



Rendering

Level of Detail

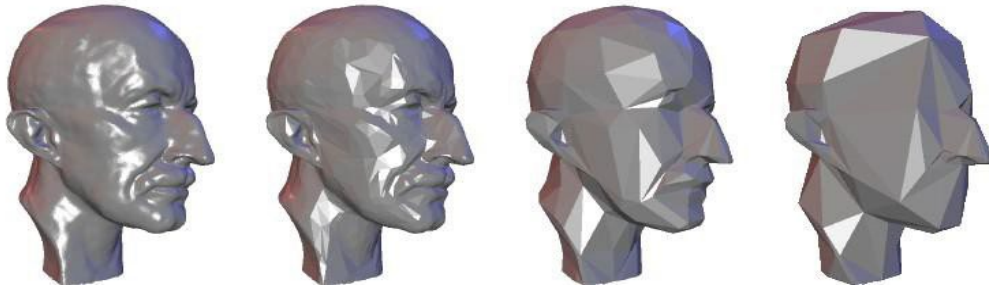
CS 415: Game Development

Professor Eric Shaffer



Run-Time LOD

- Construct multiple versions of mesh
 - Varying polygon count
- Multi-resolution hierarchies enable
 - efficient geometry processing
 - level-of-detail (LOD) rendering



Using a set of lower poly count models is “discrete level of detail”

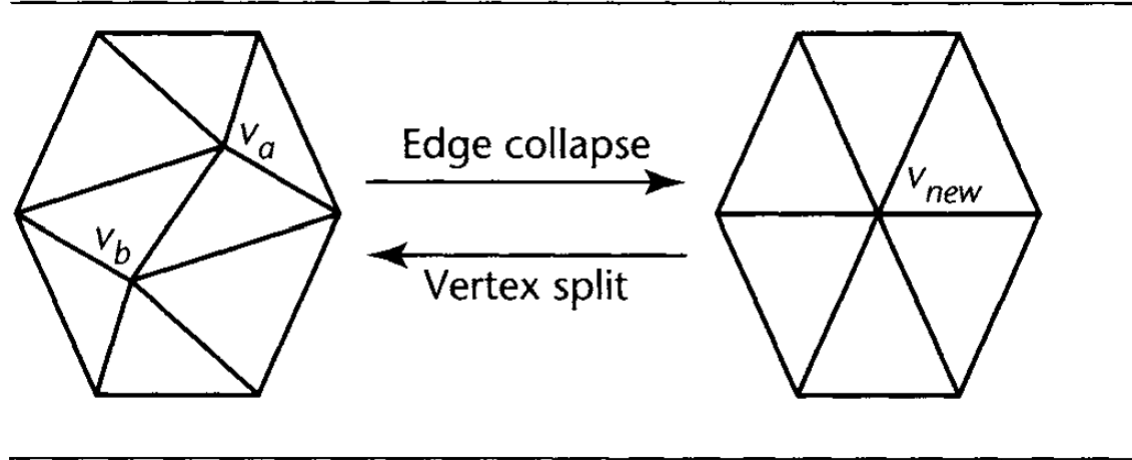
Model chosen at run-time based on some formula

- e.g. projected screen space

Using discrete level of detail can result in visual artifacts
(i.e. stuff that look bad)

Any guesses as to the most common artifact?

Mesh Simplification



Most common approach:

- Collapse an edge
- Removes 2 triangles per collapse
- Choose collapse the induces least error
- Generate new vertex position that is results in least error
- Repeat until target triangle count is reached

Continuous Level of Detail

...not really continuous

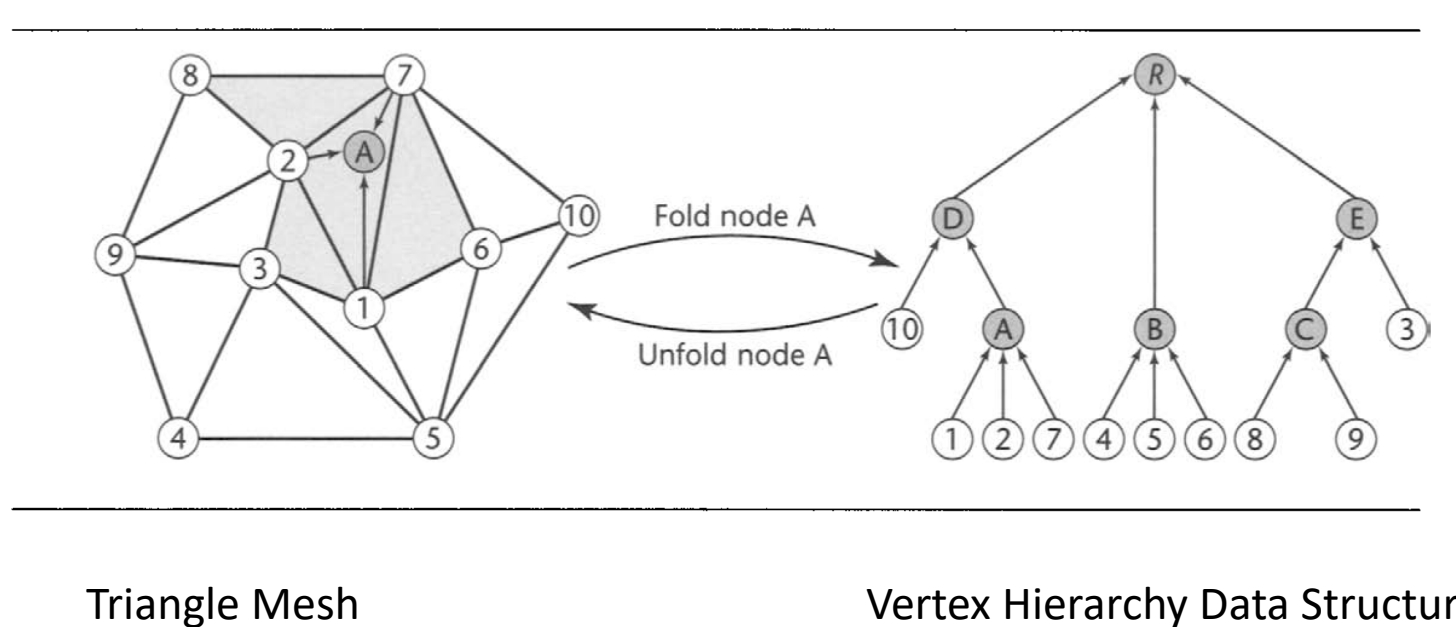
Adjusts triangle count at run-time

Makes incremental changes, smaller changes than discrete lod

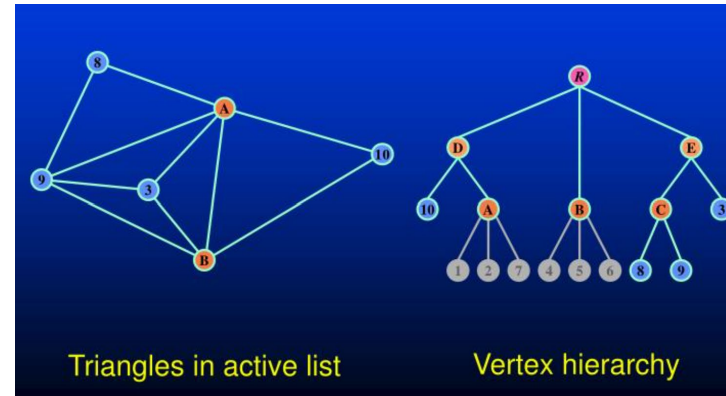
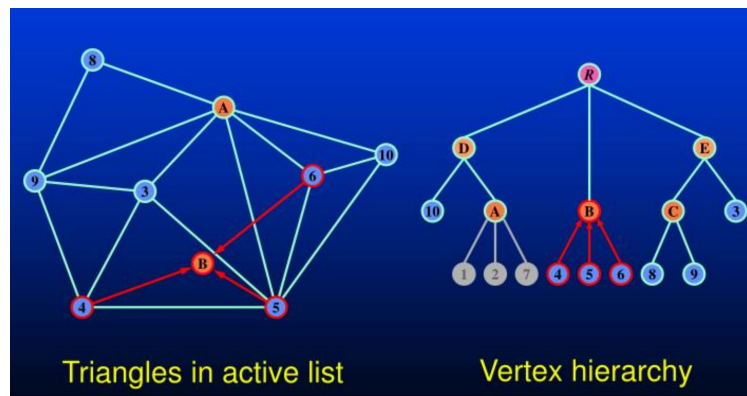
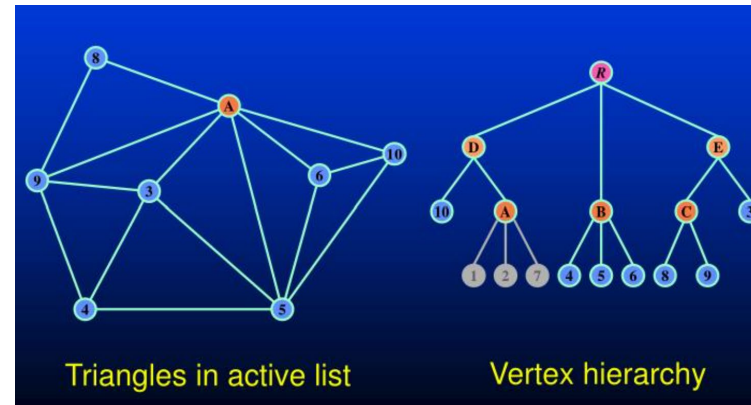
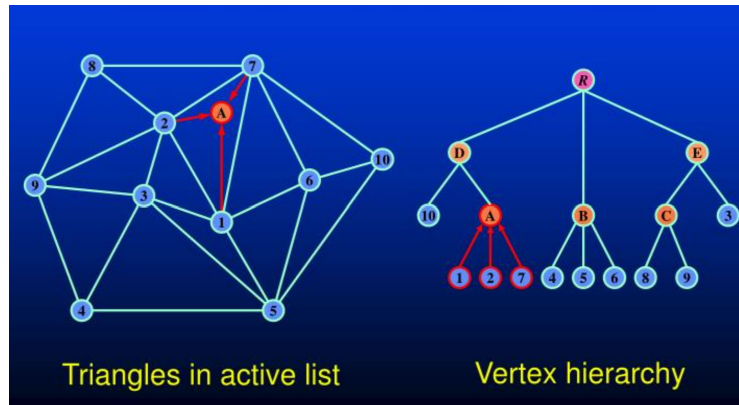
Like performing a single edge collapse based change in screen size

How Does CLOD Work?

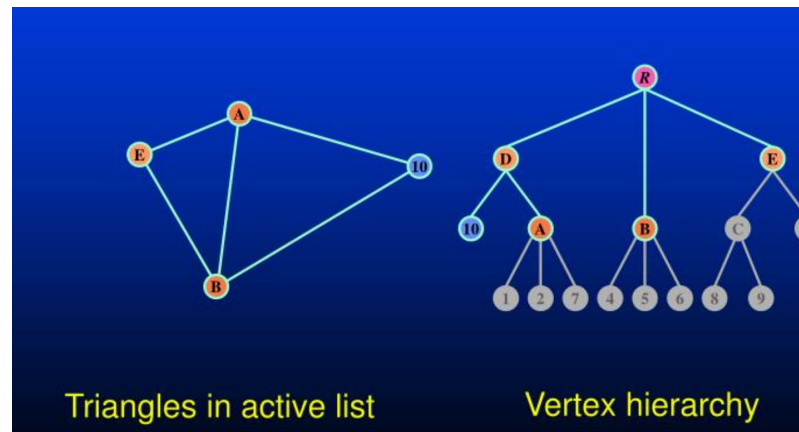
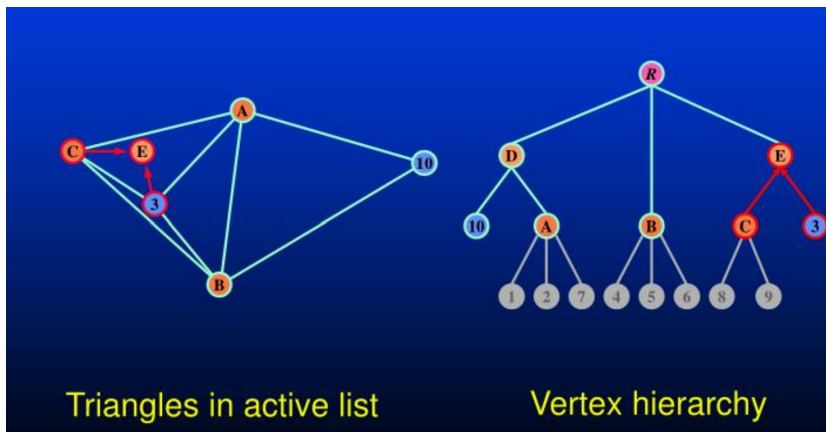
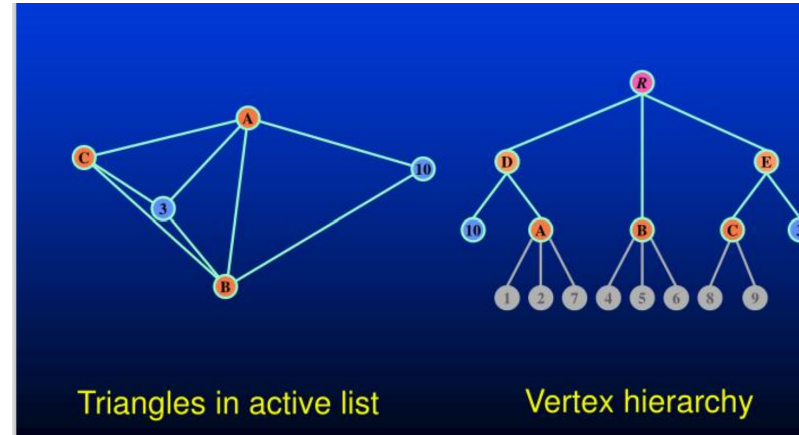
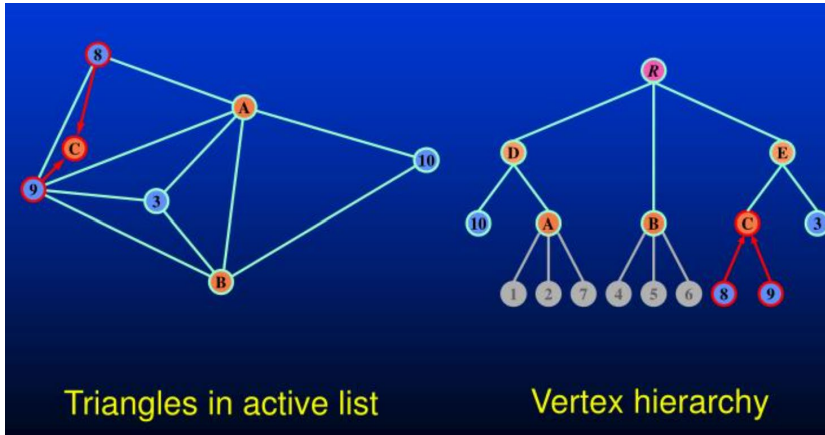
- Precompute a vertex hierarchy that encodes possible edge collapses
- Sometimes these collapses are batched for better performance



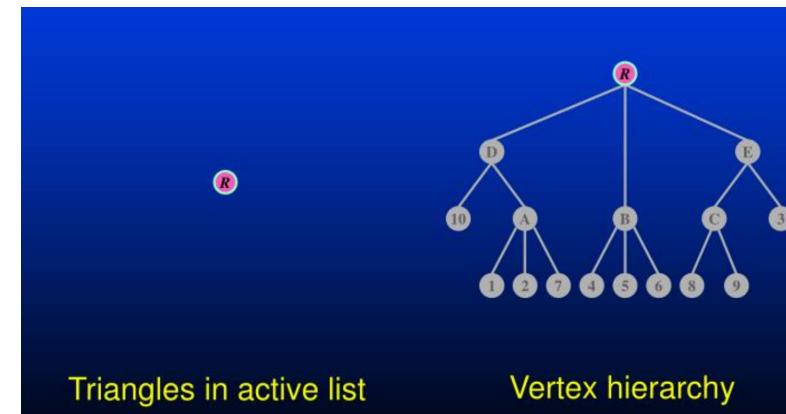
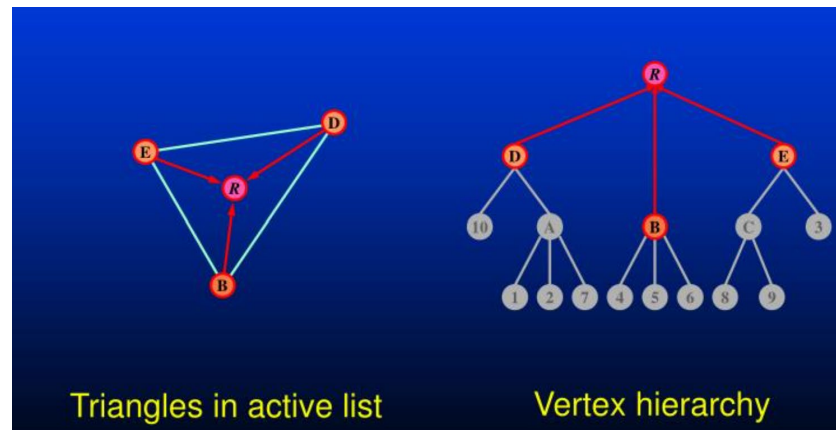
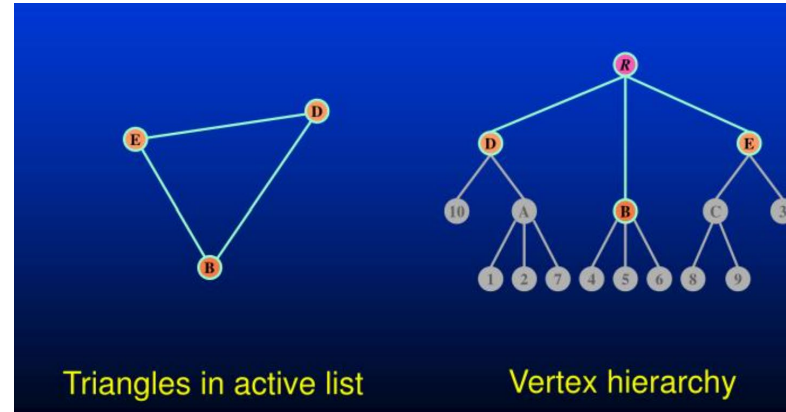
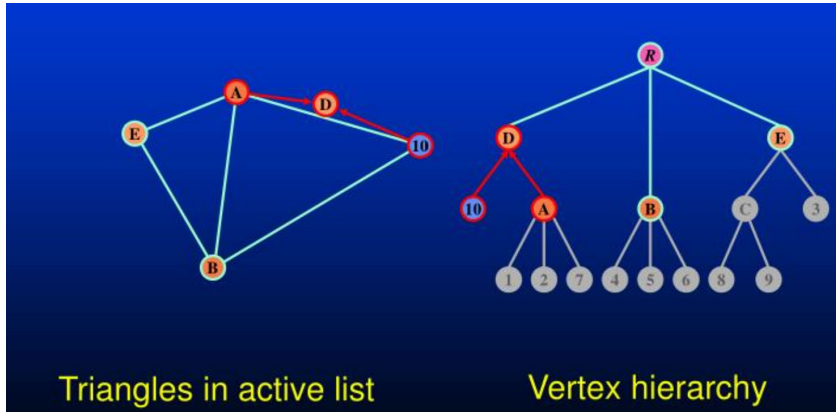
Detailed Example



Detailed Example



Detailed Example



Using the Vertex Hierarchy

A cut through the hierarchy corresponds to simplified mesh

Cut is adjusted by edge collapses and vertex splits based on viewing conditions

