**Registration Benchmark evaluation\_metrics.py Manual**

1. Program Overview

This module, **evaluation\_metrics.py**, provides utilities for computing various evaluation metrics used in point cloud registration. These metrics help in evaluating the performance of point cloud registration algorithms. It also provides functions to save these metrics to a file.

2. Function Descriptions

Main function:

get\_full\_evaluation\_metrics(pcd0: np.ndarray,

pcd1: np.ndarray,

ground\_truth\_transform: np.ndarray,

estimated\_transform: np.ndarray,

runtime: float,

downsampled\_points: int) -> dict:

This function is the primary entry point for calculating all the evaluation metrics. This function computes several evaluation metrics for a pair of point clouds. It returns a dictionary containing all of the calculated metrics.

save\_full\_eval\_metrics(metrics\_list: list,

metrics\_file\_path: str) -> None:

This function saves the evaluation metrics to a .txt file.

Auxilliary Functions:

The module includes several other internal functions for computing specific evaluation metrics such as RMSE, MAE, RRE, RTE, and a combined error metric.

get\_rmse(pcd: np.ndarray,

T\_pred: np.ndarray,

T\_gt: np.ndarray) -> float:

**get\_rte(), get\_rsme(), get\_mae(), and calculate\_error()**

Each function takes two transformation matrices as inputs: the estimated transformation and the ground truth transformation.

**point\_translation\_error()**

This function is used to calculate the translation error for a specific point between the estimated and ground truth transformations. It takes three parameters:

* **T\_pred** (np.ndarray): The predicted transformation matrix.
* **T\_gt** (np.ndarray): The ground truth transformation matrix.
* **point** (np.ndarray): The point to be transformed.

It returns a float representing the translation error for the point.

**read\_metrics\_file()**

This function is used to read a metrics file and return the processed metrics. It takes one parameter:

* **file\_path** (str): The path of the metrics file.

3. Imports and Dependencies

* Python 3.6+
* NumPy
* Open3D
* OS, Pathlib and Json Python libraries

4. Operation/Usage

To use this script, import the necessary functions and call **get\_full\_evaluation\_metrics()** with your data. Here's a simple example:

5. Output and Interpretation

The program displays the message: "Analyzing experiment results..." upon completion of experiments. The results are displayed and saved to the specified file. Heatmaps of the results are also generated and displayed.

Notes

* The code assumes that the ground truth and estimated transformations are 4x4 transformation matrices.
* If either the estimated or ground truth transformation is None, the get\_full\_evaluation\_metrics function will skip the metric calculation.
* The get\_full\_evaluation\_metrics function will return None if it is unable to calculate the metrics.
* The code uses the Open3D library for several operations such as transforming point clouds. Ensure Open3D is properly installed and working.