Mesh Editing based on Discrete Laplace and Poisson Models

Marc Alexa

TU Berlin

Presented by Barry Shih 11/12/2007

Overview

- Introduction
- Methodology
  - Laplace Vector
  - Modeling Framework
  - Linear Transformation
- Application
  - Deformation
  - Detail Transfer
  - Transplanting Surface Patches
- Conclusions

Introduction

- Goal: Manipulate and modify a surface without losing geometric details
- Absolute position of the vertices is not important
- Intrinsic surface representation
  - Multi-resolution representation
    - Locality is limited
  - Differential coordinates on 2D image

Laplace Vector

- Define coordinate i by its Laplace vector :

$$\delta_i = v_i - c_{ij} \sum_{j \in N_i} v_j, \sum_j c_{ij} = 1$$

  - The value differ in the definition of c_{ij}
  - In this paper $c_{ij} = 1/d_i$
- Transfer V into Δ

$$\Delta = (I - C) V$$

  - V can be recovered from Δ

Modeling Framework

- Goal: minimize squares
- Define the deformed geometry V' (without additional constraints)

$$V' = \min_{V'} \sum_{i=1}^{n} \left| \delta_i - \left(v'_i - \frac{1}{d_i} \sum_{j \in N_i} v'_j \right) \right|^2$$

- Solve V'

$$AV' = b \Rightarrow A^T A V' = A^T b$$

- We can prescribe position of certain v_i by adding additional row $w_i \| v'_i = \hat{v}_i$

Linear Transformation

- Goal: assign each vertex i with an individual transformation T_i
- Define V'

$$V' = \min_{V'} \sum_{i=1}^{n} \left| T_i \delta_i - \left(v'_i - \frac{1}{d_i} \sum_{j \in N_i} v'_j \right) \right|^2$$

- $AV' = b$ can be solved independent of T_i
  - Apply T_i to vertex i, allowing interactive modeling
- T_i is derived from i and its neighbors

$$T_i = \min_{T_i} \left(\left| T_i v_i - v'_i \right|^2 + \sum_{j \in N_i} \left| T_i v_j - v'_j \right|^2 \right)$$

Linear Transformation

- Transformation should be linear function to V but constrained to isotropic scales and rotations
- Define T_i :

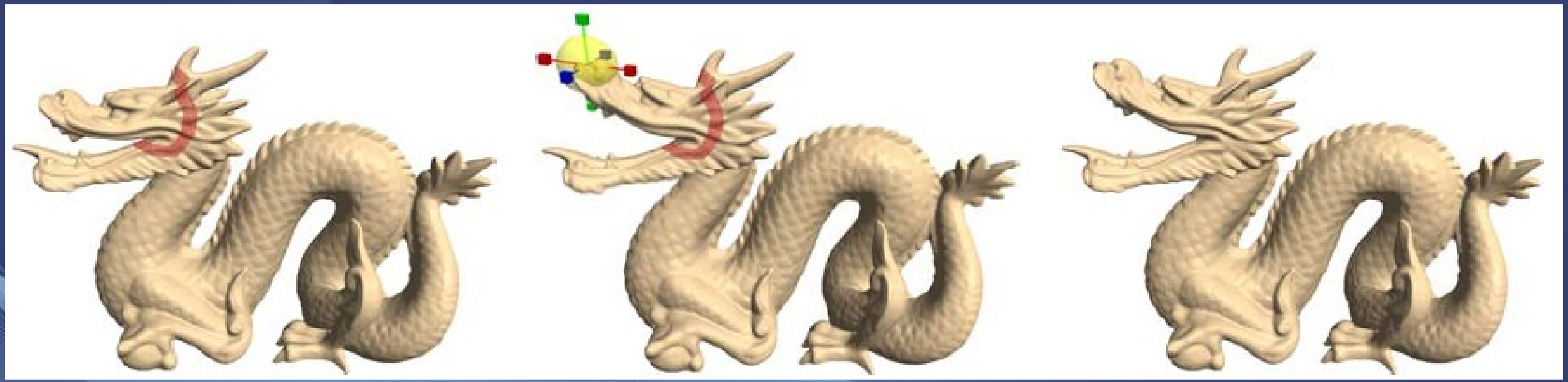
$$T_i = \begin{bmatrix} s & h_1 & -h_2 & t_x \\ -h_1 & s & h_3 & t_y \\ h_2 & -h_3 & s & t_y \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Minimize $\left| A_i (s_i, h_i, t_i)^T - b_i \right|^2$

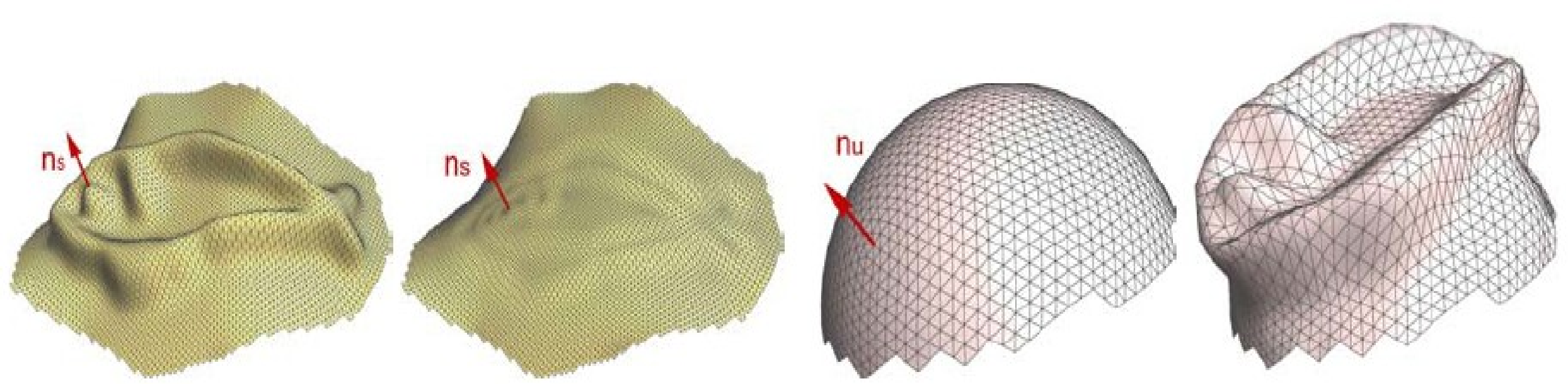
and get $(s_i, h_i, t_i)^T = (A_i^T A_i)^{-1} A_i^T b_i$

Deformation

- Define Region Of Interest by stationary anchors
- The handle is the mean of user control

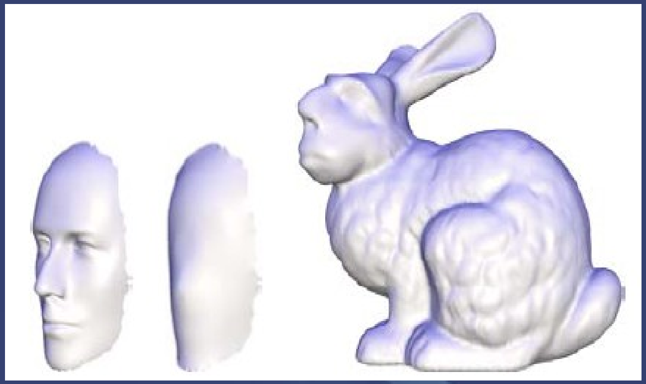


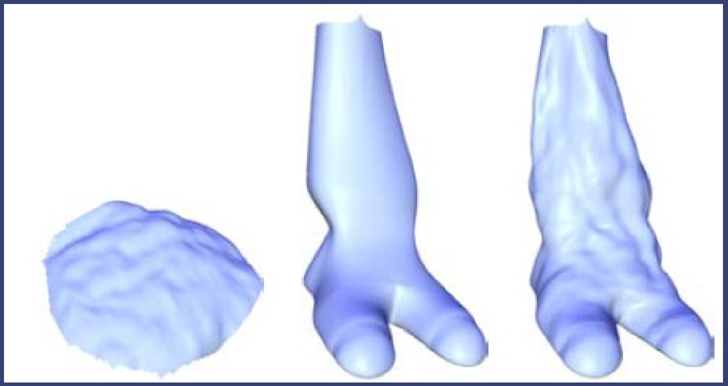
Detail Transfer



- Calculate $\xi_i = \delta_i - \tilde{\delta}_i$, we can see that $S = L^{-1}(\tilde{\delta} + \xi)$
- Rotate normal vector $n_u = R_i(n_s)$, apply the rotate function to Laplace difference $\xi'_i = R_i(\xi)$
- Calculate the desired surface U' $U' = L^{-1}(\Delta + \xi')$
- In the case with different connectivity, use mean-value coordinates to find the corresponding vertices

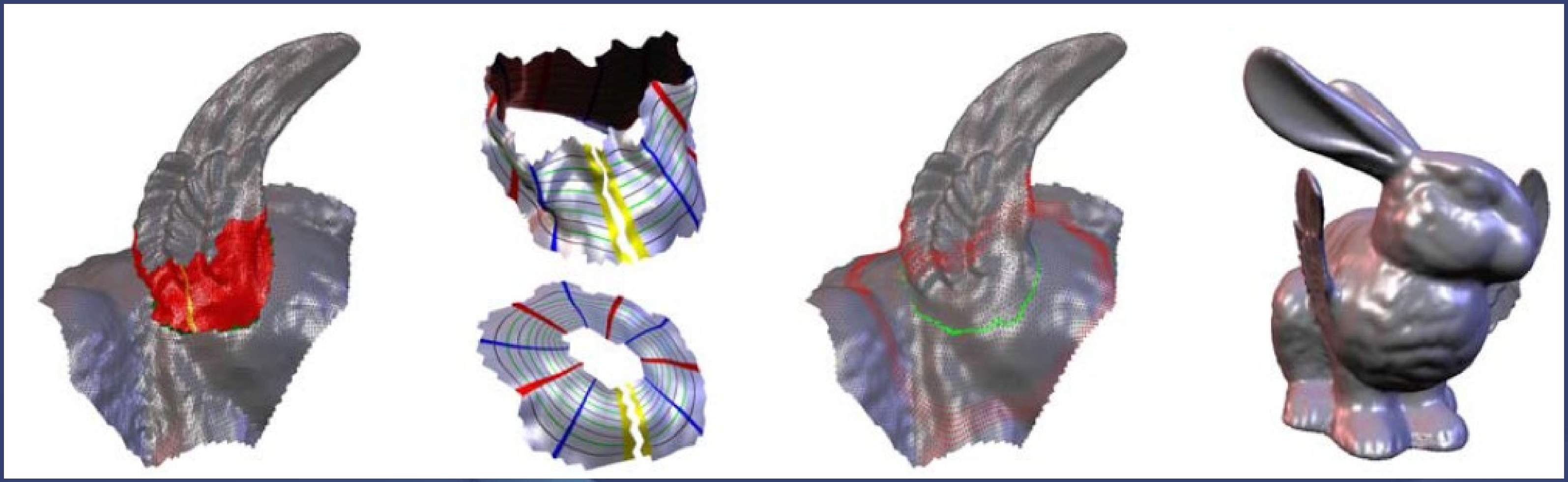
Detail Transfer





- Optimize the result by adding constraints in some case

Transplanting Surface Patches



- Zipped the patches with the target surface
- Select transitional regions
- Linearly blend the two regions

Conclusions

- Utilizing Laplace vector as intrinsic geometry representation in mesh editing will preserve surface detail as mush as possible
- Transformation being linear function of the input vertices guarantees fast computation
- Has limited result on anisotropic scaling, shear since doing so will remove normal component from Laplace representation

Thank you!