**Registration Benchmark run\_experiments.py Manual**

1. Program Overview

The run\_experiments.py script is used to conduct point cloud registration experiments. This script contains functions to run a variety of registration algorithms on selected datasets, with various inpparameters such as voxel size and transformation ranges, and to measure the performance of these algorithms. The experiments can be performed on multiple datasets using various algorithms, and the results are saved in a text file.

2. Function Descriptions

Main function:

run\_all\_experiments(dataset\_names: list, algorithms: list, voxelsize: float, range\_t: float, range\_r: float, overlap: float)

This is the main function to run the experiments on multiple datasets using multiple algorithms. It checks if the metrics file for a particular dataset and algorithm already exists; if it does, it skips that experiment, otherwise, it calls the **main\_run\_experiments()** function to execute the experiment.

main\_run\_experiments(algorithm, dataset\_name: str, voxelsize: float, range\_t: float, range\_r: float, overlap: float, metrics\_file\_path: str)

This function is used to run the experiments on a specific dataset using a particular algorithm and to save the results in a text file.

Auxilliary Functions:

run\_single\_experiment(algorithm, pcd0: o3d.geometry.PointCloud, pcd1: o3d.geometry.PointCloud, ground\_truth\_transform: np.ndarray, voxelsize: float)

This function is used to conduct a single registration experiment, i.e., to register two point clouds using a specified algorithm. It returns the evaluation metrics and the estimated transformation matrix.

run\_algorithm(algorithm, source: o3d.geometry.PointCloud, target: o3d.geometry.PointCloud, voxelsize: float)

This function is used to run the registration algorithm and measure its runtime. It returns the runtime, the transformation matrix estimated by the algorithm, and the number of points in the downsampled point clouds.

3. Imports and Dependencies

* torch: PyTorch library for neural network applications and tensor computations.
* open3d as o3d: A modern library for 3D data processing.
* numpy as np: The fundamental package for scientific computing with Python.
* time: Provides functions for working with times, and for converting between representations.
* os: Provides functions for interacting with the operating system.
* pathlib.Path: Represents the system’s path.
* utils.load\_point\_cloud: A utility function for loading point cloud data.
* utils.save\_transformations: A utility function for saving transformation matrices.
* utils.draw\_registration\_result: A utility function for drawing the registration result.
* utils.calculate\_total\_points: A utility function for calculating the total points in a pair of point clouds.
* transformation.random\_transformation\_matrix: A utility function for generating random transformation matrices.
* evaluation\_metrics.get\_full\_evaluation\_metrics: A utility function for getting a full set of evaluation metrics.
* evaluation\_metrics.save\_full\_eval\_metrics: A utility function for saving a full set of evaluation metrics.
* registration\_algs.downsample\_point\_cloud: A function for downsampling a point cloud.
* inspect: A module for introspecting live objects such as modules, classes, instances, functions, and code.

4. Operation/Usage

Invoke the function **run\_all\_experiments** with the required parameters. This function will run the specified algorithms on the given datasets. The results will be saved in a text file.

5. Output and Interpretation

The main output of this script is a set of text files containing the metrics for each experiment. These files are stored in the **./results** directory. The metrics include the runtime of the algorithm, the transformation matrix estimated by the algorithm, and various evaluation metrics. After all the experiments have been completed, a message will be printed to the console indicating that the experiments are complete.