Mesh Deformation Algorithms

Vraj Thakkar

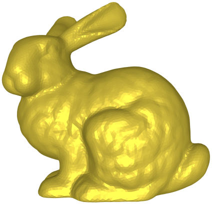
November 2024

Table of Contents

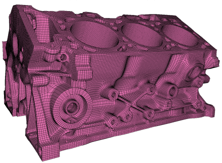
Table of Contents

What is a mesh and how do we represent it?

- A mesh is a discretized version of a surface that we can represent on our computer.
- For this presentation we will be using triangle mesh representation.
- There are other ways to represent meshes using polygon meshes, signed distance function etc.



(a) Triangular mesh of a bunny



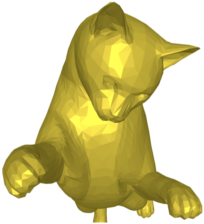
(b) Quad mesh representation(geometryfactory.com)

Figure: Illustrations of surfaces represented as polygonal meshes

Triangular Mesh

- To represent a surface as a triangular mesh we need to store geometry as well as topological information.
- Any triangular mesh M is represented by coordinates $V_{n\times 3} \in \mathbb{R}^3$ and faces as $F_{n\times 3} = \{(i,j,k)\}$ where i,j,k are the indices of the respective vertices and $i \neq j \neq k$ for any face.

(a) Triangular mesh of a cat

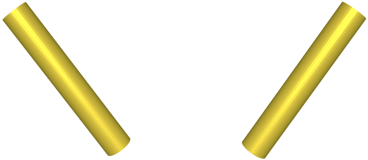


(b) Triangular mesh of the cat

Figure: Illustrations of surfaces represented as triangular meshes

Rigid Transformations

- Transformations that preserve the Euclidean distance between any two points are called rigid transformations. eg. rotation and translation.
- We are going to talk about rotation matrices which are a subset of 3×3 matrices such that $det(R) = 1$ and $RR^T = I$.
- These properties imply that these transformations preserve angles and lengths.
- $||Rv|| = \sqrt{(Rv)^T Rv} = \sqrt{v^T R^T Rv} = \sqrt{v^T v}$

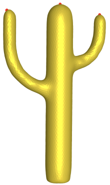


(a) Bar (b) Bar rotated

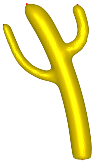
Figure: Illustrations of surfaces represented as polygonal meshes

How do we deform a mesh?

- We are given a mesh $M(V, F)$, few points $v_h \in V, h \in \{1, 2, 3...m\}$ as handle points and transformations for each handle point point $T_h, h \in \{1, 2, 3...m\}$.
- We want a mesh that contains the transformed vertices and preserves the object's shape as best as possible.



(a) Cactus with handle points(red)



(b) Cactus with rotation of $\pi/2$ on head

Figure: Mesh Deformation using Linear Blend Skinning

Skinning-based methods

- A simple solution is to do a weighted average of the transformations of each handle point.
- We will need weights to interpolate transformations at the handle points.

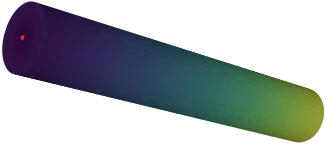


Figure: Inverse distance weights w.r.t handles

Examples

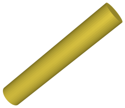


Figure: Cylinder mesh with rotations of 0 and π on handles

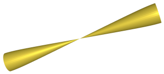


Figure: Linear Blend Skinning shows candy wrapper effect

Table of Contents

As-Rigid-As-Possible(ARAP) surface modelling

- Skinning-based methods are fast but often fail to give intuitive results.
- We can use physics-based deformation methods but they are very slow for tasks like animation.
- ARAP is an optimization-based iterative method that gives good results.

As-Rigid-As-Possible(ARAP) surface modelling

- Skinning-based methods are fast but often fail to give intuitive results.
- We can use physics-based deformation methods but they are very slow for tasks like animation.
- ARAP is an optimization-based iterative method that gives good results.
- The key idea for ARAP is to get a global non-rigid transformation via locally rigid transformations!

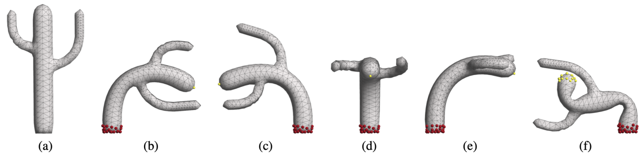


Figure: O. Sorkine M.Alexa, As-Rigid-As-Possible Surface Modeling(2004)

Deforming surfaces using ARAP

ARAP is best understood as a two-step process.

- Locally, we make the transformations as rigid as possible.
- These transformations cannot be directly applied on the local neighborhood because the surface can just break!
- So the second step is that we use these local rotations to get a connected surface.

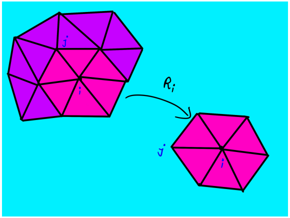


Figure: Local rigid transformation may not preserve mesh structure

ARAP energy function

- We want to find a rotation $\mathbf{R}_i$ for the cell(1-ring neighborhood) transformation $\mathcal{C}_i \to \mathcal{C}'_i$ such that $(\mathbf{p}'_i - \mathbf{p}'_j) = \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j),\ \forall j \in \mathcal{N}(i)$
- $E\left(\mathcal{C}_i, \mathcal{C}'_i\right) = \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| \left(\mathbf{p}'_i - \mathbf{p}'_j\right) - \mathbf{R}_i\left(\mathbf{p}_i - \mathbf{p}_j\right) \right\|^2$
- It may be possible that no rotation matrix describes the transformation so we do a least squares.

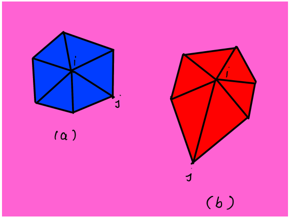


Figure: There is no rotation matrix possible

How to find rotations?(Local Step)

- We want to minimize the energy function $E\left(\mathcal{C}_i, \mathcal{C}_i'\right) = \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| \left(\mathbf{p}_i' - \mathbf{p}_j'\right) - \mathbf{R}_i \left(\mathbf{p}_i - \mathbf{p}_j\right) \right\|^2$ over $\mathbf{R}_i$ such that $\mathbf{R}_i \mathbf{R}_i^T = \mathbf{I}$
- Putting $\mathbf{p}_i' - \mathbf{p}_j' = \mathbf{e_{ij}'}$ and $\mathbf{p}_i - \mathbf{p}_j = \mathbf{e_{ij}}$ and removing the terms which do not contain $\mathbf{R_i}$ we can reduce the problem to

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

How to find rotations?(Local Step)

- Putting X and Y as the matrices with $\mathbf{e}_{ij}$'s and $\mathbf{e}'_{ij}$'s as row vectors we have

$$\underset{\mathbf{R}_i}{\operatorname{argmax}} \operatorname{Tr}\left(\mathbf{R}_i X W Y^T\right)$$

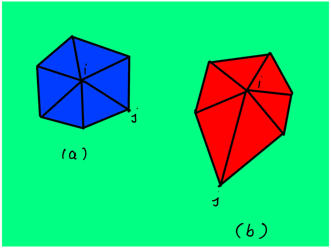
- $S = XWY^T$ is a covariance matrix and we want to find R_i such that it is maximally aligned with the basis of S.
- We do a Singular Value Decomposition of $S = U\Sigma V^T$. Rearranging the matrices we have

$$\underset{\mathbf{R}_i}{\operatorname{argmax}} \operatorname{Tr}\left(\Sigma V^T \mathbf{R}_i U\right)$$

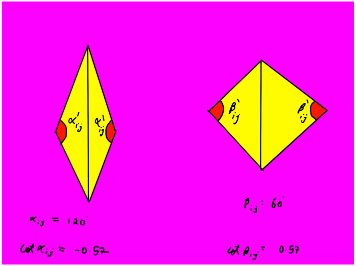
- The above equation is maximized when $\mathbf{R}_i = VU^T$.

Edge weights

- The cell weights come from the discrete Laplace-Beltrami Operator on triangle meshes. For this case, we are using the cotan Laplacian for defining the cell weights as $w_{ij} = \frac{1}{2A_i}(cot\alpha_{ij} + cot\beta_{ij})$



(a) Equally sampled[Left], Skew sampled mesh cells[Right]



(b) w_{ij} calculation

Figure: Illustrations of weight calculations and mesh sampling.

Voronoi Area

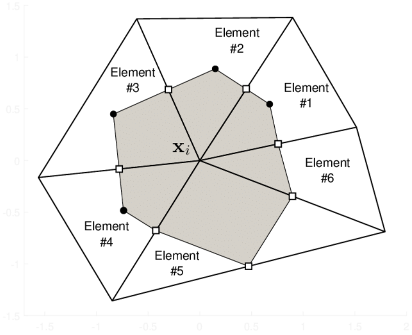
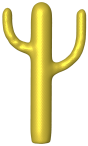


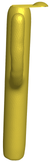
Figure: Voronoi area is the shaded region, reference:- Sudip Basu, researchgate.net

Alternating minimization in ARAP

- We don't know either $\mathbf{p_i'} - \mathbf{p_j'}$ or $\mathbf{R_i}$ which minimize the energy function.
- The Only thing we have is the mesh and the new positions of the handle points.
- Start with the new mesh $M'(V', F)$
  - In this mesh $V_h' = \textit{new positions}\ \forall h = \{1, 2, 3..m\}$, for all other vertices $V' = V$.
  - Find rotations for each cell in the mesh M using V and V'.
  - Using rotations find the new V''. Now repeat.



(a) Original Mesh M



(b) M' for first iteration

Finding new coordinates given rotations(Global Step)

- To make the transformations of the whole mesh as rigid as possible we want to minimize the energy function $E\left(\mathcal{C}_i, \mathcal{C}_i'\right)$ for each cell in the mesh.
- We have the new energy function over the whole mesh as

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

- Cells with larger areas are more important hence $w_i = A_i \ \forall i$, where A_i is the Voronoi area of the cell $\mathcal{C}_i'$.

Finding new coordinates given rotations(Global Step)

- We find the gradient of $E(\mathcal{S}')$ w.r.t the points $\mathbf{p}'$ to find the optimal vertices given rotations.

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

Finding new coordinates given rotations

- We also have $w_{ij} = w_{ji}$, therefore

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

- Setting the partial derivatives to zero w.r.t. each $\mathbf{p}'_i$

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

- This left side of the equation is the **discrete Laplace-Beltrami Operator** applied to p'_i. We can we write the equations for all vertices as

$$\mathbf{L}\mathbf{p}' = \mathbf{b}$$

ARAP at a glance

- Compute the Cotan Laplacian $\mathbf{L}$ for the mesh.
- Using the original mesh M and M' with change at handle points, find rotation matrices for each cell.
- Use these rotation matrices to find $\mathbf{b}$.
- Solve the system of equations $\mathbf{L}\mathbf{p}' = \mathbf{b}$ for new vertices $\mathbf{p}'$.
- Use $\mathbf{p}'$ as the new mesh M' for second iteration and repeat the steps.

Results

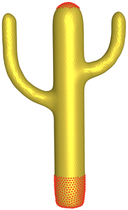


Figure: Cactus mesh

Figure: Deformation applied

Results

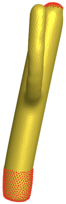


Figure: 5th iteration of ARAP

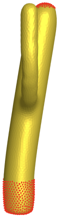


Figure: 10th iteration of ARAP

Table of Contents

ARAP variations and Future Work

- Mesh deformation is not enough, for real-world applications we need to respect the information of the material's rigidity to get intuitive deformations.
- Darshil Patel, Material based Mesh Deformation, 2022.

$$E\left(\mathcal{C}_k, \mathcal{C}'_k\right) = \sum_{(i,j)\in\mathcal{E}(k,r)} w_{ij} \left\| \left(\mathbf{p}'_i - \mathbf{p}'_j\right) - \mathbf{T_k}\left(\mathbf{p}_i - \mathbf{p}_j\right) \right\|_2^2 + \lambda \left\| \mathbf{T_k}^{\mathbf{T}}\mathbf{T_k} - \mathbf{I} \right\|_F^2$$

References

- Olga Sorkine and Marc Alexa. (2007). As-rigid-as-possible surface modeling.
- Sorkine, O., Sharf, A., Cohen-Or, D., Lischinski, D. (2004). Least-Squares Rigid Motion Using SVD.
- Chen, Shu-Yu, et al. (2013). Rigidity Controllable As-Rigid-As-Possible Shape Deformation.

Thank You

Appendix

- We don't know how to define Laplacian on a surface because it is a 2-d object embedded in a 3-d space.
- We can derive the Laplacian on a surface using the gradient of the Dirichlet energy(total surface area) and as it turns out it is an intrinsic property!
- It turns out Laplacian of a surface at a point is the mean curvature normal vector $2H\hat{n}$.
- It depends only on the first fundamental form implying that it is invariant to isometric deformations.
- So, the Laplacian of a surface becomes a natural choice for the weights because we want the vertices to be governed by an intrinsic property that is invariant to deformations(isometric in this case).

Appendix

- Even more simply the $\mathbf{L}$ in $\mathbf{L}\mathbf{p}' = \mathbf{b}$ encodes information about local neighbourhood of vertices which is invariant to isometric deformations.
- $\sum_j w_{ij}(\mathbf{p'}_i - \mathbf{p'}_j) = b$ here even though $\mathbf{p'_k}$'s are changing but they have to change such that the Laplacian of p_i is preserved.
- Equally sampled mesh means the edge lengths and angles do not change much.
- Why use Voronoi instead of the normal area? because if the triangles are thin and long, the area will still be large!! circumcenter because it is equidistant from all vertices and other interpretations in computational geometry relating to delaunay triangulations.
- $M = V^T RU$ has orthonormal row vectors and will maximize $tr(\Sigma M)$ only when $M = I$.

Appendix

- $E(f) = \int_\Omega \|\nabla f\|^2 dA$, it can be shown that: f minimizes E if $\Delta f = 0$.

Figure: Laplacian is the steady state is preserved[1]

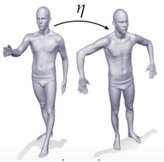


Figure: Isometric deformations[2]

1 2

[1] IGS'16 Summer School: Laplace-Beltrami: The Swiss Army Knife of Geometry Processing,Etienne Vouga (UT Austin)

[2] Lecture 18: The Laplace Operator (Discrete Differential Geometry), Keenan Crane(CMU)