• Coursework 2 - Conway's Game of Life •

Sebastiano Ferraris SN: 14108168 s.ferraris@ucl.ac.uk August 9, 2016

Part 1: Introducing the Game

Part 1: Serial Solution

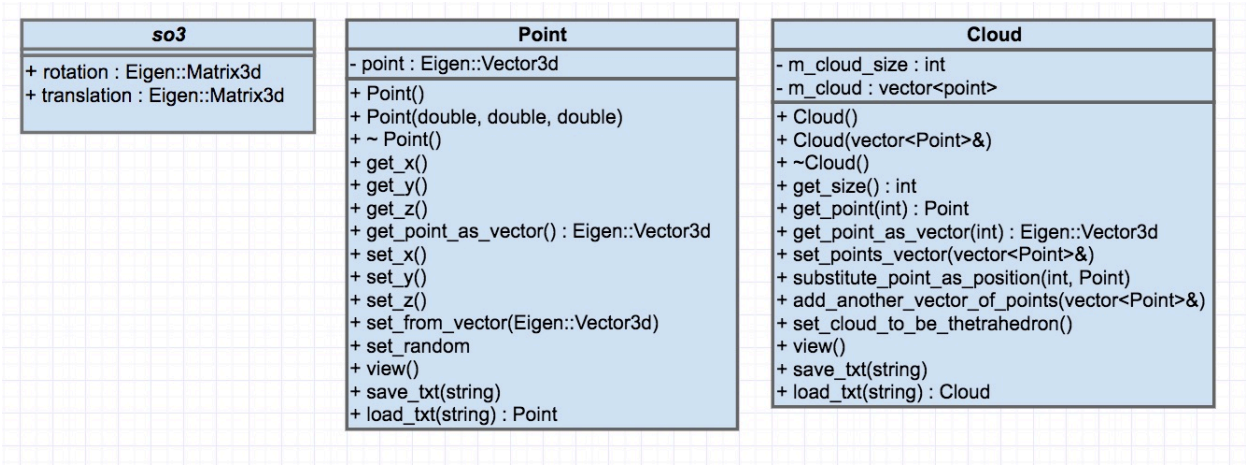

Figure 1: UML diagram with the choice of the classes proposed for the implementation of the AHB algorithm.

Using the proposed framework, I implemented the AHB algorithm [1] in a simple folder structure under *Code/registration_3d_point_sets* divided into *include*, *source* and *utils_python*. *Include* and *source* contains respectively the header files and the source code. There are two .cc files with a main, that produces the actual applications: *see_methods.cc* and *algo_AHB.cc*. These can be called from the terminal under the path *RCCPP-build/bin* in the build folder of the suggested framework. I choose an object oriented pattern, where one C++ structure, called *so3*, and two classes, called *Point* and *Cloud*, are proposed.

Part 2: Parallel Solution

Part 3: Shared Solution

Part 4: Distribuited Solution

Part 5: Accelerated Solution

Extra: between life and dead - a fuzzy game of life

References

[1] Arun, K. Somani, Thomas S. Huang, and Steven D. Blostein. "Least-squares fitting of two 3-D point sets." IEEE Transactions on pattern analysis and machine intelligence 5 (1987): 698-700.

[2] Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes." Robotics-DL tentative. International Society for Optics and Photonics, 1992.

[3] Cho, Youngsang, et al. "A multi-resolution scheme for distortion-minimizing mapping between human subcortical structures based on geodesic construction on Riemannian manifolds." Neuroimage 57.4 (2011): 1376-1392.