A Parametric Space Approach to the Computation of Multi-Scale Geometric Features

Anthousis Andreadis   Georgios Papaioannou   Pavlos Mavridis

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

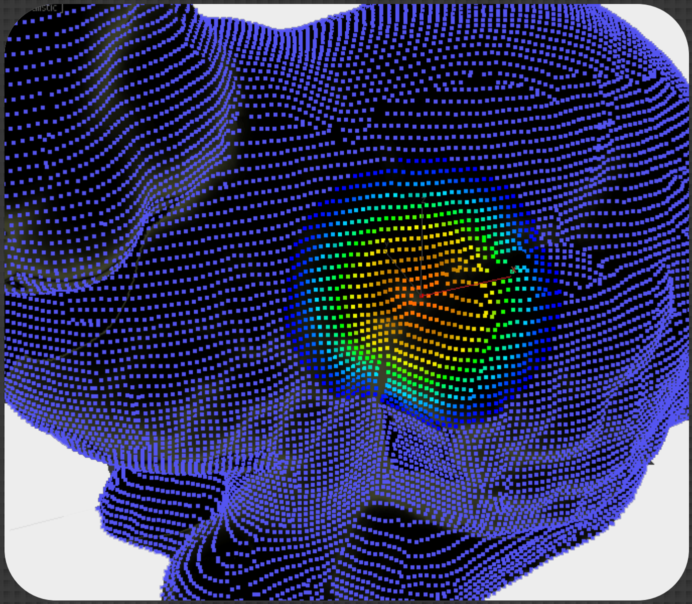
PRESIOUS

Introduction

- Geometric Features are central in a wide range of applications
  - Example Features:
    Curvature, Shape Index
  - Example Applications:
    Object Retrieval, Registration, Stylized Rendering
- Static geometry: Pre-compute
- Dynamic/Animated: Fast-computation is challenging

Feature Computation

- We focus on the general case of features with finite local support
- Key Element
  - Vertex Adjacencies/Point Neighbors
  - N-ring, Euclidean or Geodesic Distance



Related Work

- Existing methods can be classified based on the sampling method of the geometry
  - Object space
  - Volumetric
  - Screen Space
  - Parametric space

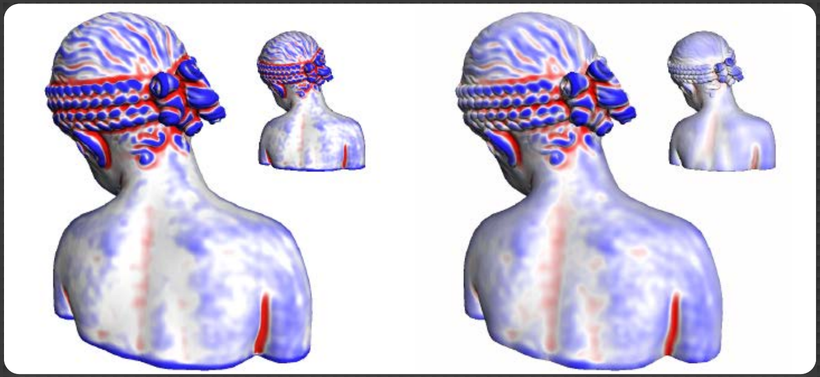
Object space

- Data structure encoding the adjacency is required (half-edge, kD-tree etc)
- These methods do not scale well as computational complexity is directly linked to
  - Geometric density
  - Area of support
- GPU mapping is non-trivial. Existing approaches do not generalize the sampling neighborhood. [Griffin et al., 2011]

Screen space

- Sample geometric information from a 2D pixel buffer.
- Adjacencies are implied by the pixel grid
  - Trivial sampling, efficient mapping to GPU's
- Disadvantage: Computations and area of support area limited to the visible point set
  - Inaccuracies, near occlusion points and at screen-space silhouettes

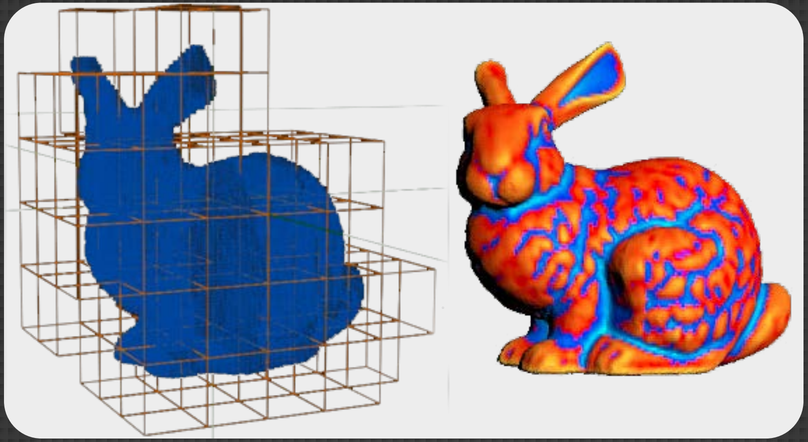
[Mellado et al., 2013]



Volumetric

- A volumetric representation is used (ex. level-set)
- Computational complexity now depends on the representation
- Disadvantages
  - Volumetric discretization is far more rough than the original surface
  - Incompatible results (ex. non-manifold surfaces)

[Museth, 2013]



Parametric Space

- Methods of this category rely on the unwrapped surface of the model on a 2D plane
- Computational complexity decoupled from the geometry
- Disadvantages
  - Neighbor discovery is not trivial
  - Cannot be performed directly on point clouds
- Existing methods are not generic
  - [Novatnack and Nishino, 2007] focus on image space techniques
  - [Hua et al.] Require specific unwrapping methodologies.

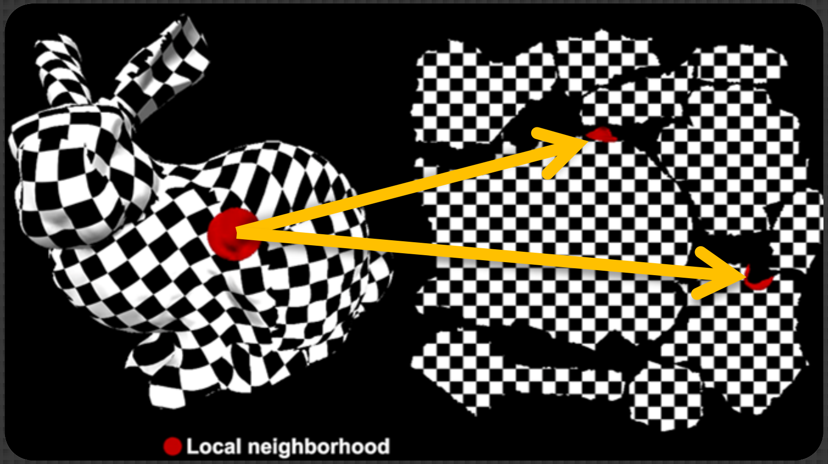
Motivation

- Design a method that is efficient, accurate and generic
  - Efficiency: Close to real-time even for large area of support for animated/deformable objects
    - Excludes Object Space
  - Accuracy: Similar results to a reference Object Space method
    - Excludes Screen Space & Volumetric
  - Generality: Not restricted to a specific feature, or parameterization

Method Overview

- Operates in parametric-space, but is agnostic to the actual mapping of the surface

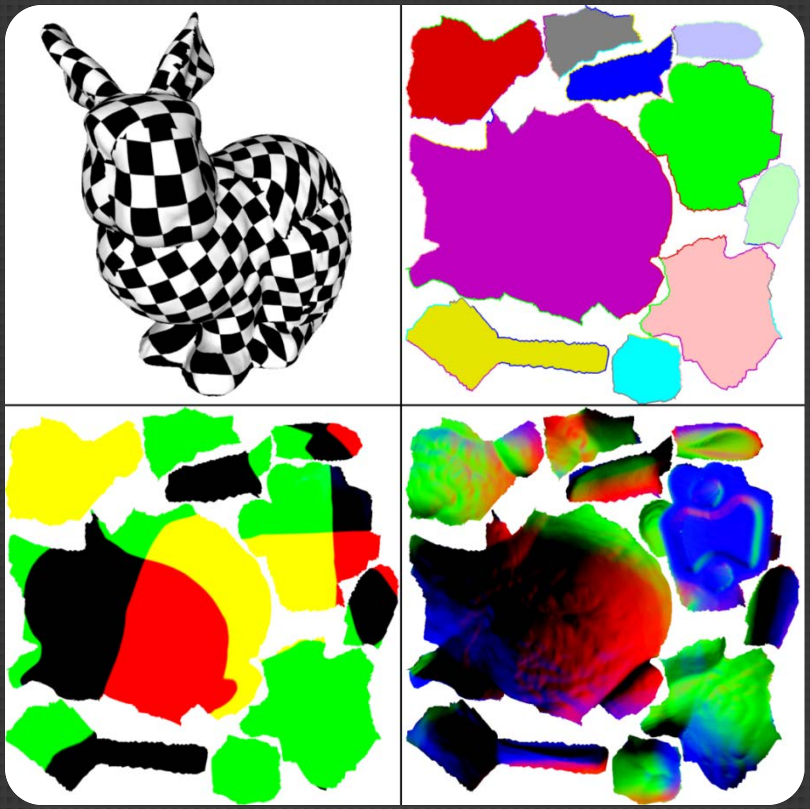
- Vertex Adjacencies → Pixel Adjacencies
  - Not a perfect world: Chart boundaries create discontinuities of geometric adjacencies



Method Overview

- Pre-Process
  - Locate affected edges and store extra information
- Real-Time
  - Create Data Buffers
    - Geometry, Normal, Adjacency
  - Recreate Adjacencies and perform Computations

Data Buffers



Data Buffers

- Information is stored in Textures
  - Geometry Buffer
    - *Object space positions, Chart id*
  - Normal Buffer
  - Adjacency Buffers
    - Discontinued Edges
      - *Adjacent chart id*
      - *Corresponding chart coordinates*
      - *Relative Scale & Rotation*
    - *Local metric distortion (LMD)*
      - *Angular distortion*
      - *u, v stretch factors*

Data Buffer Generation

- Rasterize object triangles
  - Chart boundary edges are rasterized separately to avoid disconnected regions

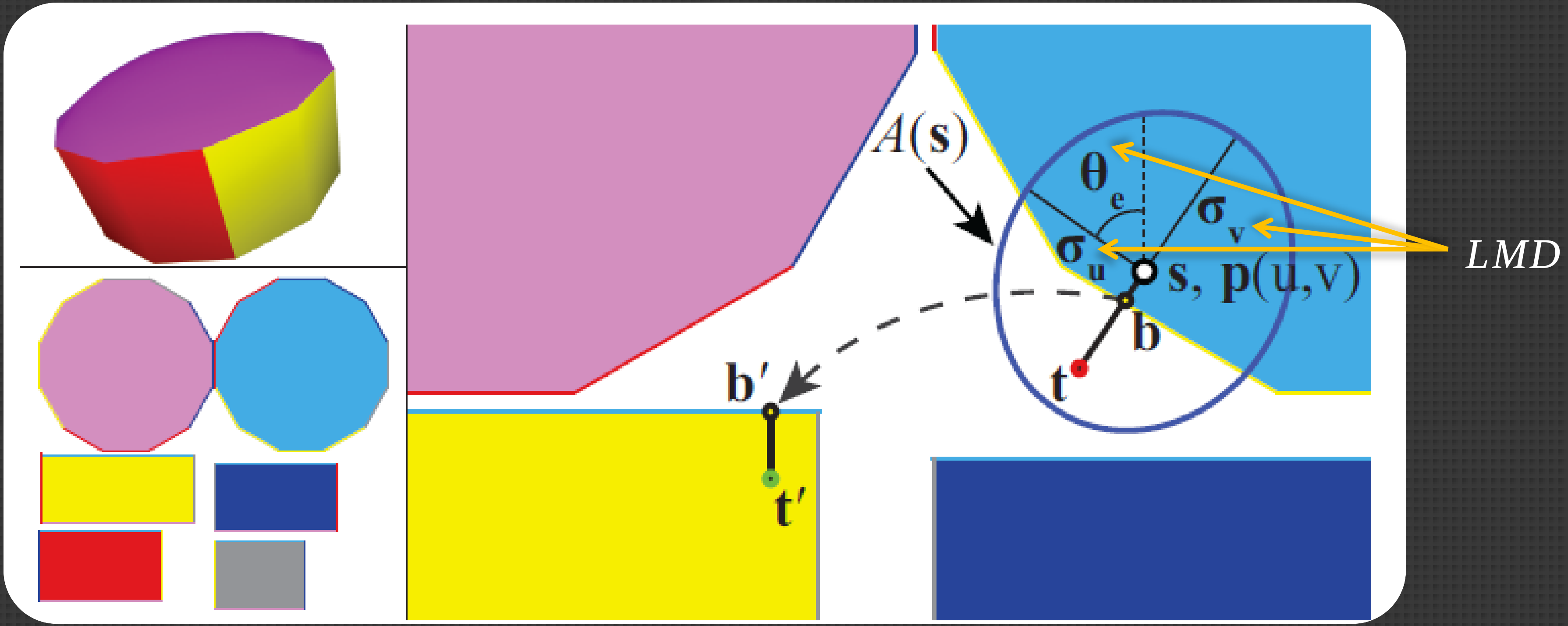
- LMD factors computed using eigen-decomposition of the *first fundamental form matrix*
  - Used for the anisotropic adjustment of scale and sampling directions

$$J_P^T J_P = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \qquad E = (\partial P(u,v)/\partial u)^2 \quad G = (\partial P(u,v)/\partial v)^2$$

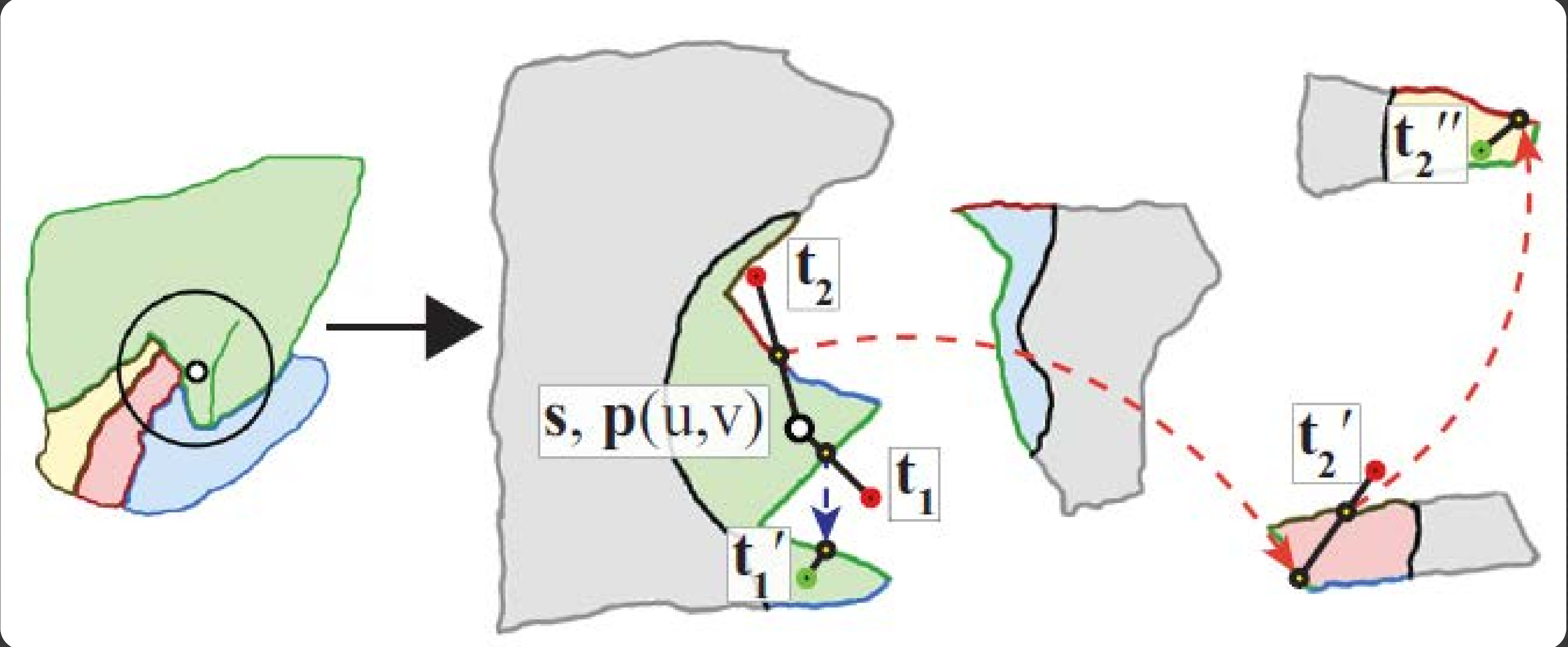
$$F = (\partial P(u,v)/\partial u) \cdot (\partial P(u,v)/\partial v)$$

Sampling the Neighborhood of a Point

$$\mathbf{t}' = \mathbf{b}' + \mathbf{R}_{\theta(\mathbf{b}\to\mathbf{b}')}\mathbf{S}_{s(\mathbf{b}\to\mathbf{b}')}(\mathbf{t}-\mathbf{b}) \qquad s(\mathbf{b}\to\mathbf{b}') = \left(\frac{\sigma_u(\mathbf{b}')}{\sigma_u(\mathbf{b})}, \frac{\sigma_v(\mathbf{b}')}{\sigma_v(\mathbf{b})}\right)$$

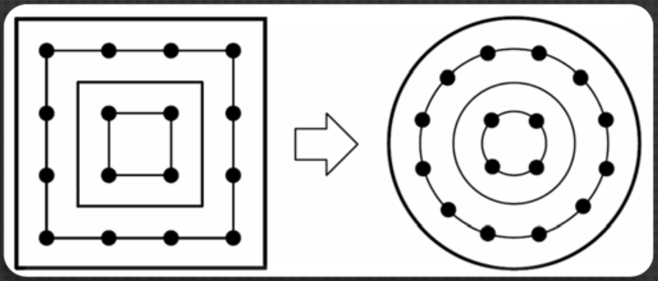


Sampling the Neighborhood of a Point



Monte Carlo Integration

- Geometric feature computation is usually performed with surface and volume integrals
- We estimate by Monte Carlo integration
- Generate random samples using a stratification scheme on a grid and transform them to disk using concentric mapping

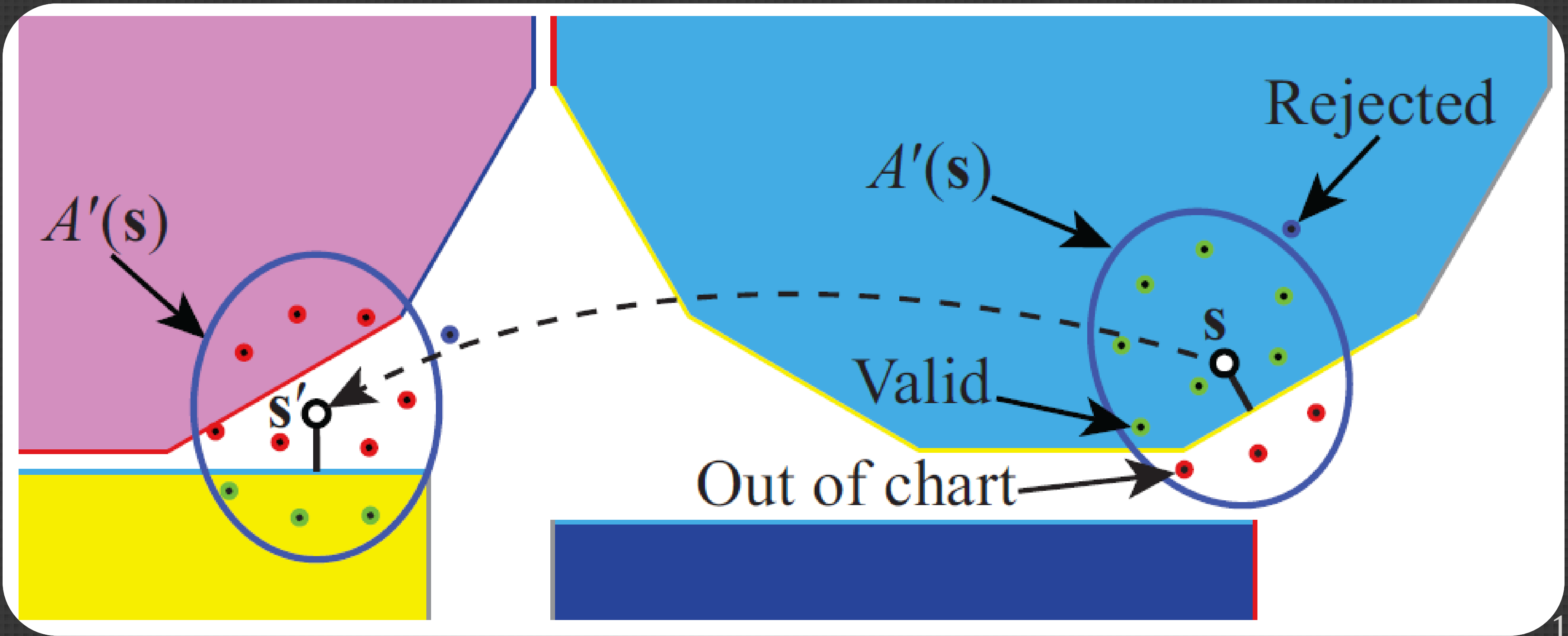


[Shirley and Chiu, 1997]

- Disk samples are anisotropically scaled and rotated according to *LMD* factors.
- We sample $A(\mathbf{s})$ ellipse using sample rejection based on the criterion of neighborhood $S(\mathbf{p})$

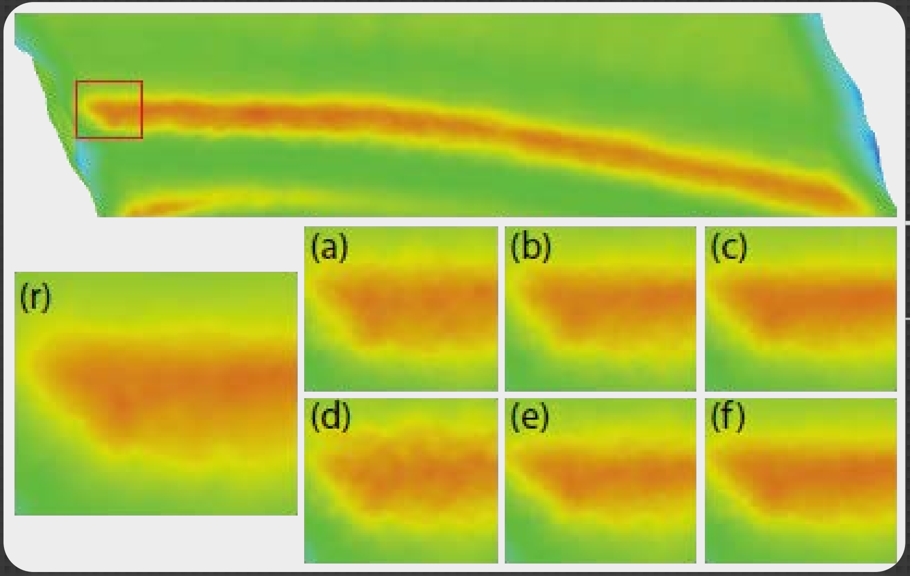
Monte Carlo Integration

$$\langle I\rangle(\mathbf{p}) = \frac{A'(\mathbf{s})}{N}\sum_{i=1}^{N} g(P(\mathbf{t}_i))$$



Adaptive Sampling

- Smooth surface areas converge faster than areas with high variance
- We use simplified two-step adaptive sampling



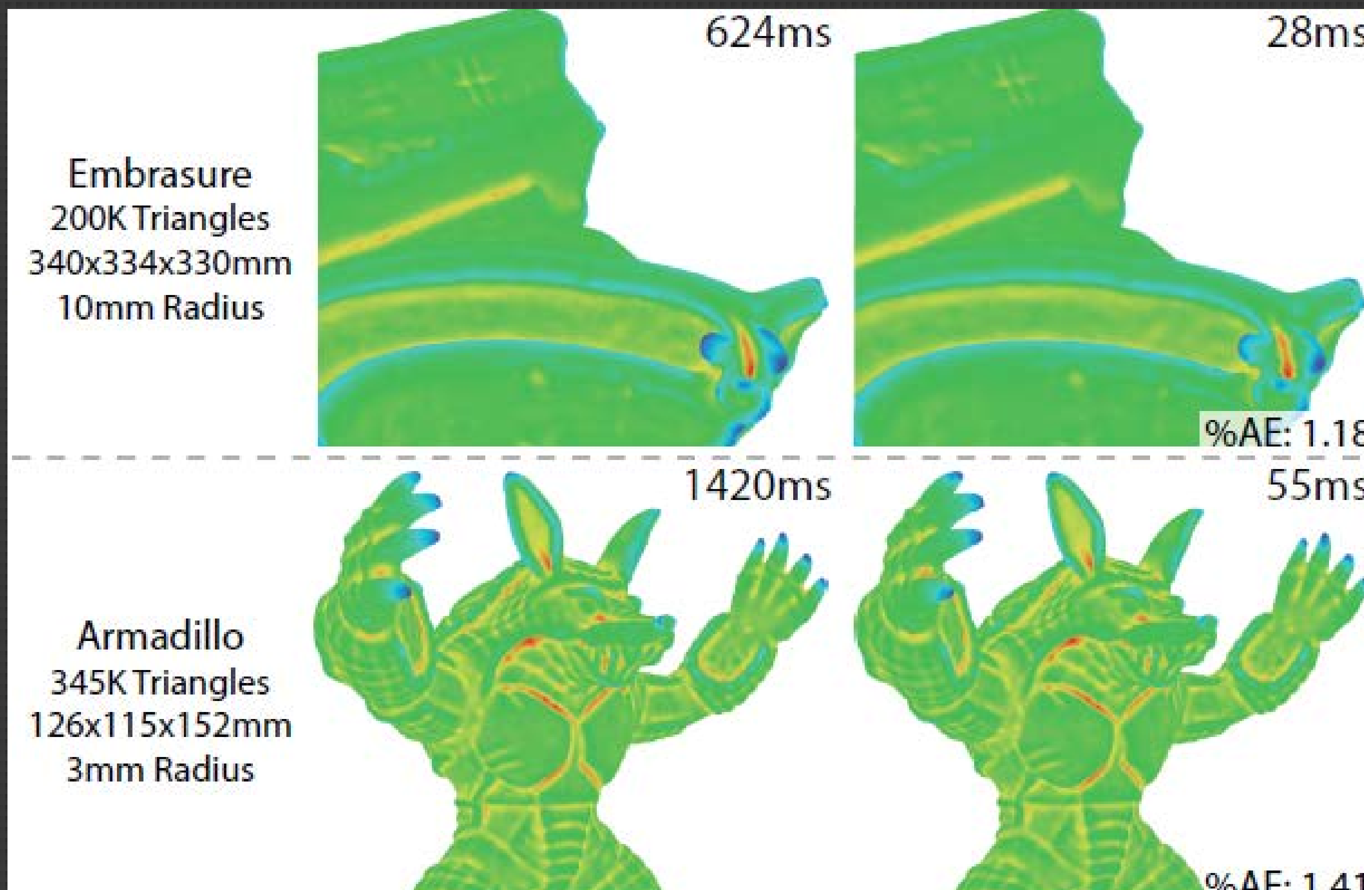
Samples	Full			Adaptive		
		Time	% AE		Time	% AE
64	(a)	17.57ms	1.172	(d)	15.94ms	1.331
100	(b)	22.17ms	1.035	(e)	19.54ms	1.110
256	(c)	50.54ms	1.005	(f)	41.44ms	1.007

Results

- Implemented Geometric Features
  - Mean Curvature
  - Local Bending Energy
  - Normalized Sphere Volume
  - Shape Index
- Comparison with multi-core CPU object space approach using Half-Edge data structure

Results (Mean Curvature)

Reference **Our Method**

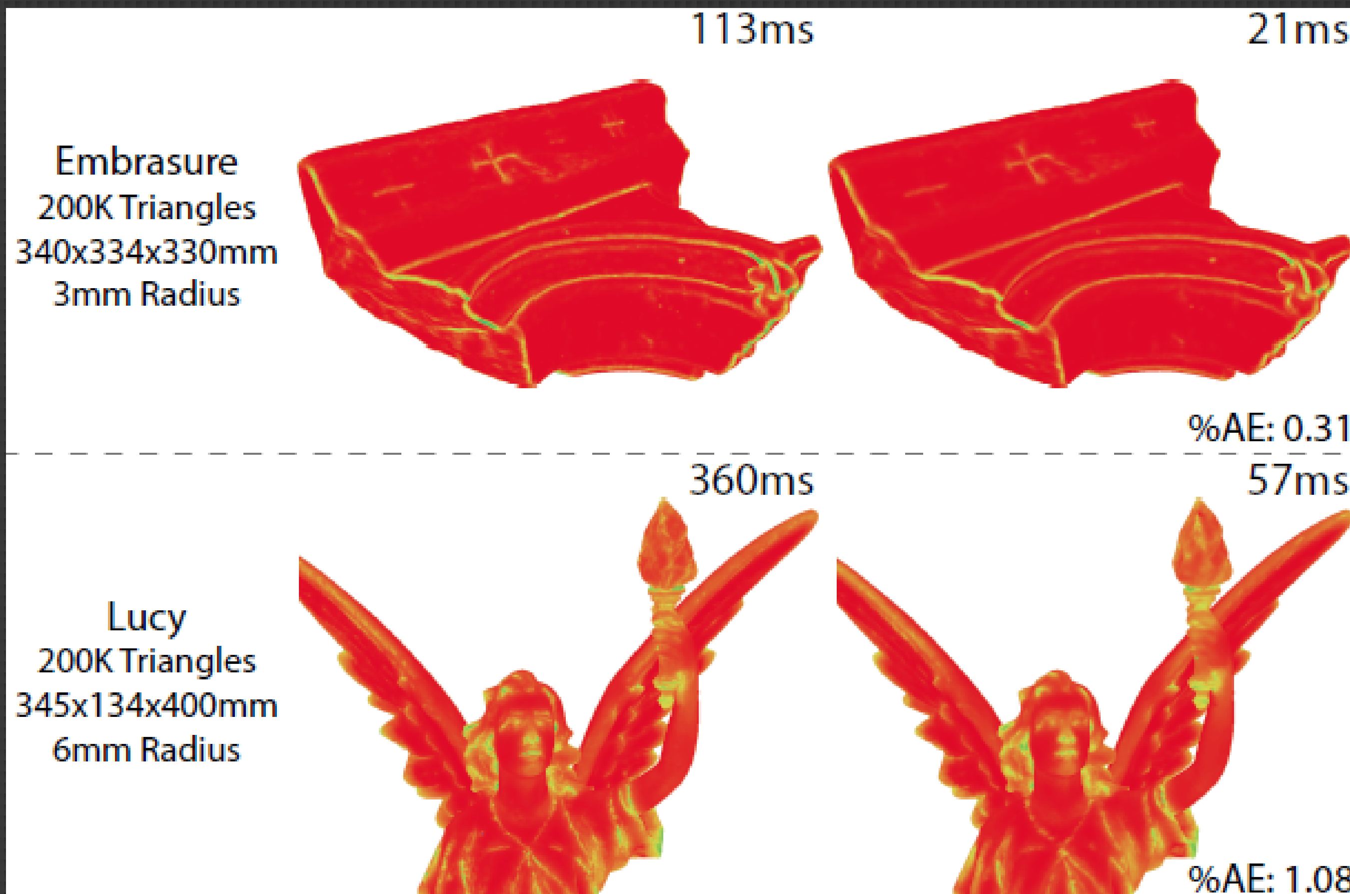


~22x

~25x

Results (Local Bending Energy)

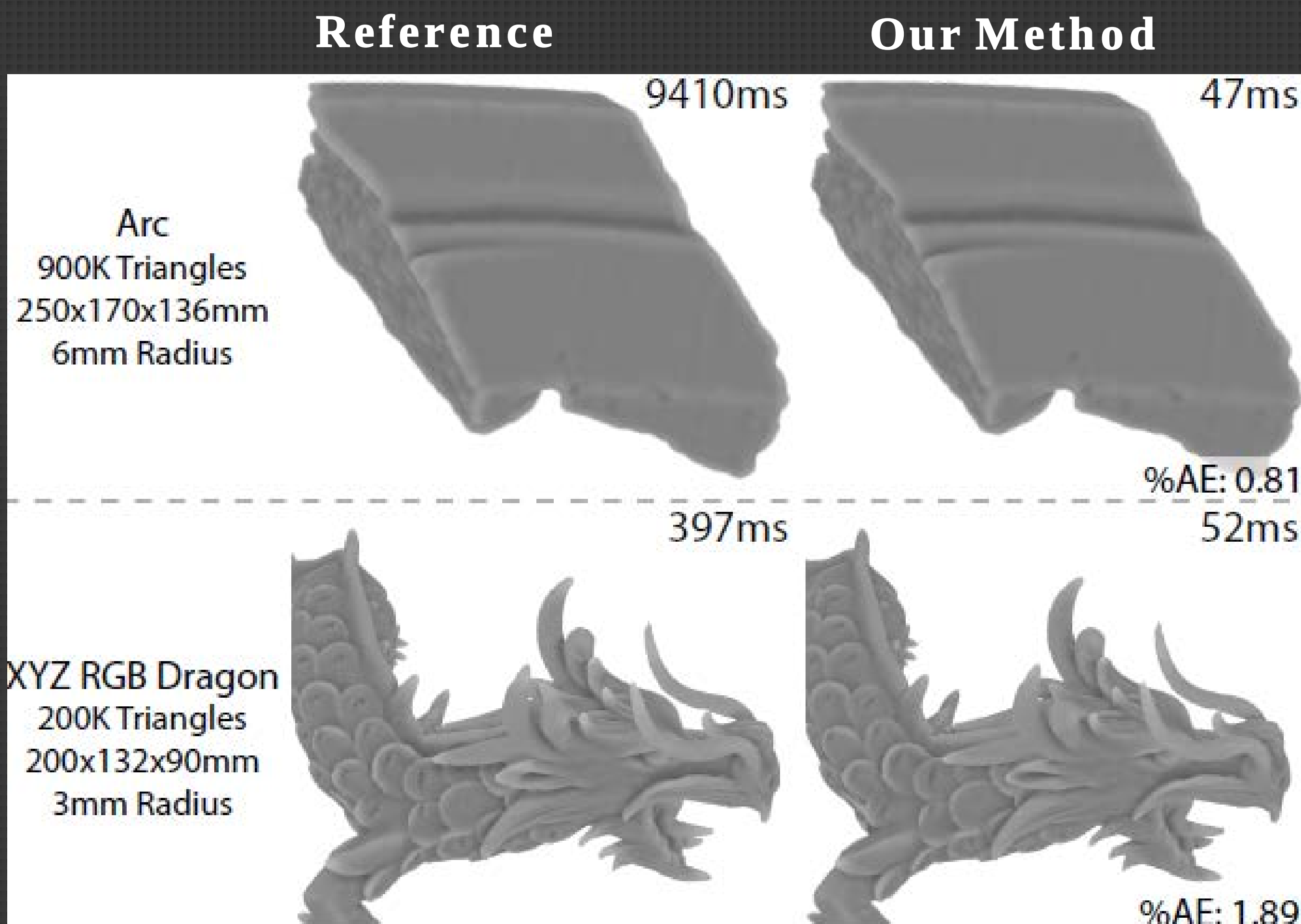
Reference **Our Method**



~5x

~6x

Results (Sphere Volume)

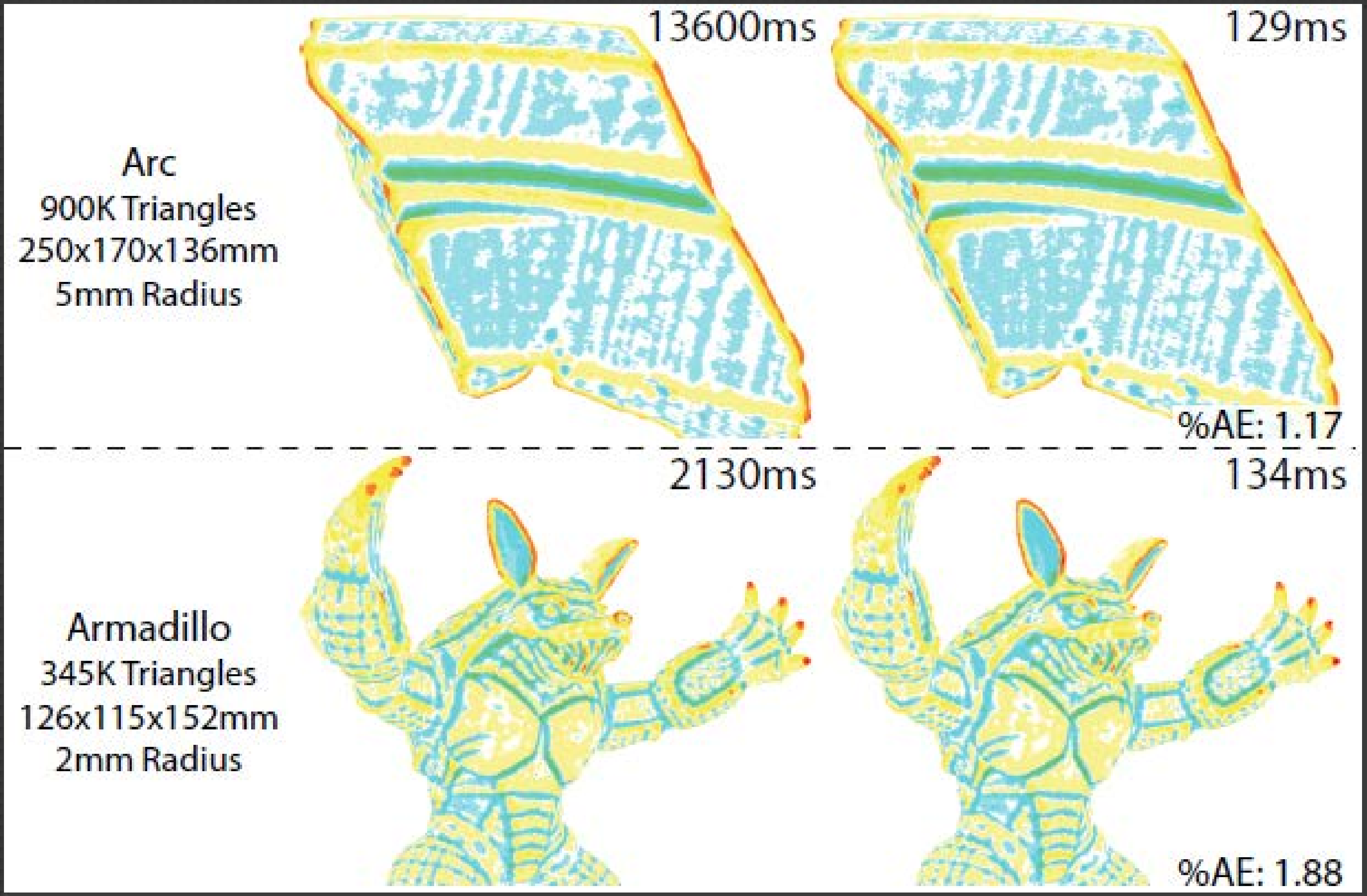


~200x

~8x

Results (Shape Index)

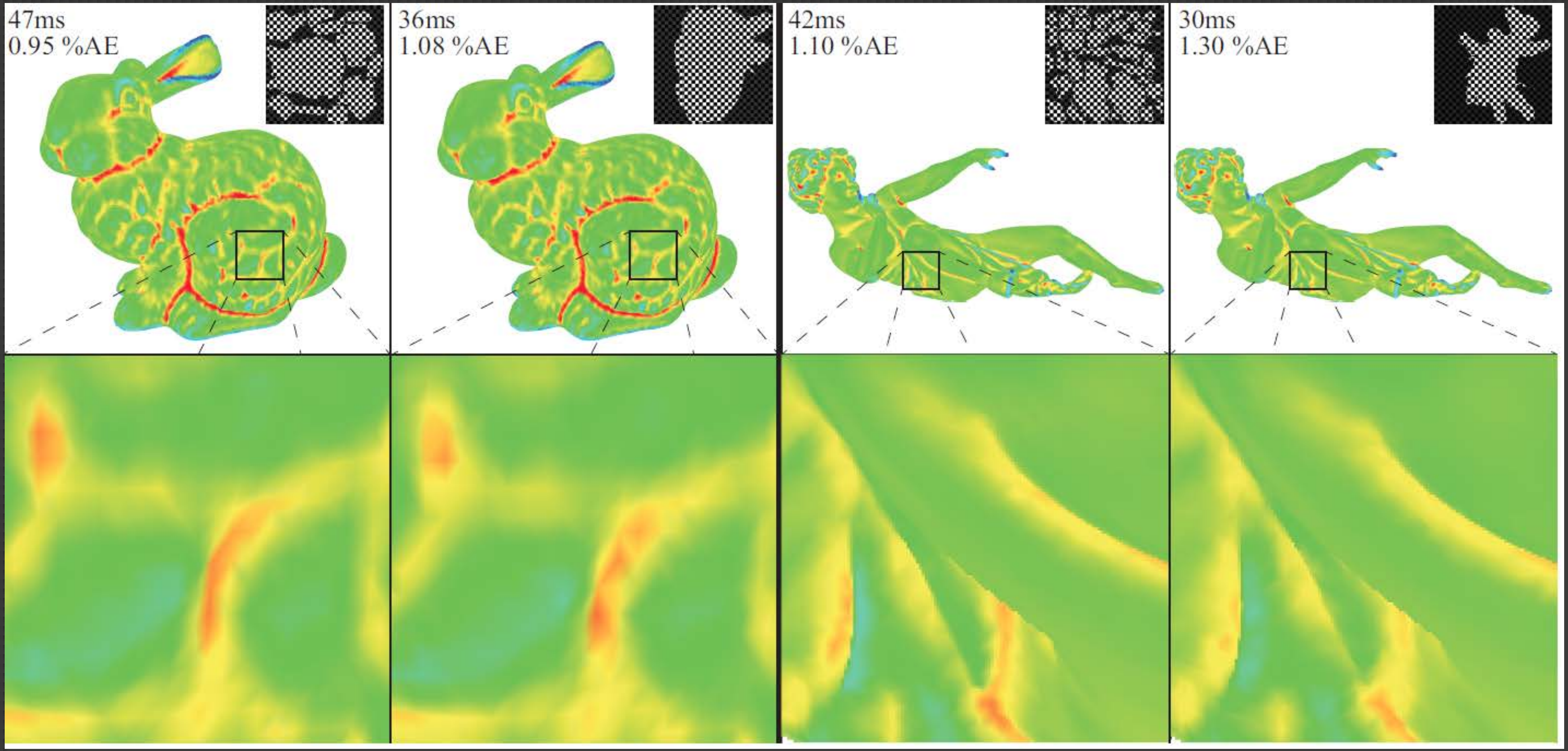
Reference **Our Method**



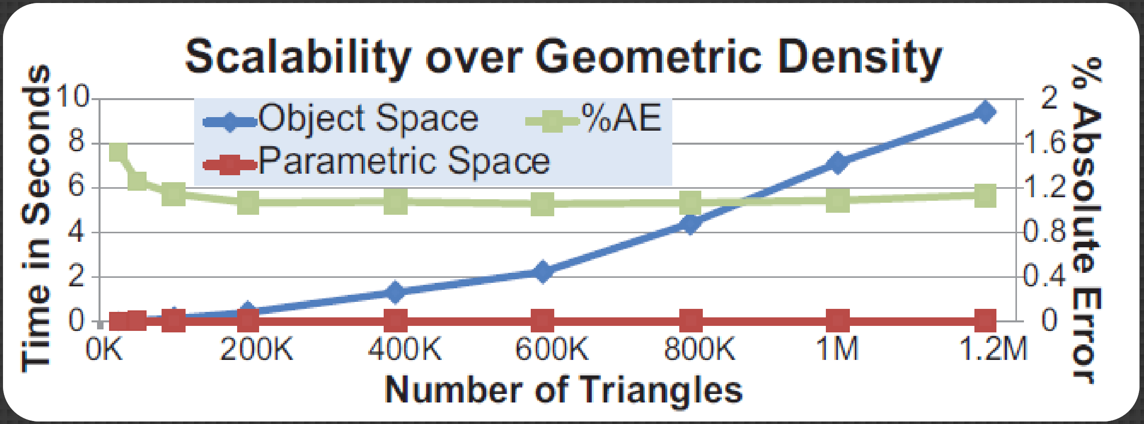
~105x

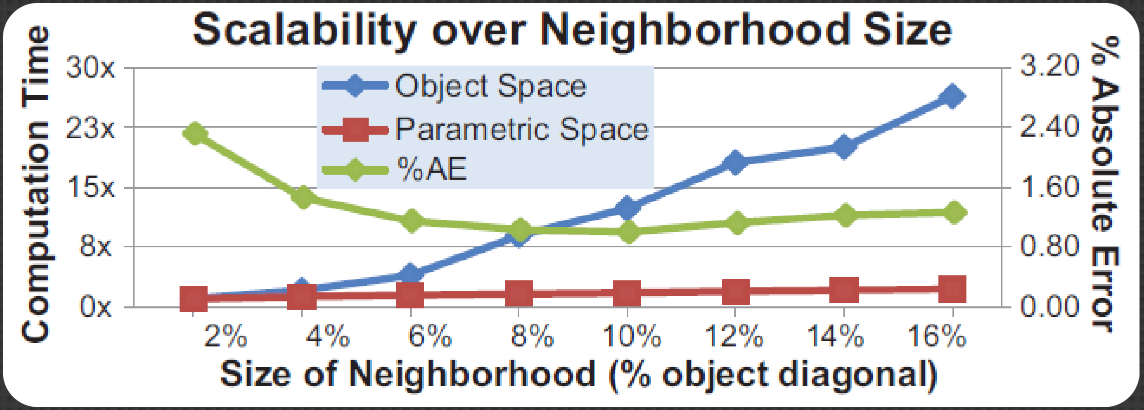
~16x

Results (Different Parameterizations)



Results (Scalability)





Thank you!

- Questions ?
- More info:
  - http://presious.eu
  - http://graphics.cs.aueb.gr