CSCI 4560/6560 Computational Geometry

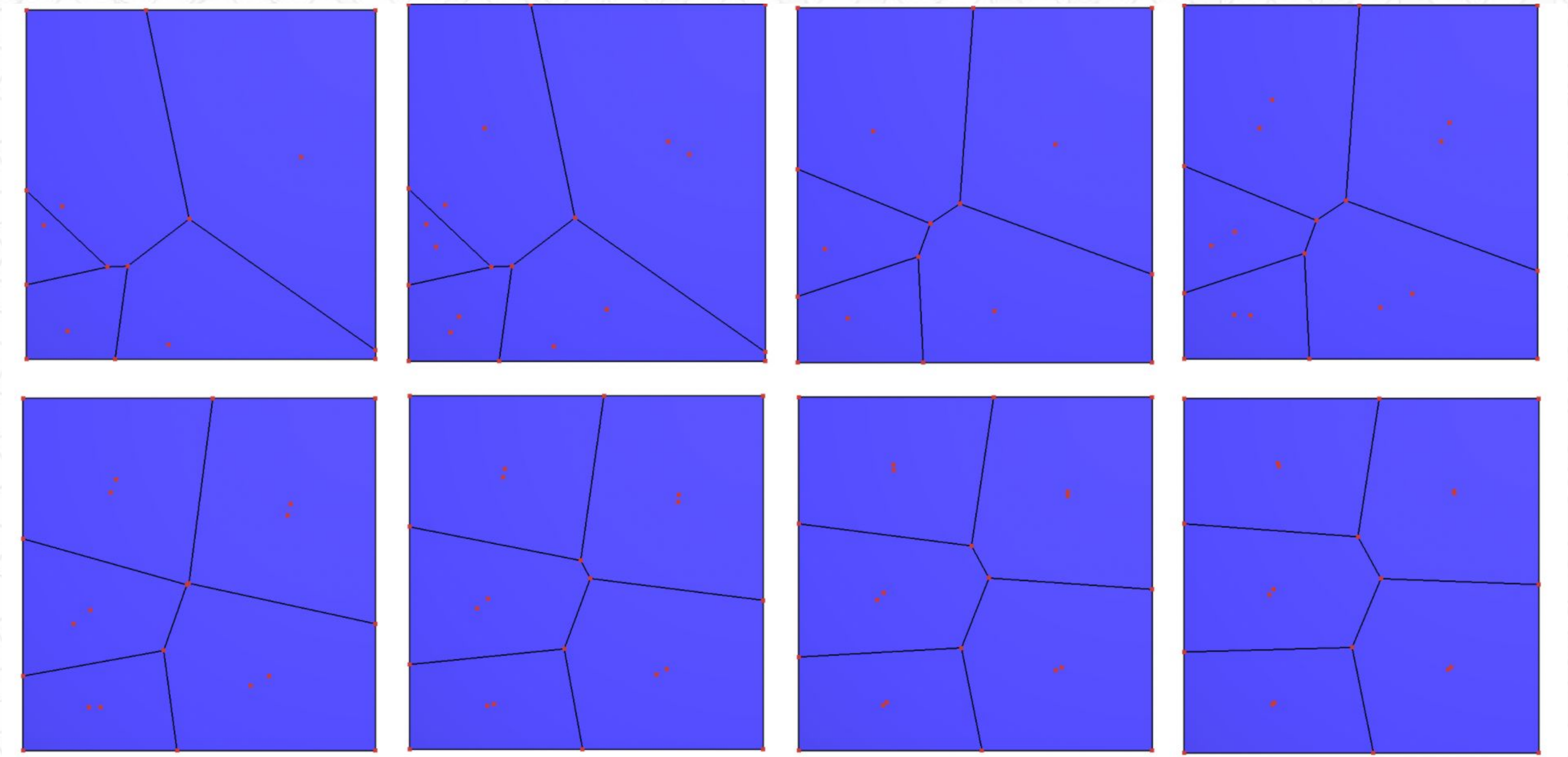
https://www.cs.rpi.edu/~cutler/classes/computationalgeometry/S22/

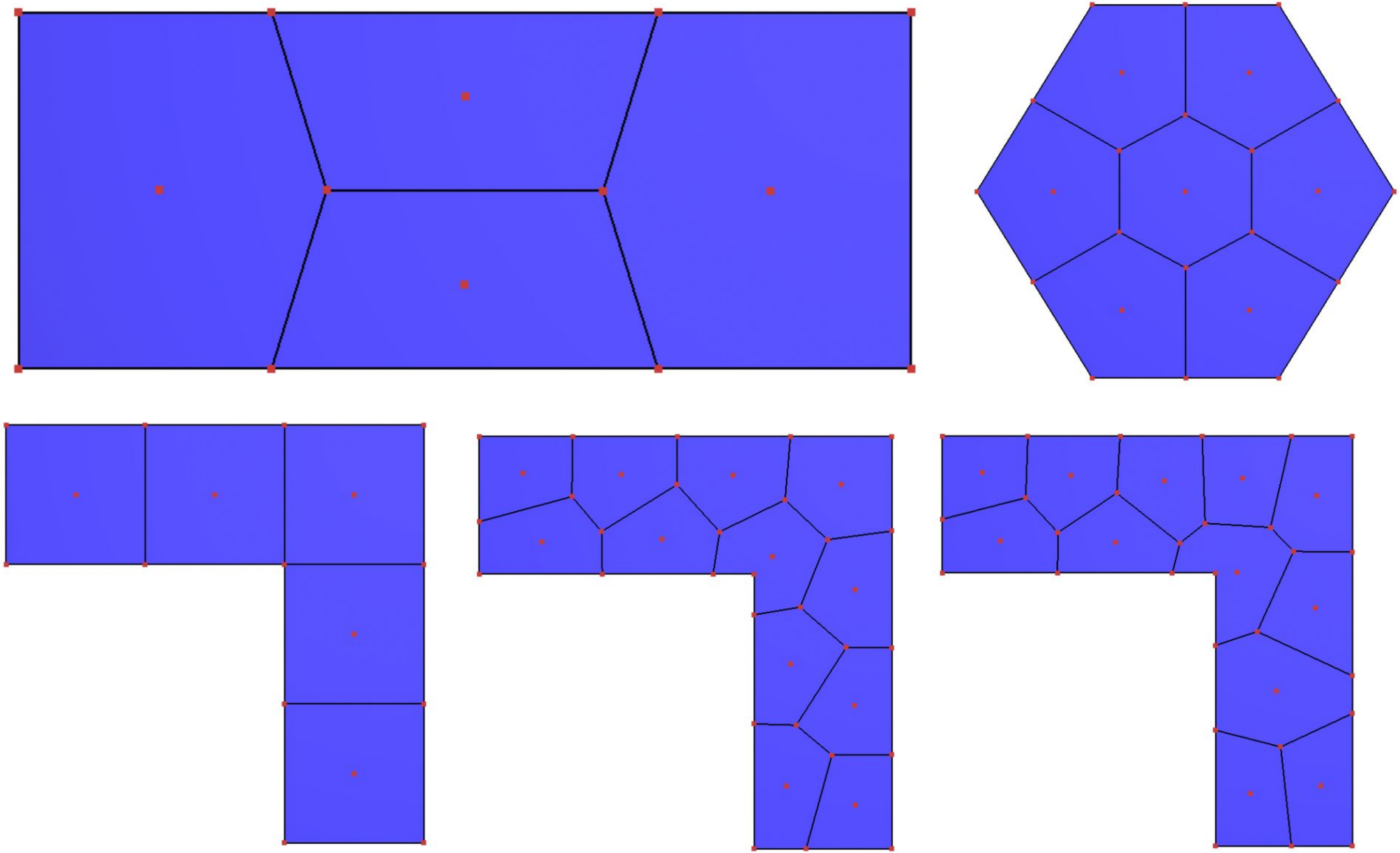
Lecture 17: Quad Trees

Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

Homework 5 Questions?



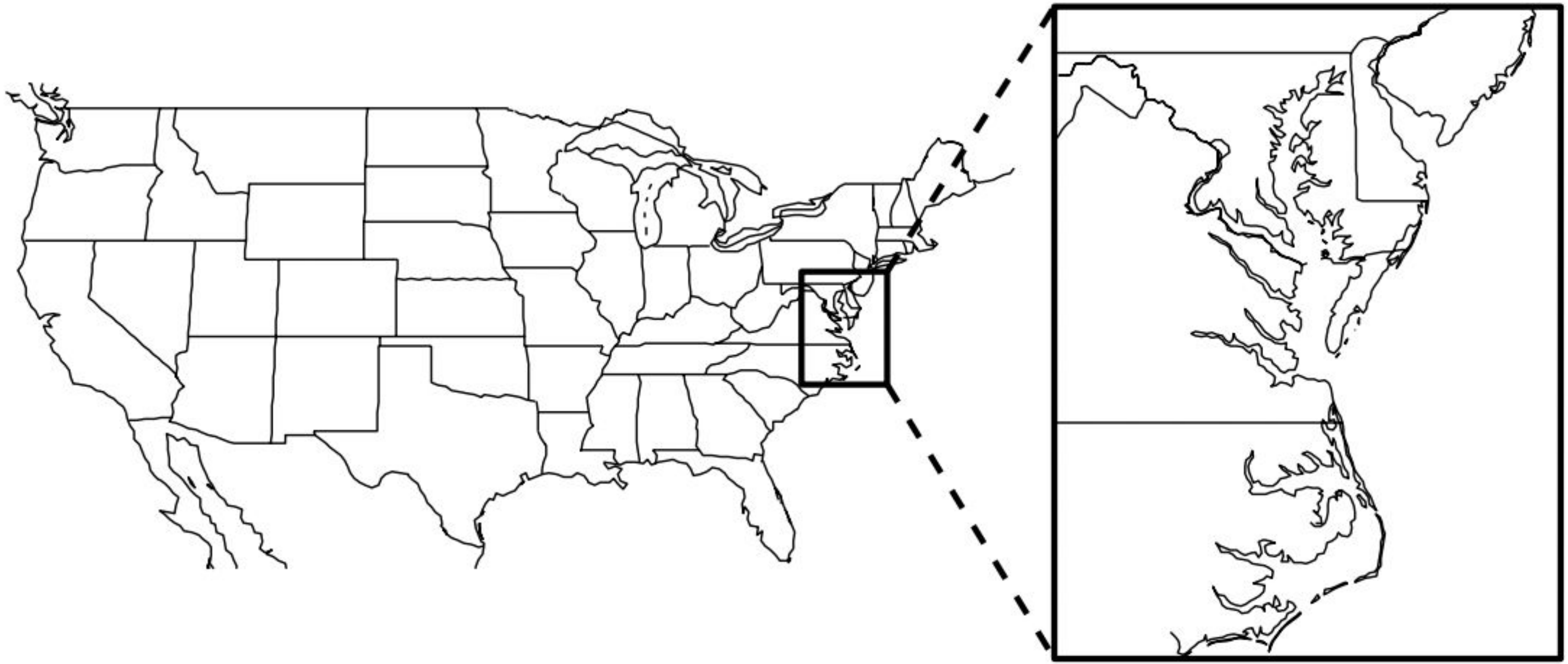


Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

Motivation: Cartography (Map-Making)

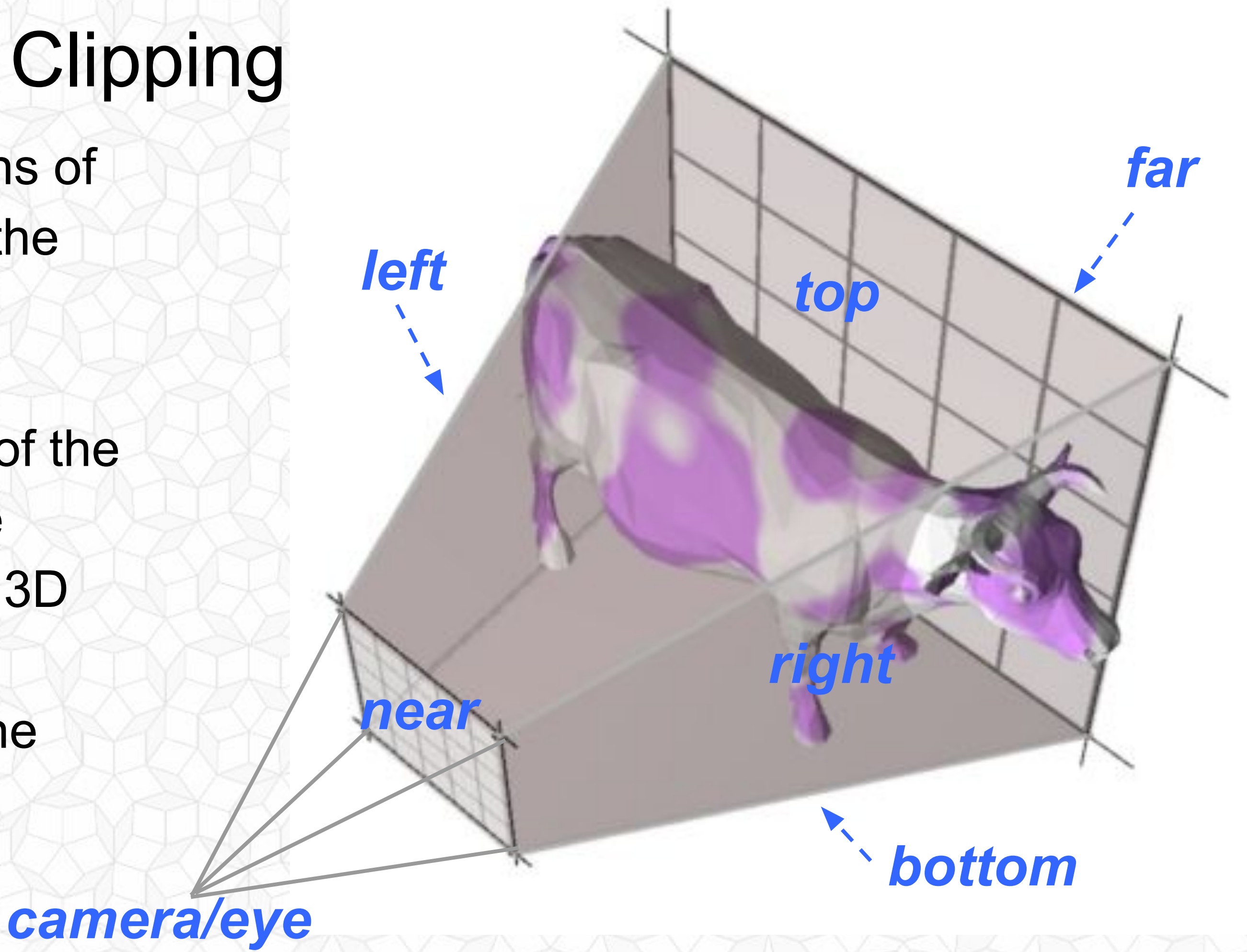
- Select a small rectangular region to display in a window at larger scale



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 10

Graphics: 3D Clipping

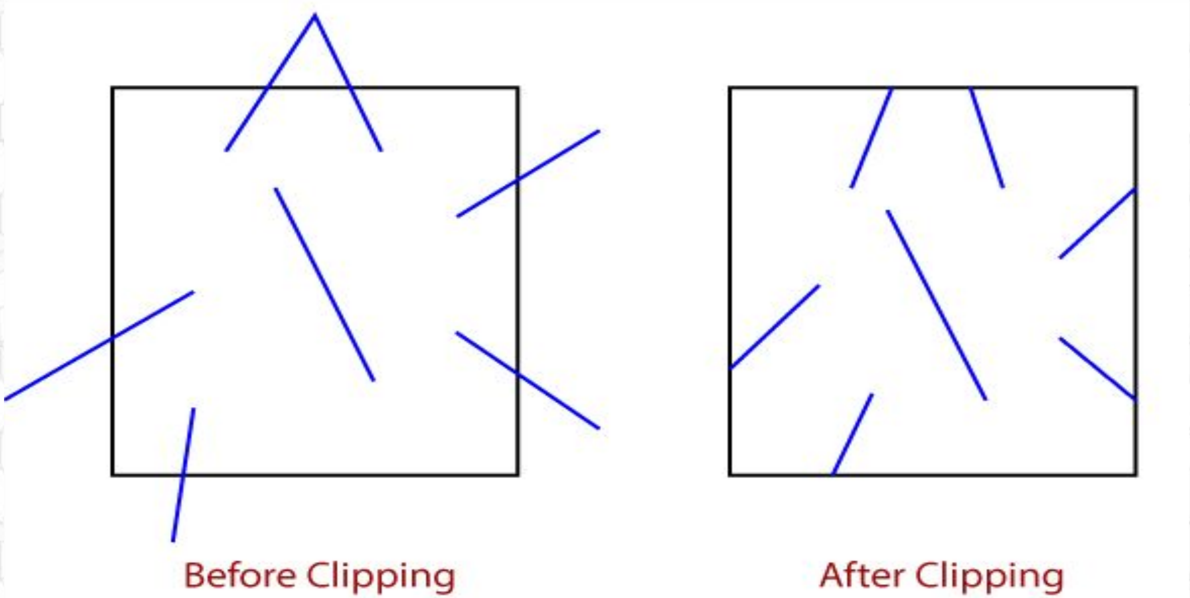
- Eliminate portions of objects outside the viewing frustum
- View Frustum
  - boundaries of the image plane projected in 3D
  - a near & far clipping plane



Graphics: 2D Clipping

Why do it?

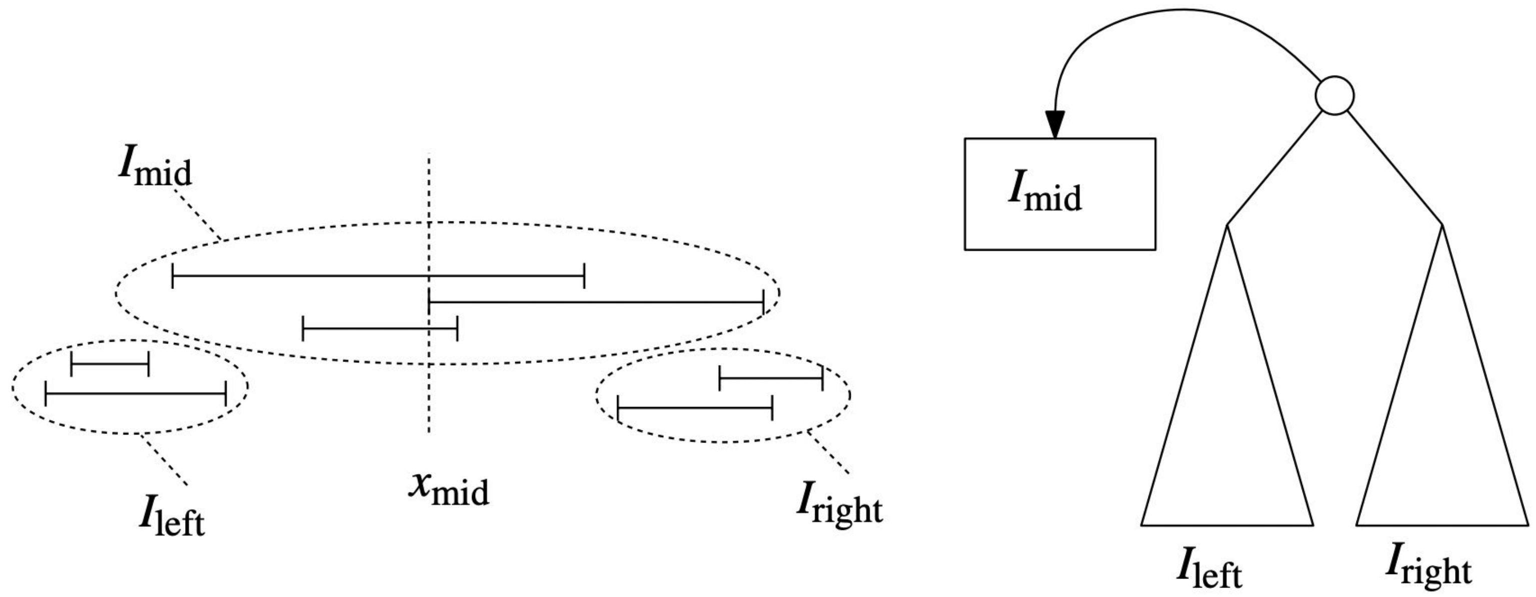
- Reduce amount of geometry going through graphics pipeline
- Prevent rendering bugs from overflow, wraparound, things behind the camera, etc.



https://www.tutorialandexample.com/clipping-in-computer-graphics

Interval Tree

- Recurse down the tree only with items that DO NOT overlap the split point.

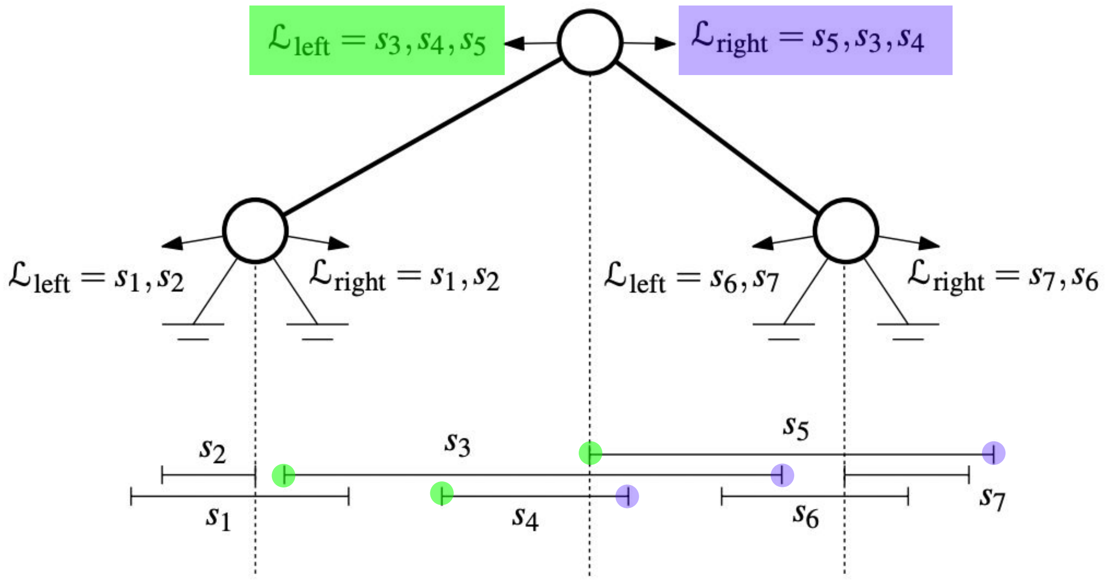


Interval Tree

- Items in I_{mid} group will stay at the current node

- Each node stores two two sorted lists:

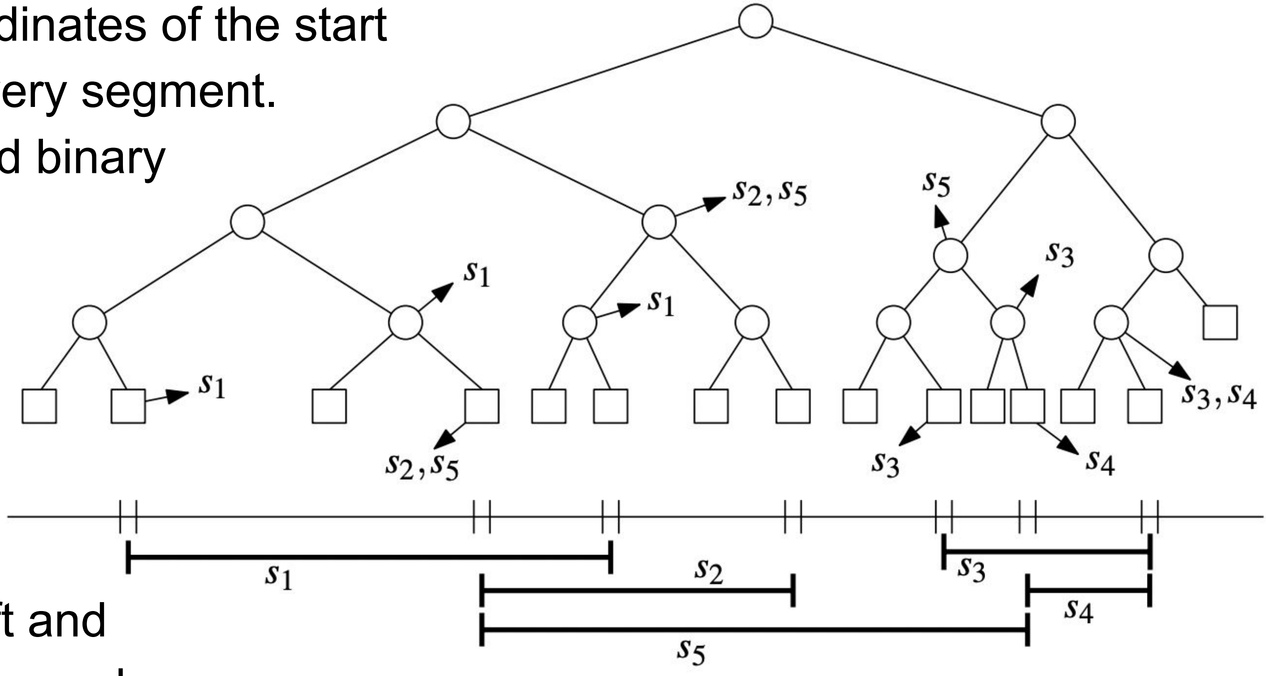
- $L_{left} = I_{mid}$ sorted by left endpoint (increasing)
- $L_{right} = I_{mid}$ sorted by right endpoint (decreasing)



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 10

Segment Tree - First Dimension (*x*)

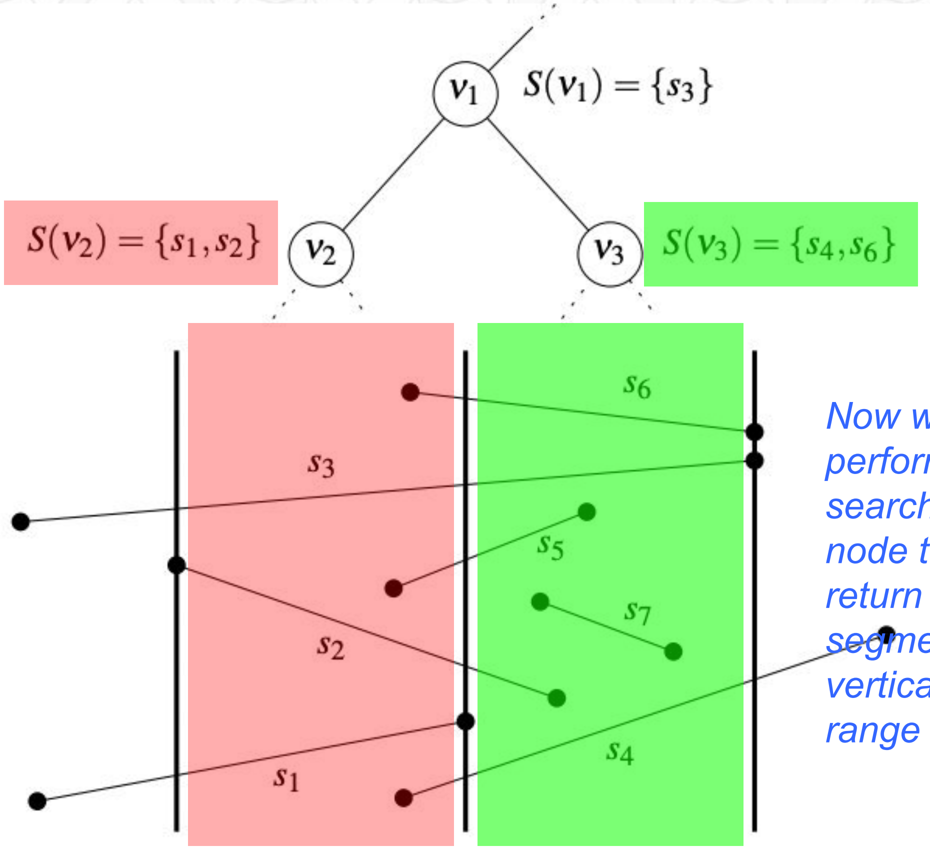
- First, sort the *x* coordinates of the start and end points of every segment.
- Construct a balanced binary search tree with these *x* values.
- Insert every segment into the structure
- If a segment overlaps both the left and right subranges of the node store it at the node (do not recurse)



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 10

Segment Tree - Second Dimension (*y*)

- To efficiently query a vertical range in addition to the horizontal range:
- Sort the segments stored at each node by *y*
- *Remember: this is only the segments that completely overlaps the node's range*
- Note: this is why we require no crossings in the input segments

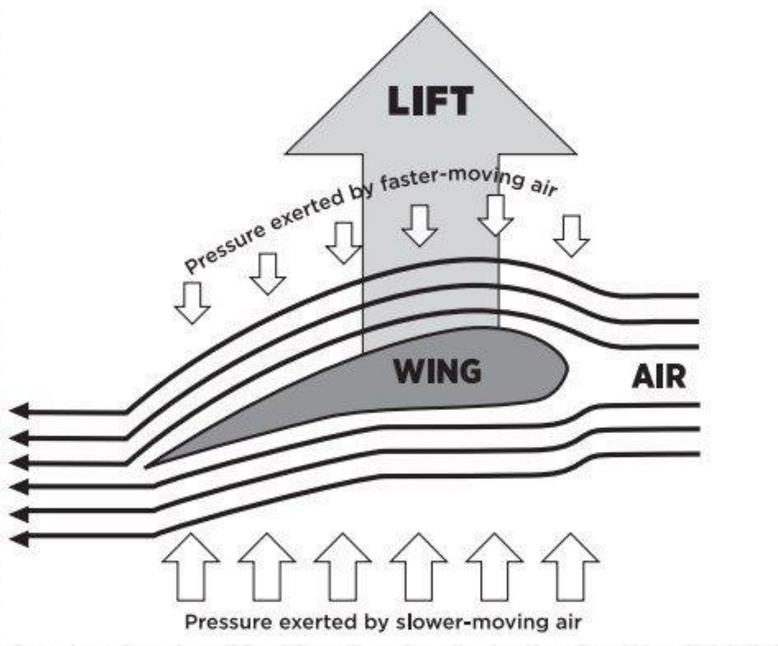


Now we can perform binary search at each node to only return the segments in the vertical query range

Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

Motivation: Finite Element Modeling (FEM) & Computational Fluid Dynamics



https://www.scienceworld.ca/resource/plane-wing-simulator/

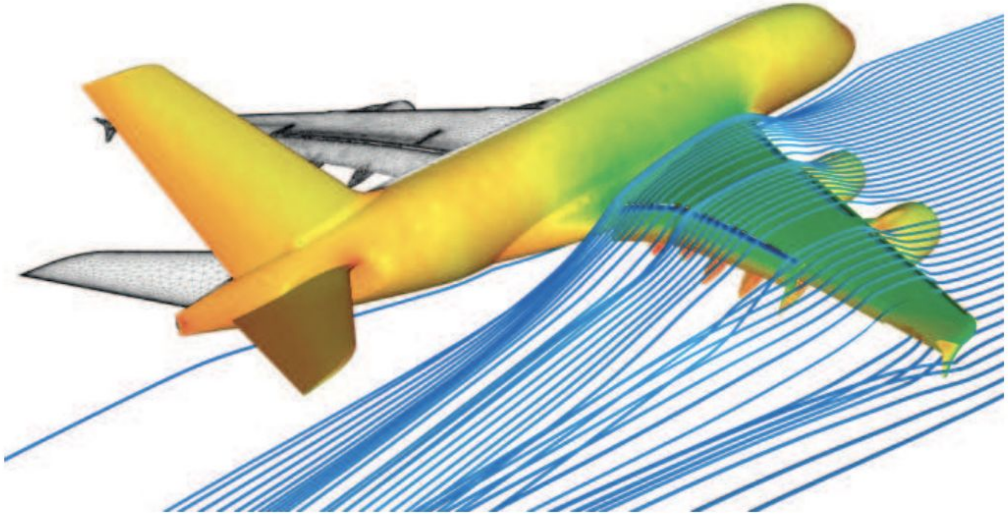
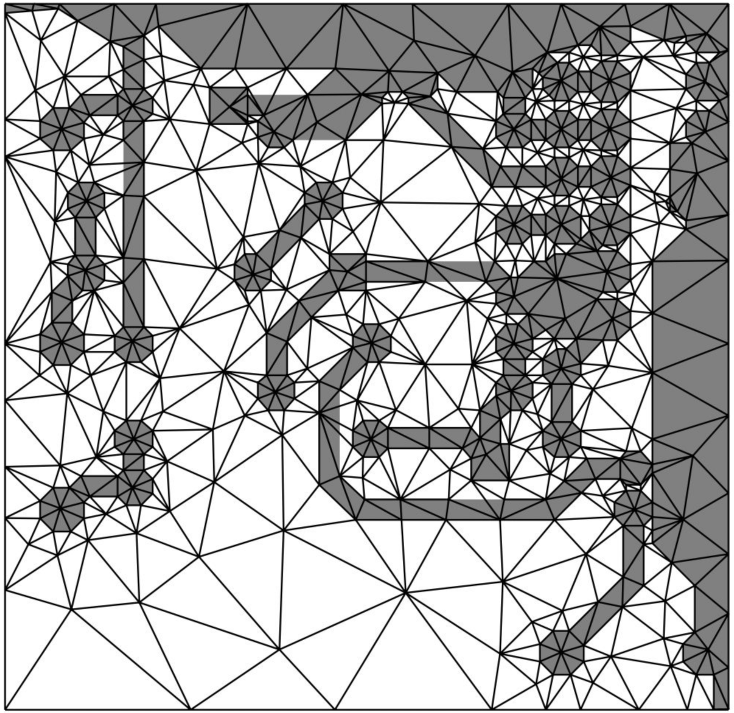


Figure 9: Numerical flow simulation for the Airbus A380 (picture credit: Airbus. Copyright: Dr. Klaus Becker, Senior Manager Aerodynamic Strategies, EGAA, Airbus, Bremen, Germany)

Motivation: Meshing Goals & Requirements

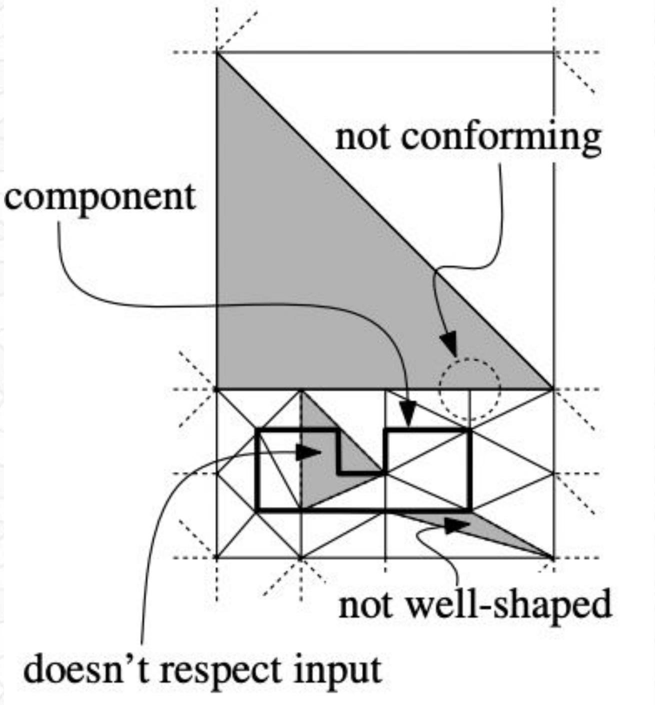
- Triangular Mesh
- Conforming (no T junctions)
- Respect (align) with input surface
- Well shaped (minimum & maximum angle requirements)
- Non-uniform
  - Fine when near input surface (ensure accurate simulation)
  - Coarse when far from the input surface (reduce computation waste)



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

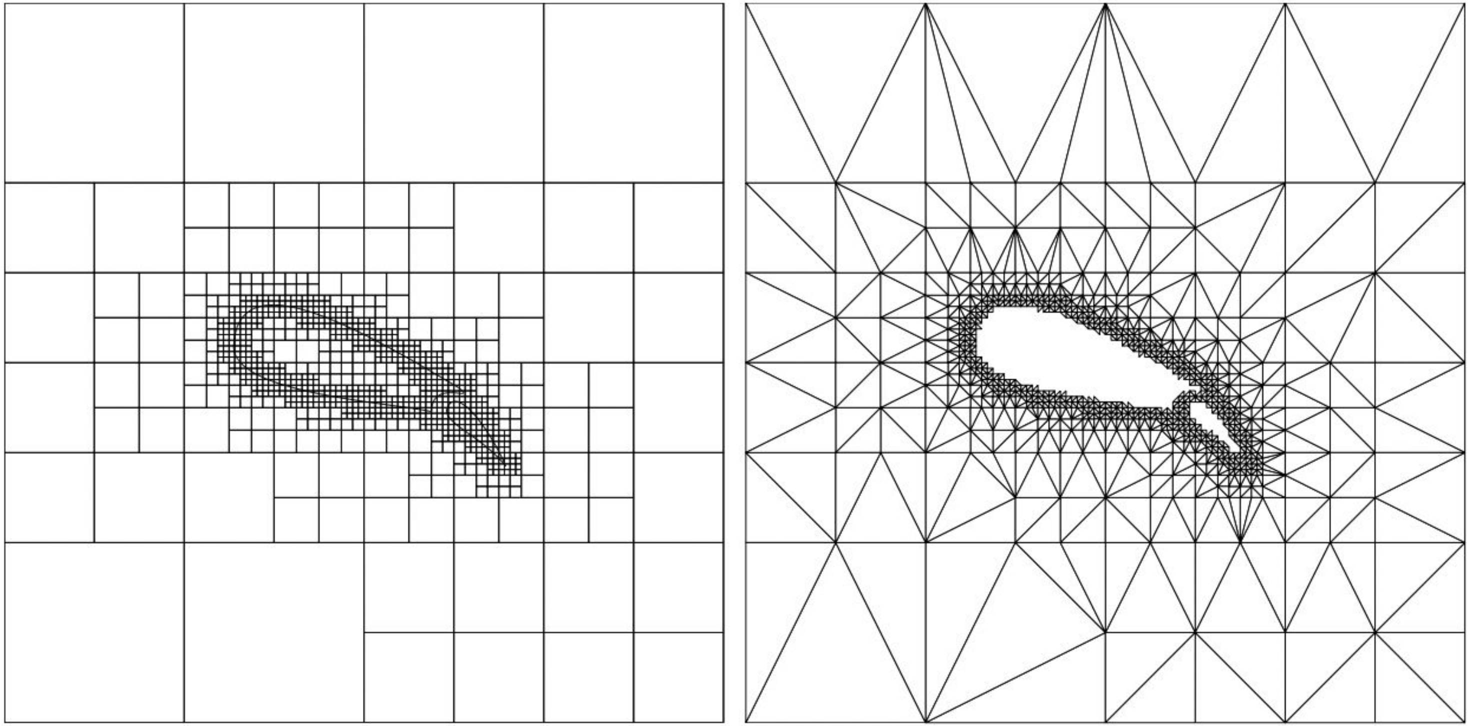
Motivation: Meshing Goals & Requirements

- Triangular Mesh
- Conforming (no T junctions)
- Respect (align) with input surface
- Well shaped (minimum & maximum angle requirements)
- Non-uniform
  - Fine when near input surface (ensure accurate simulation)
  - Coarse when far from the input surface (reduce computation waste)



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 14

Motivation: Finite Element Modeling (FEM) & Computational Fluid Dynamics



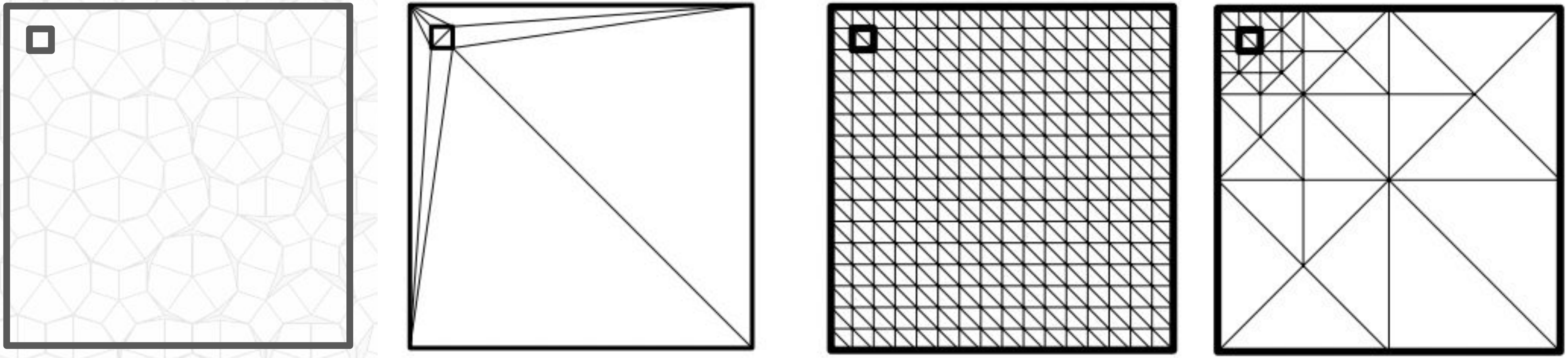
“Delaunay Refinement for Curved Complexes”, Adriano Chaves Lisboa, 2008.

Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

Uniform vs. Non-Uniform Meshing

Input | **Delaunay** | **Uniform** | **Non-Uniform**



Only uses the input vertices

Addition of "Steiner vertices" is allowed to improve shape (minimum/maximum triangle angle)

Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

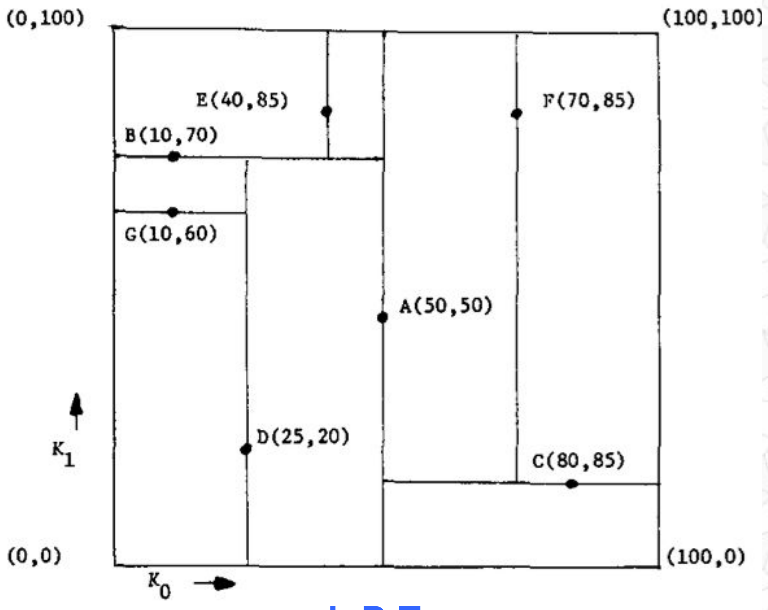
Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

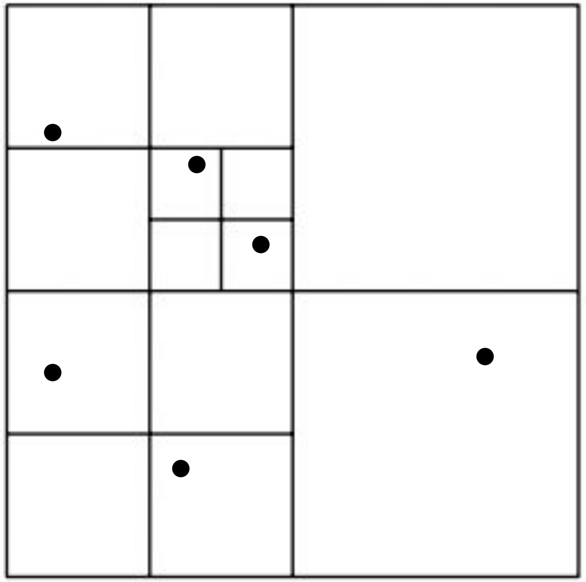
QuadTree - Basically a special case of k-D Tree

- Split all dimensions at once (instead of alternating one dimension per level)
- Always split at the midpoint (generally not perfectly balanced!)

"Multidimensional Binary Search Trees Used for Associative Searching", Communications of the ACM, Bentley 1975

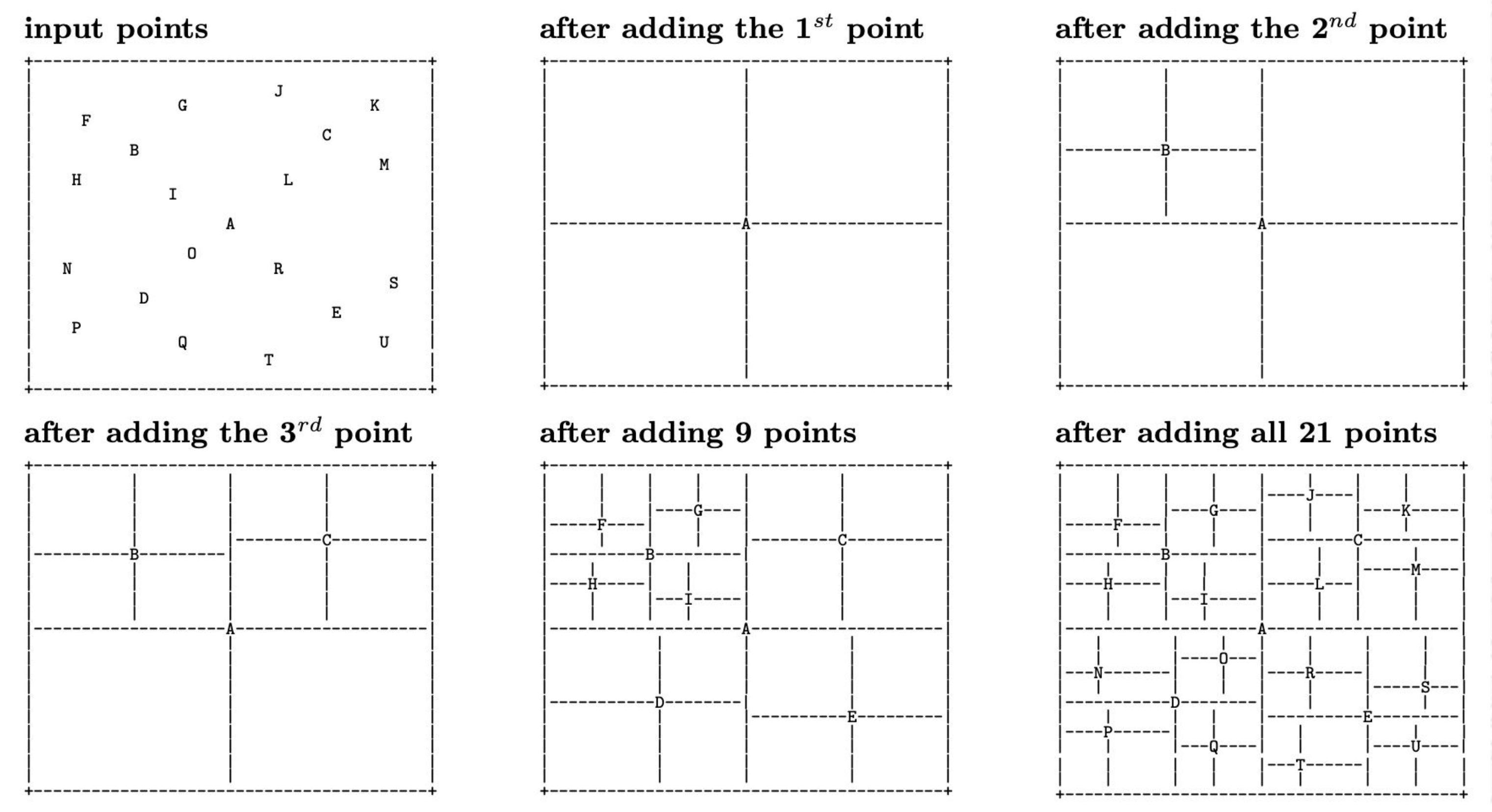


k-D Tree



Quad Tree

Data Structures Homework 8: Quad Tree



Technically this is a variant of a classic QuadTree.

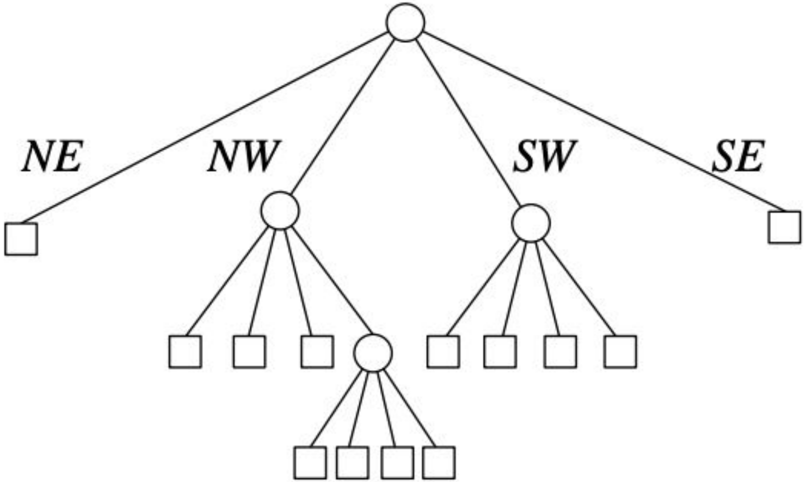
Instead of splitting at the dimension midpoint, we split at a specific data point…

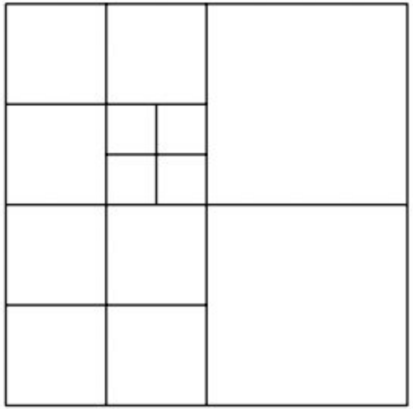
Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

QuadTree Structure Consistency

- Let's split a cell into 4 children if 2 nodes are placed into the same cell
- Points that lie on a vertical split, assigned to left child
- Points that lie on a horizontal split, assigned to bottom child



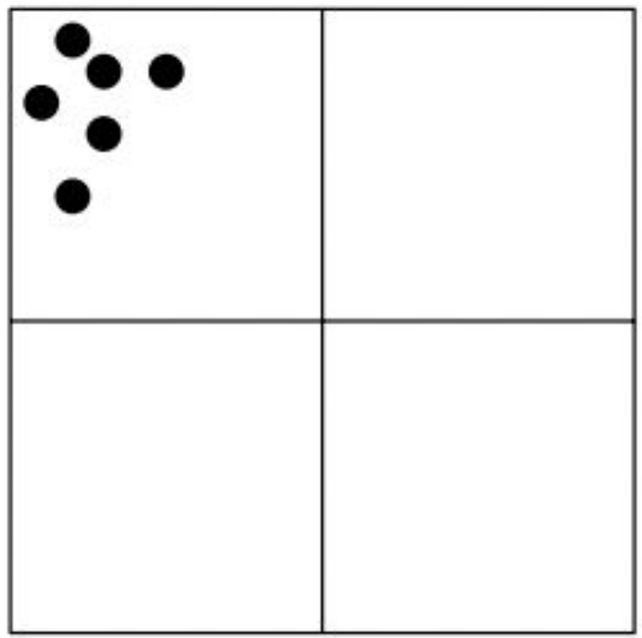


Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

Maximum Quad Tree Depth

- We always split a cell into 4 children if 2 nodes are placed into the same cell….

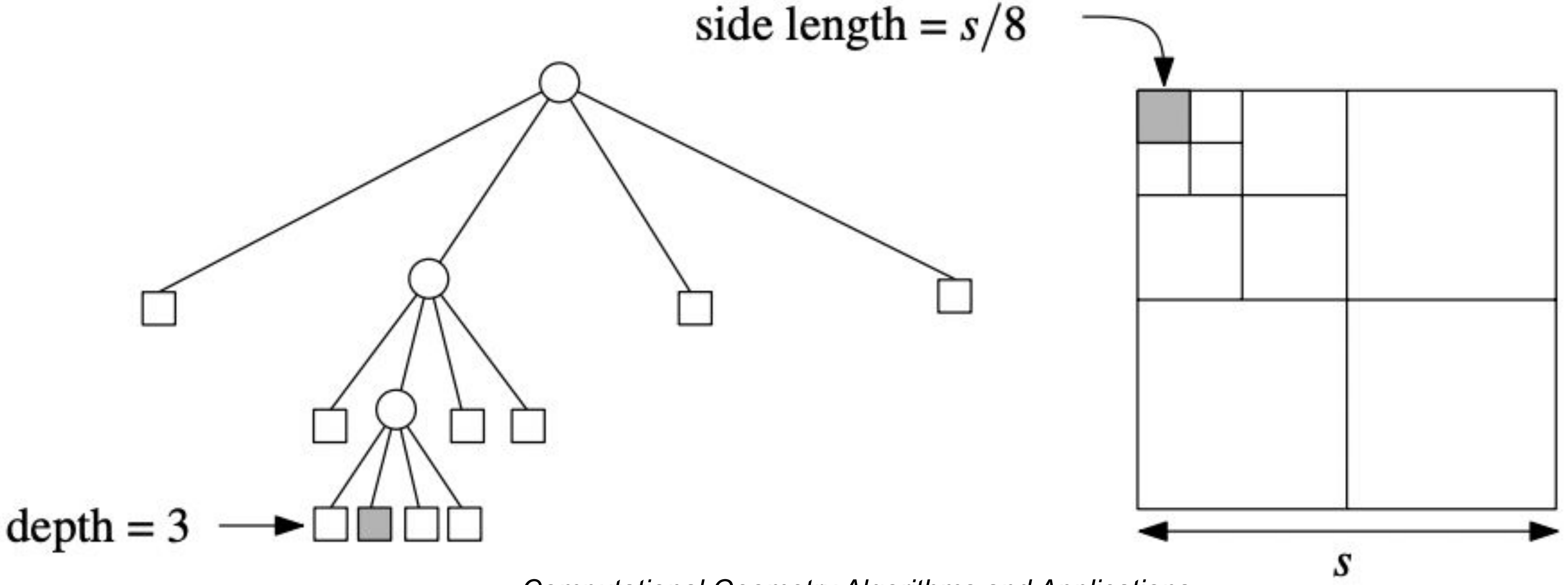
- **Lemma 14.1:** The depth of a quadtree for a set *P* of points in the plane is at most $\log(s/c) + 3/2$, where *c* is the smallest distance between any two points in *P* and *s* is the side length of the initial square that contains *P*.



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

Maximum Number of Nodes

- **Theorem 14.2:** A quadtree of depth *d* storing a set of *n* points has $O((d+1)n)$ nodes and can be constructed in $O((d+1)n)$ time.



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

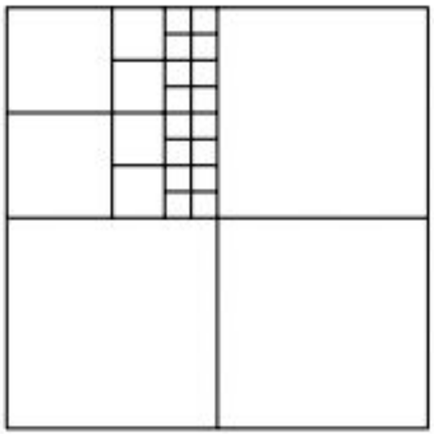
Outline for Today

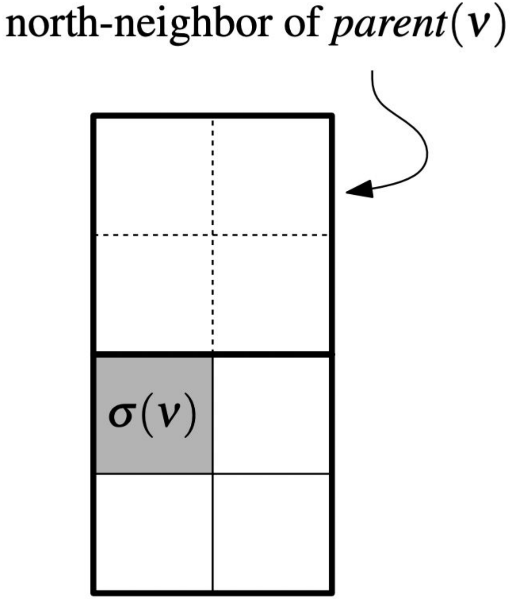
- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

QuadTree and Implicit Adjacency

- We don't need to explicitly store pointers to adjacency cell.

- However, it is disadvantageous to have adjacent cells that are subdivided to a significantly different tree level.
- Let's do something to make the tree more balanced….

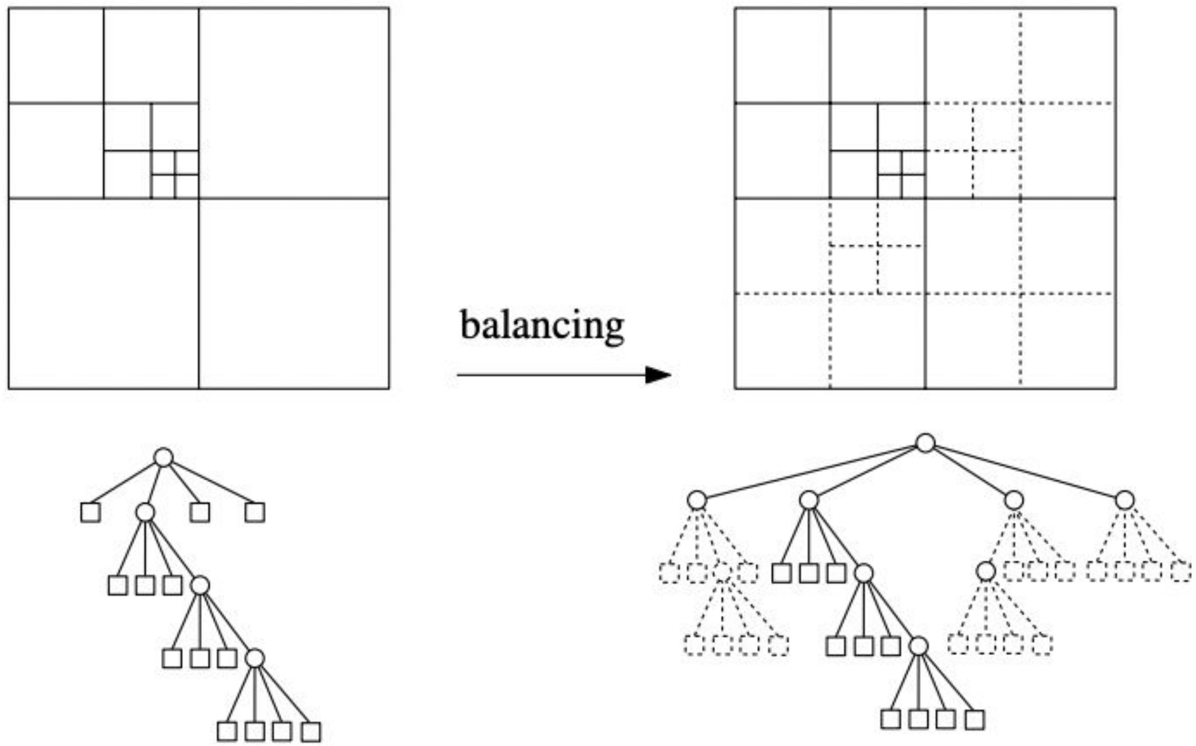




Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

Balanced Quad Tree

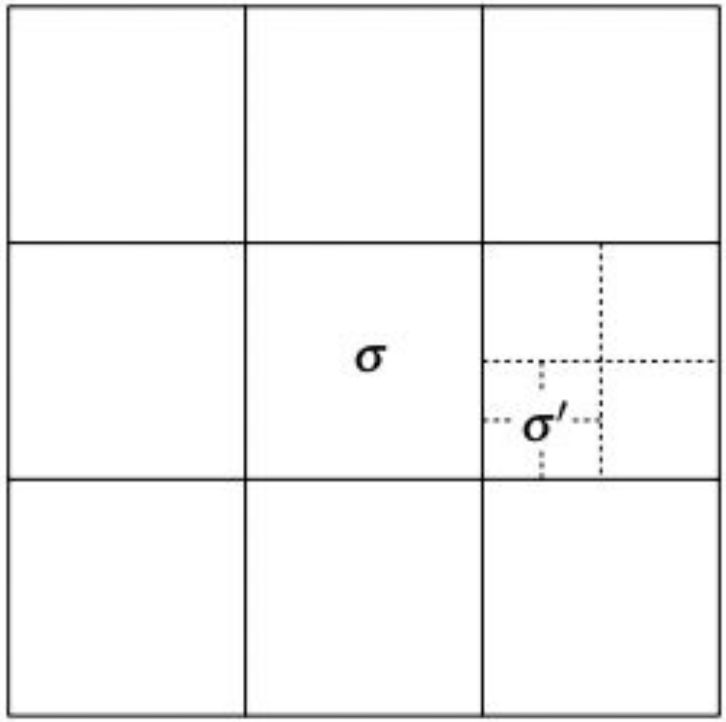
- Adjacent cells of the tree are no more than 1 split different



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 14

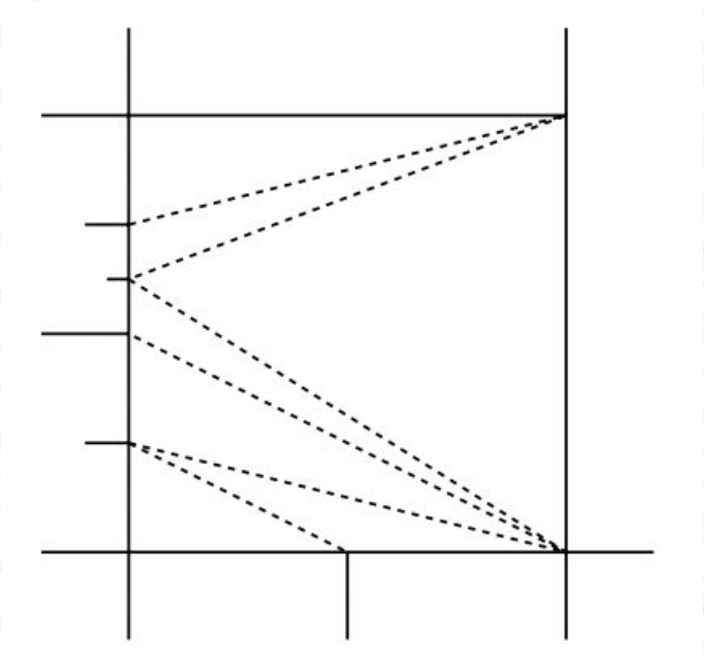
of Splits Required to Balance a QuadTree?

- **Theorem 14.4:** Let *T* be a quadtree with *m* nodes. Then the balanced version of *T* has *O(m)* nodes and it can be constructed in *O((d +1)m)* time.

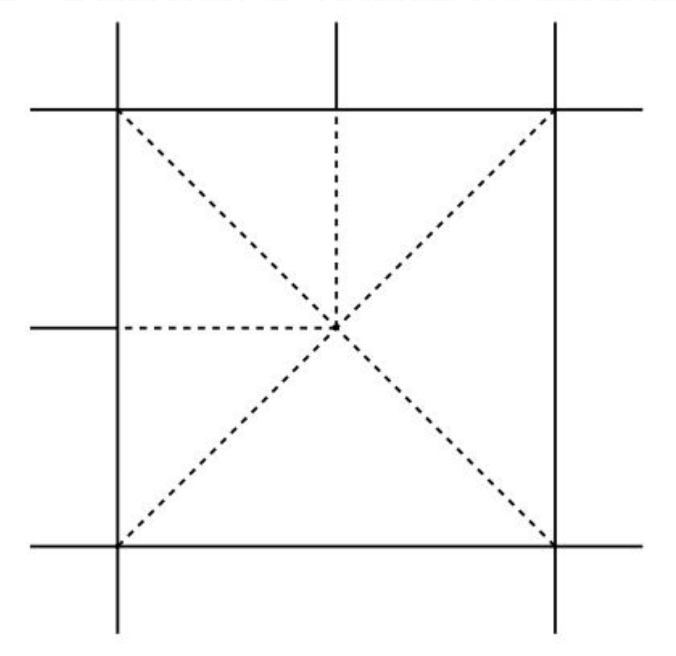


Balanced Quad Tree Triangulation

- A Balanced Quad Tree can be triangulated with all 45°/45°/90° triangles!



unbalanced Quad Tree

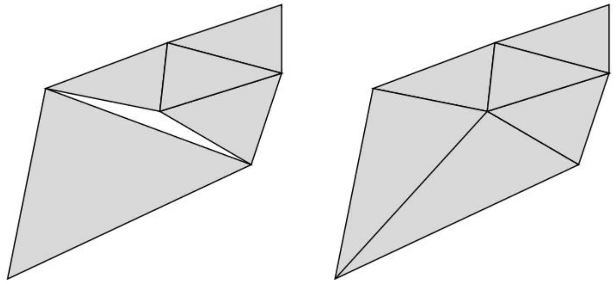


Balanced Quad Tree

Outline for Today

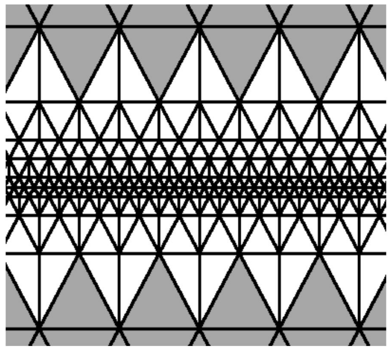
- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

√3 Subdivision, Kobbelt, SIGGRAPH 2000

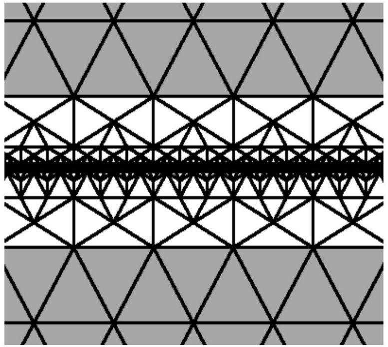


Adaptive Subdivision (Loop): Need to close gaps between different levels of refinement

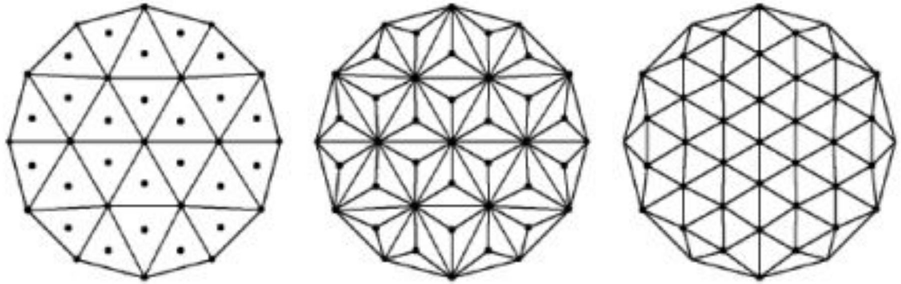
Loop: less localized refinement



√3: more localized refinement



√3 Subdivision: No intermediate special case



the split operation places a midvertex at the centre of each triangle

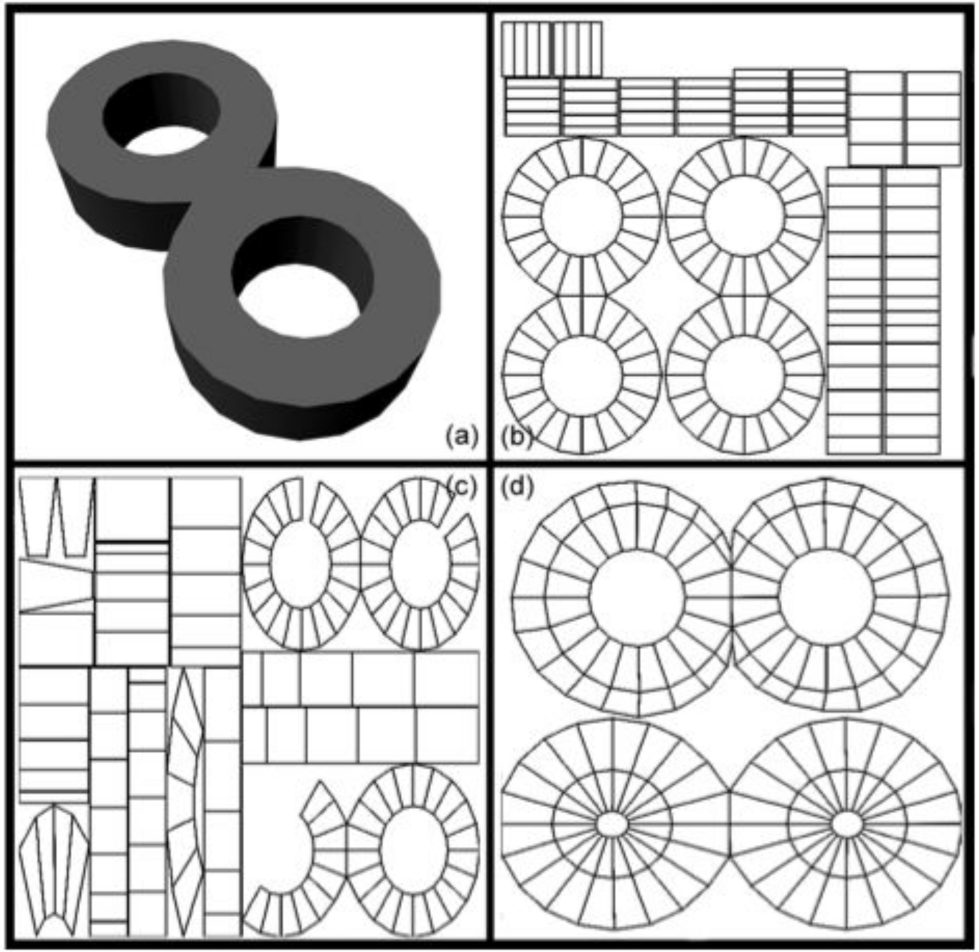
joining the midvertex to the vertices of the triangle realises the 1-to-3 split

after smoothing each old vertex, edges are flipped to connect pairs of midvertices

Traditional Texture Mapping

- Unroll / Unwrap the object to 2D
- Parameterize / Correspond 3D ↔ 2D
- “Paint” 2D texture

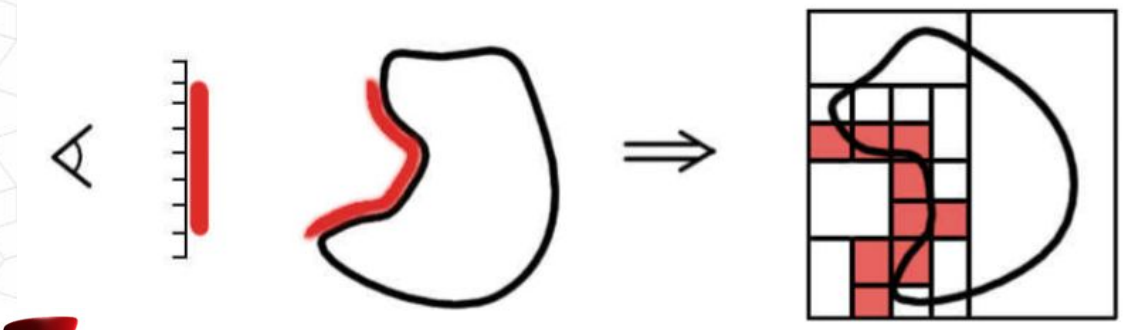
"Painting and Rendering Textures on Unparameterized Models”,
DeBry, Gibbs, Deleon, and Robins,
SIGGRAPH 2002

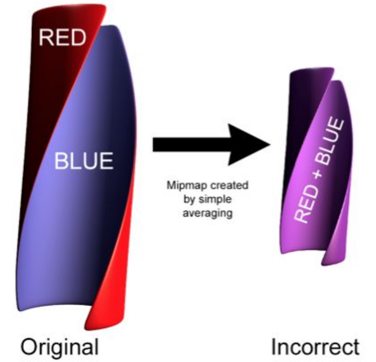


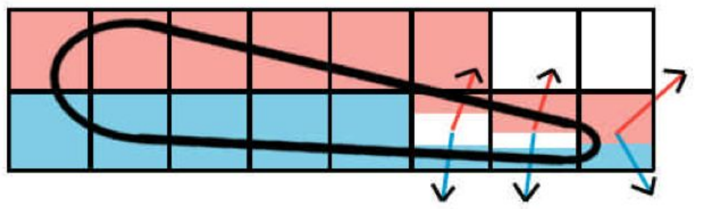
Octree Texture Mapping

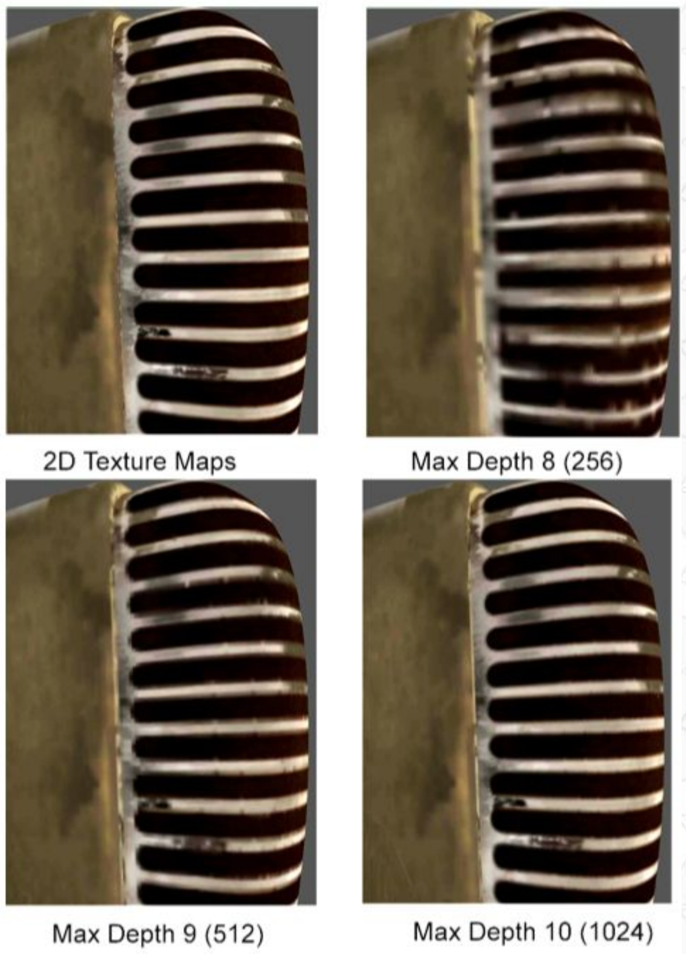
"Octree Textures", Benson & Davis, SIGGRAPH 2002

"Painting and Rendering Textures on Unparameterized Models”, DeBry, Gibbs, Deleon, and Robins, SIGGRAPH 2002







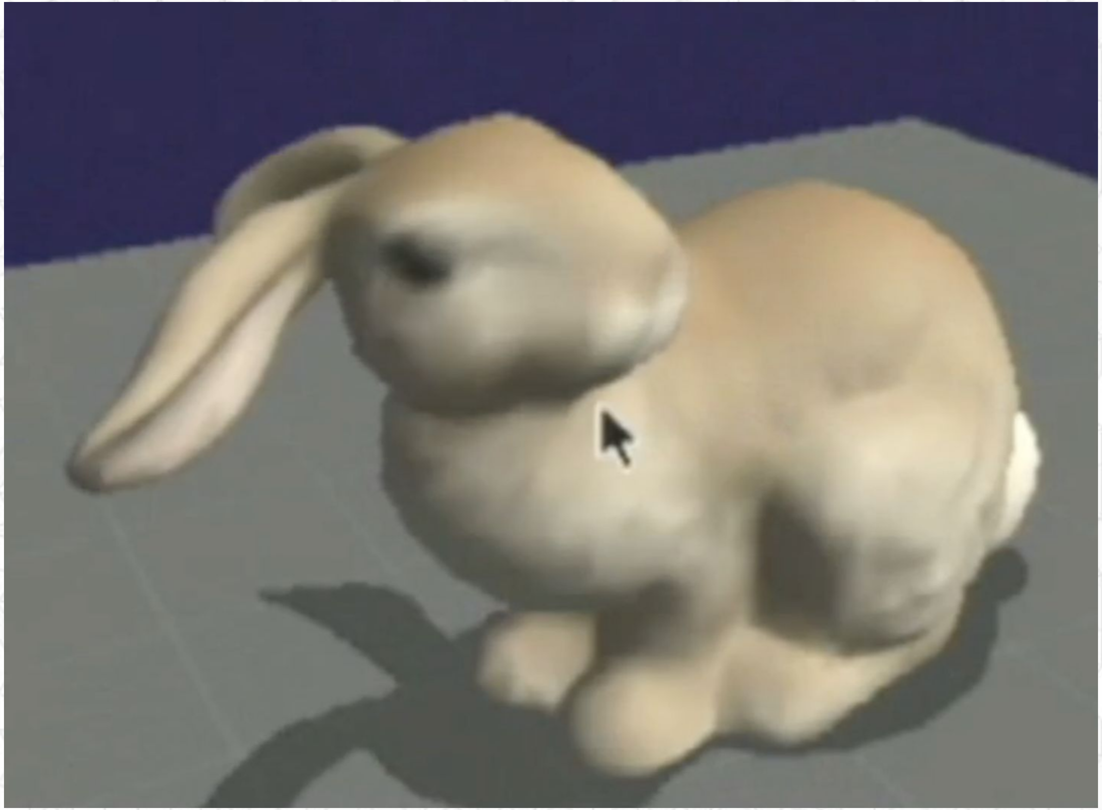


Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

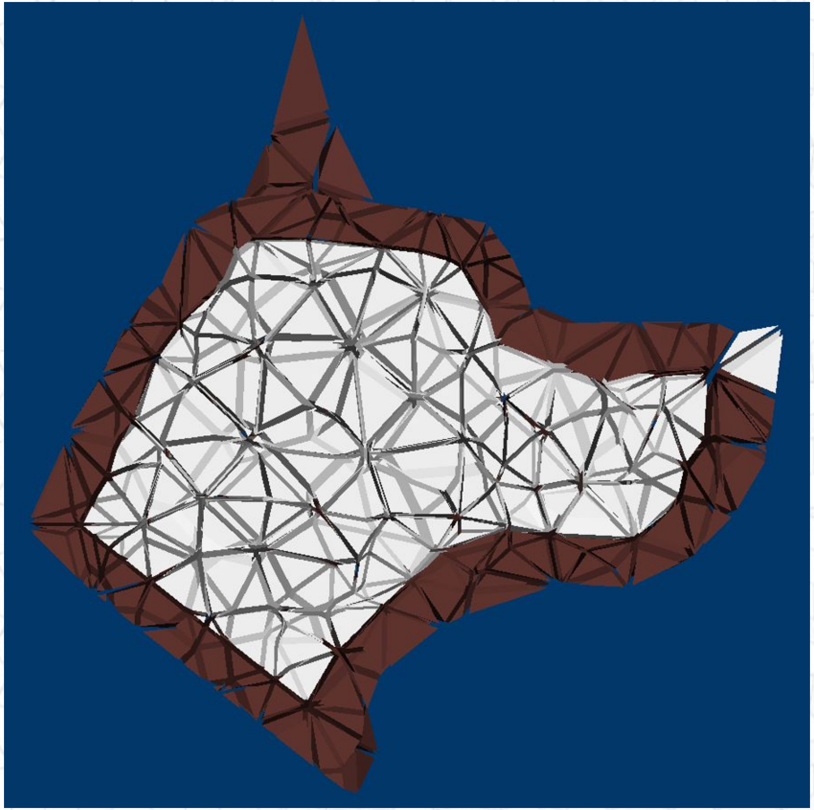
Deformation Simulation

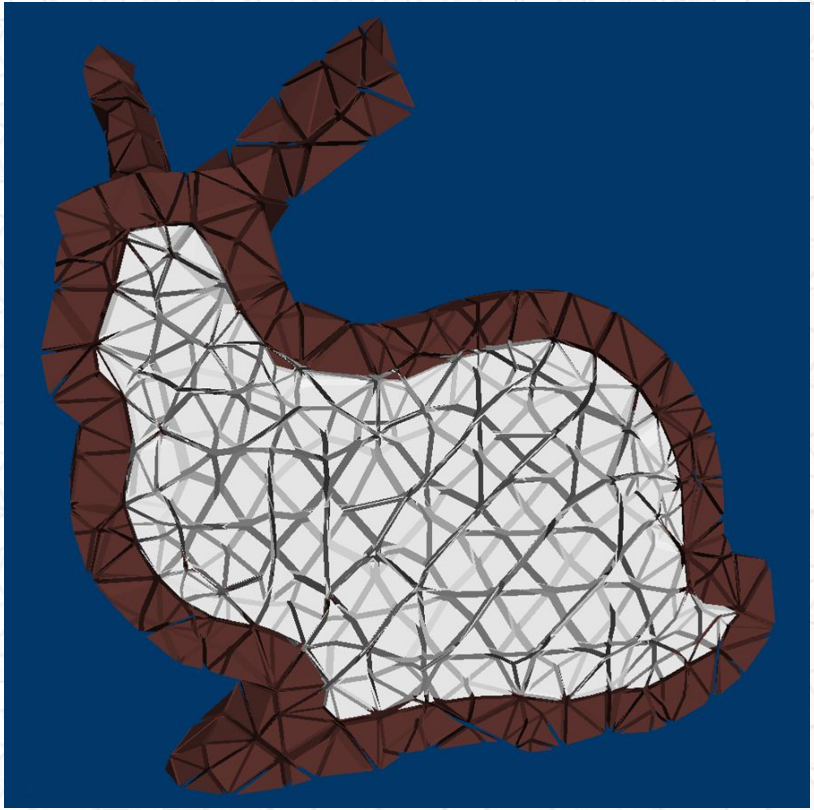
Mueller, Dorsey, McMillan, Jagnow, & Cutler
Stable Real-Time Deformations
Symposium on Computer Animation 2002



Deformation Simulation

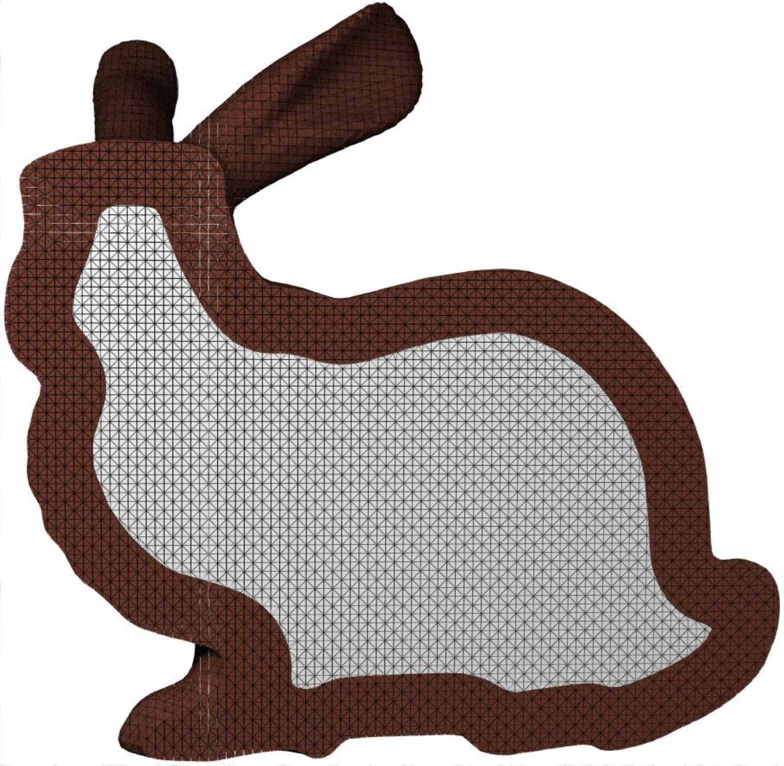
Mueller, Dorsey, McMillan, Jagnow, & Cutler
Stable Real-Time Deformations
Symposium on Computer Animation 2002



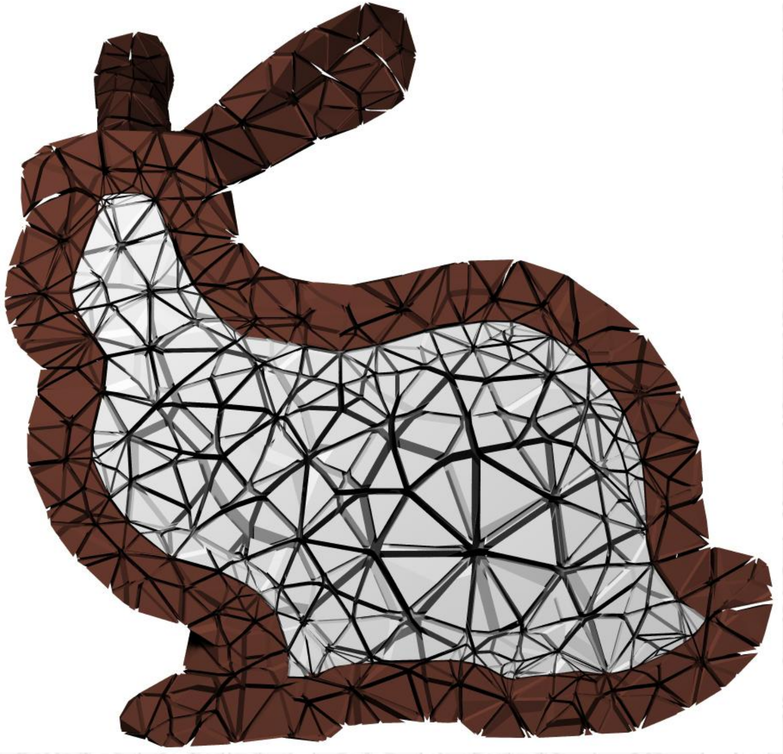


3D Mesh Simplification

"Simplification and Improvement of Tetrahedral Models for Simulation"
Cutler, Dorsey, and McMillan
SGP 2004



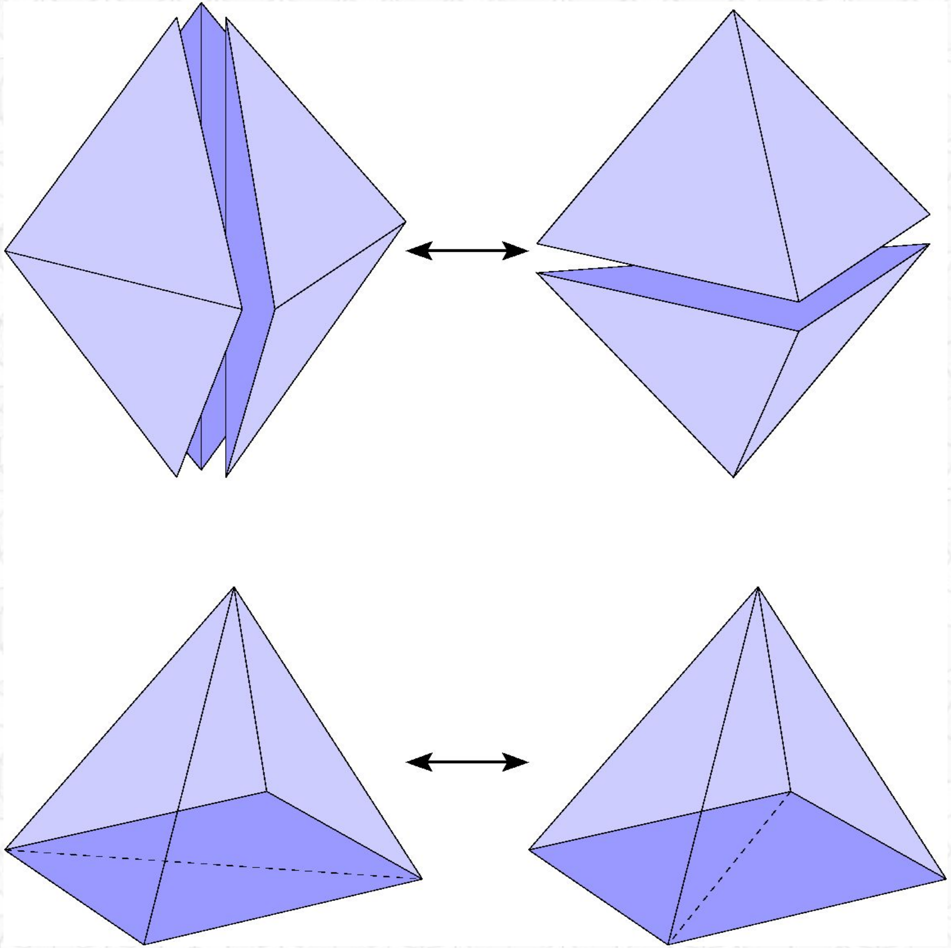
1,050K tetras
(133K faces)



10K tetras
(3K faces)

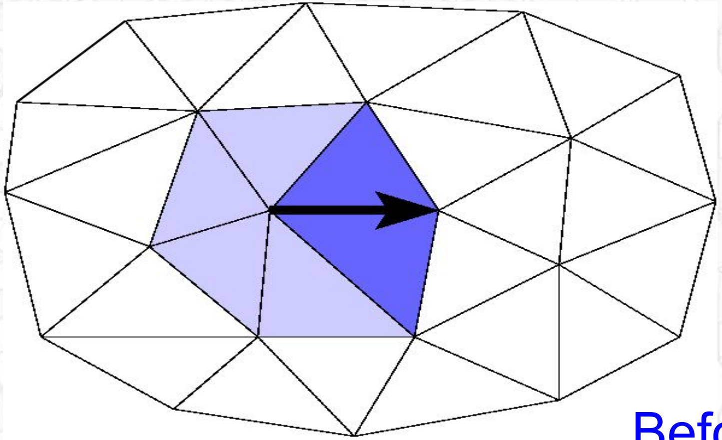
3D Mesh Operations

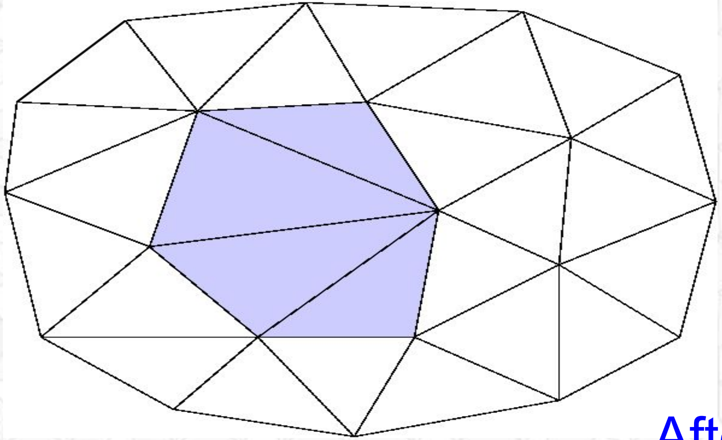
- Tetrahedral Swaps
  - Choose the configuration with the best local element shape
- Edge Collapse
- Vertex Smoothing
- Vertex Addition



3D Mesh Operations

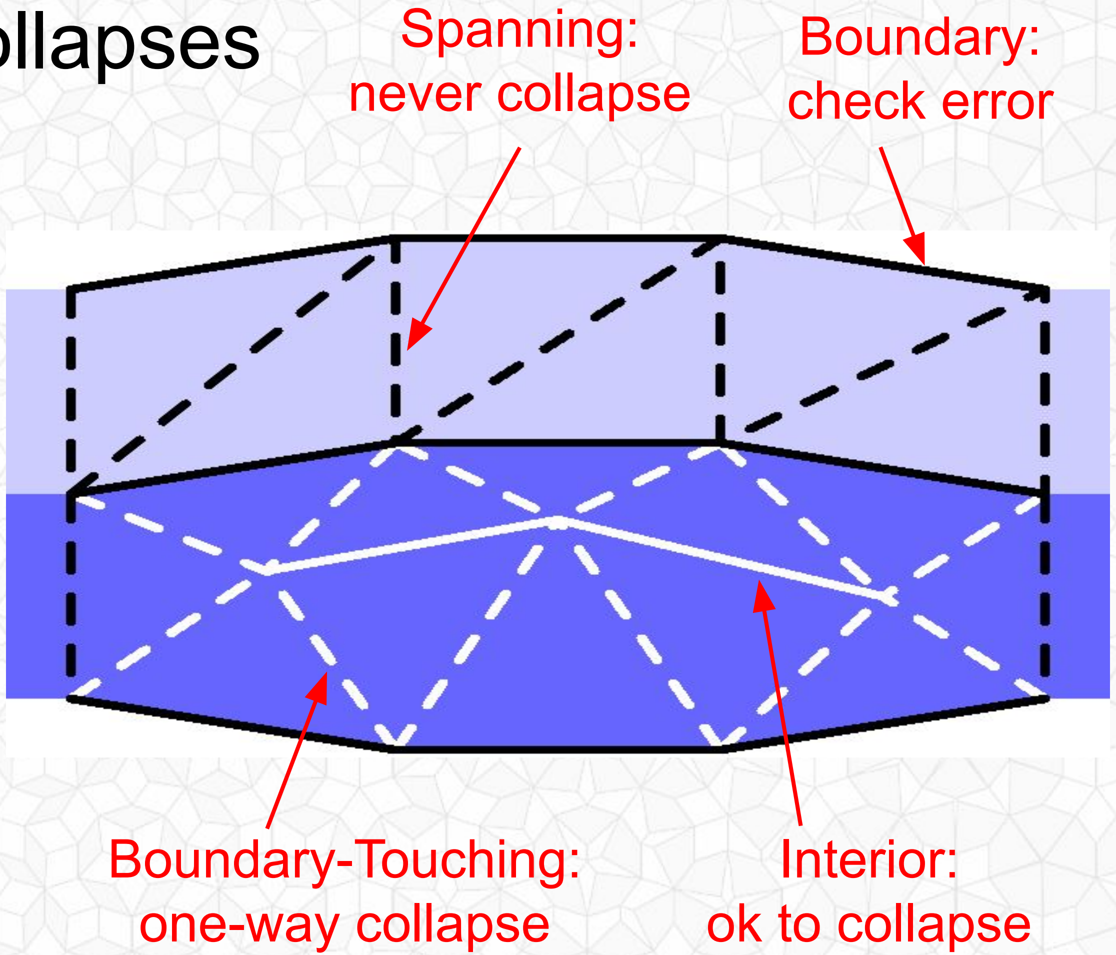
- Tetrahedral Swaps
- Edge Collapse
  - Delete a vertex & the elements around the edge
- Vertex Smoothing
- Vertex Addition





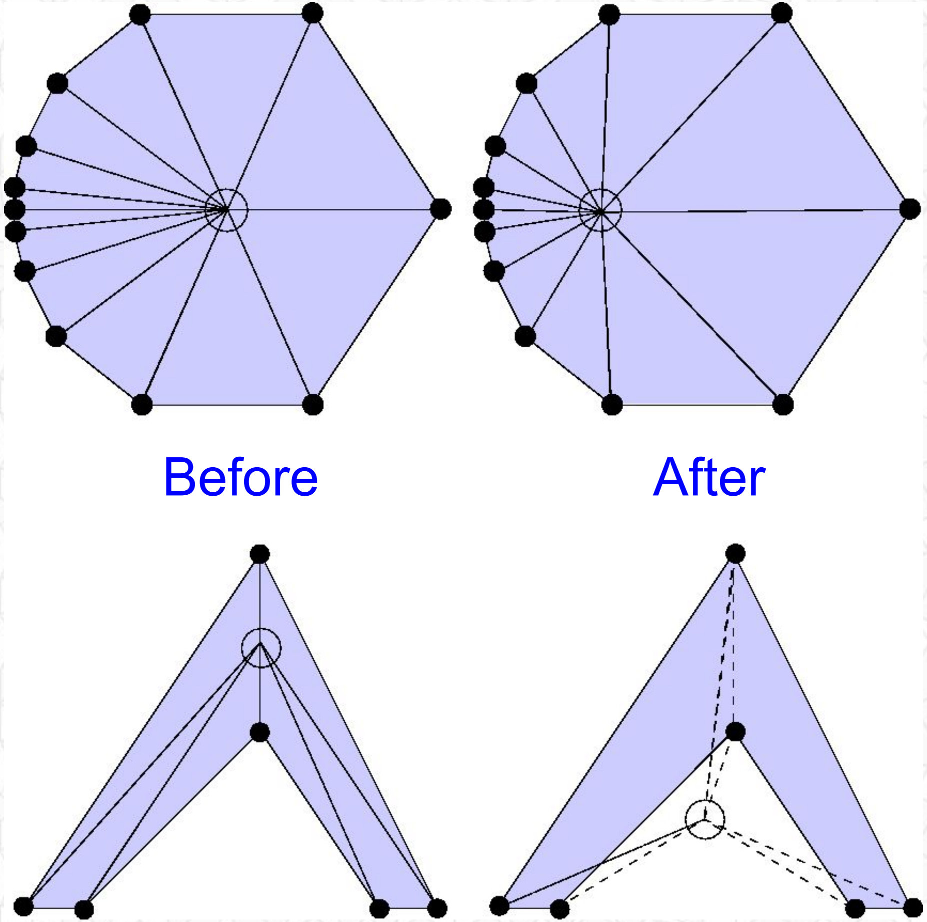
Prioritizing Edge Collapses

- Preserve topology
  - Thin layers should not pinch together
- Collapse weight
  - Edge length + boundary error
- No negative volumes
- Local element quality does not significantly worsen



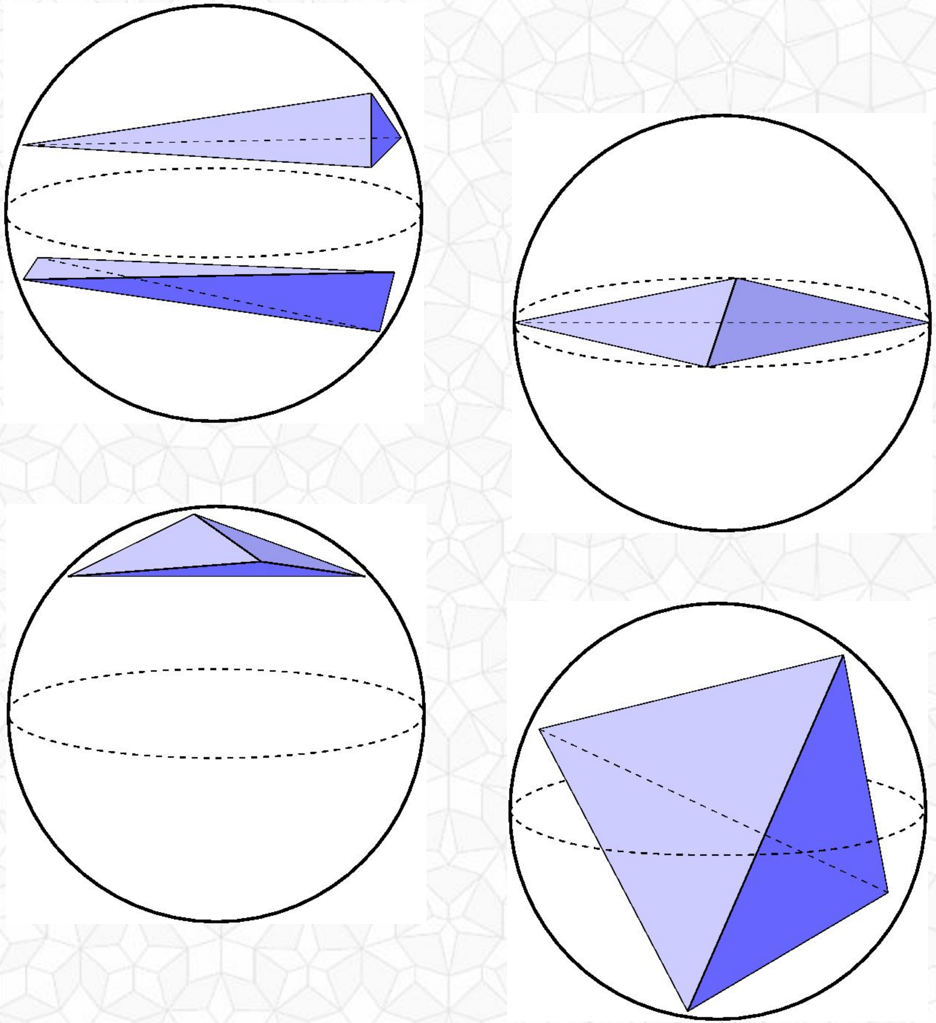
3D Mesh Operations

- Tetrahedral Swaps
- Edge Collapse
- Vertex Smoothing
    - Move a vertex to the centroid of its neighbors
    - Convex or concave, but avoid negative-volume elements
- Vertex Addition



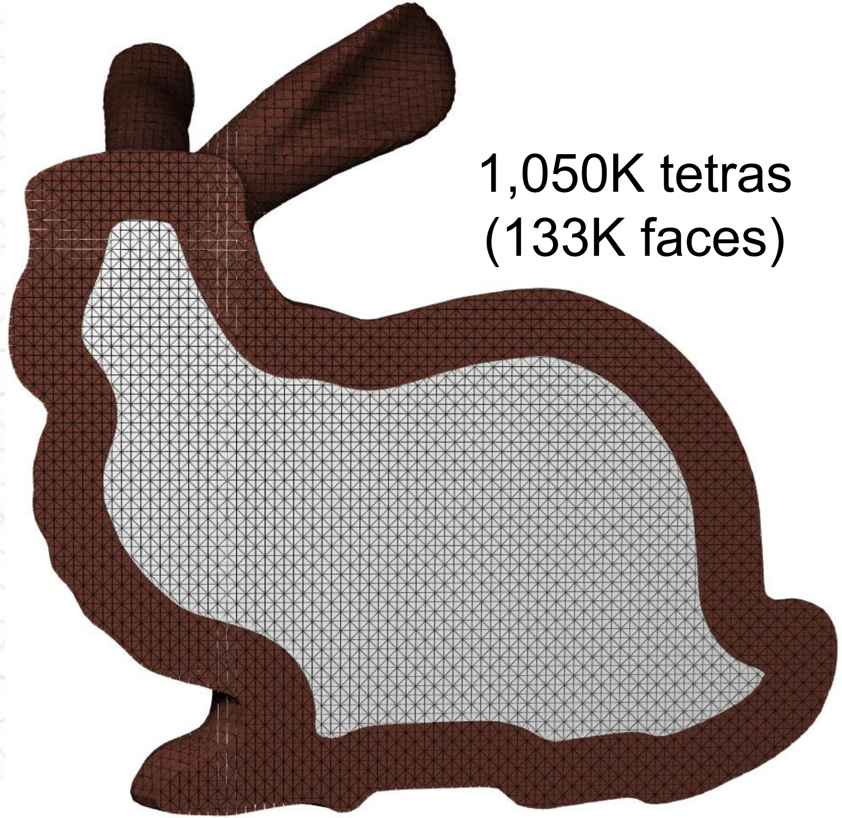
3D Mesh Operations

- Tetrahedral Swaps
- Edge Collapse
- Vertex Smoothing
- Vertex Addition
  - At the center of a tetra, face, or edge
  - Useful when mesh is simplified, but needs further element shape improvement

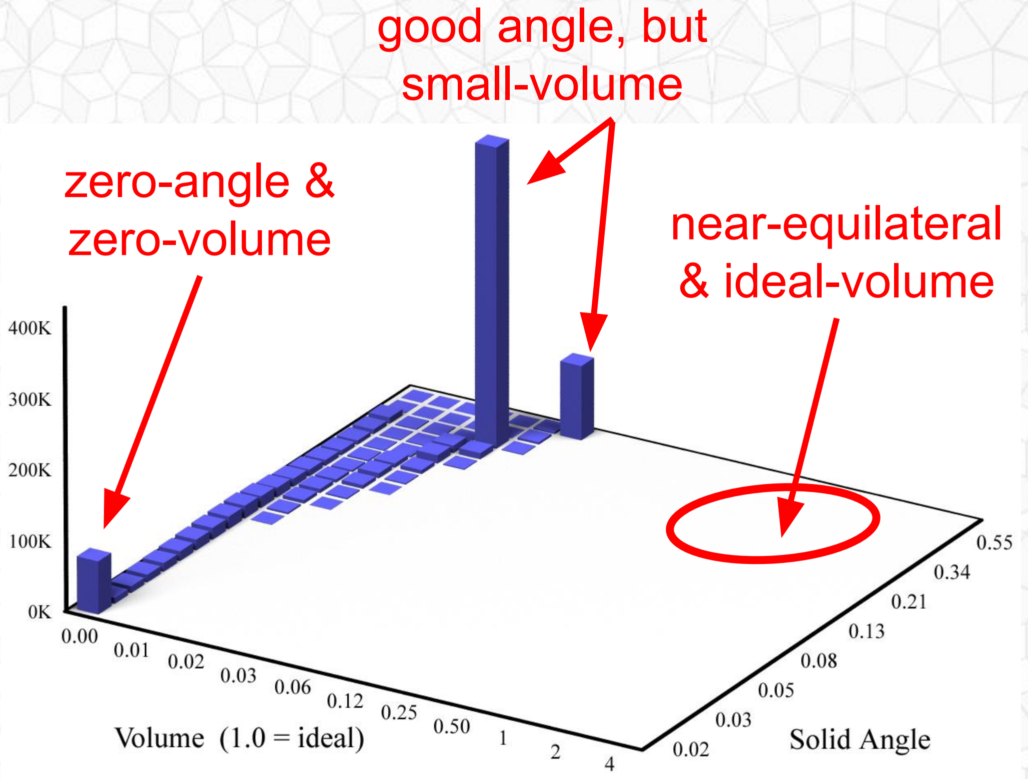


Visualization of Tetrahedra Quality

"Simplification and Improvement of Tetrahedral Models for Simulation"
Cutler, Dorsey, and McMillan
SGP 2004

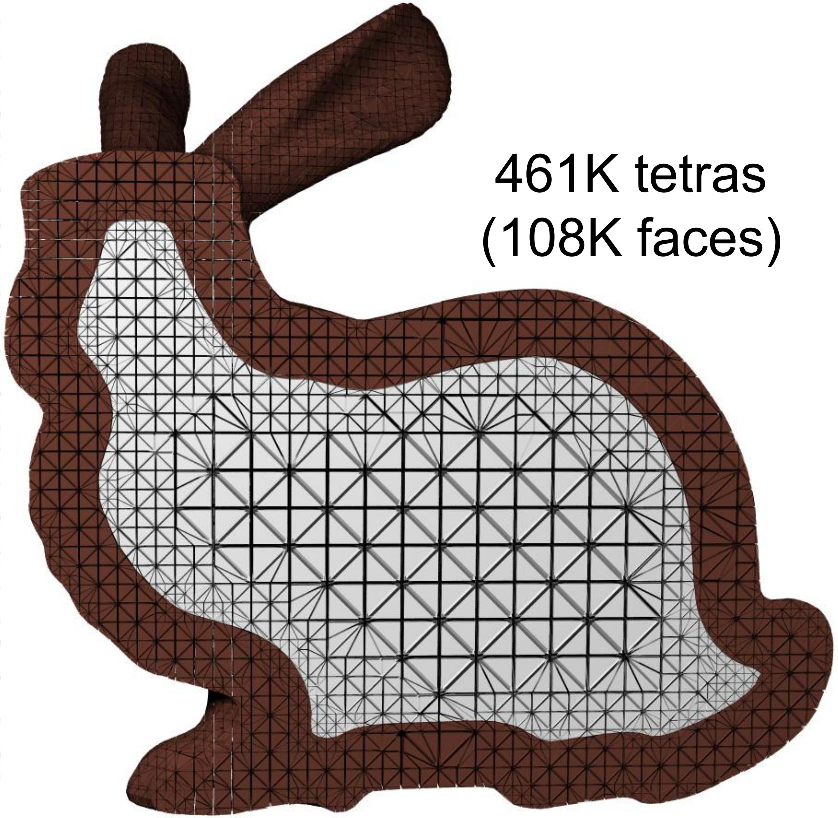


1,050K tetras
(133K faces)



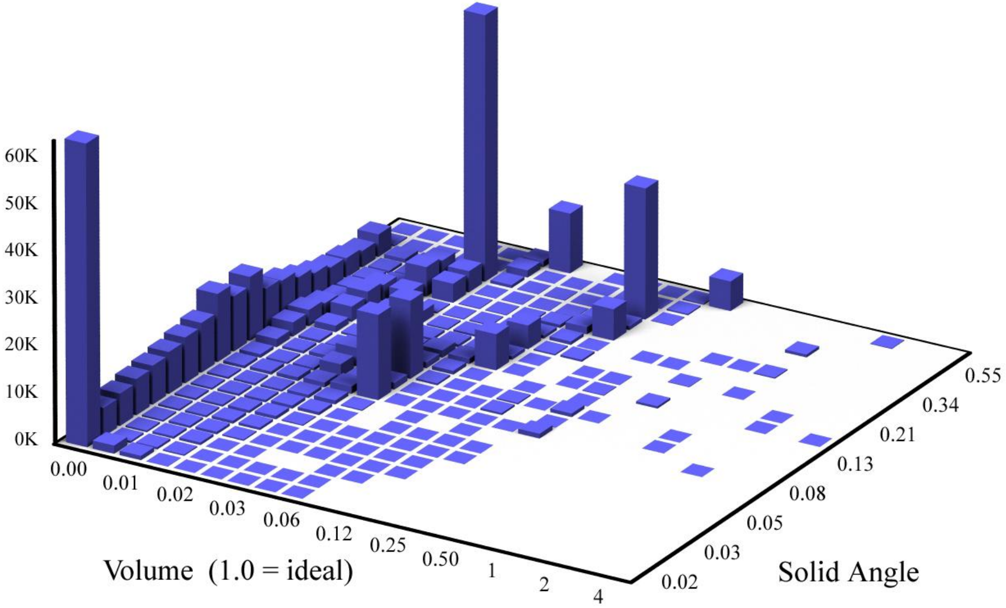
Visualization of Tetrahedra Quality

"Simplification and Improvement of Tetrahedral Models for Simulation"
Cutler, Dorsey, and McMillan
SGP 2004



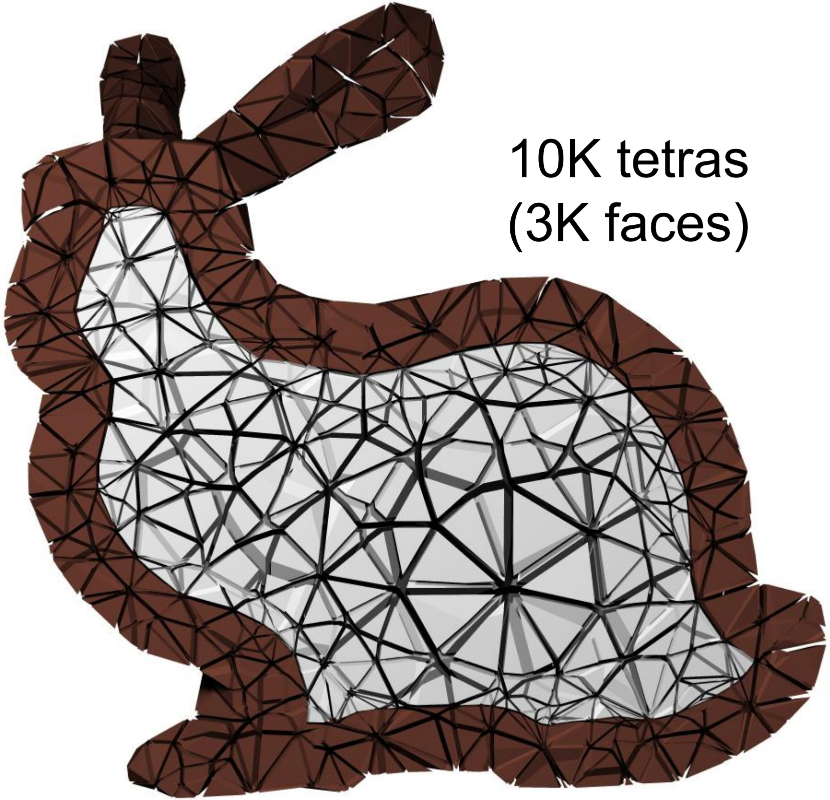
461K tetras
(108K faces)

Octree or Adaptive Distance Field (ADF)



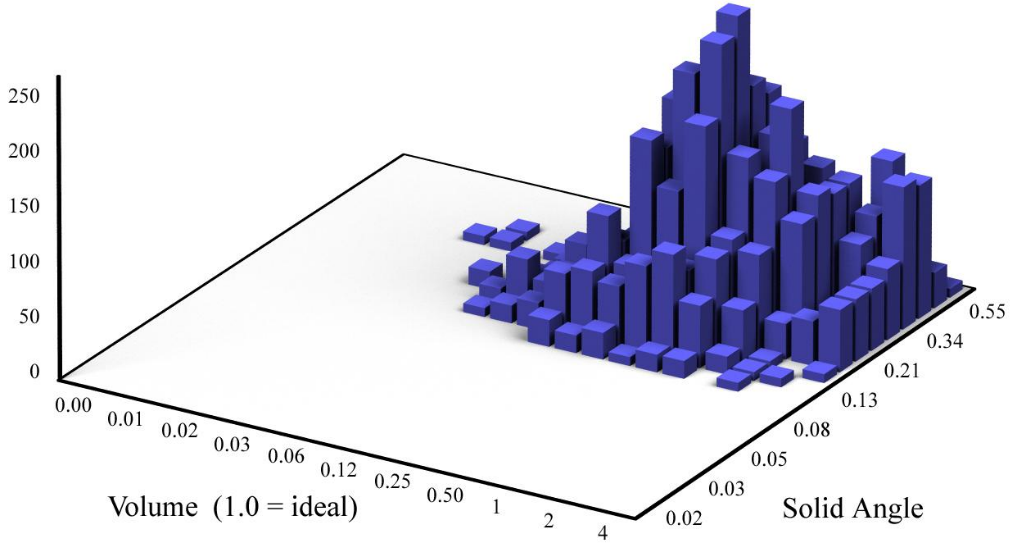
Visualization of Tetrahedra Quality

"Simplification and Improvement of Tetrahedral Models for Simulation"
Cutler, Dorsey, and McMillan
SGP 2004

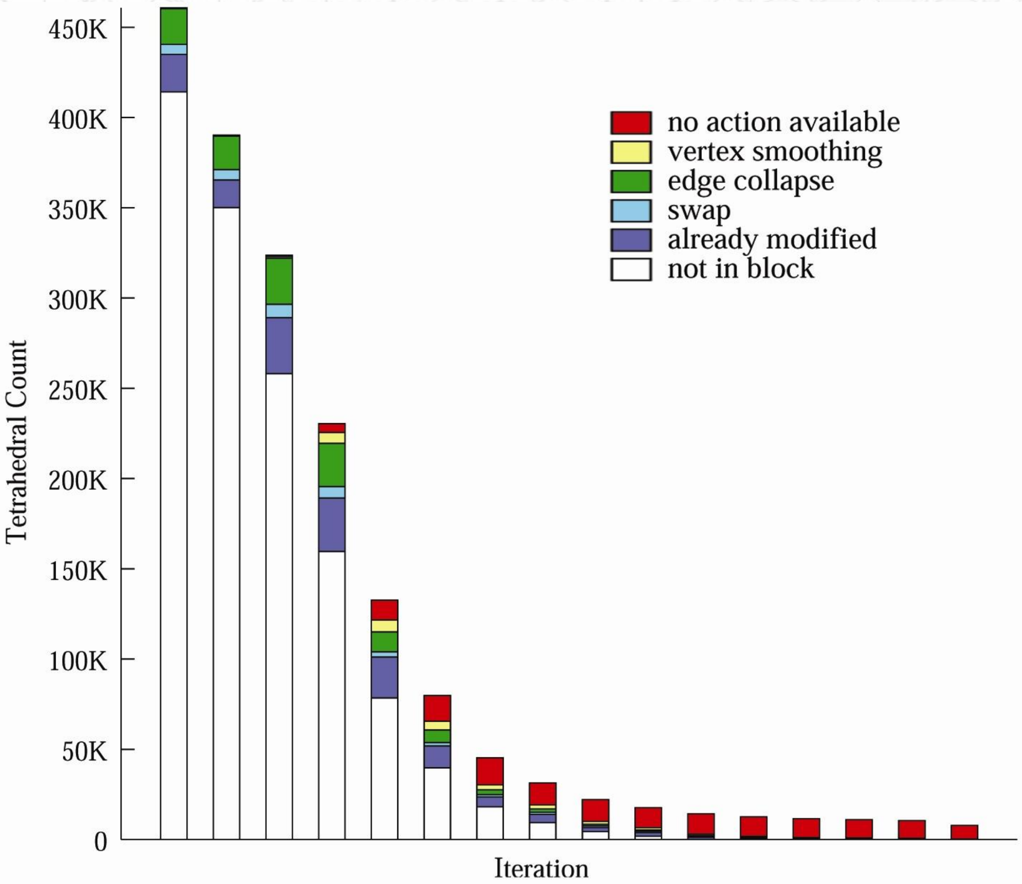


10K tetras
(3K faces)

After Simplification
& Mesh Improvement



Visualization of Simplification Algorithm

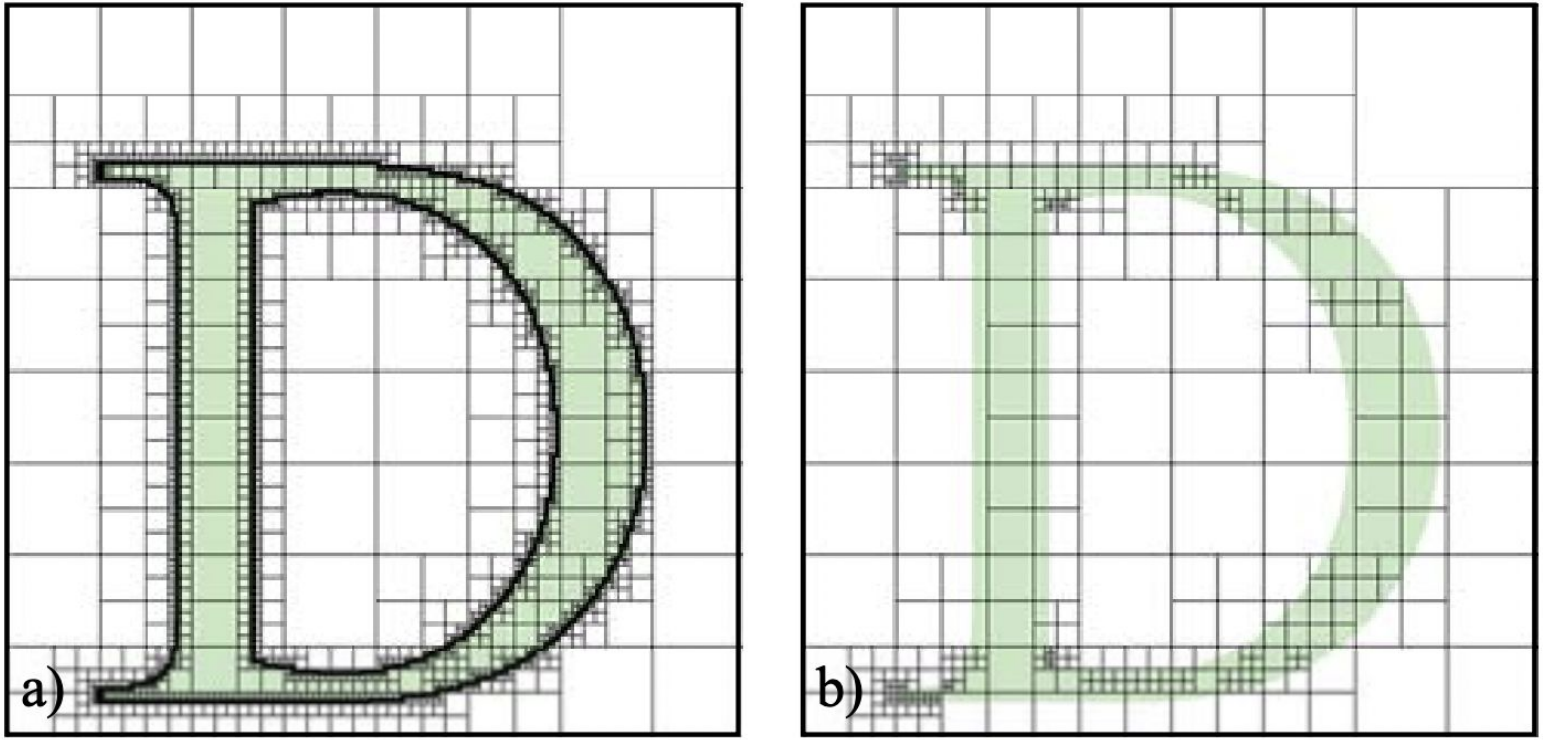


"Simplification and Improvement of Tetrahedral Models for Simulation"
Cutler, Dorsey, and McMillan
SGP 2004

Outline for Today

- Homework 5 Questions?
- Last Time: Windowing, Interval Trees & Segments Trees
- Motivation: FEM & CFD Simulation
- Uniform & Non-Uniform Meshing
- k-D Tree vs Quad Tree
- Maximum Depth, Number of Nodes
- Implicit Adjacency, Balanced Quad Tree
- Advanced Topics: √3 Subdivision & Octree Textures
- Remeshing for Interactive Deformation
- Next Time: Signed Distance Fields & Level Sets

Next Time: Signed Distance Fields & Level Sets



“Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics”,
Frisken, Perry, Rockwood, and Jones, SIGGRAPH 2001
“Designing with Distance Fields”, Frisken and Perry, 2006