As-rigid-as-possible modelling

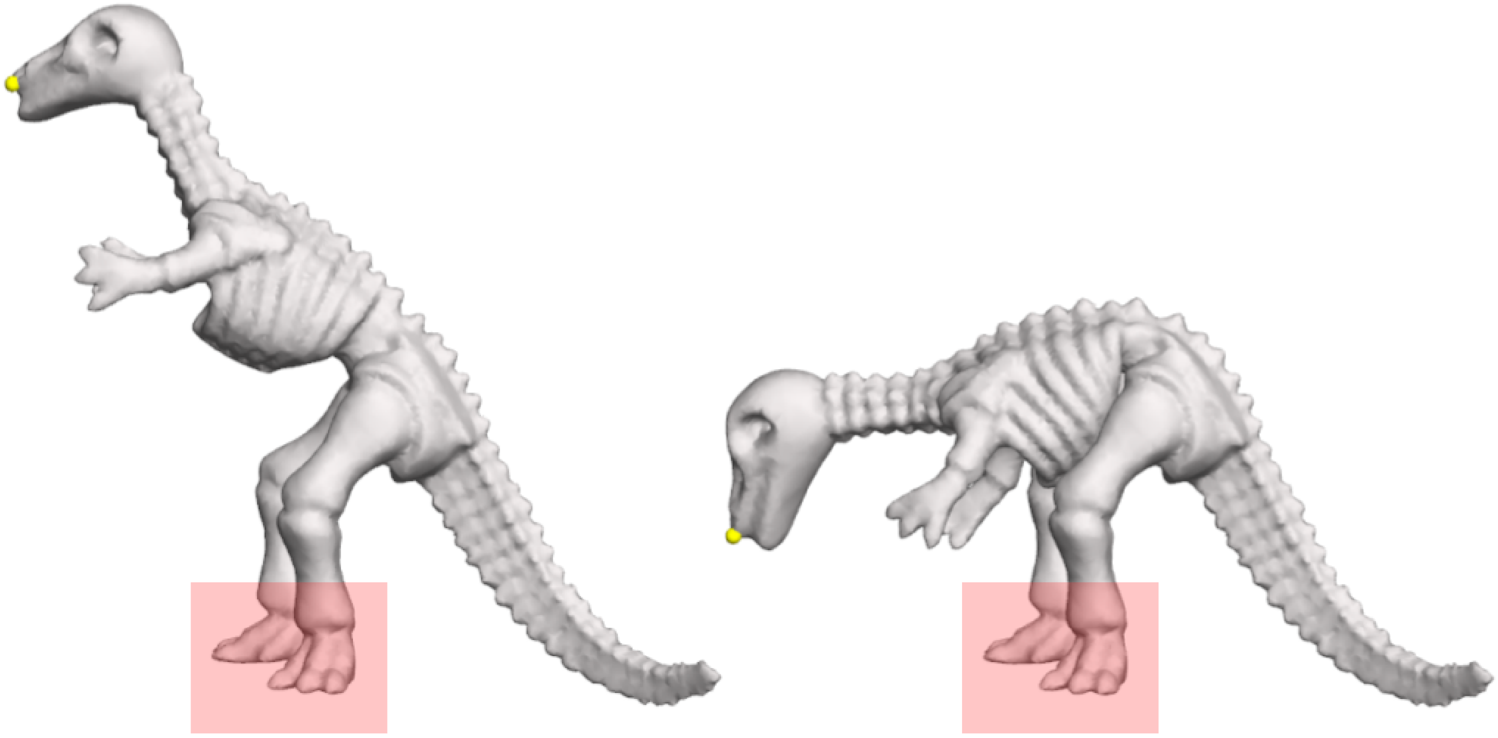
Why ARAP for today's lesson ?

- State of the art for shape direct manipulation
- Will allow me to show you how to setup a linear system (basics of numerical optimization)
- Will present the Procustes problem (basic problem in geometry)

[Sorkine & Alexa] : As-Rigid-As-Possible Surface Modeling

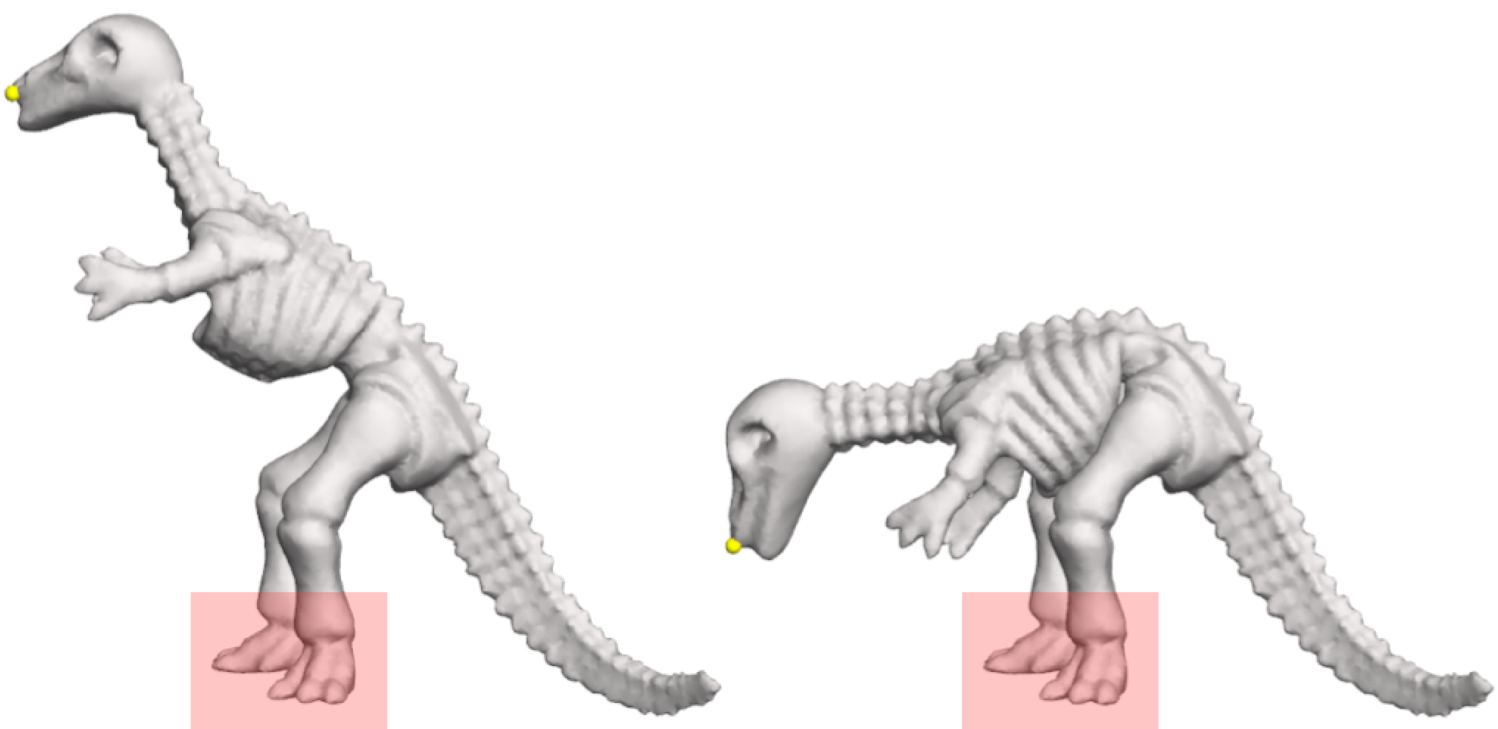
Principle

- A few vertex positions are specified by the user
- The other vertices should be placed in « a natural fashion »



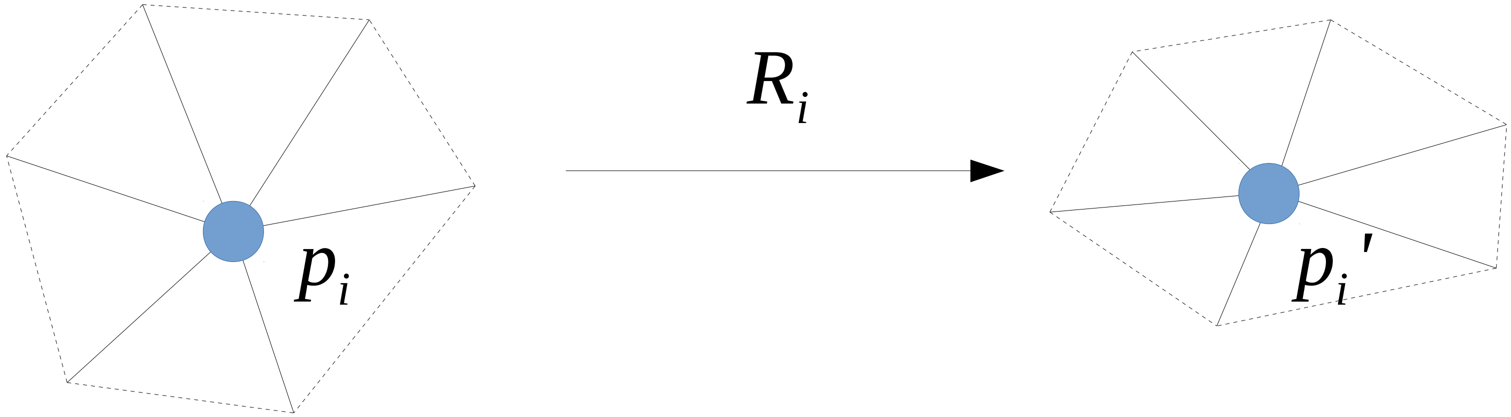
Minimizing stretch (enforcing rigidity)

- Physically « plausible »
- Impact on textures, stretch, volume



What is rigidity ?

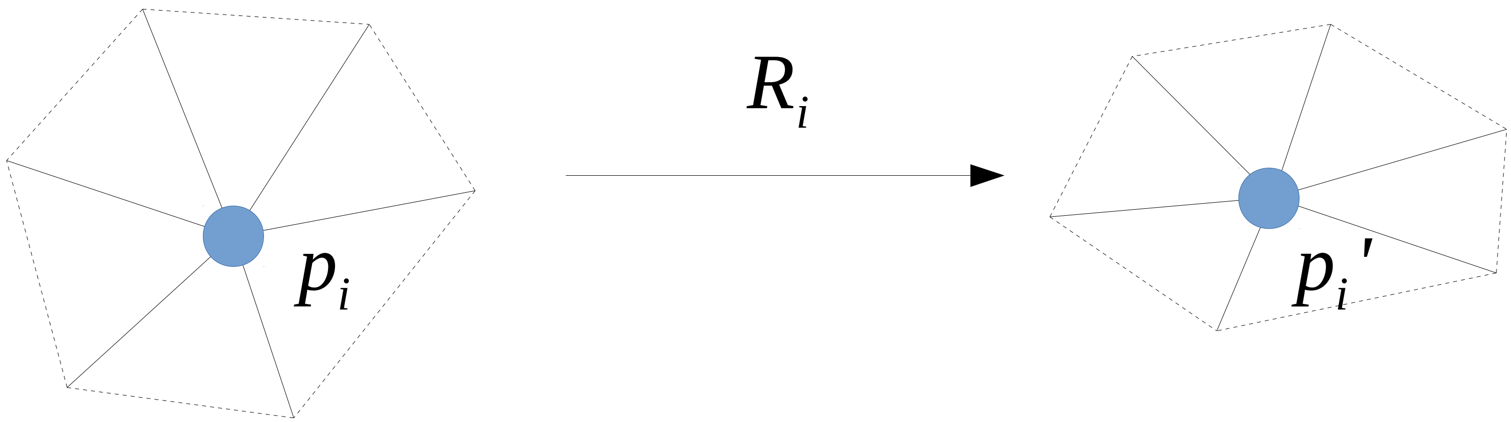
- A small piece of the shape should be globally rotated :



$$\mathbf{p}_i' - \mathbf{p}_j' = \mathbf{R}_i \left(\mathbf{p}_i - \mathbf{p}_j\right), \ \ \forall\ j \in \mathcal{N}(i)$$

Rigidity energy

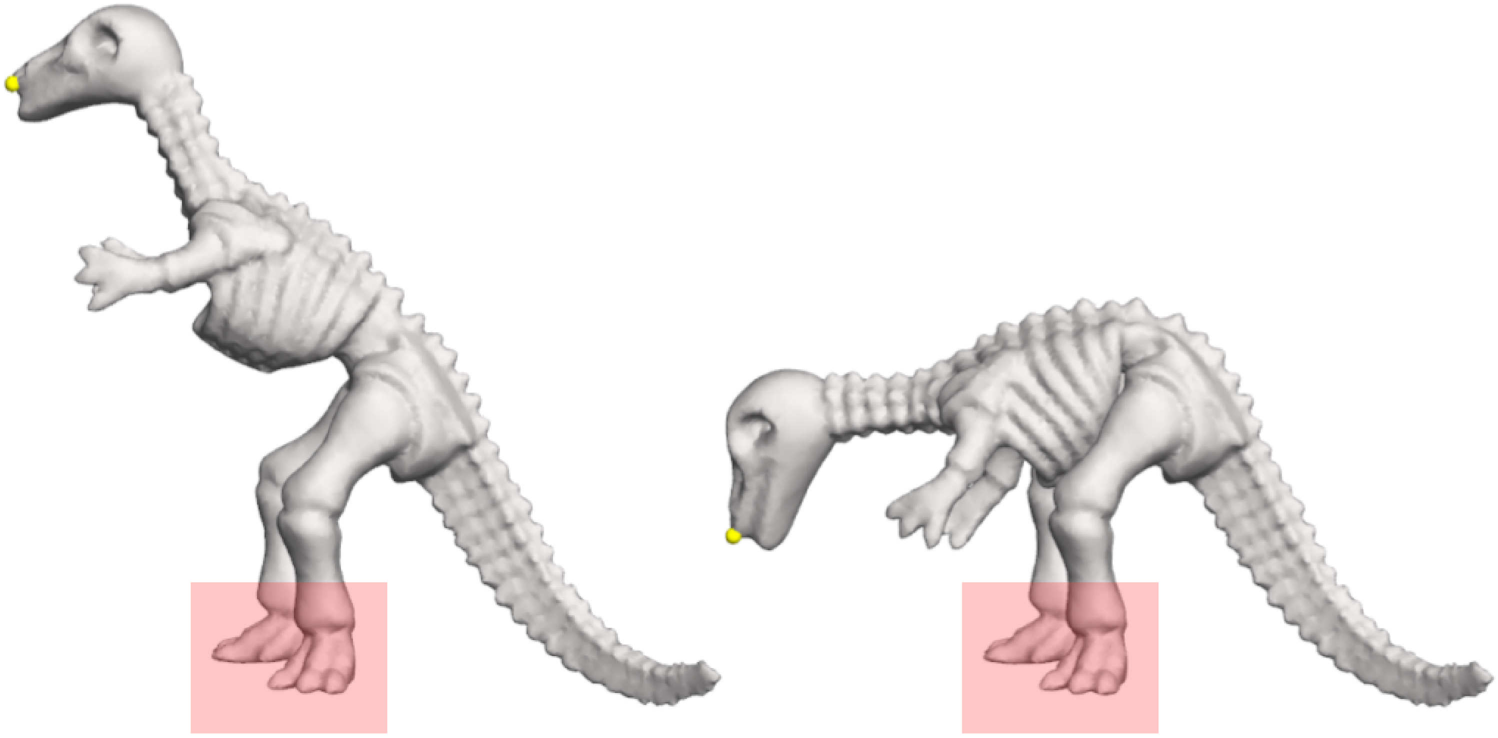
- A small piece of the shape should be globally rotated :



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

ARAP framework

- Unknowns : New positions p’ AND rotations R
- Constraints : A few specified positions



Problem

- Unknowns : New positions p’ AND rotations R
- Highly NON-LINEAR and NON-CONVEX
- Minimizing R and p’ at the same time is not feasible
- This is the rigidity energy alone, don’t forget to add the handle energies !

$$E\left(\mathcal{S}'\right) = \sum_{i=1}^{n} w_i \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| \left(\mathbf{p}'_i - \mathbf{p}'_j\right) - \mathbf{R}_i (\mathbf{p}_i - \mathbf{p}_j) \right\|^2$$

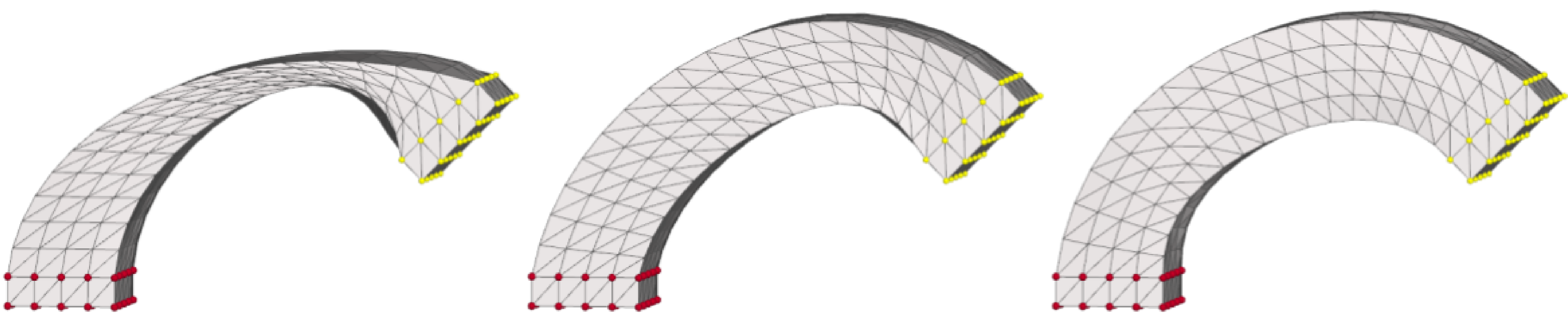
Ad hoc solution

- Fix R, optimize p’
- Fix p’, optimize R
- Fix R, optimize p’
- ...

$$E(\mathcal{S}') = \sum_{i=1}^{n} w_i \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| (\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{R}_i (\mathbf{p}_i - \mathbf{p}_j) \right\|^2$$

Ad hoc solution

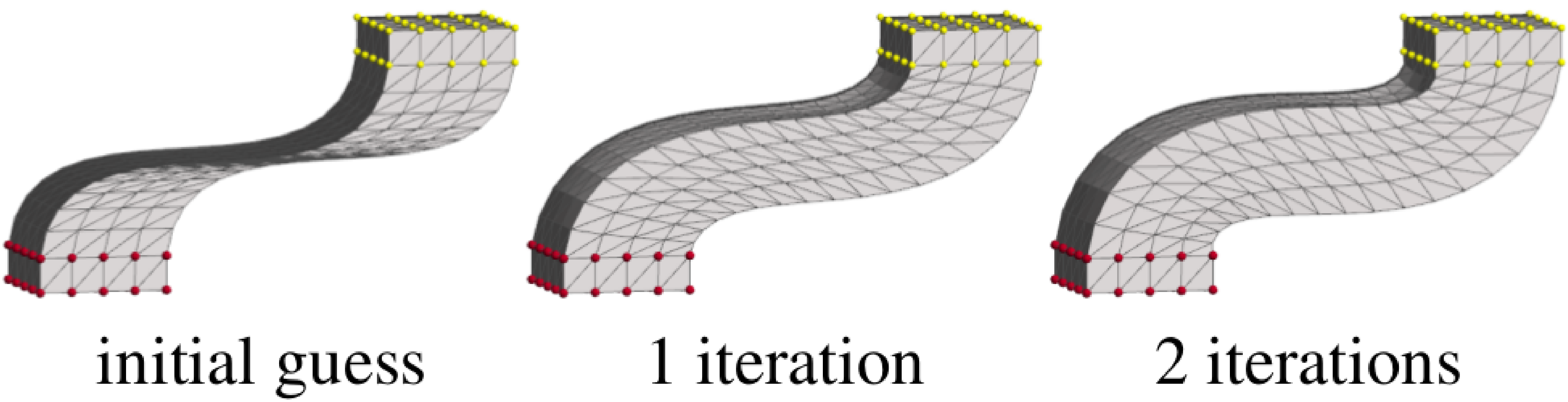
- Fix R, optimize p’
- Fix p’, optimize R
- Fix R, optimize p’
- ...



initial guess 1 iteration 4 iterations

Ad hoc solution

- Fix R, optimize p’
- Fix p’, optimize R
- Fix R, optimize p’
- ...



1) Fix R, optimize p’

- Linear system with p’ as unknowns
- Example on black board
- C++/Matlab : Cholmod, Eigen, ...
- Recall that ! Solving a linear system is the ABC of numerical optimization !

$$E\left(\mathcal{S}'\right) = \sum_{i=1}^{n} w_i \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| \left(\mathbf{p}'_i - \mathbf{p}'_j\right) - \mathbf{R}_i (\mathbf{p}_i - \mathbf{p}_j) \right\|^2$$

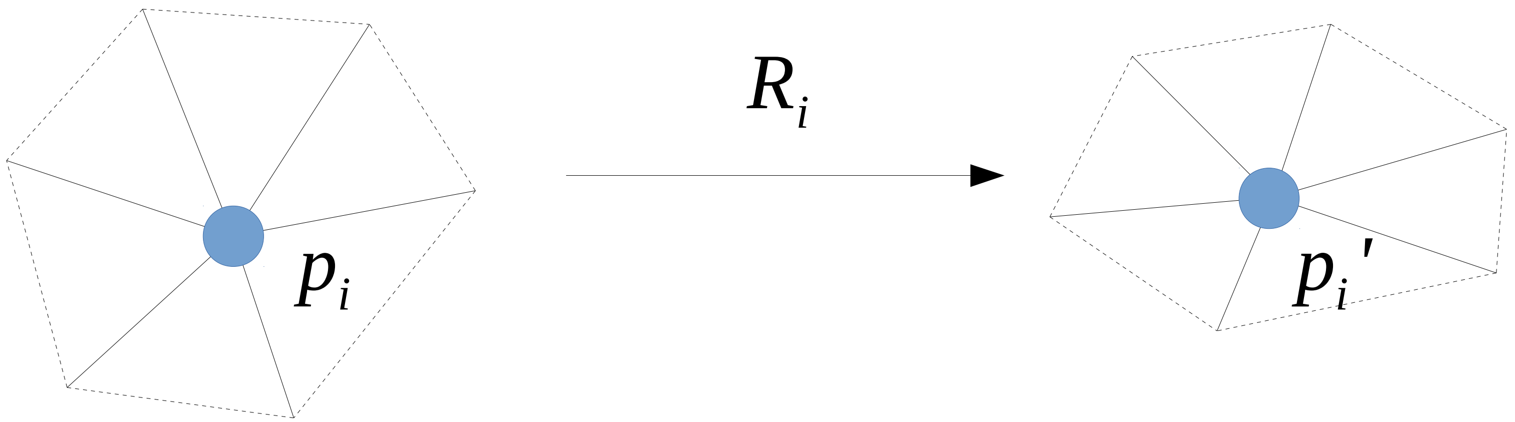
2) Fix p’, optimize R

- Can be done per vertex i

$$E\left(\mathcal{S}'\right) = \sum_{i=1}^{n} w_i \sum_{j \in \mathcal{N}(i)} w_{ij} \left\|\left(\mathbf{p}_i' - \mathbf{p}_j'\right) - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j)\right\|^2$$

2) Fix p', optimize R

- Can be done per vertex i
- Minimize E(Ci, Ci')



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

2) Fix p', optimize R

$$\sum_j w_j \| e_j' - R.e_j \|^2 \longrightarrow$$ To minimize

$$\sum_j w_j \| e_j' - R.e_j \|^2 = \sum_j w_j \| e_j' \|^2 + \sum_j w_j \| R.e_j \|^2 - 2 \sum_j w_j (R.e_j)^T . e_j'$$

$$\sum_j w_j \| e_j' - R.e_j \|^2 = const - 2 \sum_j w_j Trace(R.e_j.e_j'^T)$$

$$Trace(R.\sum_j w_j e_j.e_j'^T) \longrightarrow$$ To maximize

1) Build $S := \sum_j w_j e_j' . e_j^T$

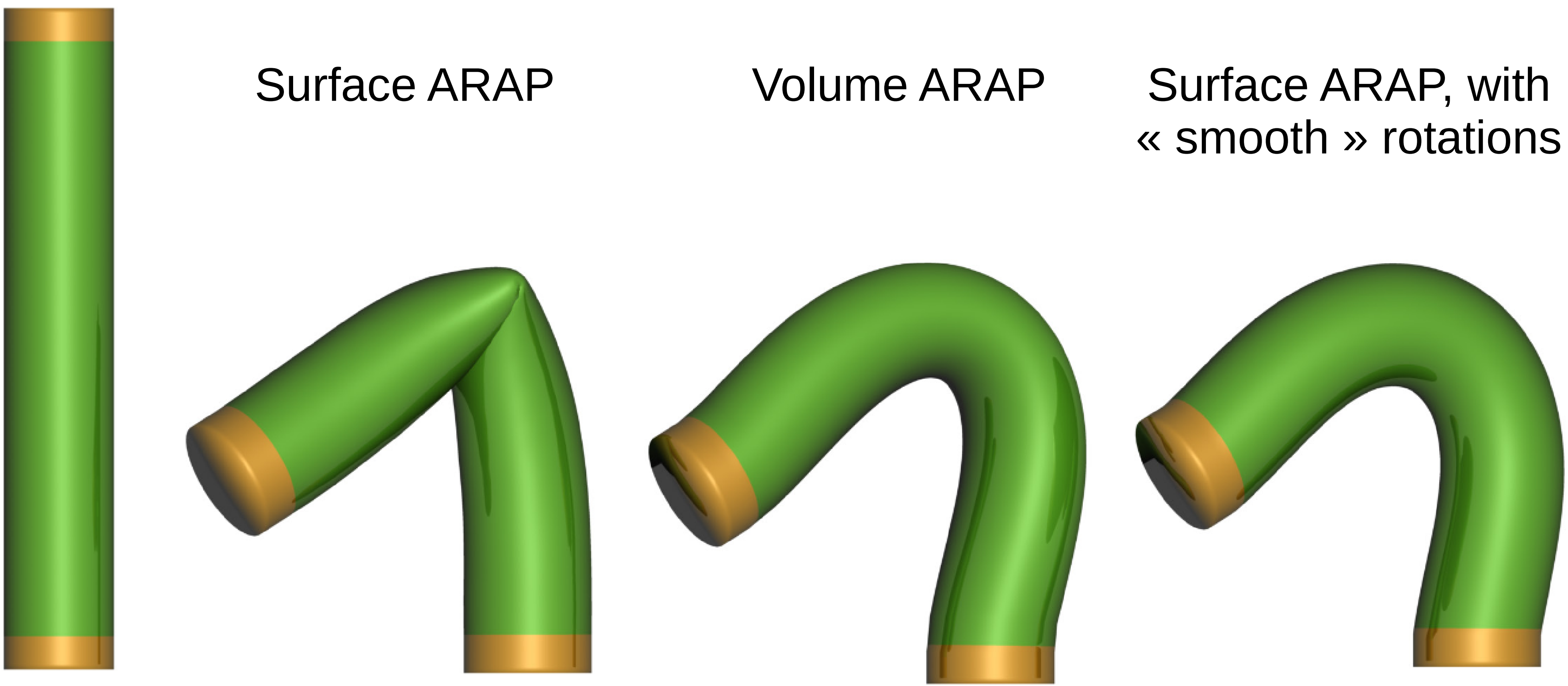
2) Compute SVD : $S := U.\Sigma.V^T$

3) Solution : $R := U.diag(1,1,det(U.V^T)).V^T$

At this point

- You know how to setup a linear system
- You know how to solve the Procustes problem
- You can implement ARAP in c++ with a few lines of code (something like 20 in Matlab)

Different flavours of ARAP



[Levi & Gotsman] : Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation