3D Registration Pipeline

Ferran Roure
froure@eia.udg.edu

Last update: February 2014

Introduction

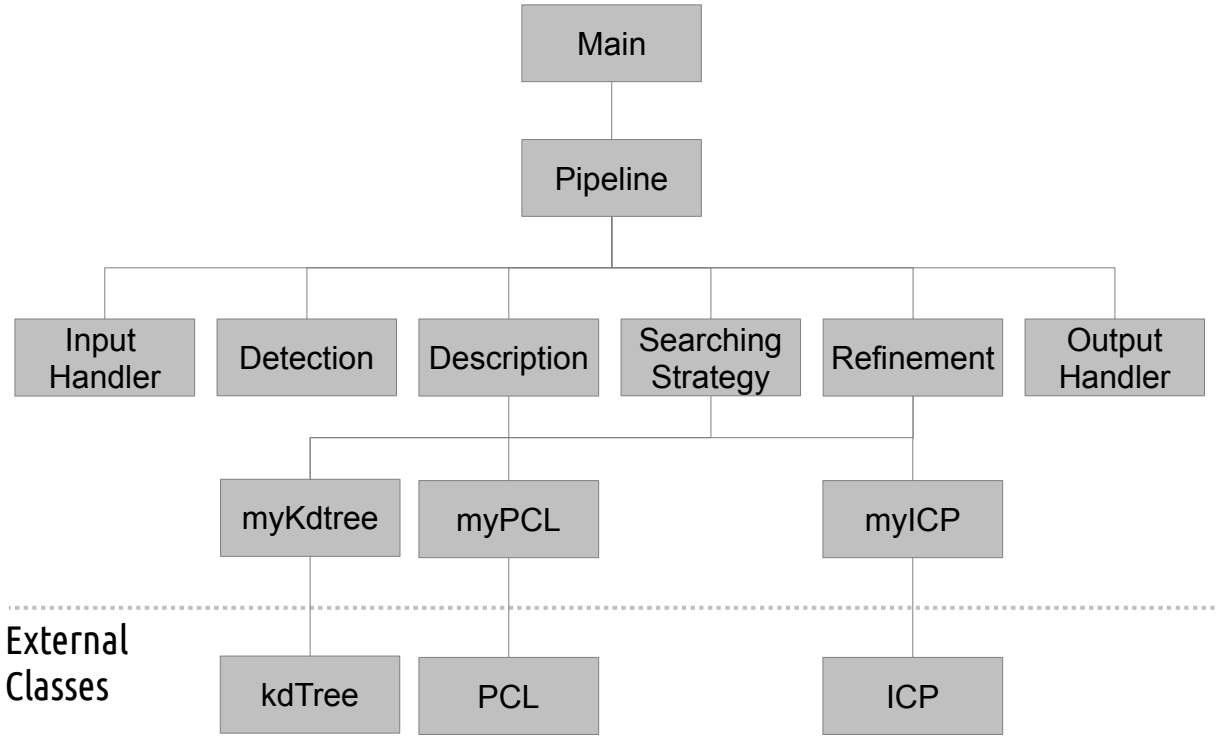
This project is a C++ framework that encompasses all registration steps. Our pipeline is prepared to test different methods in each steps of the pipeline. Six parts are clearly delimited:

- Input Handler — Manage the input data.
- Detection — Feature point detection.
- Description — Feature point description
- Searching Strategies — Search for correspondences.
- Refinement — Fine registration.
- Output handler — Manage output data.

For each of these steps, the user can add his own methods. We recommend to create a bridge class in order to transform the information between user classes and our framework.

Structure

We develop our framework with a clearly defined structure in order to make it as modular as possible.



External Classes

We use different libraries and external classes in order to complete some parts of the registration pipeline.

Kdtree

We used the ANN Approximate Nearest Neighbors library from David Mount and Sunil Arya (http://www.cs.umd.edu/~mount/ANN). A library needs to be installed.

Descriptors

We used the descriptor methods implemented in PCL Point Cloud Library (http://pointclouds.org). A library needs to be installed.

Refinement

We used the Szymon Rusinkiewicz's ICP implementation (http://www.cs.princeton.edu/~smr/papers/fasticp/).

How to run

Our Pipeline is implemented in C++ under Ubuntu Linux distribution. We use cmake (www.cmake.org) to compile our files. Please, check the CmakeFile for depending libraries.
In “build” folder, you can create the runfile using cmake. To run the program you only need to type:

```
user@machine$ ./Pipeline ./paramsFile.txt
```

paramsFile.txt contains the execution parameters for our Pipeline. Is a text file that contains different parameters separated by break lines. The configuration is:

- Use of real data
- input_file
- input_file 2
- auxiliary file
- output_file
- results_file
- Detection method
- Description method
- Searching Strategy method
- Refinement method
- Levels of 3D Grid
- Cells per axis of 3d Grid
- % of points used to calculate the Resiude
- Multiplication factor for MMD for error thresholding
- % of noise for ground truth calculation
- Radius for normal search (description step)
- Radius for descriptor search
- Use of detection step
- Use of description step
- Use of Searching Strategy step
- Use of refinement step