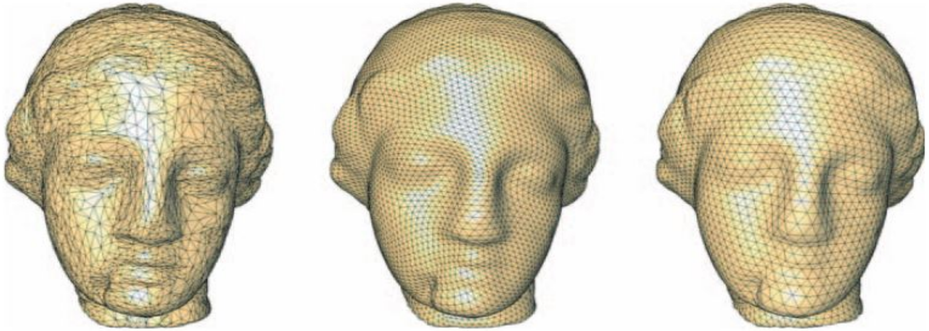
Connectivity Regularization

Intermediate Presentation



Algorithm

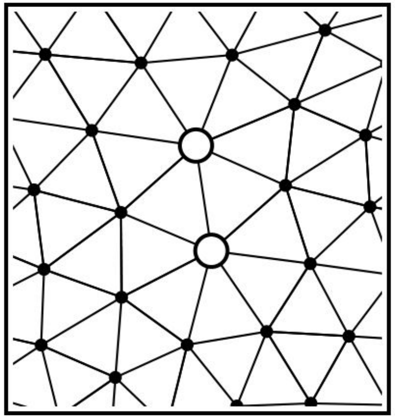
Step 1: Perform all *basic* Edge Flips

- Check edges for ones that improve the degree of the vertices when flipped
- Optimal degree of a vertex d(v) is 4 for boundary vertices and 6 for other ones

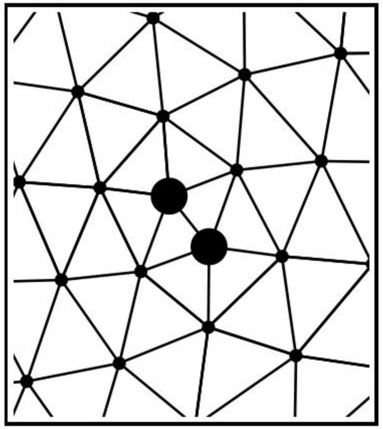
Algorithm

Step 2:

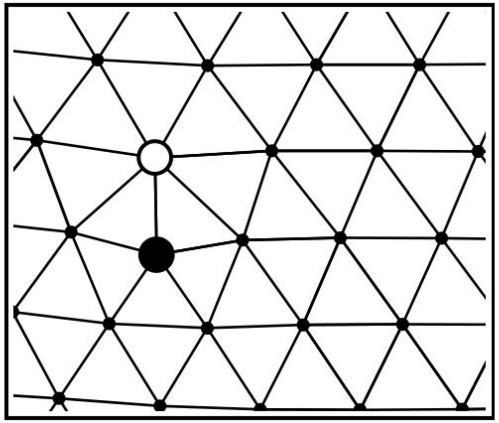
- Categorize edges:



Long



Short

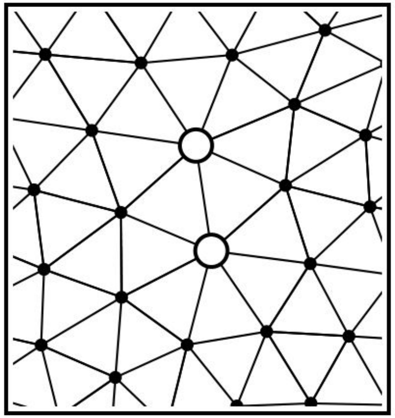


Drifting

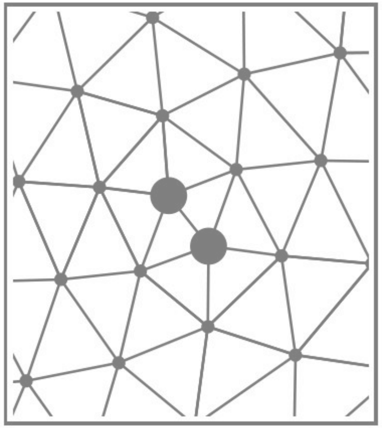
Algorithm

Step 2:

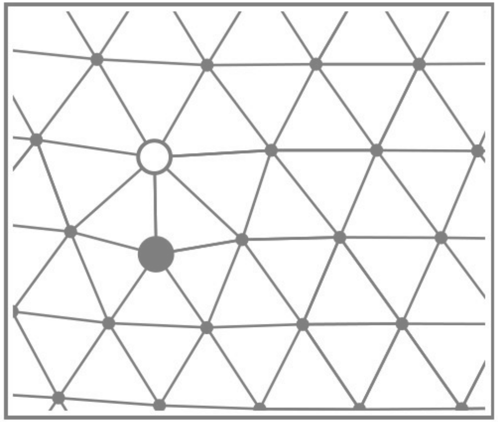
- Long edges: Split and perform basic Edge Flips



Long



Short

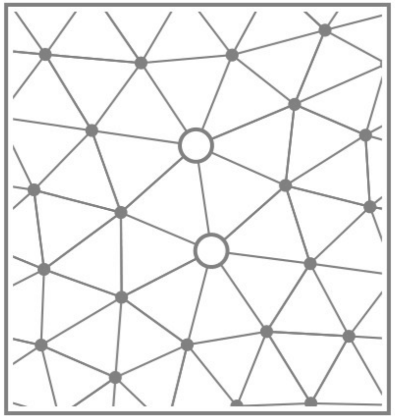


Drifting

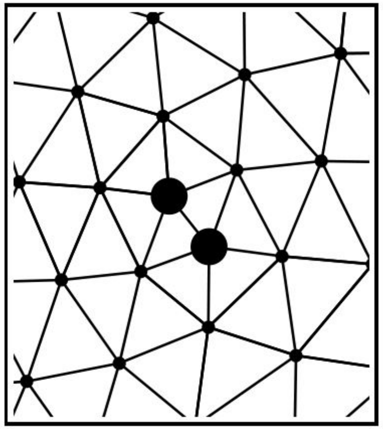
Algorithm

Step 2:

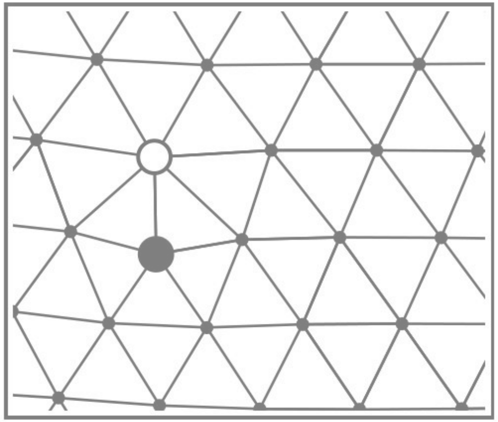
- Short Edges: Collapse and perform basic Edge Flips



Long



Short

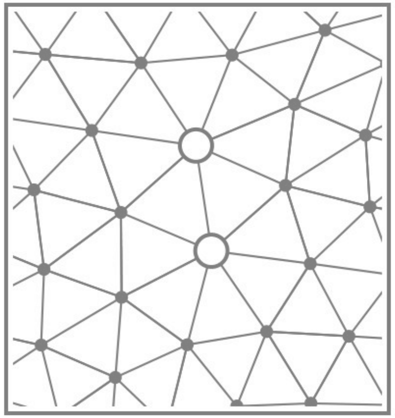


Drifting

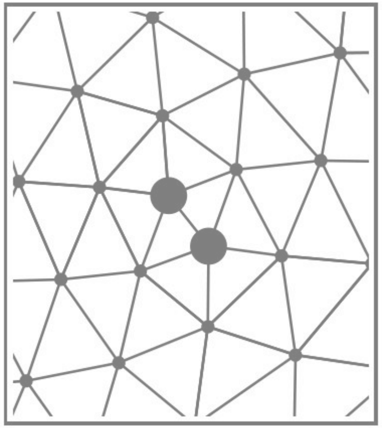
Algorithm

Step 2:

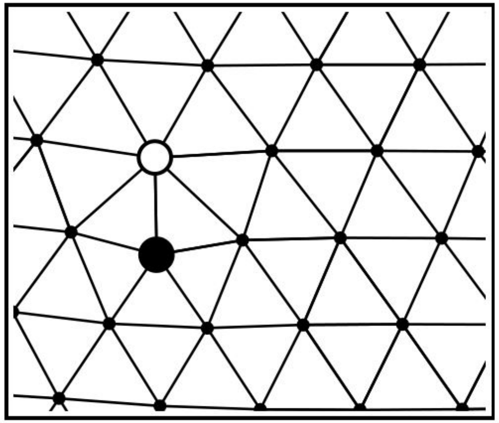
- Drifting edges: move along mesh until they meet another irregular vertex



Long



Short

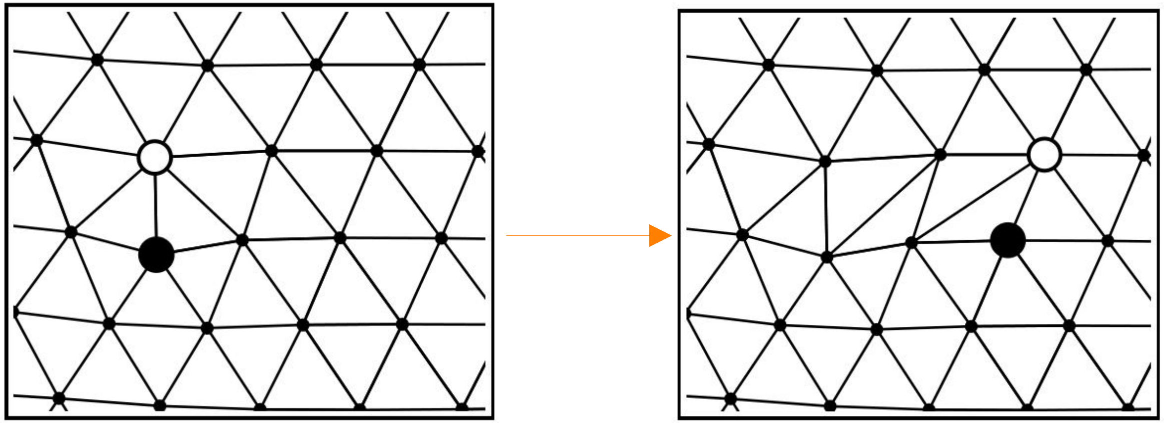


Drifting

Algorithm

Step 2:

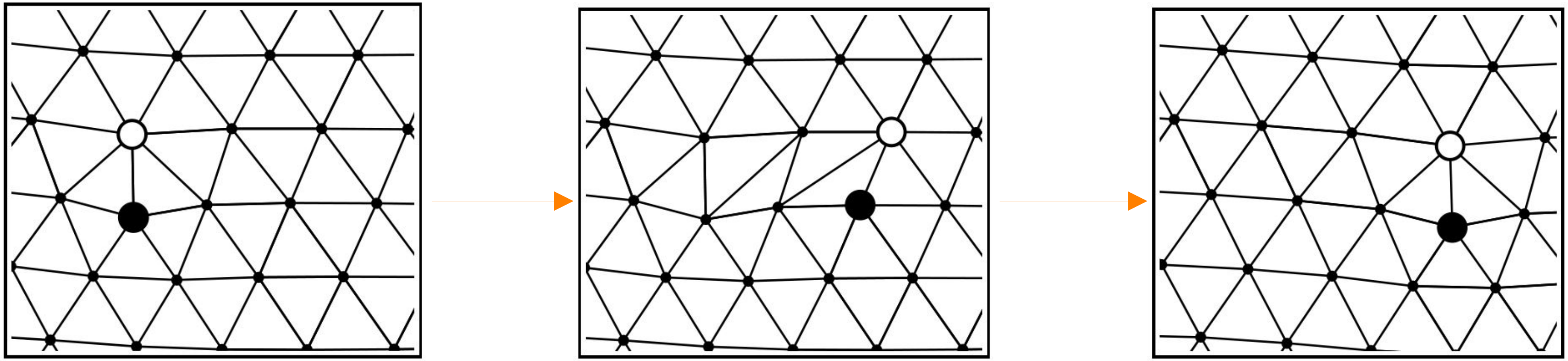
- Drifting edges: move along mesh until they meet another irregular vertex



Algorithm

Step 2:

- Apply angle-based smoothing to involved vertices after each operation



Algorithm

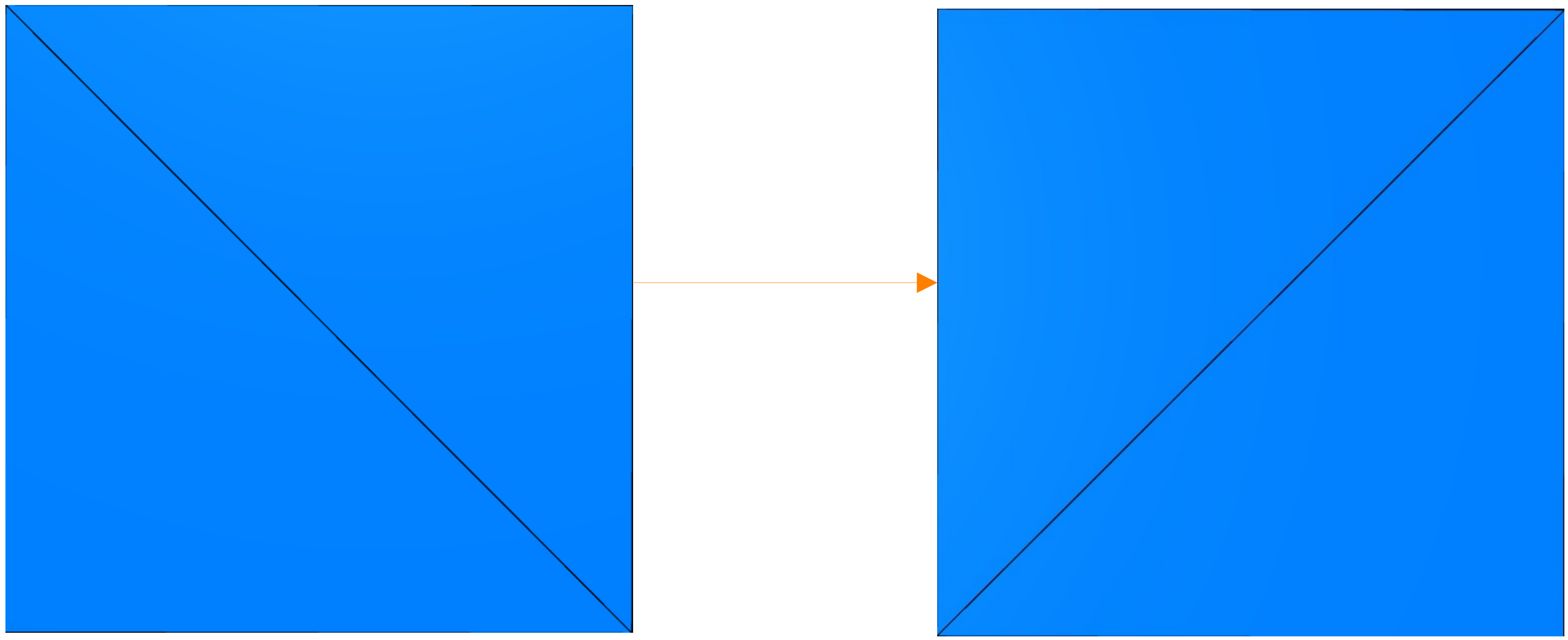
Result:

- Few irregular Vertices
- All irregular vertices surrounded by regular ones

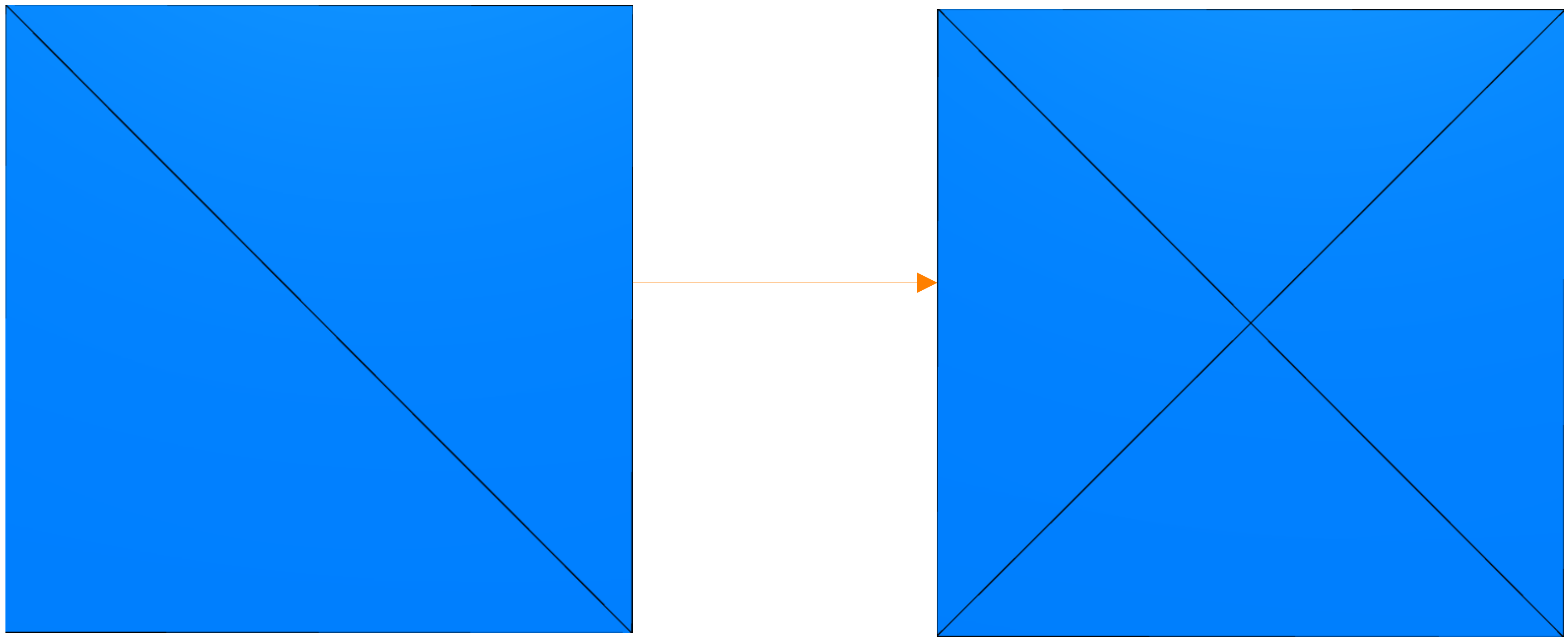
Current state

- Individual Operations are implemented:
  - Edge Flip
  - Edge Split
  - Edge Collapse
  - Angle-based Smoothing

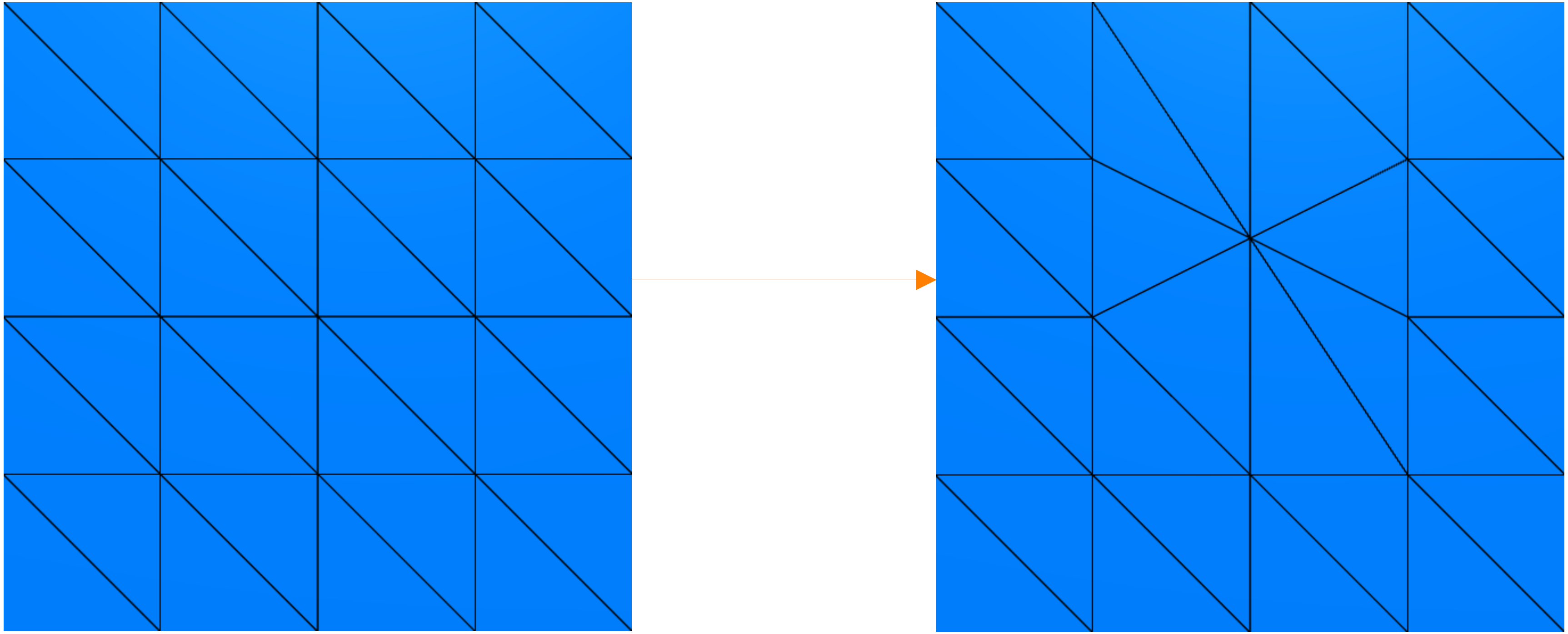
Edge Flip



Edge Split

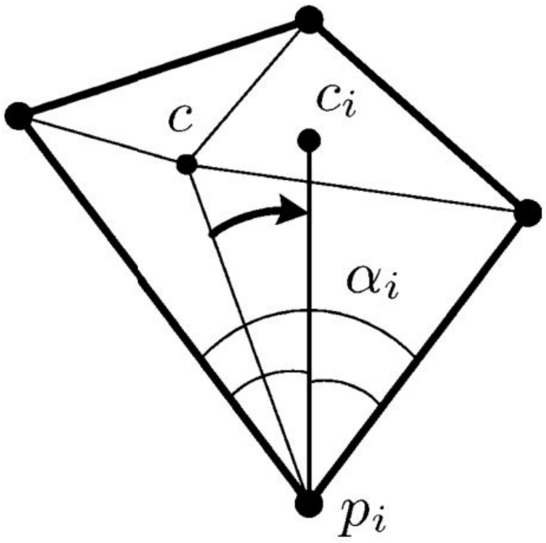


Edge Collapse



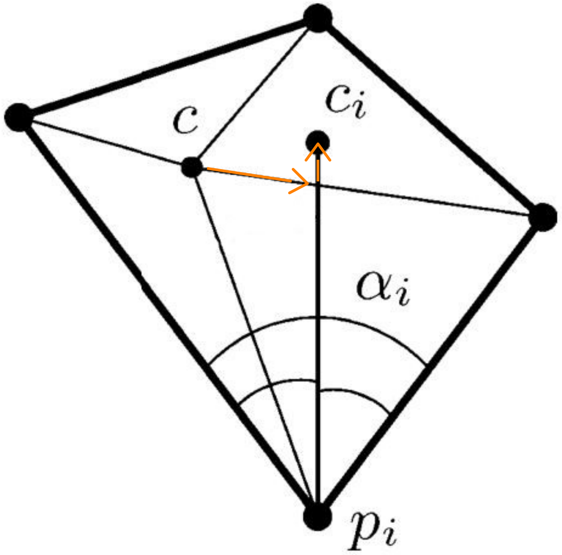
Angle based smoothing

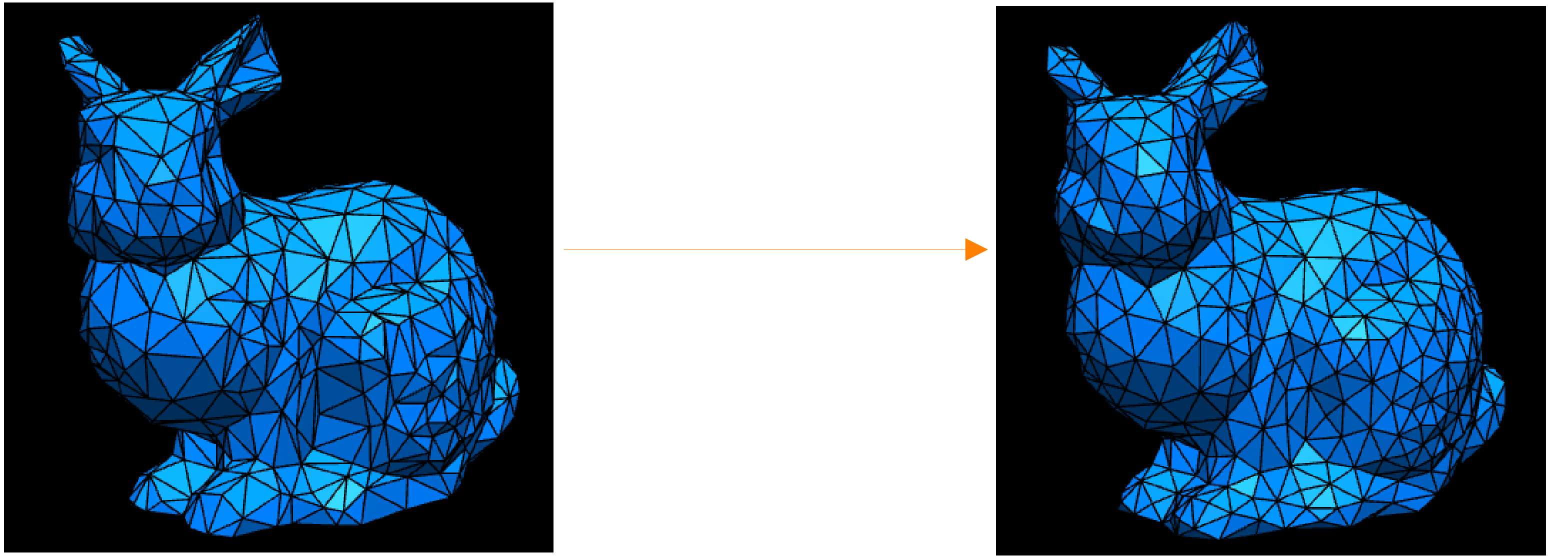
- Calculate new position of vertex c for all neighbor vertices p by rotating edge p → c
- Move c to average of calculated positions



Angle based smoothing

- Adaptation for 3D meshes:
  - Calculate new Position by moving c along the other edges
  - Extend to preserve distance





To Do

- Put pieces together
- Handle Error cases
- Controls/Parameters/Thresholds
- Improve Performance
- Code Cleanup