For instance, in case of choosing the frame sampling rate to be 2 and given N frames ($frame_1$, $frame_2$, .., $frame_N$): First, you will need to estimate the camera pose of $frame_3$ (target) from $frame_1$ (base) and use it in merging $frame_1$ with $frame_3$ to get $frame_{1,3}$. Next, estimate the camera pose of $frame_5$ (target) from $frame_3$ (base) and use it to merge $frame_{1,3}$ with $frame_5$ to get $frame_{1,3,5}$ and so on until reaching $frame_N$. Does the camera pose estimation change?

2.2

Iteratively merge and estimate the camera poses for the consecutive frames (point-clouds). Supposed there are N frames ($frame_1$, $frame_2$,. . ., $frame_N$). First, estimate the camera pose of $frame_2$ (target) from $frame_1$(base). Next, merge $frame_1$ and $frame_2$ into $frame_{1,2}$. Then, estimate the camera pose of $frame_3$(target) from $frame_{1,2}$(base) and use it to merge $frame_{1,2}$ and $frame_3$ into $frame_{1,2,3}$ and so on until reaching the final frame ($frame_N$). Do the estimated camera poses change in comparison with the previous estimates (Section 2.1)? Does this estimation produce better results?

3 Questions

1. What are the drawbacks of the ICP algorithm?
2. How do you think the ICP algorithm can be improved, beside the techniques mentioned in [3], in terms of efficiency and accuracy?
3. Feel free to improve the results of ICP using your suggestions to above questions.

References

[1] J Andreas Bærentzen, Jens Gravesen, François Anton, and Henrik Aanæs. *Guide to computational geometry processing: foundations, algorithms, and methods*. Springer Science & Business Media, 2012.

[2] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14:239–256, 1992.

[3] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.