CSCI-5229 Final Project
Mesh Deformation

Ryo Suzuki
Graduate School of Computer Science
ryo.suzuki@colorado.edu

Demo

Mesh Deformation

3D model sculpture

3D model character animation

How it works

As-Rigid-As-Possible Surface Modeling

Olga Sorkine and Marc Alexa

TU Berlin, Germany

Abstract

Modeling tasks, such as surface deformation and editing, can be analyzed by observing the local behavior of the surface. We argue that defining a modeling operation by asking for rigidity of the local transformations is useful in various settings. Such formulation leads to a non-linear, yet conceptually simple energy formulation, which is to be minimized by the deformed surface under particular modeling constraints. We devise a simple iterative mesh editing scheme based on this principle, that leads to detail-preserving and intuitive deformations. Our algorithm is effective and notably easy to implement, making it attractive for practical modeling applications.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – geometric algorithms, languages, and systems

1. Introduction

When we talk about *shape*, we usually refer to a property that does not change with the orientation or position of an object. In that sense, preserving shape means that an object is only rotated or translated, but not scaled or sheared. In the context of interactive shape modeling it is clear, however, that a shape has to to be stretched or sheared to satisfy the modeling constraints placed by the user. Users intuitively expect the deformation to preserve the shape of the object locally, as happens with physical objects when a smooth, large-scale deformation is applied to them. In other words, small parts of the shape should change as rigidly as possible.

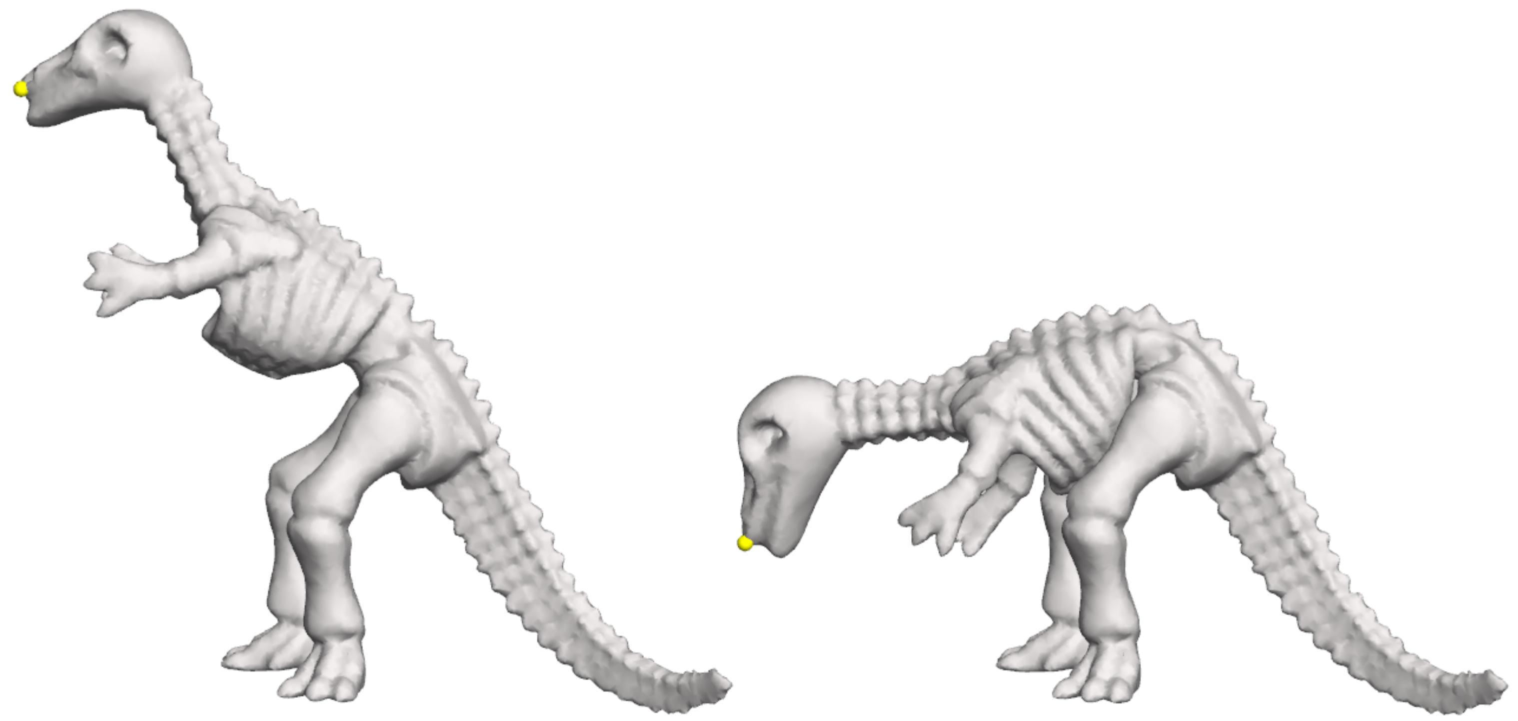
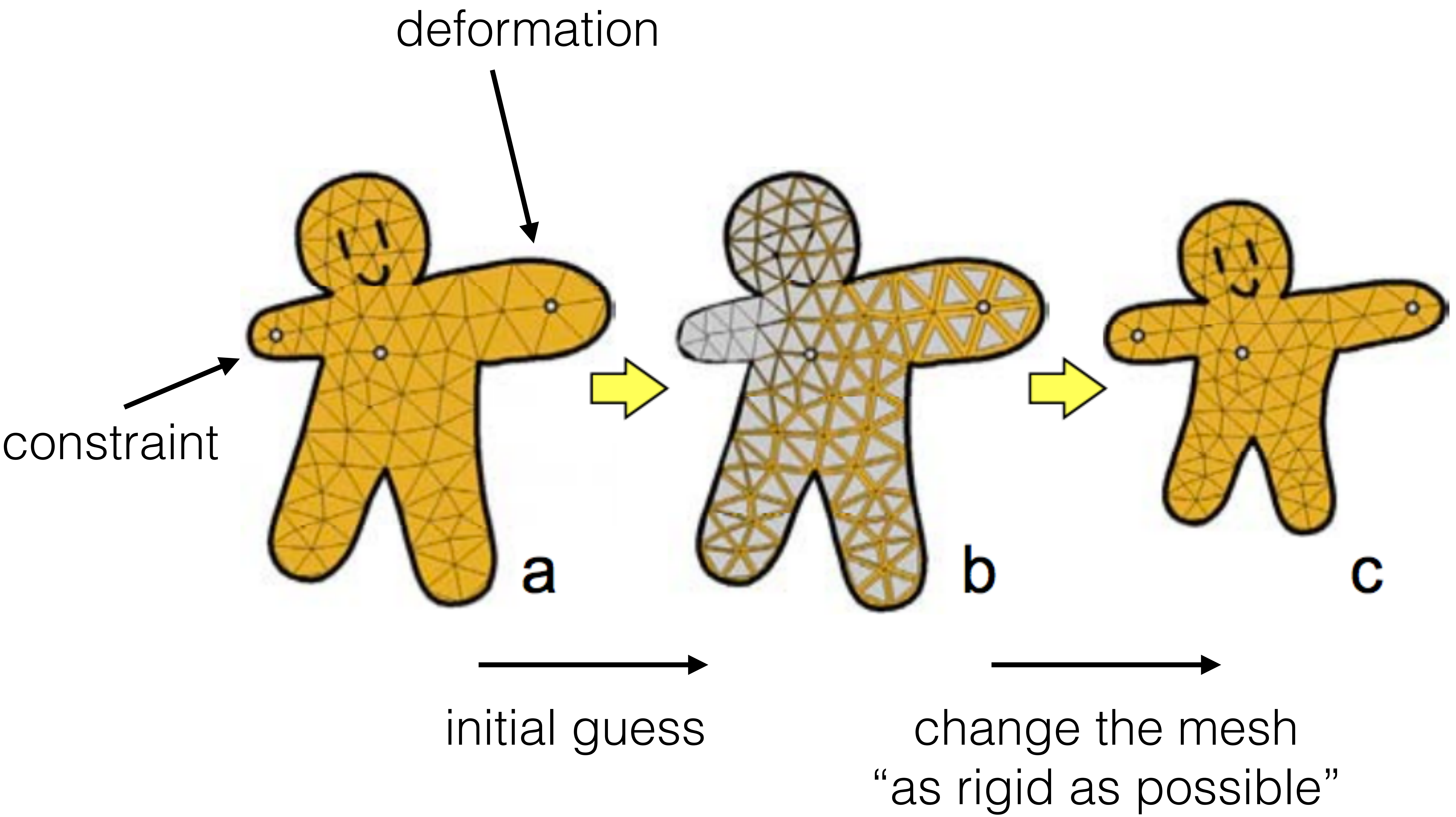
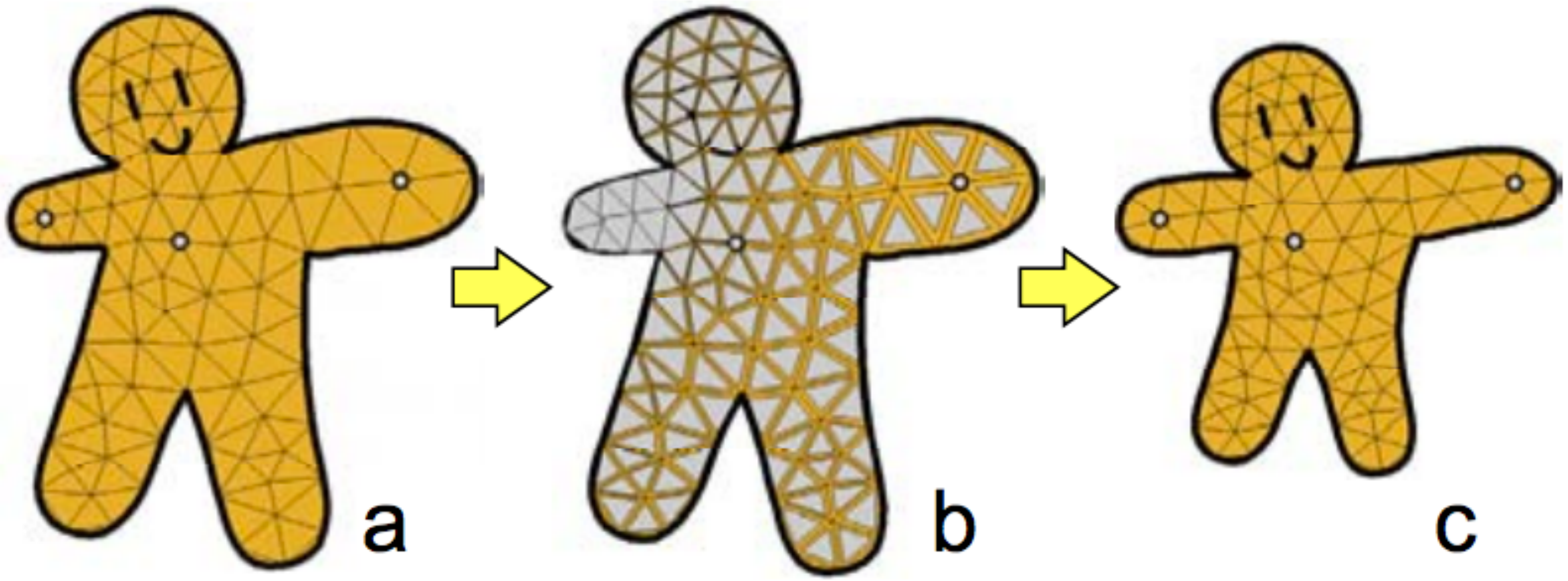


Figure 1: *Large deformation obtained by translating a single vertex constraint (in yellow) using our as-rigid-as-possible technique.*


deformation
constraint
a
b
c
initial guess
change the mesh
“as rigid as possible”

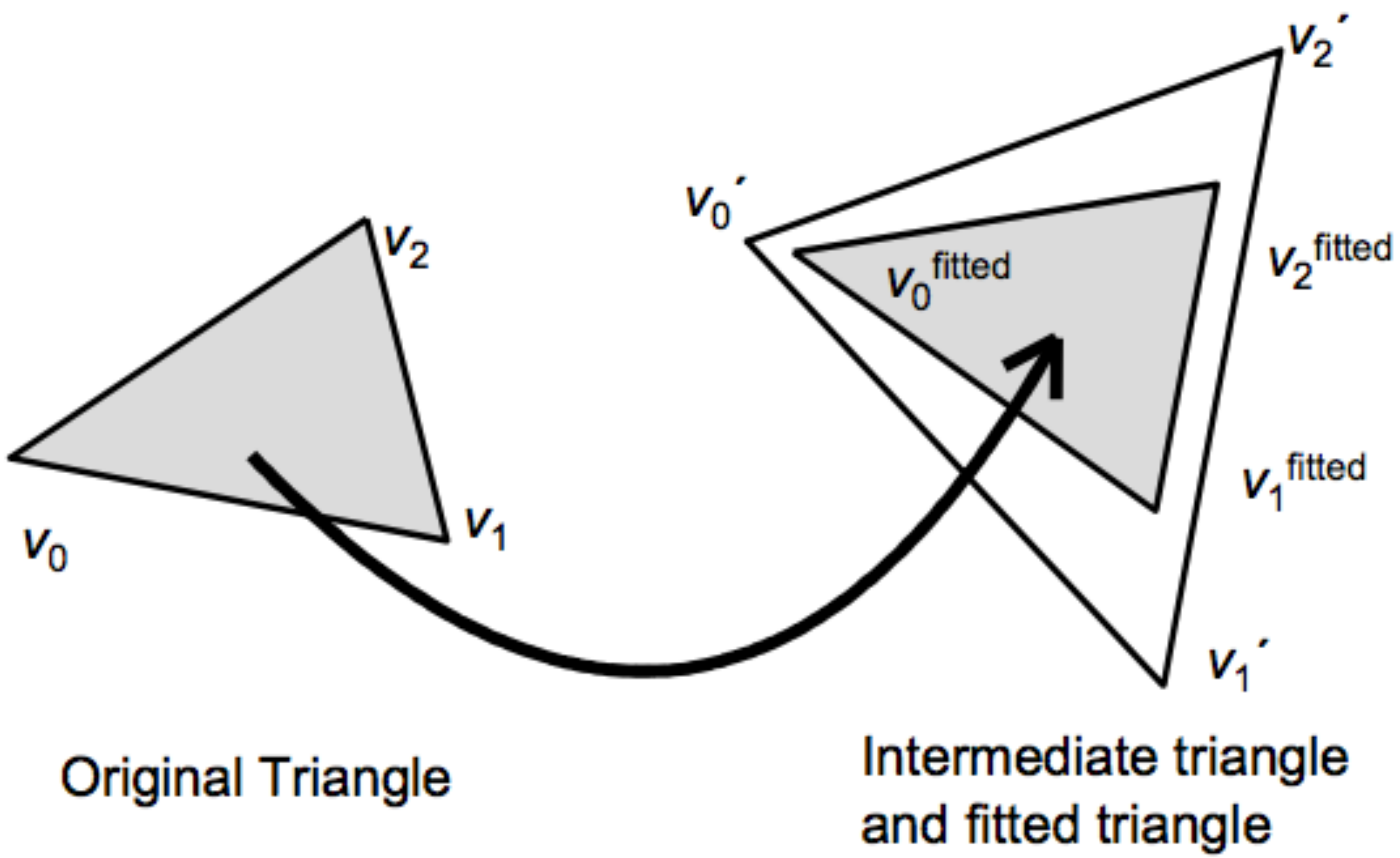
as rigid as possible = minimize scale and rotation



initial guess

change the mesh
"as rigid as possible"

as rigid as possible = minimize scale and rotation



1. **Compute Laplacian weight matrix**
2. Minimize the energy function

$$E(\mathcal{C}_i, \mathcal{C}_i') = \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| (\mathbf{p}_i' - \mathbf{p}_j') - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j) \right\|^2. \quad (3)$$

$$\underset{\mathbf{R}_i}{\operatorname{argmin}} \sum_j -2w_{ij}\, \mathbf{e}_{ij}'^T\, \mathbf{R}_i\, \mathbf{e}_{ij} = \underset{\mathbf{R}_i}{\operatorname{argmax}} \sum_j w_{ij}\, \mathbf{e}_{ij}'^T\, \mathbf{R}_i\, \mathbf{e}_{ij} =$$

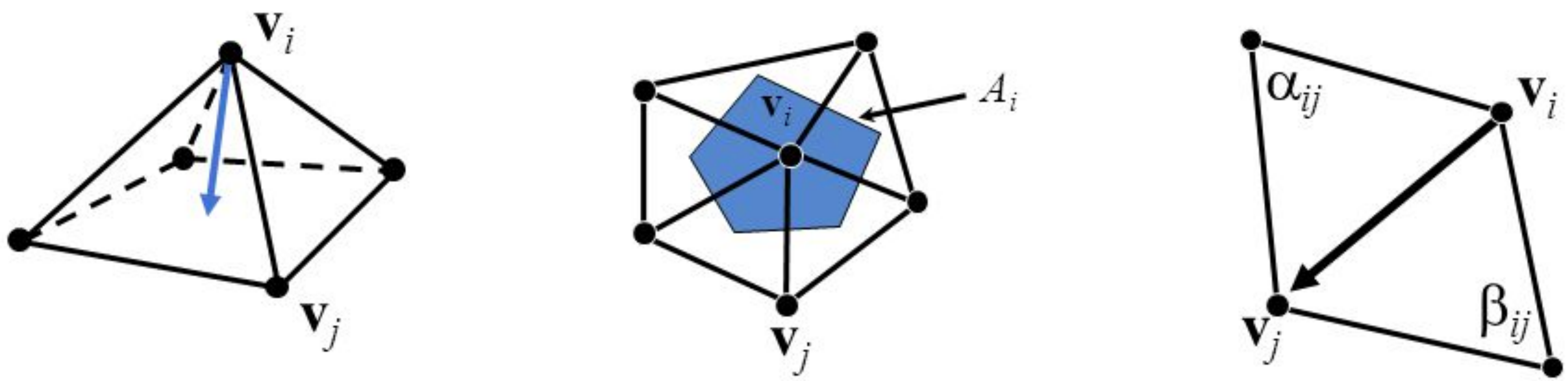
$$= \underset{\mathbf{R}_i}{\operatorname{argmax}}\ Tr\left(\sum_j w_{ij}\, \mathbf{R}_i\, \mathbf{e}_{ij}\, \mathbf{e}_{ij}'^T \right) =$$

$$= \underset{\mathbf{R}_i}{\operatorname{argmax}}\ Tr\left(\mathbf{R}_i \sum_j w_{ij}\, \mathbf{e}_{ij}\, \mathbf{e}_{ij}'^T \right).$$

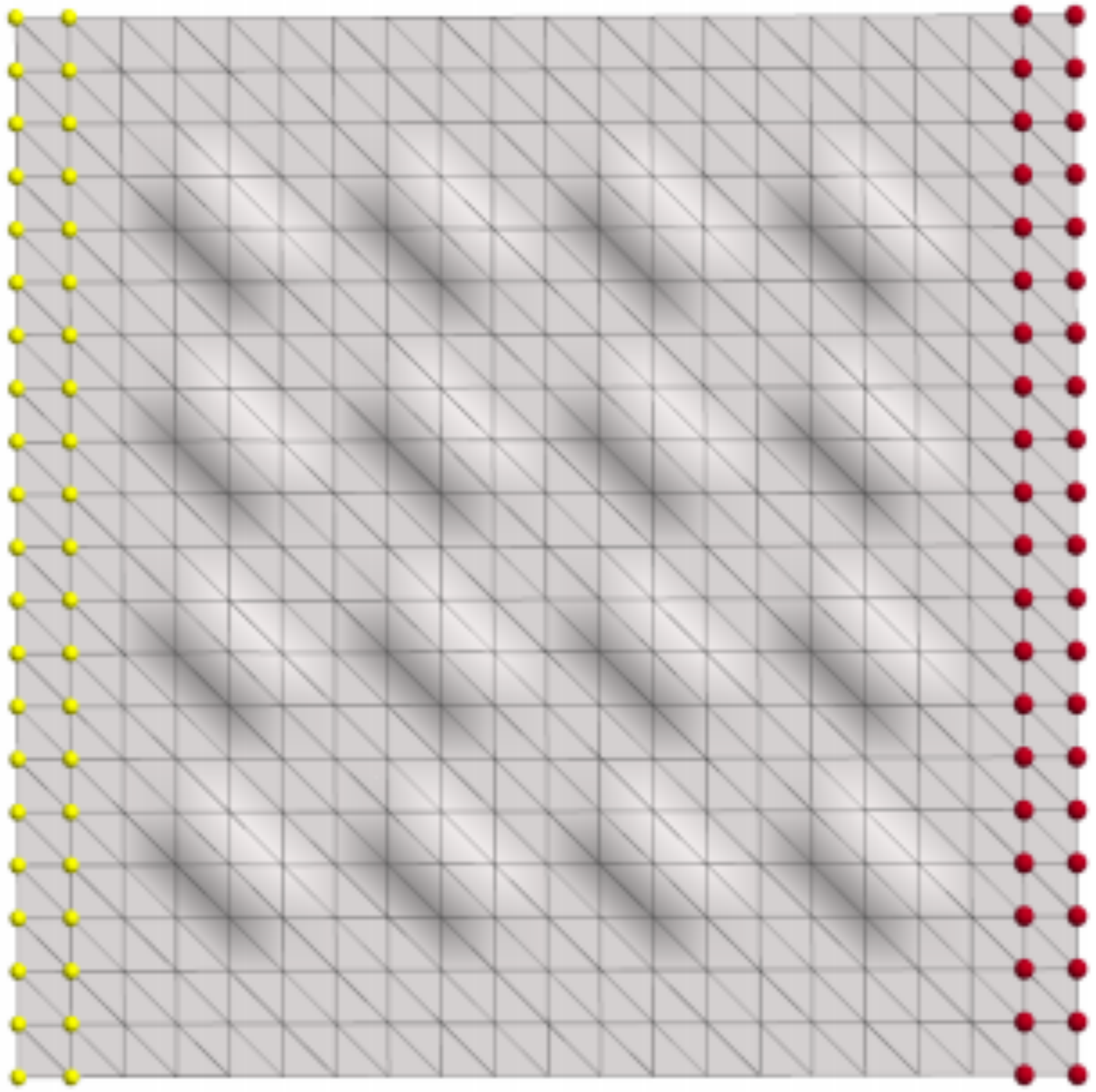
Laplacian Matrix

- Cotangent formula

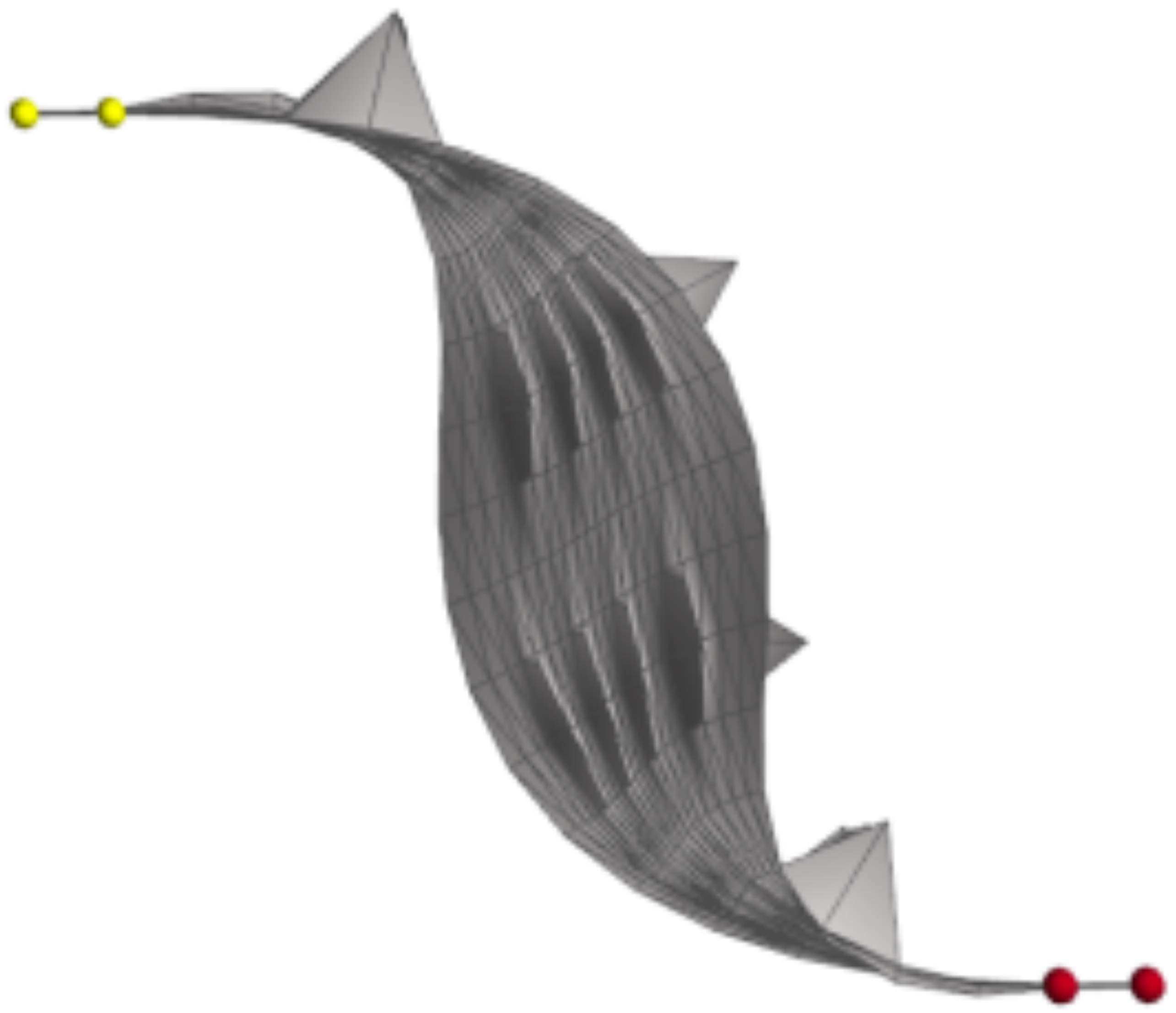
$$L_c(\mathbf{v}_i) = \frac{1}{A_i} \sum_{j \in \mathcal{N}(i)} \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{v}_j - \mathbf{v}_i)$$



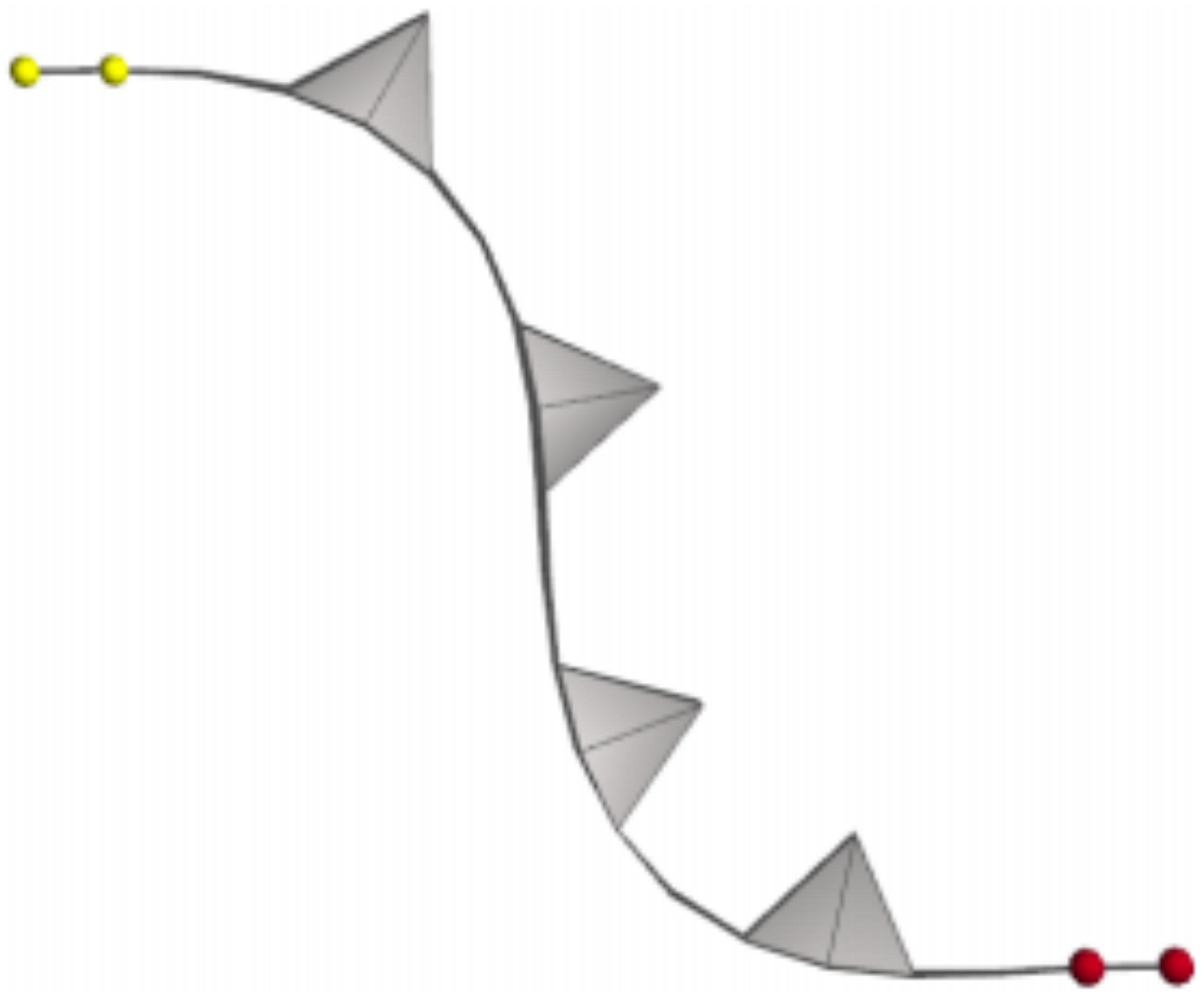
Laplacian Matrix



original mesh

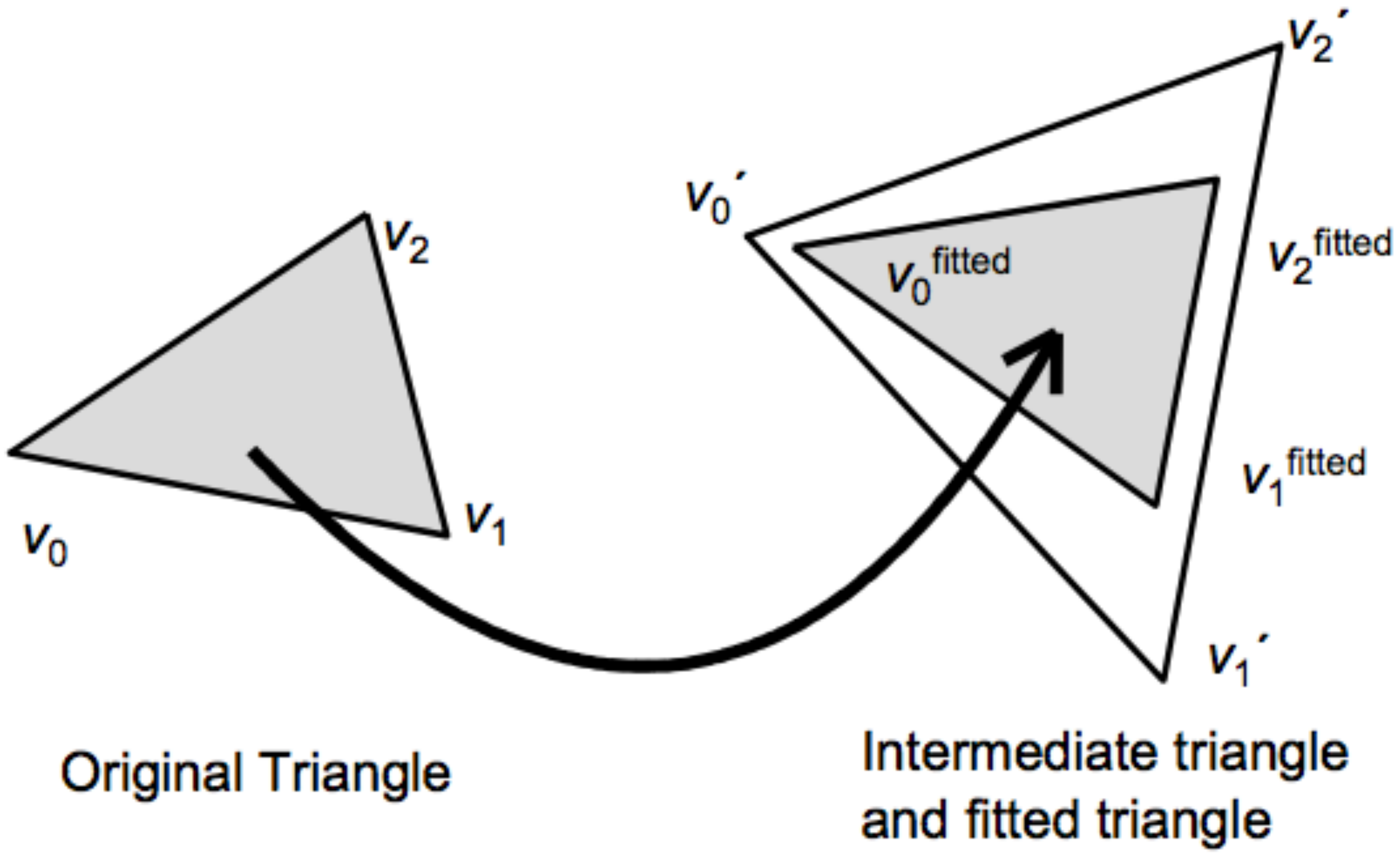


uniform weights



cotan weights

as rigid as possible = minimize scale and rotation



1. Compute Laplacian weight matrix
2. **Minimize the energy function**

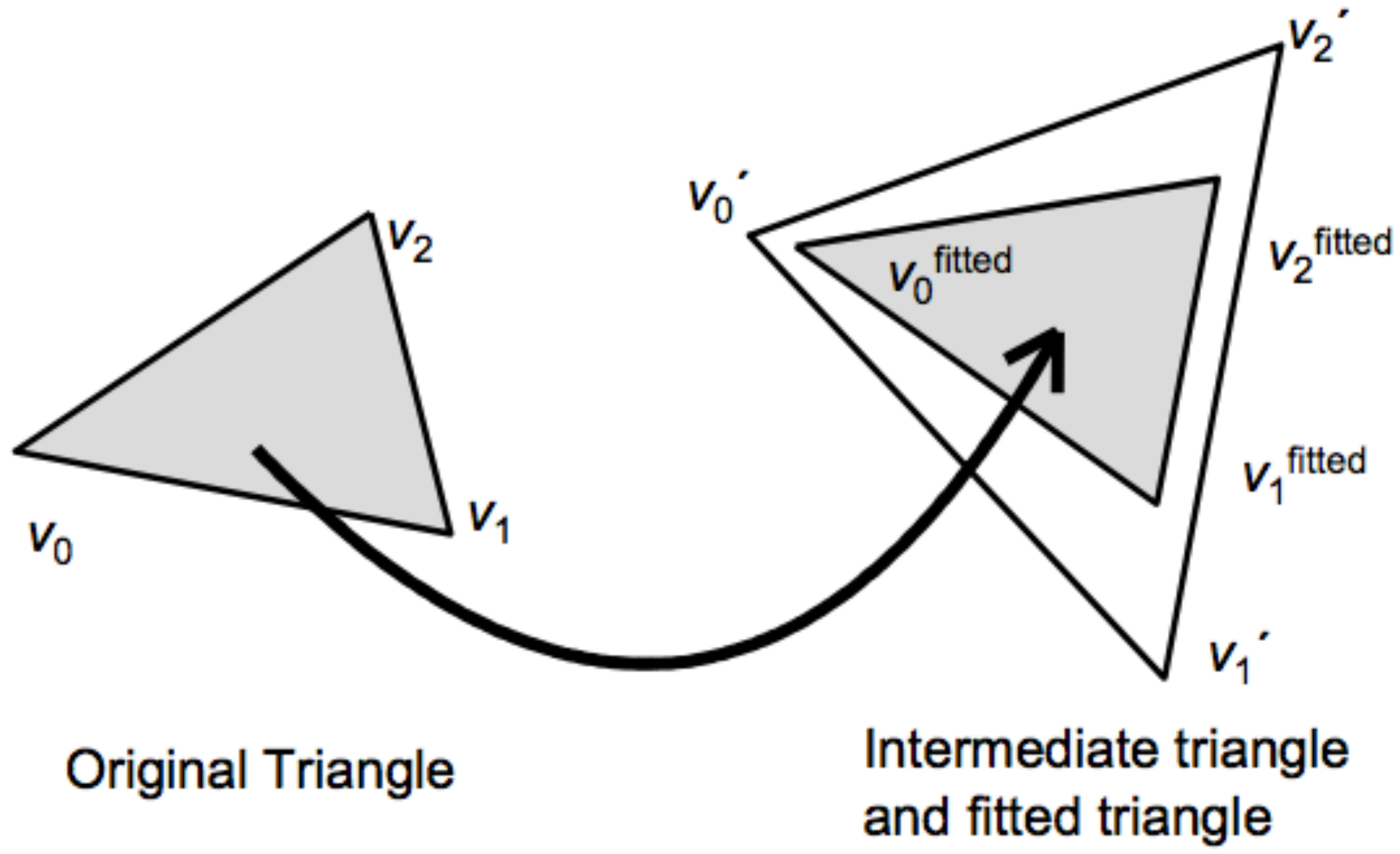
$$E(\mathcal{C}_i,\mathcal{C}_i') = \sum_{j\in\mathcal{N}(i)} w_{ij} \left\|(\mathbf{p}_i' - \mathbf{p}_j') - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j)\right\|^2 . \quad (3)$$

$$\frac{\partial E(\mathcal{S}')}{\partial \mathbf{p}_i'} = \sum_{j\in\mathcal{N}(i)} 4w_{ij}\left((\mathbf{p}_i' - \mathbf{p}_j') - \frac{1}{2}(\mathbf{R}_i + \mathbf{R}_j)(\mathbf{p}_i - \mathbf{p}_j)\right)$$

$$\sum_{j\in\mathcal{N}(i)} w_{ij}(\mathbf{p}_i' - \mathbf{p}_j') = \sum_{j\in\mathcal{N}(i)} \frac{w_{ij}}{2}(\mathbf{R}_i + \mathbf{R}_j)(\mathbf{p}_i - \mathbf{p}_j)$$

$$\mathbf{L}\,\mathbf{p}' = \mathbf{b}\,, \quad (9)$$

as rigid as possible = minimize scale and rotation



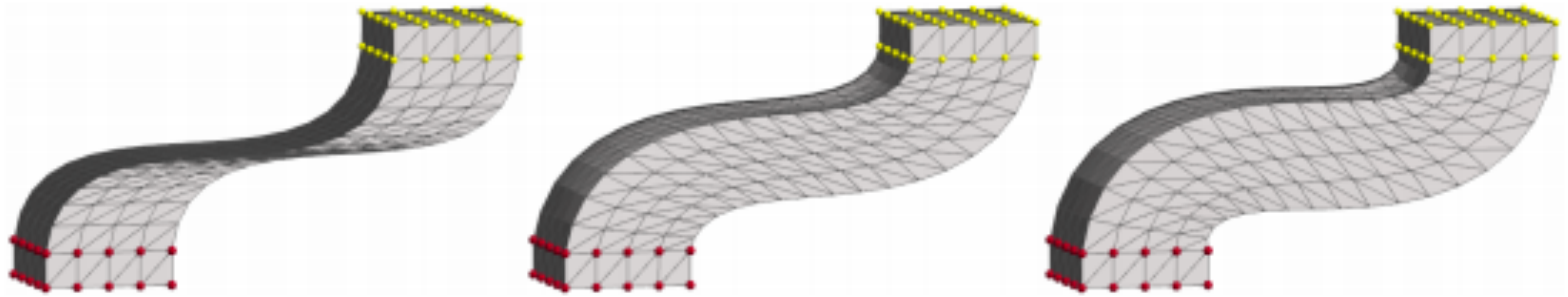
1. Compute Laplacian weight matrix
2. **Minimize the energy function**

$$E(\mathcal{C}_i, \mathcal{C}_i') = \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| (\mathbf{p}_i' - \mathbf{p}_j') - \mathbf{R}_i (\mathbf{p}_i - \mathbf{p}_j) \right\|^2 . \quad (3)$$

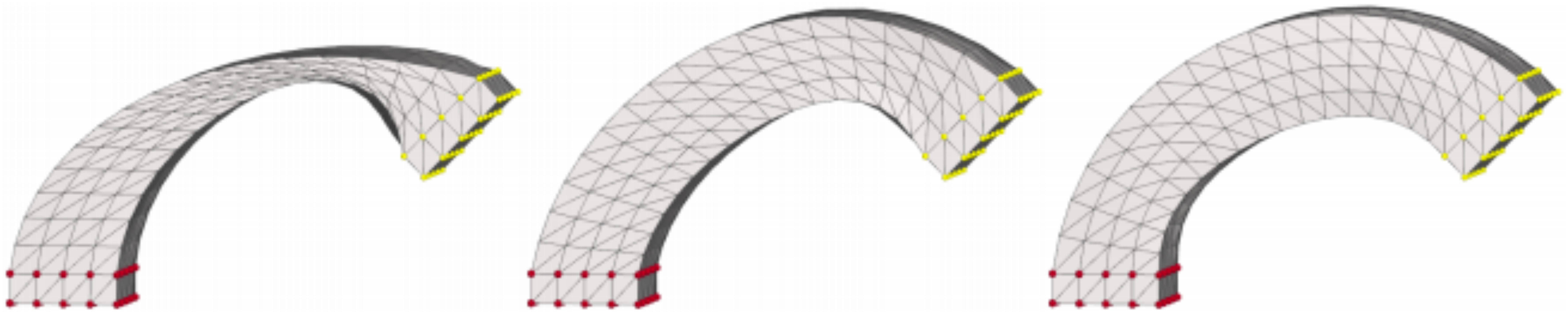
$$\frac{\partial E(\mathcal{S}')}{\partial \mathbf{p}_i'} = \sum_{j \in \mathcal{N}(i)} 4 w_{ij} \left((\mathbf{p}_i' - \mathbf{p}_j') - \frac{1}{2} (\mathbf{R}_i + \mathbf{R}_j) (\mathbf{p}_i - \mathbf{p}_j) \right)$$

$$\boxed{\begin{gathered} \sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{p}_i' - \mathbf{p}_j') = \sum_{j \in \mathcal{N}(i)} \frac{w_{ij}}{2} (\mathbf{R}_i + \mathbf{R}_j)(\mathbf{p}_i - \mathbf{p}_j) \\ \mathbf{L}\,\mathbf{p}' = \mathbf{b}, \quad (9) \end{gathered}}$$

solve linear equation iteratively

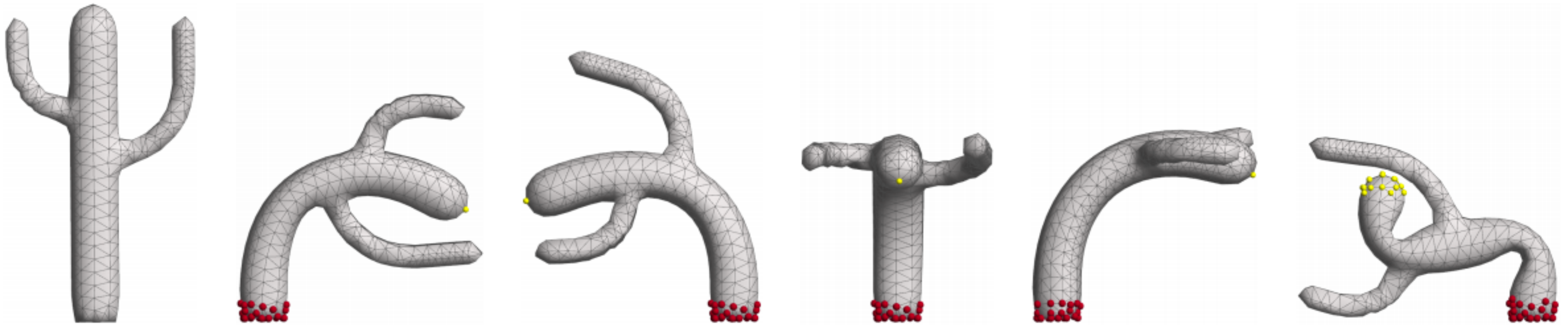


initial guess 1 iteration 2 iterations



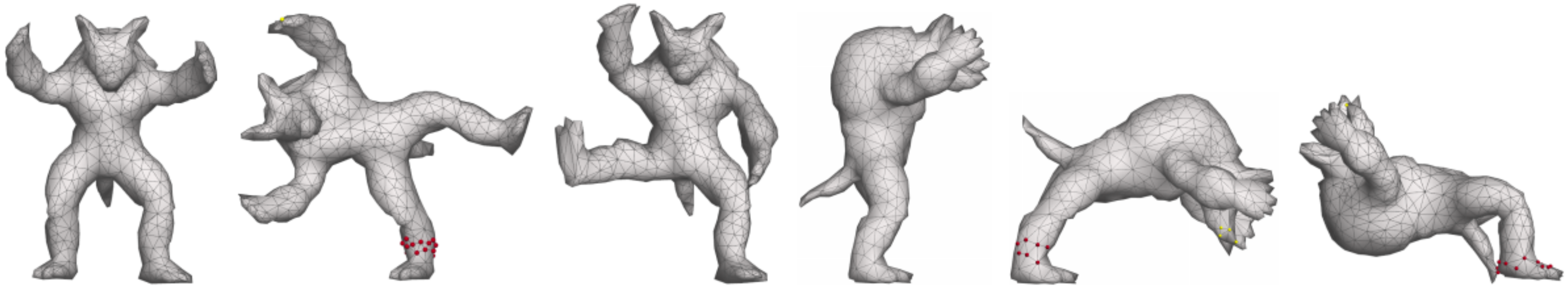
initial guess 1 iteration 4 iterations

Results



(a) (b) (c) (d) (e) (f)

Figure 7: *Bending the Cactus. (a) is the original model; yellow handles are translated to yield the results (b-f). (d) and (e) show side and front views of forward bending, respectively. Note that in (b-e) a* single vertex *at the tip of the Cactus serves as the handle, and the bending is the result of translating that vertex, no rotation constraints are given.*



(a) (b) (c) (d) (e) (f)

Figure 8: *Editing the Armadillo. (a) and (d) show views of the original model; the rest of the images display editing results, with the static and handle anchors denoted in red and yellow, respectively.*

github.com/ryosuzuki/cad-app