

## Consistent Mesh Parameterizations

Emil Praun    Wim Sweldens    Peter Schröder  
Princeton    Bell Labs    Caltech

## Motivation

### Digital Geometry Processing (DGP)

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

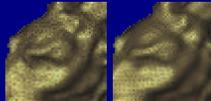


- Requires smooth parameterizations

## Parameterizations

### Smooth sampling pattern

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



- Efficient algorithms



hierarchical editing



progressive transmission

## Parameterizations

### What about multiple objects?

- Computing the mean

$$\frac{1}{n} (\text{head}_1 + \text{head}_2 + \text{head}_3 + \dots) = \text{mean\_head}$$

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

**Need consistent parameterizations!**

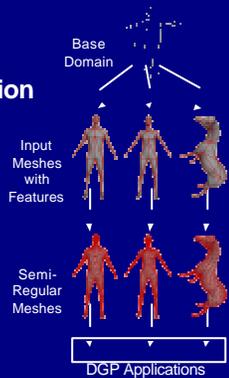
## Goal

### 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

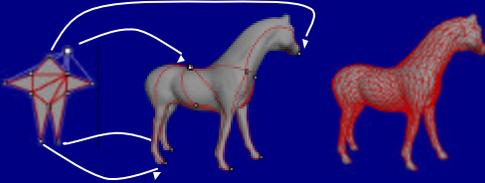
- [Lee 2000] / [Guskov 2000]

## Approach

Identify feature points (user)

Trace curves for base domain edges

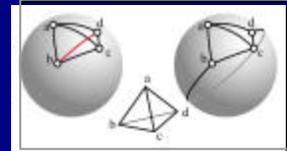
Parameterize interior of patches



## Tracing Curves

Net topologically equivalent to base domain

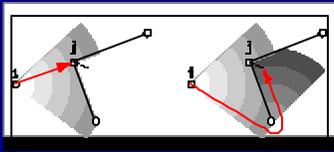
- Curves intersect only at vertices
- Same neighbor ordering around vertices



## Tracing Curves

Restricted “brush fire” (BFS traversal):

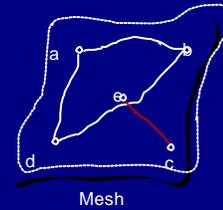
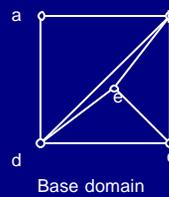
- Do not cross other curves
- Start & end in correct sector



## Problem: Encircling

To avoid, first trace spanning tree

Proof of correctness in the paper

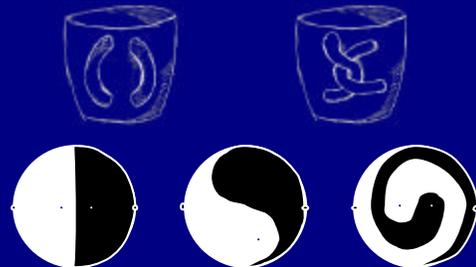


## Topological Equivalence

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

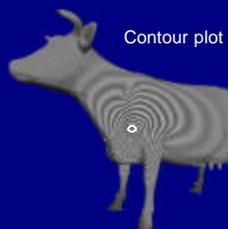
1. Feature points repel curves
2. Introduce curves in order of length
3. Delay edges of flipped triangles

## 1. Features Repel Curves

Use embedding in  $\hat{A}^n$

Compute

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



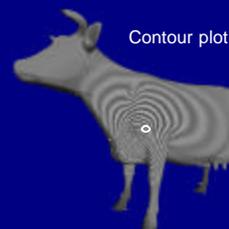
## 1. Features Repel Curves

Initialize:

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

Relax for mesh surface

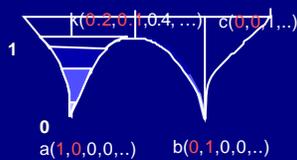
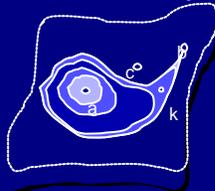
- Linear system
- Floater's weights



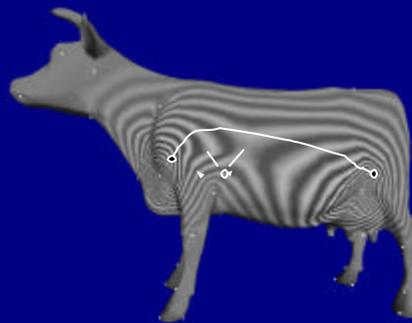
## 1. Features Repel Curves

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

$$\text{Priority } P(k) = 1 - I_a(k) - I_b(k)$$



## 1. Features Repel Curves



## 2. Prioritize Curves by Length

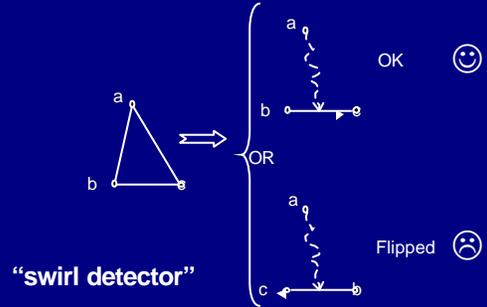
First stage: complete spanning tree

Second stage: complete whole net

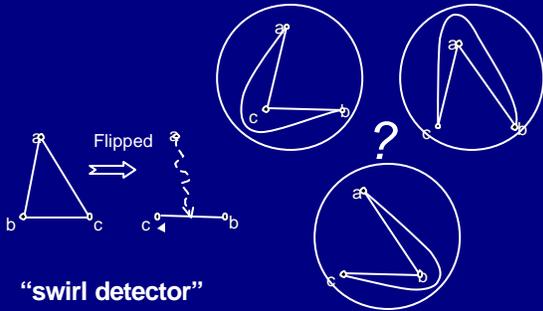
For each stage, keep priority queues

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

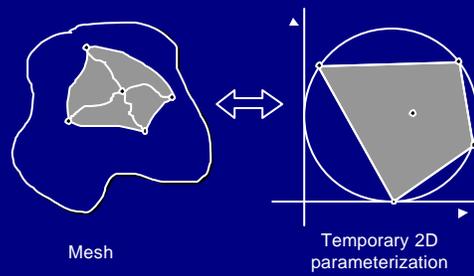
## 3. Delay Edges of Flipped Triangles



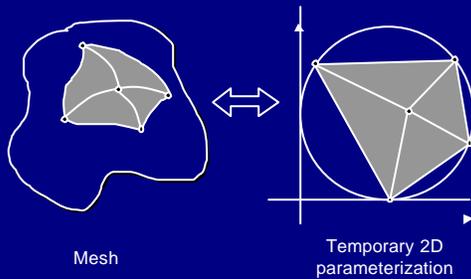
## 3. Delay Edges of Flipped Triangles



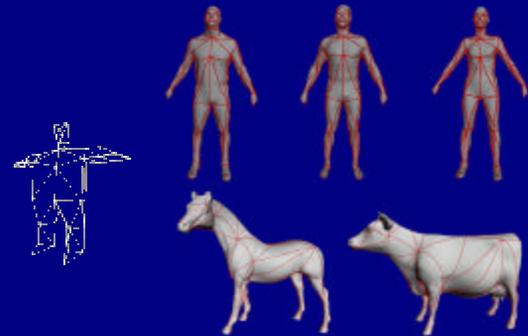
## Edge Straightening

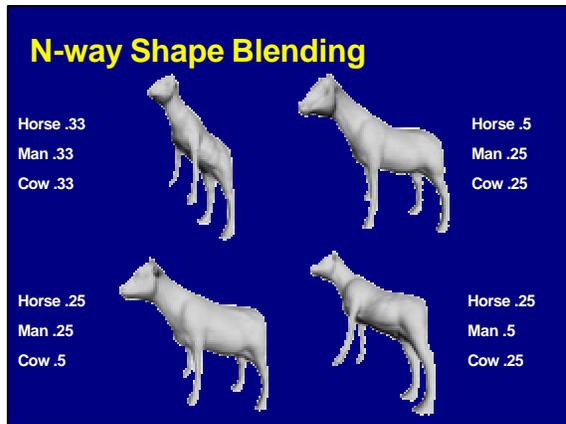
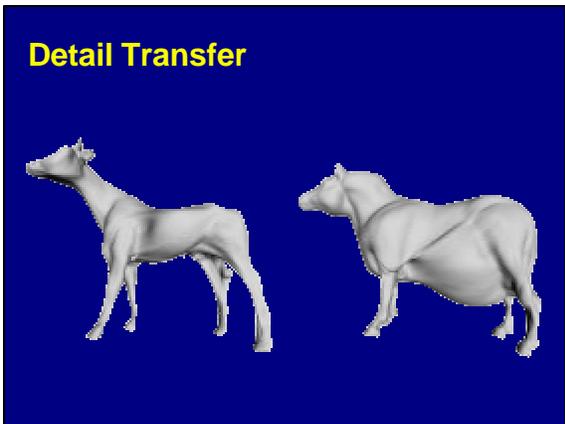
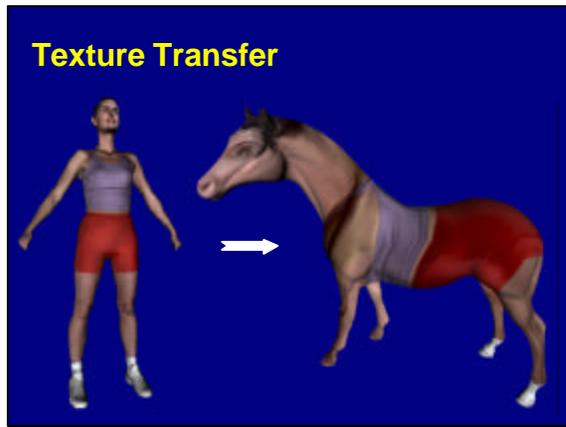
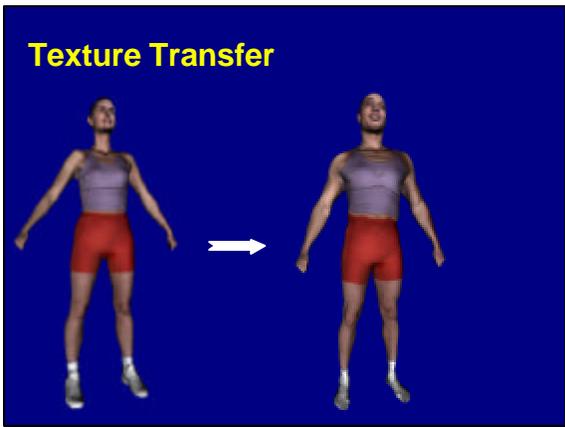
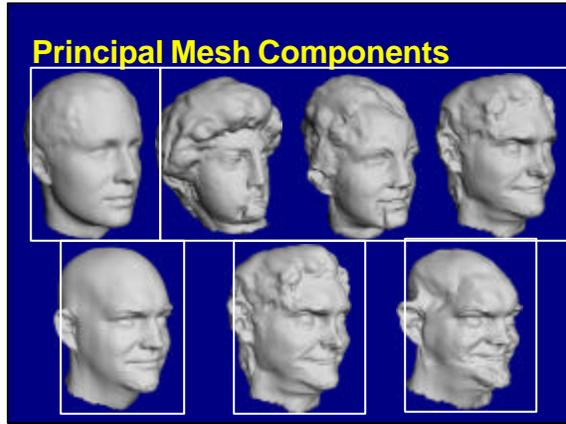
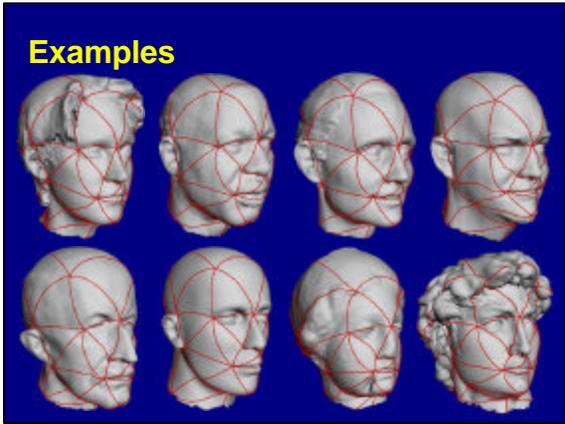


## Edge Straightening



## Examples





## 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,

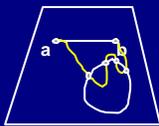
Tom Funkhouser, Lee Markosian,

& the Princeton crowd

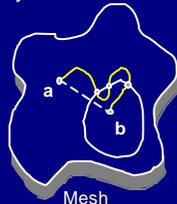
## Proof

Spanning tree prevents encircling

- Before tree completion, no cycles



Base domain

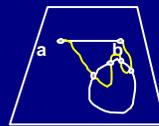


Mesh

## Proof

Spanning tree prevents encircling

- Before tree completion, no cycles



Base domain



Mesh