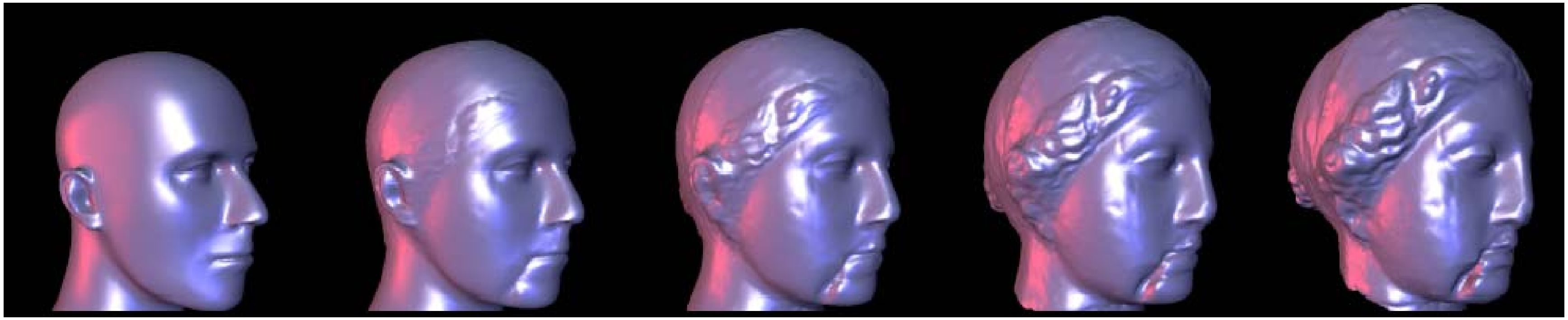
Graphics&Geometric Computing Lab
@USTC

Mesh Morphing

Ligang Liu
Graphics&Geometric Computing Lab
USTC
http://staff.ustc.edu.cn/~lgliu

Morphing

- Given two objects produce sequence of intermediate objects that gradually evolve from one object to the other
  - Interpolate object shapes
  - Interpolate object attributes
    - Color, texture, normal, etc.



Terminologies

- Morphing
- Metamorphosis
- Shape blending
- Shape averaging
- Shape interpolation
- Shape transition

Applications

- Scientific visualization
- Education
- Entertainment
- Shape modeling
- Key frame animation
  - gives the animator the ability to “fill” an animation between key-framed objects

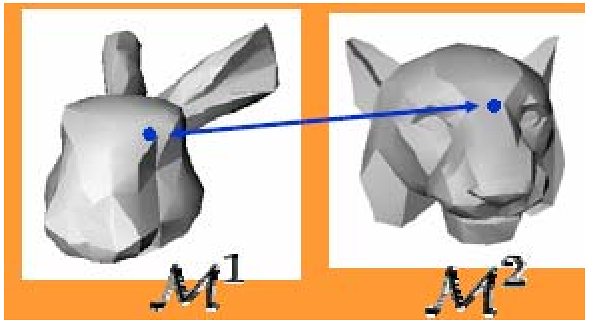
Rules for Good Morphing

- Natural
  - Keep as much as possible of the two shapes during the transformation
    - Volume, curvature, area, etc...
  - Subjective aesthetic criteria
- User control
  - intuitive
  - not too heavy
  - can be adapted to user's knowledge

Morphing

- Input: two meshes source & target
  - Frames at t_0 and t_n
- Output: sequence of intermediate meshes
  - Frames t_1 to t_{n-1}
- Intermediate mesh:
  - For each point on source/target model specify location at time t_i consistent with source & target

Morphing: Sub-Problems



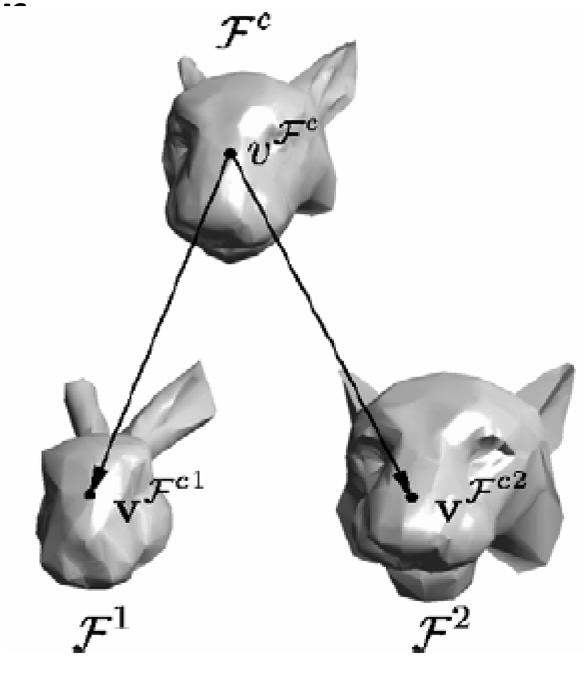
- Correspondence problem
  - – For each point on source/target meshes find corresponding point on second mesh = Parameterization
- Path problem
  - – Specify trajectory in time for each point
  - – For mesh – specify vertex trajectory

Morphing

- Vertex correspondence:
  - Each vertex on source mesh mapped to vertex on target (and vice versa)
  - Have common connectivity
- Algorithm stages
  - Compute mapping between source & target
  - Compute common connectivity (remesh models)
  - Compute trajectory for each vertex

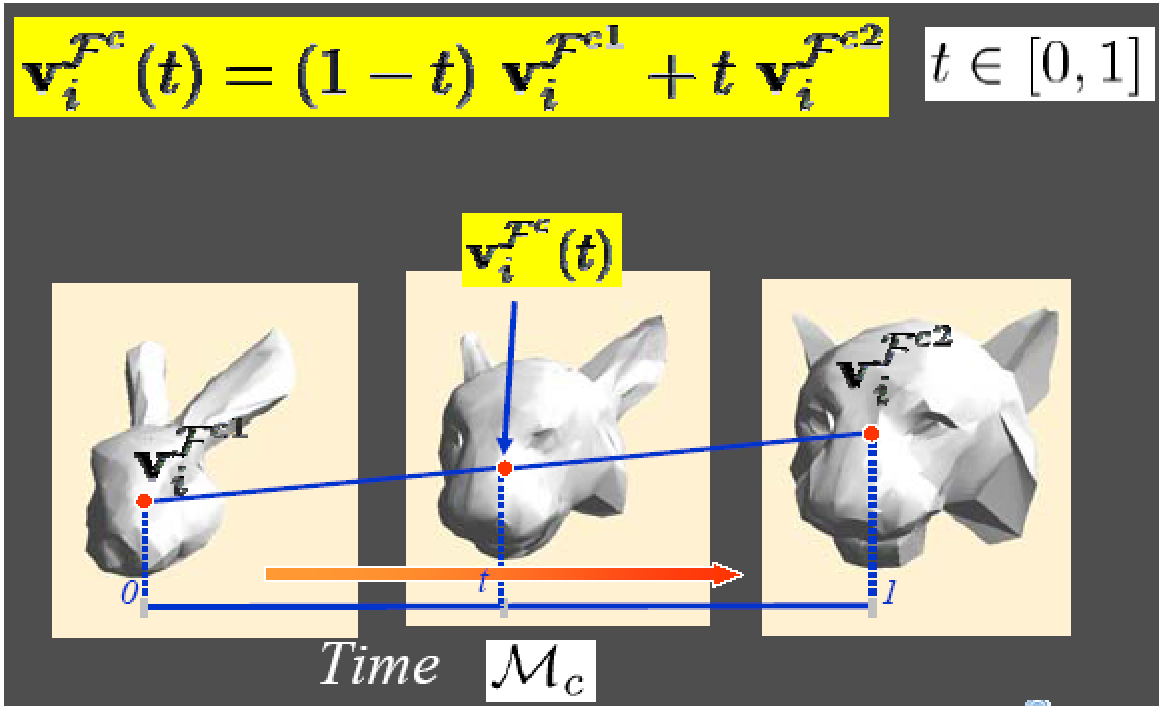
Path

- All vertices on source & target have one-to-one correspondence with each othe
- Each vertex has two 3D coords v^{Fc1} (source) and v^{Fc2} (target)



Linear Interpolation

- Linear interpolation between corresponding points



$$\mathbf{v}_i^{\mathcal{F}^c}(t) = (1-t)\,\mathbf{v}_i^{\mathcal{F}^{c1}} + t\,\mathbf{v}_i^{\mathcal{F}^{c2}}$$

$t \in [0,1]$

$\mathbf{v}_i^{\mathcal{F}^c}(t)$
$\mathbf{v}_i^{\mathcal{F}^{c1}}$
$\mathbf{v}_i^{\mathcal{F}^{c2}}$
0
t
1
Time
$\mathcal{M}_c$

Trajectory

- Linear
  - simple but has many drawbacks
- Better methods exist in 2D
- Very little done in 3D

Correspondence: Parameterization

- To compute map between source mesh S and target mesh T parameterize both on common domain D:

$$F_s\!:S \rightarrow D$$

$$F_t\!:S \rightarrow D$$

$$F_{st} = F_t^{-1} F_s$$

- Common domain options
  - 2D patch(es) – works for genus 0 + boundary
    - Use convex boundary (why?)
  - Sphere
  - Base mesh

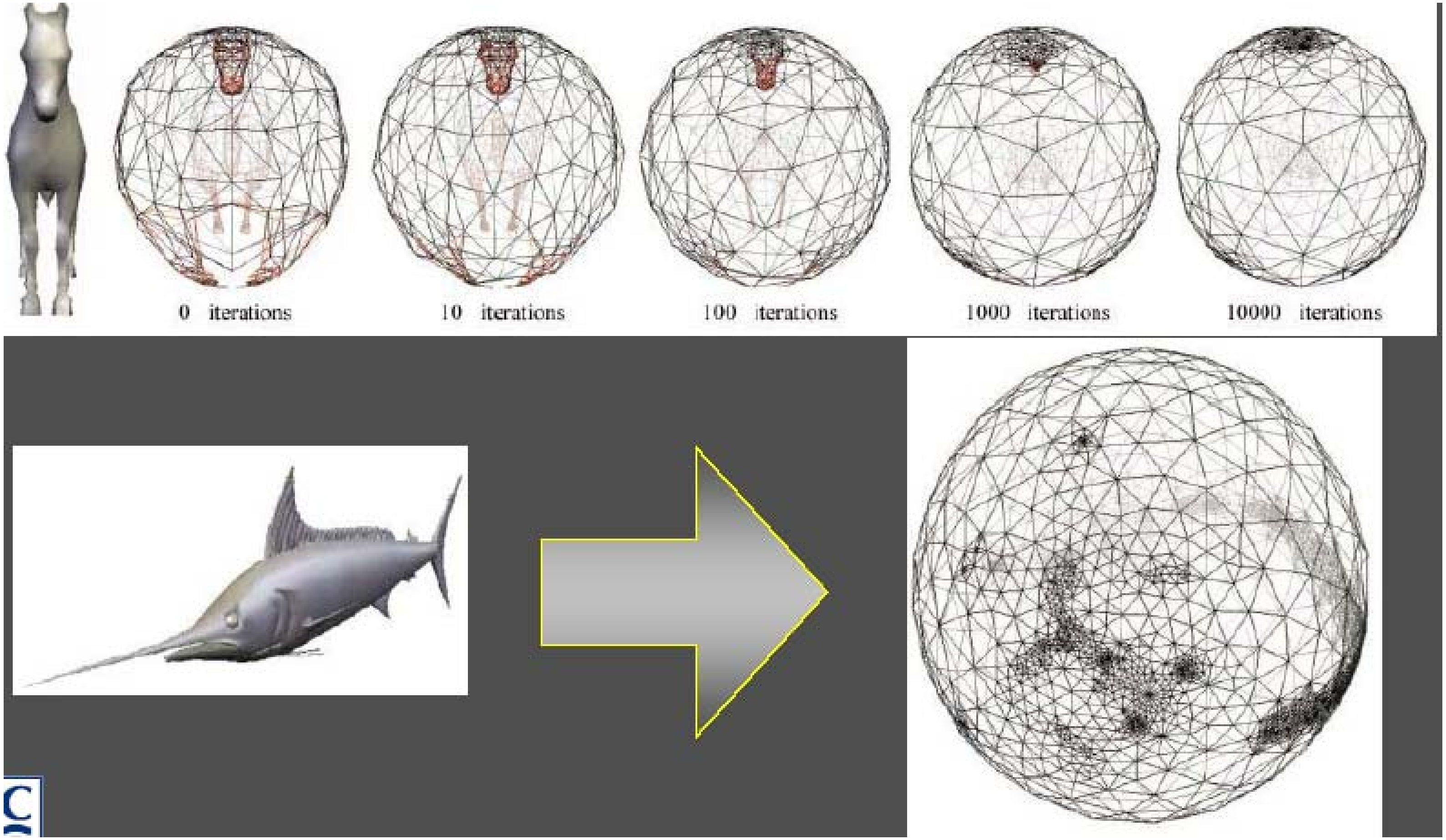
Spherical Embedding

- Project model onto unit sphere through center of mass
- Repeat till embedding is valid:
  - – Embedding is valid if & only if all faces are oriented the same way
  - – Recalculate all vertices
    - Set

$$P_i = (1 - c)P_i + c\frac{\sum_{\{i,j\}\in E} v_j \left\| v_i - v_j \right\|}{\sum_{\{i,j\}\in E} \left\| v_i - v_j \right\|}$$

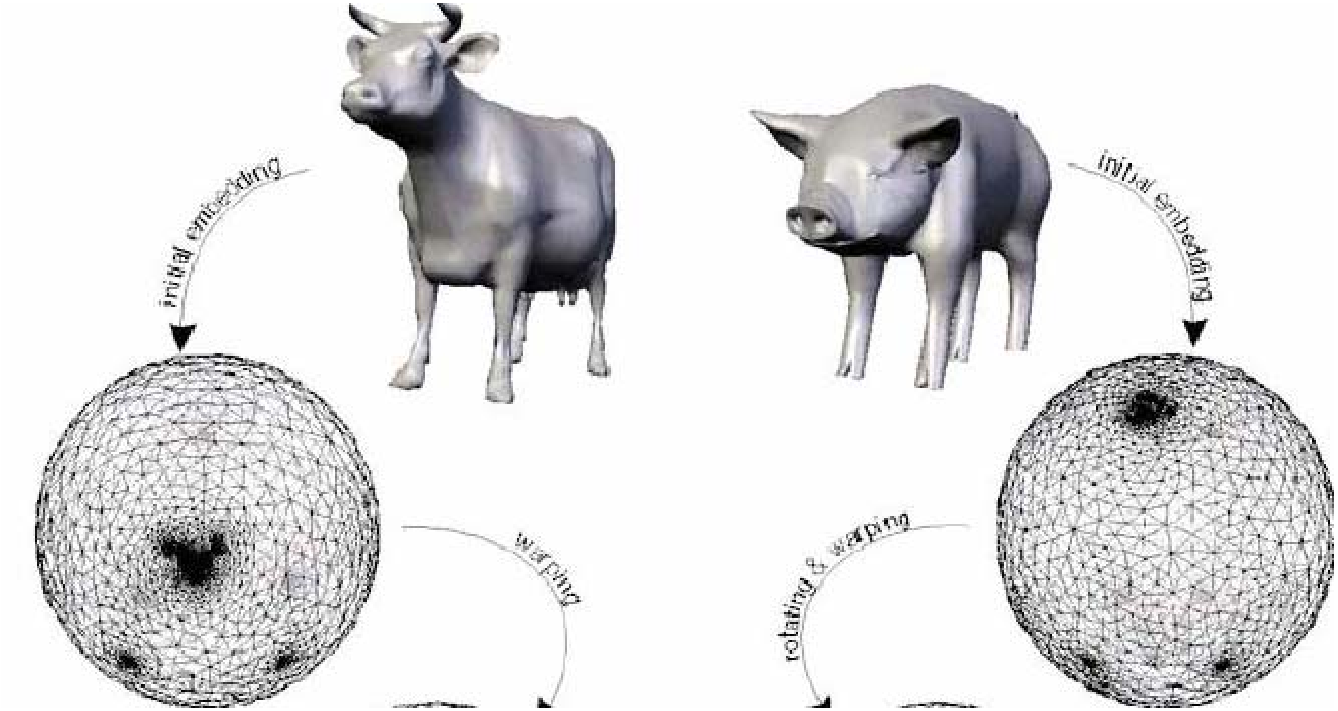
    - Project to sphere

Embedding


0 iterations
10 iterations
100 iterations
1000 iterations
10000 iterations

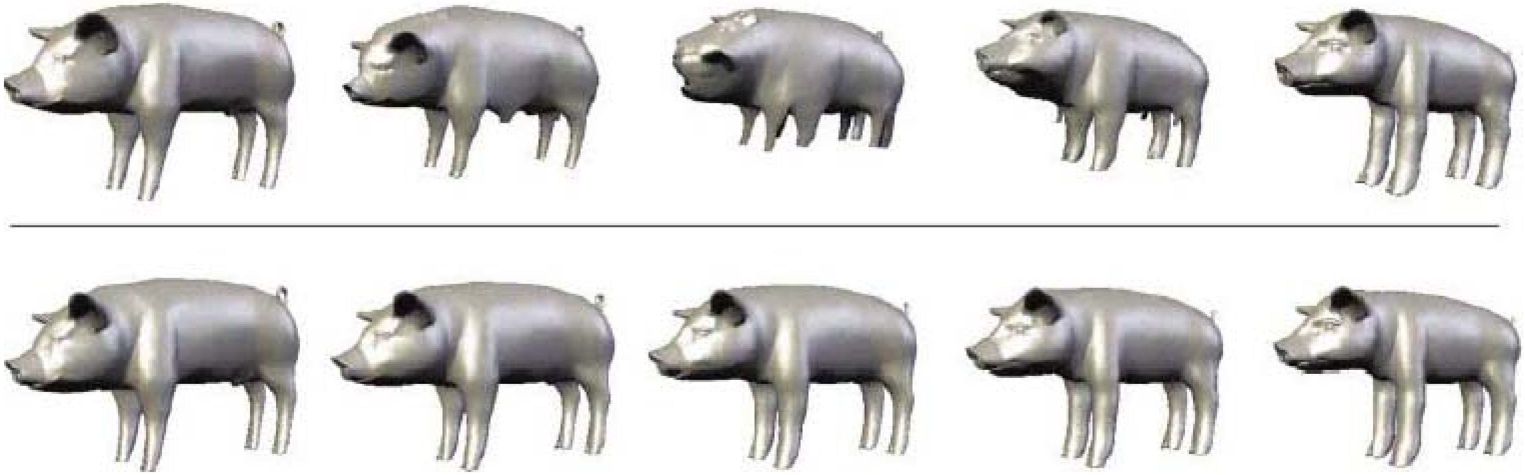
Spherical Embedding

- Drawbacks
  - Can fail
  - No geometry preservation



Feature Alignment

- For natural looking morph must align matching features

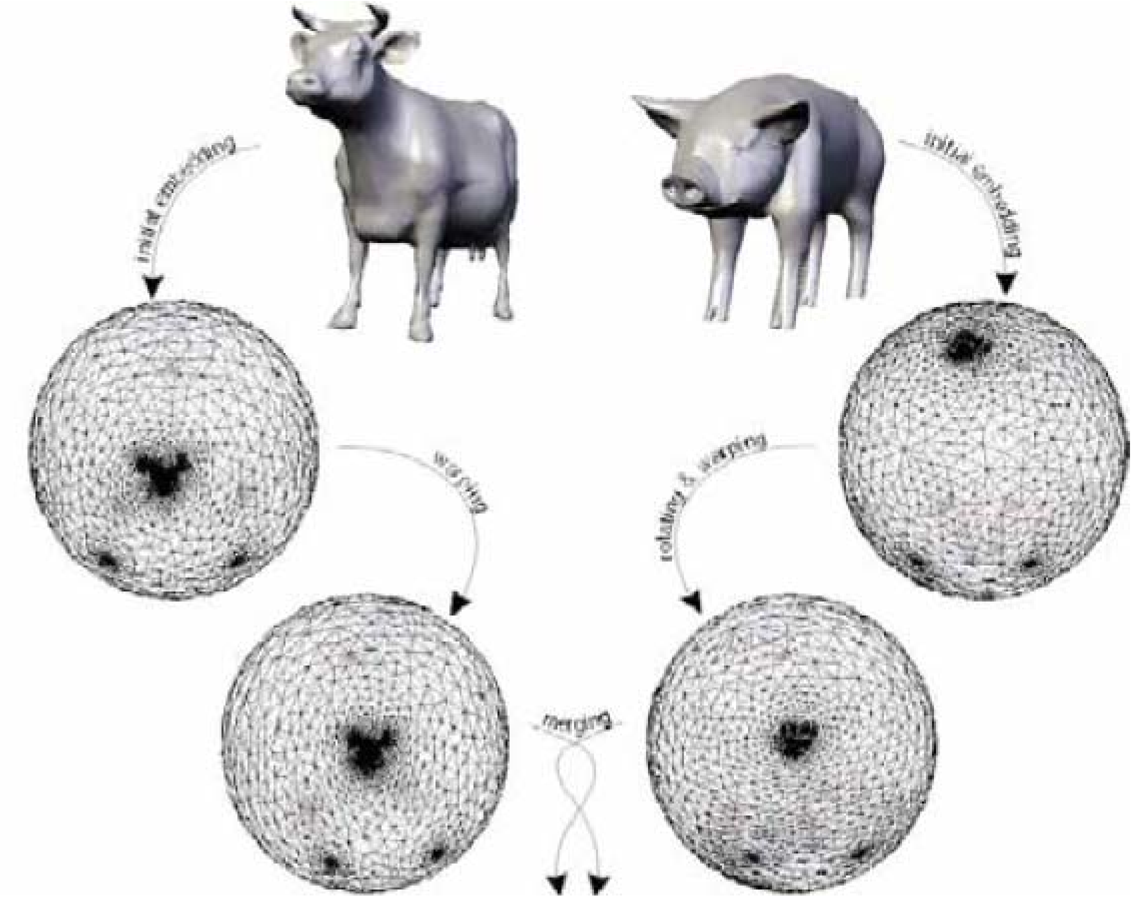


Align Features

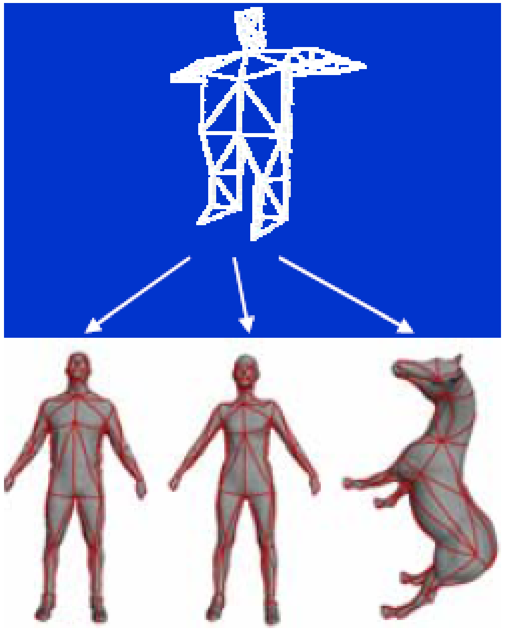
- Rotate one sphere to minimize sum of squared distances between corresponding feature vertices
- Warp surfaces of both spheres to reduce distances between corresponding feature vertices
  - Define region of influence of feature vertex
  - Move vertex towards matching feature vertex
  - Move region vertices (with diminishing influence)
  - If generate fold-over reduce motion step or region size
  - Repeat till vertices are aligned or step/region size too small

Feature Alignment

- Works only when features are relatively close by



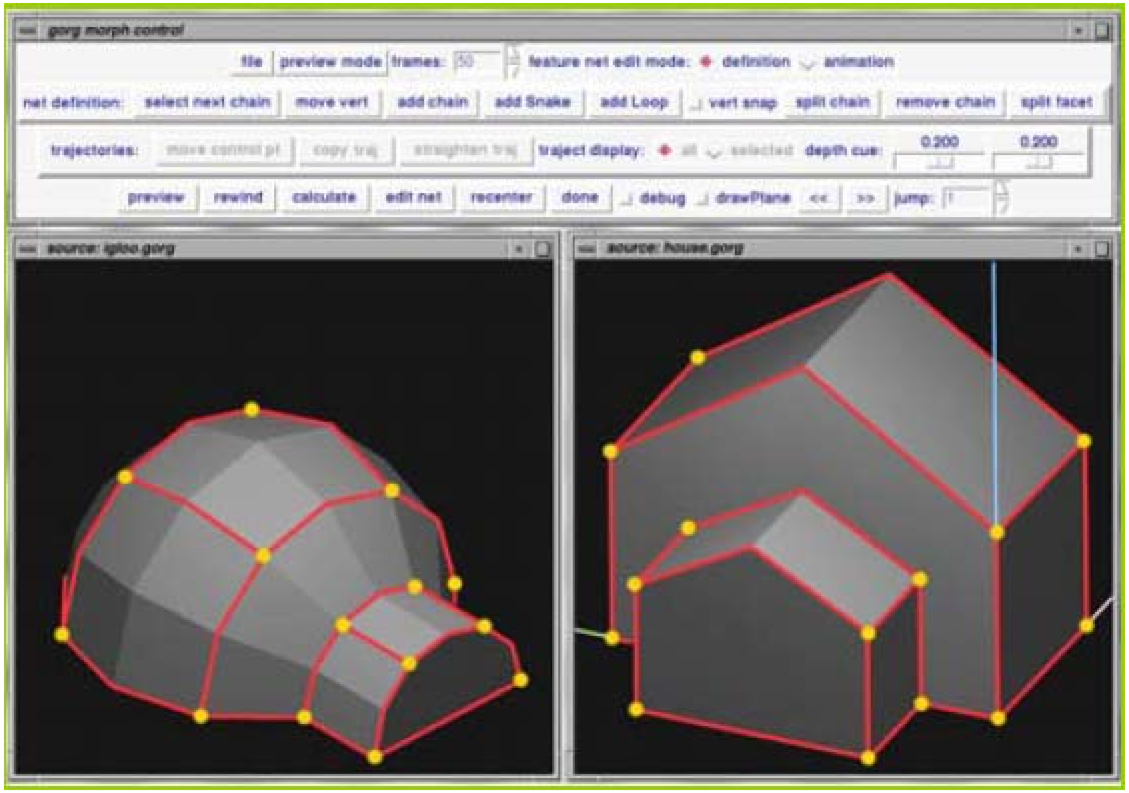
Base Mesh



- Define single base mesh connectivity for both source and target
- Split source and target models into patches with base connectivity
- Map patches to base mesh faces & use combined mapping to define correspondence
- Features: use to define base mesh vertices

Manual Patch Specification

- Define feature-net decomposing input meshes into matching patches



gorg morph control

frames: 50

feature net edit mode: ● definition ○ animation

trajecct display: ● all ○ selected

depth cue: 0.200 0.200

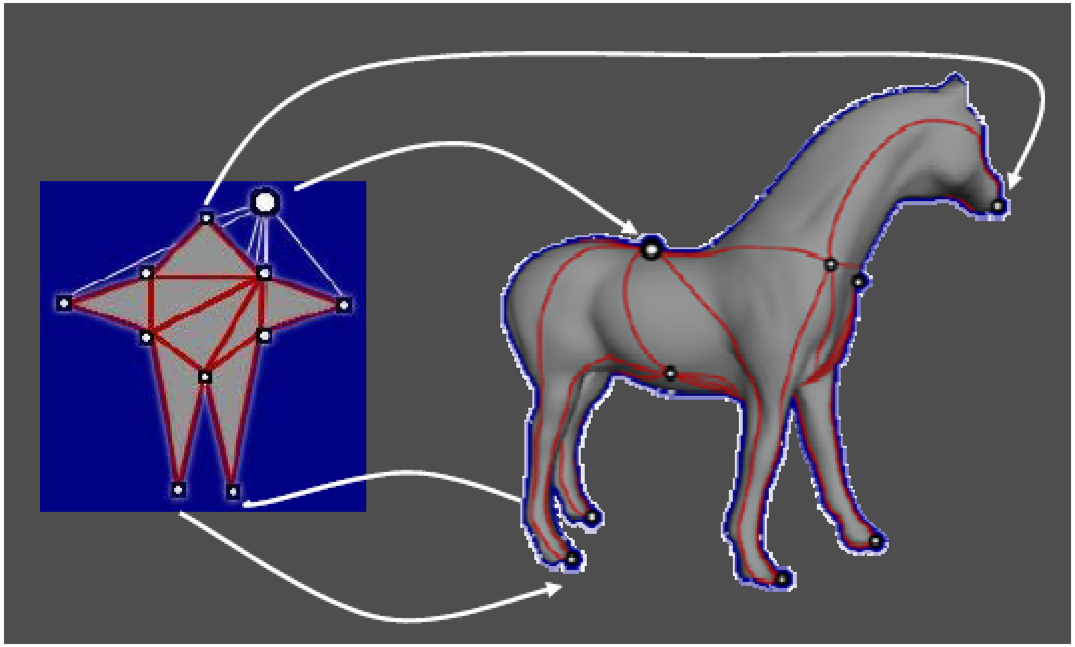
jump: 1

source: igloo.gorg

source: house.gorg

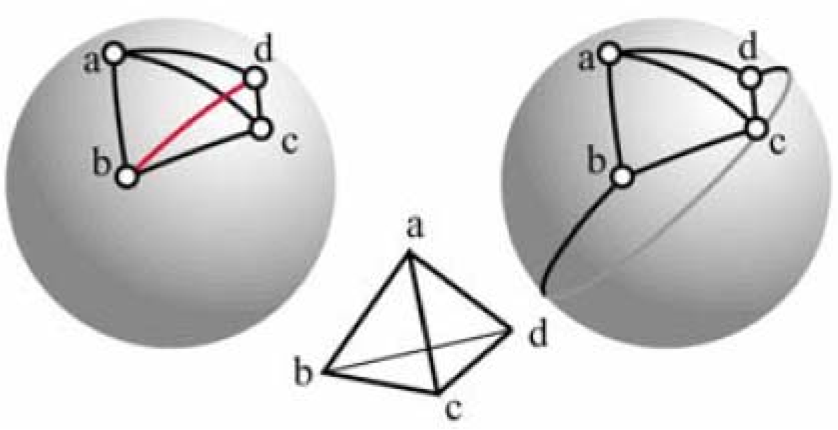
Patch Computation

- User input: base mesh
- Algorithm: trace base mesh edges as paths on source/target models



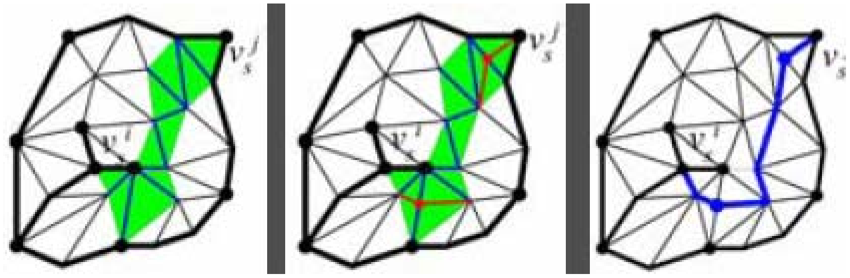
Tracing Paths

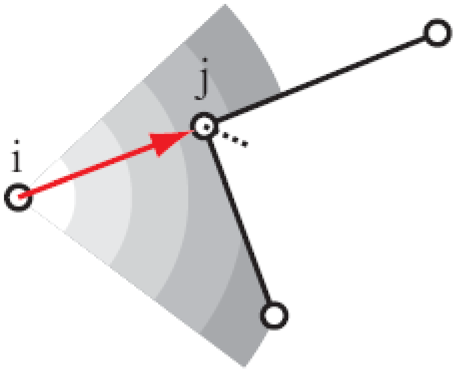
- Paths net topologically equivalent to base domain
  - Paths intersect only at vertices
  - Same neighbor ordering around vertices
  - Correct orientation

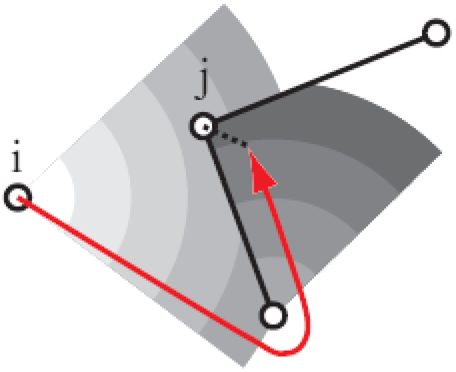


Tracing Paths

- Trace face paths from vertex i to j
  - Convert to edge paths
- Use restricted BFS traversal:
  - Do not cross existing paths
  - Start & end in correct sector

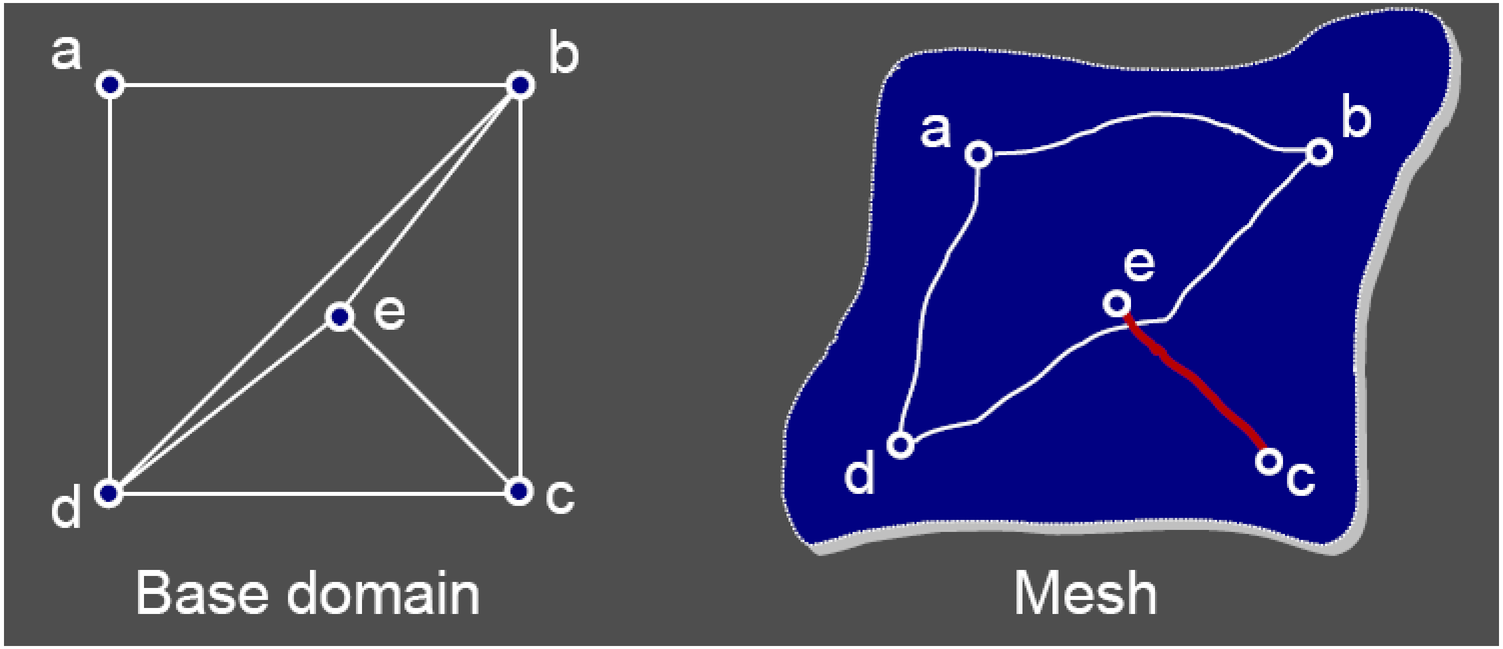






Problem: Encircling

- To avoid, first trace spanning tree



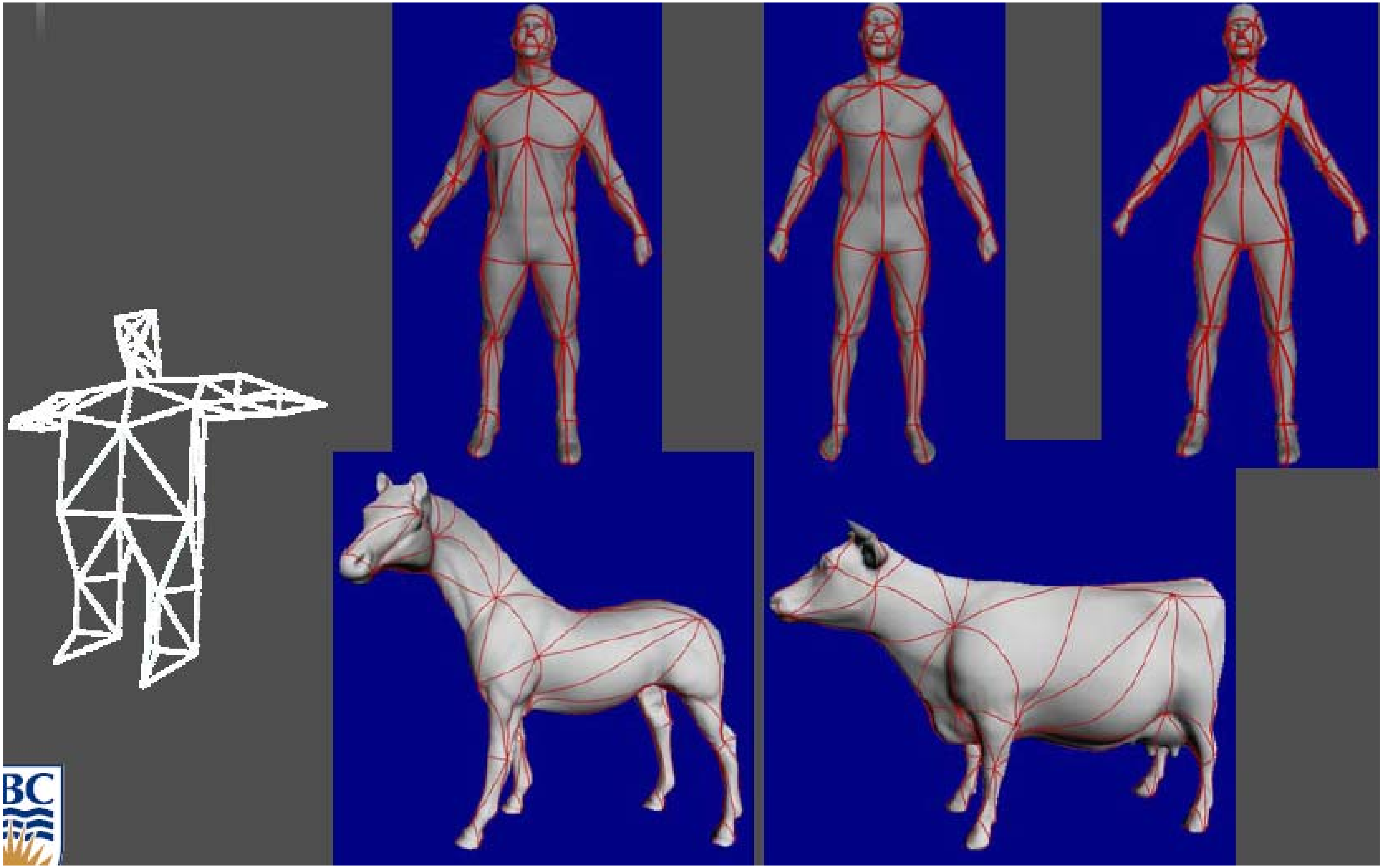
Algorithm Issues

- Guarantee topological equivalence of traced net and base domain
  - Trace curves with restricted BFS
  - Complete spanning tree before adding cycles
- Patch shape
  - Introduce curves in order of length (shorter is better)

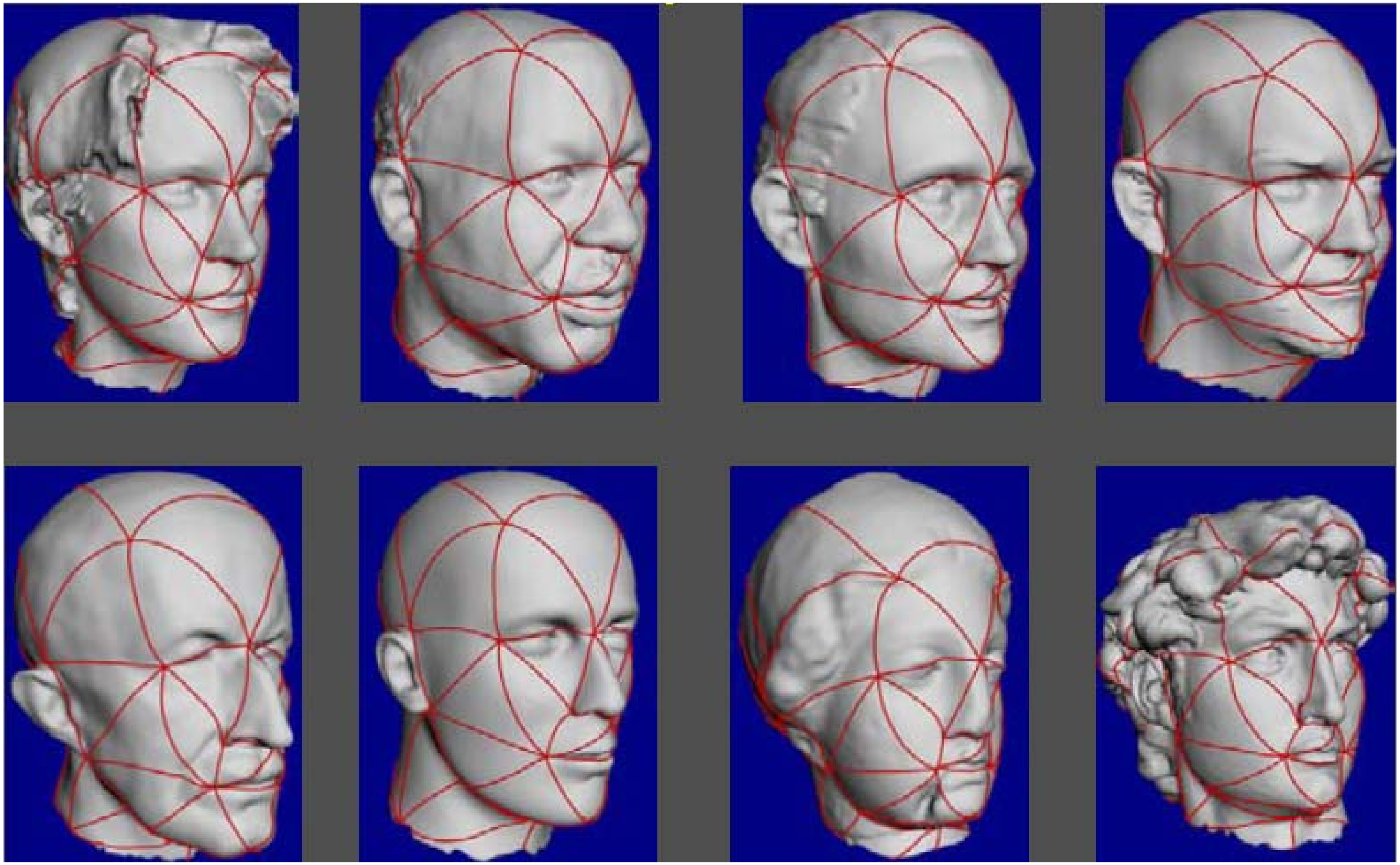
Algorithm Stages

- First stage: complete spanning tree
- Second stage: complete whole net
  - For each stage, keep priority queues
    - Queues contain candidate curves
    - May need to update to enforce topology
- Third stage: Edge straightening

Examples



Example



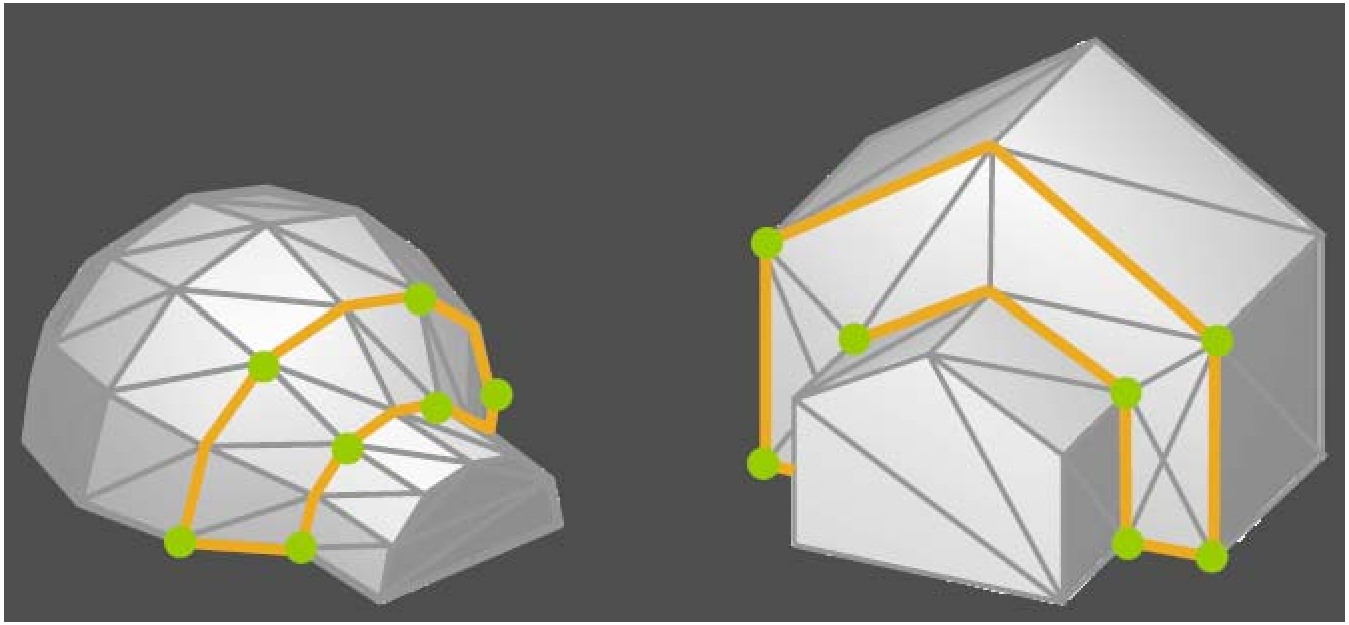
Common Connectivity

- Input: two models parameterized on common domain
- Output: both models remeshed with common connectivity (preserving point correspondence)
- Methods:
  - Overlay
  - Subdivision meshing

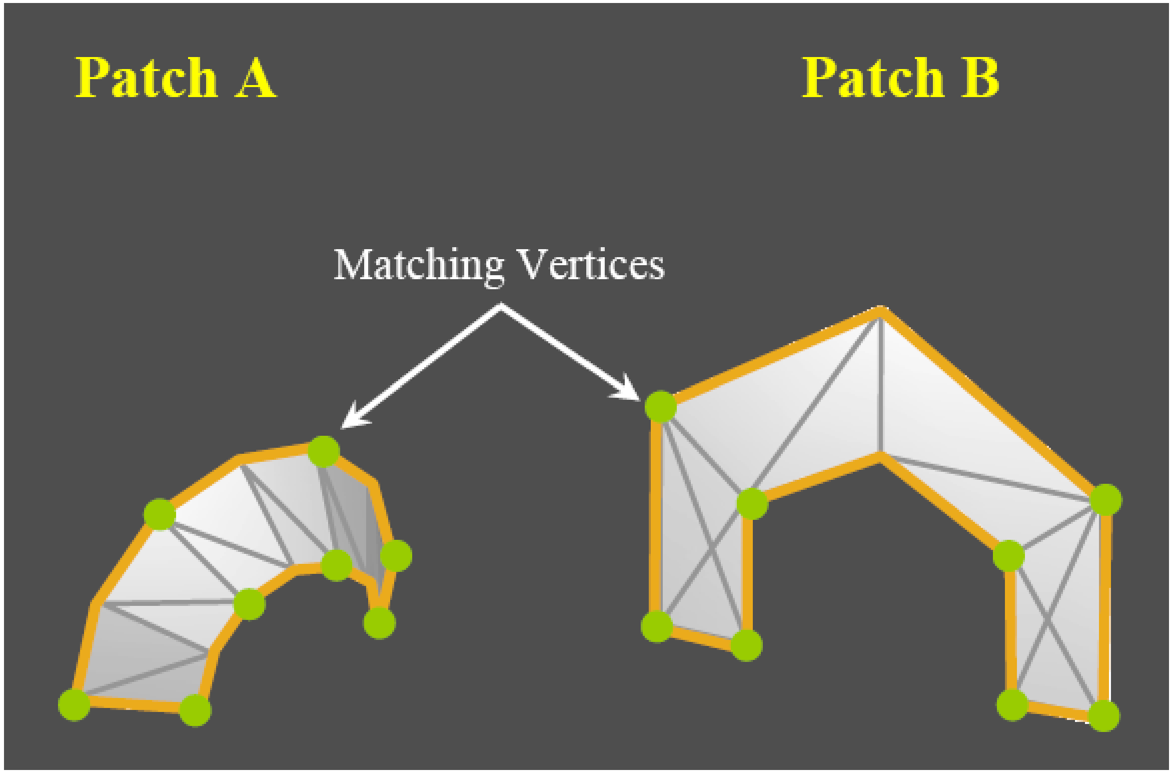
Overlay

- Map both models to base domain
  - Sphere: use spherical mapping
  - Base mesh: use one pair of patches at a time
- Merge vertex-edge graphs
- Reconstruct facets
- Project back using barycentric coordinates on original source/target mesh triangles

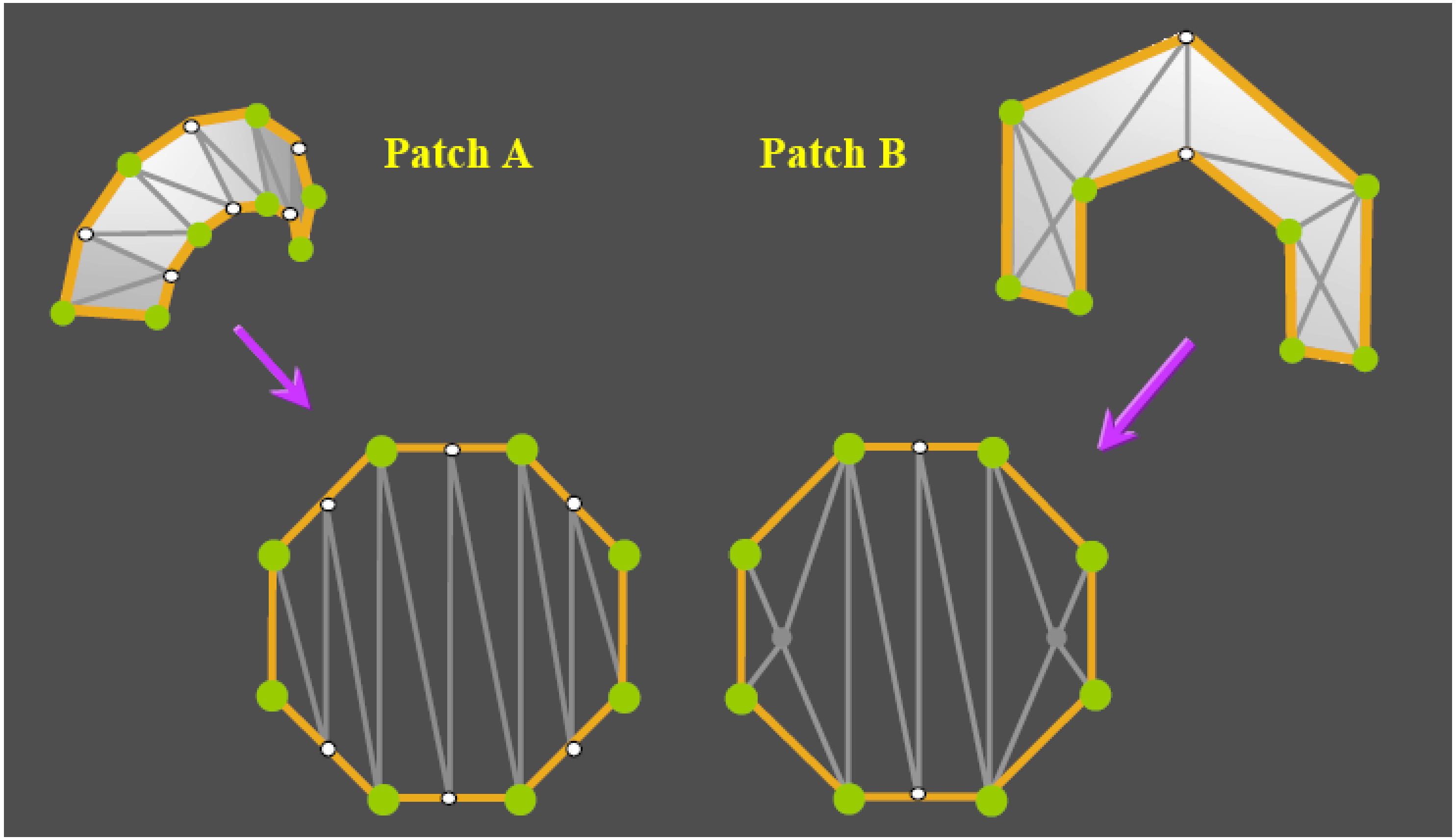
Correspondence Computation



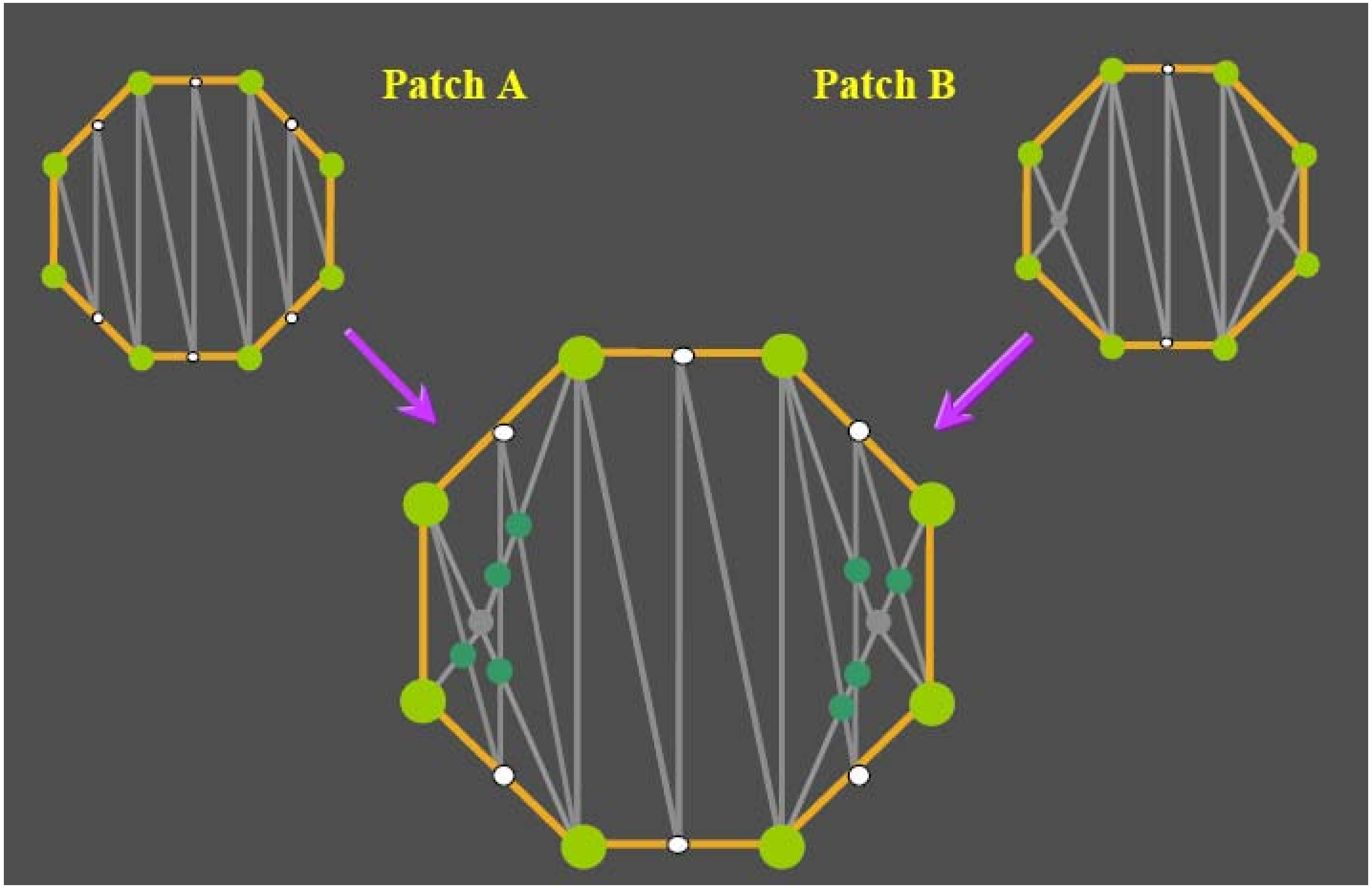
Correspondence Computation



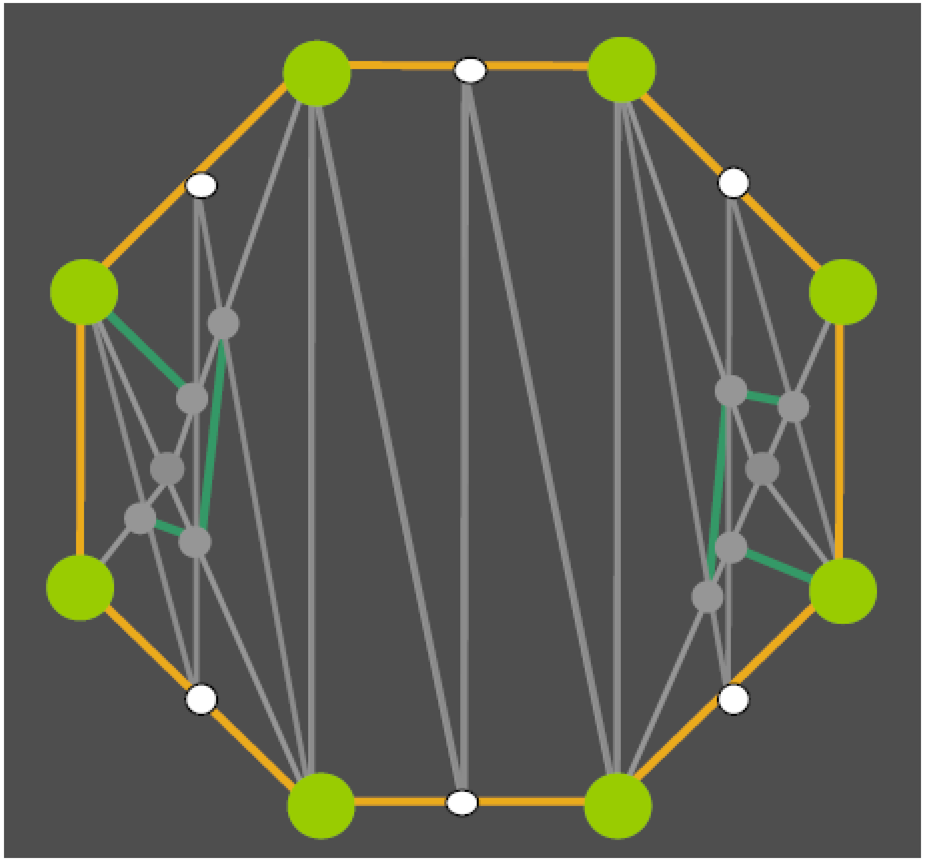
Mapping



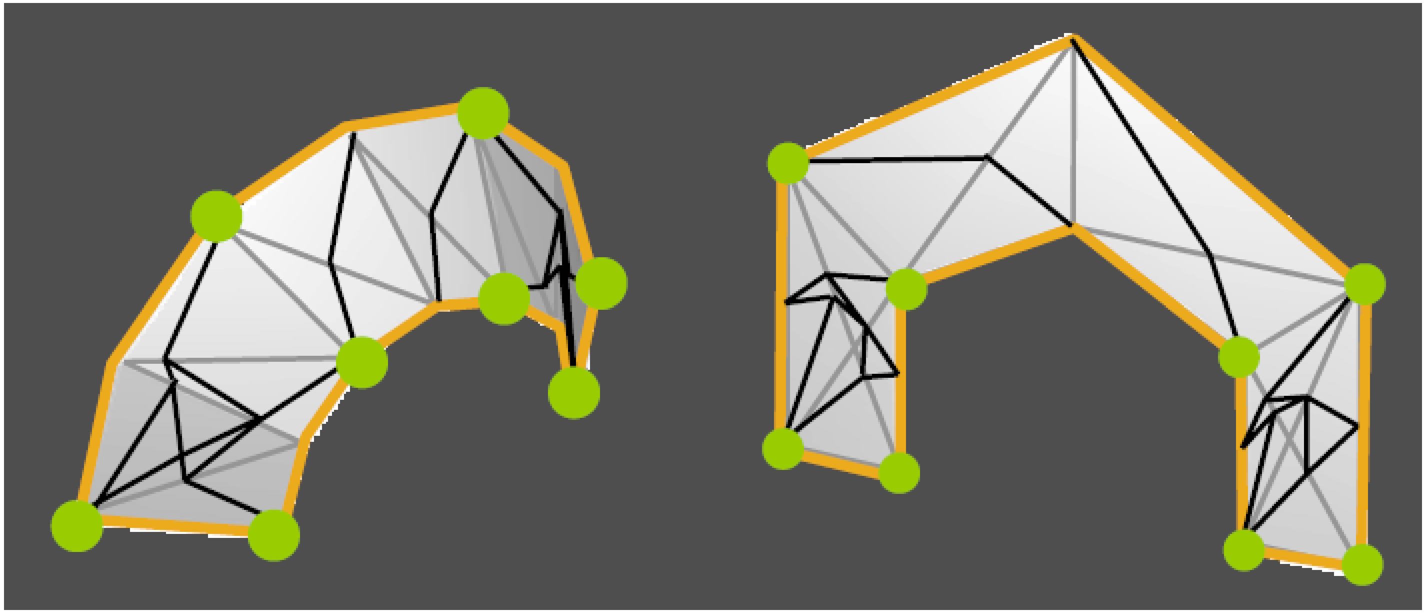
Merging



Reconstruction

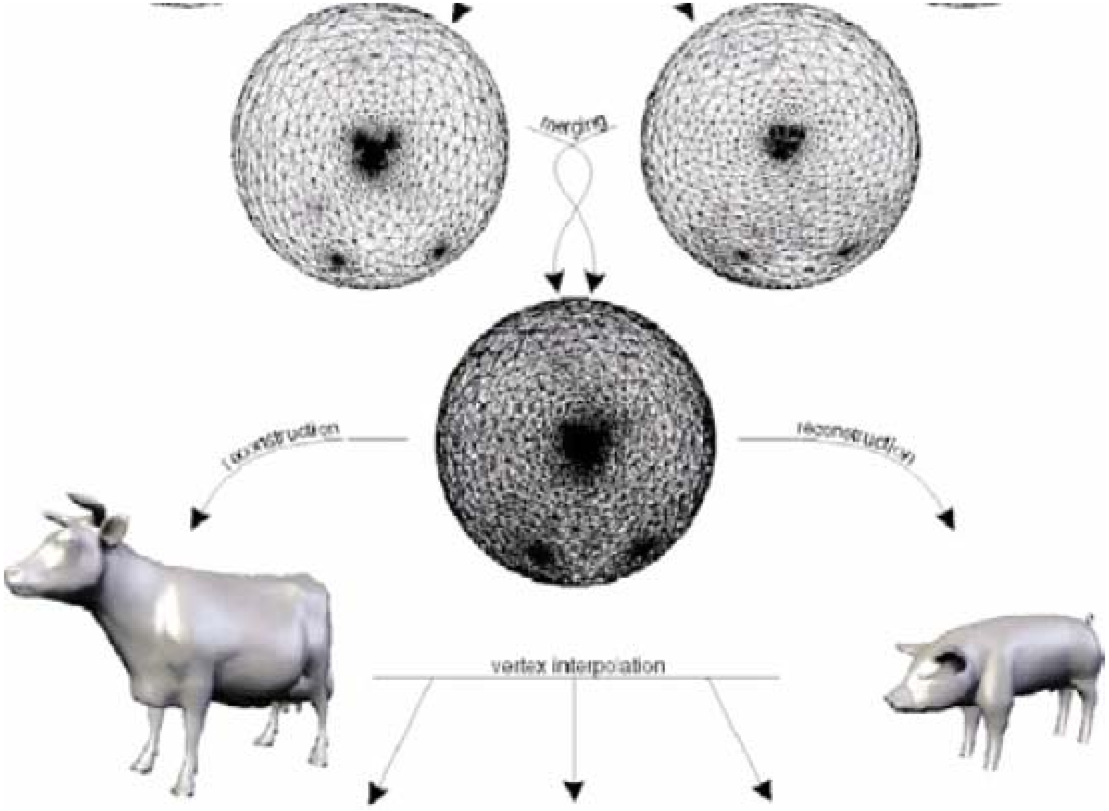


Completed Correspondence



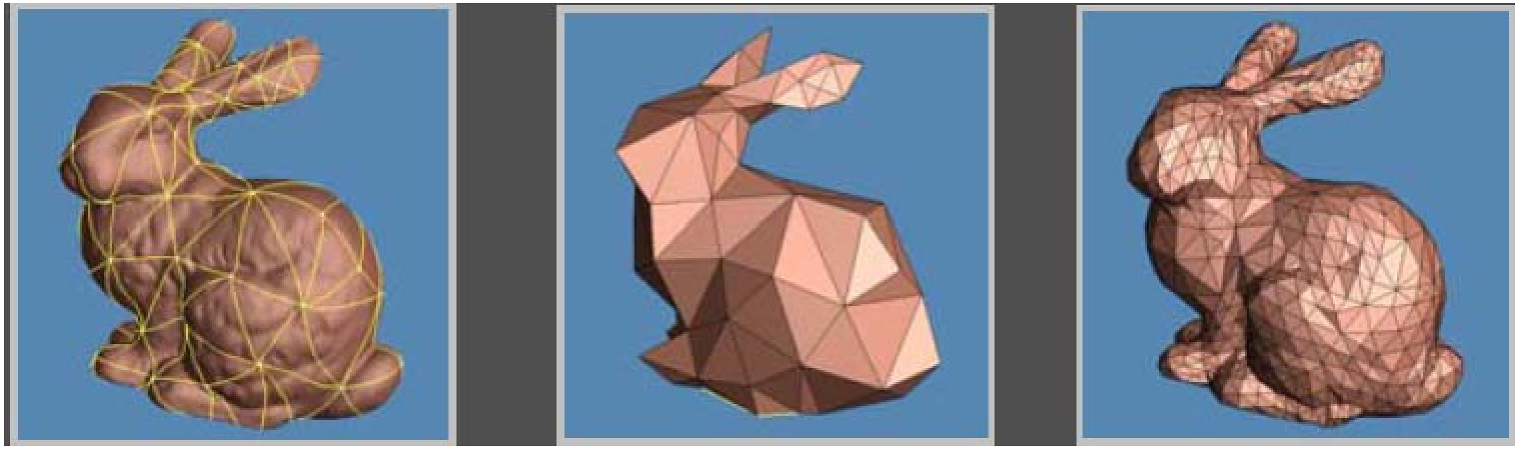
Sphere

- All computations (intersections, etc..) on sphere



Subdivision Remeshing

- Works for triangular base mesh
- Mesh each base mesh triangle to required density using subdivision pattern
- Project back to source/target meshes using parameterization

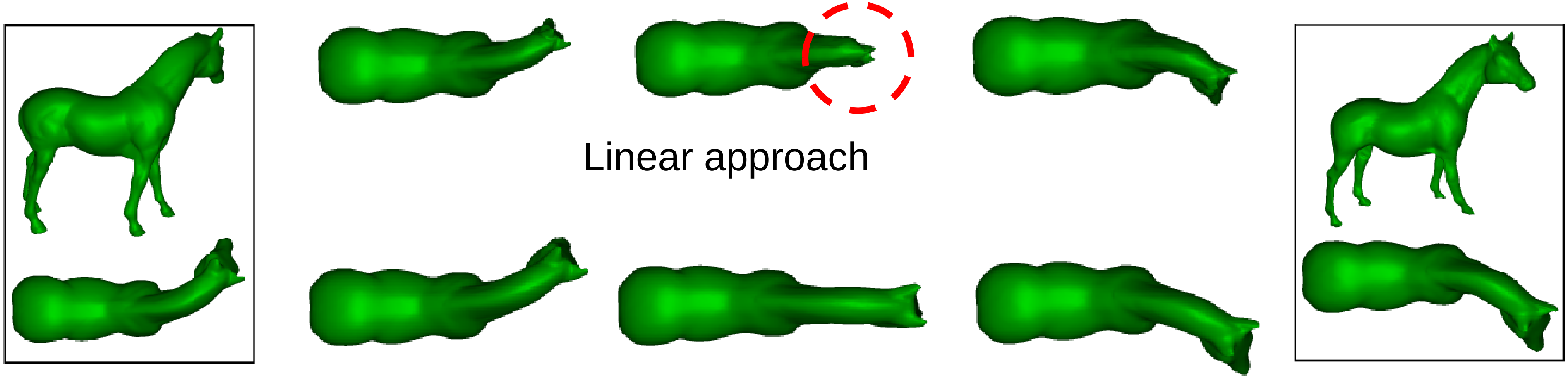


Methods Pro/Cons

- Overlay
  - Pros:
    - Preserves source/target geometry
    - Not ‘very’ affected by parameterization distortion
  - Cons:
    - Increases mesh size by x10
    - Very labor intensive to implement
- Subdivision
  - Pros:
    - Simple
    - Nice mesh if patch layout is good
  - Cons:
    - Approximation only – depend on resolution
    - Depends very strongly on patch shape & parametric distortion

Dual Laplacian Morphing

- Jianwei Hu, Ligang Liu, Guozhao Wang. Dual Laplacian Morphing for Triangular Meshes. Proceedings of CASA 2007.



Linear approach

Novel approach

Demo

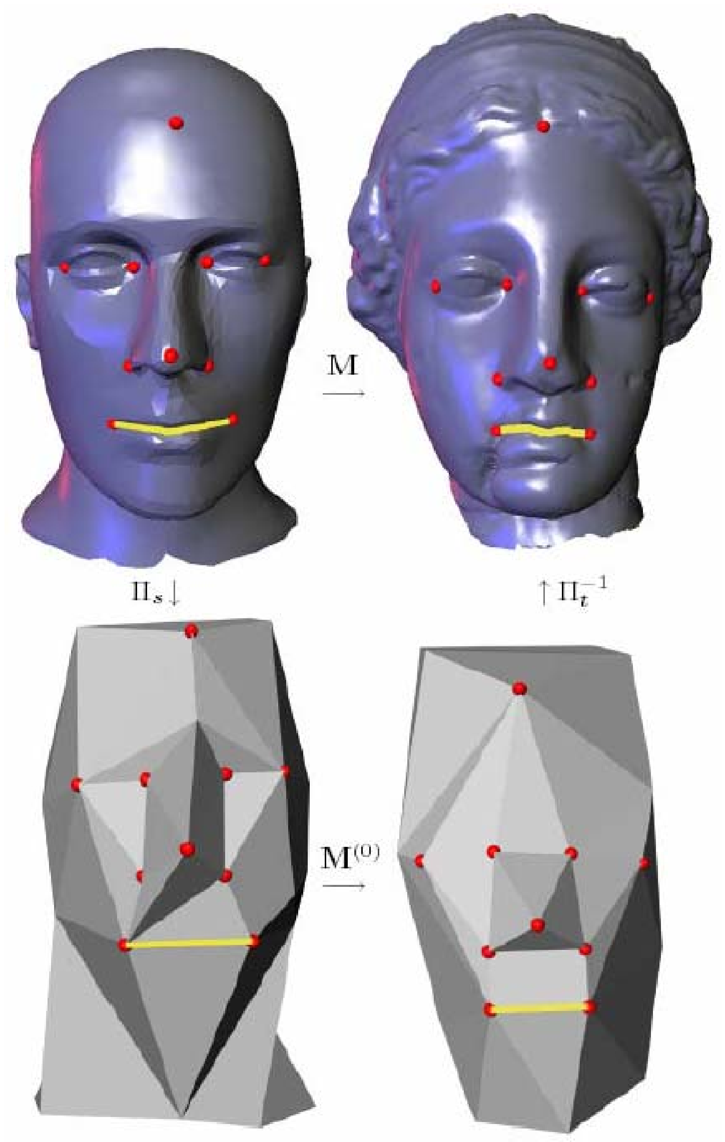
Summary

- Mesh morphing – important animation component
- Two parts:
  - Matching
    - Hard
    - Some algorithms exist
  - Trajectory computation
    - Most algorithms assume linear (given good matching)
    - Better algorithms exist in 2D – no adequate 3D equivalents

More…

Multiresolution Method

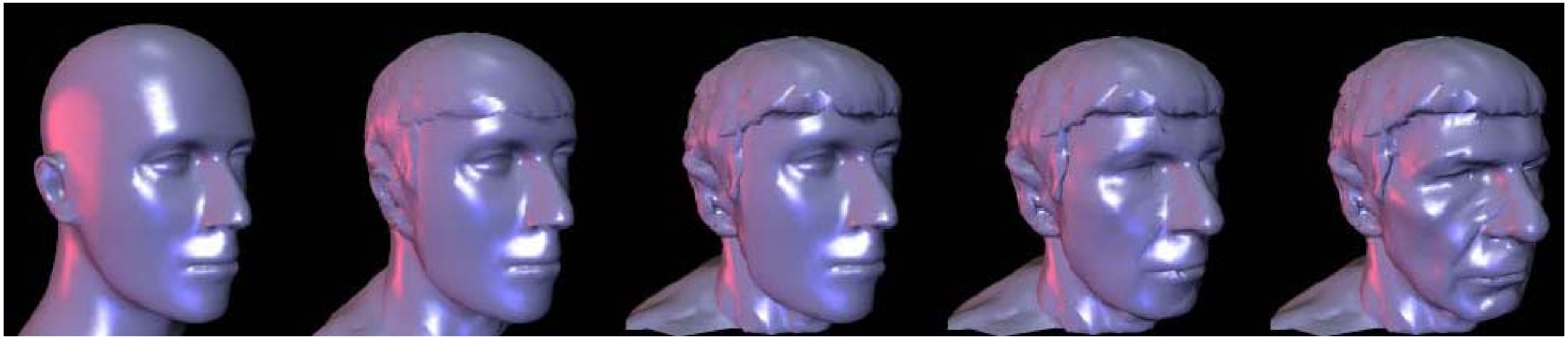
Lee et al. 1999

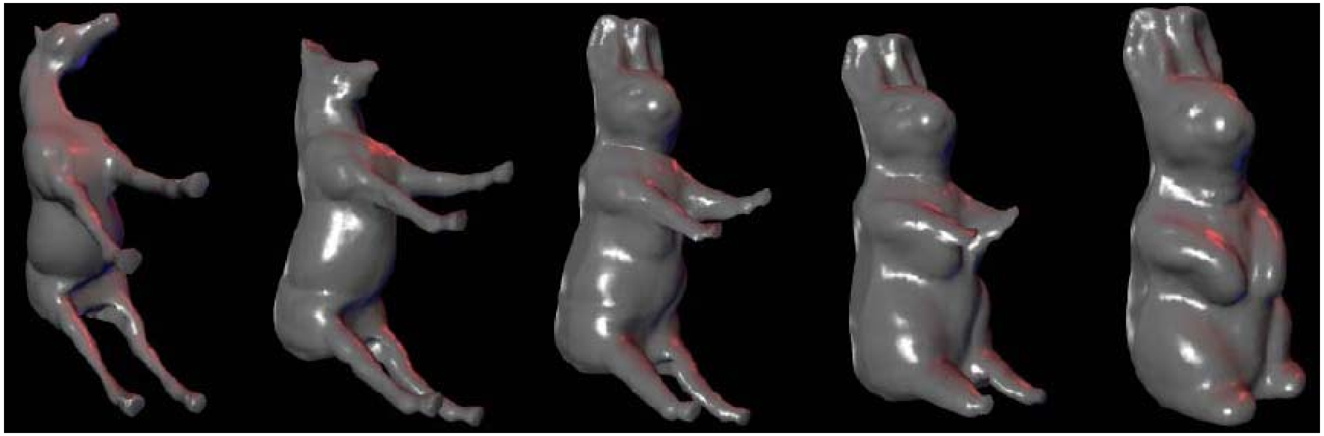


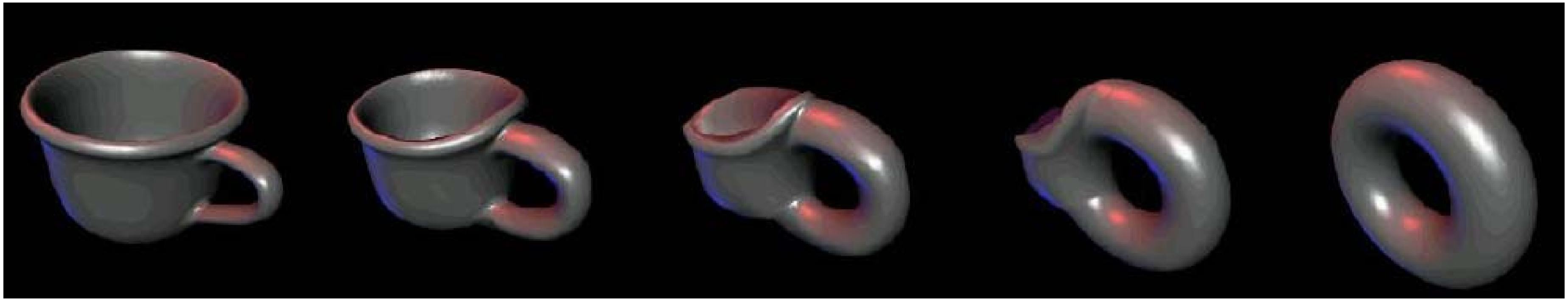
Π_s and Π_t^{-1} are computed using MAPS

$M = \Pi_t^{-1} M^{(0)} \Pi_s$

Results

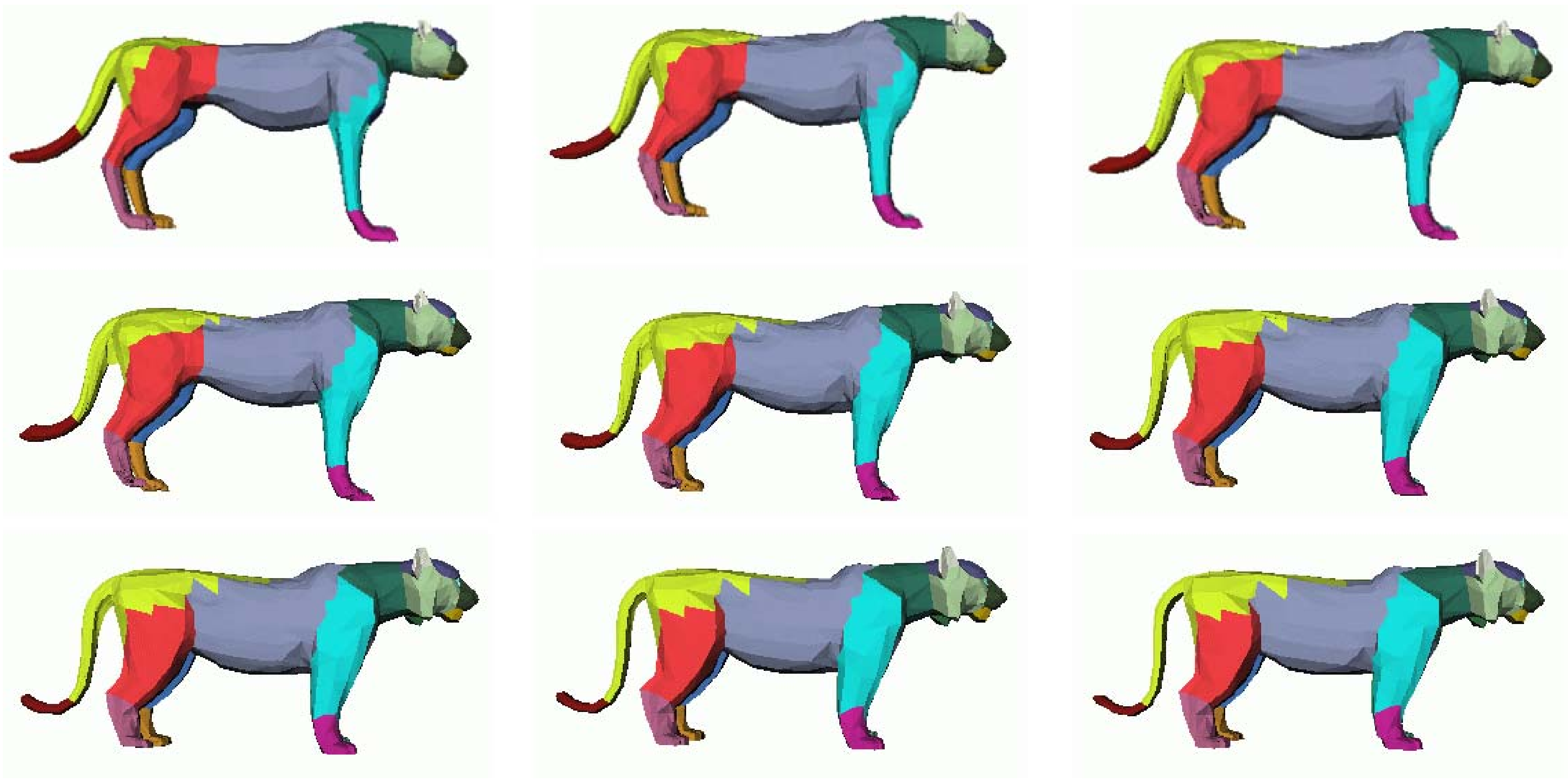






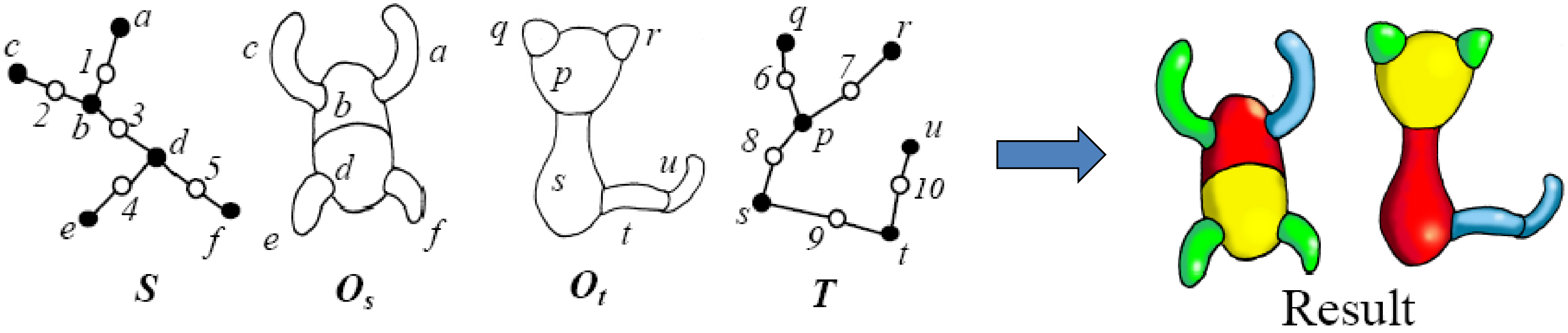
Decomposition Based

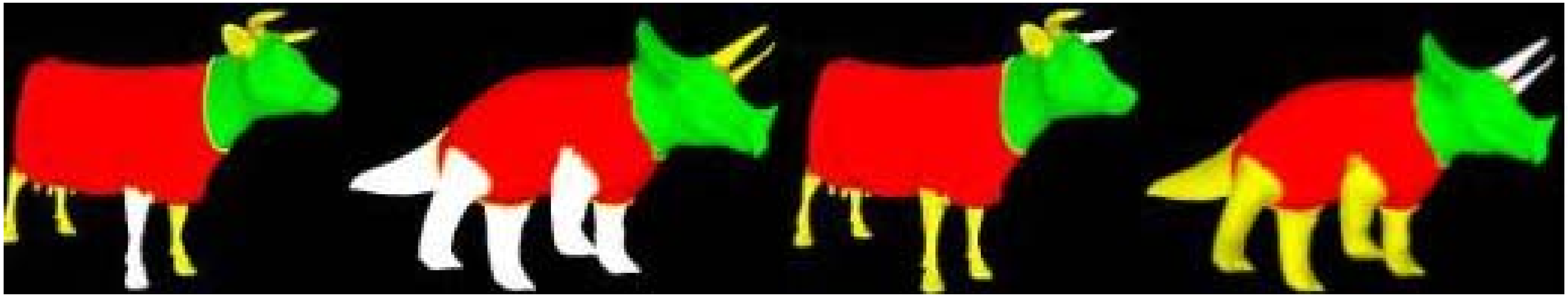
[Shlafman et al. 2002]



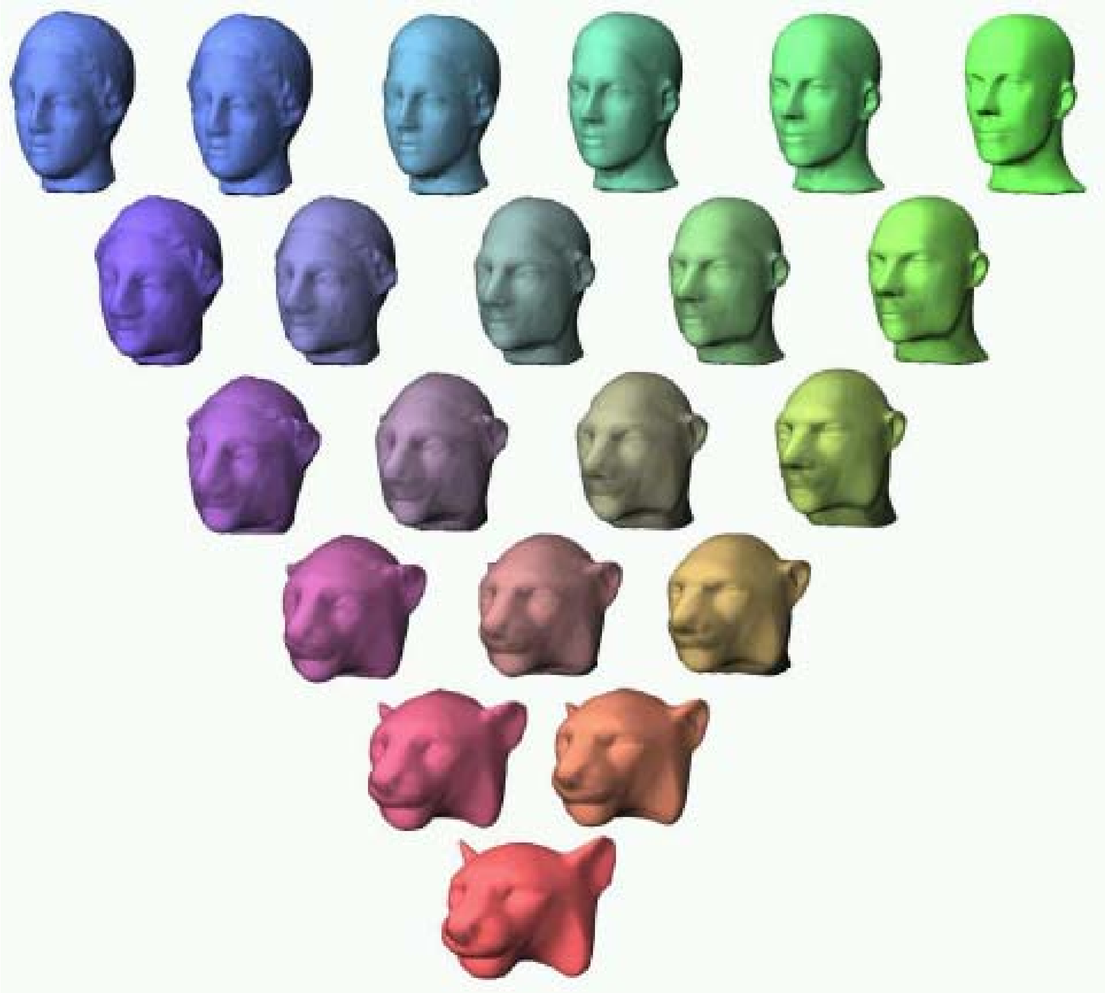
Component Based

[Zhao et al. 2003]





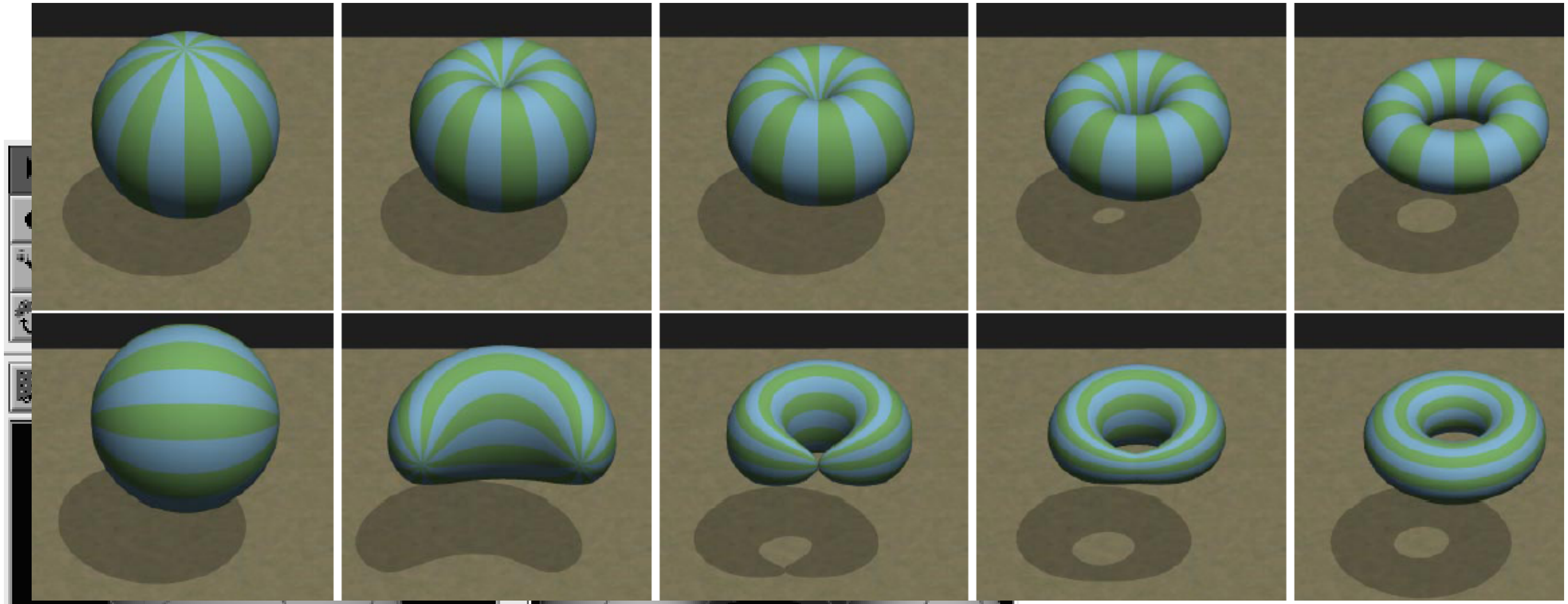
Polymorph

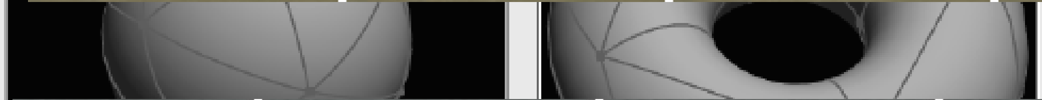


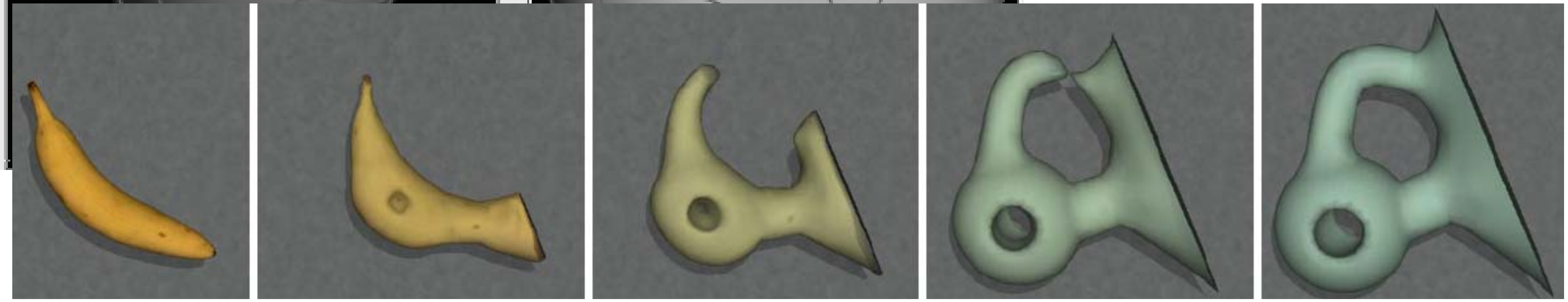
Discussions

Mesh Morphing with Different Topologies

[DeCarlo et al. 1996]





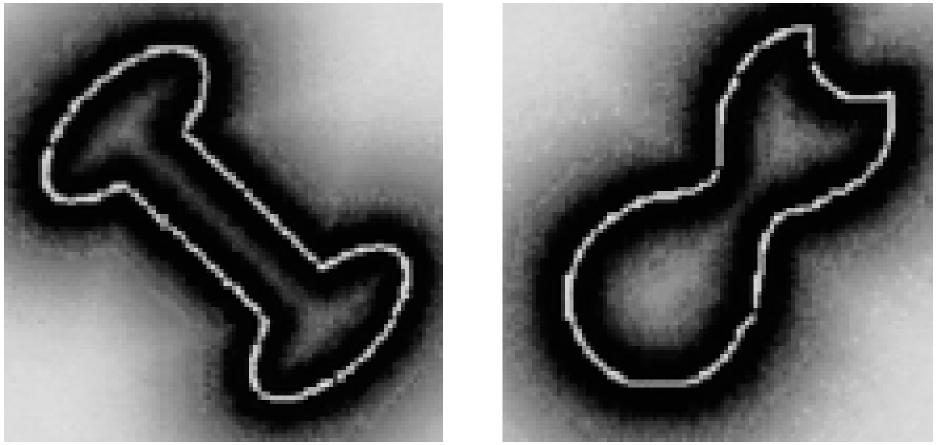


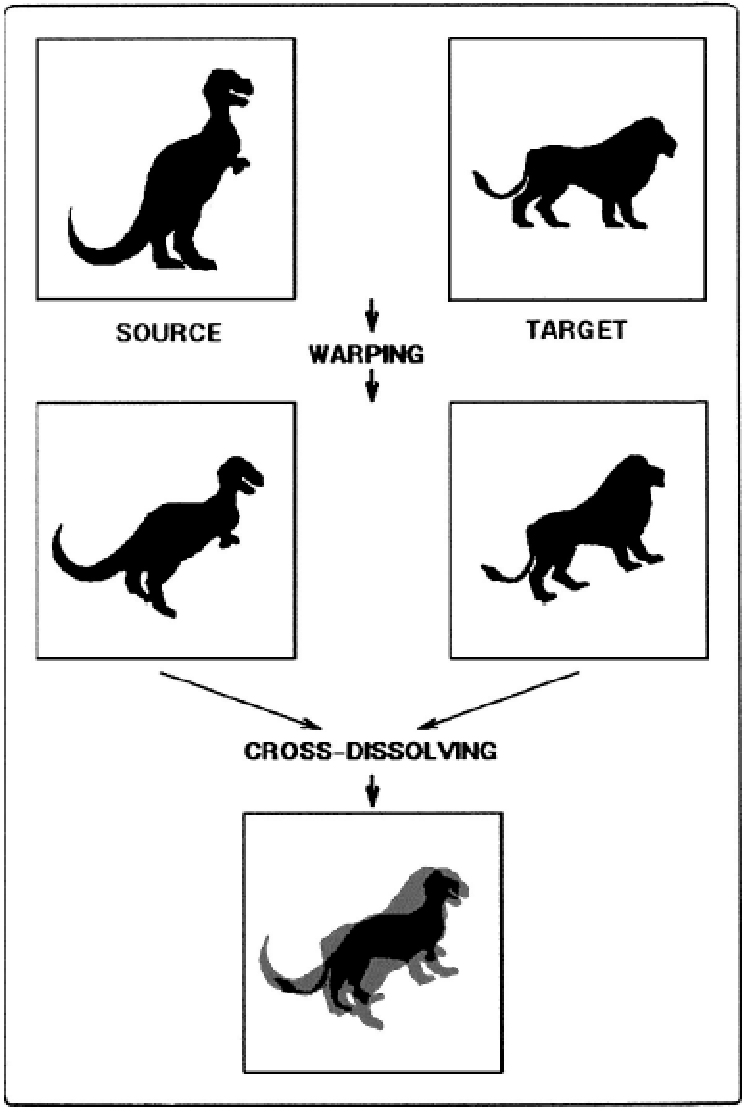
Implicit Approaches

Distance Field

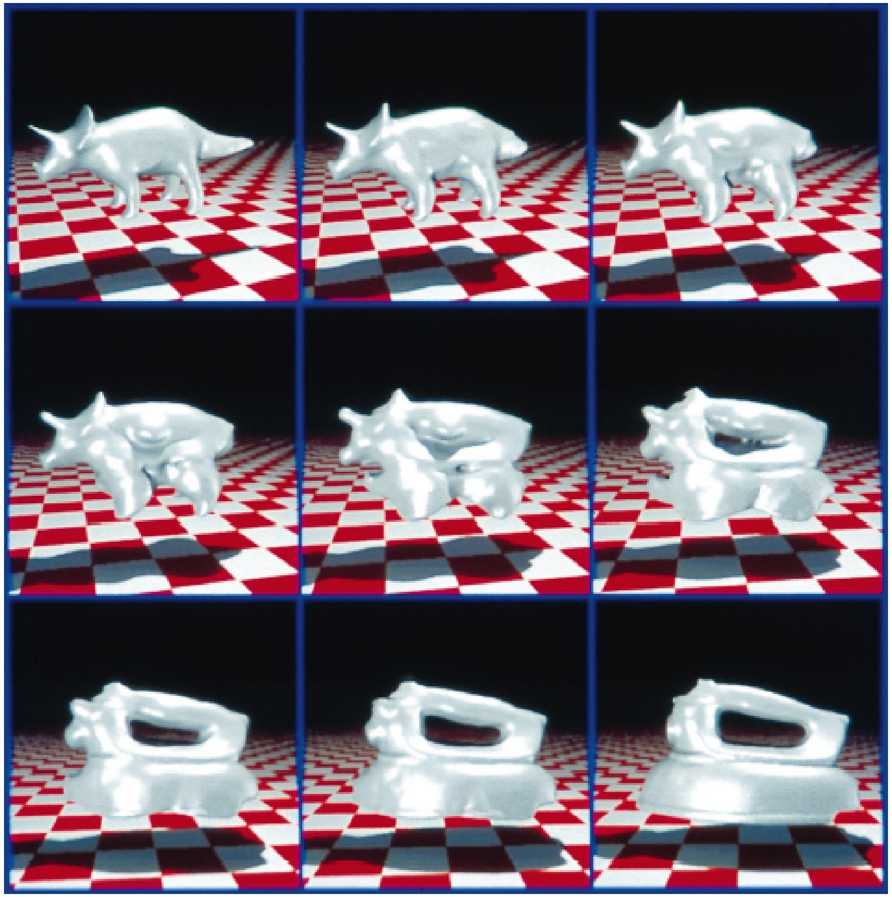
[Cohen-Or et al. 1998]

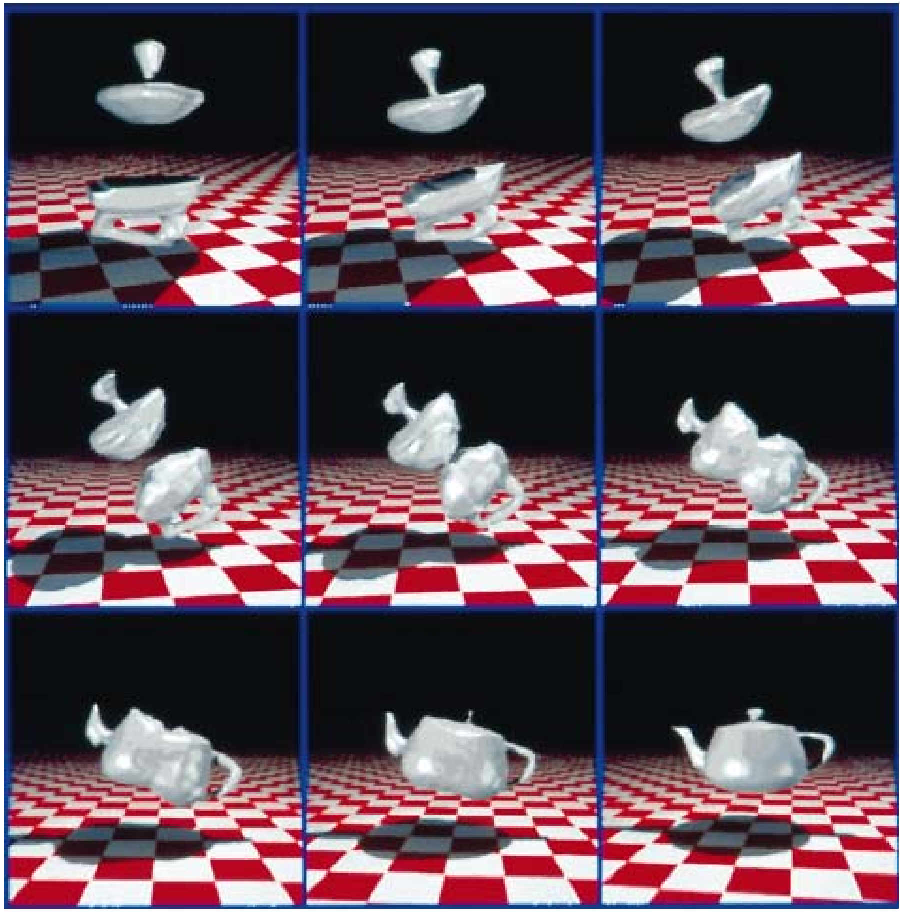
- Distance field of a shape





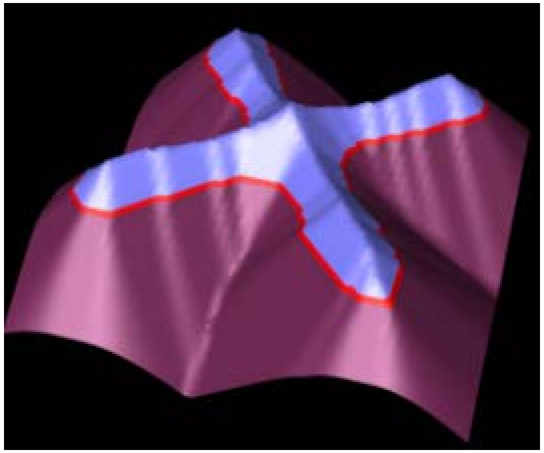
Distance Field

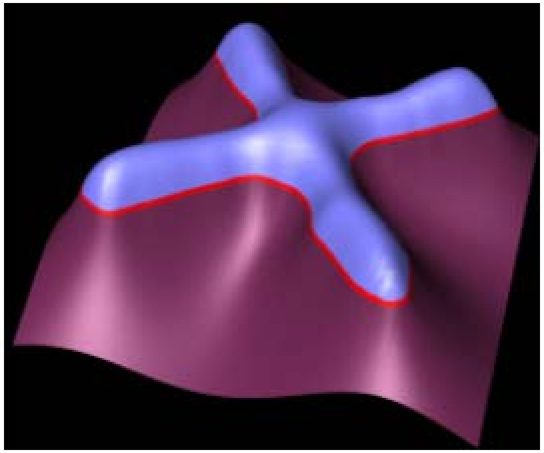


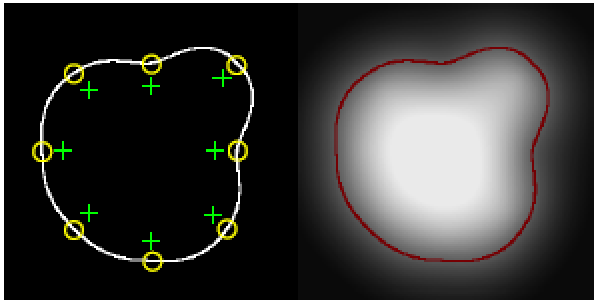


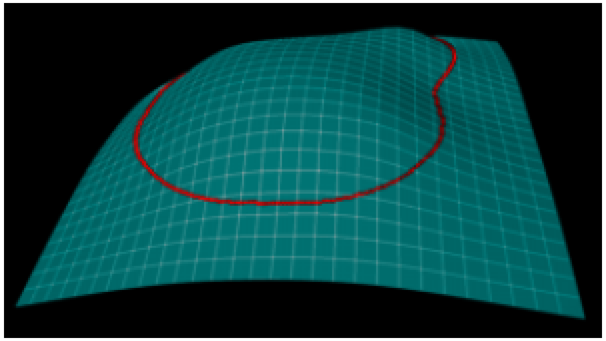
Variational Implicit Function

[Turk et al. 1999]



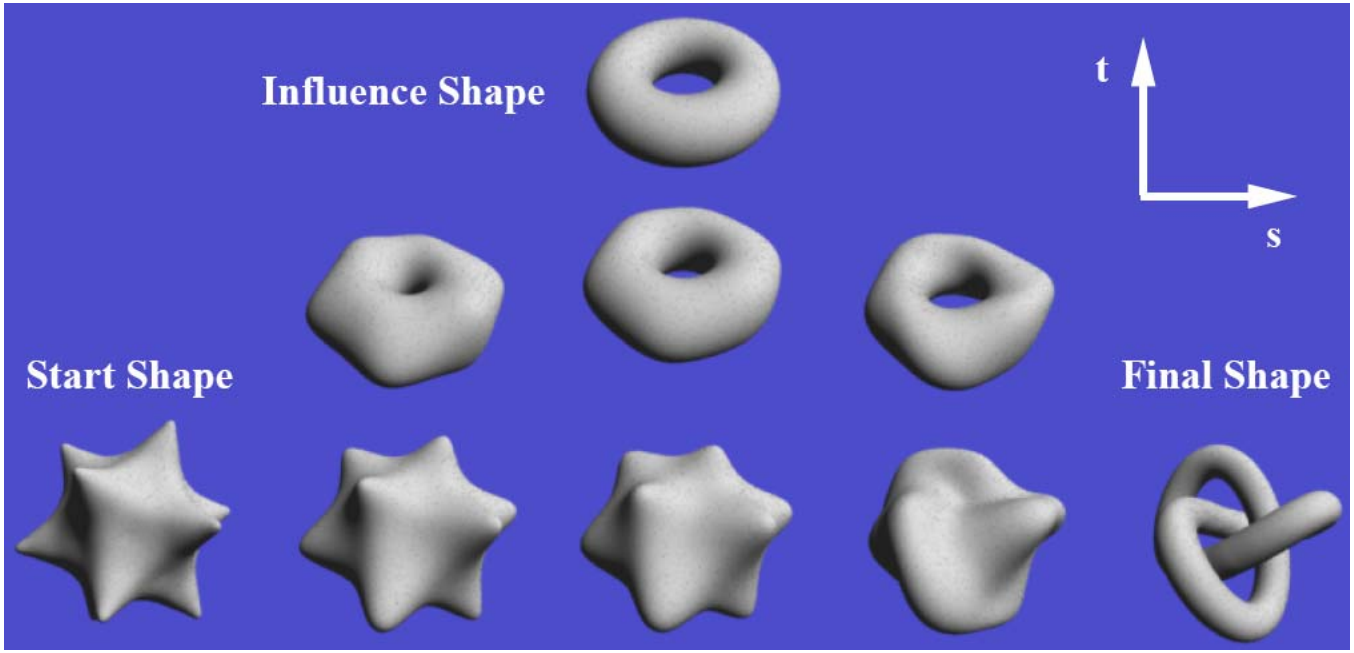


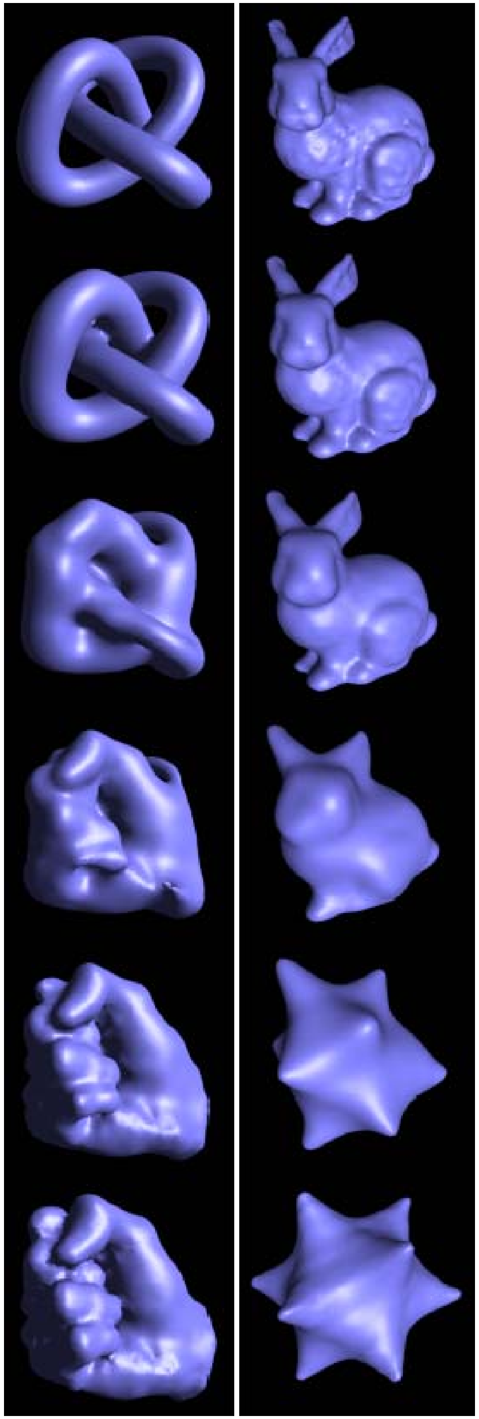




$$f(\mathbf{x}) = \sum_{j=1}^{n} d_j \phi(\mathbf{x} - \mathbf{c}_j) + P(\mathbf{x})$$

Examples





More…

- Level set
- RBF

Q&A