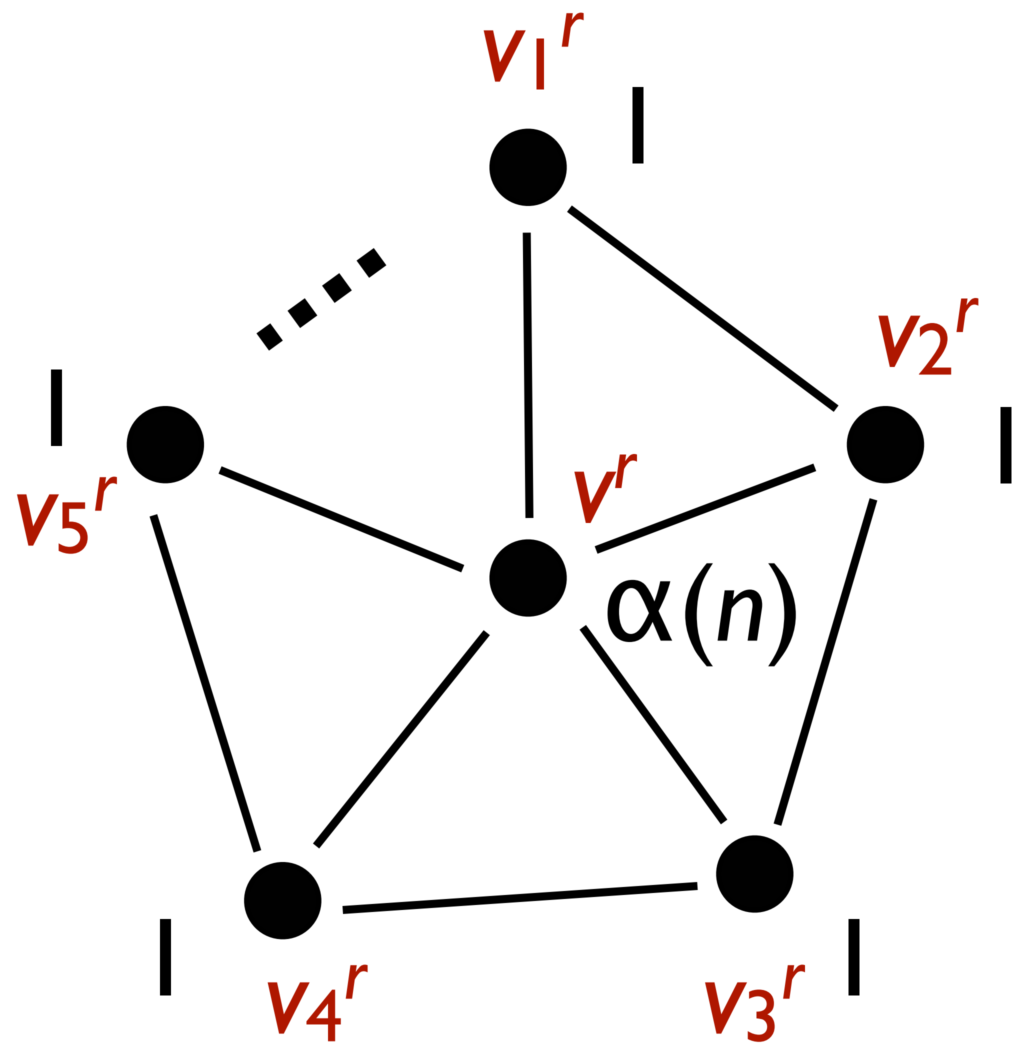
Piecewise Smooth Surface Reconstruction

Hugues Hoppe et al.

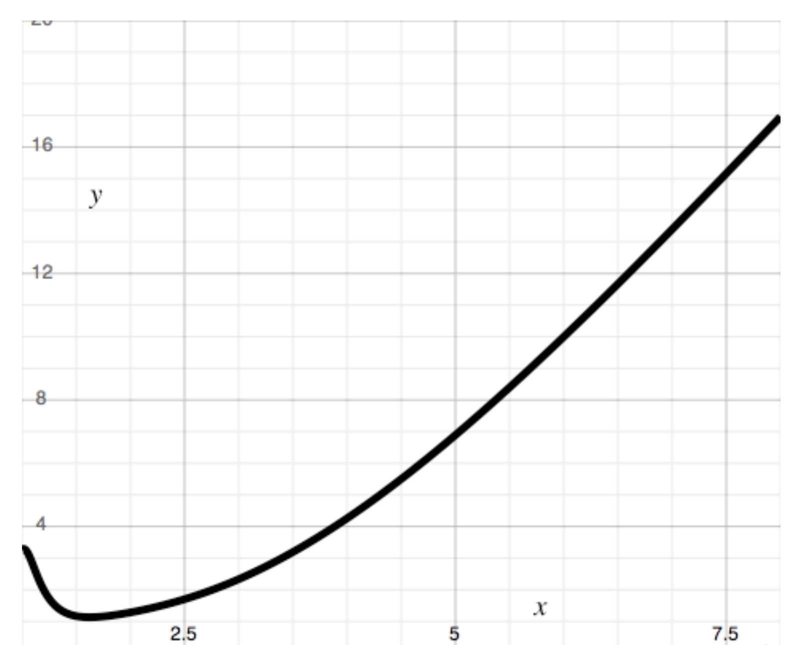
Loop's Subdivision Scheme

- A mesh M is iteratively refined into smoother meshes M^r
- Each vertex in M^{r+1} is a weighted average of "nearby" vertices in M^r
- Weights are given by masks

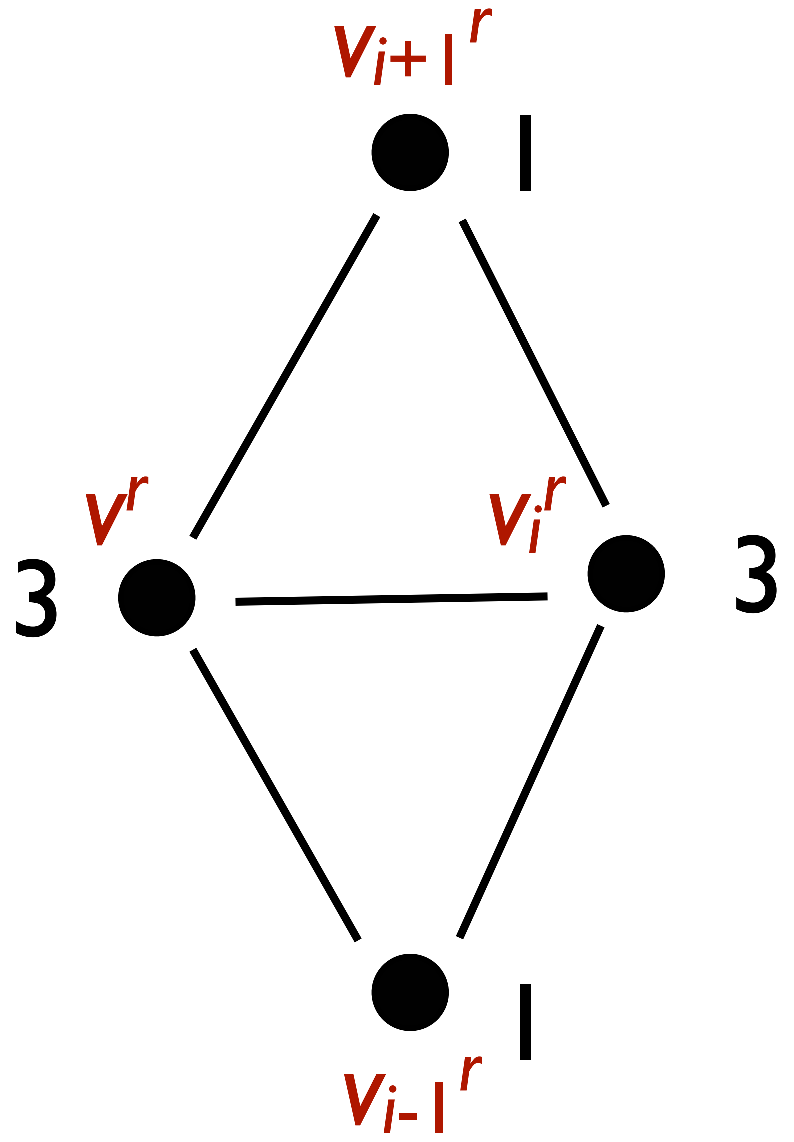
Subdivision masks



$$v^{r+1} = (\alpha(n)\, v^r + v_1{}^r + \ldots + v_n{}^r)/(\alpha(n) + n)$$

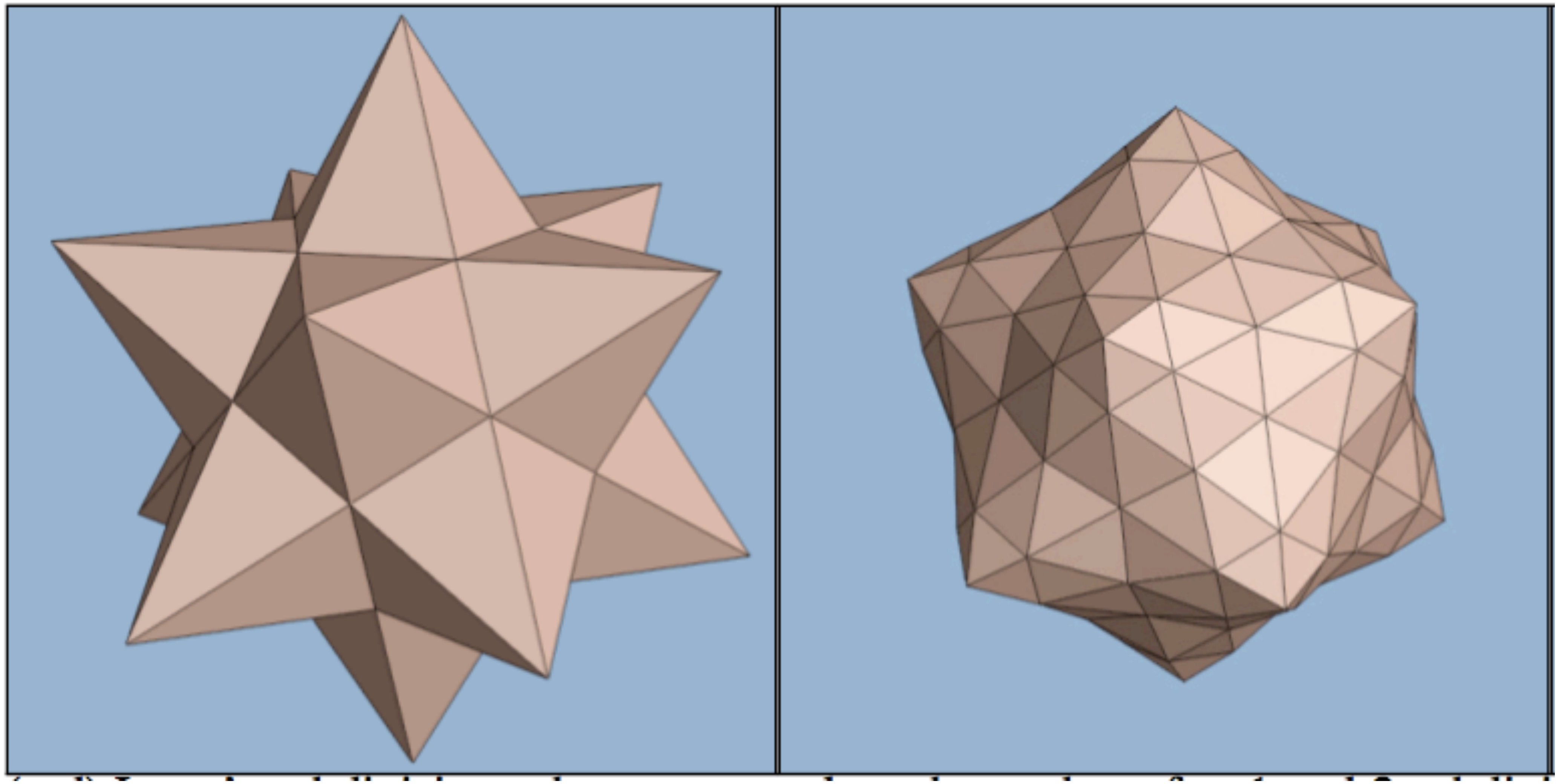


Subdivision masks



$$v_i^{r+1} = (3v^r + 3v_i^r + v_{i-1}{}^r + v_{i+1}{}^r)/8$$

Subdivision masks



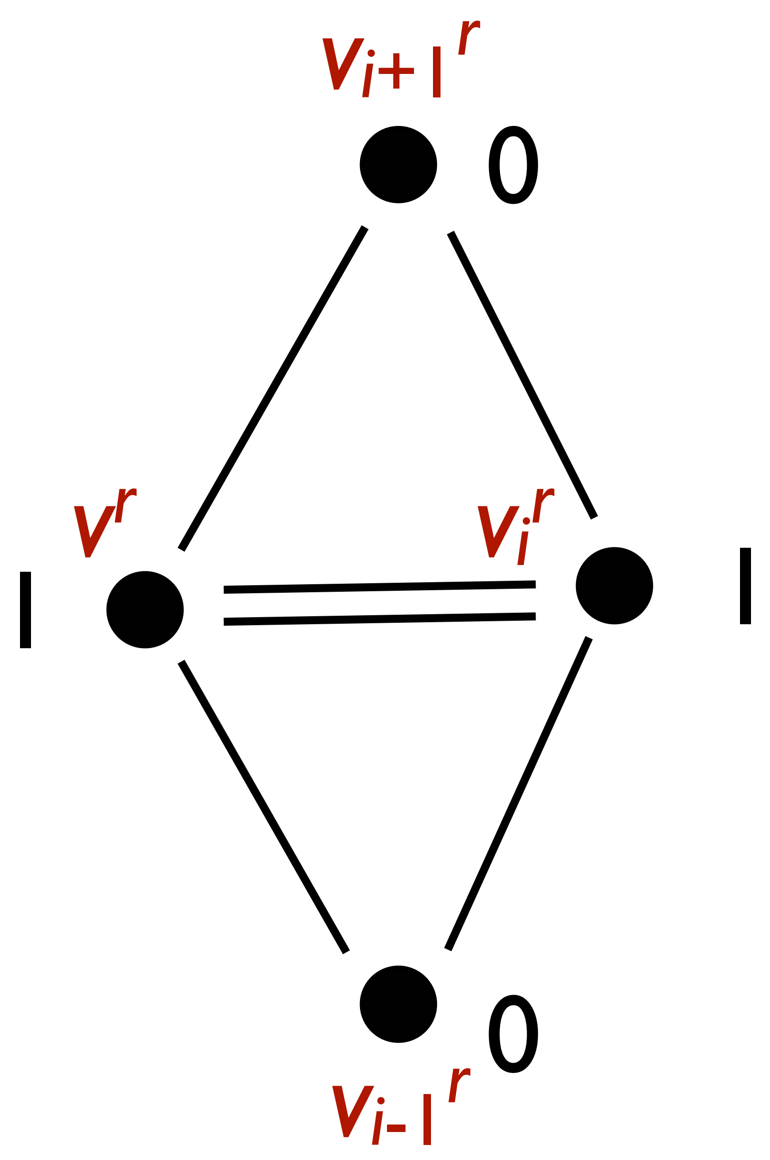
Limits of the surface

- It turns out that we can calculate exactly the "eventual" position v^{∞} of v using eigenanalysis (yay math!)
- We can also get exact normal vectors (e.g., for Phong shading)

Creases

- This paper modifies Loop surfaces so you can mark edges as "creases"
- Only difference is that masks are different for points and edges on creases
  - Vertices on one side of a crease cannot affect vertices on the other side

Crease edge mask



$$v_i^{r+1} = (v^r + v_i^r)/2$$

Crease vertices similarly "ignore" non-crease vertices

The Problem

- Given a set V of vertices, find a mesh M which, when used as a Loop surface:
  - is concise (few control vertices)
  - has few crease edges
  - minimizes the “distance” from V to M^∞

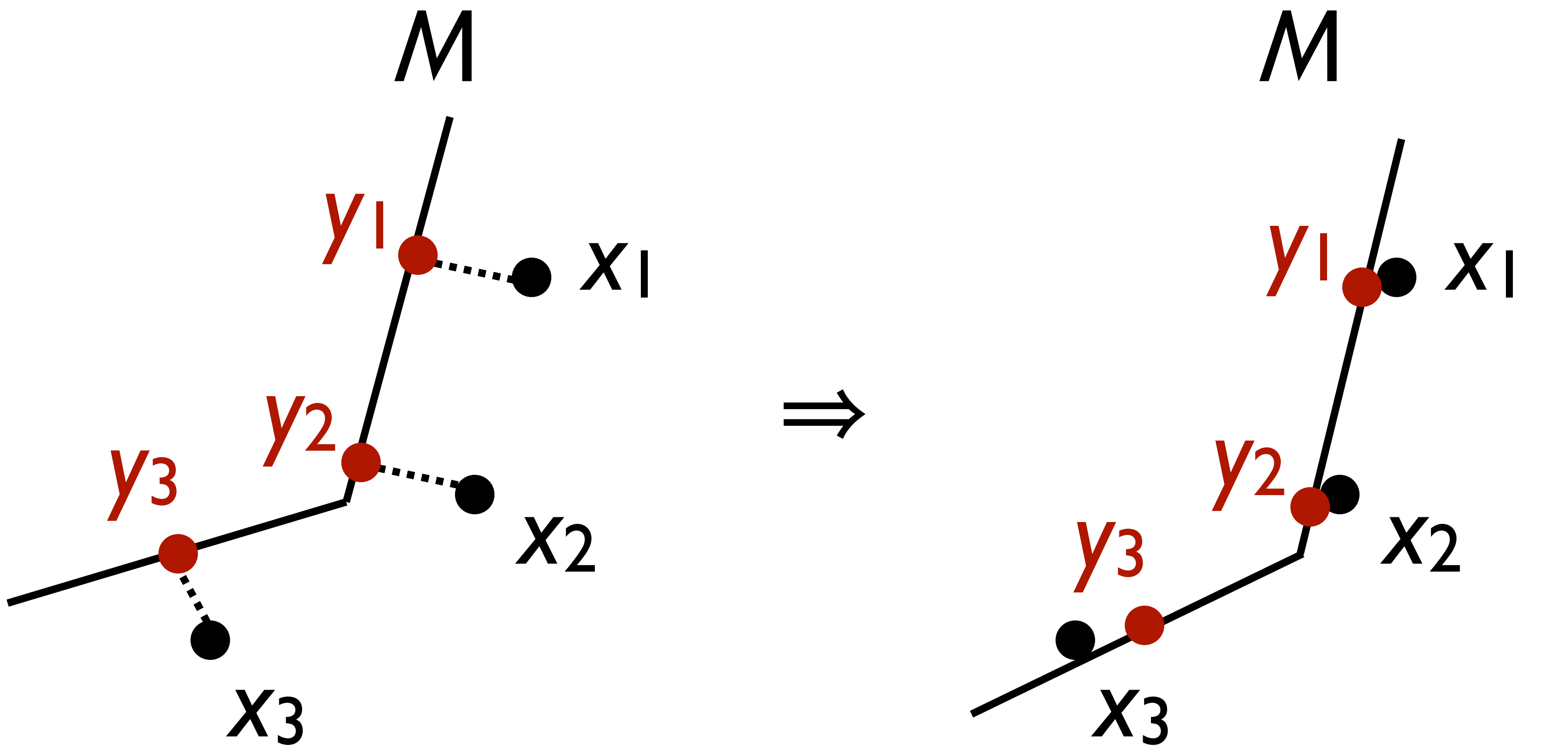
Energy function

- $E(M, V) = E_{dist}(M, V) + c_{rep}m + c_{sharp}e$
  - m is number of vertices in M
  - e is number of crease edges in M

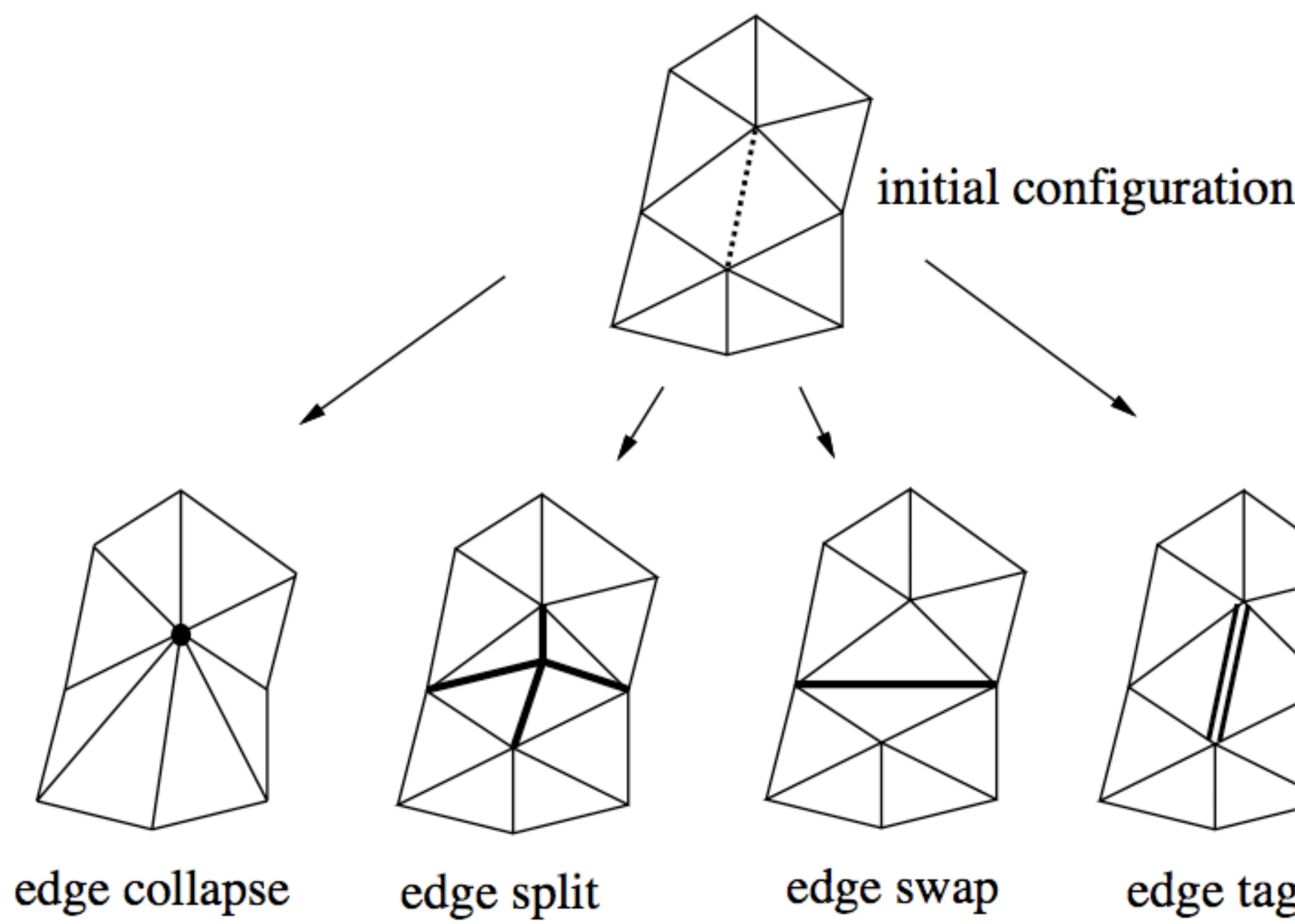
Mesh Optimization

- If we keep the connectivity of the mesh constant and move the vertices, it turns out to be an iterative least-squares problem

Mesh Optimization



Mesh Optimization



Pretty Pictures