Consistent Mesh Parameterizations

Emil Praun — Princeton

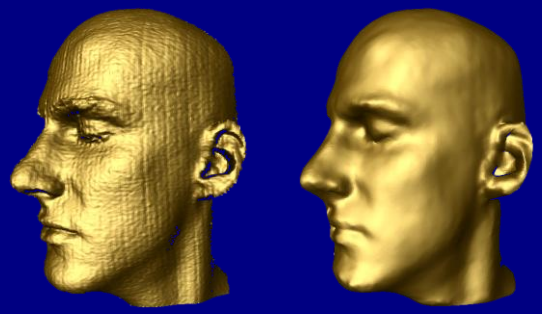
Wim Sweldens — Bell Labs

Peter Schröder — Caltech

Motivation

Digital Geometry Processing (DGP)

- Do for surfaces what DSP does for sound, images, and video



denoising

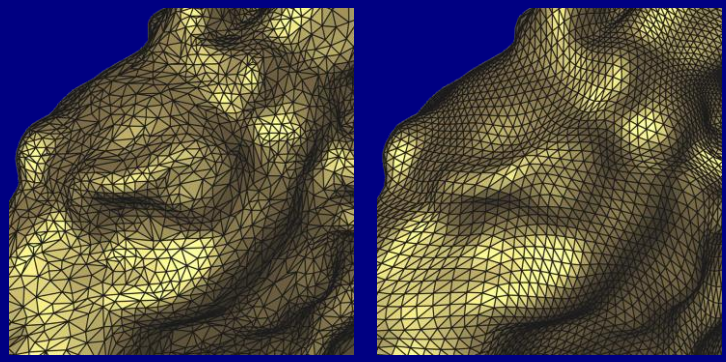
enhancement

- Requires smooth parameterizations

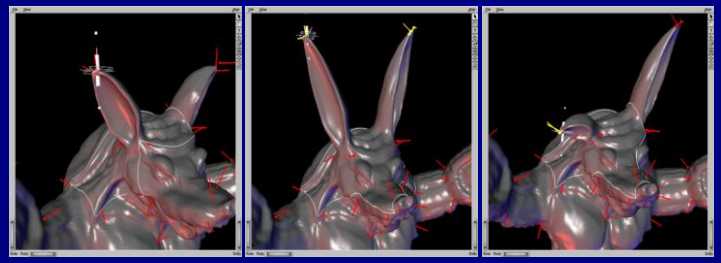
Parameterizations

Smooth sampling pattern

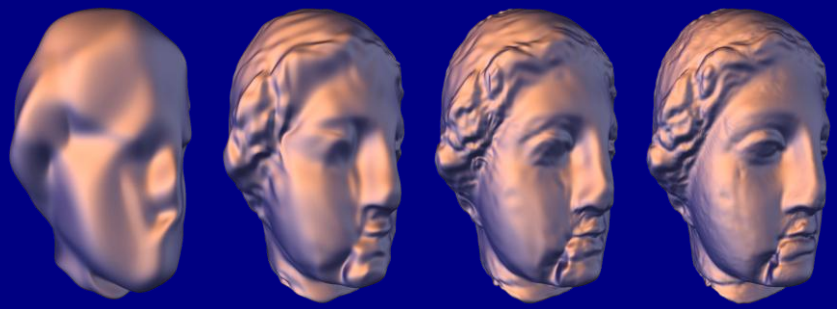
- Individual surface setting
  - *coarse mesh (base domain)*
  - *semi-regular refinement*
- Efficient algorithms



irregular semi-regular



hierarchical editing

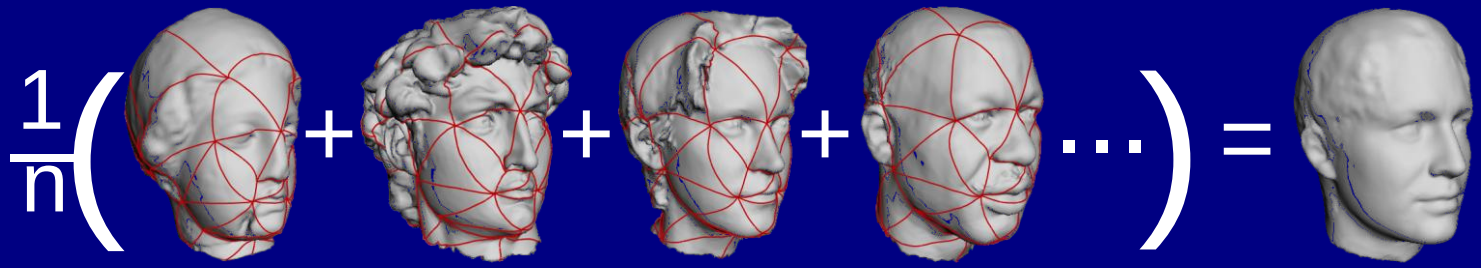


progressive transmission

Parameterizations

What about multiple objects?

- Computing the mean



- ... and many other algorithms
  - *blending, principal components, etc.*

Need consistent parameterizations!

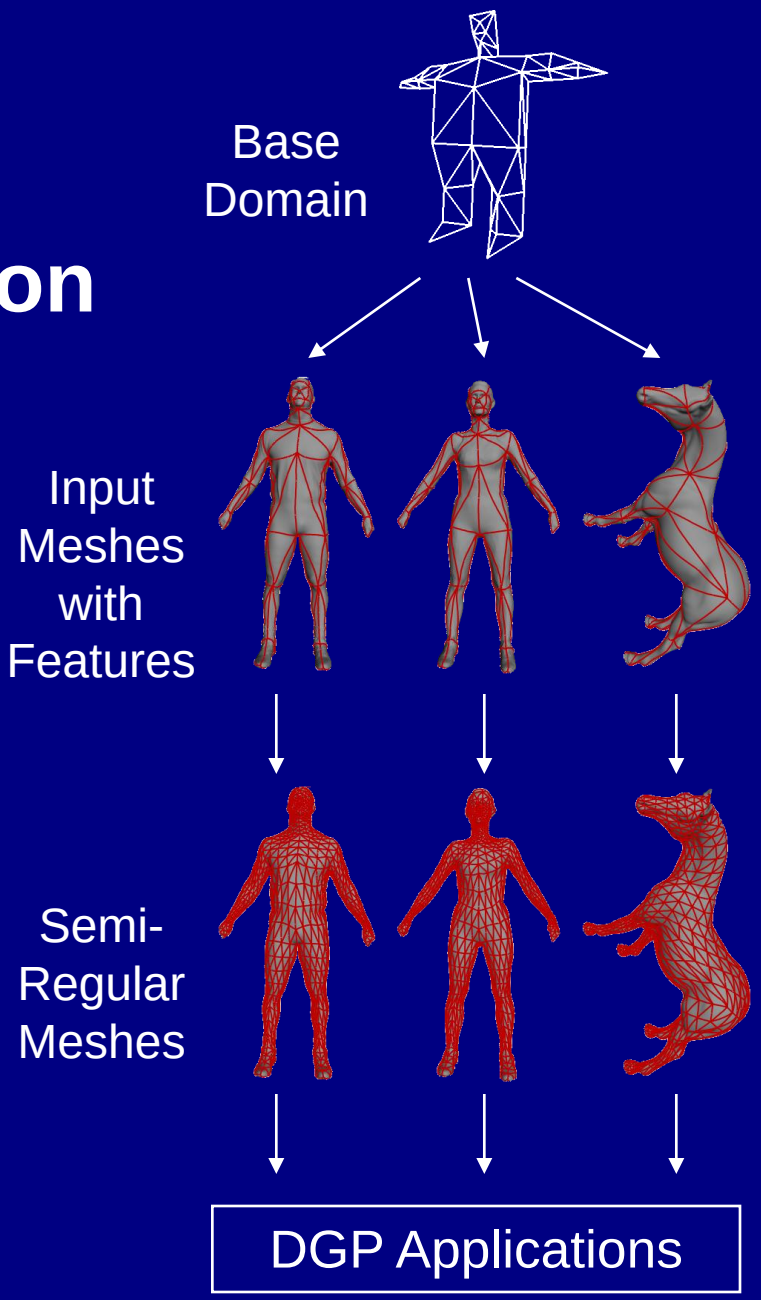
Goal

Consistent parameterization

- same base domain
- correspondences
  - *vertices, edges*
- smooth & fair

Common sampling

- samples 1-1



Previous Work

Mesh Simplification, Progressive Meshes, …

- [Hoppe 94-98]

MAPS, Morphing

- [Lee 98, 99]

Disp. Subdivision Surfaces / Normal Meshes

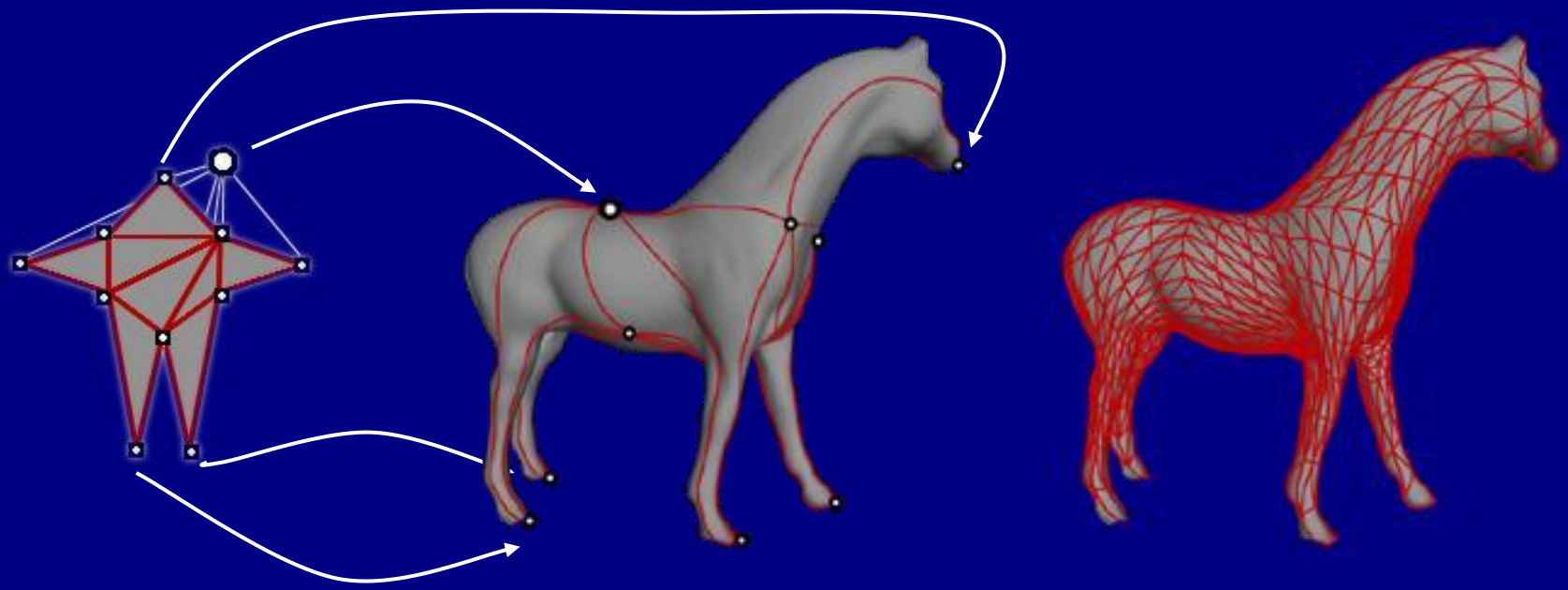
- [Lee 2000] / [Guskov 2000]

Approach

Identify feature points (user)

Trace curves for base domain edges

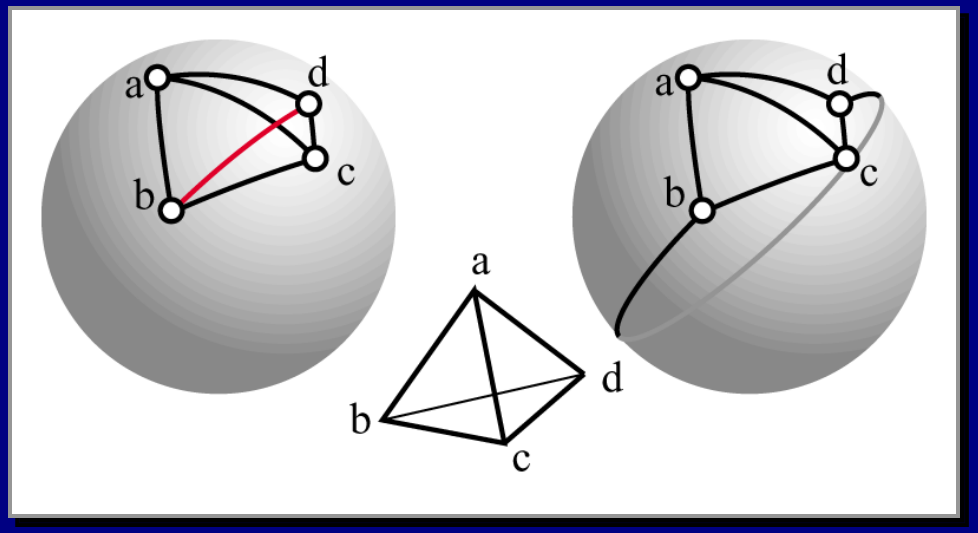
Parameterize interior of patches



Tracing Curves

Net topologically equivalent to base domain

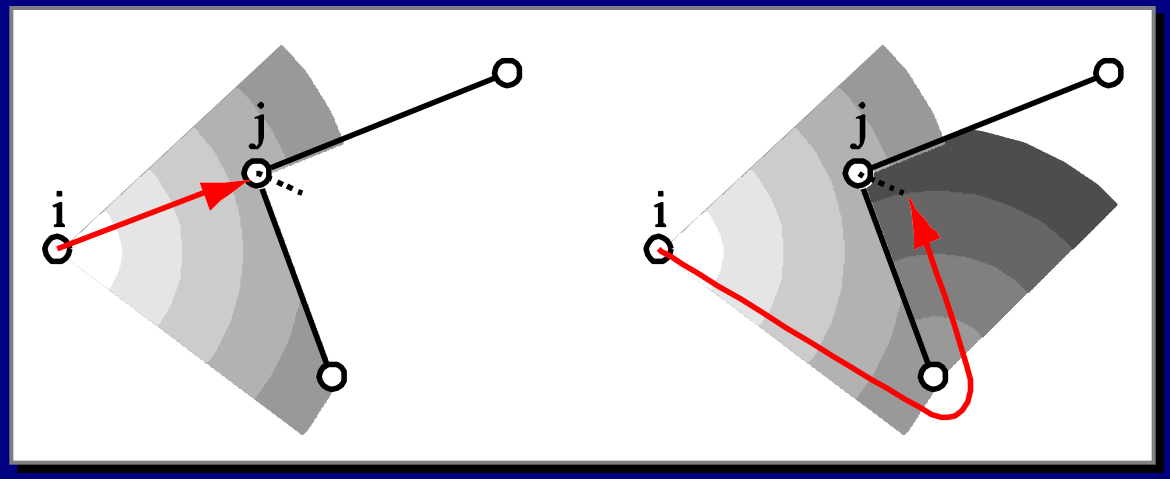
- Curves intersect only at vertices
- Same neighbor ordering around vertices



Tracing Curves

Restricted “brush fire” (BFS traversal):

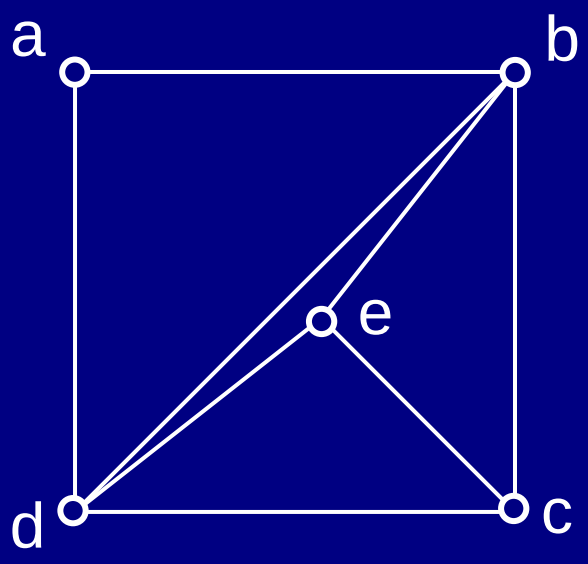
- Do not cross other curves
- Start & end in correct sector



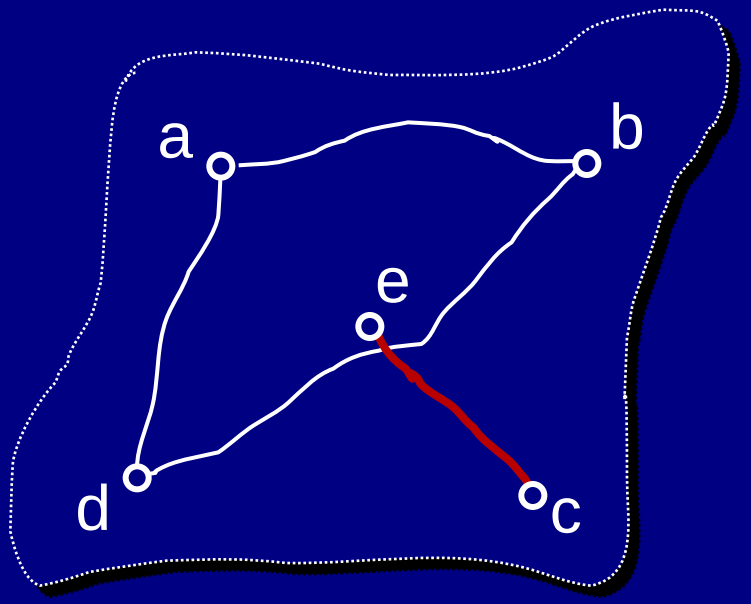
Problem: Encircling

To avoid, first trace spanning tree

Proof of correctness in the paper



Base domain



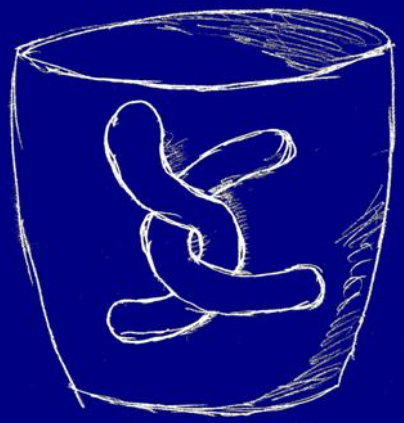
Mesh

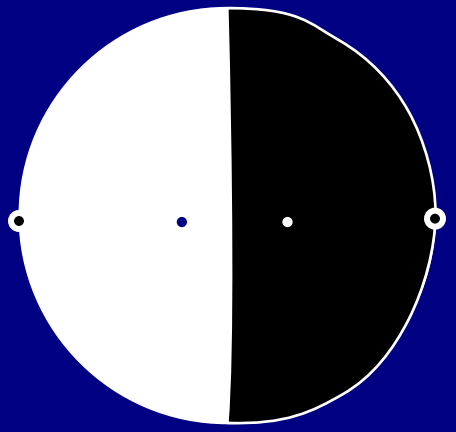
Topological Equivalence

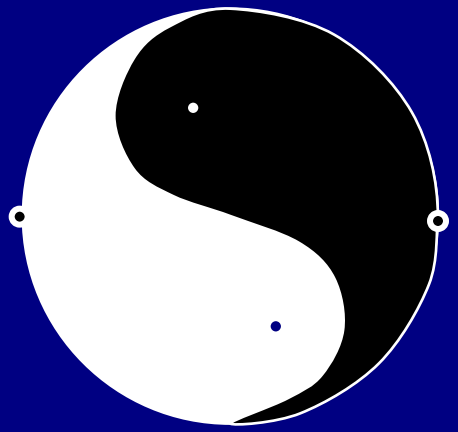
Guarantee topological equivalence of traced net and base domain

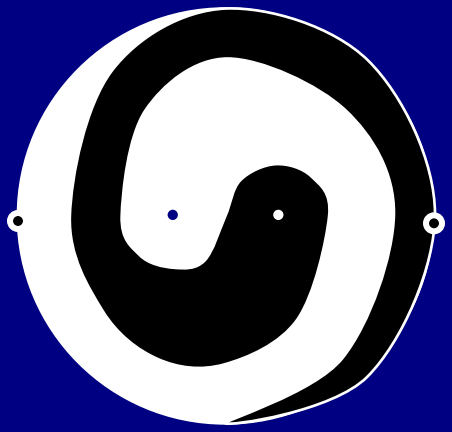
- Trace curves w/ restricted brush fire
- Complete spanning tree before adding cycles

Is “Topological” Enough?









Swirl Operator

Simple relaxation doesn’t undo swirls

Infinity of possible configurations

- We want the least unnecessary swirls
- Optimization very hard; use heuristics

Heuristics

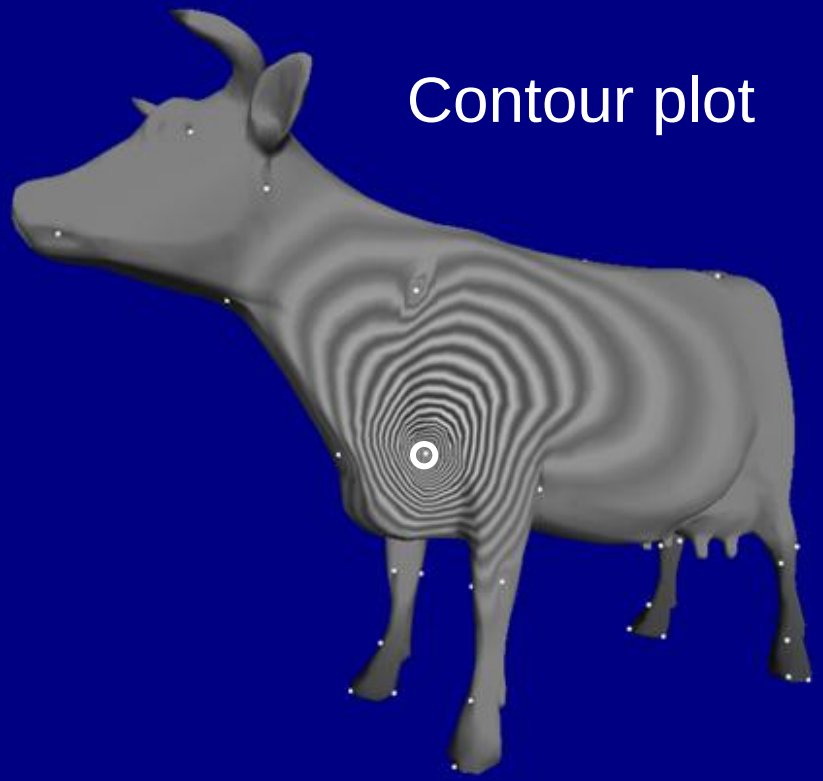
1. Feature points repel curves
2. Introduce curves in order of length
3. Delay edges of flipped triangles

1. Features Repel Curves

Use embedding in $\mathfrak{R}^n$

Compute

- $I_i(k)$ = “influence” of feature i on vertex k



Contour plot

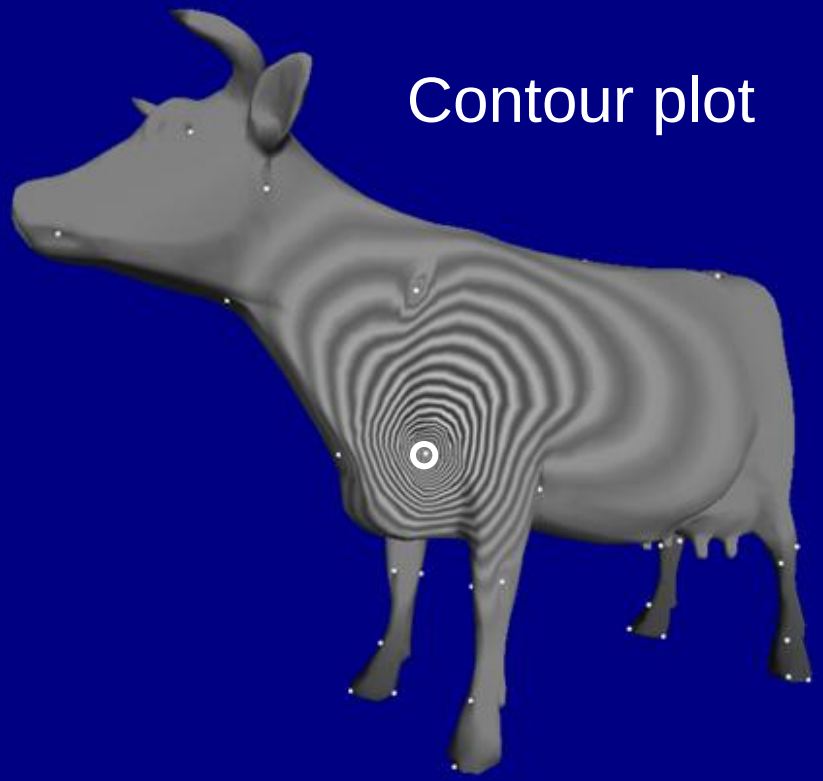
1. Features Repel Curves

Initialize:

- $I_i(i) = 1$
- $I_i(\text{feature } j) = 0$

Relax for mesh surface

- Linear system
- Floater's weights

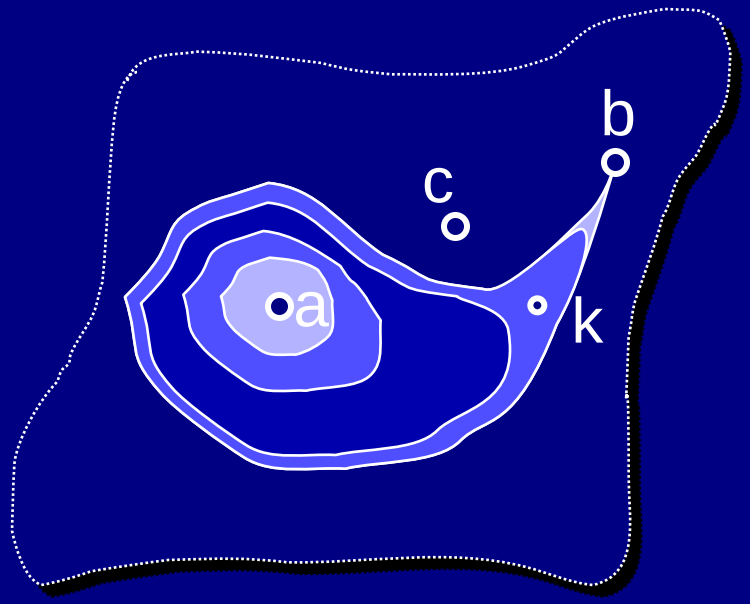


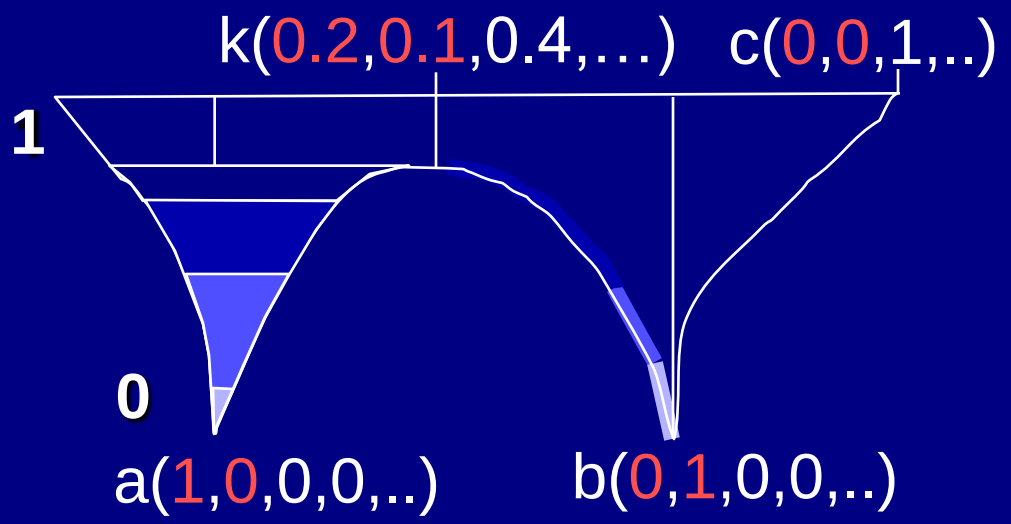
Contour plot

1. Features Repel Curves

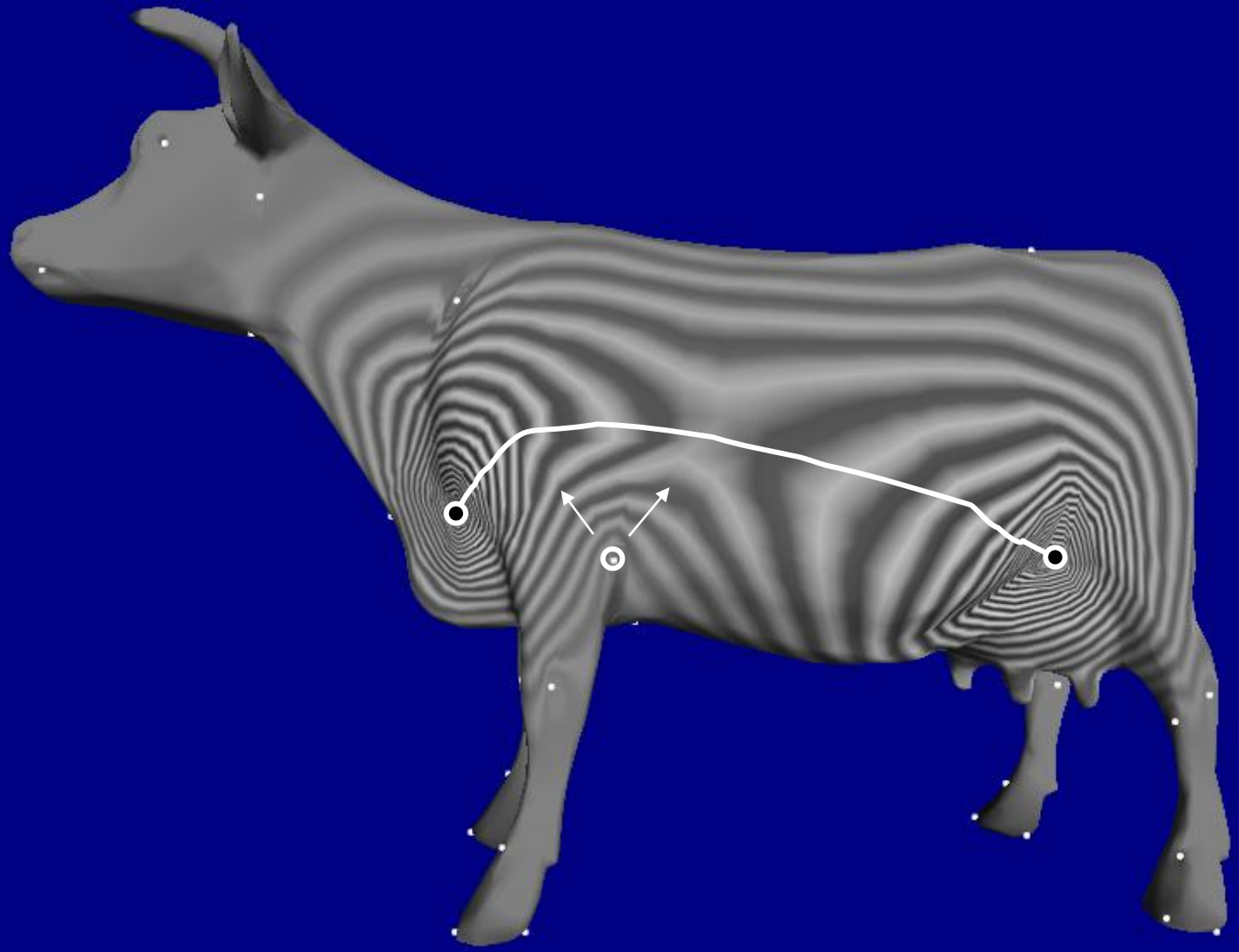
Trace curve (a,b): brush fire with variable propagation speed

Priority $P(k) = 1 - I_a(k) - I_b(k)$





1. Features Repel Curves



2. Prioritize Curves by Length

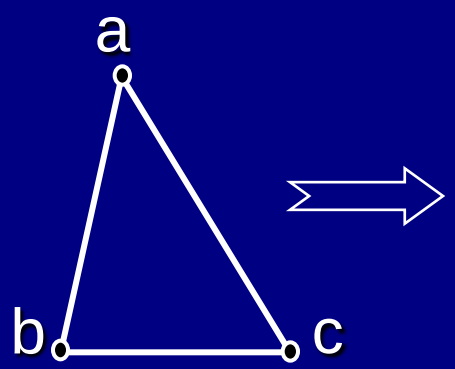
First stage: complete spanning tree

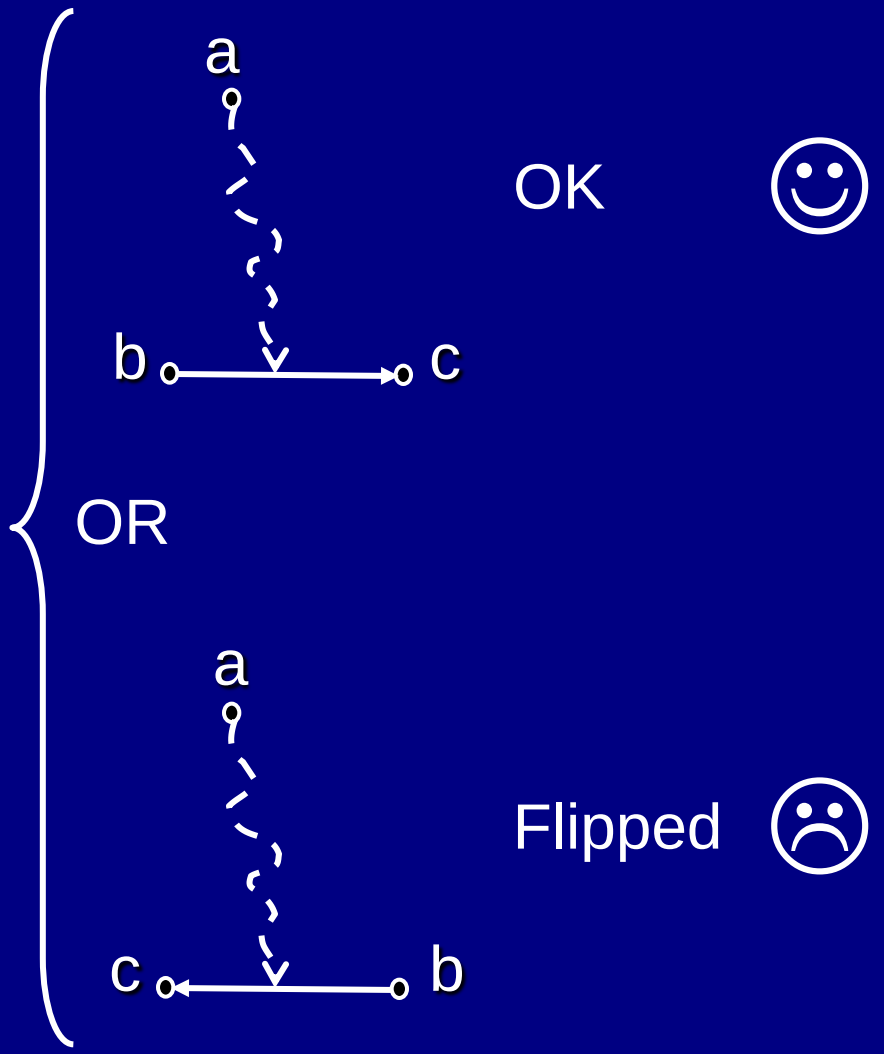
Second stage: complete whole net

For each stage, keep priority queues

- Queues contain candidate curves
- May need to update to enforce topology

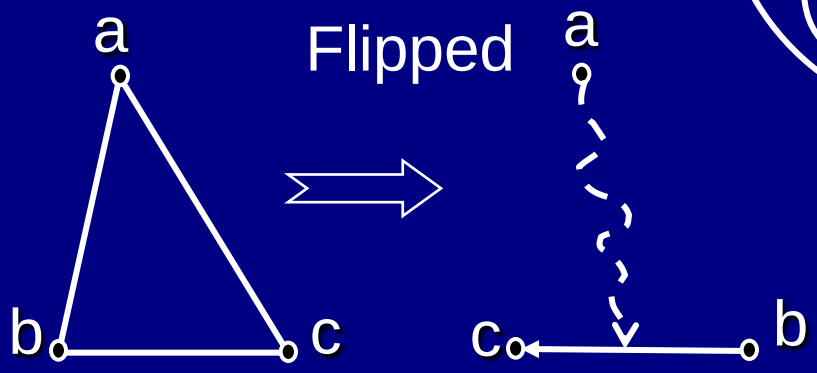
3. Delay Edges of Flipped Triangles

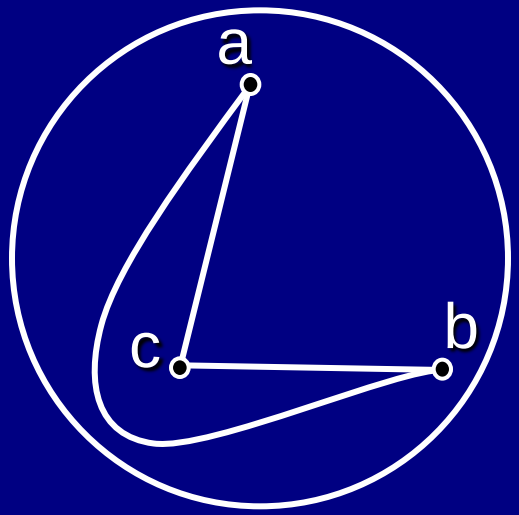




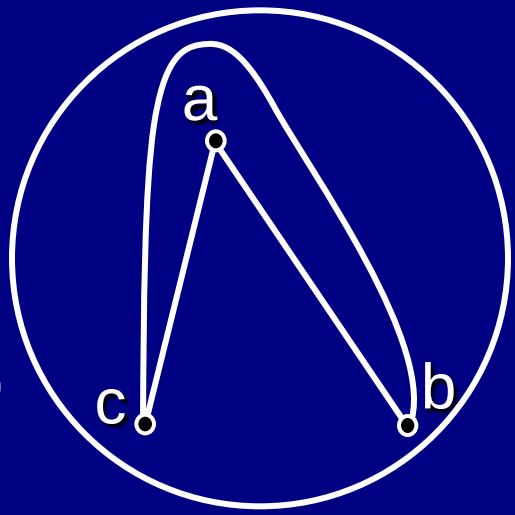
“swirl detector”

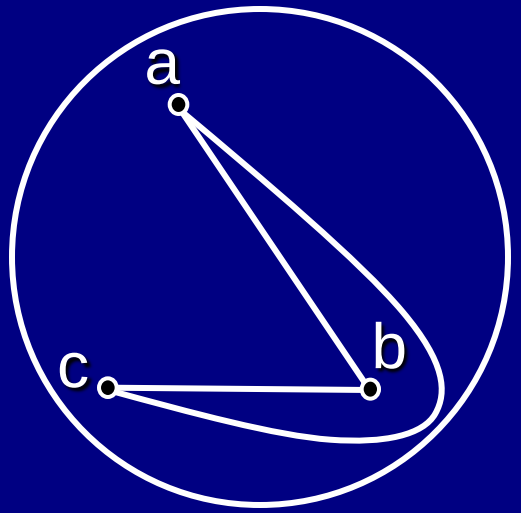
3. Delay Edges of Flipped Triangles





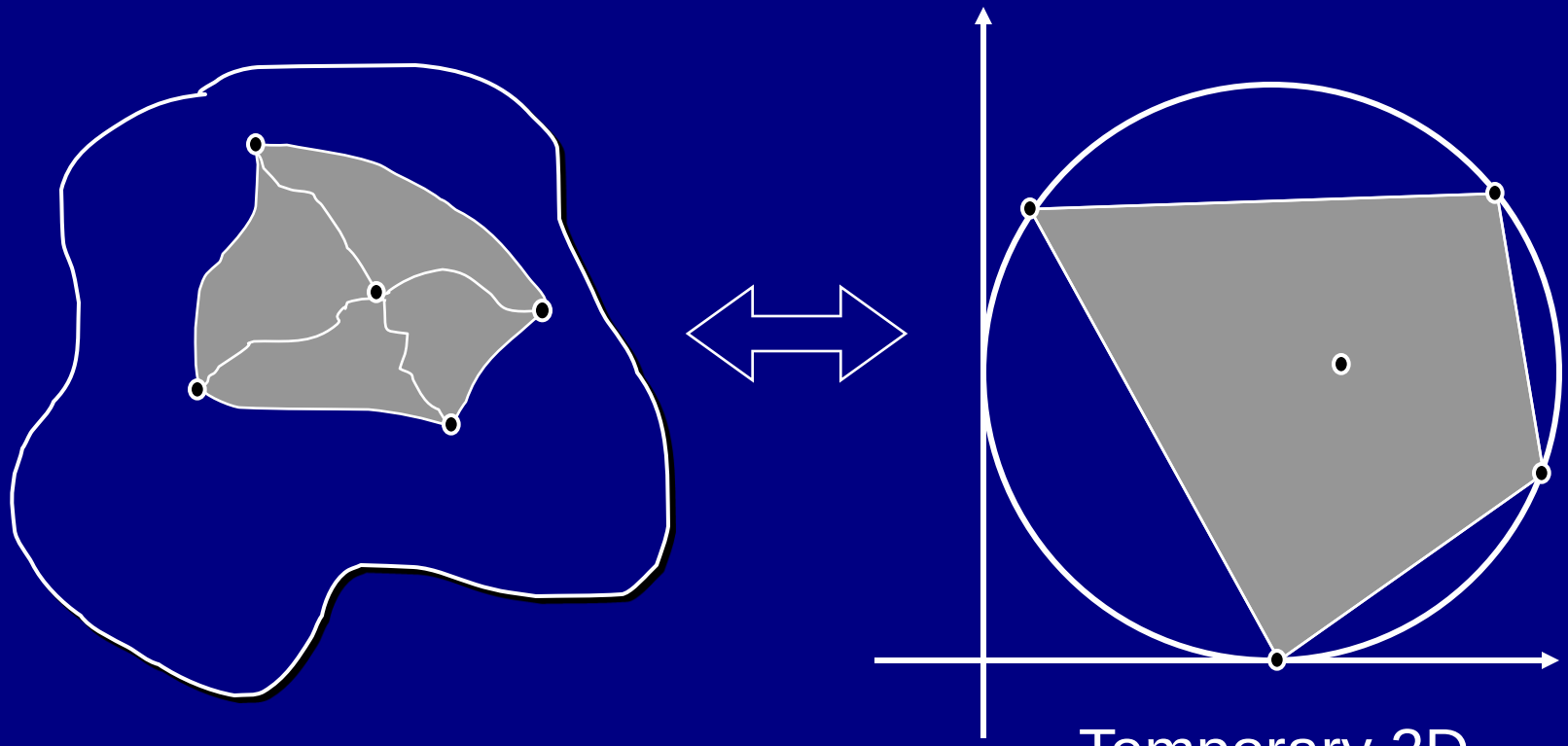
?





“swirl detector”

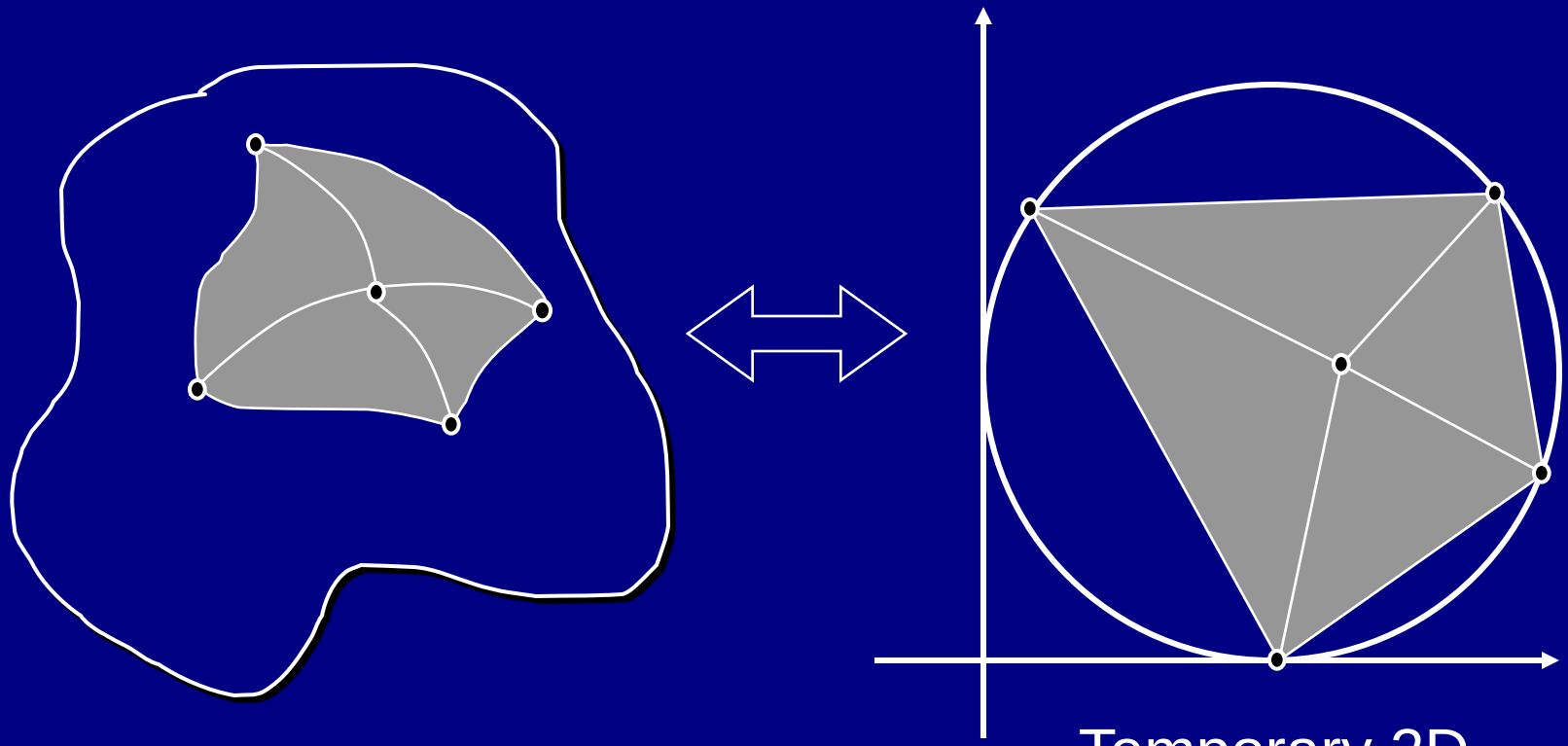
Edge Straightening



Mesh

Temporary 2D parameterization

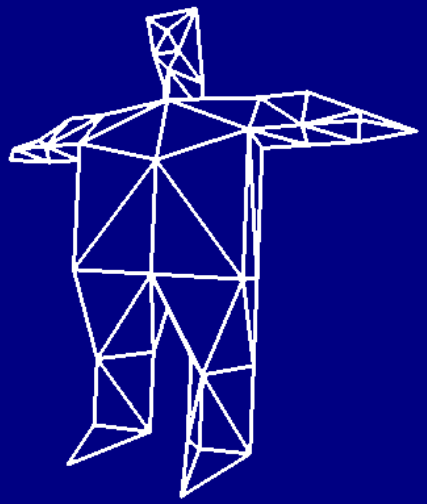
Edge Straightening

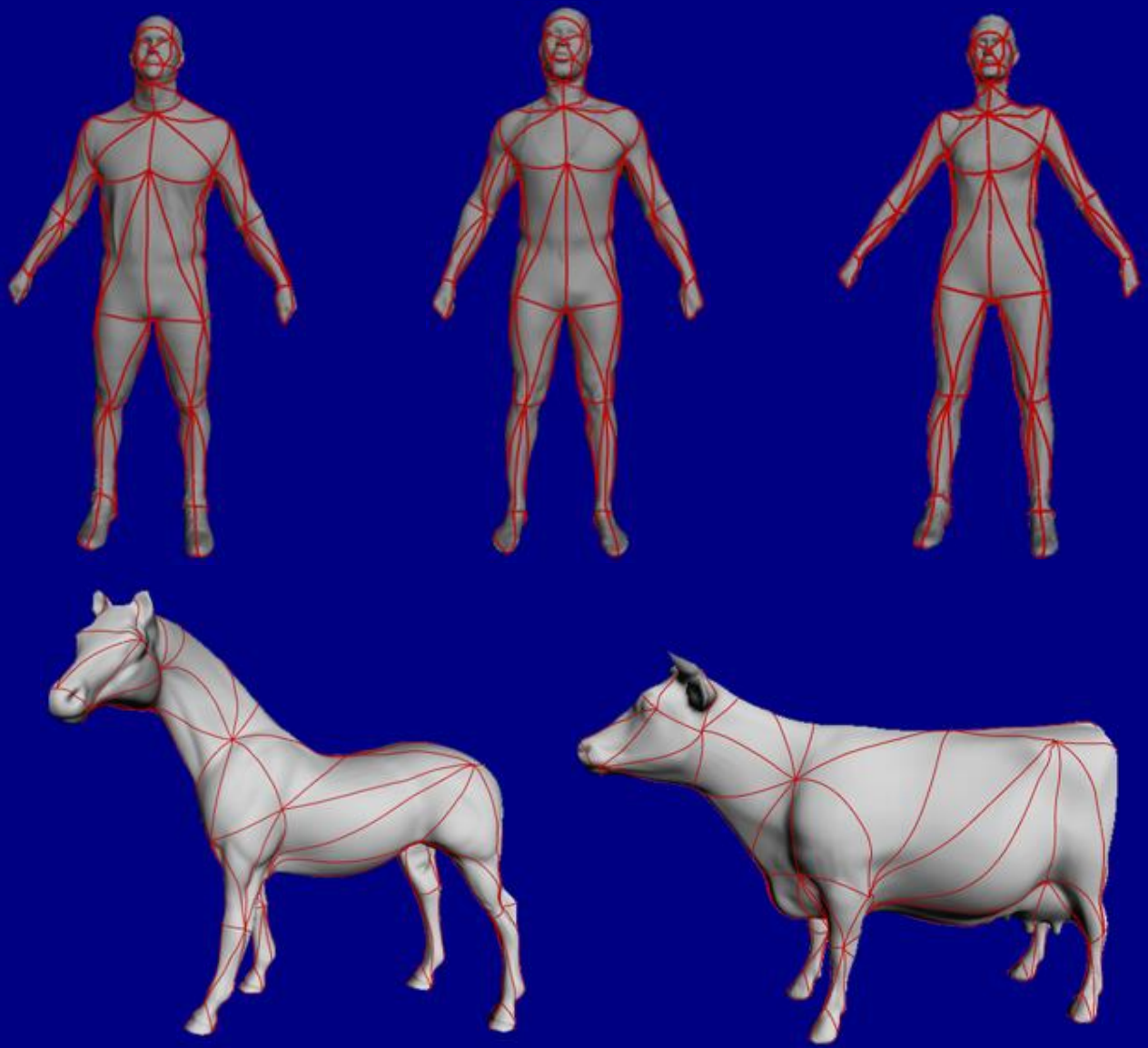


Mesh

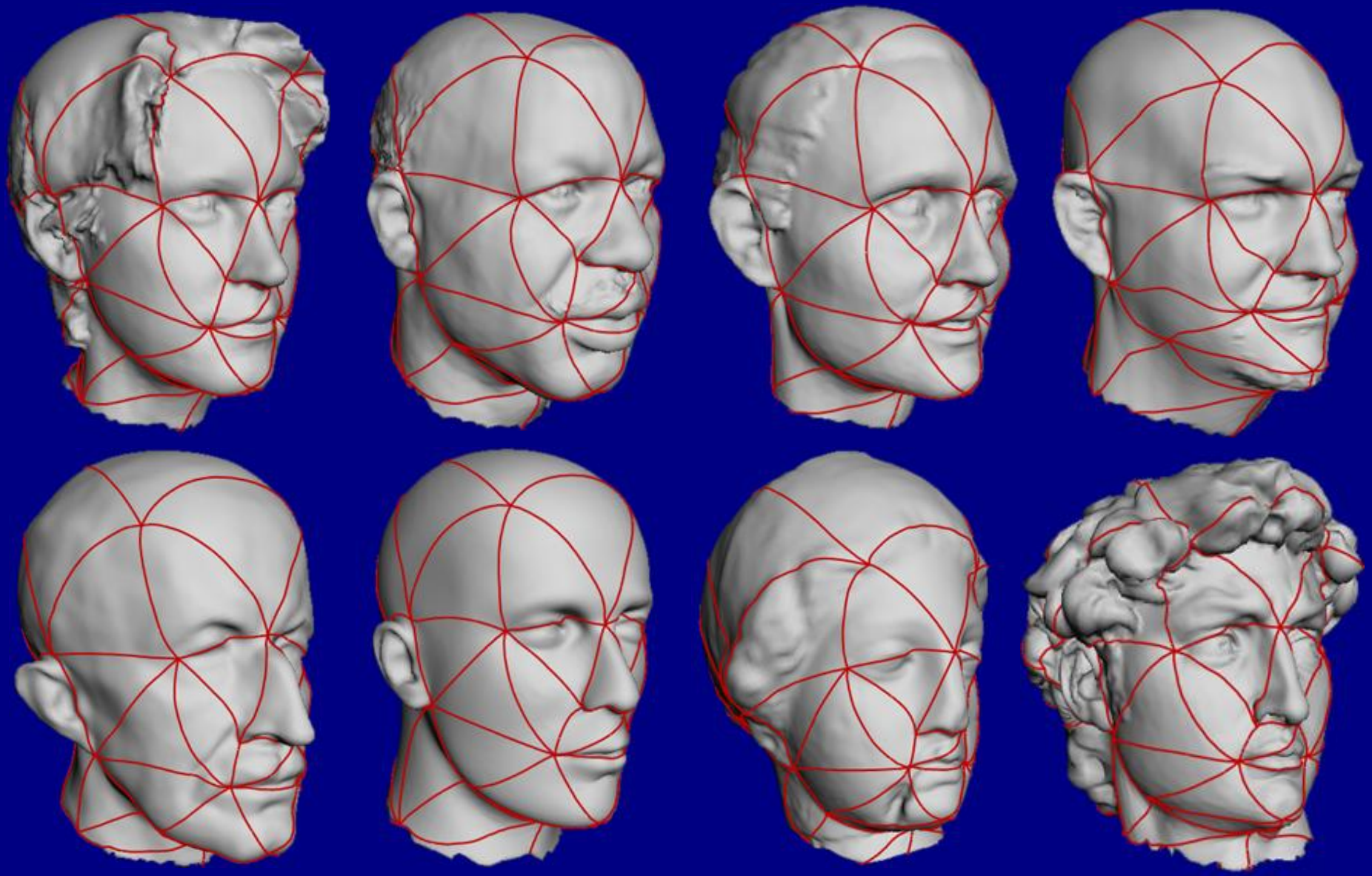
Temporary 2D parameterization

Examples

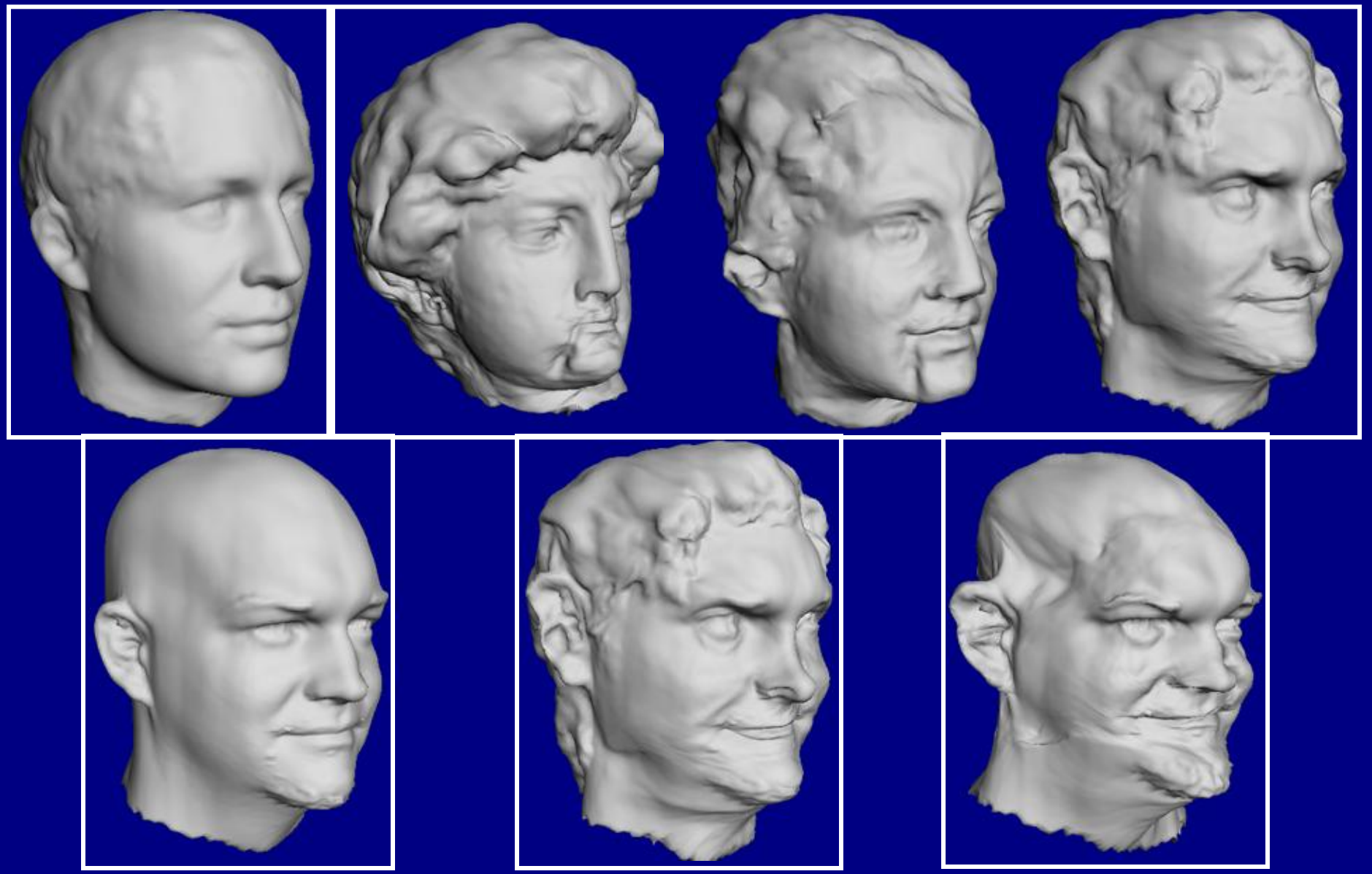




Examples

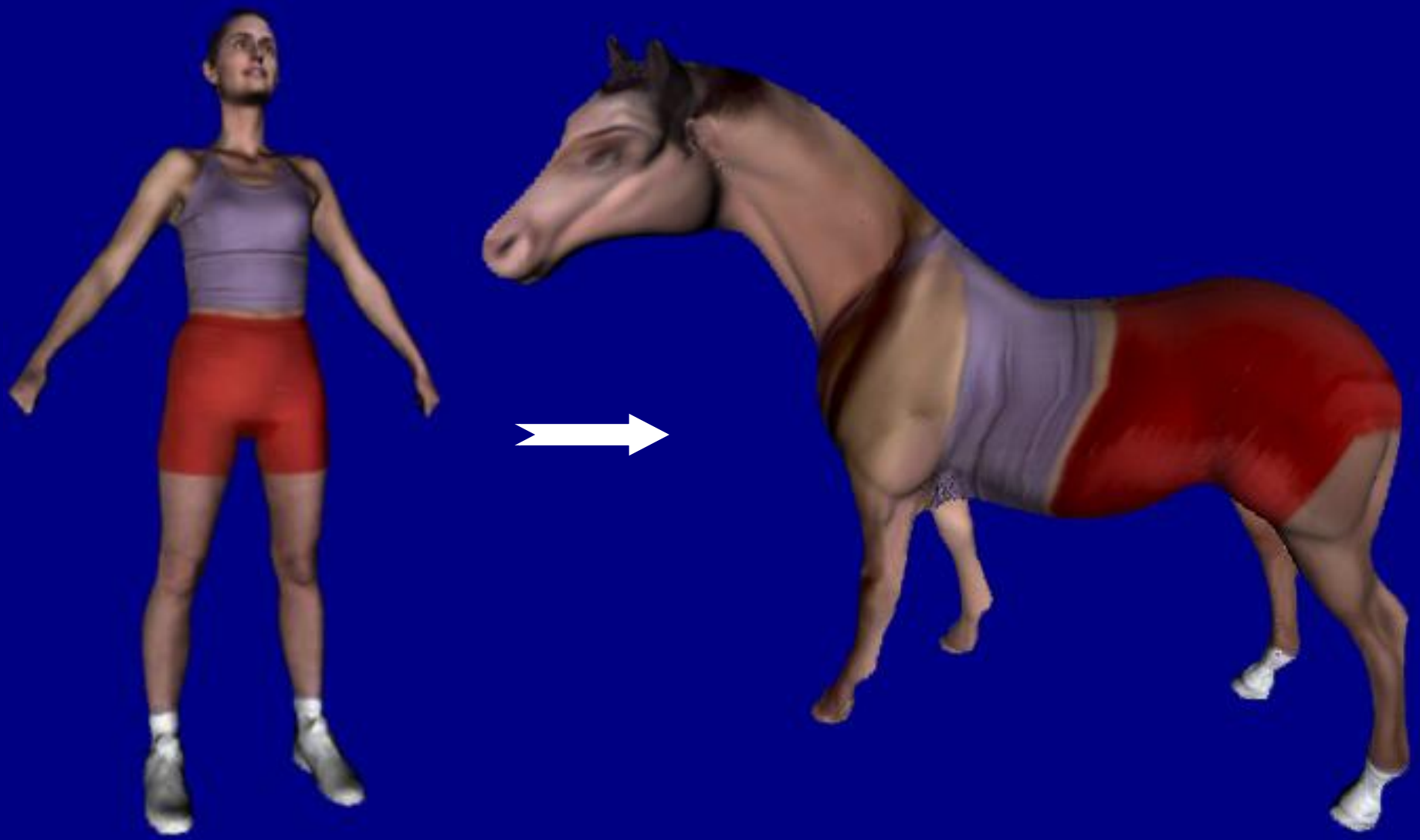


Principal Mesh Components

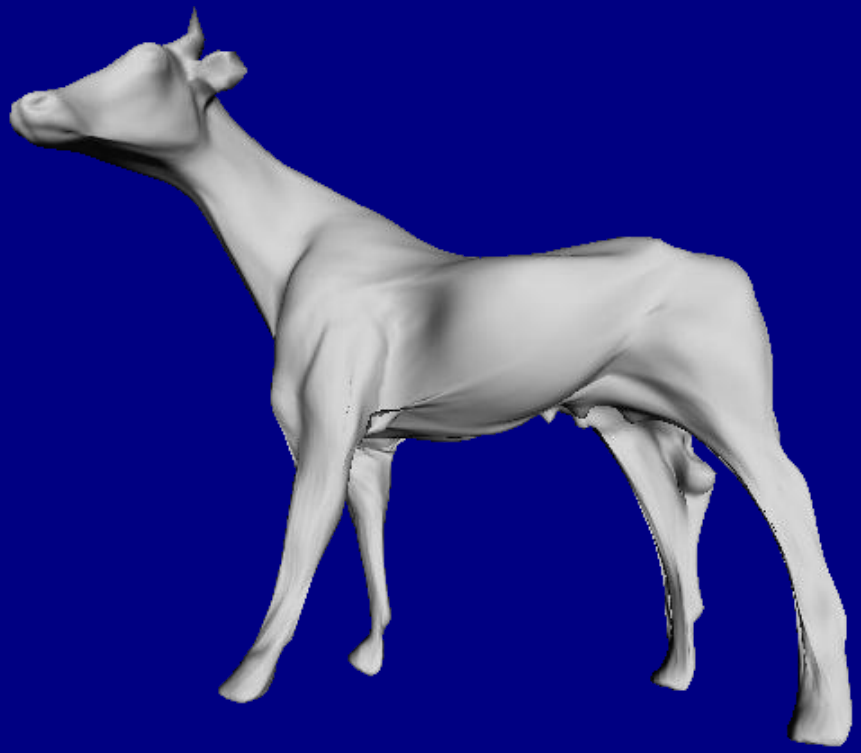


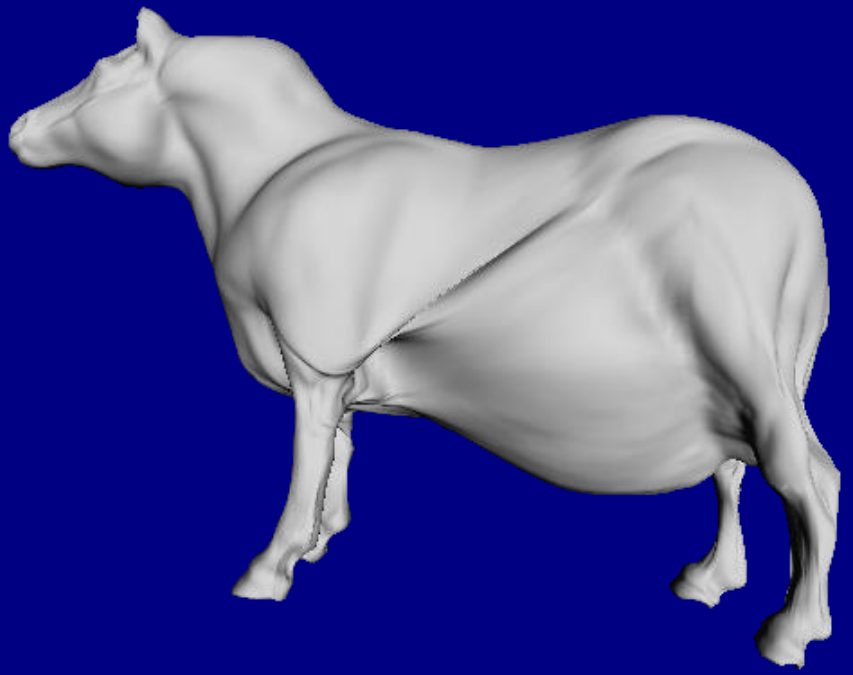
Texture Transfer

Texture Transfer



Detail Transfer



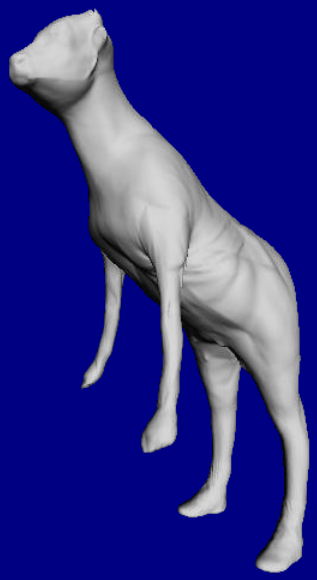


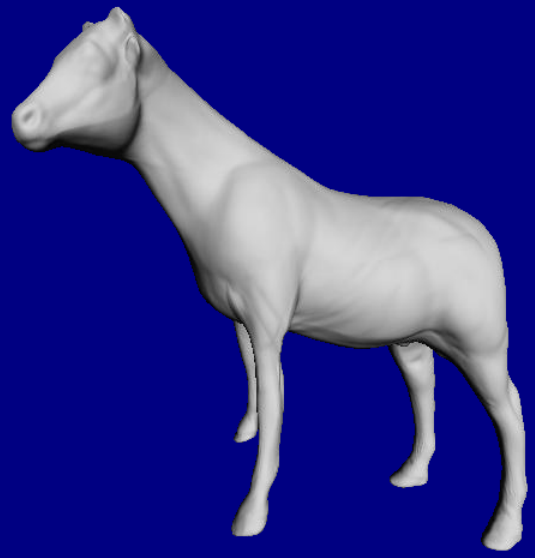
N-way Shape Blending

Horse .33

Man .33

Cow .33





Horse .5

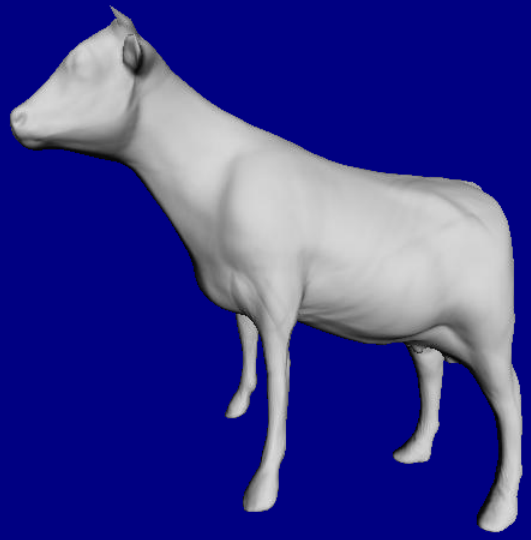
Man .25

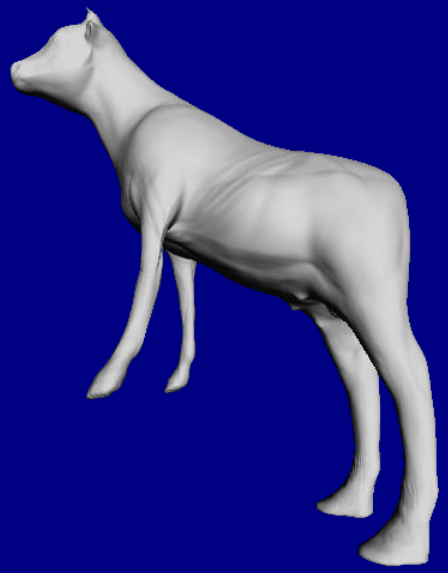
Cow .25

Horse .25

Man .25

Cow .5





Horse .25

Man .5

Cow .25

Future Work

- Higher genus, boundaries, missing feature points, additional feature points.
- Transfer of animation controls
- Use of principal component analysis for indexing and recognition in large database
- Compression of multiple shapes

Acknowledgements

Support: Bell Labs

Models: Cyberware, Stanford, Freiburg U.

Code: Igor Guskov

Help:

Adam Finkelstein,

Tom Funkhouser, Lee Markosian,

& the Princeton crowd