

## IFT 6112: Assignment 3

Due: Nov, 8<sup>th</sup>, 2018

### Objective:

Develop and implement a mesh deformation tool. Implement the As-Rigid-As-Possible method (ARAP).

### Tasks:

1. Develop a mesh deformation tool based on local coordinates.
2. Use an iterative method with a rotation update mechanism.
3. Use an off-the-shelf linear solver to efficiently solve the resulting linear system(s).
4. Develop a user-friendly API/visualization mechanism for your algorithm.

### Instructions:

1. Develop an anchor-based mesh deformation tool for closed manifold meshes. Start by implementing a method which has a single solve per anchor displacement (no rotation update). Only when this method works you can add support for iterated solver with rotation updates.
2. For solving the linear system(s) you obtain you can use either the Eigen library or another solver, as long as you are familiar with it. Pay attention to efficiency. What solver is the most efficient for this case? (Write out answers in the Readme file).
3. You can limit your method to changing one anchor at a time. This way you can have a fixed rotation axis for you Region of Interest (ROI) - think how this can simplify your updates.
4. Bonus:
  - There is a bonus for handling multiple anchors as well as for result quality and speed.
  - Note: please make sure your basic algorithm works before implementing the bonuses. *There will be no bonus points given if the basic method has problems.*
5. Your API should support the following mesh operations, performed repeatedly and in any order (*pay attention to API ease of use*):
  - Specify an anchor vertex or triangle.
  - Specify a region of influence.
  - Perform a deformation by moving the anchor around (see "Selection and Deformation" below)

- You can also provide a mechanism to prescribe/change anchor normals or provide a mechanism where the user specifies only anchor rotations (no translation).
6. To visualize the deformation process and ensure correctness you should provide a mechanism to highlight the ROI and anchor. You should provide an interface option (e.g. key-press) allowing the user to switch the visualization on and off.

### **Selection and Deformation in Cartel:**

Selection of multiple vertices is already implemented in ControlState and main. Selection is done while the right mouse button is pressed. Remember that clearing selection is a menu item and bound to the key 'C'. You may want to change that code slightly to allow for multiple (different) selections or create events to save the current selection in EditMesh for different purposes (anchor(s), ROI).

You will need to create controls to drive the deformation itself. You have several options, a few suggestions are:

- You can check how mouse movement is detected in ControlState.cpp for selection and implement something similar but checking for left mouse button to be pressed.
- You can use keyboard events or interface buttons or sliders for movement in different axis, but that might only allow for discrete steps.
- You can input (through the console) an amount and axis of deformation.

You can use the same meshes as before or download whatever you like.

As usual, make sure you understand every line of the code you write.

Submission: private message to me on Piazza with a source code attached.

### **Reference:**

Olga Sorkine and Marc Alexa. 2007. *As-rigid-as-possible surface modeling*. In Proceedings of the fifth Eurographics symposium on Geometry processing (SGP '07). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 109-116.