Hierarchical Computation of Conformal Spherical Embeddings

Takashi Kanai

Keio University SFC, Japan

Mathematical Methods for Curves and Surfaces 2004

Tromso, Norway

1-6 July, 2004

Outline

- Background, related work, contribution
- Conformal spherical parameterization [Gu et al. 2004]
- Hierarchical approach
- Results and discussion
- Conclusion and future work

Parameterization

... maps (a part of) a mesh to a simpler primitive

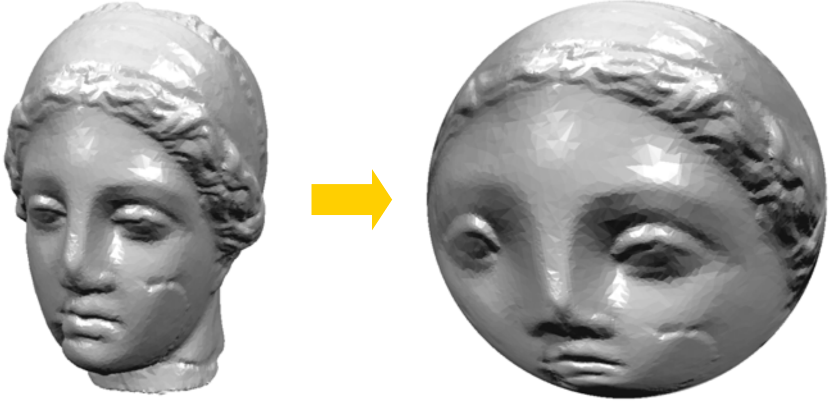
(plane, sphere, cylinder, octahedron, ...)

- Fundamental technique of DGP
- Used for Applications such as texture mapping, remeshing, morphing, surface reconstruction etc.

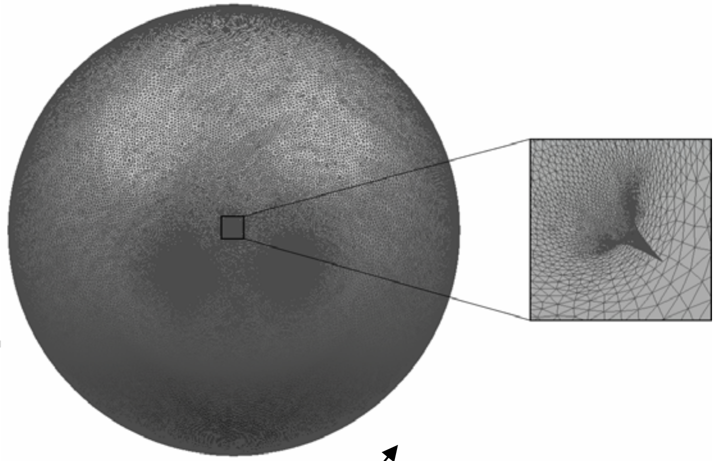
Spherical Parameterization

... maps a genus zero mesh to a sphere

- Consistent for a whole region of a mesh
  - Can perform some geometric processing applications easily (ex. remeshing, morphing)
  - need not to consider about the boundary



Related Work



[Kent et al., SIGGRAPH 92]

... can only apply for a star-shape object

[Alexa, Vis. Comp. 2000]

... simple and fast, but low-quality and flipping in some cases

[Praun and Hoppe, SIGGRAPH 2003]

... stretch-minimizing spherical parameterization

[Gotsman et al. SIGGRAPH 2003]

... generalization of Tutte's mapping

[Gu et al. IEEE TMI 2004]

... simple solution for conformal mapping

Our Contribution

- Hierarchical computation of conformal spherical parameterization
  - Extension to [Gu et al. 2004]
  - Keeps conformity
  - Robust and fast
  - (User-specified) parameter-independent

    ... free to try-and-error!

Conformal mapping

A mapping $f : M \mapsto S^2$

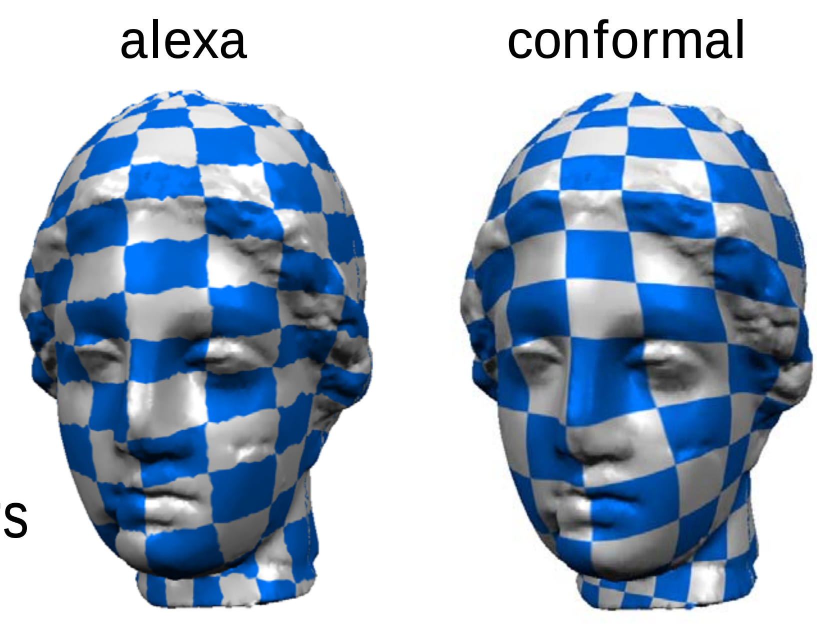
M : mesh S^2 : sphere

f is *conformal* if

$$\mathbf{I}^M = \mu(u,v)\mathbf{I}^{S^2}$$

$\mathbf{I}^M, \mathbf{I}^{S^2}$:
The first fundamental form

$\mu(u,v)$:
A scalar function for parameters $(u,v) \in S^2$



An Approach of [Gu et al. 2004]

- Based on steepest decent method
- Two steps approach:
  - Tutte mapping
  - Conformal mapping
- Simple iterative procedure

Tutte mapping algorithm

1. Compute Gauss map: $N : M \mapsto S^2$
   (a set of vertex normals)
   ⟶ initial parameter value $x(v)$
2. For each vertex v
   derivative
   update: $x'(v) = x(v) + c_t \overline{Dx(v)} \delta x$

$$Dx(v) = \Delta x(v) - \left(\Delta x(v) \cdot x(v) \right) x(v)$$

$$\Delta x(v) = \sum_{e=(u,v)} \left(x(u) - x(v) \right)$$

c_t : Tutte parameter (user-specified)

Tutte mapping algorithm (cont'd)

3. Compute Tutte Energy

$$E = \sum_e \| x(u) - x(v) \|^2$$

If $|E - E_0| < \varepsilon$, terminate the algorithm.

else, return 2.

Conformal mapping algorithm

- The algorithm is almost the same with Tutte mapping
- Initial value: the result of Tutte mapping

$$x'(v) = x(v) + \underline{c_c} Dx(v)\delta x$$

c_c : Conformal parameter (user-specified)

$$\Delta x(v) = \sum_{e=(u,v)} \underline{(a^{\alpha}_{v,u} + a^{\beta}_{v,u})} \big(x(u) - x(v)\big)$$

Harmonic Energy:

$$E = \sum_{e} \underline{(a^{\alpha}_{v,u} + a^{\beta}_{v,u})} \| x(u) - x(v) \|^2$$

Discussion: Gu et al's approach

User-specified parameters c_t, c_c

- have to be set to appropriate values

  ... difficult

  - too small c_t, c_c ... slow iteration
  - too large c_t, c_c ... computation failure (not embedding)

- depends on mesh geometry

→ Try-and-error

Hierarchical Approach

Similar to [Sander et al. 2002, Ray and Levy 2003, Praun and Hoppe 2003]

- Use Progressive Mesh [Hoppe 96]
  - Coarse-to-fine (multi-level) strategy

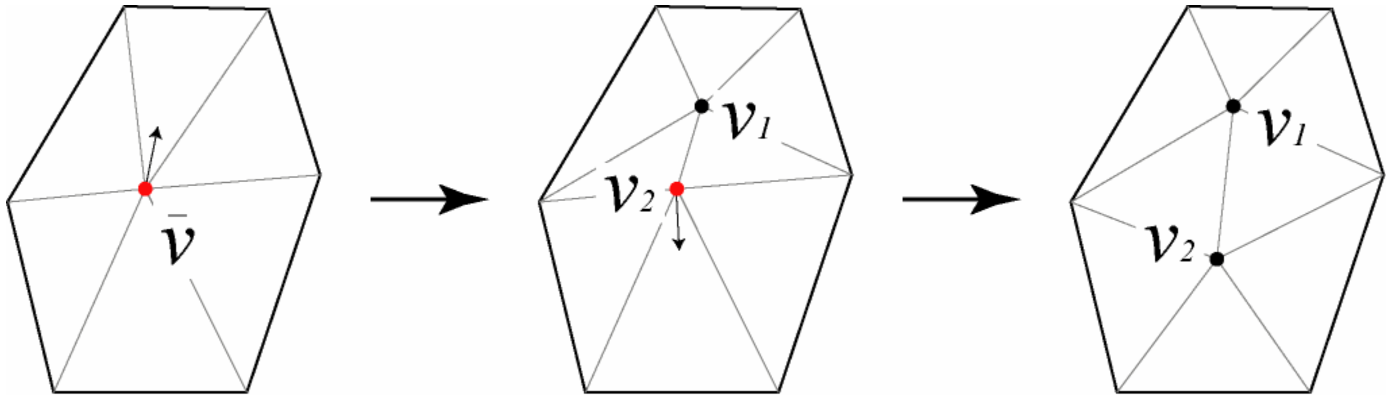
    Results in one level are used as initial guess in finer level
- Global Optimization
  - Computed in each level
  - Based on using priority queue

Computing Initial Value

- Simplify an original mesh to create a progressive mesh
- Start from a coarse mesh (roughly 100-1000 vertices)
- Use [Alexa 2000] to compute a spherical embedding of a coarse mesh
  - Most robust for a coarse mesh
  - Quality is not so important in this stage

Vertex-Based Optimization

- Use vertex split operation to increase vertices of a mesh
- Apply vertex-based optimization for each of two newly-created vertices
- the number of mesh vertices in each level is multiplied by a constant factor (eg. 2)
  (200, 400, 800, 1600 ...)



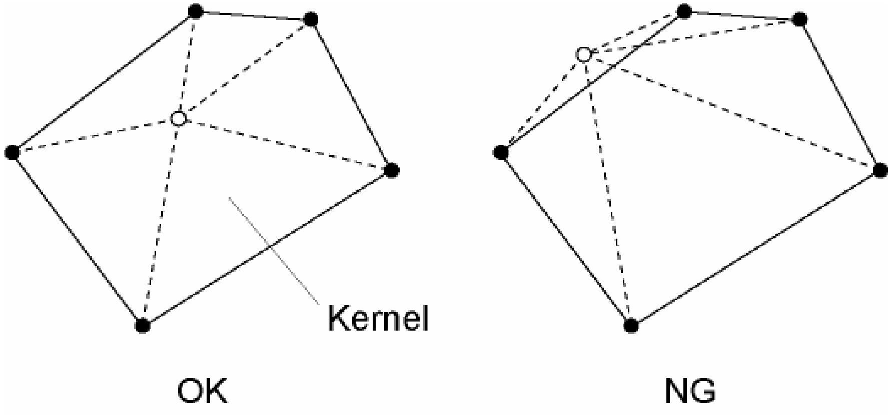
Vertex-Based Optimization (Cont'd)

Apply Gu et al's approach for a vertex

- Initial guess: a parameter of its parent's
- Update parameter … the same formula as Gu et al.'s approach
- Optimization terminates if $|E - E_0| < \varepsilon$

  E: Tutte (or Harmonic) Energy defined for neighbor vertices

- Check whether a new parameter is inside a *kernel*

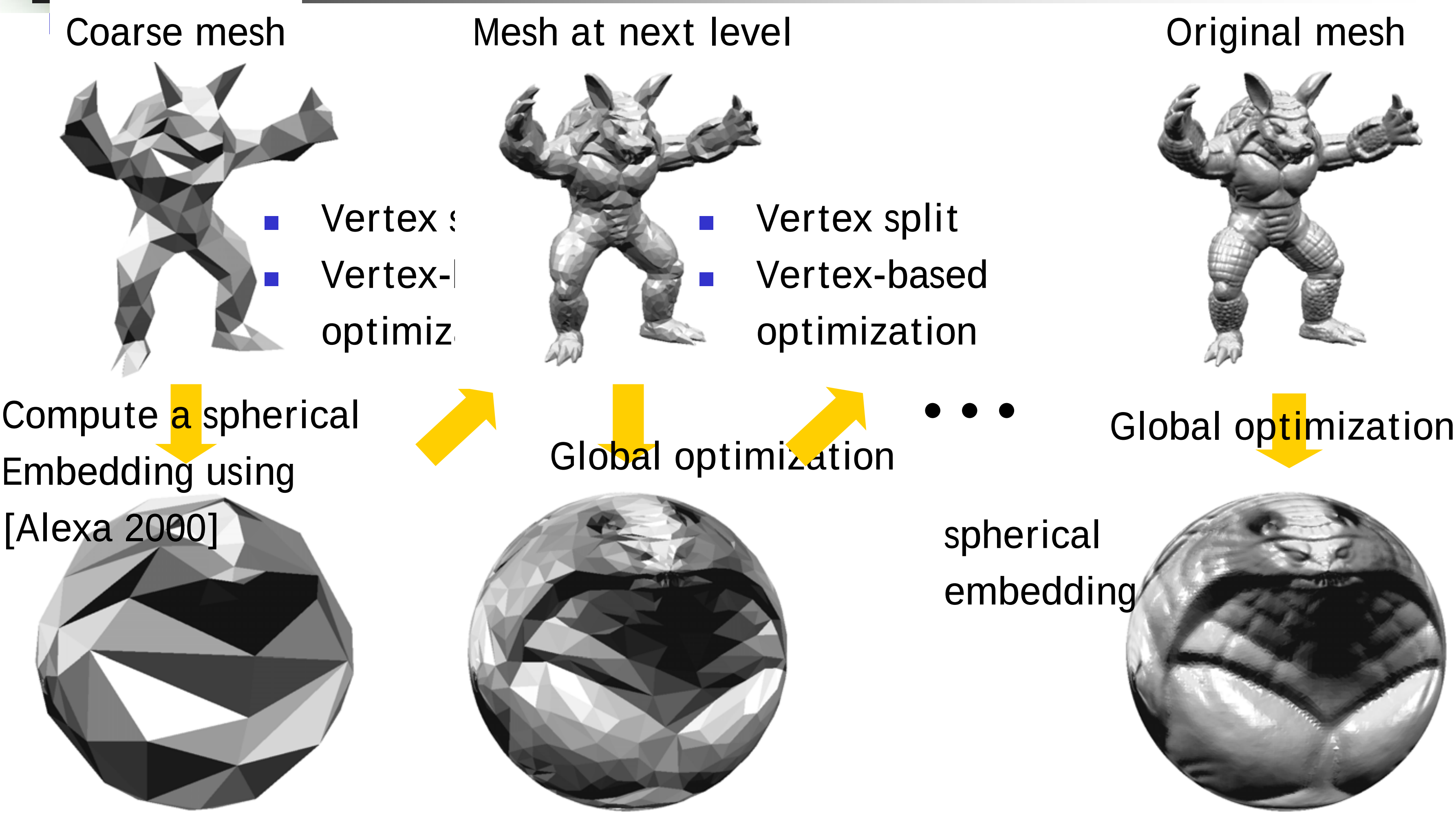


Global Optimization

1. Compute $dE = E(v)\text{-}E_0(v)$ for each vertex
2. Store dE to priority queue as a key
3. Apply delete min. (update vertex)
4. Update dE for neighbor vertices
5. Optimization terminates if $dE < \varepsilon$.

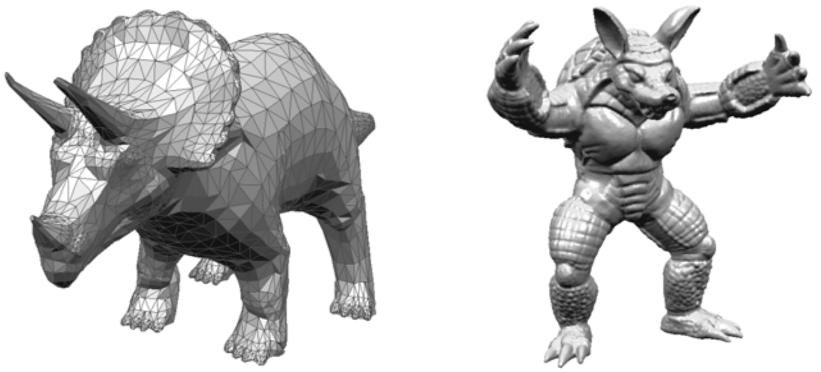
   Else, return 3.

Algorithm Overview



Experiments

- Two models
- Gu et al.'s approach
  - Different c_t and c_c
- Our approach
  - Hierarchical solution: yes *or* no
  - Initial solution: original *or* progressive
  - Global optimization: original *or* pri. queue



2,832 vertices   172,974 vertices

Results #1: Triceratops

Gu et al.'s approach

	coeffs.		embed.	time (sec.)		
	c_t	c_c		tutte	conf.	total
(1)	5.0×10^{-2}	5.0×10^{-3}	×	-	-	-
(2)	3.0×10^{-2}	3.0×10^{-4}	×	19.1	-	-
(3)	3.0×10^{-2}	2.0×10^{-4}	○	19.0	129.9	148.9
(4)	2.0×10^{-2}	2.0×10^{-4}	○	23.1	121.3	144.4
(5)	1.0×10^{-2}	1.0×10^{-4}	○	40.8	155.1	195.9
(6)	5.0×10^{-3}	1.0×10^{-4}	×	73.6	161.1	234.7
(7)	1.0×10^{-3}	1.0×10^{-4}	×	235.4	171.7	407.1
(8)	1.0×10^{-4}	1.0×10^{-4}	×	1034.6	314.0	1348.6
(9)	1.0×10^{-4}	1.0×10^{-5}	×	961.1	403.1	1364.2

Results #1: Triceratops

Our approach

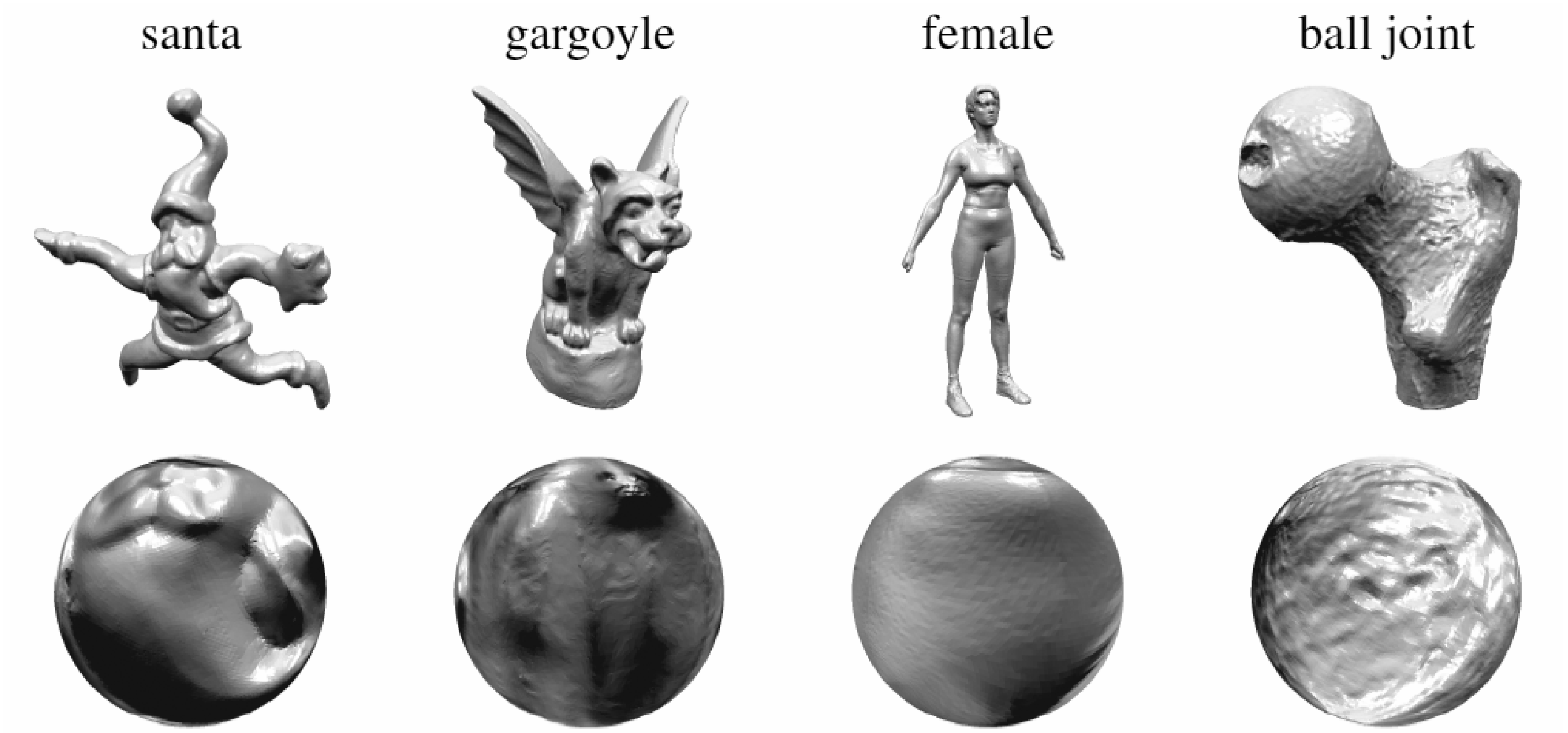
coeffs		hier.	init.	optim.	embed.	total time (sec.)
c_t	c_c					
2.0×10^{-2}	2.0×10^{-4}	no	prog.	orig.	○	372.1
2.0×10^{-2}	2.0×10^{-4}	yes	prog.	orig.	○	177.2
1.0×10^{-1}	1.0×10^{-2}	no	prog.	pri.	○	176.0
1.0×10^{-1}	1.0×10^{-2}	yes	prog.	pri.	○	57.3

- Narrow range of parameters (Gu et al.) ... needs try-and-error (c_t: 1.0x10^{-2}--3.0x10^{-2}, c_c: 1.0x10^{-4}--2.0x10^{-4})
- Computation time:
  144.4 s (Gu et al. with try-and-error parameter settings)
  vs.
  57.3 s (Ours without try-and-error)

Results #2: Armadillo

	type	c_t	c_c	hier.	init.	optim.	embed.	time (sec.)
(1)	Alexa	-	-	no	orig.	orig.	×	1,389.0
(2)	Gu et al.	2.0×10^{-1}	1.0×10^{-3}	no	orig.	orig.	○	5,518.7
(3)	Gu et al.	1.0×10^{-1}	1.0×10^{-3}	no	orig.	orig.	×	8,385.3
(4)	Hier. Conf.	1.0×10^{-1}	1.0×10^{-3}	no	prog.	orig.	○	20,370.5
(5)	Hier. Conf.	1.0×10^{-1}	1.0×10^{-3}	yes	prog.	orig.	○	3,861.1
(6)	Hier. Conf.	1.0×10^{-1}	1.0×10^{-1}	no	prog.	pq.	○	2,065.0
(7)	Hier. Conf.	1.0×10^{-1}	1.0×10^{-1}	yes	prog.	pq.	○	830.1

Other Results



	pai	horse	venus	bunny	santa	gargoyle	female	ball joint
#vertices	12,344	19,851	33,591	34,817	75,781	100,002	121,723	137,062
total time (sec.)	100.0	206.3	228.7	1,322.3	354.3	907.2	719.4	923.3

Conclusion and Future Work

Conclusion

- Hierarchical Computation of Conformal Spherical Embeddings
  - Robust (always holds embeddings)
  - Fast (roughly 3-10 times faster than Gu et al.'s original approach)
  - Parameter-independent (NO try-and-error)

Future Work

- Other types of parameterizations (area-preserving, mean value coordinates, …)
- Constrained parameterization
- Applications using spherical parameterization

Acknowledgements

- Prof. Xianfeng Gu
  (Univ. of Florida, USA)
- Mr. Alexandre Gouaillard
  (INSA de Lyon, France)

... for useful discussion

Thank you for your attention!