Deformation

Xiao-Ming Fu

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
    - Meshless mapping
- Volumetric Deformation
    - Tetrahedral mapping

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
    - Meshless mapping
- Volumetric Deformation
    - Tetrahedral mapping

Definition

- The deformation of a given surface S into the desired surface S'
  - a displacement $d(p)$ on each vertex $p \in S$
  - $S' = \{p + d(p) | p \in S\}$
- The user controls the deformation by
  - prescribing displacements $\bar{d}_i$ for a set of vertices $p_i \in H \subset S$.
  - constraining certain parts F stay fixed.
  - handles
- The main question: determine the displacements for vertices in $S \backslash (H \cup F)$.

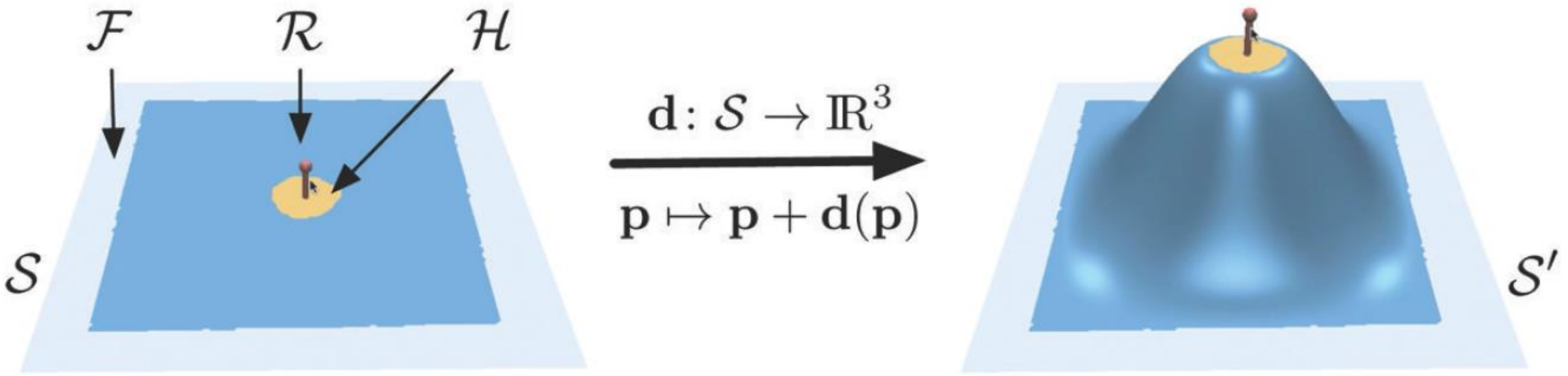


Figure 9.1. A given surface $\mathcal{S}$ is deformed into $\mathcal{S}'$ by a displacement function $\mathbf{d}(\mathbf{p})$. The user controls the deformation by moving a handle region $\mathcal{H}$ (yellow) and keeping the region $\mathcal{F}$ (gray) fixed. The unconstrained deformation region $\mathcal{R}$ (blue) should deform in an intuitive, physically-plausible manner.

Two classes of shape deformations

- Surface-based deformations
  - The displacement is defined on each vertex
  - A high degree of control, since each vertex can be constrained individually.
  - The robustness and efficiency of the involved computations are strongly affected by the mesh complexity and the triangle quality of the original surface S.

- Space deformations
  - Displacement is defined on each point in the space.
  - Smooth.

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
    - Meshless mapping
- Volumetric Deformation
    - Tetrahedral mapping

Transformation Propagation

- Propagating the user-defined handle transformation:

✓1. specify the support region of the deformation

✓2. specify a handle region within the support region

✓3. the handle is transformed using some modeling interface

✓4. propagate the transformation of handle and damp within the support region

  - ✓leading to a smooth blending between the transformed handle and the fixed region

Figure 9.2. After specifying the blue support region and the green handle region (left), a smooth scalar field is constructed that is 1 at the handle and 0 outside the support (center). Its isolines are visualized in black and red, where red is the $\frac{1}{2}$-isoline. This scalar field is used to propagate and damp the handle's transformation within the support region (right). (Image taken from [Botsch et al. 06b]. ©2006 ACM, Inc. Included here by permission.)

Smooth blend

- Controlled by a scalar field:
  - 1 is at the handle;
  - 0 is at the fixed region;
  - smoothly blends between 1 and 0 within the support region.
- One method:
  - $d_F(p)$: distance from p to the fixed region
  - $d_H(p)$: distance from p to the handle

$$s(p) = \frac{d_F(p)}{d_F(p) + d_H(p)}$$

Harmonic field

$$\Delta s(p_i) = 0 \quad \forall p_i \in R$$
$$s(p_i) = 1 \quad \forall p_i \in H$$
$$s(p_i) = 0 \quad \forall p_i \in F$$

Simple to implement

Discussion

- simple and efficient to compute
- distance-based propagation of transformations will typically not result in the geometrically most intuitive solution.

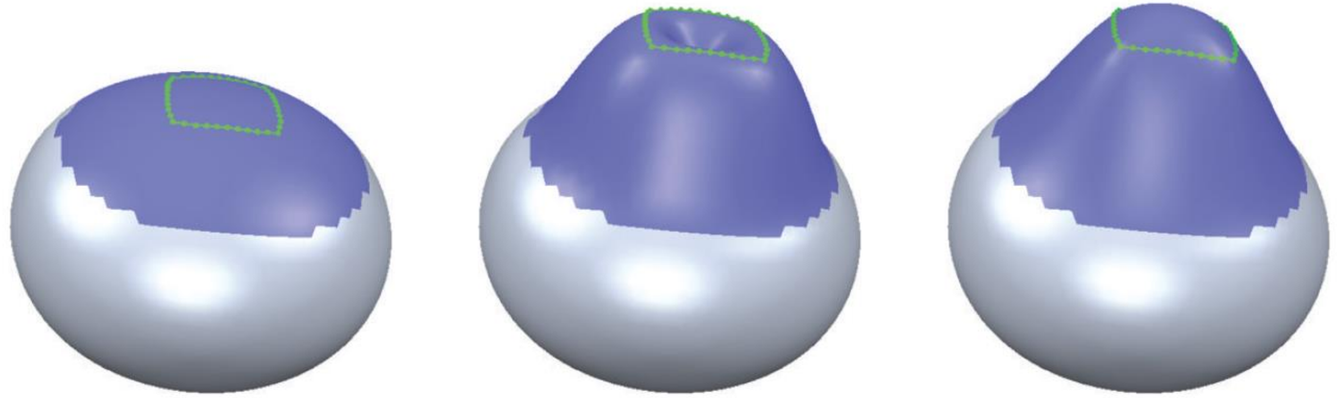


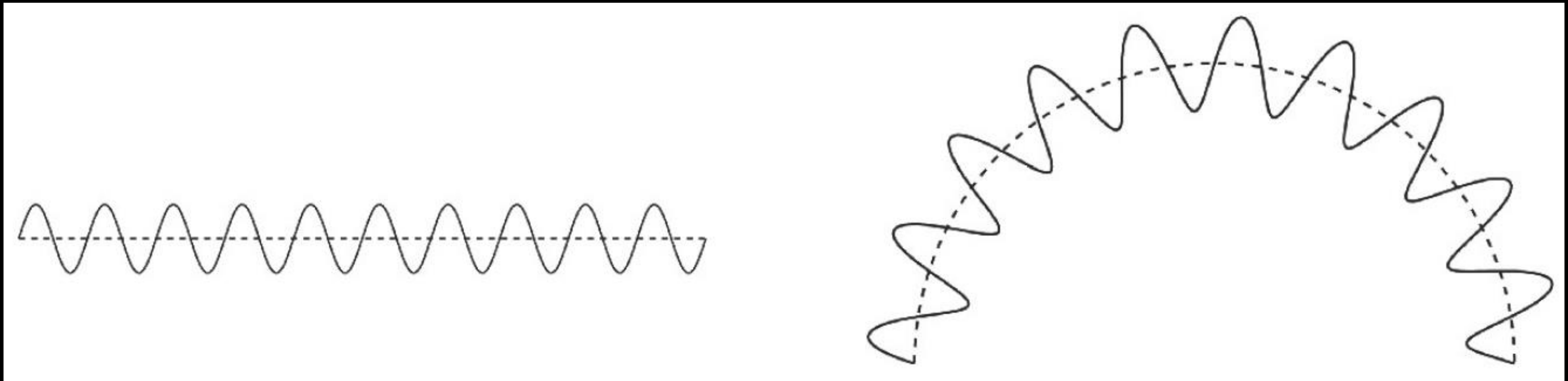
Figure 9.3. A sphere is deformed by lifting a closed handle polygon (left). Propagating this translation based on geodesic distance causes a dent in the interior of the handle polygon (center). A more intuitive solution can be achieved by minimizing physically-motivated deformation energies (right). (Image taken from [Botsch 05].)

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
  - Meshless mapping
- Volumetric Deformation
  - Tetrahedral mapping

Multi-scale deformations

- Main idea: decompose the object into two frequency bands using the smoothing and fairing techniques.
  - the low frequencies correspond to the smooth global shape;
  - the high frequencies correspond to the fine-scale details.
- Goal: deform the low frequencies (global shape) while preserving the high-frequency details



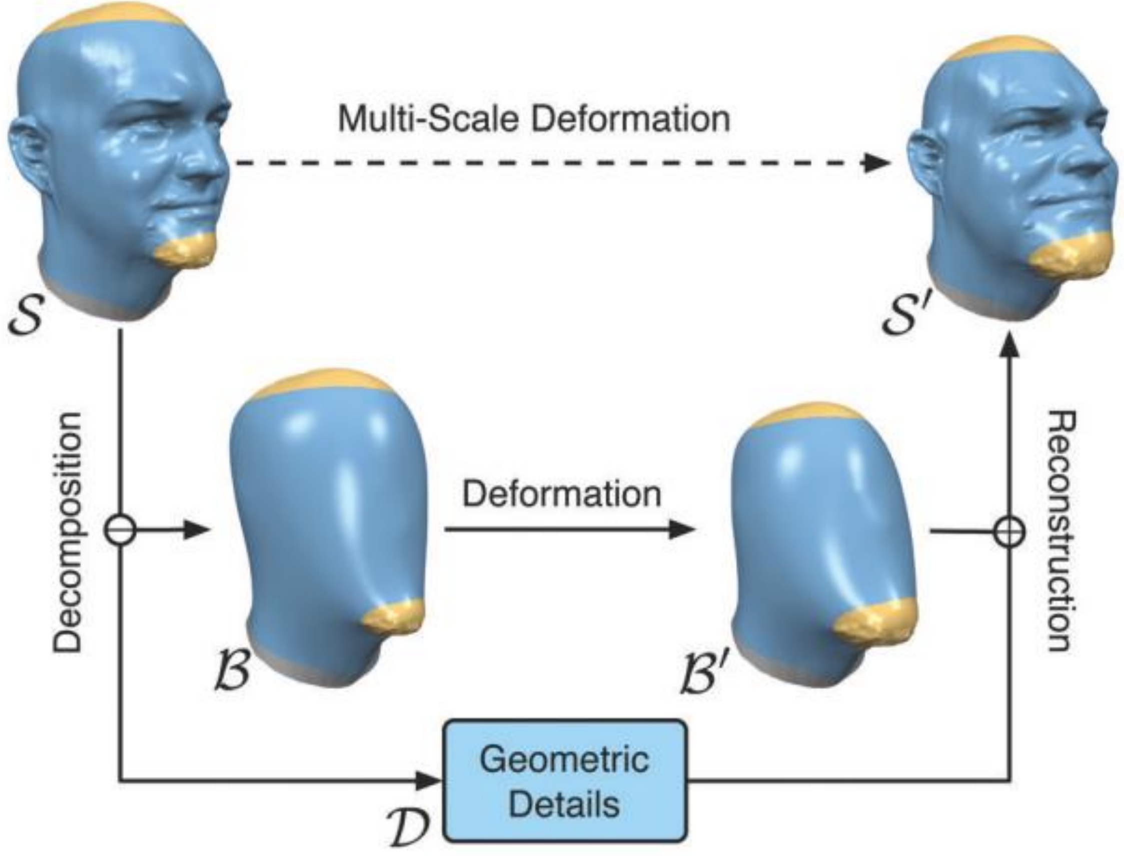


Figure 9.7. A general multi-scale editing framework consists of three main operators: the decomposition operator, which separates the low and high frequencies; the editing operator, which deforms the low frequency components; and the reconstruction operator, which adds the details back onto the modified base surface. Since the lower part of this scheme is hidden in the multi-scale kernel, only the multi-scale edit in the top row is visible to the designer. (Image taken from [Botsch and Sorkine 08]. ©2008 IEEE. Model courtesy of Cyberware.)

Pipeline

- First a low-frequency representation of the given surface S is computed by removing the high frequencies, yielding a smooth base surface B. The geometric detail $D = S \ominus B$.
- Deform the B to B'
- Adding the geometric details onto B': $S' = B' \oplus D$

- $\ominus$: decomposition
- $\oplus$: reconstruction
- mesh smoothing or fairing

Representation for the geometric detail

- The straightforward representation: a vector-valued displacement function
  - associates a displacement vector to each point on the base surface.
  - per-vertex displacement vectors
  - $p_i = b_i + h_i, p_i \in S, b_i \in B, h_i \in R^3$
- Encoded in local frame

$$h_i = \alpha_i n_i + \beta_i t_{i,1} + \gamma_i t_{i,2}$$

n_i:normal

$t_{i,1}, t_{i,2}$: two tangent vectors

Encoded in local frame

- When the base surface S is deformed to S'

$$h_i' = \alpha_i n_i' + \beta_i t_{i,1}' + \gamma_i t_{i,2}'$$

$$p_i' = b_i' + h_i'$$

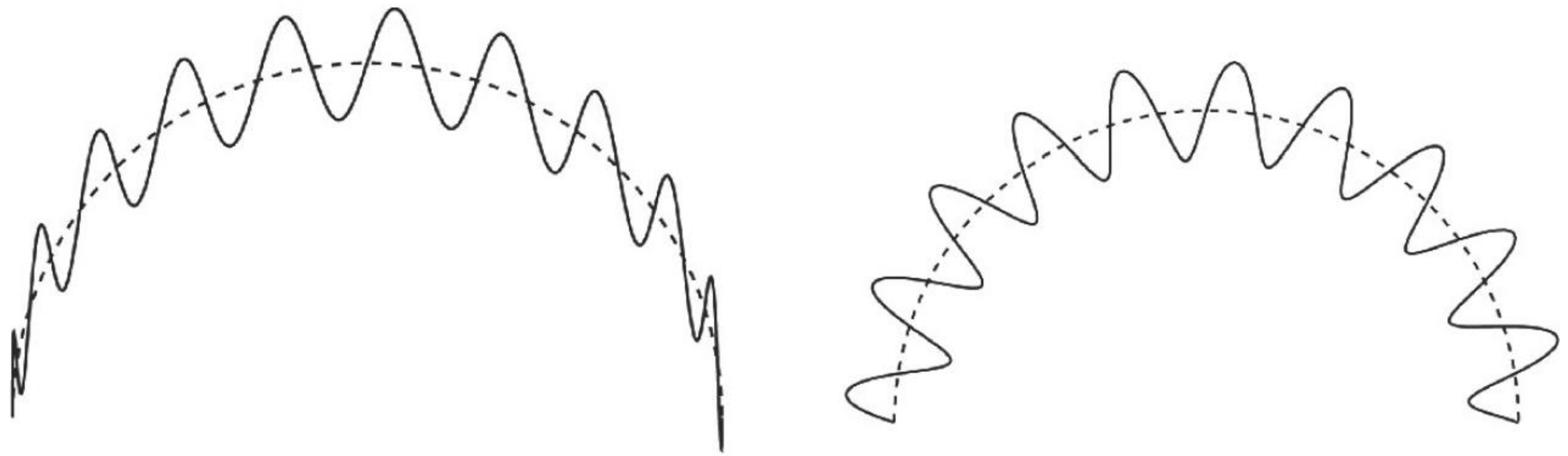


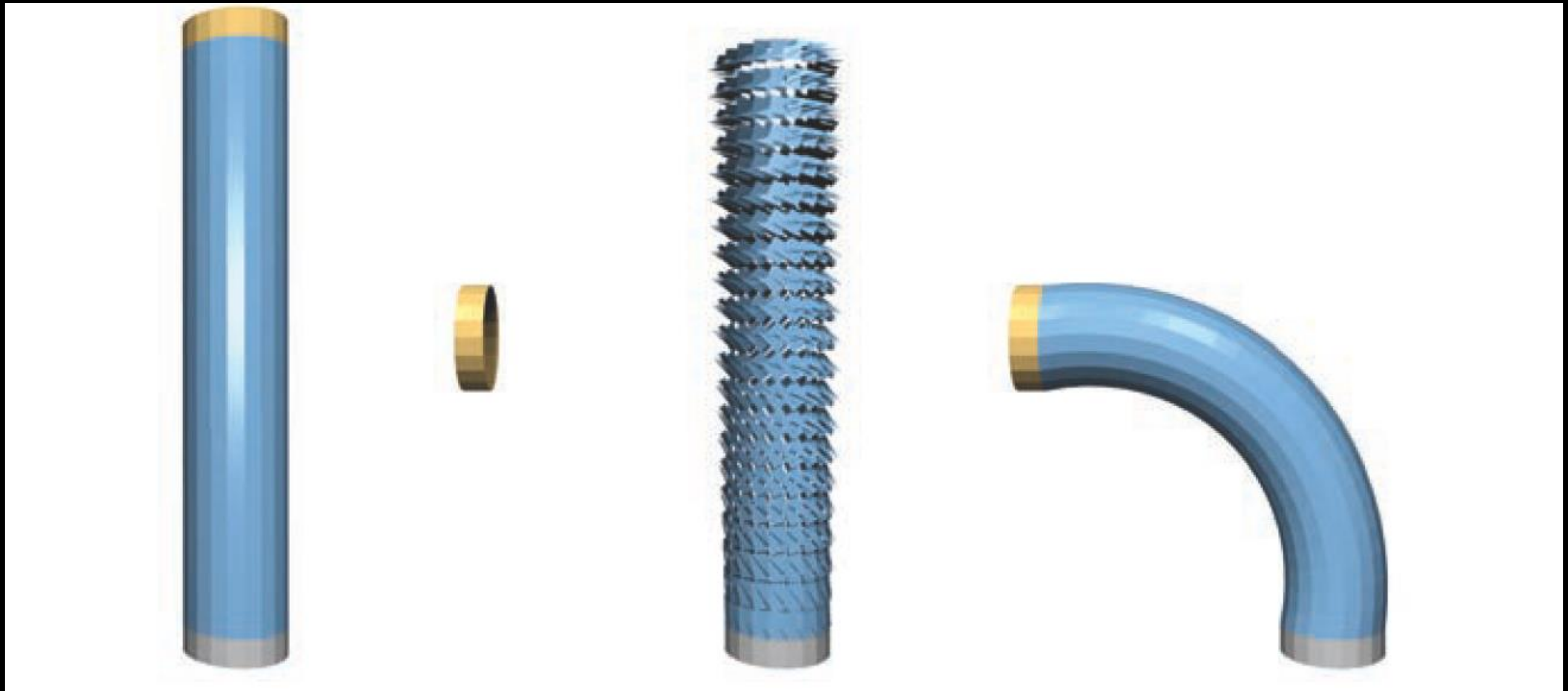
Figure 9.8. Representing the displacements with regard to the global coordinate system does not lead to the desired result (left). The geometrically intuitive solution is achieved by storing the details with regard to local frames that rotate according to the local tangent plane's rotation of $\mathcal{B}$ (right). (Image taken from [Botsch 05].)

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
    - Meshless mapping
- Volumetric Deformation
    - Tetrahedral mapping

Gradient-Based Deformation

- deform the surface by
  - 1. manipulating the original surface gradients
  - 2. finding the deformed surface that matches the target gradient field in the least-squares sense



Gradient-Based Deformation

- Gradient of the coordinate function on facet f_i

$$\nabla p_i = \begin{bmatrix} \nabla p_{x,i} \\ \nabla p_{y,i} \\ \nabla p_{z,i} \end{bmatrix} =: J_i \in R^{3\times 3}$$

The rows of J_i are just the gradients of the x-, y- and z-coordinates.

- Manipulation: $J_i' = M_i J_i$
  - M_i: local rotation/scale/shear, (discussed later)
  - breaking up the mesh. (similar to ARAP parameterization)

Find new vertex positions p_i'

$J_i' = M_i J_i$

- Goal: the gradient of p_i' are as close as possible to J_i'

: new vertex position

A_i: the area of facet f_i

$\nabla p_i'$: the gradient is defined on the original surface (just replace the function value)

Solving Laplace equation for x, y, z.

Poisson equation.

Laplacian-Based Deformation

- manipulate per-vertex Laplacians instead of per-face gradients
- 1. compute initial Laplace coordinates $\delta_i = \Delta(p_i)$
- 2. manipulate them to $\delta_i' = M_i \delta_i$, (discussed later)
- 3. find new coordinates p_i' that match the target Laplacian coordinates

$$E = \sum_i A_i \|\Delta(p_i') - \delta_i'\|_F^2$$

A_i: local average area for vertex i.

bi-Laplacian system

Uniform Laplace or cot Laplace

The cot weight $\Delta(p_i')$ is same to $\Delta(p_i)$.

Local Transformations M_i for face

- Propagation of deformation gradients.
- The user manipulates the handle by prescribing an affine transformation
$$T(x) = Mx + t$$

M: gradient of $T(x)$

- propagate this matrix over the deformable region and damp it using the smooth scalar field.
- Rotations should be interpolated differently than scalings.

Propagation of deformation gradients

- Polar decomposition:

$$M = R \cdot S$$
$$R = UV^T$$
$$S = V\Sigma V^T$$

Where

$$M = U\Sigma V^T$$

R: rotation; S: scaling

rotation and scaling components are then interpolated separately

Propagation of deformation gradients

- Rotation: quaternion interpolation R_i
- Scaling: linear interpolation $S_i = (1 - s_i)S + s_i \cdot Id$
- $M_i = R_i \cdot S_i$

- Discussion:
  - works very well for rotations
  - insensitive to handle translations

Local Transformations M_i for vertex

- Implicit optimization: simultaneously optimize for both the new vertex positions p_i' and the local rotations M_i.

$$E = \sum_i A_i \|\Delta(p_i') - M_i \delta_i\|_F^2$$

- To avoid a nonlinear optimization and rigid transformation is desired
  - linearized similarity transformations, skew-symmetric matrices

$$M_i = \begin{bmatrix} s_i & -h_{i,z} & h_{i,y} \\ h_{i,z} & s_i & -h_{i,x} \\ -h_{i,y} & h_{i,x} & s_i \end{bmatrix}$$

Local Transformations M_i for vertex

- To determine s_i, $h_{i,x}$, $h_{i,y}$ and $h_{i,z}$:

$$M_i\left(p_i - p_j\right) = p_i' - p_j', \forall j \in \Omega(i)$$

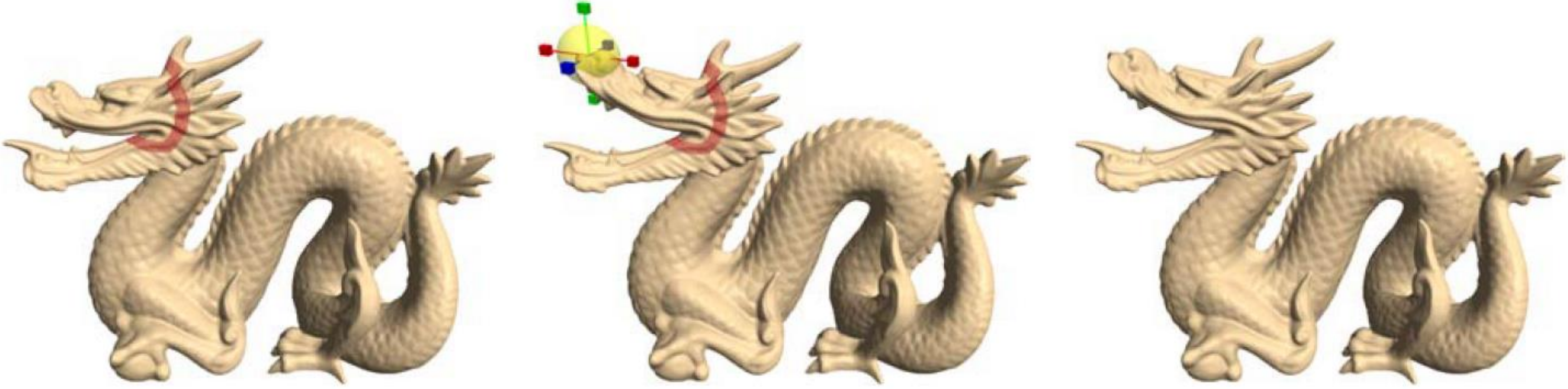
Then:

M_i is a linear combinations of p_i'.

$E = \sum_i A_i ||\Delta(p_i') - M_i \delta_i||_F^2$ becomes a quadratic energy.

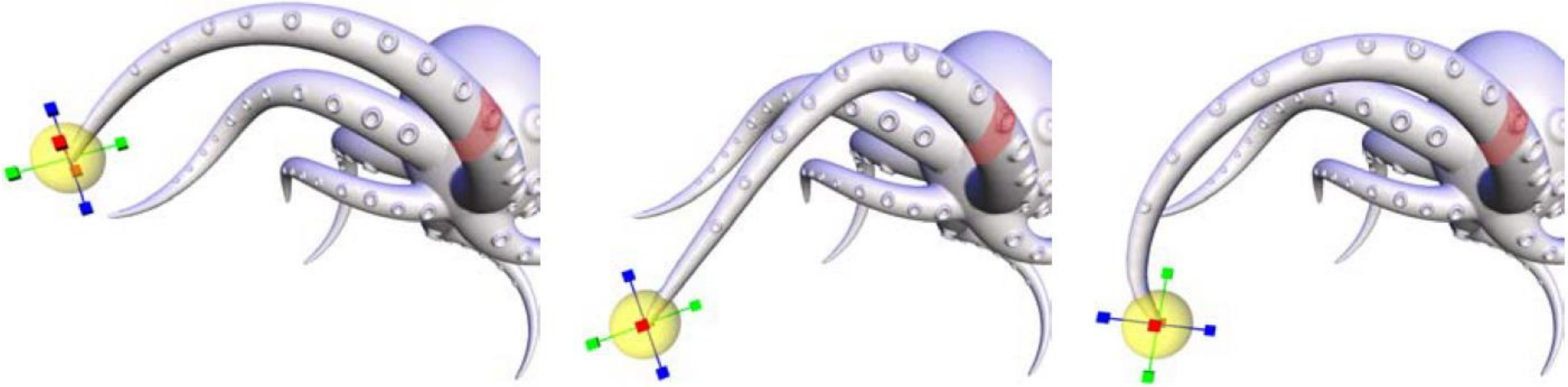
Linear least-squares problem, which can be solved efficiently.

The linearized transformations lead to artifacts in the case of large rotations.



(a) (b) (c)

Figure 1: *The editing process. (a) The user selects the region of interest – the upper lip of the dragon, bounded by the belt of stationary anchors (in red). (b) The chosen handle (enclosed by the yellow sphere) is manipulated by the user: translated and rotated. (c) The editing result.*



(a) (b) (c)

Figure 2: *Different handle manipulations. (a) The region of interest (arm), bounded by the belt of stationary anchors, and the handle. (b) Translation of the handle. (c) Subsequent handle rotation. Note that the detail is preserved in all the manipulations.*

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
  - Meshless mapping
- Volumetric Deformation
  - Tetrahedral mapping

Deformation transfer

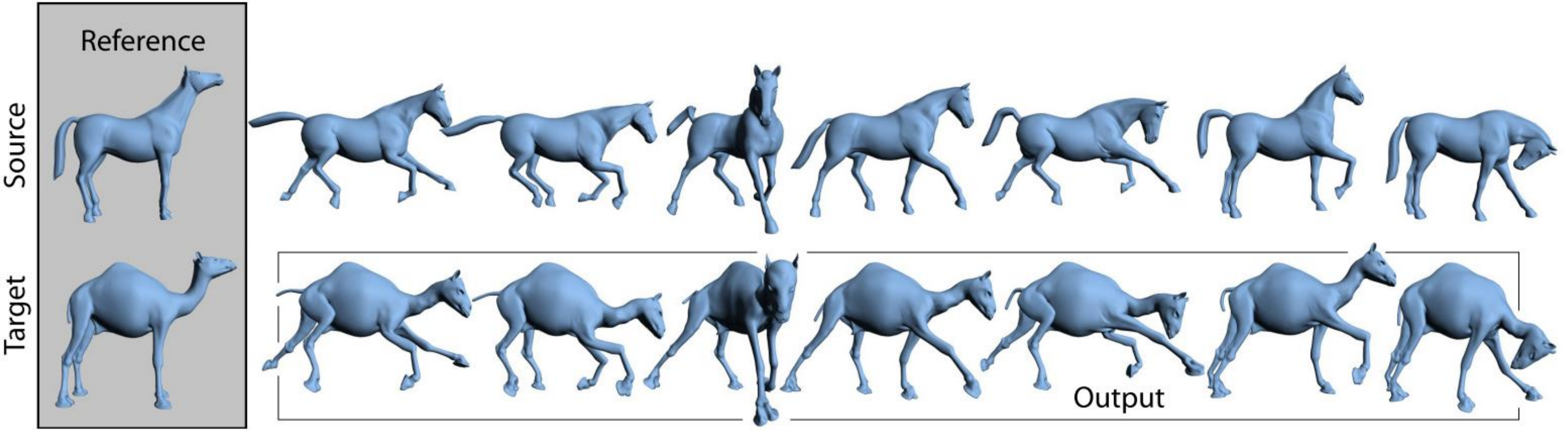
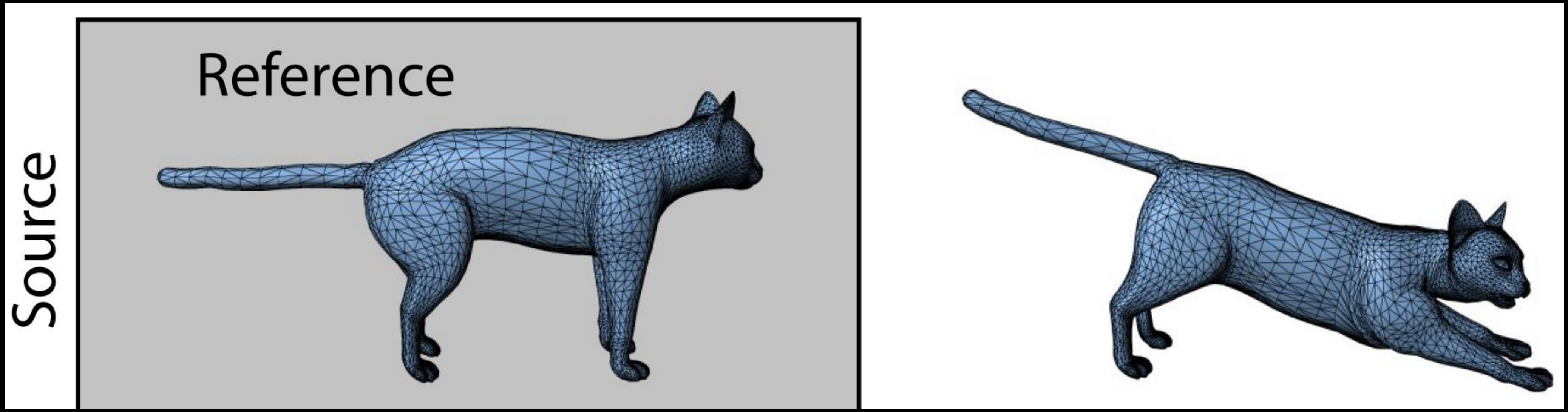


Figure 1: Deformation transfer copies the deformations exhibited by a source mesh onto a different target mesh. In this example, deformations of the reference horse mesh are transfered to the reference camel, generating seven new camel poses. Both gross skeletal changes as well as more subtle skin deformations are successfully reproduced.

Deformation transfer

- The goal of deformation transfer: transfer the change in shape exhibited by the source deformation onto the target.
- Input: source deformation
  - a collection of affine transformations tabulated for each triangle of the source mesh.



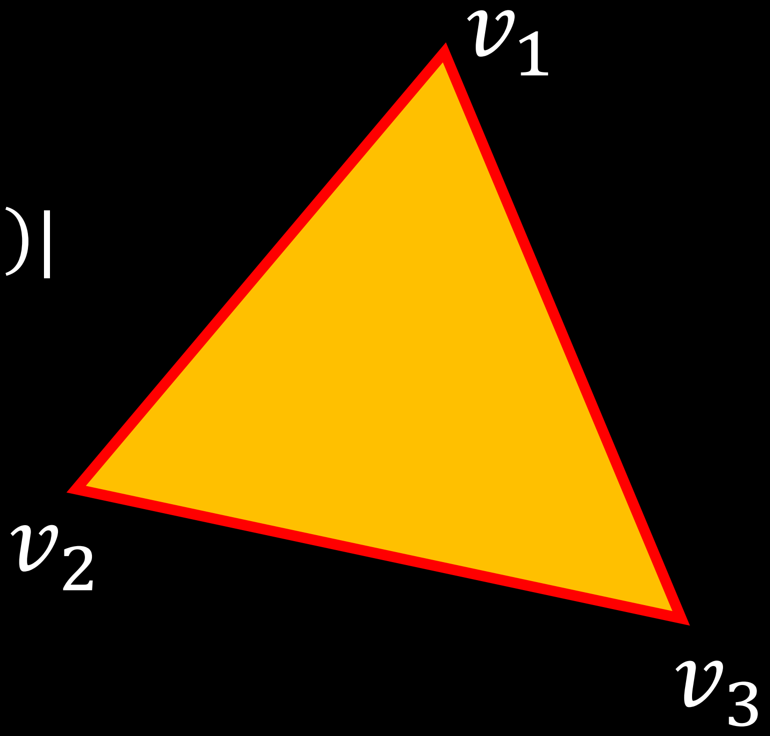
- The three vertices of a triangle before and after deformation do not fully determine the affine transformation.

Affine transformation

- $v_i \in$ undeformed, $\tilde{v}_i \in$ deformed
- add a fourth vertex in the direction perpendicular to the triangle.

$$v_4 = v_1 + n$$
$$n = (v_2 - v_1) \times (v_3 - v_1)/|(v_2 - v_1) \times (v_3 - v_1)|$$

1. an analogous computation for $\tilde{v}_4$
2. How to compute the Affine transformation?



Transfer

- Transfer the source transformations via the correspondence map to the target.

$$\mathrm{E} = \sum_{i} \left\| S_j - T_j \right\|_F^2$$

1. Require one-to-one correspondence between source and target model.
2. Least squares.

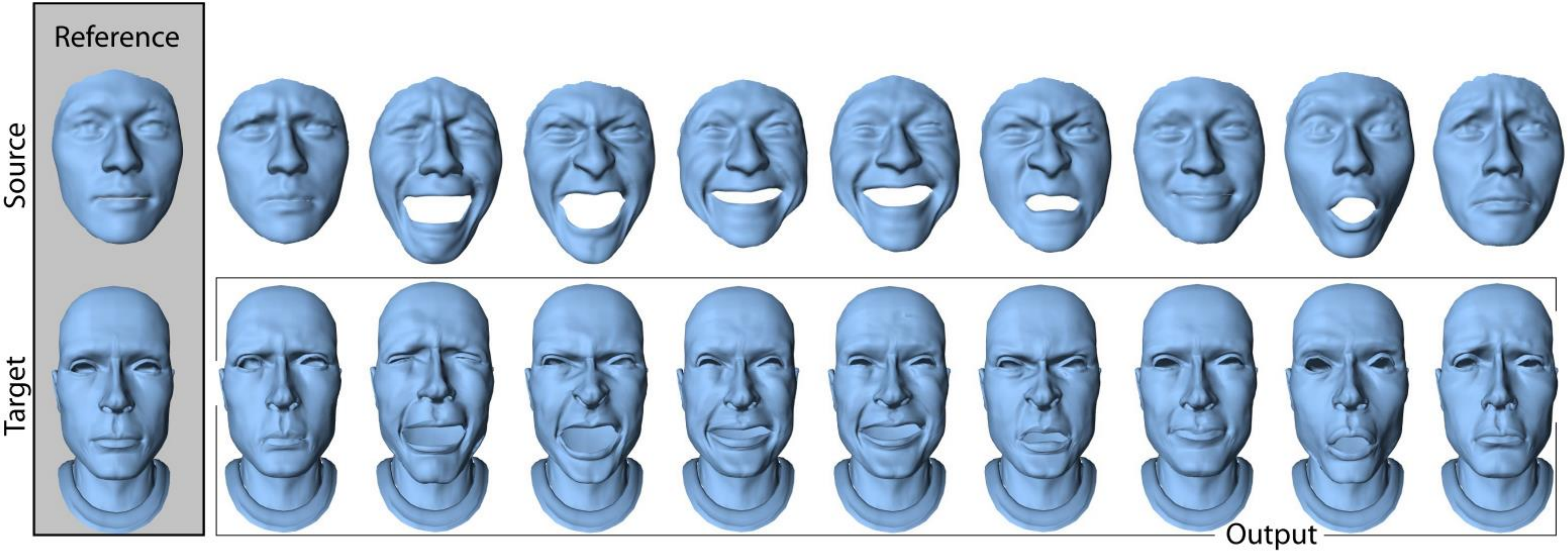


Figure 7: Scanned facial expressions cloned onto a digital character.

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
    - Meshless mapping
- Volumetric Deformation
    - Tetrahedral mapping

As-Rigid-As-Possible Surface Modeling

- Goal: preserve shape meaning that an object is only rotated or translated, but not scaled or sheared.
  - small parts of the shape should change as rigidly as possible
- Energy:

$$E = \sum_{i=1}^{N_v} w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i (p_i - p_j) \right\|^2$$

w_i, w_{ij}: fixed cell and edge weights.

w_{ij}: cot weight; w_i: local average area

Variables: R_i and p_i'

Local step

$$E = \sum_{i=1}^{N_v} w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i (p_i - p_j) \right\|^2$$

- Given p_i', compute R_i

$$E_i = \sum_{j \in \Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i (p_i - p_j) \right\|^2$$

Set $e_{ij}' = p_i' - p_j'$, $e_{ij} = p_i - p_j$,

$$E_i = \sum_{j \in \Omega(i)} w_{ij} \left(e_{ij}' - R_i e_{ij} \right)^T \left(e_{ij}' - R_i e_{ij} \right)$$

$$= \sum_{j \in \Omega(i)} w_{ij} \left(e_{ij}'^T e_{ij}' - 2 e_{ij}'^T R_i e_{ij} + e_{ij}^T e_{ij} \right)$$

Local step

$$E = \sum_{i=1}^{N_v} w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p'_i - p'_j) - R_i (p_i - p_j) \right\|^2$$

$$\arg\min_{R_i} \sum_{j \in \Omega(i)} w_{ij} \left(\cancel{e'^T_{ij} e'_{ij}} - 2 e'^T_{ij} R_i e_{ij} + \cancel{e^T_{ij} e_{ij}} \right)$$

$$= \arg\max_{R_i} \sum_{j \in \Omega(i)} w_{ij} 2 e'^T_{ij} R_i e_{ij} = \arg\max_{R_i} Tr \left(\sum_{j \in \Omega(i)} w_{ij} R_i e_{ij} e'^T_{ij} \right)$$

$$= \arg\max_{R_i} Tr \left(\sum_{j \in \Omega(i)} w_{ij} R_i e_{ij} e'^T_{ij} \right) = \arg\max_{R_i} Tr \left(R_i \sum_{j \in \Omega(i)} w_{ij} e_{ij} e'^T_{ij} \right)$$

Local step

$$E = \sum_{i=1}^{N_v} w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p'_i - p'_j) - R_i (p_i - p_j) \right\|^2$$

Set $S_i = \sum_{j \in \Omega(i)} w_{ij} e_{ij} e'^{T}_{ij}$ and $S_i = U_i \Sigma_i V_i^T$.

If M is a positive-symmetric-definite matrix then for any orthogonal R, $Tr(M) \; > Tr(RM)$.

The rotation matrix R_i maximizing $Tr(R_i S_i)$ is obtained when $R_i S_i$ is symmetric positive semi-definite.

$\Longrightarrow R_i = V_i U_i^T$.

Global step

$$E = \sum_{i=1}^{N_v} w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p'_i - p'_j) - R_i (p_i - p_j) \right\|^2$$

- Given R_i, compute p'_i
- Linear squares, easy to solve

- Initial deforamation
  - 1. Previous frame (for interactive manipulation)
  - 2. Naive Laplacian editing
  - …

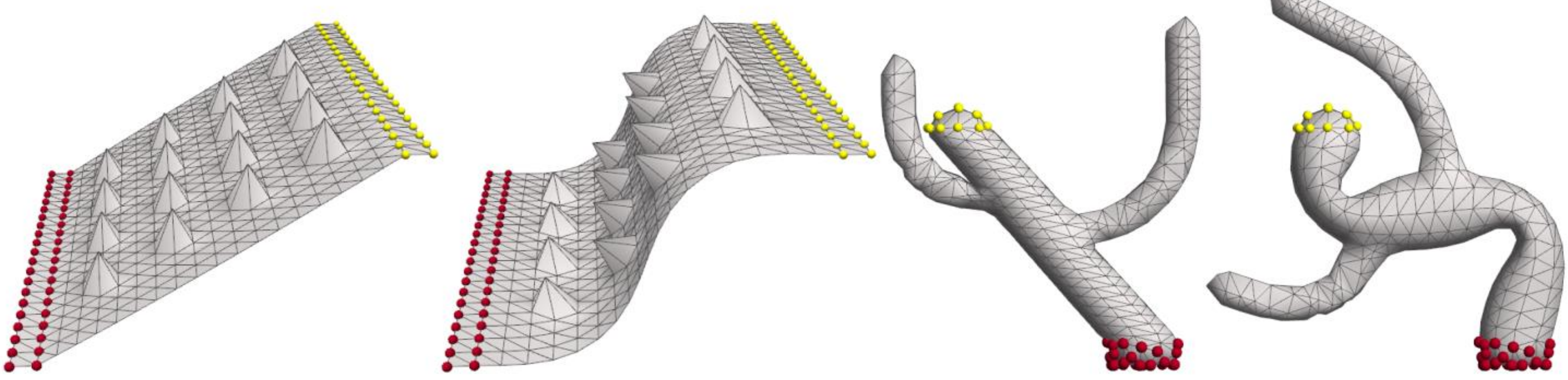


Figure 5: *Comparison with Poisson mesh editing. The original models appear in Figures 2 and 7. The yellow handle was only translated; this poses a problem for rotation-propagation methods such as [YZX*04, ZRKS05, LSLCO05].*

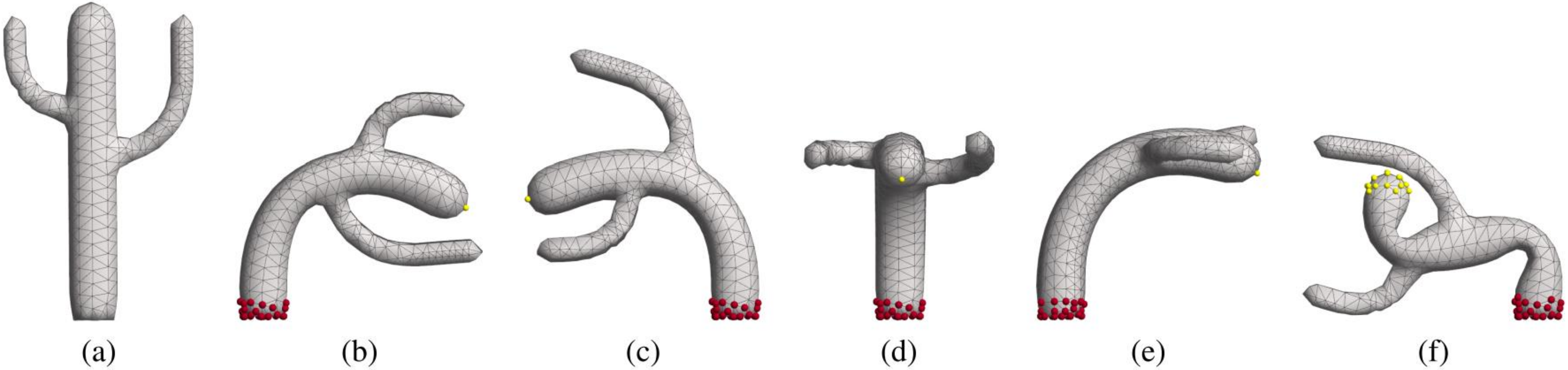


Figure 7: *Bending the Cactus. (a) is the original model; yellow handles are translated to yield the results (b-f). (d) and (e) show side and front views of forward bending, respectively. Note that in (b-e) a* single vertex *at the tip of the Cactus serves as the handle, and the bending is the result of translating that vertex, no rotation constraints are given.*

Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
  - Meshless mapping
- Volumetric Deformation
  - Tetrahedral mapping

Space deformations

- Deform the ambient space and thus implicitly deform the embedded objects.

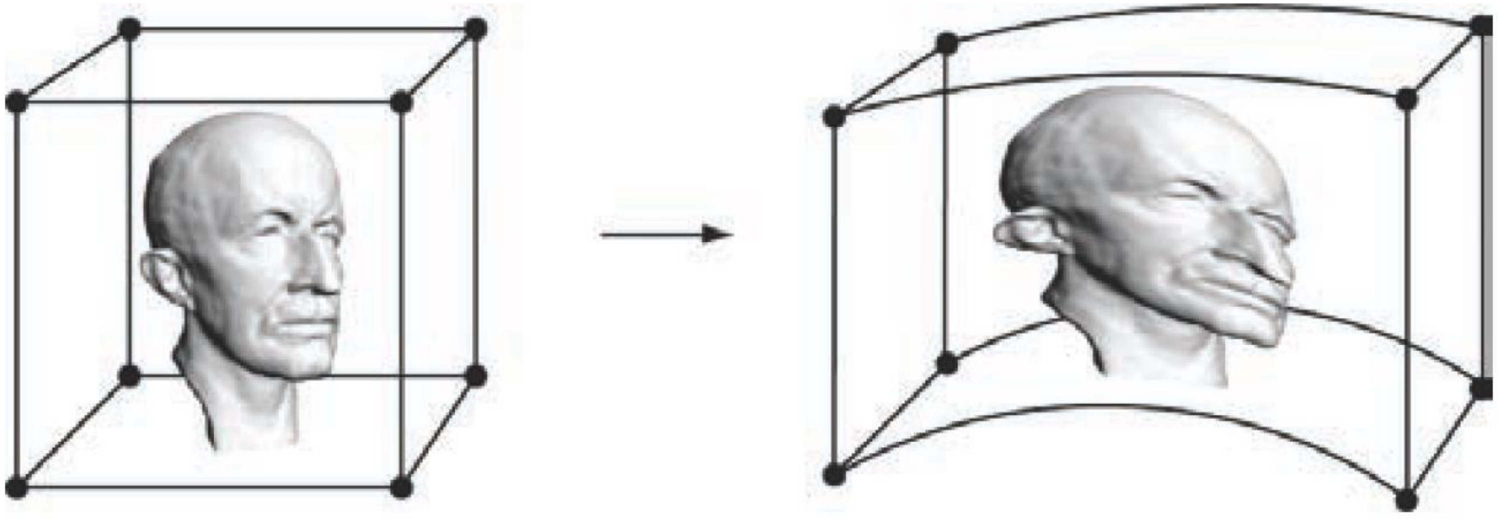


Figure 9.12. Space deformations warp the embedding space around an object and thus implicitly deform the object. (Image taken from [Botsch et al. 06b]. ©2006 ACM, Inc. Included here by permission.)

Lattice-Based Freeform Deformation

- Freeform deformation represents the space deformation by a trivariate tensor-product spline function

$$d(u,v,w) = \sum_i \sum_j \sum_k \delta c_{ijk} N_i(u) N_j(v) N_k(w)$$

1. N_i are B-spline basis functions
2. $\delta c_{ijk} = c'_{ijk} - c_{ijk}$ displacements of the control points c_{ijk}
3. Original vertex p_i satisfying

$$p_i = \sum_i \sum_j \sum_k c_{ijk} N_i(u) N_j(v) N_k(w)$$

New vertex $p'_i = p_i + d(u,v,w) = \sum_i \sum_j \sum_k c'_{ijk} N_i(u) N_j(v) N_k(w)$

Deformation

- A handle-based interface for direct manipulation.
- Input a set of displacement constraints: $\bar{d}_i$ for $\mathrm{H} \cup F = \{p_1, \dots, p_m\}$.
- Least squares:

$$E = \sum_{l=1}^{m} \left| \bar{d}_i - \sum_i \sum_j \sum_k \delta c_{ijk} N_i(u) N_j(v) N_k(w) \right|^2$$

After getting c'_{ijk}, the deformed surface is determined.

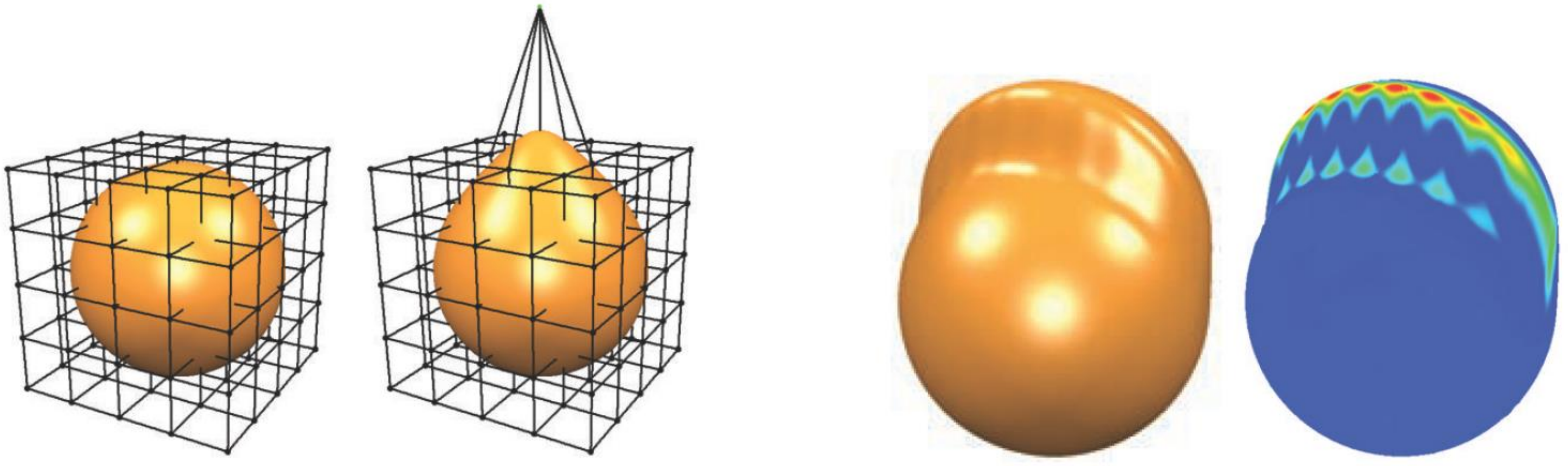


Figure 9.13. In the FFD approach a 3D control grid is used to specify a volumetric displacement function (left). The regular placement of grid basis functions can lead to alias artifacts in the deformed surface (right). (Image taken from [Botsch 05].)

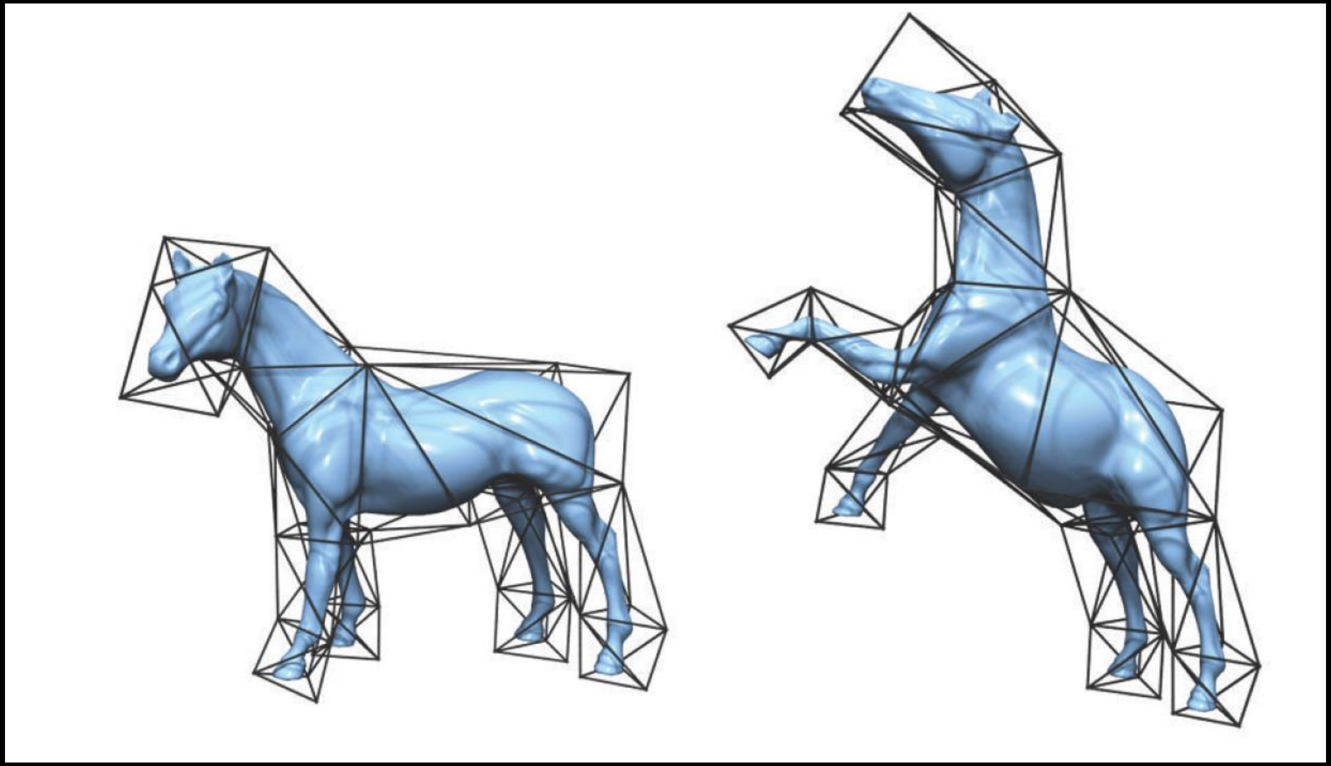
Discussion

- Two drawbacks:
  - Displacement constraints cannot be satisfied exactly.
  - The placement of basis functions on a regular grid.

- How to support concave region?

Cage-Based Freeform Deformation

- A generalization of the lattice-based freeform deformation
- This cage typically is a coarse, arbitrary triangle mesh enclosing the object to be modified.



Deformation

- The vertices p_i of the original mesh S:

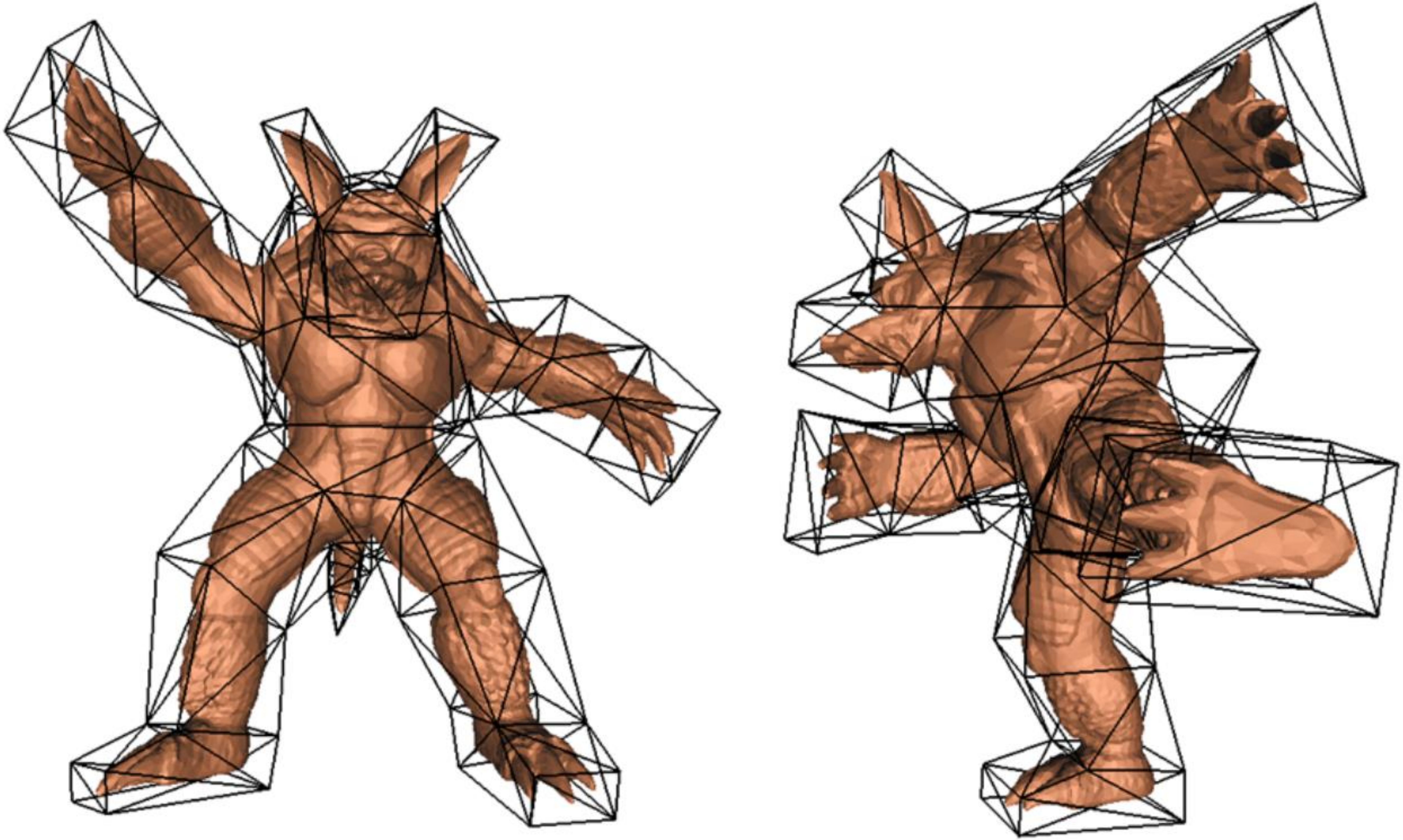
$$p_i = \sum_{l=1}^{n} c_l \varphi_l(p_i)$$

n: the vertex number of cage mesh

$\varphi_l(p_i)$: generalized barycentric coordinates

- Deform by manipulating the cage vertices $c_l \longmapsto c_l + \delta c_l$, displacement:

$$d(p_i) = \sum_{l=1}^{n} \delta c_l \varphi_l(p_i)$$



Outline

- Definition
- Transformation Propagation
- Multi-Scale Deformation
- Differential Coordinates
- Deformation transfer
- As-Rigid-As-Possible surface deformation
- Freeform Deformation
  - Meshless mapping
- Volumetric Deformation
  - Tetrahedral mapping

How to implement ARAP tetrahedral deformation?