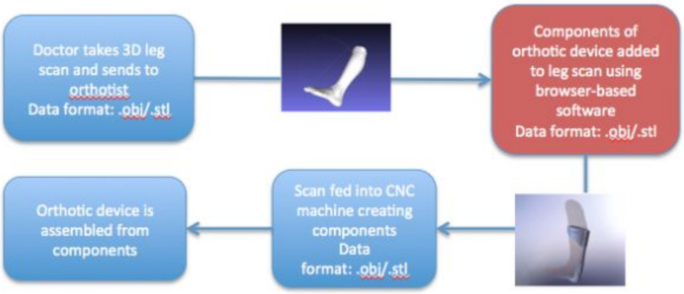
A Comparison of Mesh Simplification Algorithms

Nicole Ortega

Project Summary, Group 16

Browser Based Constructive Solid Geometry for Anatomical Models

- Orthotics for cerebral palsy patients
- Fusiform developed a process to reduce waste, reduce time and increase efficiency of orthotic design/fabrication
- Currently: ~10 hour process to create orthotic in SolidWorks
- Browser based software to add pre-designed orthotic components



Doctor takes 3D leg scan and sends to orthotist
Data format: .obj/.stl

Components of orthotic device added to leg scan using browser-based software
Data format: .obj/.stl

Scan fed into CNC machine creating components
Data format: .obj/.stl

Orthotic device is assembled from components

Paper

Cignoni, P., C. Montani, and R. Scopigno. "A Comparison of Mesh Simplification Algorithms." *Computers & Graphics* 22.1 (1998): 37-54. Web.

Goal of paper:

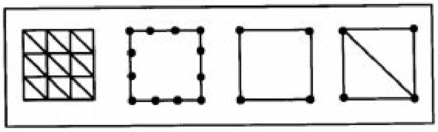
- Characterization of fundamental simplification methods
- Comparison of six simplification methods using three sample surfaces

Application to project:

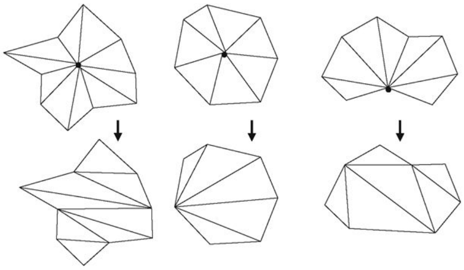
- Use method that will perform best on anatomical models for use in browser-based environment

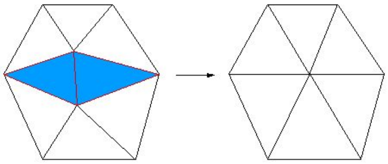
---

Simplification Methods



- Coplanar facets merging
- Controlled vertex/edge/face decimation
  - removal of vertices, collapsing edges/faces
- Re-tilling
- Energy function optimization
- Vertex clustering



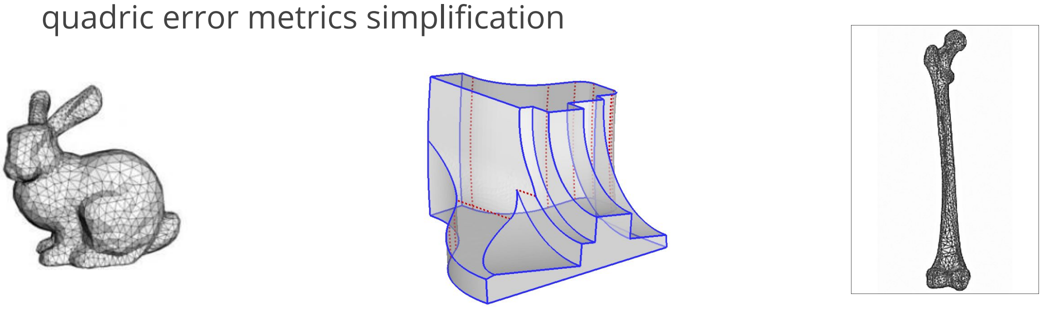


Characterization of Algorithms

- Method Characterization
    - Optimization goal - min size given error or vice-versa
    - Incremental simplification - iterations
    - Topological features - vertices, edges, faces, vertex pairs
- Approximation error
    - Local, global, other
    - Bounded (envelope)
- Preservation of mesh
    - Preservation of global mesh topology (mesh decimation but not vertex clustering)
    - Relocation of vertices
    - Preservation Solid/features edges or angles
- Multiresolution output
- Speed and availability

Evaluation of Simplification Code

- Meshes (3) - Bunny, Fandisk and Femur
- Simplification code (6) - Mesh decimation, simplification envelopes, multiresolution decimation, mesh optimization, progressive meshes, and quadric error metrics simplification



Evaluation of Simplification Code - Metro Tool

- Uniform and general tool to evaluate approximation precision
- Gives surface at different levels of detail (number of vertices and triangles)
- No knowledge of method used
- Finds the approximation error to evaluate differences
  - Definition: M_i and M_j are meshes. They are approximations of each other iff every point on M_i is within a distance *e* of some point of M_j and vice-versa
  - Samples original mesh and computes pt-to-surface distance with simplified mesh
- Output to compare likeness
  - $N_{vertices}$ $N_{triangles}$ E_{max} E_{avg} Time Edge Length Area Mem (kb)

Algorithm

Characterization

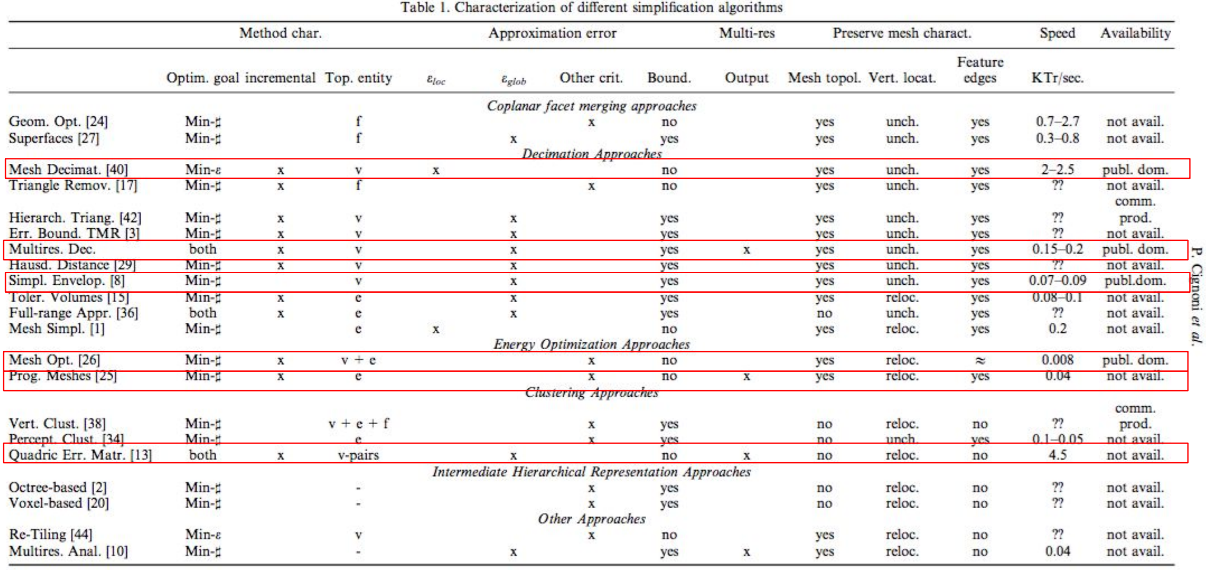


Table 1. Characterization of different simplification algorithms

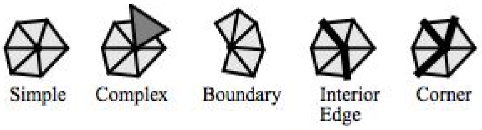
	Method char.			Approximation error				Multi-res	Preserve mesh charact.			Speed	Availability
	Optim. goal	incremental	Top. entity	ε_{loc}	ε_{glob}	Other crit.	Bound.	Output	Mesh topol.	Vert. locat.	Feature edges	KTr/sec.	
Coplanar facet merging approaches													
Geom. Opt. [24]	Min-♯		f			x	no		yes	unch.	yes	0.7–2.7	not avail.
Superfaces [27]	Min-♯		f		x		yes		yes	unch.	yes	0.3–0.8	not avail.
Decimation Approaches													
Mesh Decimat. [40]	Min-ε	x	v	x			no		yes	unch.	yes	2–2.5	publ. dom.
Triangle Remov. [17]	Min-♯	x	f			x	no		yes	unch.	yes	??	not avail.
Hierarch. Triang. [42]	Min-♯	x	v		x		yes		yes	unch.	yes	??	comm. prod.
Err. Bound. TMR [3]	Min-♯	x	v		x		yes		yes	unch.	yes	??	not avail.
Multires. Dec.	both	x	v		x		yes	x	yes	unch.	yes	0.15–0.2	publ. dom.
Hausd. Distance [29]	Min-♯	x	v		x		yes		yes	unch.	yes	??	not avail.
Simpl. Envelop. [8]	Min-♯		v		x		yes		yes	unch.	yes	0.07–0.09	publ.dom.
Toler. Volumes [15]	Min-♯	x	e		x		yes		yes	reloc.	yes	0.08–0.1	not avail.
Full-range Appr. [36]	both	x	e		x		yes		no	unch.	yes	??	not avail.
Mesh Simpl. [1]	Min-♯		e	x			no		yes	reloc.	yes	0.2	not avail.
Energy Optimization Approaches													
Mesh Opt. [26]	Min-♯	x	v + e			x	no		yes	reloc.	≈	0.008	publ. dom.
Prog. Meshes [25]	Min-♯	x	e			x	no	x	yes	reloc.	yes	0.04	not avail.
Clustering Approaches													
Vert. Clust. [38]	Min-♯		v + e + f			x	yes		no	reloc.	no	??	comm. prod.
Percept. Clust. [34]	Min-♯		e			x	yes		no	unch.	yes	0.1–0.05	not avail.
Quadric Err. Matr. [13]	both	x	v-pairs		x		no	x	no	reloc.	no	4.5	not avail.
Intermediate Hierarchical Representation Approaches													
Octree-based [2]	Min-♯		-			x	yes		no	reloc.	no	??	not avail.
Voxel-based [20]	Min-♯		-			x	yes		no	reloc.	no	??	not avail.
Other Approaches													
Re-Tiling [44]	Min-ε		v			x	no		yes	reloc.	no	??	not avail.
Multires. Anal. [10]	Min-♯		-		x		yes	x	yes	reloc.	no	0.04	not avail.

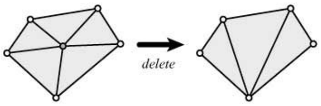
P. Cignoni *et al.*

Mesh Decimation [2]

Decimation - Min *e*, incremental, *v*, local e, not bounded *e*, preservation, no relocation *v*

1. Characterize the local vertex geometry and topology
    a. Complex vertices not deleted
2. Evaluate using decimation criteria
    a. Vertex distance to avg plane (simple) or vertex distance to edge (boundary/interior)
3. Delete vertex (& all triangles) if less than *d*
4. Triangulate the resulting hole
5. Repeat



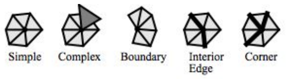


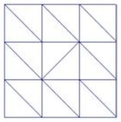
Multiresolution Decimation [3]

Decimation-Min #/e, incremental, v, global/bounded e, multires out, preservation, no relocation *v*

JADE (Just Another DEcimator) - enhanced decimation algorithm

1. Uses classification of vertex topology used by Shroeder
2. Evaluate using global approximation error criterion
3. Vertex selection to reduce accumulated error
    a. min local and accumulated global errors on top
4. Vertex deletion and triangulation using edge flipping
5. Repeat





Simplification Envelopes [4]

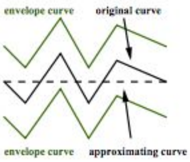
Decimation-Min #, v, global/bounded e, preservation, no relocation v

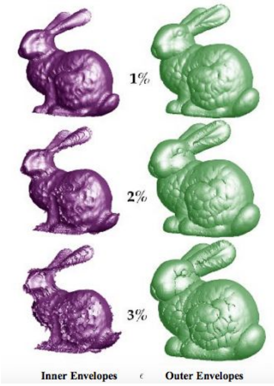
Surround mesh with two envelopes (inner & outer)

    User specific distance e from mesh

Perform simplification limited to envelope

    Preserve global topology

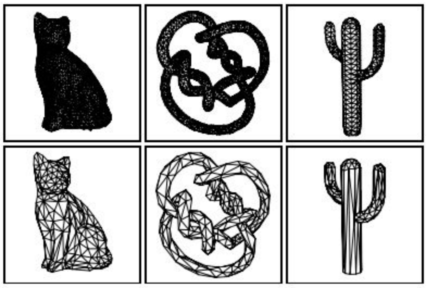




Mesh Optimization [5]

Energy Op - Min #, incremental, v/e, other e, no bounded e, preservation, relocation v

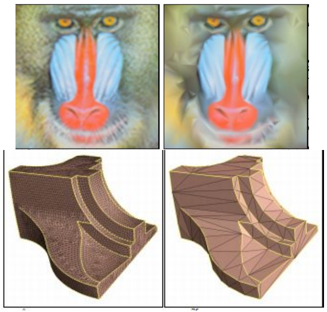
- Minimize energy function
- $E(K V) = E_{dist}(K, V) + E_{rep}(K) + E_{spring}(K, V)$
  - distance energy - sum of squared distance of the points from the mesh
  - representation energy - penalizes meshes with large number of vertices
  - spring energy - like placing on each edge of the mesh a spring of rest length 0 and tension k to regulate optimization to desired local minimum



Progressive Meshes [6]

Energy Op-Min #, incremental, e, other e, no bounded e, multires out, preservation, relocation

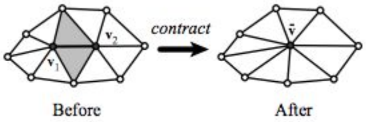
- Add more terms to the energy function by Hoppe
- $E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)$
  - preserve attributes of mesh
  - scalar energy - measures the accuracy of its scalar attributes
    - ie. diffuse color, normal, texture coordinates, and shading parameters
  - disc energy - measures the geometric accuracy of its discontinuity curves
    - sharp edges

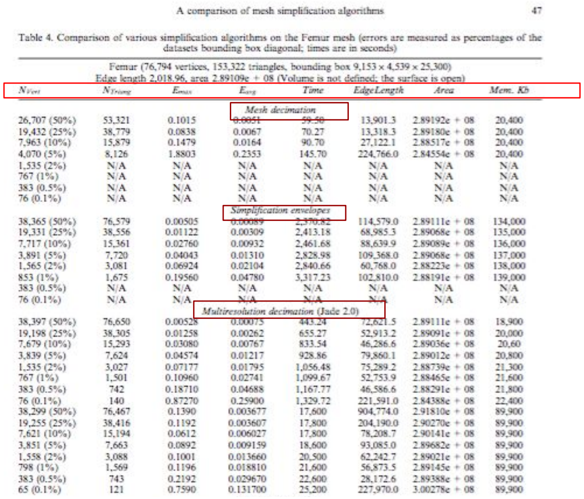


Quadric Error Metrics Simplification [7]

Clustering-Min #/e, incremental, v-pairs, global e, not bounded, multires out, no preservation, relocation v

1. Compute the Q matrices for all the initial vertices
   a. Q is the quadric error, sum of fundamental error quadrics
   b. quadric error: $\Delta v = v^T \mathbf{Q} v$
2. Select valid pairs
   a. $(v1, v2)$ is an edge, or $||v1 - v2|| < t$, where t is a threshold parameter
3. Compute the optimal contraction v′ for each valid pair
   a. error: $v'^T(Q_1 + Q_2)v'$ = cost of pair
4. Pairs ordered by cost → Min cost at top
5. Contract the pair of least cost
6. Update and repeat





A comparison of mesh simplification algorithms 47

Table 4. Comparison of various simplification algorithms on the Femur mesh (errors are measured as percentages of the datasets bounding box diagonal; times are in seconds)

Femur (76,794 vertices, 153,322 triangles, bounding box 9,153 × 4,539 × 25,300)
Edge length 2,018.96, area 2.89109e + 08 (Volume is not defined; the surface is open)

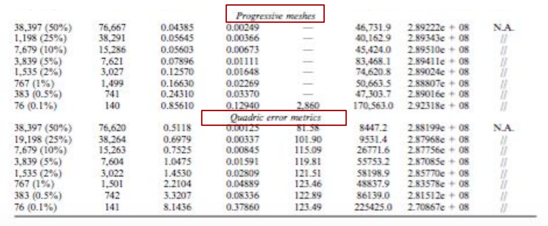
N_{Vert}	N_{Triang}	E_{max}	E_{avg}	*Time*	*EdgeLength*	*Area*	*Mem. Kb*
Mesh decimation							
26,707 (50%)	53,321	0.1015	0.0051	59.50	13,901.3	2.89192e + 08	20,400
19,432 (25%)	38,779	0.0838	0.0067	70.27	13,318.3	2.89180e + 08	20,400
7,963 (10%)	15,879	0.1479	0.0164	90.70	27,122.1	2.88517e + 08	20,400
4,070 (5%)	8,126	1.8803	0.2353	145.70	224,766.0	2.84554e + 08	20,400
1,535 (2%)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
767 (1%)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
383 (0.5%)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
76 (0.1%)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Simplification envelopes							
38,365 (50%)	76,579	0.00505	0.00089	2,370.82	114,579.0	2.89111e + 08	134,000
19,331 (25%)	38,556	0.01122	0.00309	2,413.18	68,985.3	2.89068e + 08	135,000
7,717 (10%)	15,361	0.02760	0.00932	2,461.68	88,639.9	2.89089e + 08	136,000
3,891 (5%)	7,720	0.04043	0.01310	2,828.98	109,368.0	2.89068e + 08	137,000
1,565 (2%)	3,081	0.06924	0.02104	2,840.66	60,768.0	2.88223e + 08	138,000
853 (1%)	1,675	0.19560	0.04780	3,317.23	102,810.0	2.88191e + 08	139,000
383 (0.5%)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
76 (0.1%)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Multiresolution decimation (Jade 2.0)							
38,397 (50%)	76,650	0.00528	0.00075	443.24	72,621.5	2.89111e + 08	18,900
19,198 (25%)	38,305	0.01258	0.00262	655.27	52,913.2	2.89091e + 08	20,000
7,679 (10%)	15,293	0.03080	0.00767	833.54	46,286.6	2.89036e + 08	20,60
3,839 (5%)	7,624	0.04574	0.01217	928.86	79,860.1	2.89012e + 08	20,800
1,535 (2%)	3,027	0.07177	0.01795	1,056.48	75,289.2	2.88739e + 08	21,300
767 (1%)	1,501	0.10960	0.02741	1,099.67	52,753.9	2.88465e + 08	21,600
383 (0.5%)	742	0.18710	0.04688	1,167.77	46,586.6	2.88291e + 08	21,800
76 (0.1%)	140	0.87270	0.25900	1,329.72	221,591.0	2.84388e + 08	22,400
38,299 (50%)	76,467	0.1390	0.003677	17,600	904,774.0	2.91810e + 08	89,900
19,255 (25%)	38,416	0.1192	0.003607	17,800	204,190.0	2.90270e + 08	89,900
7,621 (10%)	15,194	0.0612	0.006027	17,800	78,208.7	2.90141e + 08	89,900
3,851 (5%)	7,663	0.0892	0.009159	18,600	93,085.0	2.89682e + 08	89,900
1,558 (2%)	3,088	0.1001	0.013660	20,500	62,242.7	2.89021e + 08	89,900
798 (1%)	1,569	0.1196	0.018810	21,600	56,873.5	2.89145e + 08	89,900
383 (0.5%)	743	0.2192	0.029670	22,600	28,172.6	2.89388e + 08	89,900
65 (0.1%)	121	0.7590	0.131700	25,200	227,970.0	3.00278e + 08	89,900

Femur numerical output:

$N_{vertices}$	$N_{triangles}$
E_{max}	E_{avg}
Time	Edge Length
Area	Mem (kb)

Errors as percentage of dataset bounding box diagonals

TIme in seconds



Progressive meshes							
38,397 (50%)	76,667	0.04385	0.00249	—	46,731.9	2.89222e + 08	N.A.
1,198 (25%)	38,291	0.05645	0.00366	—	40,162.9	2.89343e + 08	//
7,679 (10%)	15,286	0.05603	0.00673	—	45,424.0	2.89510e + 08	//
3,839 (5%)	7,621	0.07896	0.01111	—	83,468.1	2.89411e + 08	//
1,535 (2%)	3,027	0.12570	0.01648	—	74,620.8	2.89024e + 08	//
767 (1%)	1,499	0.16630	0.02269	—	50,663.5	2.88807e + 08	//
383 (0.5%)	741	0.24310	0.03370	—	47,303.7	2.89016e + 08	//
76 (0.1%)	140	0.85610	0.12940	2,860	170,563.0	2.92318e + 08	//
Quadric error metrics							
38,397 (50%)	76,620	0.5118	0.00125	81.58	8447.2	2.88199e + 08	N.A.
19,198 (25%)	38,264	0.6979	0.00337	101.90	9531.4	2.87968e + 08	//
7,679 (10%)	15,263	0.7525	0.00845	115.09	26771.6	2.87756e + 08	//
3,839 (5%)	7,604	1.0475	0.01591	119.81	55753.2	2.87085e + 08	//
1,535 (2%)	3,022	1.4530	0.02809	121.51	58198.9	2.85770e + 08	//
767 (1%)	1,501	2.2104	0.04889	123.46	48837.9	2.83578e + 08	//
383 (0.5%)	742	3.3207	0.08336	122.89	86139.0	2.81512e + 08	//
76 (0.1%)	141	8.1436	0.37860	123.49	225425.0	2.70867e + 08	//

Overview | Methods/Char | Algorithms | Results | Analysis | Application | Pros/Cons

---

Findings

- Mesh decimation and simplification error- failed to reach high simplification rates on femur data
  - Both remove vertices in random order
  - partial solution: iterate multiple times
- Progressive Meshes and Mesh Optimization- best average error
- Simplification Envelope and Multires Decimation - best results when high accuracy needed
- Quadratic error
  - fast speed and small error for Fandisk
  - fast speed and large error for meshes with open boundaries (femur and bunny)
  - authors claim large error can be fixed - insert perpendicular planes at boundary edge and assigning large cost

Overview | Methods/Char | Algorithms | Results | Analysis | Application | Pros/Cons

Analysis - Femur

Femur mesh chosen

- Femur: 76k vertices, 153k triangles
- Leg scan: 47k vertices, 92k triangles
- Both Anatomical Models

Simplification needed to run on browser: ~10k vertices

*Speed needed to load on browser: <1min

Preservation of mesht: preserve shape features needed in this medical application

*Most important

Overview | Methods/Char | Algorithms | Results | Analysis | Application | Pros/Cons

---

Analysis - Femur

	E_{max}	E_{avg}	Time	Mem. kb
Mesh Decimation	5	6	1	1
Simplification Env.	1 (Best)	5	4	4
Multiresolution Dec.	2	3	3	2
Mesh Optimization	4	1	6	3
Progressive mesh	3	2	5	N/A
Quadric Error Metric	6 (Worst)	4	2	N/A

*Mesh Decimation and Simplification Envelope fail to reach high simplification. All compared at 10% simplification to keep simplification requirement in mind

** Nvert and Area not considered because too similar, Edge Length not considered because too variable

Overview | Methods/Char | Algorithms | Results | Analysis | Application | Pros/Cons

Applications

Things to consider: Preservation of Mesh, accuracy and speed

Preservation of Mesh -- not applicable to Quadric Error Metric code

Accuracy -- All methods had similar errors (slight differences)

Speed -- Very different times, most important quality

Top Pick: Mesh Decimation

Pros

Thorough

Good overview of methods

Detailed tables

Simplification codes cover range of methods

Cons

Too broad - needed clearer focus

Crowded/confusing tables - hard to read

Characterization explanation

- Why is it important? Only explained for preserving mesh topology

Methods of simplification codes not summarized

Summary table of findings

Lack of analysis of results

Citations

[1] Cignoni, P., C. Montani, and R. Scopigno. "A Comparison of Mesh Simplification Algorithms." *Computers & Graphics* 22.1 (1998): 37-54. Web.

[2] Schroeder, William J., Jonathan A. Zarge, and William E. Lorensen. "Decimation of Triangle Meshes." *ACM SIGGRAPH Computer Graphics SIGGRAPH Comput. Graph.* 26.2 (1992): 65-70. Web.

[3] Ciampalini, A., P. Cignoni, C. Montani, and R. Scopigno. "Multiresolution Decimation Based on Global Error." *The Visual Computer* 13.5 (1997): 228-46. Web.

[4] Cohen, Jonathan, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. "Simplification Envelopes." *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96* (1996). Web.

[5] Hoppe, Hugues, Tony Derose, Tom Duchamp, John Mcdonald, and Werner Stuetzle. "Mesh Optimization." *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93* (1993). Web.

[6]Hoppe, H., Progressive meshes. In ACM Computer Graphics Proc.. Annual Conference Series (Siggraph '96), 1996, pp. 99±108.

[7] Garland, Michael, and Paul S. Heckbert. "Surface Simplification Using Quadric Error Metrics." *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97* (1997). Web.

Questions?