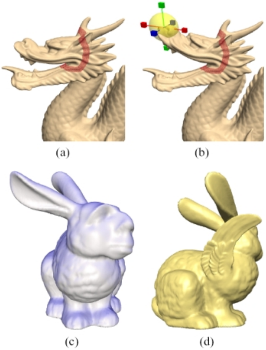
Skeleton Based As-Rigid-As-Possible Volume Modeling

Shaoting Zhang, Andrew Nealen and Dimitris Metaxas

Computer Science Department, Rutgers University

Introduction

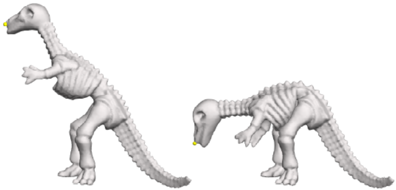
- As-rigid-as-possible (ARAP) shape modeling is a popular technique to obtain natural deformations. There have been many excellent methods.



- Olga Sorkine, et al,: Laplacian Surface Editing. SGP2004

Introduction

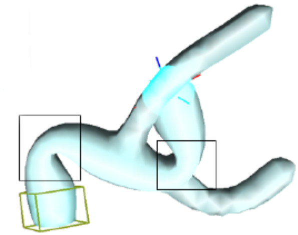
- As-rigid-as-possible (ARAP) shape modeling is a popular technique to obtain natural deformations. There have been many excellent methods.



- Olga Sorkine, et al,: Laplacian Surface Editing. SGP2004
- Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007

Introduction

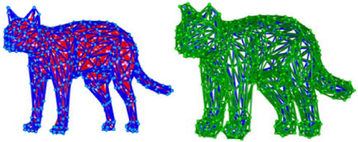
- As-rigid-as-possible (ARAP) shape modeling is a popular technique to obtain natural deformations. There have been many excellent methods.
- We are interested in the volume preservation.

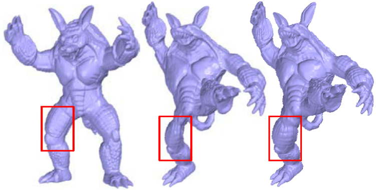


- Olga Sorkine, et al,: Laplacian Surface Editing. SGP2004
- Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007

Introduction

- As-rigid-as-possible (ARAP) shape modeling is a popular technique to obtain natural deformations. There have been many excellent methods.
- We are interested in the volume preservation.
- VGL is a good approach. However, we do not want to break the manifoldness of ARAP surface modeling or sacrificing the speed.

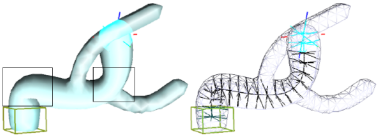




- Olga Sorkine, et al,: Laplacian Surface Editing. SGP2004
- Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
- Kun Zhou, et al,: Large mesh deformation using the volumetric graph Laplacian. SIGGRAPH 2005

Introduction

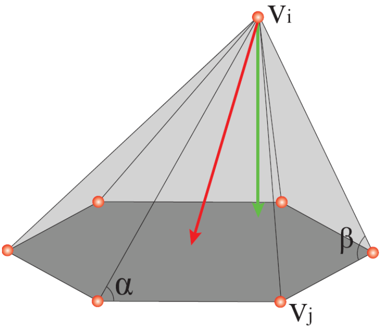
- As-rigid-as-possible (ARAP) shape modeling is a popular technique to obtain natural deformations. There have been many excellent methods.
- We are interested in the volume preservation.
- VGL is a good approach. However, we do not want to break the manifoldness of ARAP surface modeling or sacrificing the speed.
- We do it by leveraging the skeleton information.



- Olga Sorkine, et al,: Laplacian Surface Editing. SGP2004
- Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
- Kun Zhou, et al,: Large mesh deformation using the volumetric graph Laplacian. SIGGRAPH 2005

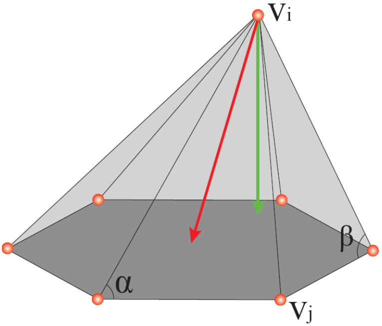
Linear LSE with C^0 continuity

- Laplacian coordinates represent each point as the weighted difference between such point and its neighborhoods.



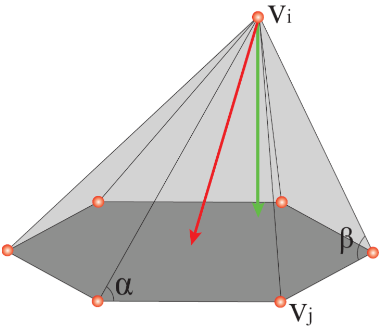
Linear LSE with C^0 continuity

- Laplacian coordinates represent each point as the weighted difference between such point and its neighborhoods.
- Given original coordinates V, the connectivity, and m control points, the reconstructed object V' can be obtained by minimizing: $\|LV' - \delta\|_2^2 + \Sigma_{i=1}^{m}\|v'_{c_i} - v_{c_i}\|_2^2$



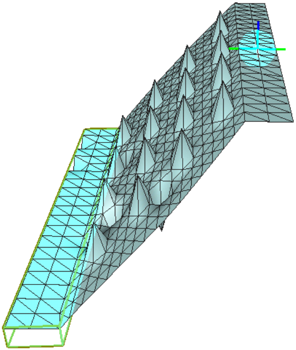
Linear LSE with C^0 continuity

- Laplacian coordinates represent each point as the weighted difference between such point and its neighborhoods.
- Given original coordinates V, the connectivity, and m control points, the reconstructed object V' can be obtained by minimizing: $\|LV' - \delta\|_2^2 + \Sigma_{i=1}^{m}\|v'_{c_i} - v_{c_i}\|_2^2$
- $\begin{bmatrix} L \\ l_c \end{bmatrix} V' = \begin{bmatrix} \delta \\ V_c \end{bmatrix}$



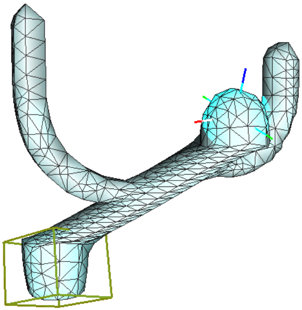
Linear LSE with C^0 continuity

- Laplacian coordinates represent each point as the weighted difference between such point and its neighborhoods.
- Given original coordinates V, the connectivity, and m control points, the reconstructed object V' can be obtained by minimizing: $\|LV' - \delta\|_2^2 + \Sigma_{i=1}^{m}\|v'_{c_i} - v_{c_i}\|_2^2$
- $\begin{bmatrix} L \\ I_c \end{bmatrix} V' = \begin{bmatrix} \delta \\ V_c \end{bmatrix}$
- When rotations are large, the deformation may not be natural.



Linear LSE with C^0 continuity

- Laplacian coordinates represent each point as the weighted difference between such point and its neighborhoods.
- Given original coordinates V, the connectivity, and m control points, the reconstructed object V' can be obtained by minimizing: $\|LV' - \delta\|_2^2 + \Sigma_{i=1}^{m}\|v'_{c_i} - v_{c_i}\|_2^2$
- $\begin{bmatrix} L \\ I_c \end{bmatrix} V' = \begin{bmatrix} \delta \\ V_c \end{bmatrix}$
- When rotations are large, the deformation may not be natural.



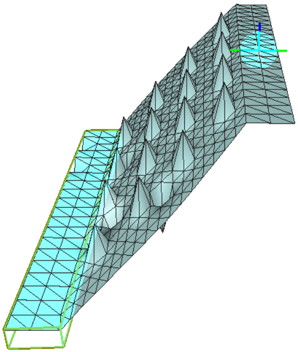
Rotation and edge length constraints

- Iterate two steps to recover rotations:

Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
Olga Sorkine: Least-Squares Rigid Motion Using SVD

Rotation and edge length constraints

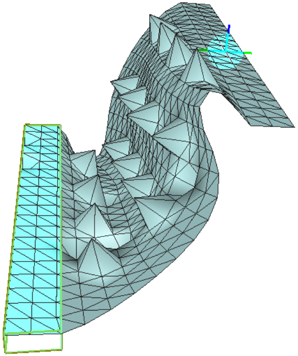
- Iterate two steps to recover rotations:
- Step1: Initial guess from solving naive LSE.



Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
Olga Sorkine: Least-Squares Rigid Motion Using SVD

Rotation and edge length constraints

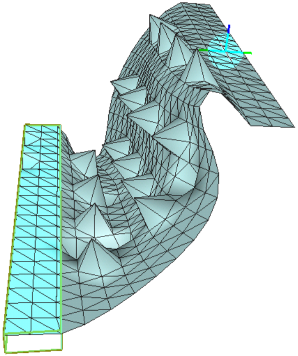
- Iterate two steps to recover rotations:
- Step1: Initial guess from solving naive LSE.
- Step2: Find optimal rotations, then update the linear system (edge length preserving).



Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
Olga Sorkine: Least-Squares Rigid Motion Using SVD

Rotation and edge length constraints

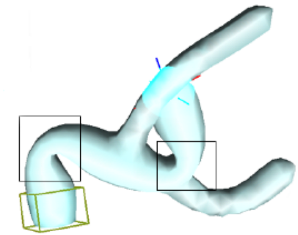
- Iterate two steps to recover rotations:
- Step1: Initial guess from solving naive LSE.
- Step2: Find optimal rotations, then update the linear system (edge length preserving).
- Robustness, simplicity, efficiency.



Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
Olga Sorkine: Least-Squares Rigid Motion Using SVD

Rotation and edge length constraints

- Iterate two steps to recover rotations:
- Step1: Initial guess from solving naive LSE.
- Step2: Find optimal rotations, then update the linear system (edge length preserving).
- Robustness, simplicity, efficiency.
- However, there is no volume preserving constraint.



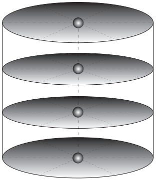
Olga Sorkine and Marc Alexa: As-Rigid-As-Possible Surface Modeling. Eurographics SGP2007
Olga Sorkine: Least-Squares Rigid Motion Using SVD

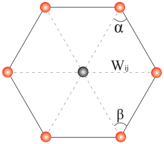
Skeleton and volume constraints

- Use volumetric mesh? Manifoldness, computational complexity, etc.

Skeleton and volume constraints

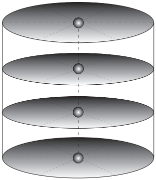
- Use volumetric mesh? Manifoldness, computational complexity, etc.
- Use both the skeleton and edge length constraint to roughly preserve the volume, without breaking the manifoldness of ARAP or increasing the computation complexity.

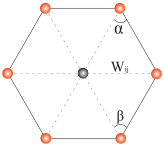




Skeleton and volume constraints

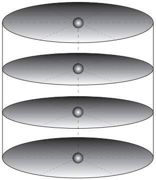
- Use volumetric mesh? Manifoldness, computational complexity, etc.
- Use both the skeleton and edge length constraint to roughly preserve the volume, without breaking the manifoldness of ARAP or increasing the computation complexity.
- $$\begin{bmatrix} L & \mid & 0 \\ l_c & \mid & 0 \\ & L_s & \end{bmatrix} \begin{bmatrix} V' \\ V'_s \end{bmatrix} = \begin{bmatrix} \delta \\ V_c \\ \delta_s \end{bmatrix}$$

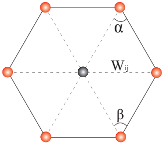




Skeleton and volume constraints

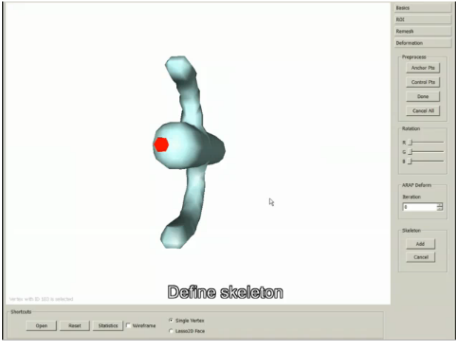
- Use volumetric mesh? Manifoldness, computational complexity, etc.
- Use both the skeleton and edge length constraint to roughly preserve the volume, without breaking the manifoldness of ARAP or increasing the computation complexity.
- $$\begin{bmatrix} L & | & 0 \\ l_c & | & 0 \\ & L_s & \end{bmatrix} \begin{bmatrix} V' \\ V'_s \end{bmatrix} = \begin{bmatrix} \delta \\ V_c \\ \delta_s \end{bmatrix}$$
- One-way coupling property.





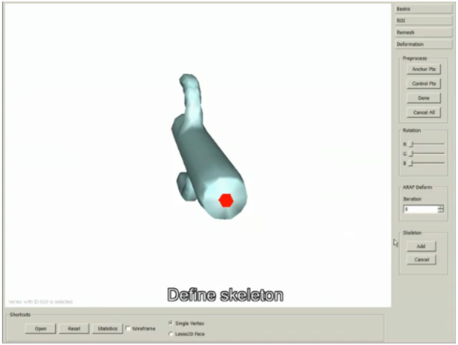
Mesh editing framework

- Manually define the skeleton.



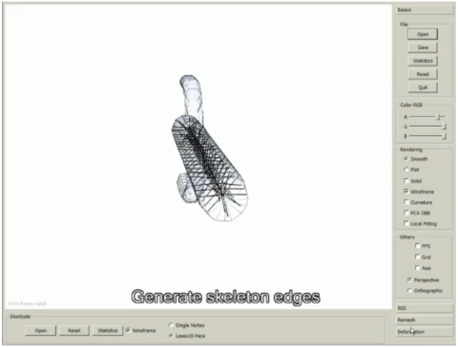
Mesh editing framework

- Manually define the skeleton.



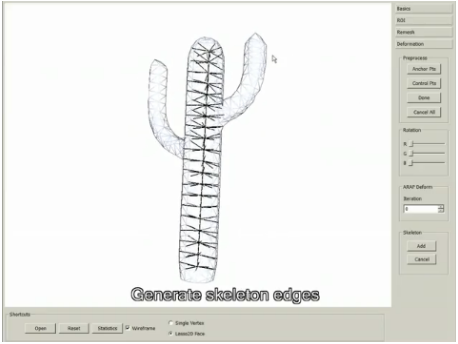
Mesh editing framework

- Manually define the skeleton.
- Evenly generate skeleton points, and connect them with surface vertices automatically.



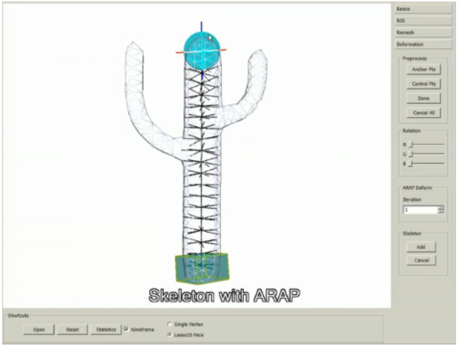
Mesh editing framework

- Manually define the skeleton.
- Evenly generate skeleton points, and connect them with surface vertices automatically.



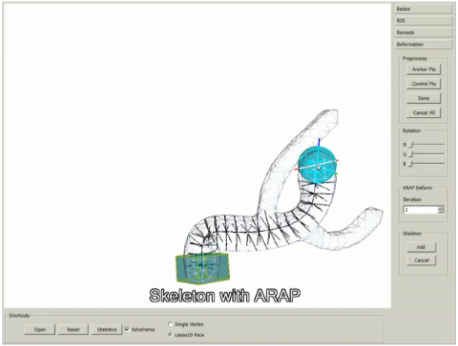
Mesh editing framework

- Manually define the skeleton.
- Evenly generate skeleton points, and connect them with surface vertices automatically.
- Manually select anchor points (bottom) and control points (top).



Mesh editing framework

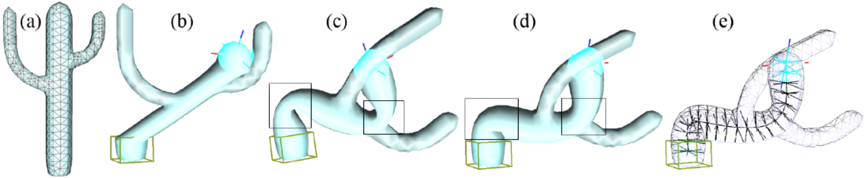
- Manually define the skeleton.
- Evenly generate skeleton points, and connect them with surface vertices automatically.
- Manually select anchor points (bottom) and control points (top).
- Interactively deform the shape.



Experimental settings

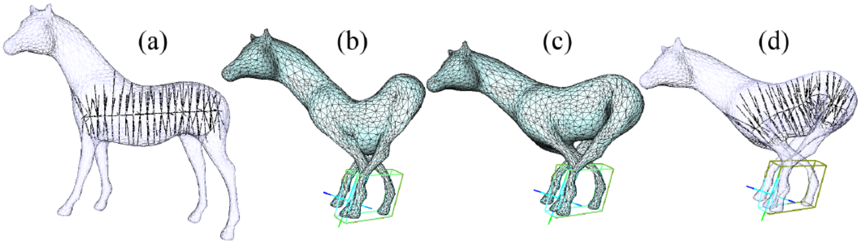
- The C++ implementation was run on a Intel Core2 Quad 2.40GHz CPU with 8G RAM.
- We compare the linear LSE, ARAP surface modeling and our method.
- We tested on the cactus model (620 vertices, $1,236$ polygons) and the horse model ($2,482$ vertices, $4,960$ polygons).
- The relative root mean square errors of edge lengths and volume magnitudes are reported.

Results



Model	RRMS-E	RE-V	Times
(b)	0.126	0.453	0.017
(c)	0.074	0.131	0.024
(d)	**0.075**	**0.056**	**0.025**

Results



Model	RRMS-E	RE-V	Times
(b)	0.068	0.356	0.117
(c)	**0.040**	**0.125**	**0.121**

Conclusions

- We proposed an approach to approximately preserve the volume without breaking the manifoldness of traditional ARAP or increasing the computational complexity.
- Our method is easy-to-implement and may be useful to systems relying on ARAP techniques.
- Limitations: Skeleton generation; complex skeletons; self intersection.

Thanks for listening.