HAAG Weekly Report (Simplified) – Omar Moursy – 3D Modeling

Time-Log

What did you do this week?

- Updated website homepage for 3D Modeling project https://sites.gatech.edu/3dmodeling/
- Uploaded the Weekly reports and meeting recordings for Week 4.
- Had a progress update meeting with Nikita and Steve.
- Read through resources shared by Dr. Porto such as PyCPD paper, GitHub repository, article explaining coherent point drift and downloaded the mouse model datasets.
- Cloned Nikita's fork of the PyCPD repository.
- Tested the point cloud registration on the source file semilandmarks.json and target file B6AF1_J.ply_align.json based on instructions provided by Nikita.
- Downloaded 3D Slicer to explore the dataset we will be working with further

What are you going to do next week

- Add the Researchers section of the 3D Modeling website and upload any missing documents
- Test the PyCPD code on more of the mouse model dataset.
- Import the mouse models dataset into 3D slicer to visualize the registration problem
- Approach the problem of breaking down the Statistical Shape Model provided by Dr. Porto into Python code that explains 95% of the variation
- Meeting with Dr. Porto on Tuesday to discuss progress and next steps

Blockers, things you want to flag, problems, etc.

- None for this week

Abstracts:

PyCPD: Pure NumPy Implementation of the Coherent Point Drift Algorithm

https://joss.theoj.org/papers/10.21105/joss.04681

Background

Point cloud registration is a common problem in many areas of computer science, particularly computer vision. Point clouds come from many types of data such as LIDAR commonly used for self-driving vehicles, and other sorts of 3D scanners (e.g., structured light) are commonly used to map the surface of physical objects. Point clouds are also used to represent the surface of an anatomical structure extracted from a medical image. Point cloud registration finds a transformation from one point cloud to another. Point cloud registration has use cases in many fields from self-driving vehicles to medical imaging and virtual reality. Typically, point cloud registration is classified into rigid (only rotations or translations), affine (rigid + shearing and scaling) and non-rigid also called deformable registration (non-linear deformation).

Point cloud registration typically requires 2 point clouds. The first point cloud is the "fixed" or "target" point cloud and the second is the "moving" or "source" point cloud. We try to find the transformation that will best align the moving (or source) point cloud with the fixed point cloud. One of the most well known rigid point cloud registration algorithms is the Iterative Closest Point (ICP) algorithm (Besl & McKay, 1992; Chen & Medioni, 1992). ICP is an

iterative algorithm where the following steps are iterated:

(1) for every point on the moving point cloud find the closest point on the fixed point cloud (2) use a least squares approach to find the optimal transformation matrix (rotation, trans-

lation, scaling, shear) to align the point correspondences found in (1) (3) apply the transformation from (2) to the moving point cloud

These steps are repeated until the root mean squared point-to-point distances from (1) converge.

The coherent point drift (CPD) algorithm was created by Myronenko and Song (Myronenko & Song, 2010) to overcome many of the limitations of ICP and other previous registration methods (Besl & McKay, 1992; Chen & Medioni, 1992; Fitzgibbon, 2003; Rusinkiewicz & Levoy, 2001). Namely, these other methods did not necessarily generalize to dimensions greater than 3 and they were prone to errors such as noise, outliers, or missing points. The CPD algorithm is a probabilistic multidimensional algorithm that is robust and works for both rigid and non-rigid registration. In CPD the moving point cloud is modelled as a Gaussian Mixture Model (GMM) and the fixed point cloud is treated as observations from the GMM. The optimal transformation parameters maximize the Maximum Likelihood / Maximum A Posteriori (MAP) estimation that the observed point cloud is drawn from the GMM. A key point of the CPD algorithm is that it forces the points to move coherently by preserving topological structure. The CPD algorithm is also an iterative algorithm that iterates between an expectation (E) step and a maximization (M) step until convergence is achieved. The E-step estimates the posterior probability distributions of the GMM centroids (moving points) given the data (fixed points) then the M-step updates the transformation to maximize the posterior probability that the data belong to the GMM distributions. The E- and M-steps are iterated until convergence.

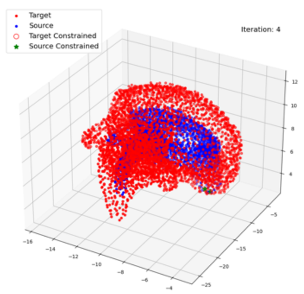
What did you do and prove it

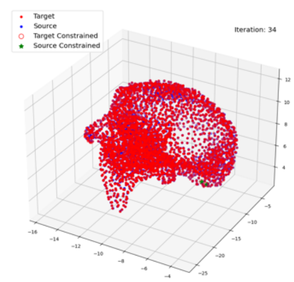
Updated website homepage for [3D Modeling project](). I also uploaded the missing [weekly reports]() and [meeting recordings]().

We had a team meeting to discuss the progress and setup a better plan to ensure no one is falling behind. The recording is also on the website.

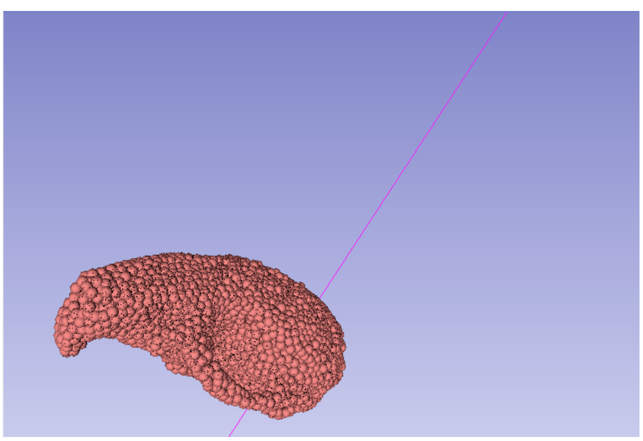
Read through resources shared by Dr. Porto such as PyCPD paper, GitHub repository, article explaining coherent point drift and downloaded the mouse model datasets.

Cloned GitHub repo and tested example using the source file semilandmarks.json and target file B6AF1_J.ply_align.json based on instructions provided by Nikita.





Downloaded 3D Slicer to import the mouse model dataset and explore them further.



Links to the paper, GitHub repo and article:

https://joss.theoj.org/papers/10.21105/joss.04681

https://siavashk.github.io/2017/05/14/coherent-point-drift/

https://github.com/Nikitos1865/pycpd-Porto/tree/master