As-Rigid-As-Possible Surface Modeling

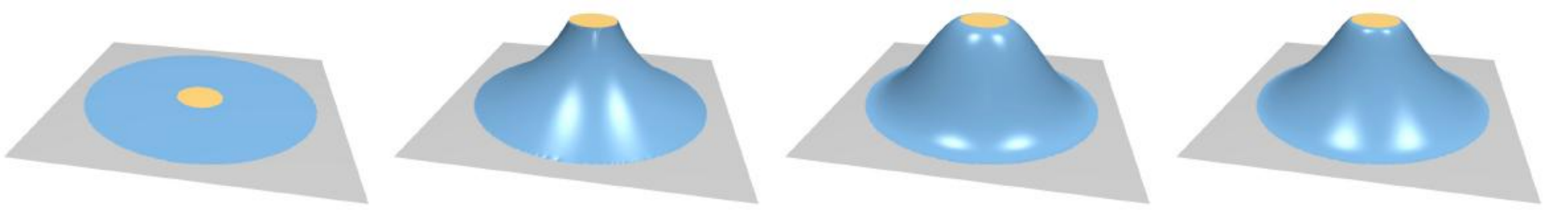
Olga Sorkine and Marc Alex

Presenter: Ming Chuang

Class: 600.657 Mesh Processing

Goal

- A shape modeling framework that supports intuitive and detail-preserving deformation.



[Botsch and Sorkine, 08]

Previous (surface-based) Works

- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:

Previous (surface-based) Works

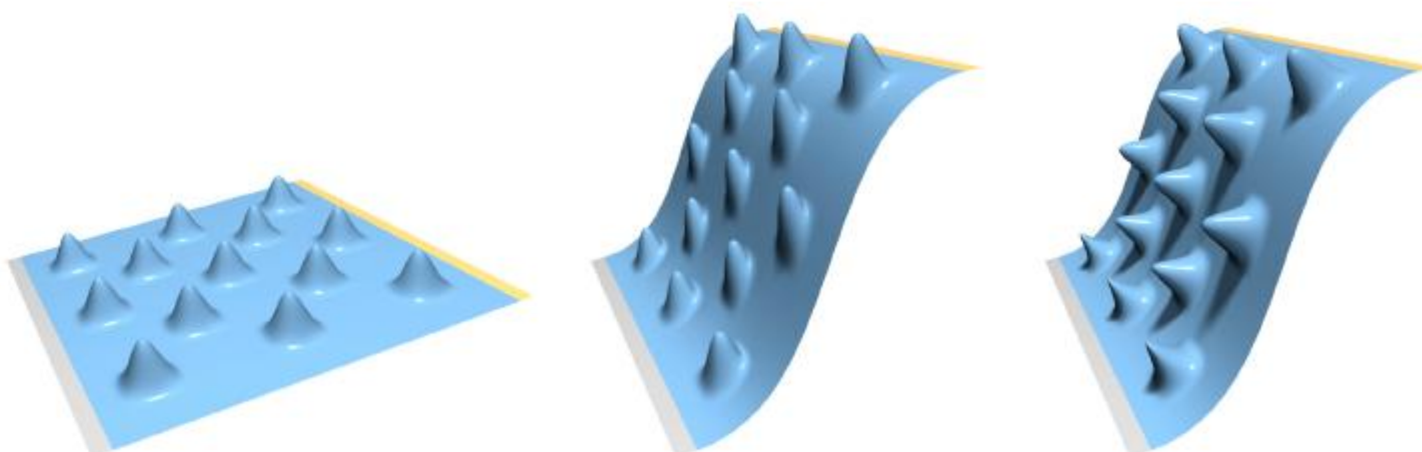
- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:
  - Physically-based methods that directly minimize stretching and/or bending energy

Previous (surface-based) Works

- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:
  - Physically-based methods that directly minimize stretching and/or bending energy
    - Problem: Not so detail-preserving
    - Remedy: Multi-resolution Editing

Previous (surface-based) Works

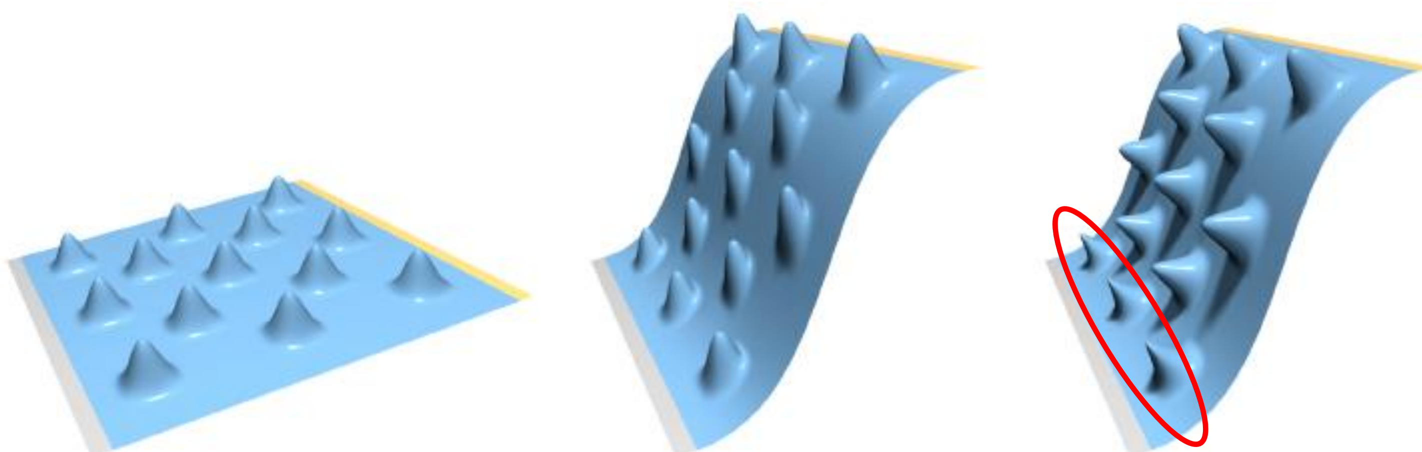
- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:
  - Physically-based methods that directly minimize stretching and/or bending energy



[Botsch and Sorkine, 08]

Previous (surface-based) Works

- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:
  - Physically-based methods that directly minimize stretching and/or bending energy



[Botsch and Sorkine, 08]

Previous (surface-based) Works

- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:
  - Physically-based methods that directly minimize stretching and/or bending energy
  - Differential Coordinates Editing methods that modify differential properties instead of spatial coordinate.

Previous (surface-based) Works

- For the sake of interactive frame-rates, most of previous works were based on **linear** methods. Two categories:
  - Physically-based methods that directly minimize stretching and/or bending energy
  - Differential Coordinates Editing methods that modify differential properties instead of spatial coordinate.
    - Still suffer from linearization…(interpolating either gradients or rotations)

Observation

- When we talk about “detail-preserving” deformation, only **rotation** and **translation** should be involved. Scaling and shearing should not be allowed, since they destroy local structures.

Observation

- When we talk about “detail-preserving” deformation, only **rotation** and **translation** should be involved. Scaling and shearing should not be allowed, since they destroy local structures.
- This motivates us to preserve local **rigidity** as much as possible.

Observation

- When we talk about “detail-preserving” deformation, only **rotation** and **translation** should be involved. Scaling and shearing should not be allowed, since they destroy local structures.
- This motivates us to preserve local **rigidity** as much as possible.
- The paper proposes to break a surface into overlapping **cells** and seek to keep the transformation in each cell rigid.

Measurement of Rigidity

- We define each cell covers the triangles incident upon a vertex (i.e. the one-ring neighborhood).
- Given the cell C_i corresponding to vertex *i* that deforms to C_i', we know there exists a rotation R_i' such that

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

  if the transformation of that cell is perfectly rigid.

Measurement of Rigidity

- This leads to an energy function that is minimized when the rigidity is maximized:

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

- Summing up energy functions from all cells, we obtain a global energy functional:

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

Minimization Scheme

- The energy functional is non-linear and depends only on p_i' : the new positions of the vertices.
- A two-stage, alternating scheme is proposed to decrease energy iteratively. This is achieved by separating the optimization of R_i (though it depends on p_i') from the optimization of p_i'

Stage 1: Optimization of R_i

- Given vertex positions, we would like to find optimal rigid transformations $\{R_i\}$

- Rewrite the energy function:

$$\sum_j w_{ij} \left(\mathbf{e}'_{ij} - \mathbf{R}_i \mathbf{e}_{ij}\right)^T \left(\mathbf{e}'_{ij} - \mathbf{R}_i \mathbf{e}_{ij}\right) =$$

$$= \sum_j w_{ij} \left(\mathbf{e}'^T_{ij}\,\mathbf{e}'_{ij} - 2\,\mathbf{e}'^T_{ij}\,\mathbf{R}_i\,\mathbf{e}_{ij} + \mathbf{e}^T_{ij}\,\mathbf{R}^T_i\,\mathbf{R}_i\,\mathbf{e}_{ij}\right) =$$

$$= \sum_j w_{ij} \left(\mathbf{e}'^T_{ij}\,\mathbf{e}'_{ij} - 2\,\mathbf{e}'^T_{ij}\,\mathbf{R}_i\,\mathbf{e}_{ij} + \mathbf{e}^T_{ij}\,\mathbf{e}_{ij}\right).$$

- Dropping constants then

$$\underset{\mathbf{R}_i}{\operatorname{argmin}} \sum_j -2 w_{ij}\,\mathbf{e}'^T_{ij}\,\mathbf{R}_i\,\mathbf{e}_{ij} = \underset{\mathbf{R}_i}{\operatorname{argmax}} \sum_j w_{ij}\,\mathbf{e}'^T_{ij}\,\mathbf{R}_i\,\mathbf{e}_{ij} =$$

$$= \underset{\mathbf{R}_i}{\operatorname{argmax}}\ Tr\left(\sum_j w_{ij}\,\mathbf{R}_i\,\mathbf{e}_{ij}\,\mathbf{e}'^T_{ij}\right) =$$

$$= \underset{\mathbf{R}_i}{\operatorname{argmax}}\ Tr\left(\mathbf{R}_i \underbrace{\sum_j w_{ij}\,\mathbf{e}_{ij}\,\mathbf{e}'^T_{ij}}_{S_i}\right).$$

- $Tr(R_i S_i)$ can be maximized by making $R_i S_i$ symmetric PSD via SVD

Stage 2: Optimization of P_i'

- Given $\{R_i\}$, we would like to find optimal vertex positions P_i'

- Differentiating the energy function, we have

$$\begin{aligned}\frac{\partial E(\mathcal{S}')}{\partial \mathbf{p}'_i} &= \frac{\partial}{\partial \mathbf{p}'_i}\left(\sum_{j\in\mathcal{N}(i)} w_{ij}\left\|(\mathbf{p}'_i-\mathbf{p}'_j)-\mathbf{R}_i(\mathbf{p}_i-\mathbf{p}_j)\right\|^2 + \right.\\ &+ \left.\sum_{j\in\mathcal{N}(i)} w_{ji}\left\|(\mathbf{p}'_j-\mathbf{p}'_i)-\mathbf{R}_j(\mathbf{p}_j-\mathbf{p}_i)\right\|^2\right) = \\ &= \sum_{j\in\mathcal{N}(i)} 2w_{ij}\left((\mathbf{p}'_i-\mathbf{p}'_j)-\mathbf{R}_i(\mathbf{p}_i-\mathbf{p}_j)\right) + \\ &+ \sum_{j\in\mathcal{N}(i)} -2w_{ji}\left((\mathbf{p}'_j-\mathbf{p}'_i)-\mathbf{R}_j(\mathbf{p}_j-\mathbf{p}_i)\right).\end{aligned}$$

- Since the weights are symmetric

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

- Setting it to zero, we achieve:

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

- Differentiating the energy function, we have

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

- Since the weights are symmetric

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

- Setting it to zero, we achieve: *L p' = b*

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

Discussion

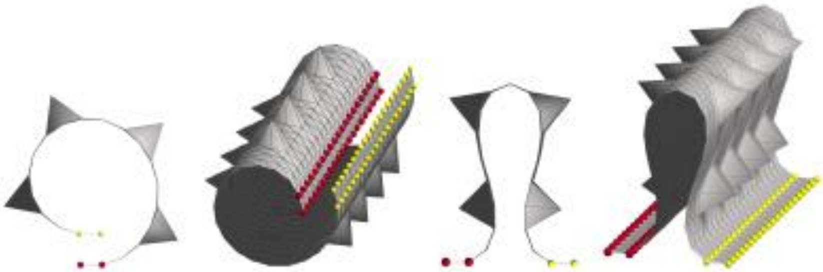
- The first stage involves solving a series of (3x3) SVD problem. The complexity is linear.
- The second stage involves solving a Poisson equation. It is important to note that the system matrix never changes, only the constraint does.
- This suggests pre-factorization of the system. Though expensive to compute, it is an one-time task.

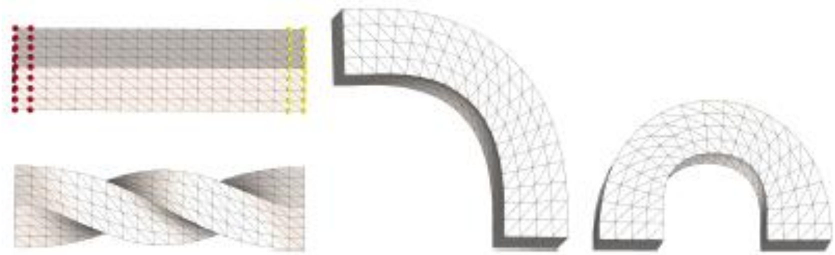
Discussion

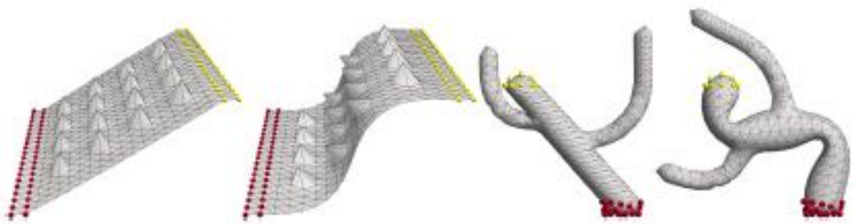
- Cholesky factorization is used to decompose the system matrix into two triangle matrices.
- As a result, solving for a new constraint amounts to 3(?) times of back-substitution, which has a quadratic(!?) complexity.

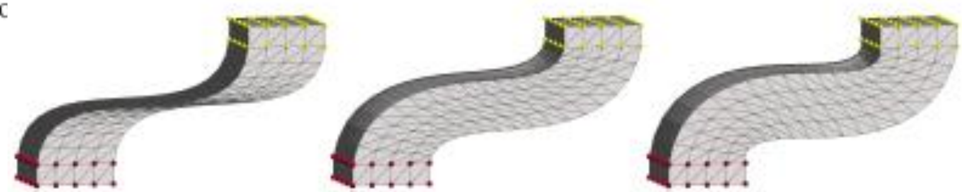
Discussion

- Cholesky factorization is used to decompose the system matrix into two triangle matrices.
- As a result, solving for a new constraint amounts to 3(?) times of back-substitution, which has a ~~quadratic(!?)~~ complexity.
  - Note that the matrix is sparse. In practice, it is permuted first to ensure the resulting factorization will be sparse too. So the complexity will be linear in the number of vertices.






Poisson
our method
Poisson
our metho


initial guess
1 iteration
2 iterations


initial guess
1 iteration
4 iterations

Conclusion

- The paper proposes a shape deformation modeling framework that preserves local rigidity as much as possible (and thus detail-preserving).
- The advantages of the approach are
  1. Robustness(?): guaranteed to converge to something…(a good initial guess is important!)
  2. Simplicity: easy to implement
  3. Efficiency: able to pre-factorize the system
- The visual results are comparable to the fully non-linear technique [Botsch et al., 07]