## HAAG Weekly Report Week 9 & 10

Steve Foryoung

sforyoung@gatech.edu

Mar 14, 2025

## Time-Log

- What did you do this week?
  - □ Completed v2 of CPD algorithm which uses Statistical Shape Model for registration.
  - $\Box$  Advanced progress on using rmse for testing.
  - □ Coordinated meetings with Dr. Porto and Teammates.
- What are you going to do next week
  - □ Compare different PCA CPD approaches and find out which one does better.
  - □ Find different error measurement techniques.
- Blockers, things you want to flag, problems, etc.
  - $\Box$  No current blockers.

## Abstracts

Guangrun Xu, Jianmin Huang, and Yueni Lu "Improved coherent point drift for 3D point clouds registration", Proc. SPIE 11928, International Conference on Image Processing and Intelligent Control (IPIC 2021), 119280D (22 October 2021);

Link: https://doi.org/10.1117/12.2611477

Coherent Point Drift (CPD) is one of the popular robust point cloud registration algorithms in recent years. However, the algorithm uses fast Gaussian transformation to calculate the matrix-vector product, resulting in slower overall registration efficiency. We propose an improved coherent point drift (ICPD) algorithm, which introduces faster Gaussian lattice filtering to calculate the above product and uses the global squared iterative method to reduce the number of iterations of the CPD algorithm. In addition, the outlier w is not accurately expressed in CPD. We propose an iterative outlier formula to solve this problem. Experiments show that the improved algorithm is about two orders of magnitude faster than the CPD algorithm, 1-2 times faster than the ICP algorithm, and shows superior performance in environments with different noise and outlier distributions.

## What did you do and prove it

Primitive version of the v2 of the PCA registration here: <u>https://github.com/Nikitos1865/pycpd-</u> Porto/blob/master/pycpd/pca\_registration\_v2.py. Example run here: <u>https://github.com/Nikitos1865/pycpd-</u> Porto/blob/master/examples/pca\_skull\_deformable\_3D\_v2.py