CSE 291 Assignment-2 (Problem 3)

Saqib Azim
sazim@ucsd.edu
Benchmark Username: sirius1707

1 Method

The objective in this problem is to estimate the poses of objects in the test dataset using training data. My approach relies on the fact that we have objects with same ids as test data in the training data and we know the ground-truth pose information for the train dataset. The overall procedure can be described in following points.

- Extract all the file names in the training, test and validation data.
- Take the training data, iterate over all the scenes in the training data. For each instance of an object, extract the image coordinates in the pixel coordinate frame (PCF) using the label segmentation image.
- Using the intrinsic matrix and the depth image provided for each scene, convert each object's pixel coordinates from PCF to camera coordinate frame (CCF).
- Using the extrinsic matrix provided for each scene, transform these points in the CCF to world coordinate frame (WCF). The 3D points obtained in the WCF are then scaled by multiplying with the inverse of the scaling factor provided for each object in the scene. Now, the obtained 3D points in the WCF are rotated and translated version of their canonical form.
- We transform the points in the WCF to their canonical original form by multiplying with the inverse of the ground-truth pose provided for every object in the scene in training data.
- For each of the 79 objects present, we store their various canonical original point cloud from different scenes in a list. This acts as a dictionary later when we estimate the pose of an object in the test scene data.
- In order to speed up the training and test process, we stored the training, validation and test data in separate pickle files in a one-time computation. Later, we reload train and test pickle files during training extraction and testing and use these to load the data. This saves significant I/O time as only once we have to load images (depth, label and metadata) one-by-one directly from the memory.
- During test, we want to estimate the 6D poses of each object in all the 200 test scenes. For each object in a test scene, we follow similar process as for the training data and estimate their 3d coordinates in WCF from their 2d pixel coordinates using intrinsic, extrinsic matrix and scale factor.
- Now, the sad part is we do not have ground-truth pose information (transformation matrix from their canonical original form to rotated + translated form) for these point clouds. Here we use the list of partial point clouds for each object that we created during train process.
- For each object, there exists a list of partial point clouds created during training. We use Iterative Closest Point (ICP) algorithm to estimate the transformation between test point cloud (referred as target) and each of the training point clouds (referred to as source).
- For initial transformation provided to the ICP algorithm, we used the ground truth pose for the source point cloud in the training data. The threshold value provided to the ICP is a hyperparameter. We used thresh_value = 0.001, 0.008, 0.01, 0.02, etc.

- The test process is quite exhaustive as it requires performing ICP between every object's point cloud in a test scene and a list of train point clouds containing 1000's of point cloud for each object. We use ICP implementation from Open3D library as it is fast and much more efficient as compared to my own implementation.
- For time reduction and accuracy improvement (to some extent), we have uniformly downsampled the test point cloud and training point clouds and then performed ICP. The downsample factor is a hyperparameter and we tested with downsample = 4,8,16,32,64. As expected, with 64, speed is fast but accuracy is poor whereas with downsample=4, the accuracy is high but it's super slow on a CPU.
- We used the fitness parameter provided by the Open3D's ICP to find the best point cloud (maximum fitness) among the list of training point clouds.

2 Experiments

Results on Validation dataset:

With ICP thresh = 0.001, DOWNSAMPLE = 32, RRE = 0.7291051 and RTE = 0.00114901
With ICP thresh = 0.01, DOWNSAMPLE = 4, RRE = 0.521865 and RTE = 0.00100371
With ICP thresh = 0.01, DOWNSAMPLE = 16, RRE = 0.6926190 and RTE = 0.00253707