# Parallel computation of spherical parameterizations for mesh analysis

Theodoros Athanasiadis and Ioannis Fudos, Dept of Computer Science, University of Ioannina

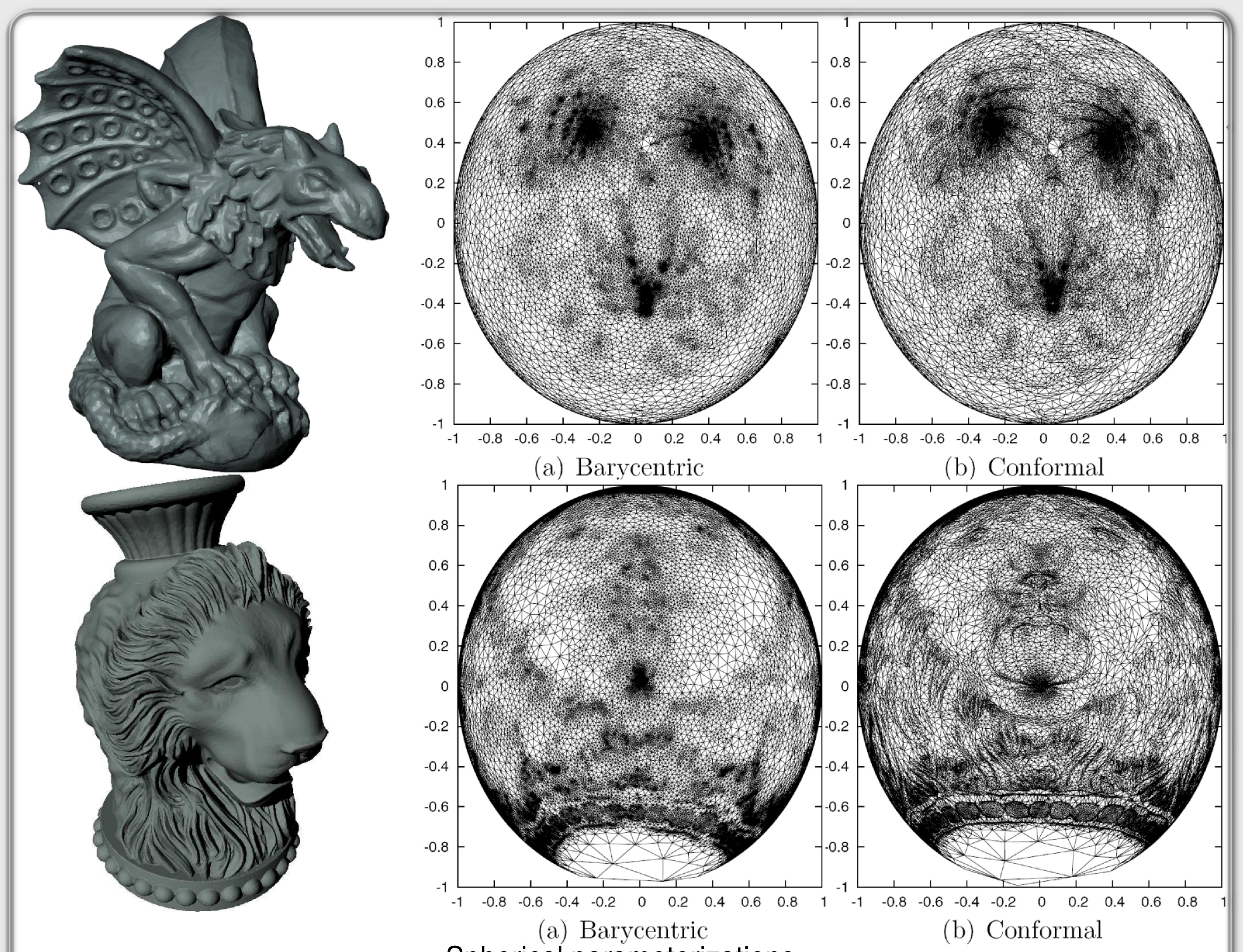
## Introduction

Mesh parameterization is of great importance to a broad spectrum of applications. In this work, we present a novel approach to spherical mesh parameterization based on an iterative quadratic solver that is efficiently parallelizable on modern massively parallel architectures. We introduce a number of heuristics that exploit various system characteristics of the underlying architectures to speed up the parallel realization of our algorithms. Furthermore, we demonstrate the applicability of our approach to real-time feature detection, mesh decomposition and similarity-based 3D object retrieval.

### Spherical parameterization

Fast mesh parameterization is central to many applications such as remeshing, filtering, texture mapping, compression, mesh completion and morphing. The existing spherical mesh parameterization methods can roughly be classified into two categories:

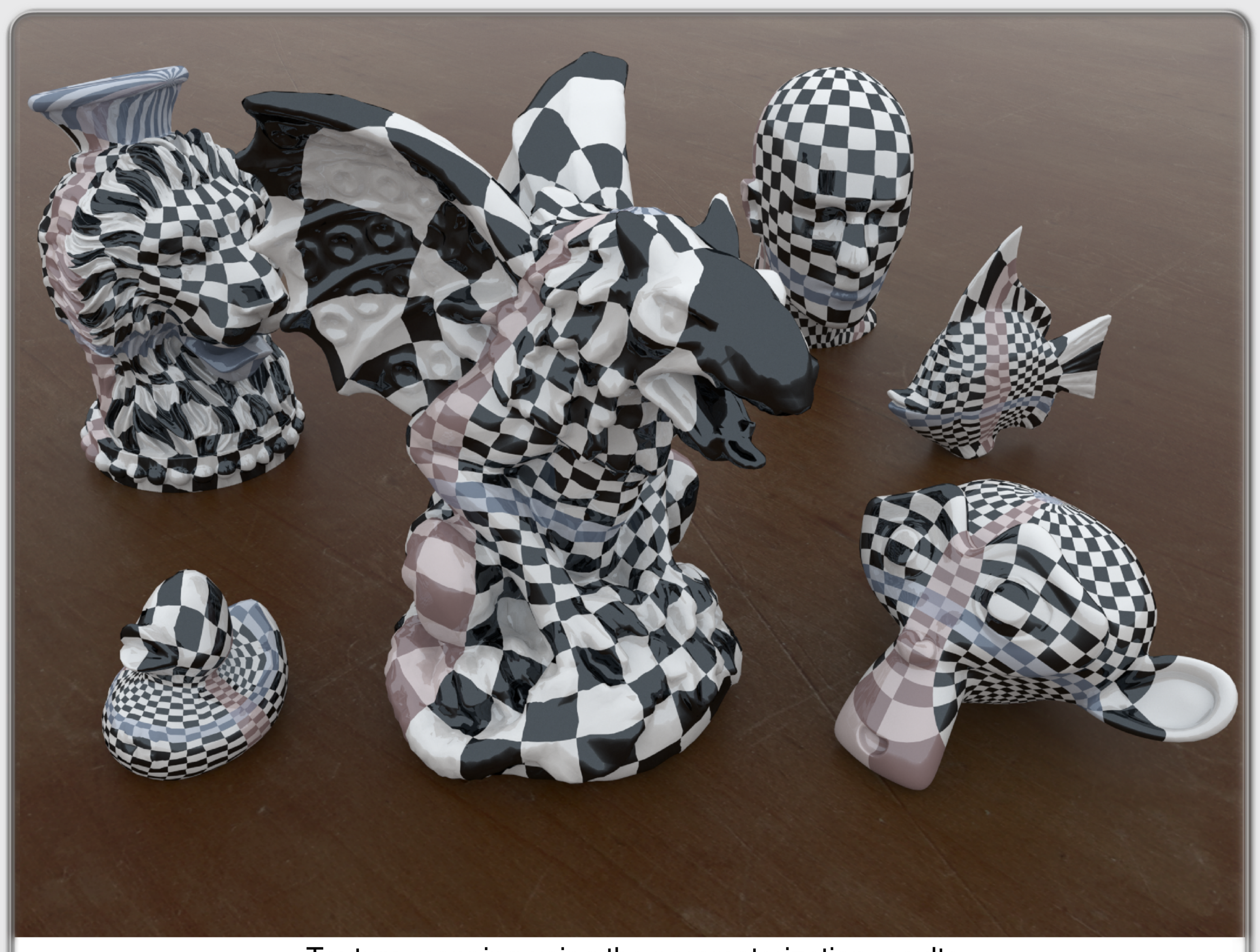
- Methods that attempt to extend planar methods
- Methods that use some kind of non-linear optimization




Spherical parameterizations

Typical methods of the former category introduce cuts in the mesh and increase the distortion of the parameterization, an effect that may be undesirable by some applications. Therefore, it is advantageous to directly parameterize the meshes on the spherical domain to allow seamless continuous parameterizations of genus-0 meshes.

In the latter category the spherical parameterization problem is usually approached as an energy minimization problem (minimizing angular or area distortion) subject to the vertices lying on the sphere. Nevertheless, the usual approach for computing the parameterizations using non-linear optimization has a high computation cost due to the non linearity of the constraints.



Texture mapping using the parameterization results

To effectively employ a fast parallelizable scheme, we model the problem as a set of equality constrained quadratic (saddle point) problems subject to a set of linear constraints. The problems are then solved using a sparse linear solver well suited to massively parallel architectures such as modern GPUs [1].

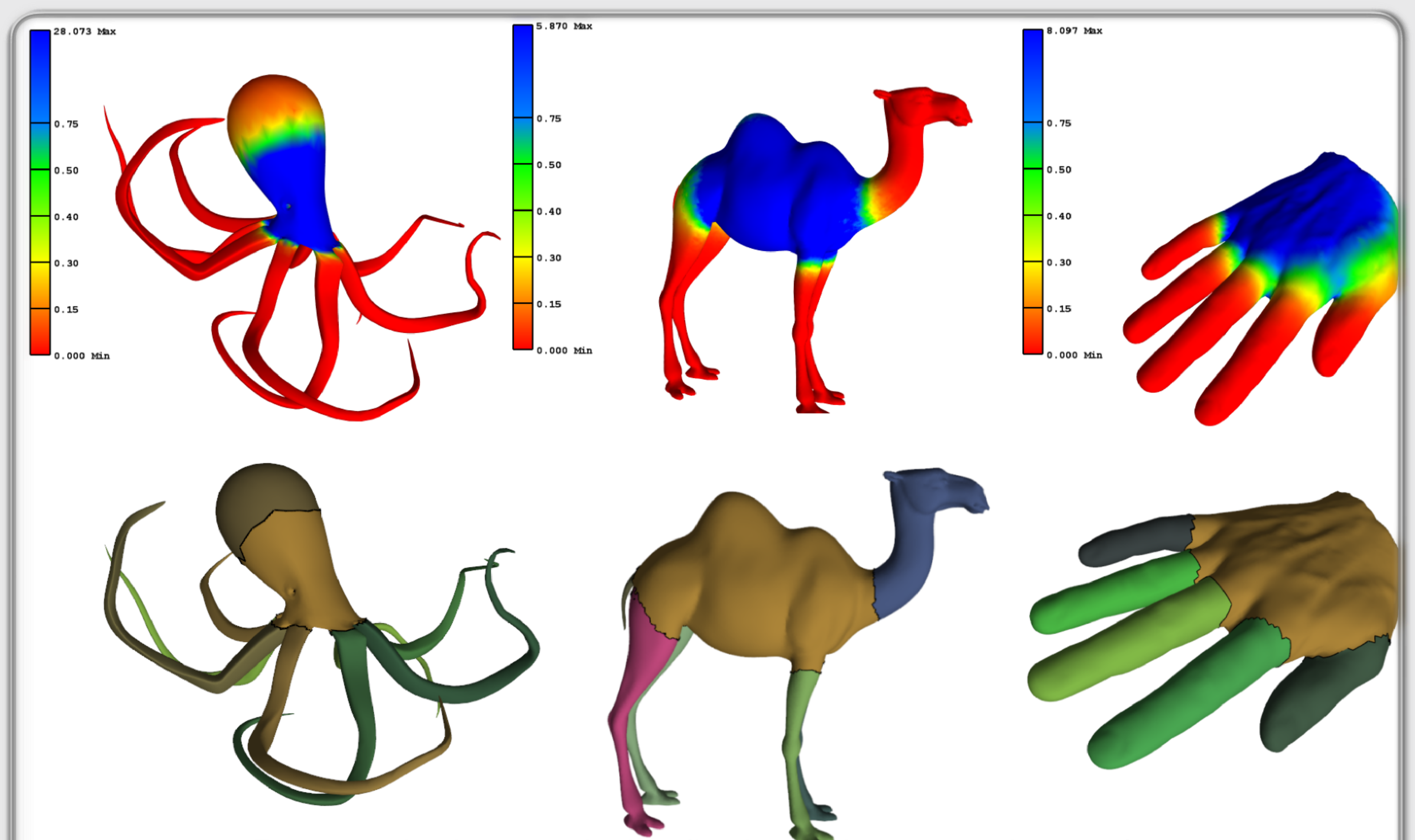
### Implementation

As an API for our implementation, we have used OpenCL to achieve almost direct portability of our core source to both GPU and CPU based architectures. Moreover, we have investigated various optimization principles optimize the performance of the proposed algorithm on the GPU:

- Cache hit ratio - we improve it by reordering the vertices for better locality
- Synchronization cost - we can reduce it by using a sparse residual check policy

### Mesh segmentation

Any spherical parameterization is expected to create some dense concentrations of faces on the sphere due to the prominent extremities of the mesh. The extruding parts of the meshes, for example the limbs, are usual to be mapped to relatively small regions on the sphere. This effect is more evident when the angular distortion of the parameterization is minimized. Therefore, the spherical embedding of a mesh contains a substantial amount of information about its geometric shape. We have carried out a number of experiments with a region growing approach that takes advantage of the above observation. The method starts from initial vertices (seeds) on the sphere and expands while a threshold in the variation of the area stretch deformation is satisfied. This results in a number of regions that represent object extremities.

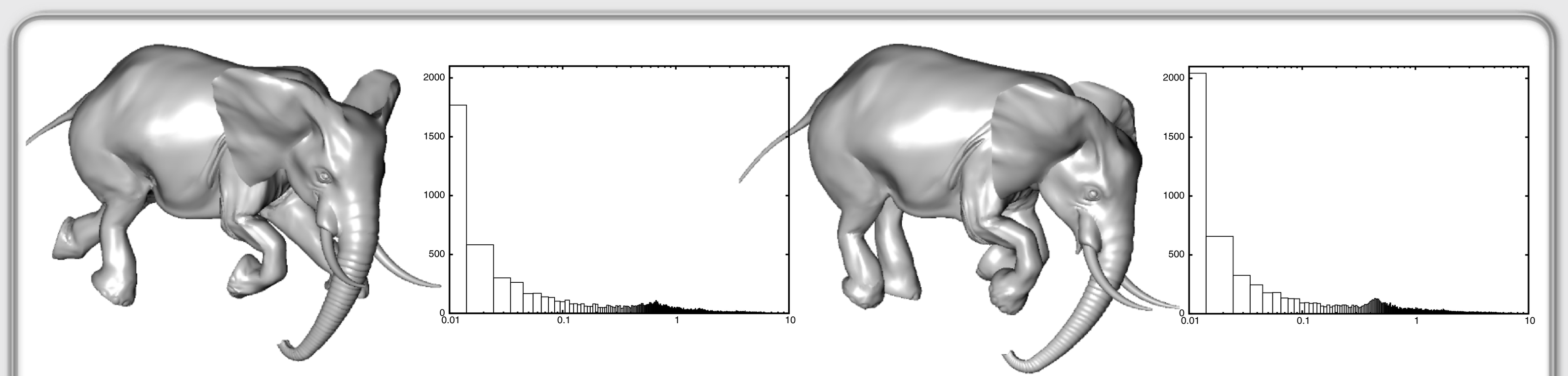


Parameterization stretch visualization and segmentation results

### Shape Search

Key idea:

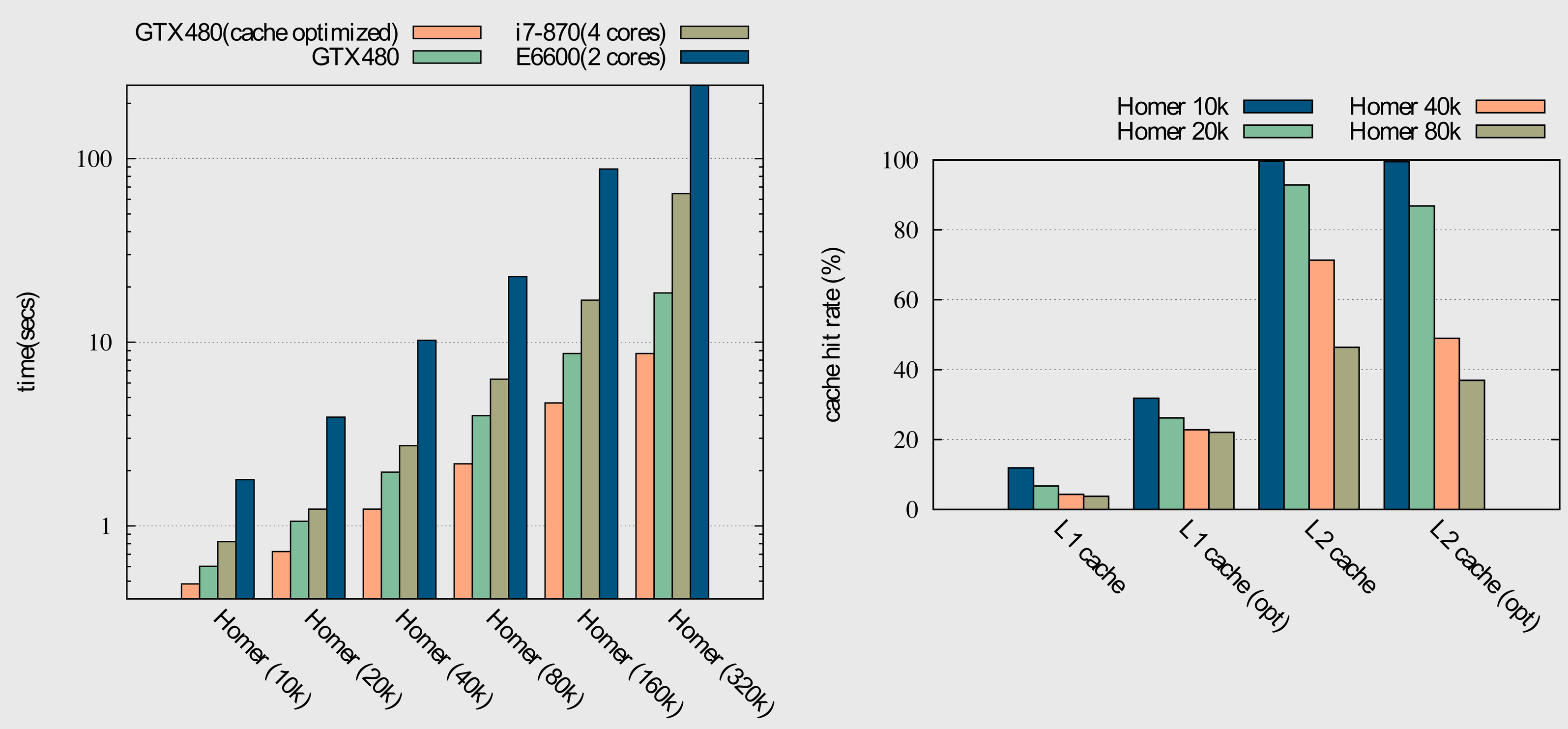
- Compare signatures derived from the parameterizations
- A signature can be a histogram of the area stretch factor
- With uniform or random sampling the signature can become tessellation independent



Histograms of different elephant poses

### Results

We have successfully used our scheme to parameterize meshes of up to 400K triangles on a modern GPU in less than 25 secs. Furthermore, we have carried out a large number of experiments to validate that our iterative method converges to the actual bijective mapping.




### Future Work

- Compare results with more sophisticated solvers (e.g. Multigrid,Krylov)
- Extend to arbitrary genus objects
- Implement a non linear solver for feature guided parameterizations

### Software, Source and Video

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

### References

[1] Th. Athanasiadis and I. Fudos - *Parallel Computation of Spherical Parameterizations for Mesh Analysis* - Computers & Graphics Journal vol. 35, issue 3, special issue on Shape Modeling International 2011

The research project is co-funded by the European Union - European Social Fund (ESF) & National Sources, in the framework of the program "HRAKLEITOS II" of the "Operational Program Education and Life Long Learning" of the Hellenic Ministry of Education, Life Long Learning and religious