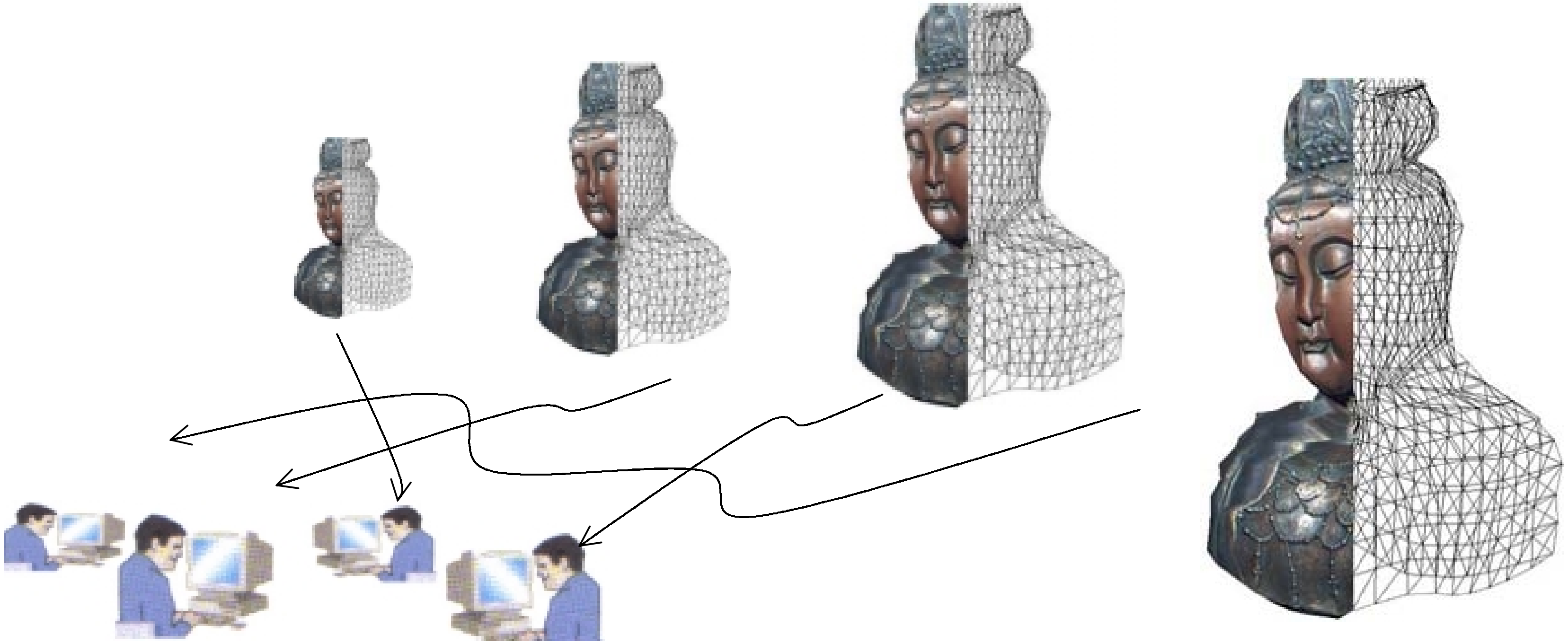
3D Model Simplification & Efficient Transmission

Irene Cheng

Department of Computing Science
University of Alberta



What are the goals ?

(1) Display physical objects in the virtual world, e.g.
machine parts, cultural artifacts, design models,
moviemaking, and video game industries.

(2) 3D instead of 2D images.

(3) Cater for different display devices.

(4) Adapt to different fluctuating network speeds.

(5) Provide best-effort visual quality based on given constraints, i.e. time, bandwidth.

(6) Associate high resolution real texture with mesh.

What are the challenges ?

(1) Display physical objects in the virtual world.
  - Surface data acquisition.

(2) 3D instead of 2D images.
  - External shape and surface characteristics.

(3) Cater for different display devices.
  - Screen resolution.

(4) Adapt to different network speeds.
  - Levels-of-detail (LOD).

(5) Provide best-effort visual quality.
  - Measuring criteria.

(6) Associate high resolution real texture with mesh.
  - Limited resources — Tradeoff.

Data Acquisition

- The Digital Michelangelo Project.

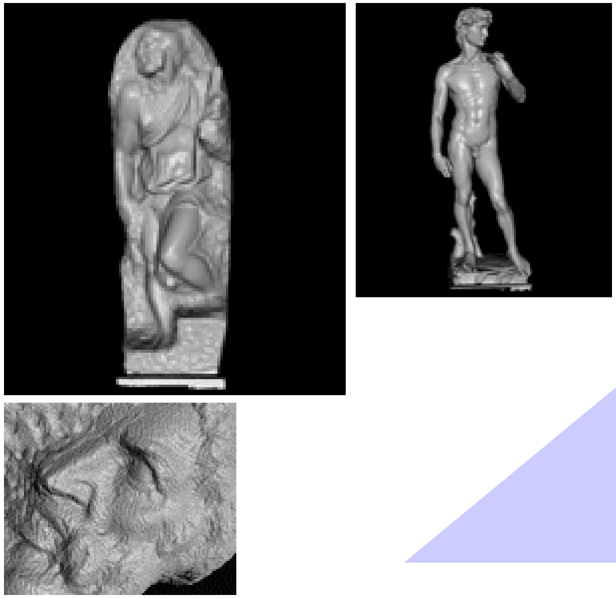
  http://graphics.stanford.edu/projects/mich/

  - 250 gigabytes data
  - 372 million polygons and 3.7 gigabytes for the statue of St. Matthew.
  - 1,000,000 polygons and 10 megabytes of the David (23’).
  - A team of 30 from Stanford U & U of Washington, led by Prof. Marc Levoy, spent 1998-99 in Italy scanning.
  - Study of scanning methods started in 1992.

Data Acquisition (continued)

http://graphics.stanford.edu/projects/mich/

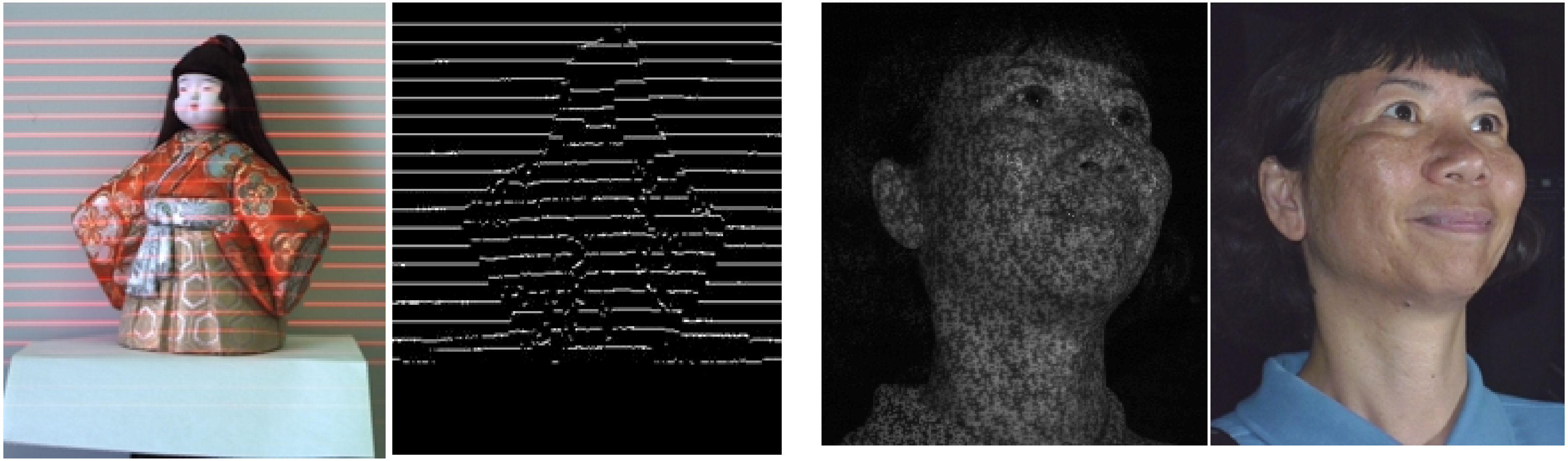
Cyberware



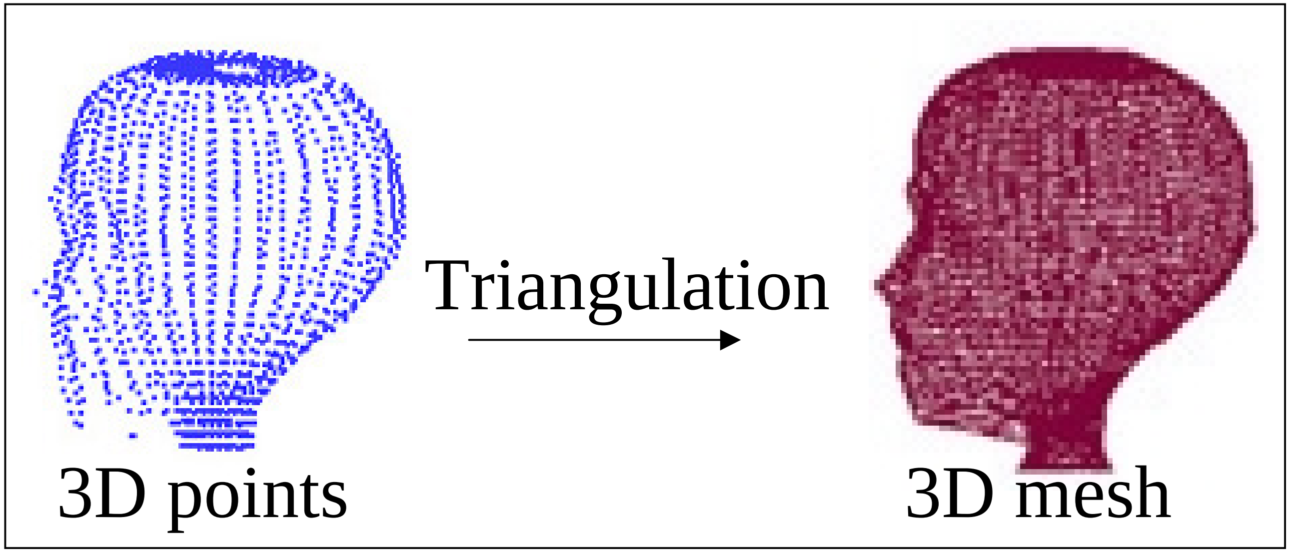
Faro

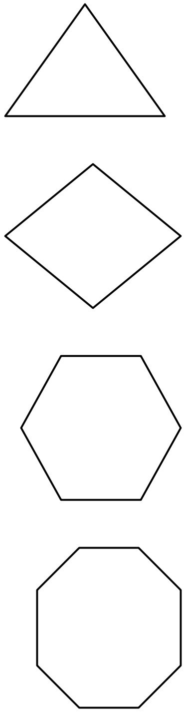
Data Acquisition (continued)

- Laser scan, structure light, pattern projection, multiple images, etc.
  - In contrast with IBR, e.g. light field.

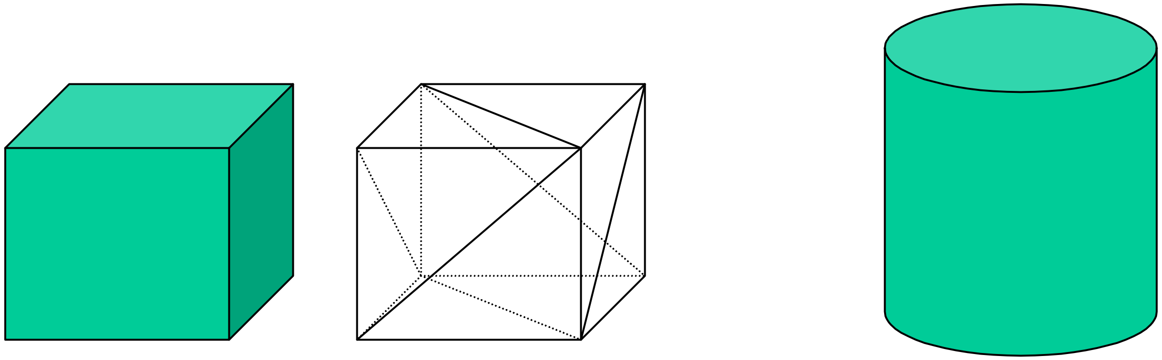


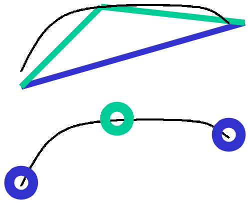
External Shape & 3D Geometry





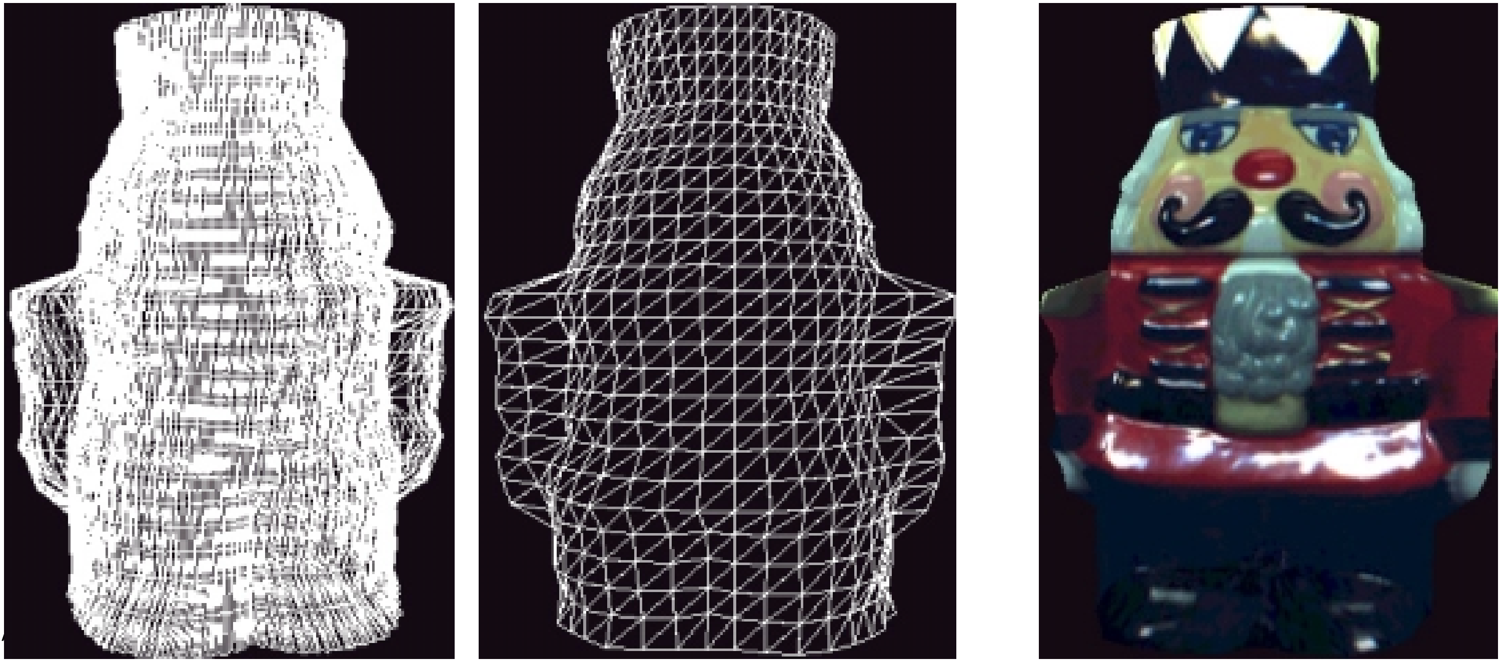
Feature point representation :





Feature point extraction & Simplification techniques

- Discrete LOD v.s. Continuous LOD
- View dependent v.s. View-independent
- Image-based v.s. Geometry-based
- Sub-sampling — regular mesh

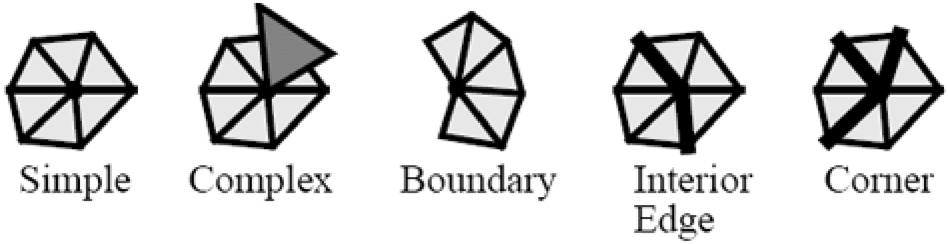


Feature point extraction & Mesh simplification techniques

- Decimation of triangle meshes (Schroeder et.al.1992)
  - Use local operation on geometry to reduce the # of ∆s.
  - Preserve the original topology.
  - Make multiple passes over all vertices.
  - A vertex & the associated ∆s are deleted if the specified decimation criteria are met.
  - The resulting hole is patched by local triangulation.
  - Terminate when the required # of ∆s is reduced.

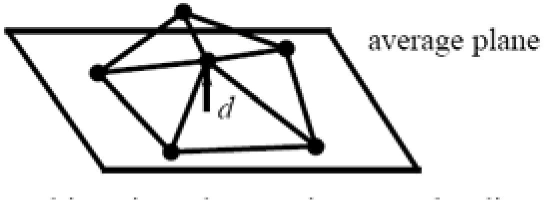
Decimation of triangle meshes (continued)

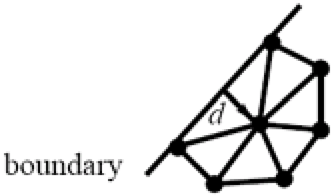
- Decimation criteria
  - Each vertex is assigned one of the following:
    - Simple (interior edge/(2 feature edges) and corner/...), complex, & boundary.
    - Complex vertex is not deleted.



Decimation of triangle meshes (continued)

- Decimation criteria
  - If a vertex is within the specified distance *d* to the average plane (of the surrounding Δs), it may be deleted.

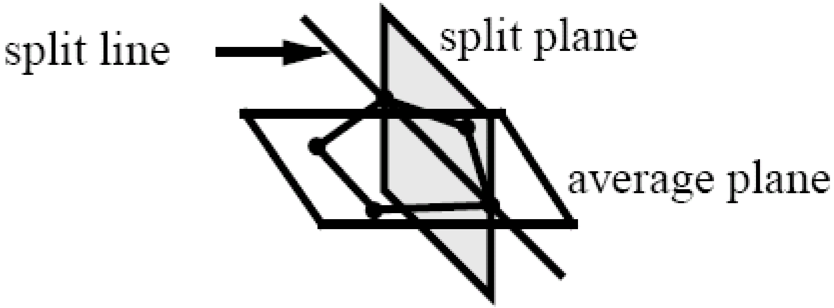




  - Boundary & interior edge vertices use the distance to edge criterion.
  - Relative small Δs with large feature angles, contributing little to surface property, are removed.

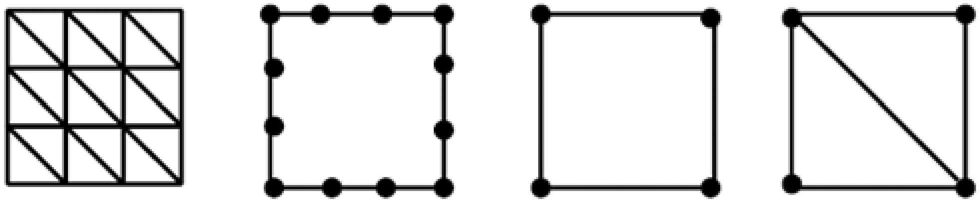
Decimation of triangle meshes (continued)

- Triangulation
  - Divide-and-conquer until only 3 vertices left.
  - The best splitting plane yields the max. aspect ratio.
    - The aspect ratio is the min. distance of the loop vertices to the split plane, divided by the length of the split line (constrained to > 0,1).

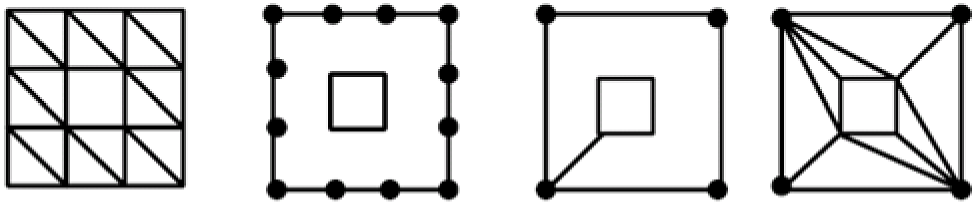


Feature point extraction & Mesh simplification techniques

- Geometric optimization (Hinker et.al.1993)
  - Merge coplanar and nearly coplanar Δs.

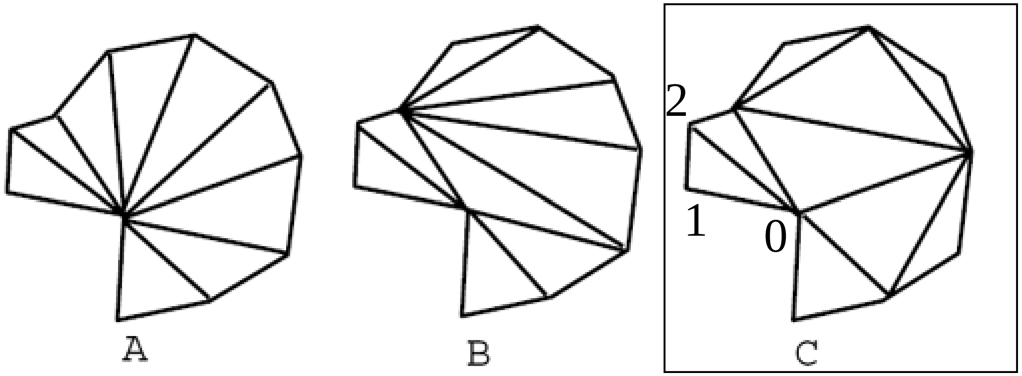


  - Start from a representative normal, add an adjacent Δ if the inter-normal angle is within a specified ε.
  - Replace the representative normal with the average normal.
  - Preserve holes.



Geometric optimization (continued)

- Triangulation
  - Traverse the Δ 0-1-2.
  - Form segment 0-2 if it does not intersect any other segments. Delete vertex 1.
  - Repeat the above for Δ 2-3-4.
  - If intersection occurs, the starting vertex is increased by one, i.e. 3-4-5.



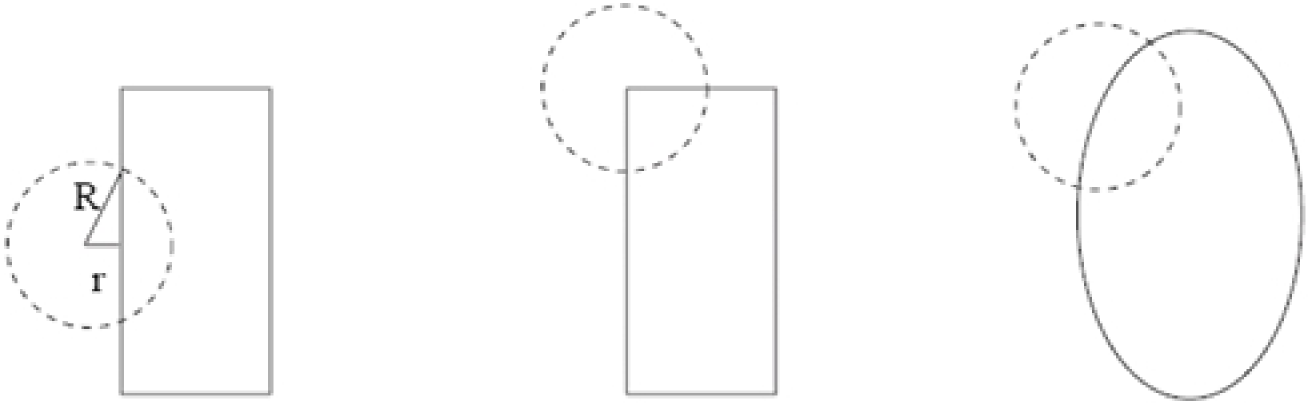
  - Can produce poorly proportioned Δs, but easy and fast to implement (c).

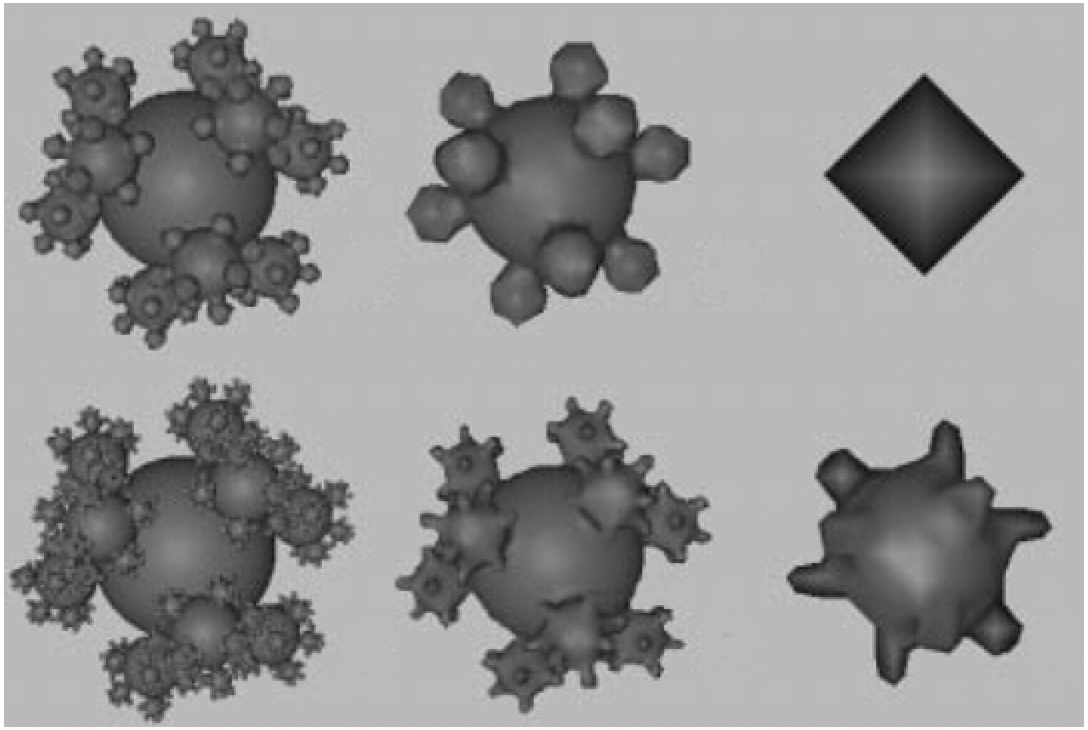
Feature point extraction & Mesh simplification techniques

- Voxel based object simplification (He et.al. 1995)
  - Gradual elimination of details exceeding a certain frequency.
  - Simplify the genus of objects; topology preservation could present efficient simplification.
  - Apply low-pass filtering $f(i, j, k)$ to remove detailed features, i.e. (i, j, k) is a grid point inside the filter.
  - Cannot output infinitely high frequencies, i.e.those generated by sharp edges. Thus works best for objects with no sharp discontinuities.

Voxel based object simplification (continued)

e.g. A hyber-cone filter with radius R and distance r from 3D surface.



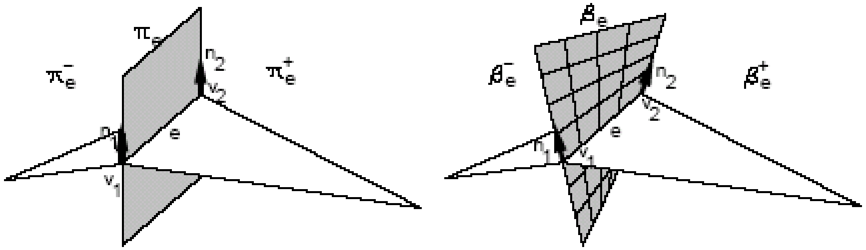


Feature point extraction & Mesh simplification techniques

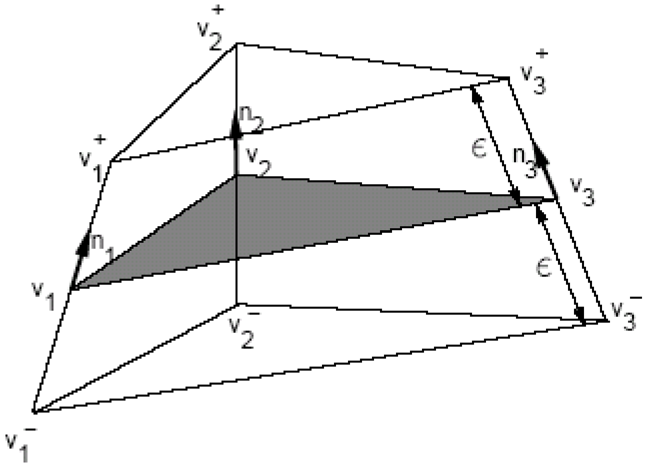
- Simplification Envelopes (Cohen et.al. 1996)
  - Generate hierarchy of LOD.
  - Guarantee an approximation is within ε (+/-) distance from the original model.
  - Preserve genus.
  - Prevent self-intersection.
  - Preserve sharp features.
  - Allow variation of approximation distance across different portions of a model.

Simplification Envelopes (continued)

- Envelope Computation



▨ Edge Half-spaces



$$\text{Coord}(v_i^+) = \text{Coord}(v_i) + \varepsilon\mathbf{n}(v_i) \text{ and } \mathbf{n}(v_i^+) = \mathbf{n}(v_i)$$

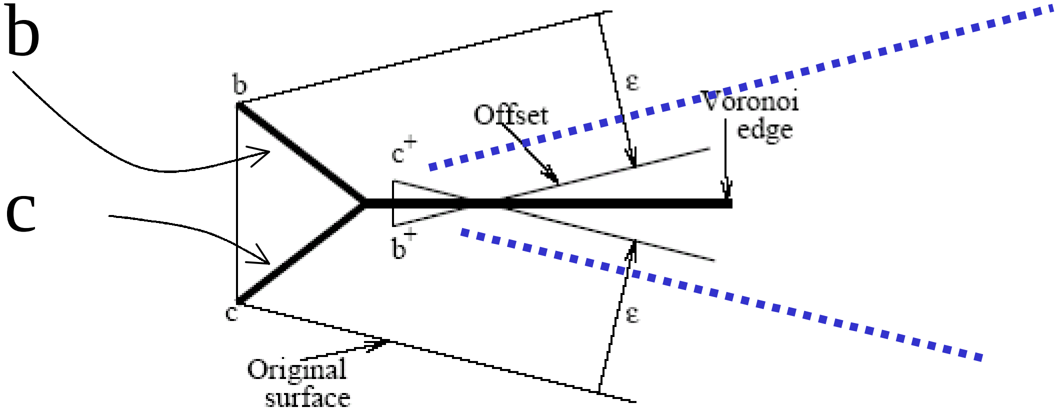
-ε can be similarly defined in the opposite direction.

Simplification Envelopes (continued)

- Avoid self-intersection

normal of b

normal of c

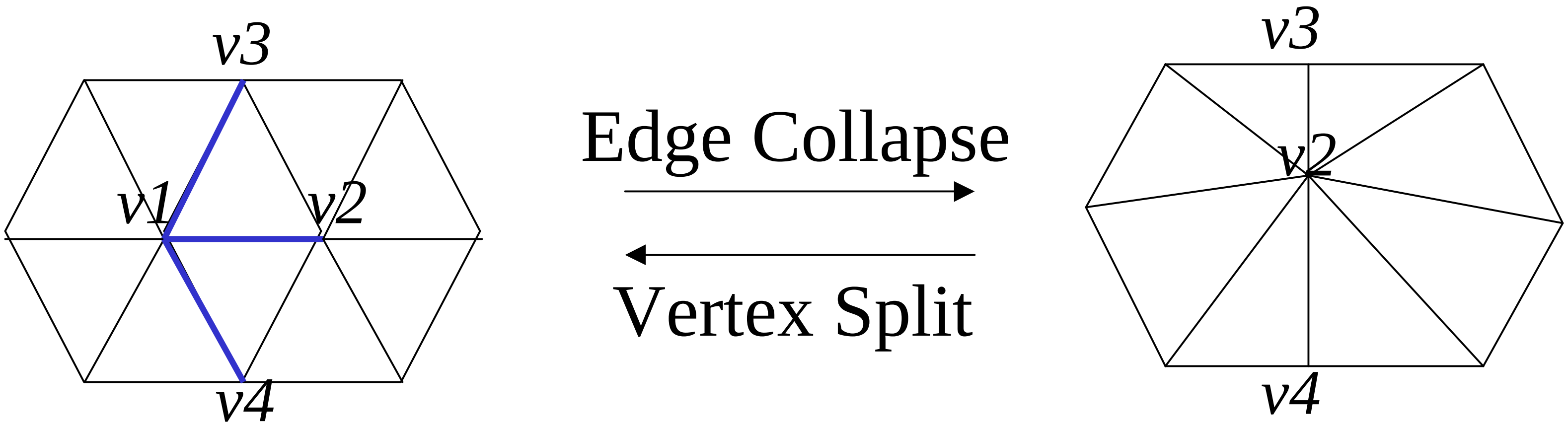


- Hole creation & hole filling.

Simplification Envelopes (continued)

Feature point extraction & Mesh simplification techniques

- Progressive meshes (Hoppe 1996)
  - Lossless CLOD.
  - Apply edge collapse (simplification) & vertex split (refinement).



Progressive meshes (continued)

- Energy Minimization
  - Estimate energy cost ΔE for each edge collapse transformation, and store in priority queue.
  - In each iteration, perform the transformation with lowest ΔE.
  - Recompute the priorities of edges in the neighborhood of this transformation.

Feature point extraction & Mesh simplification techniques

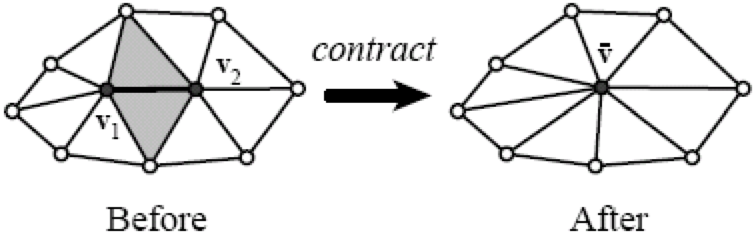
- Simplification using Quatric Error Metrics (Garland et.al. 1997)
  - Use iterative contractions of vertex pairs.
  - Maintain surface error approximations using quadric matrices.
  - Able to join unconnected regions of models.
  - Select the set of valid pairs at initialization time, based on the assumption that, in a good approximation, points do not move far from their original positions.

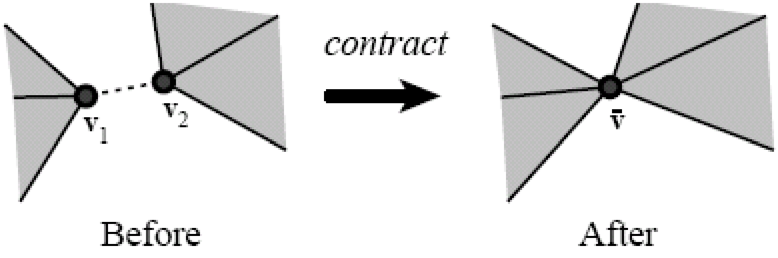
Simplification using Quatric Error Metrics
(continued)

- Cost/error of contraction Δv for (v1, v2) → v
  - Associate a cost with each vertex.
  - Compute an initial cost by accumulating the planes for the Δs which meet at that vertex. Note that the initial error estimate for each vertex is 0, because the vertex lies in the planes of all its incident Δs.
  - Define the error of the vertex w.r.t this set of planes as the sum of squared distances to the planes.

Simplification using Quatric Error Metrics
(continued)

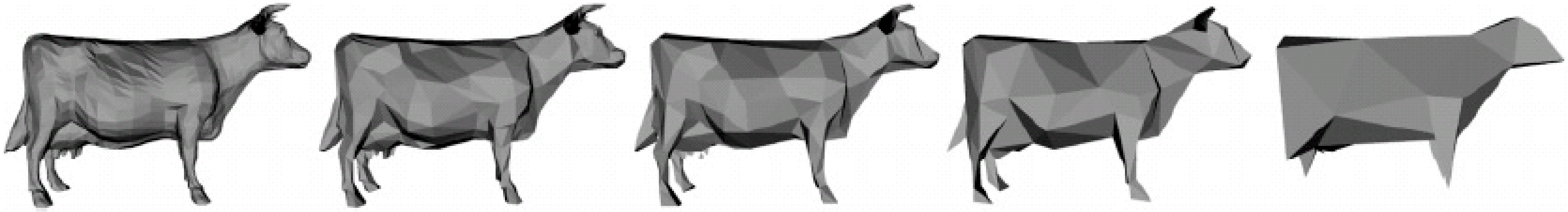
- Cost/Error of contraction Δv for $(v1, v2) \rightarrow v$
  - Select $v1$, $v2$ or $(v1 + v2)/2$ as position of v, depending on which one produces the lowest value of Δv.
  - $Qv = Q1 + Q2$.





Simplification using Quatric Error Metrics

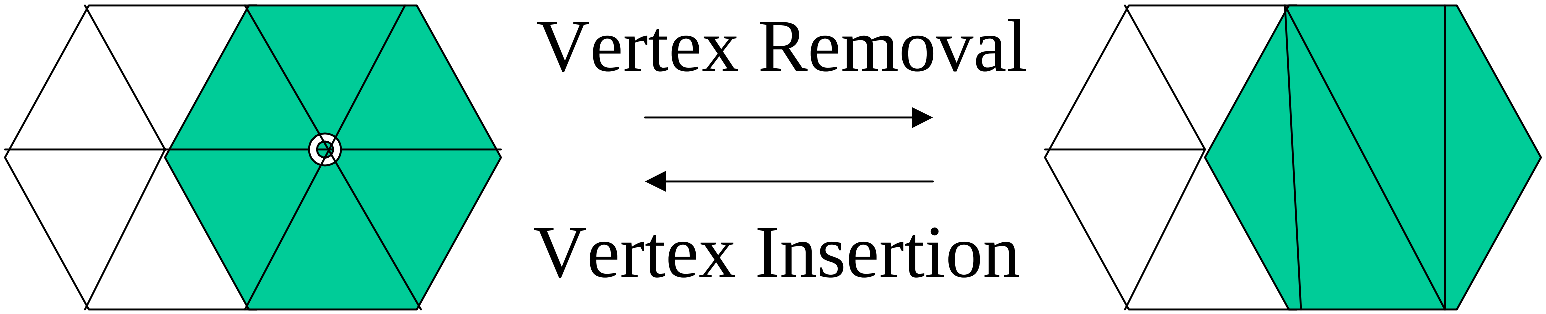
(continued)



Feature point extraction & Mesh simplification techniques

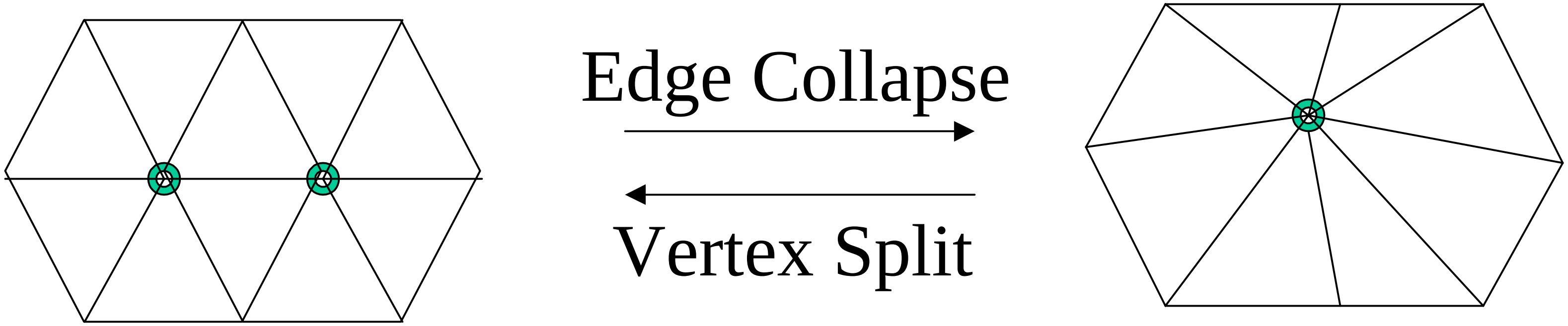
- Model simplification using vertex clustering (Low et.al. 1997)
    - Determine the closeness of the vertices.
    - Vertices are grouped together based on their proximity.
    - A new representative vertex is created to replace them.

Simplification & Refinement



Vertex Removal

Vertex Insertion



Edge Collapse

Vertex Split

Discussion — Pros & Cons

1. Vertex relocation
2. Continuous LOD
3. Prevent drastic simplification
4. Use priority queue
5. Preserve topology

A. Simplification envelope
B. Progressive meshes
C. Quadric error
D. Vertex clustering
E. Voxel based
F. Geometric optimization
G. Decimation

- How to evaluate the performance of these techniques ?
- What factors contribute to efficient transmission ?
- Can we assess quality solely based on geometric computation ?

Perceptual Metrics

Visual comparison:

Same geometry but decreasing texture quality. Nutcracker texture image size:
66KB, 27KB and 11KB

Same texture quality but different geometry resolutions:

Which has higher resolution:
Left ?
Right ?

Perceptual Metrics

Visual comparison:

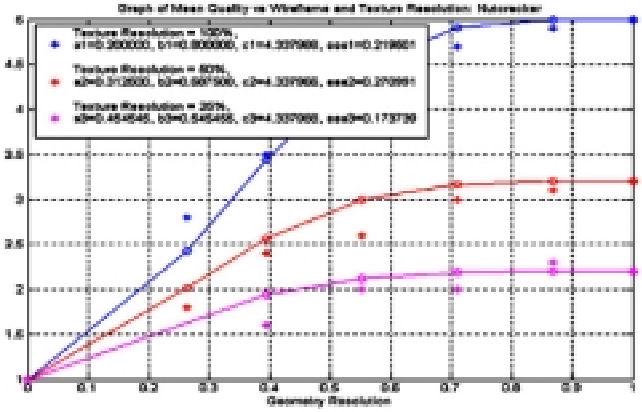
Same geometry but decreasing texture quality. Nutcracker texture image size:
66KB, 27KB and 11KB

Same texture quality but different geometry resolutions: 18,720 and 1,872 triangles

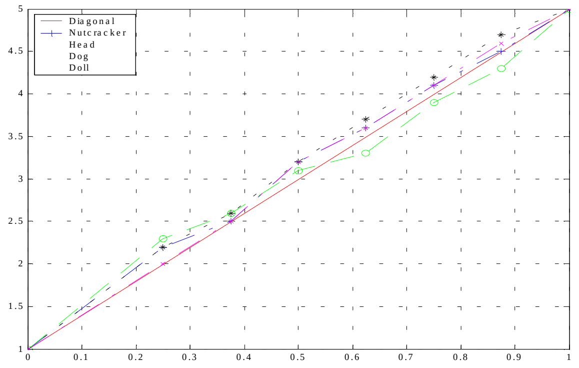
Perceptual Metrics (continued)

Y. Pan, I. Cheng, and A. Basu, “Quantitative metric for estimating perceptual quality of 3D objects,” IEEE Transactions on Multimedia, in press, 2004.

- Past perceptual experiments showed that increasing geometry resolution has no significant effect after reaching a certain threshold, while increasing texture resolution continues to improve visual fidelity.



geometry



texture