Parallel Computation of Spherical Parameterizations for Mesh Analysis

Th. Athanasiadis and I. Fudos
University of Ioannina, Greece

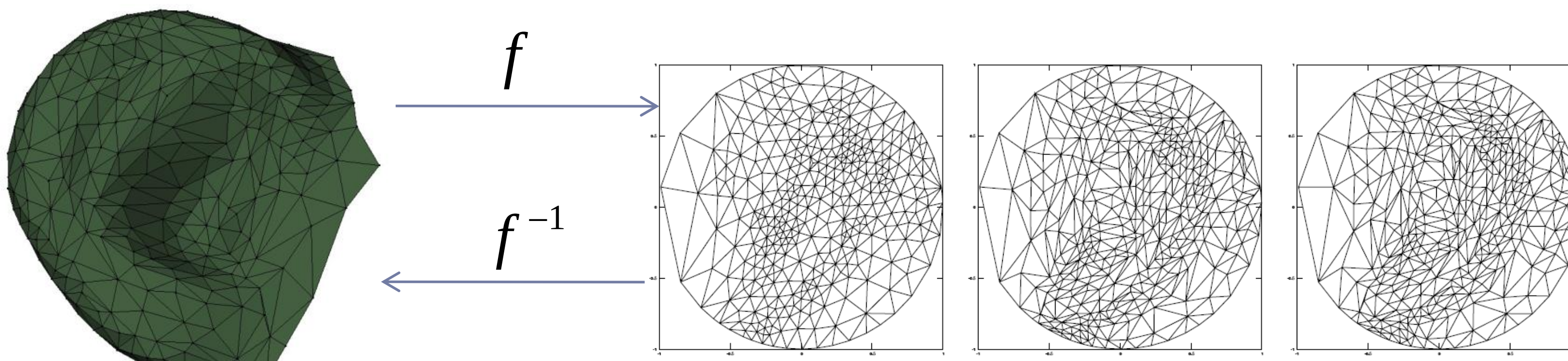
Introduction

- Mesh parameterization is a powerful geometry processing tool

- Applications
  - Remeshing
  - Texture mapping
  - Segmentation
  - Shape search
  - Morphing

- **Conclusion**: fast mesh parameterization is central to many applications

Parameterization definitions



(a) Barycentric (b) Conformal (c) Mean value

S Ω

- Surface $S \in \mathbb{R}^3$
- Parameter domain $\Omega \in \mathbb{R}^2$
- Bijective Mapping $f : S \to \Omega$ and $f^{-1} : \Omega \to S$

(one-to-one correspondence between Ω and S)

Planar parameterization bijectivity

- If we map the boundary vertices to a convex polygon with the same order and express the interior vertices as a convex combination of their neighbors:

$$v_i = \sum_{j \in N_i} w_{ij} v_j$$

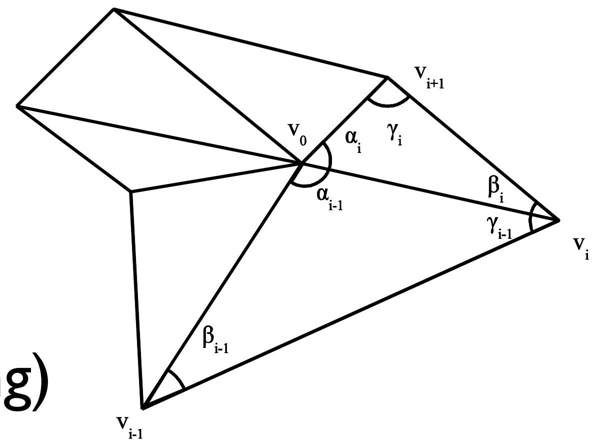
$$\sum_{j \in N_i} w_{ij} = 1$$

$$w_{ij} > 0$$

- Then the map is bijective (without foldovers).
- This can be expressed as a linear system of equations
- The weights can affect the distortion of the parameterization (e.g. minimize the angular distortion)

Parameterization deformation

- The constants affect the deformation of the parameterization



- Authalic (area preserving)

$$w_{i0} = \frac{Area(v_{i-1}, v_i, v_{i+1})}{Area(v_{i-1}, v_i, v_0)\,Area(v_i, v_{i+1}, v_0)} = \frac{\cot\gamma_{i-1} + \cot\beta_i}{\| v_i - v_0 \|^2}$$

- Conformal (angle preserving):

$$w_{i0} = \cot\beta_{i-1} + \cot\gamma_i$$

Physical interpretation

- Consider the edges of the triangle mesh as springs that are connected at the vertices

- If we fix the boundary of the spring network in the plane, then the interior will relax in the energetically most efficient configuration

- If we assume each spring to have potential energy $1/2wl^2$, where *w* is a spring constant and *l* is the length of the spring, then the overall spring energy of the mesh is minimized

$$f(v_1, v_2, \ldots, v_n) = \sum_{(v_i, v_j) \in E} w_{ij} \| v_i - v_j \|^2$$

Planar parameterization extensions

- Planar parameterization is well suited for meshes with a boundary but can be extended to genus-0 objects:
  - Cut a path between two selected poles
  - Remove a triangle from the mesh and map the mesh on the unit triangle
- Problems:
  - Undesirable distortion is introduced by the cuts
  - Unit triangle tends to cluster the remaining vertices in the center

Spherical parameterization

- Express each vertex of S as a convex combination of its neighbors:

$$v_i = \frac{\sum_{j \in N_i} \lambda_{ij} v_j}{\| \sum_{j \in N_i} \lambda_{ij} v_j \|}$$

$$\sum_{j \in N_i} \lambda_{ij} = 1$$

$$\lambda_{ij} = \lambda_{ji}$$

$$\lambda_{ij} > 0$$

- The above can also be expressed as a set of non linear equations:

$$a_i x_i - \sum_{j \in N_i} \lambda_{ij} x_j = 0$$

$$a_i y_i - \sum_{j \in N_i} \lambda_{ij} y_j = 0$$

$$a_i z_i - \sum_{j \in N_i} \lambda_{ij} z_j = 0$$

$$x_i^2 + y_i^2 + z_i^2 = 1$$

Solutions

- We can directly try to solve the problem with non linear optimization techniques

- Problems
  - Degenerate solutions exist (e.g. collapsed solution)
  - Non linear constraints
  - High computation cost

Our approach

- Solve an easier problem with only linear constraints

- **Observation** (spherical tangent planes):

$$f\left(\frac{v_1}{\|v_1\|},\ldots,\frac{v_n}{\|v_n\|}\right) \leq f(v_1,\ldots,v_n)$$

$$\left\| \frac{v_i}{\|v_i\|} - \frac{v_j}{\|v_j\|} \right\|^2 \leq \|v_i - v_j\|^2$$

$$\|v_i\| \geq 1, \|v_j\| \geq 1$$

- The solution to this problem does not lie on the spherical domain. Nevertheless, after normalizing the vertices the spring energy is decreased

- The solution space is constrained and we avoid many degenerate configurations due to the constraints

Algorithm overview

- The equality constrained energy minimization problem is a saddle point problem:

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} r \\ q \end{pmatrix}$$

- $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix containing symmetric weights (e.g. barycentric)

- $B \in \mathbb{R}^{n \times m}$ is a matrix containing the original coordinates (tangential planes normals)

- $r \in \mathbb{R}^{n}$ is a matrix related to the fixed vertices (if we have any)

- $q = \ 1,\ldots,1^{\ T}, q \in \mathbb{R}^{m}$ contains the plane distances

Algorithm overview

- What we gain:
  - The problem is easier to solve
  - Requires only a sparse linear solver
  - Certain degenerate solutions are avoided

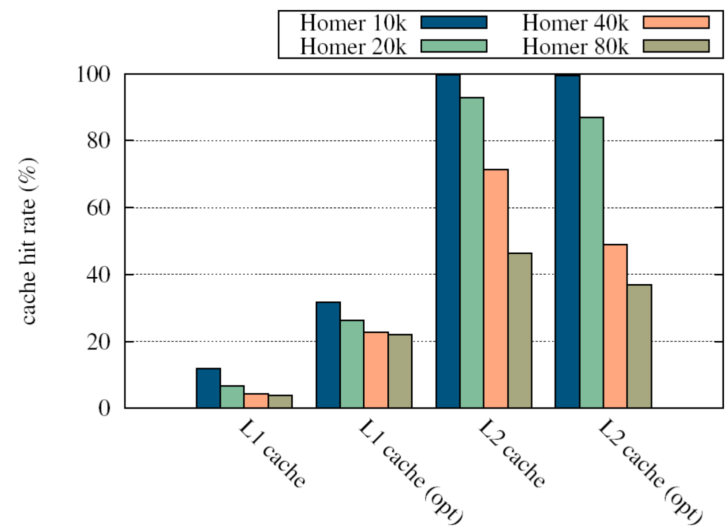
- What we lose:
  - Potentially slower convergence speed (in terms of iterations)
  - We do not directly solve the problem but the procedure converges to the original problem solution
  - Certain degenerate solutions can still occur

Implementation details

- The problem is sparse and therefore more difficult to efficiently map on the GPU
- Better mathematical solvers are often poor GPU candidates (e.g. ILU)
- The challenge is trading off performance with iteration count
- It can be shown that under certain conditions the Jacobi converges to the solution of the saddle point problem
- Further implementation challenges
  - GPU - host synchronization
  - Cache hit ratio

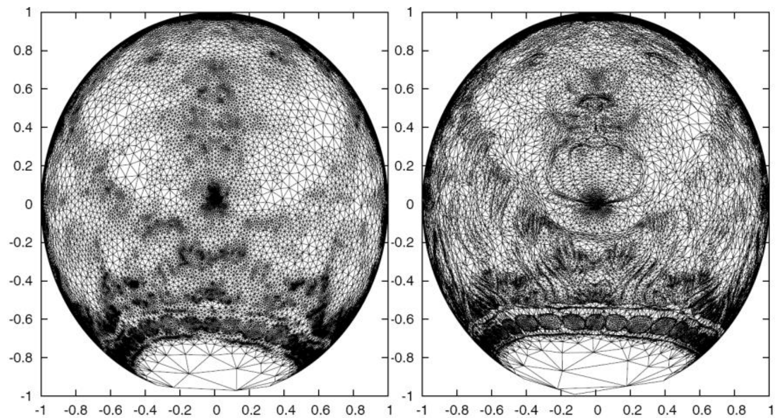
Implementation challenges

- Synchronization cost
  - Can be reduced by using a sparse residual check policy (requires a synchronization)

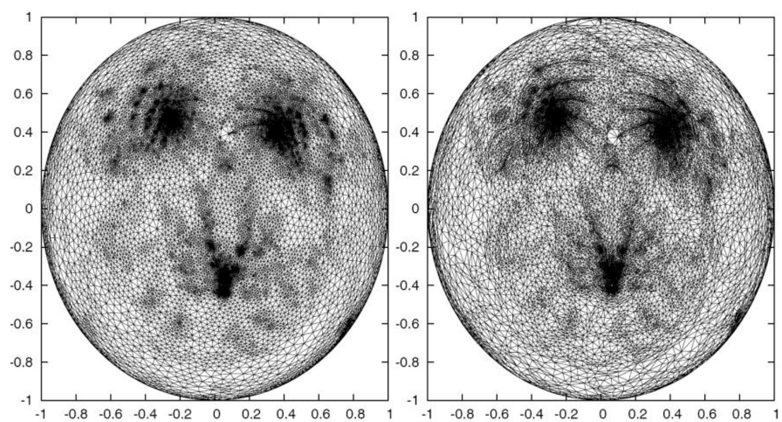


- Cache efficiency
  - The algorithm is *vertex-bound*
  - Can be improved by reordering the vertices for better locality
  - Finding the optimal reordering is an NP-complete problem but heuristic based algorithms usually provide substantial speed gains

Parameterizations



(a) Barycentric (b) Conformal

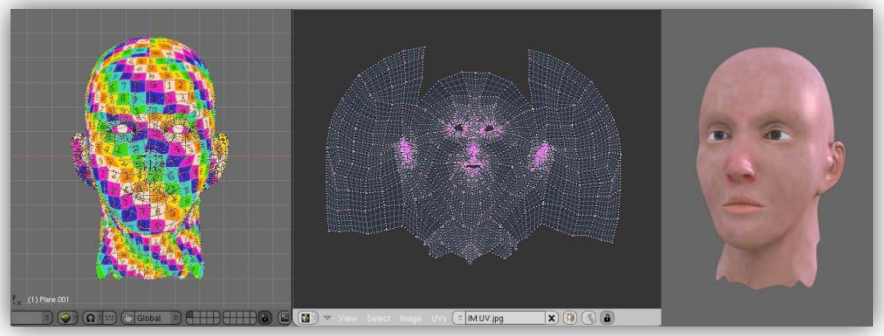


(a) Barycentric (b) Conformal

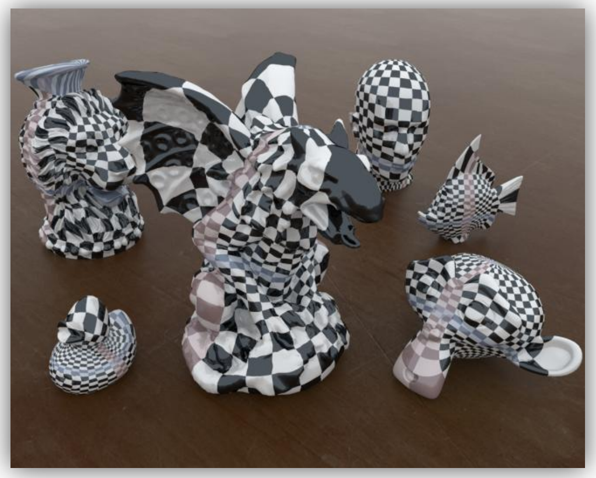
Applications

- Spherical parameterization applications
  - Texture mapping
  - Mesh segmentation
  - Shape search

Applications - Texture mapping



- Texture mapping (Least squares planar conformal parameterization) of Blender modeler



- Seamless, continuous texture mapping of Genus-0 models (conformal parameterization) with our approach.

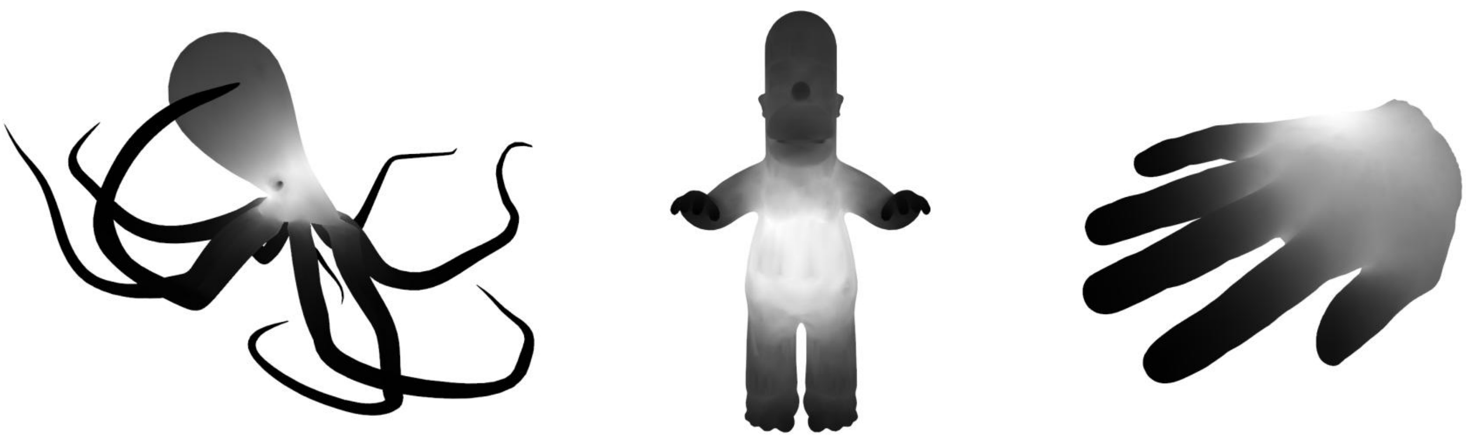
Applications

- Spherical parameterization applications
  - Texture mapping
  - Mesh segmentation
  - Shape search

Mesh analysis motivation

- **Idea:** the spherical embedding represents a pose invariant representation of the mesh.

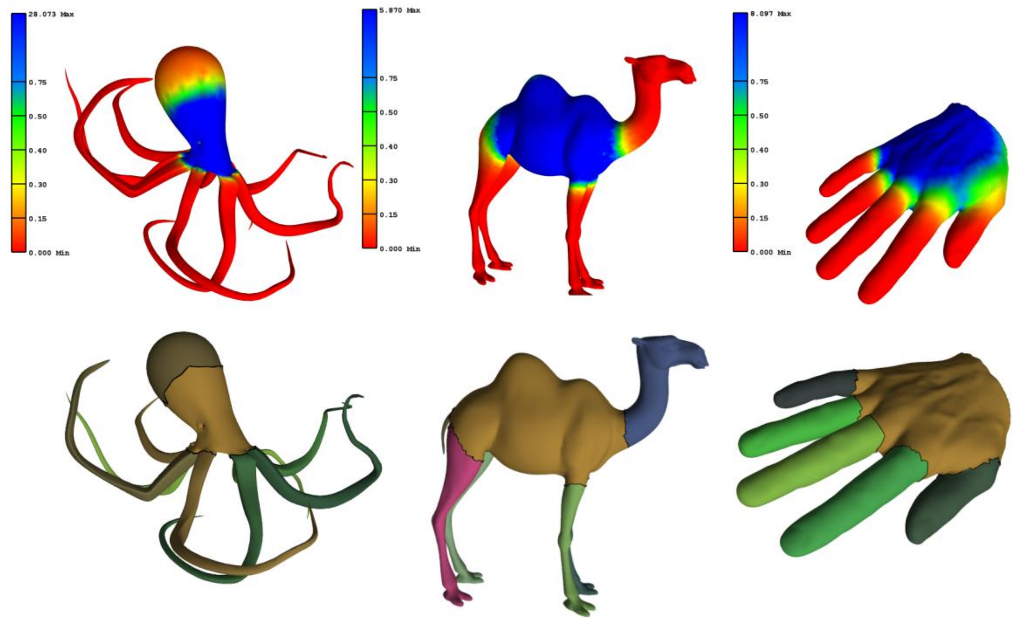
- **Observation:** due to the prominent extremities any, spherical embedding is expected to create some dense concentrations of faces.



Mesh segmentation algorithm

- Compute the *Area stretch factor* $A(v_i)$ of each vertex v_i (average area stretch deformation of its adjacent faces).
- Region growing approach that starts from *seeds* and expands while a threshold in the variation of the area stretch factor is satisfied.
- A *seed* is a vertex if and only if $A(v_i)$ exhibits a local minimum or maximum at v_i .

Mesh segmentation stretch visualization



- Visualization of the ratio between the mapped area and the original surface and the final segmentation.

- Red means that a very small spherical area is assigned to a large area in the original model

Advantages - Limitations

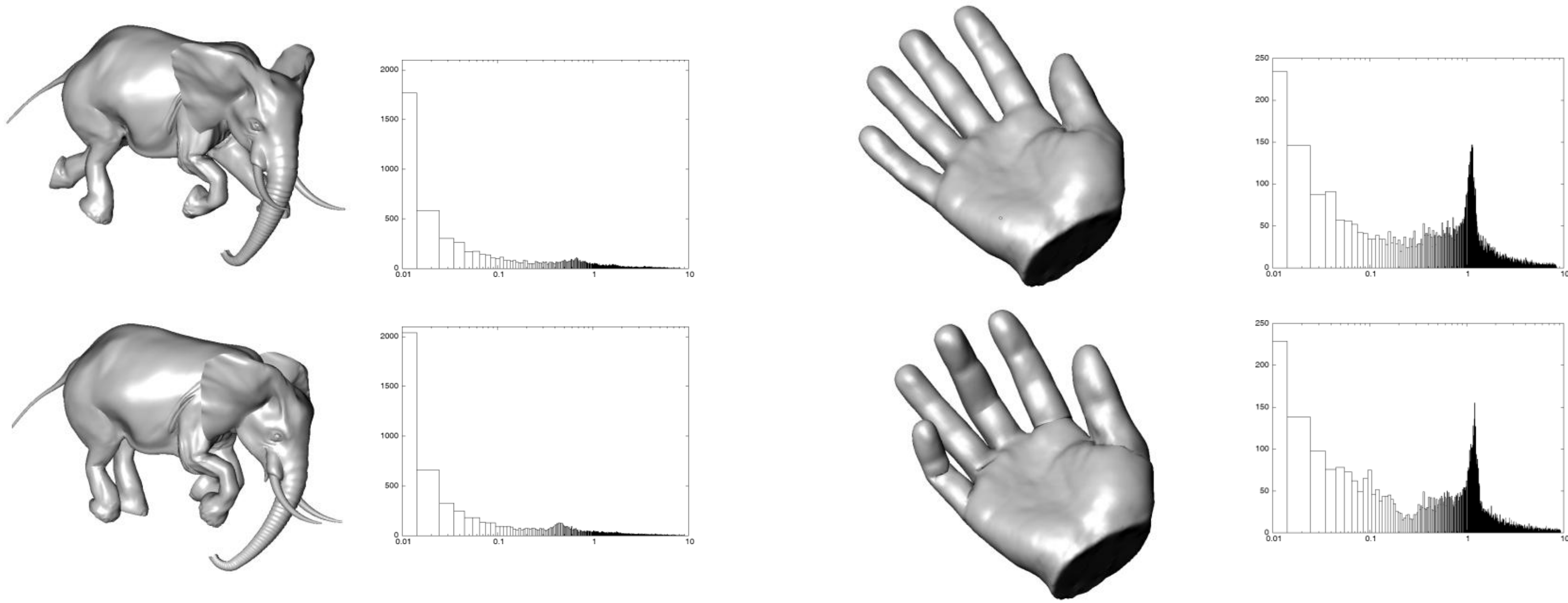
- Advantages
  - Simple metric (geometric path from sphere)
  - Suitable for models with limbs
  - Fast even for large meshes (> 100k triangles)
  - No need for decimation to reduce the segmentation cost

- Disadvantages
  - Post processing must be used for application specific mesh-segment refinement (e.g. for mechanical objects)

Applications

- Spherical parameterization applications
  - Texture mapping
  - Mesh segmentation
  - Shape search

Shape search



- Key idea:
  - Compare signatures derived from the parameterizations
  - A signature can be a histogram of the area stretch factor
  - The signature can be tessellation independent with uniform or random sampling

Results

Table 1: Numerical results for finding a spherical parameterization on the GPU with different models

model	map	# vertices	# faces	# iterations	L_2 res ($\times 10^{-8}$)	time (secs)
Suzanne	Barycentric	7573	15142	4	5	0.575
Suzanne	Conformal	7573	15142	3	5	0.589
Gargoyle	Barycentric	24990	49976	4	2	1.706
Gargoyle	Conformal	24990	49976	3	6	2.326
Igea	Barycentric	25586	51168	3	4	0.908
Igea	Conformal	25586	51168	2	3	0.936
Lion Vase	Barycentric	38952	77900	3	3	1.567
Lion Vase	Conformal	38952	77900	3	3	2.053
Homer	Barycentric	78850	157696	3	1	4.923
Homer	Conformal	78850	157696	3	4	10.920
Buste	Barycentric	183580	367156	3	1	13.759
Buste	Conformal	183580	367156	2	1	22.667

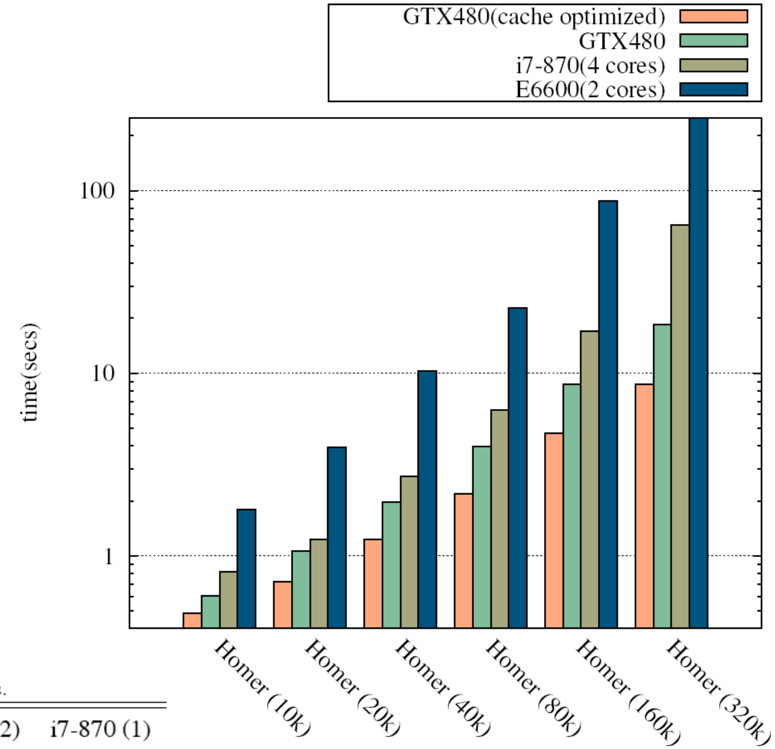
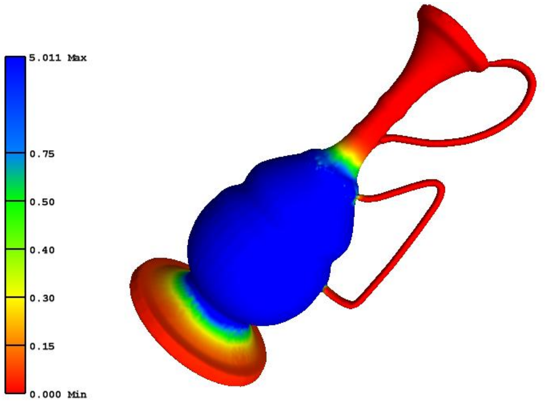


Table 3: Comparison of running times (in secs) between GPU and CPU with different core configurations.

model	map	# vertices	# faces	# iterations	GTX 480	i7-870 (4)	i7-870 (2)	i7-870 (1)
Gargoyle	Barycentric	10002	20000	4	0.946	1.186	1.950	3.135
Gargoyle	Conformal	10002	20000	4	0.949	1.045	1.685	2.714
Torso	Barycentric	11362	22720	4	0.718	1.107	1.731	2.808
Torso	Conformal	11362	22720	3	0.870	1.123	1.747	2.745
Skull	Barycentric	20002	40000	3	0.649	1.076	1.719	2.904
Skull	Conformal	20002	40000	2	0.643	0.920	1.373	2.230
Bunny	Barycentric	67038	134074	3	1.217	3.616	6.635	12.038
Bunny	Conformal	67038	134074	2	2.158	3.778	7.737	14.118

Conclusions and future work

- We have presented a fast spherical parameterization algorithm for genus-0 meshes with application in mesh analysis
- Arbitrary genus meshes
- Different types of parameterizations (e.g. authalic)



- Shape search evaluation (in terms of hit and misses)

Thank You

Questions:

thathana@cs.uoi.gr

fudos@cs.uoi.gr

Software:

www.cs.uoi.gr/~fudos/smi2011.html