MAPS: Multiresolution Adaptive Parameterization of Surface

Aaron W. F. Lee   Wim Sweldens   Peter Schröder

Lawrence Cowsar   David Dobkin

Introduction

- Construct hierarchy of models of different fineness in $O(N \log N)$ time and space complexity
- Construct smooth parameterization of the original mesh
- Allows for uniform or adaptive remeshing after simplification
- Respect either manual tagging of features or automatic feature detection

Outline

- Notation
- Constructing the Mesh Hierarchies
- Parameterization
- Preserving Image Features
- Remeshing
- Result

Notation (1/4)

- Triangular Mesh
  - represented as a pair (P,K)
  - P is the set of all N points $pi = (xi,yi,zi)$
  - K is the complex defining connectivity by 3 types of simplicies
    - $\{i\}$is the vertex pi
    - $\{i,j\}$is the edge from $\{i\}$ to $\{j\}$
    - $\{i,j,k\}$is the triangle with vertices $\{i\}$, $\{j\}$,$\{k\}$

Notation (2/4)

- $\{k\}$ and $\{j\}$ are Neighbors if there is an edge between them
- An Independent Set of vertices is a set where no vertices are neighbors
- A set is Maximally Independent if no superset of it is independent

Notation (3/4)

- 1-ring neighborhood

$$N(i) = \{j \mid \{i,j\} \in K\}.$$

  - i.e. all vertices adjacent to {*i*}
- star(*i*)

$$\operatorname{star}(i) = \bigcup_{i \in s,\, s \in K} s.$$

  - i.e. all vertices, edges, and faces that include {*i*}, including {*i*} itself

Notation (4/4)

- $k(i)$, curvature estimate
  $k(i) = k1 + k2$
- Estimated by finding a tangent plane to $\{i\}$ and then using a polynomial to approximate star(i)and then finding curvature of the polynomial
- Claims to be conservative curvature estimate

Constructing the Mesh Hierarchies(1/5)

- Notation: Meshes of level l denoted (P^l, K^l)
  - Original (finest) mesh: (P^L, K^L)
  - Coarsest mesh (base domain):? (P^0, K^0)
- Traditionally progressive meshing involved *edge collapse* MAPS uses *vertex removal* and then retriangulates the resulting hole
- Similar to hierarchy given by Dobkin & Kirkpatrick (DK) which ensures an $O(\log N)$ bound on the number of levels

Constructing the Mesh Hierarchies(2/5)

- MAPS Approach:
  - Make a priority queue with a weight *w*(l,*i*) for each vertex {*i*}
  - *a*(*i*) is the area of the 1-ring neighborhood
  - k(*i*) is the curvature of the 1-ring neighborhood
  - They suggest l=0.5
  - Smallest weights get removed first

$$w(\lambda, i) = \lambda \frac{a(i)}{\max_{p_i \in P^l} a(i)} + (1-\lambda) \frac{\kappa(i)}{\max_{p_i \in P^l} \kappa(i)}.$$

Constructing the Mesh Hierarchies(3/5)

- After weights are computed for all vertices, mark the one with smallest weight
- Continue looking at vertices in order of increasing weight and mark the ones that are independent of all other marked vertices
- Remove marked vertices and all adjacent edges

Constructing the Mesh Hierarchies(4/5)

- Use conformal map, *ui*

  Enumerate cyclically the *Ki* vertices in the 1-ring

$$N(i) = \{j_k \mid 1 \leq k \leq K_i\}$$

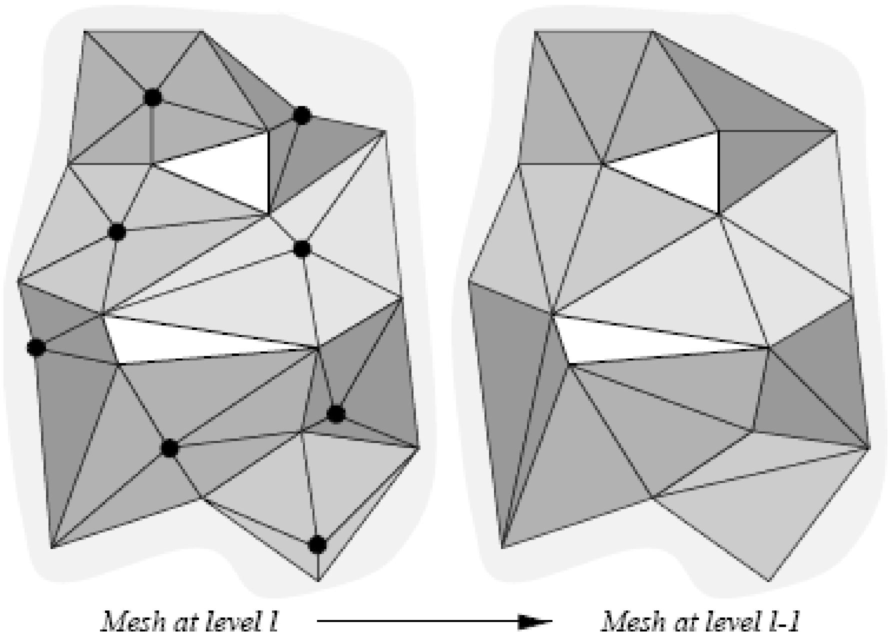
$\mu_i(p_i) = 0$ and $\mu_i(p_{jk}) = r_k^{\,a} e^{i\theta ka}$

where $r_k = \| p_i - p_{jk} \|$, $a = 2\pi / \theta_{Ki}$,

$$\theta_k = \sum_{l=1}^{k} \angle(p_{j_{l-1}}, p_i, p_{j_l}),$$

Constructing the Mesh Hierarchies(5/5)

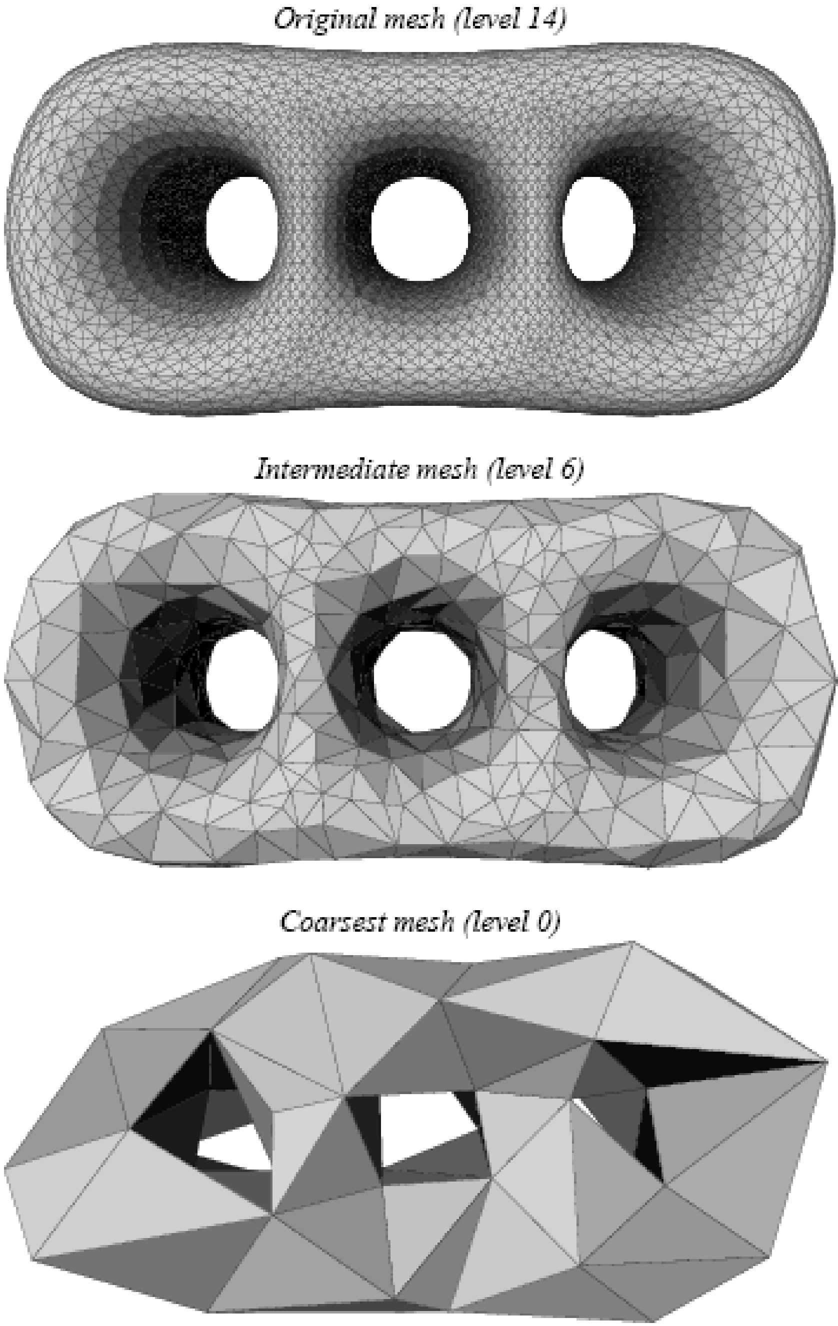
- Now that the hole is flat, retriangulate it
- Use Constrained Delaunay Triangulation



Hierarchy

- *Coarsest mesh (level 0)*

  Is base domain



Parameterization (1/3)

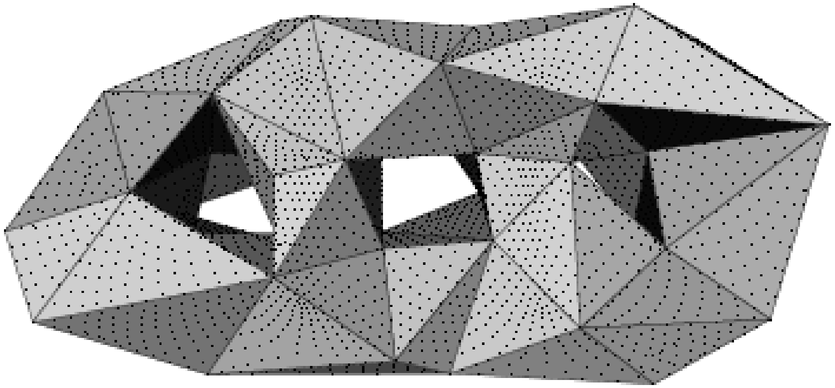
- Use barycentric coordinates
- Want to have a mapping P^l from the top level L to mesh level l which will allow us to map points between meshes at any level of the hierarchy

Parameterization (2/3)

- Constructing Π^{l-1} for vertex $\{i\}$:
  - $\{i\}$ was in previous level, nothing to do
    - $\Pi^{l-1}(p_i) = \Pi^{l}(p_i) = p_i$
  - $\{i\}$ just got removed in the current level
    - $\Pi^{l-1}(p_i) = \alpha p_j + \beta p_k + \gamma p_m$ where p_i is in $\{j,k,m\}$ in the new level,

Parameterization (3/3)

- {*i*} was removed before previous level
  - If the triangle that contained {*i*} at the previous level is still in the new level, do nothing.
  - Otherwise, assign barycentric coordinates based on the new triangle that {*i*} is in.



Preserving Image Features(1/2)

- User manually tags features
  - Vertices
    A certain vertex can be crucial to the character of an image
  - Edge Paths
    Want to keep certain boundaries or contours present in the base domain
  - Solution: Mark vertices as unremovable at all stages of mesh simplification

Preserving Image Features(2/2)

- Automatic detection of features
  - Vertices:
    Set some curvature threshold.
    If the curvature of the 1-ring of a vertex *exceeds* the threshold, mark it as unremovable.
  - Edge Paths:
    Set a threshold for the dihedral angle of an edge.
    If the dihedral angle of an edge is *less than* the threshold, mark the edge.

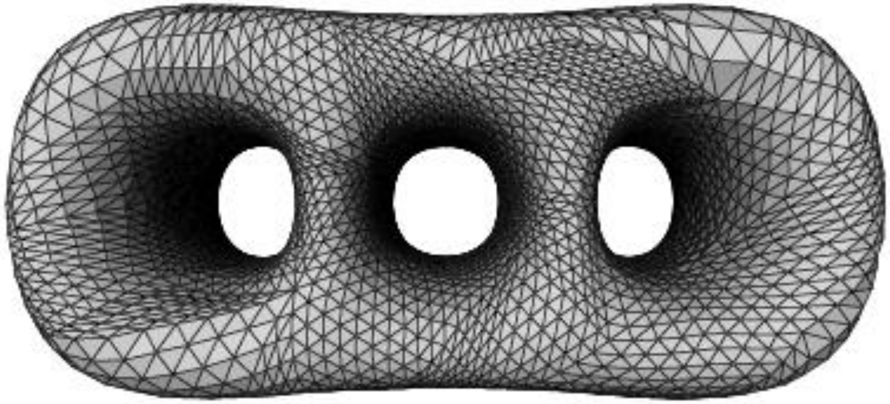
Remeshing(1/5)

- Uniform
  - Use Loop subdivision variant
  - Subdivide an edge to get new vertex q
    But make the new point q close to the original mesh since we have that information
  - Find $\{i, j, k\}$ in the original mesh that contains q
    Then $q = \alpha \prod(p_i) + \beta \prod(p_j) + \gamma \prod(p_k)$ [in base domain]
    $\prod^{-1}(q) = \alpha p_j + \beta p_j + \gamma p_k$ [in top mesh]
- Use $\prod^{-1}(q)$ as new vertex

Remeshing(2/5)

- Smooth

  Parameterization not smooth across base domain triangles



Remeshing(3/5)

- Cases on the stencil needed to compute and smooth a new point:
  - All points in the stencil are within the same base domain triangle
    - Then use the normal Loop rule
  - The stencil is in two base domain triangles
    - Flatten the triangles at the edge
    - Compute the new point
    - Determine the top-level triangle which contains the new point and compute the barycentric coords of the new point
  - The stencil is on multiple base domain triangles
    - Use the conformal mapping to flatten and then subdivide

Remeshing(4/5)

- Adaptive
  - Define an error metric $E(t)$: (for triangle t in the base domain):

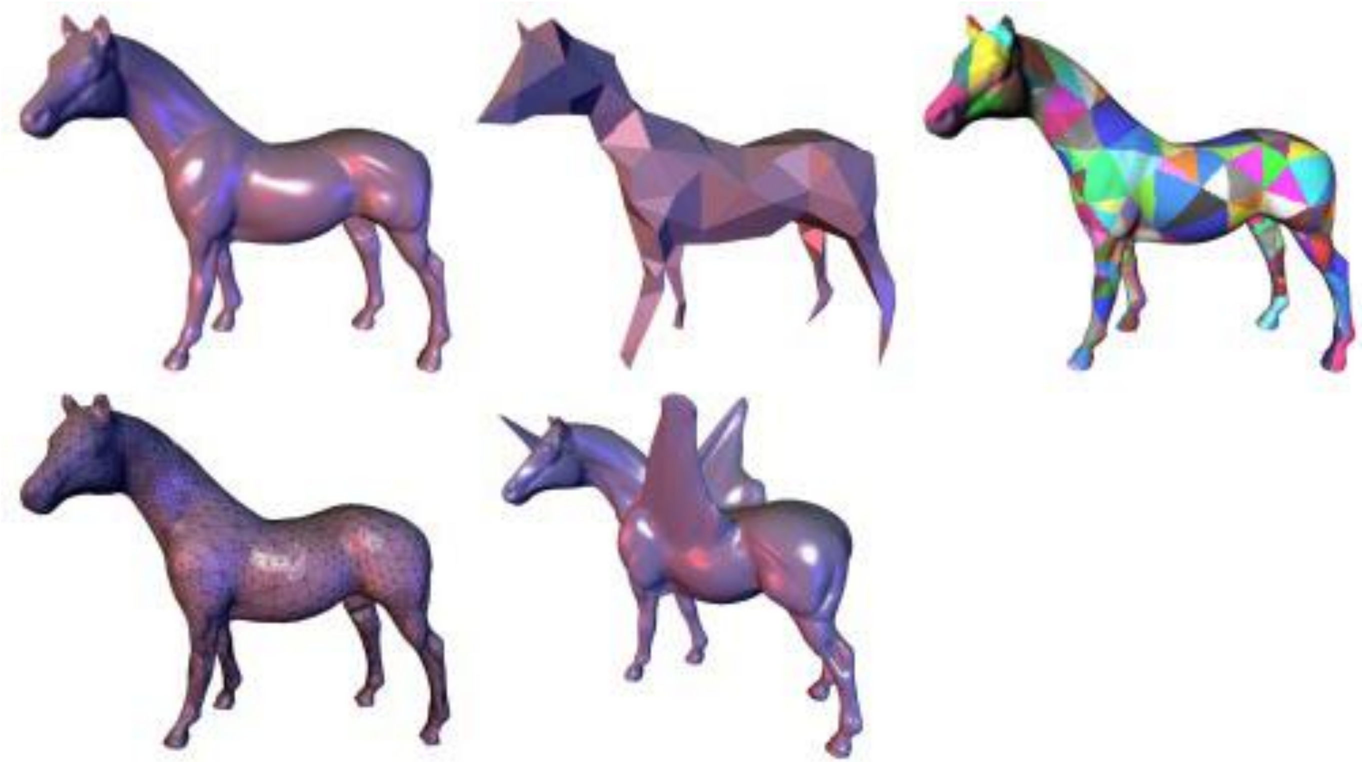
$$E(t) = \max_{p_i \in P^L \text{ and } \Pi(p_i) \in \varphi(|t|)} \operatorname{dist}(p_i, \varphi(|t|)).$$

  - In other words, the maximum distance between a base domain triangle and the finest triangle that gets mapped to it

Remeshing(5/5)

- Also need an error threshold ε
- Algorithm:
  - Compute $E(t)$ for all triangles of current mesh level (starting with the base domain)
  - If $E(t)/B > \varepsilon$ where B is the longest bounding box side,
    - Then Loop subdivide the current triangle t
    - Take each vertex p (in P^L and t) and assign it to the closest child triangle of t
- Loop (iterate, not subdivide) until the ε threshold is met for all triangles in the remeshing

Result



Result

Dataset	Input size (triangles)	Hierarchy creation	Levels	P^0 size (triangles)	Remeshing tolerance	Remesh creation	Output size (triangles)
3-hole	11776	18 (s)	14	120	(NA)	8 (s)	30720
fandisk	12946	23 (s)	15	168	1%	10 (s)	3430
fandisk	12946	23 (s)	15	168	5%	5 (s)	1130
head	100000	160 (s)	22	180	0.5%	440 (s)	74698
horse	96966	163 (s)	21	254	1%	60 (s)	15684
horse	96966	163 (s)	21	254	0.5%	314 (s)	63060

Table 1: *Selected statistics for the examples discussed in the text. All times are in seconds on a 200 MHz PentiumPro.*