Scene Graphs

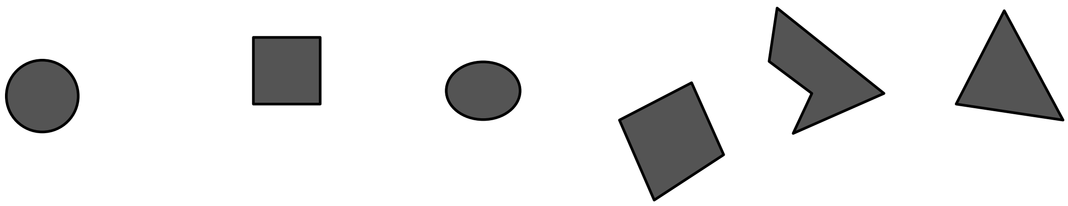
Thomas A. Funkhouser

Today's Lecture

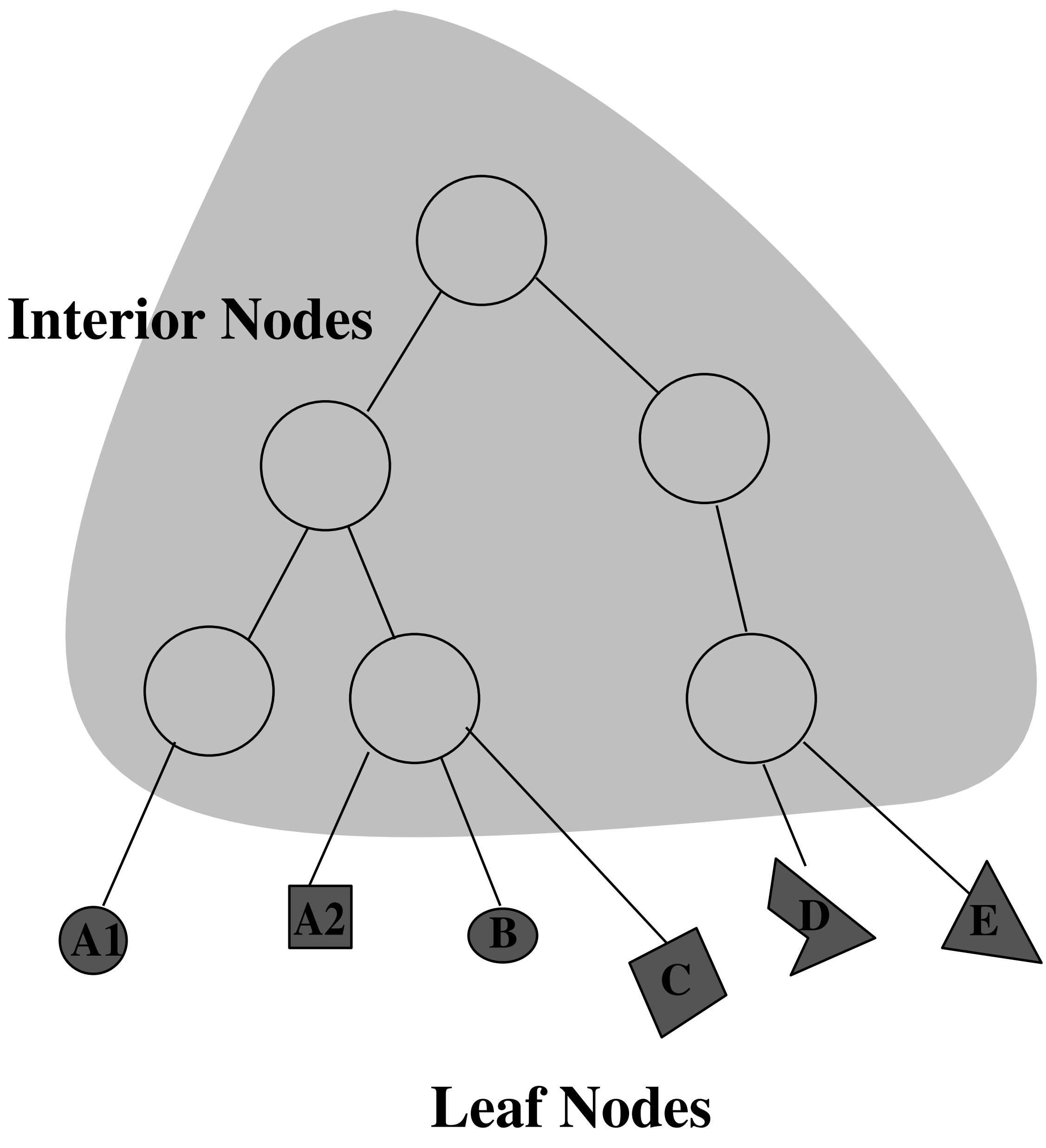
- **Scene Graph Overview**
  - Primitives
  - Operations
  - Considerations
- **Scene Graph Optimization**
  - Function
  - Bounding volumes
  - Levels of Detail
  - Attributes
  - Lights
- **Thought Exercise**

Scene Description

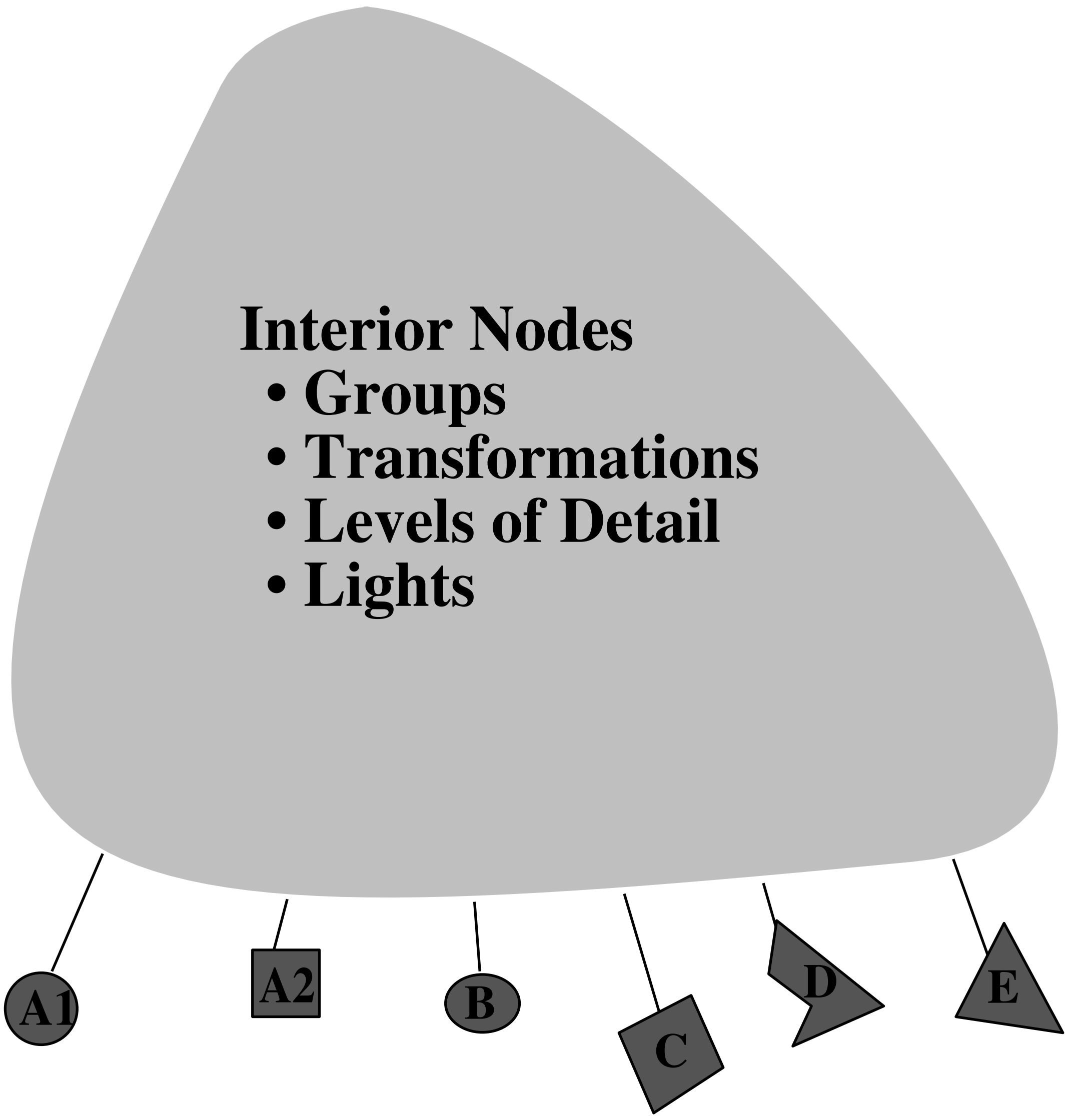
- **Set of Primitives**
- **Specify for each primitive**
  - Transformation
  - Lighting attributes
  - Surface attributes
    - Material (BRDF)
    - Texture
    - Texture transformation



Scene Graph

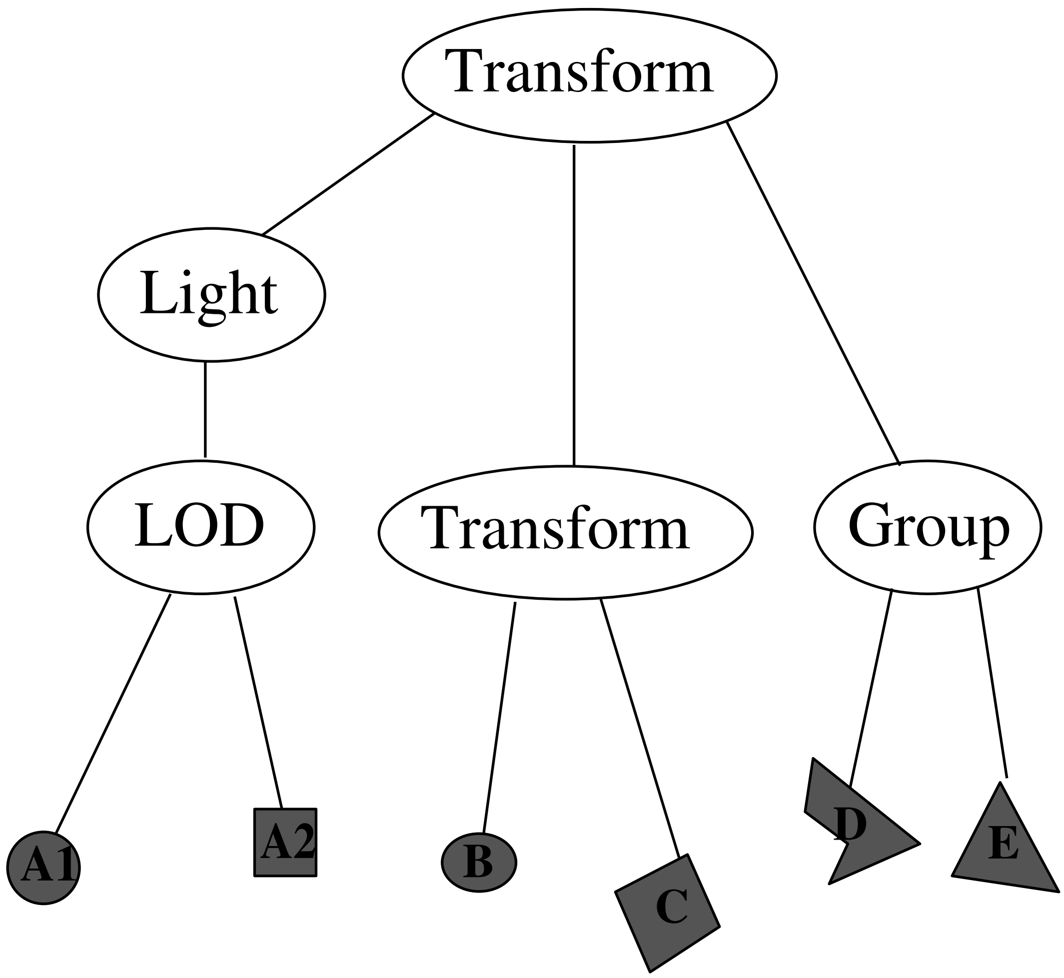


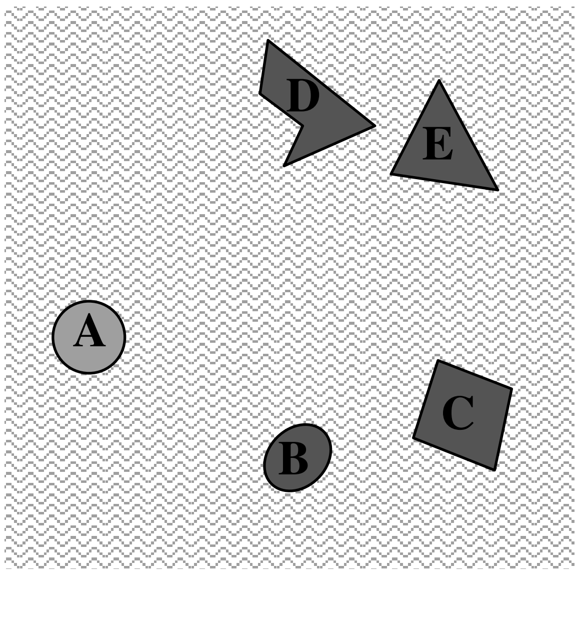
Scene Graph (VRML 2.0)



Leaf Nodes
- **Geometry with surface attributes**

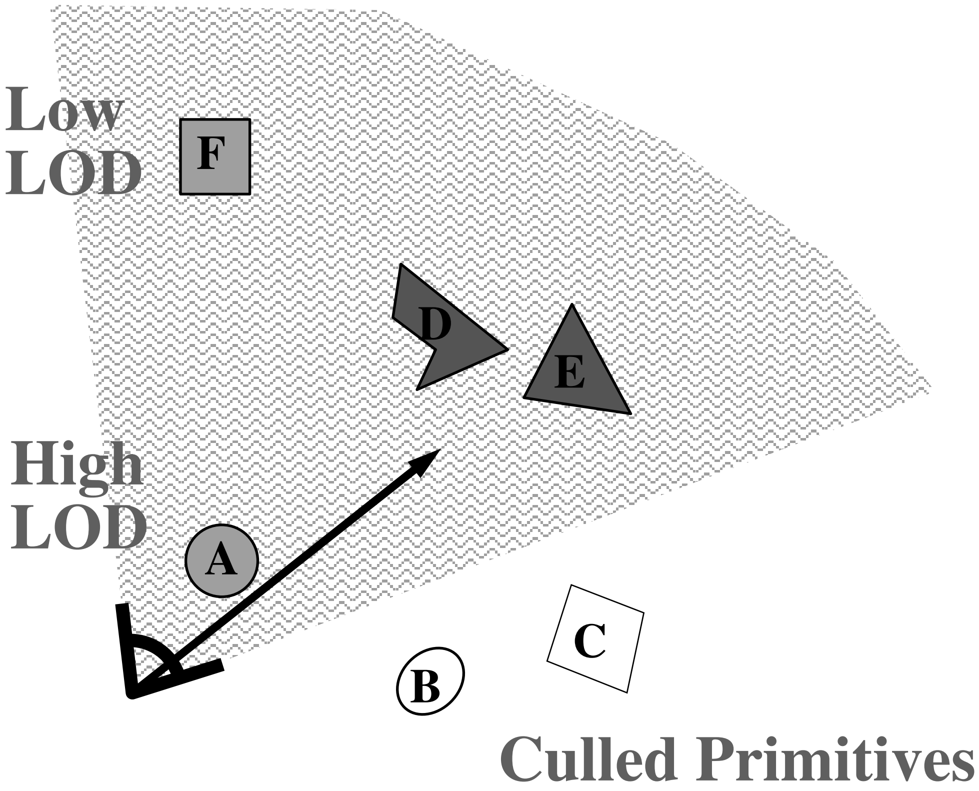
Example Scene Graph





Scene Graph Traversal

- **Simulation**
  - Animation
- **Intersection**
  - Collision detection
  - Picking
- **Image Generation**
  - Culling
  - Detail elision
  - Attributes

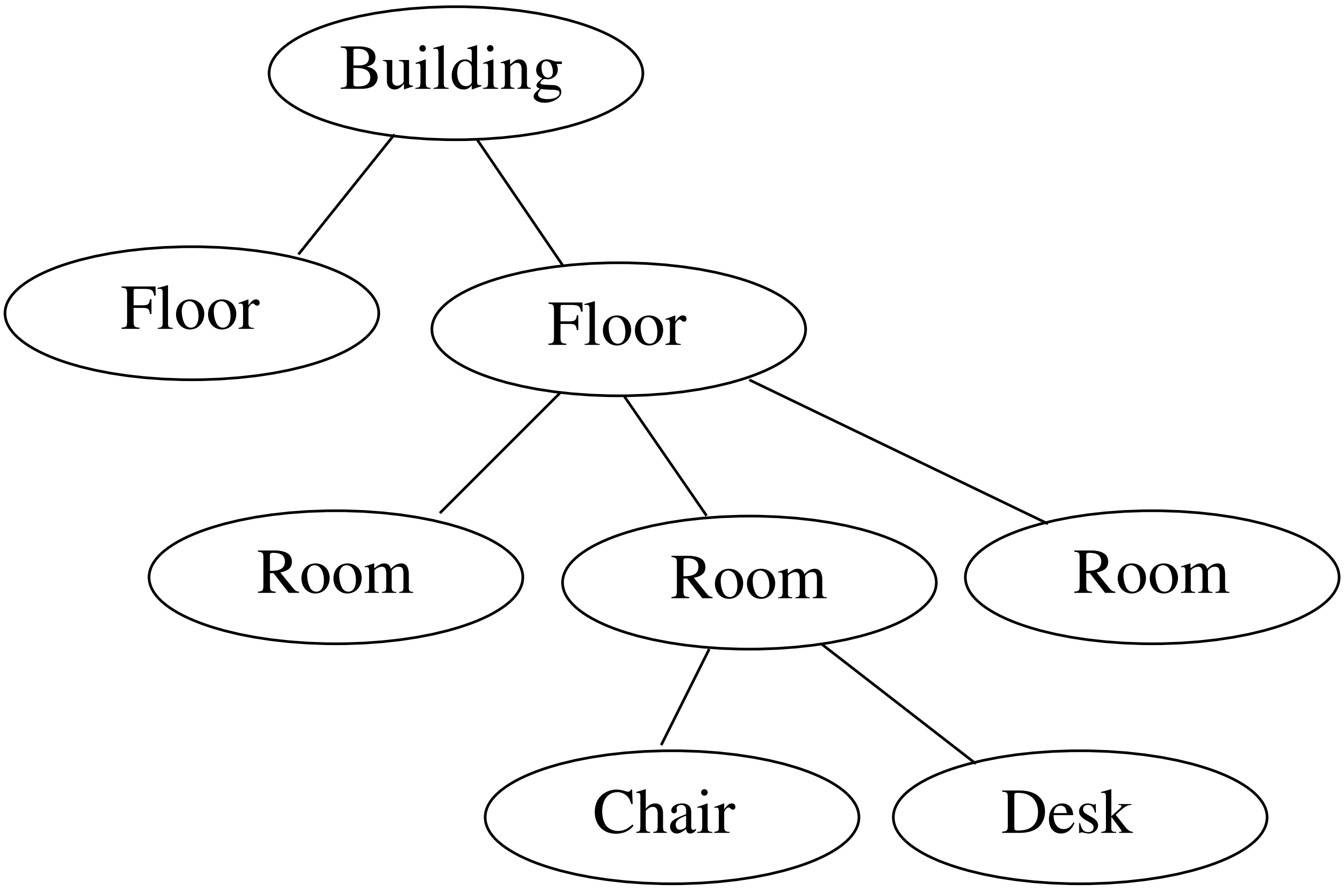


Scene Graph Considerations

- **Functional Organization**
  - Semantics
- **Bounding Volumes**
  - Culling
  - Intersection
- **Levels of Detail**
  - Detail elision
  - Intersection
- **Attribute Management**
  - Eliminate redundancies

Functional Organization

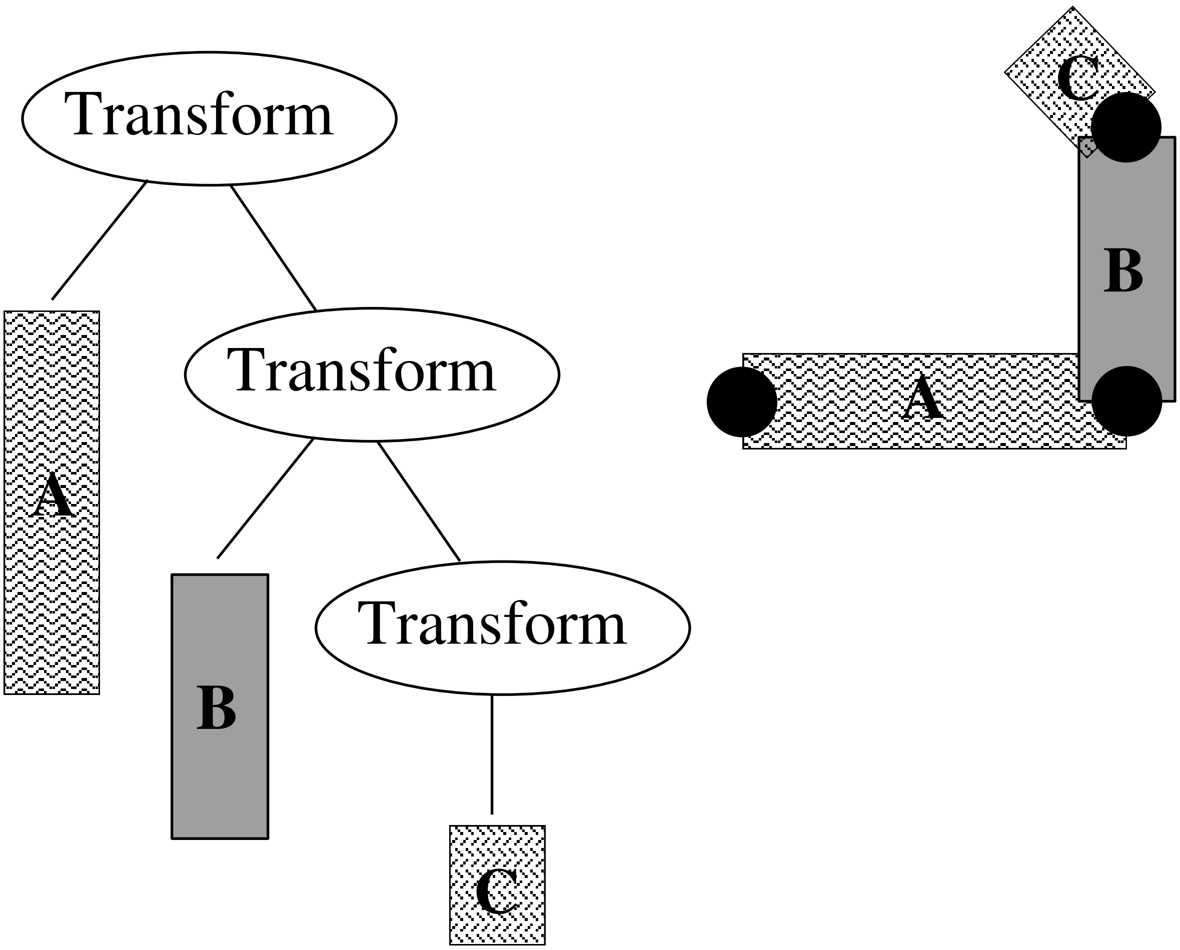
- **Semantics:**
  - Logical parts
  - Named parts



Functional organization
is only for human –> Optimize

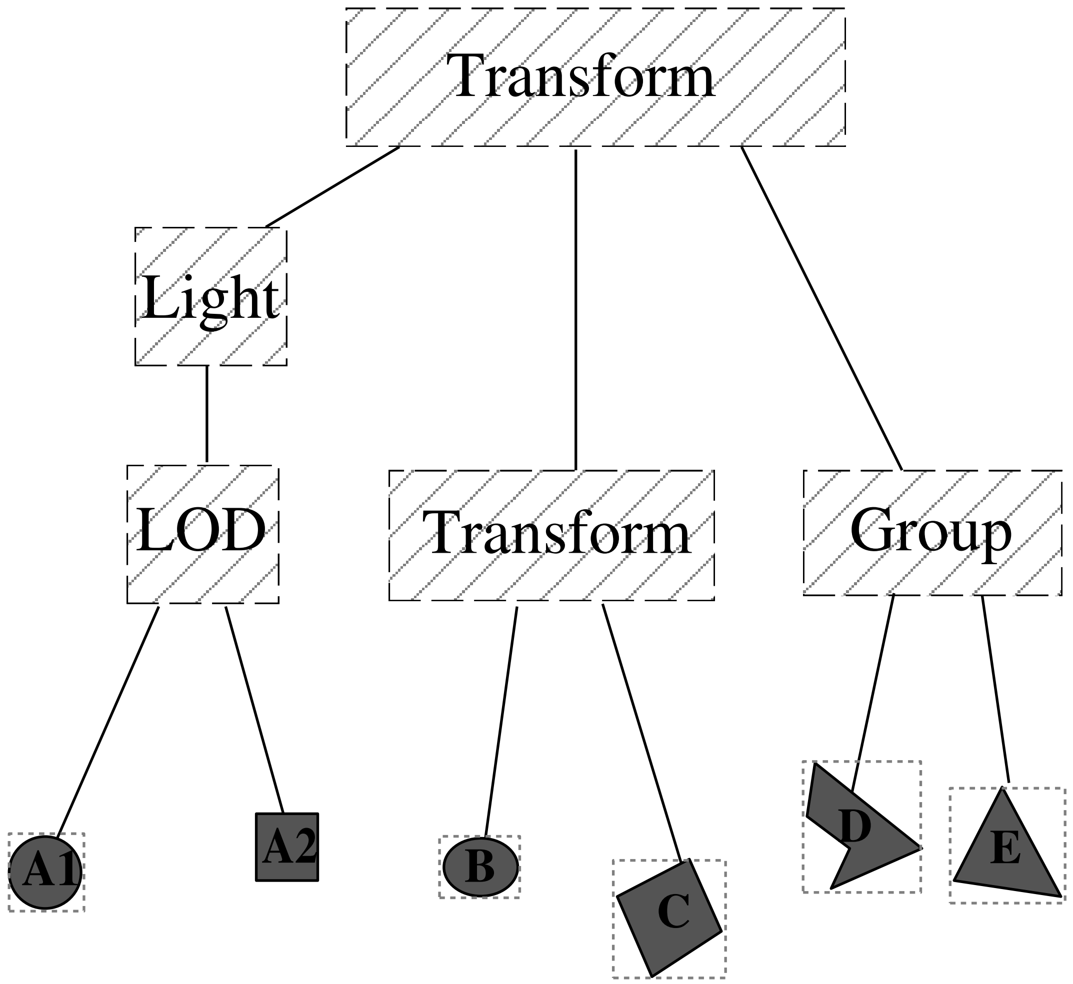
Functional Organization

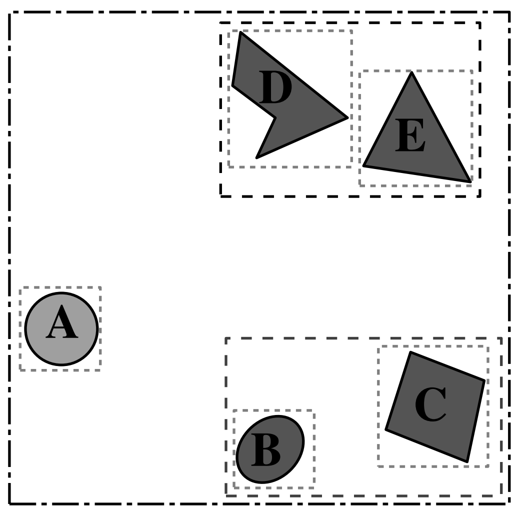
- **Articulated Transformations**
  - Animation



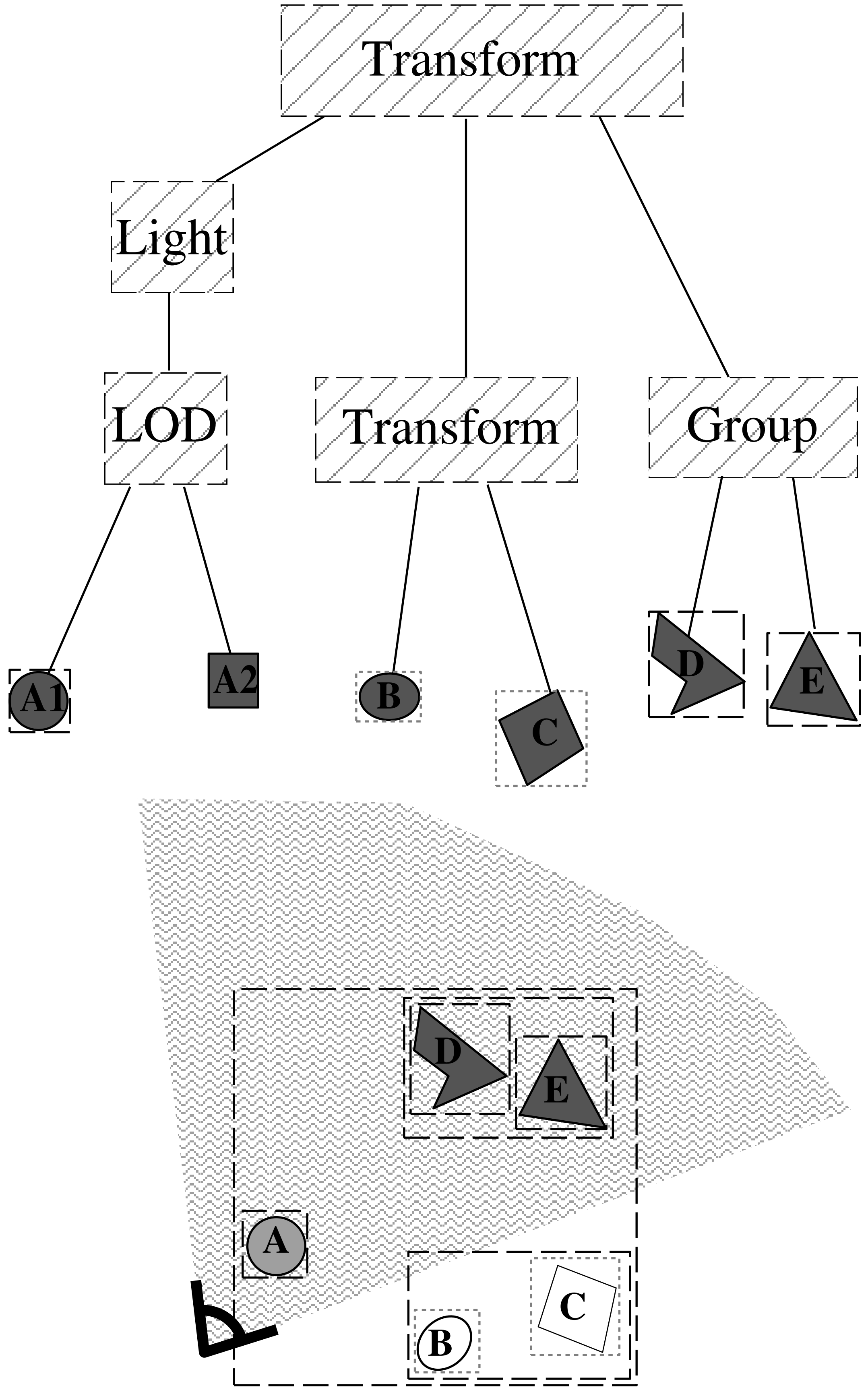
Difficult to optimize animated objects

Bounding Volume Hierarchies

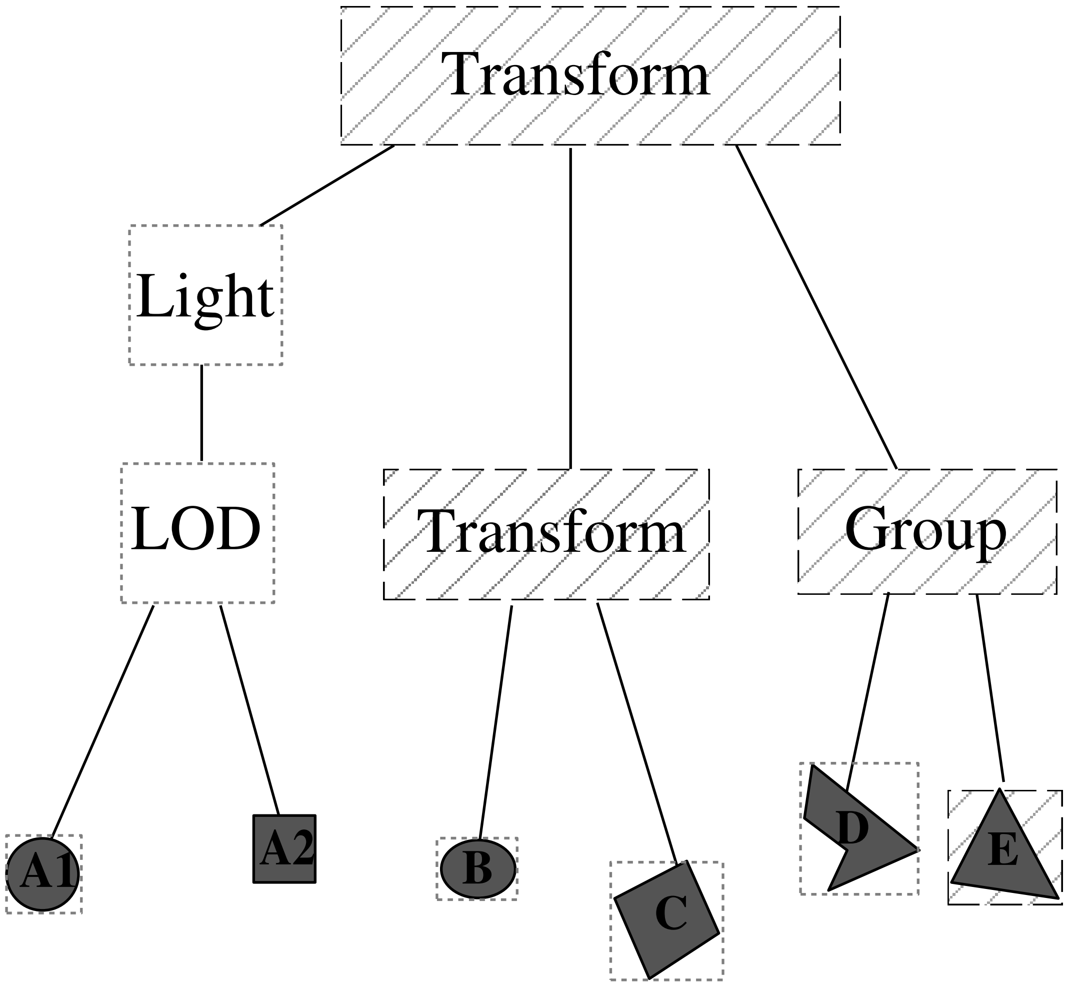


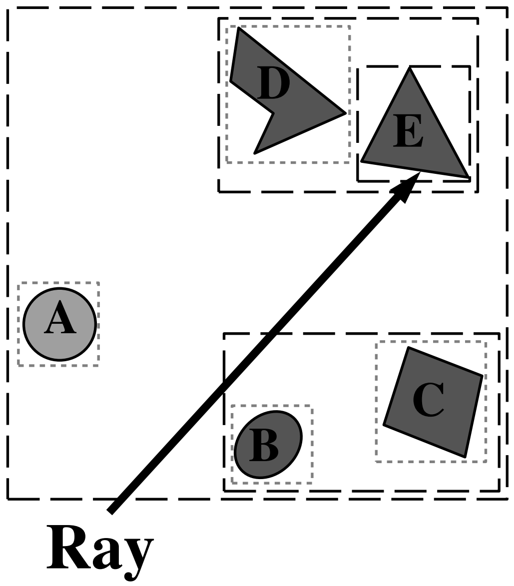


View Frustum Culling



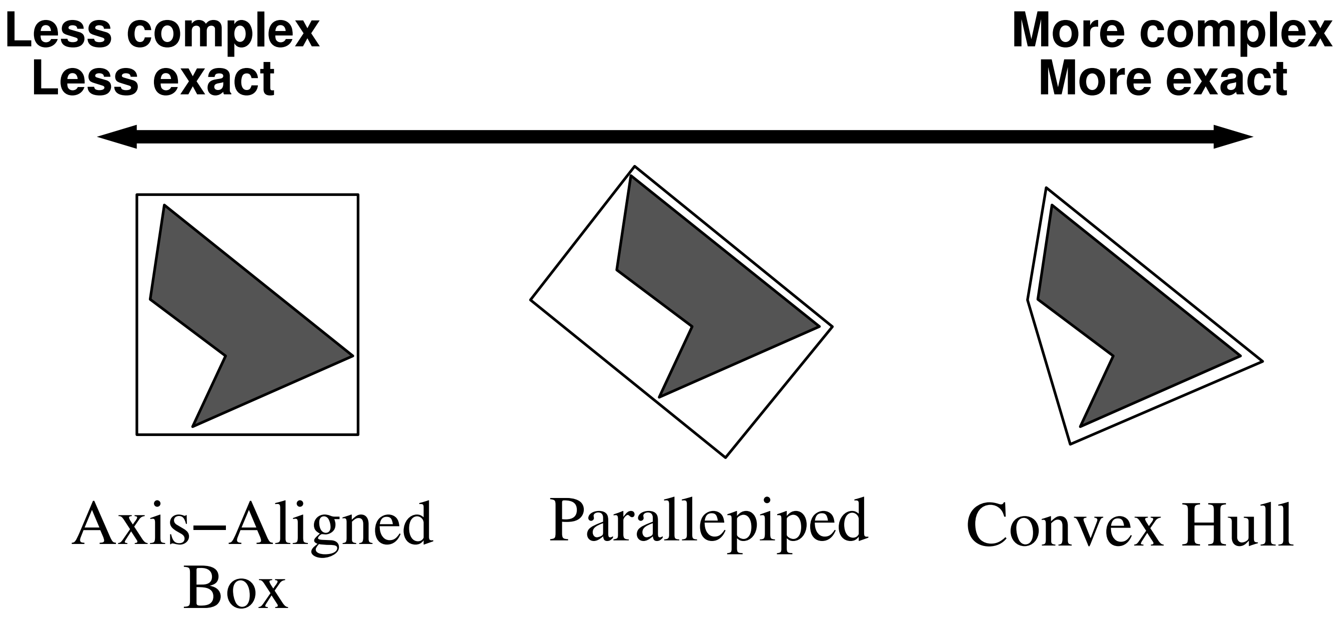
Ray Intersection





Hierarchy Construction

1. **Choose Bounding Volume(s)**
   - Spheres
   - Boxes
   - Parallelpipeds
   - Oriented boxes
   - Ellipsoids
   - Convex hulls



Hierarchy Construction

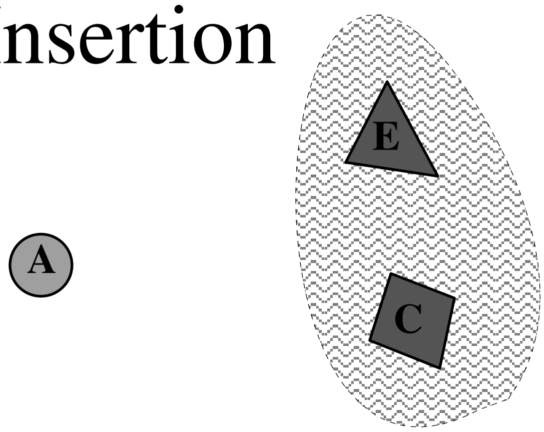
2. **Construct Hierarchy of Objects**
   – According to heuristics

- **Ray Tracing Heuristics** **(Kay & Kajiya)**
  – Subtrees should contain objects near each other.
  – The volume of each node should be minimal.
  – The sum of the volume of all bounding volumes should be minimal.
  – Properties of nodes near root should be weighted more.
  – Time spent constructing the tree should more than pay for itself.

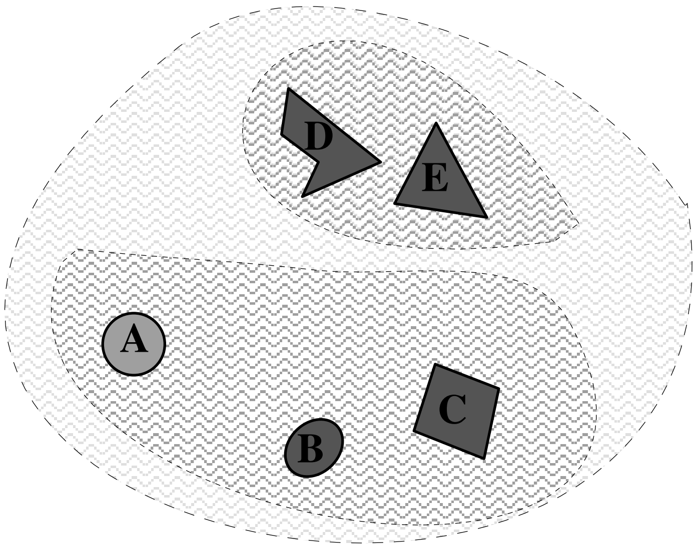
- **Radiosity Heuristics** **(Sillion & Drettakis)**
  – Proximity
  – Homogeneity

Algorithms

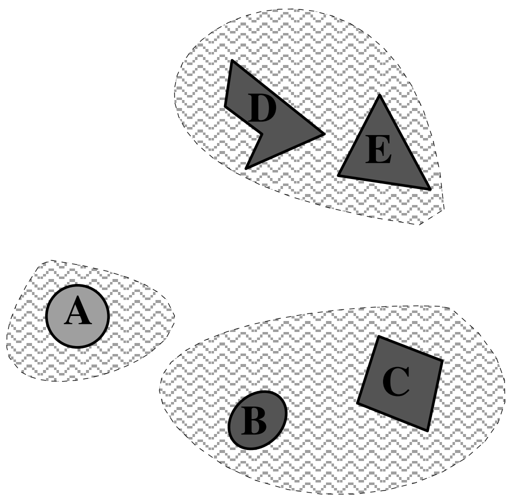
- Greedy Insertion



- Recursively partition objects

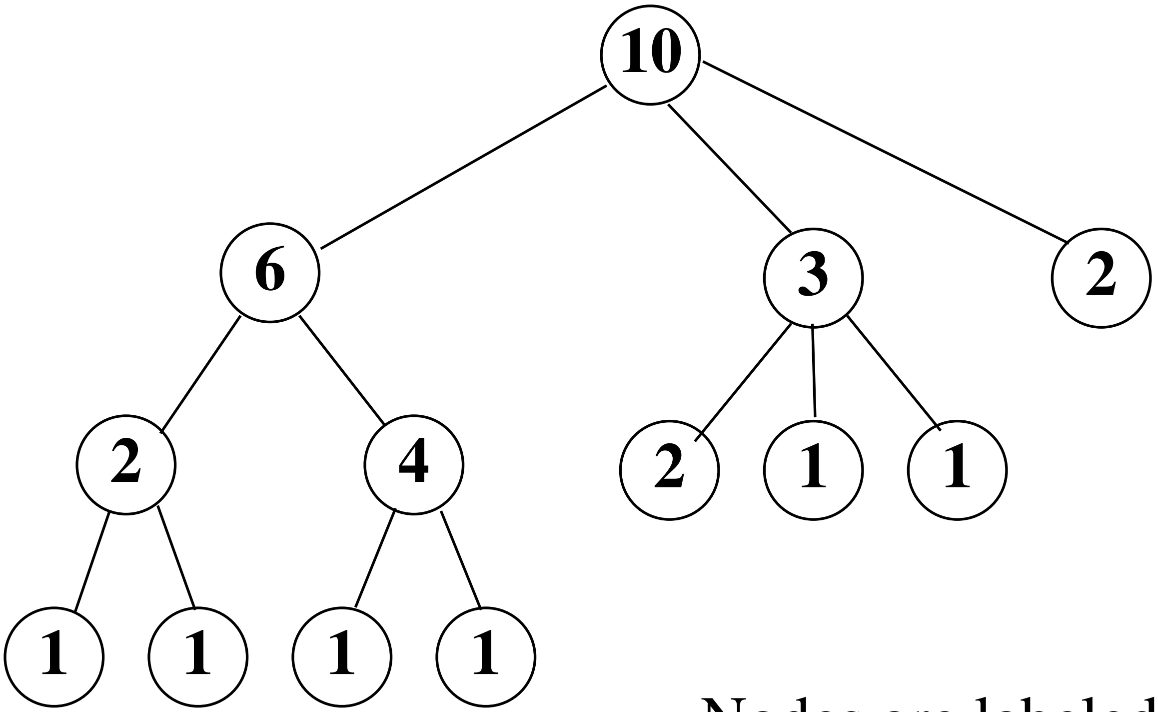


- Recursively group objects



Greedy Insertion Algorithm

- **Goldsmith & Salmon**
  - Insert each object into hierarchy to minimize some metric
  - Sensitive to order of objects



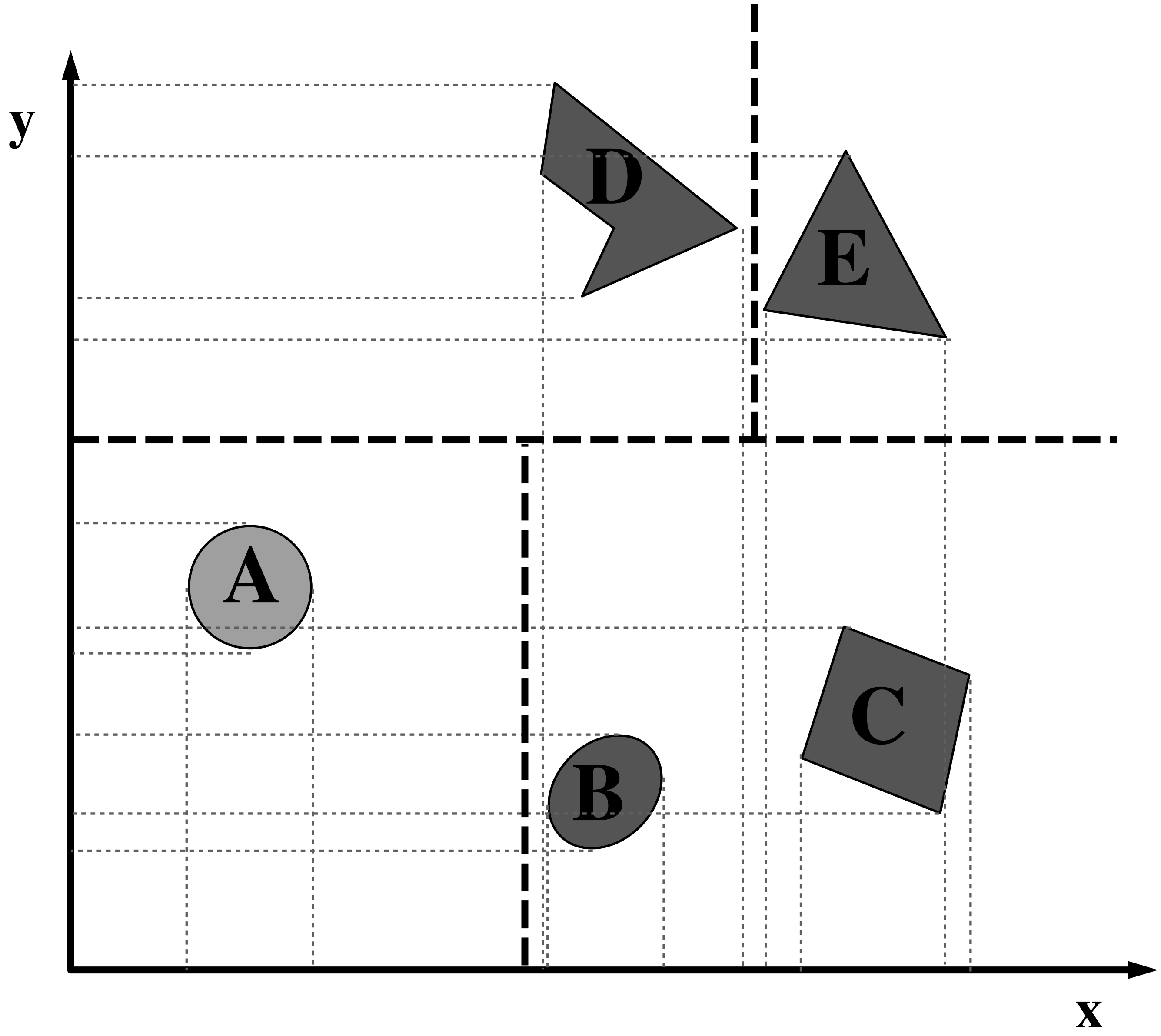
Nodes are labeled by likelihood they are intersected by a ray

Expect 7.3 intersections per ray

(1) + (3) + (.6*2 + .3*3) + (.2*2 + .4*2)

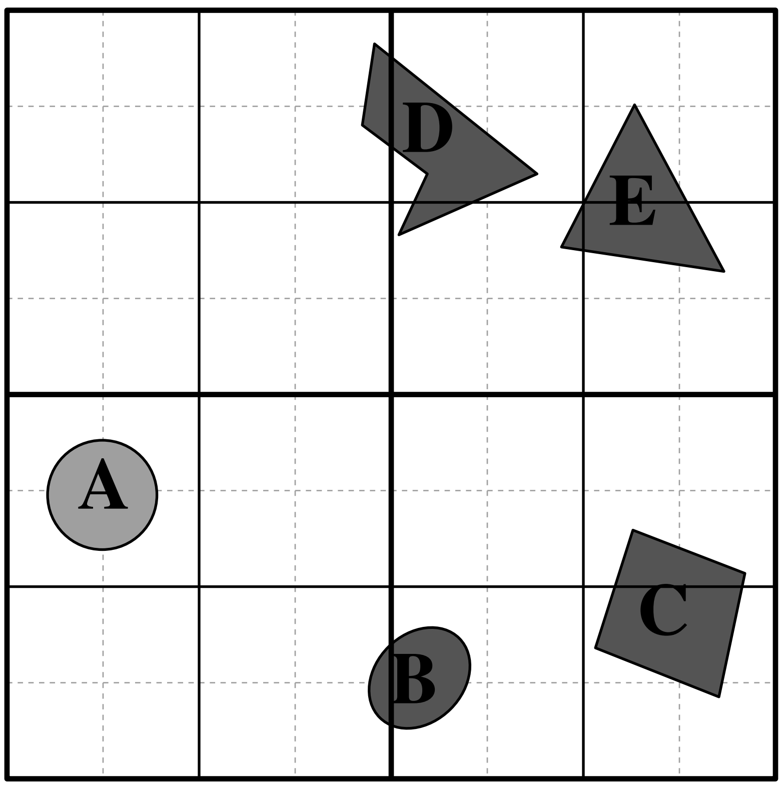
Median Cut Algorithms

- **Kay & Kajiya ('86)**
  - Median cut in X, then Y, then X, ...
- **Shade et al. ('96)**
  - Search for 'best' gaps for a small set of plane orientations



Octree Grouping Algorithm

- **Sillion & Drettakis ('95)**
  - Load objects into octree
  - Group subtrees residing in the same cell at the same level

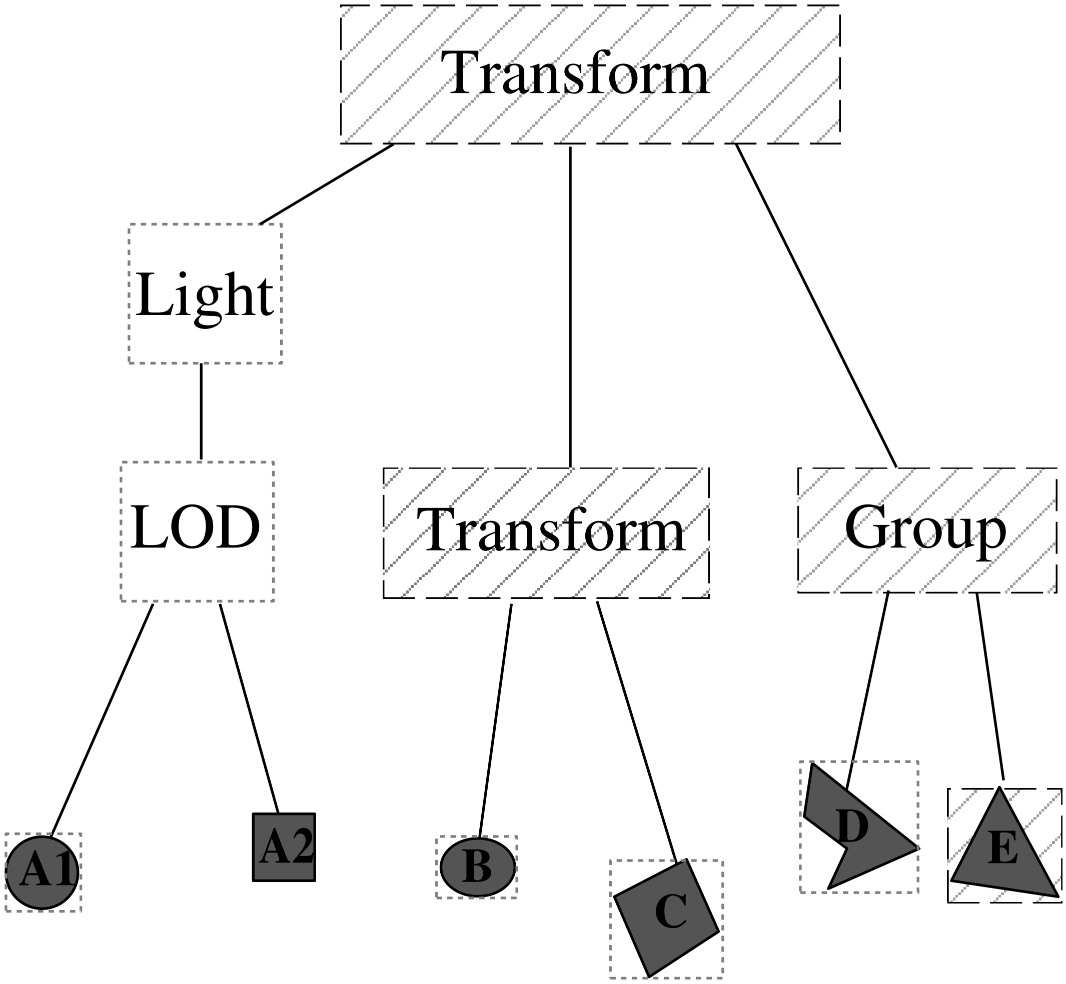


Open Question #1

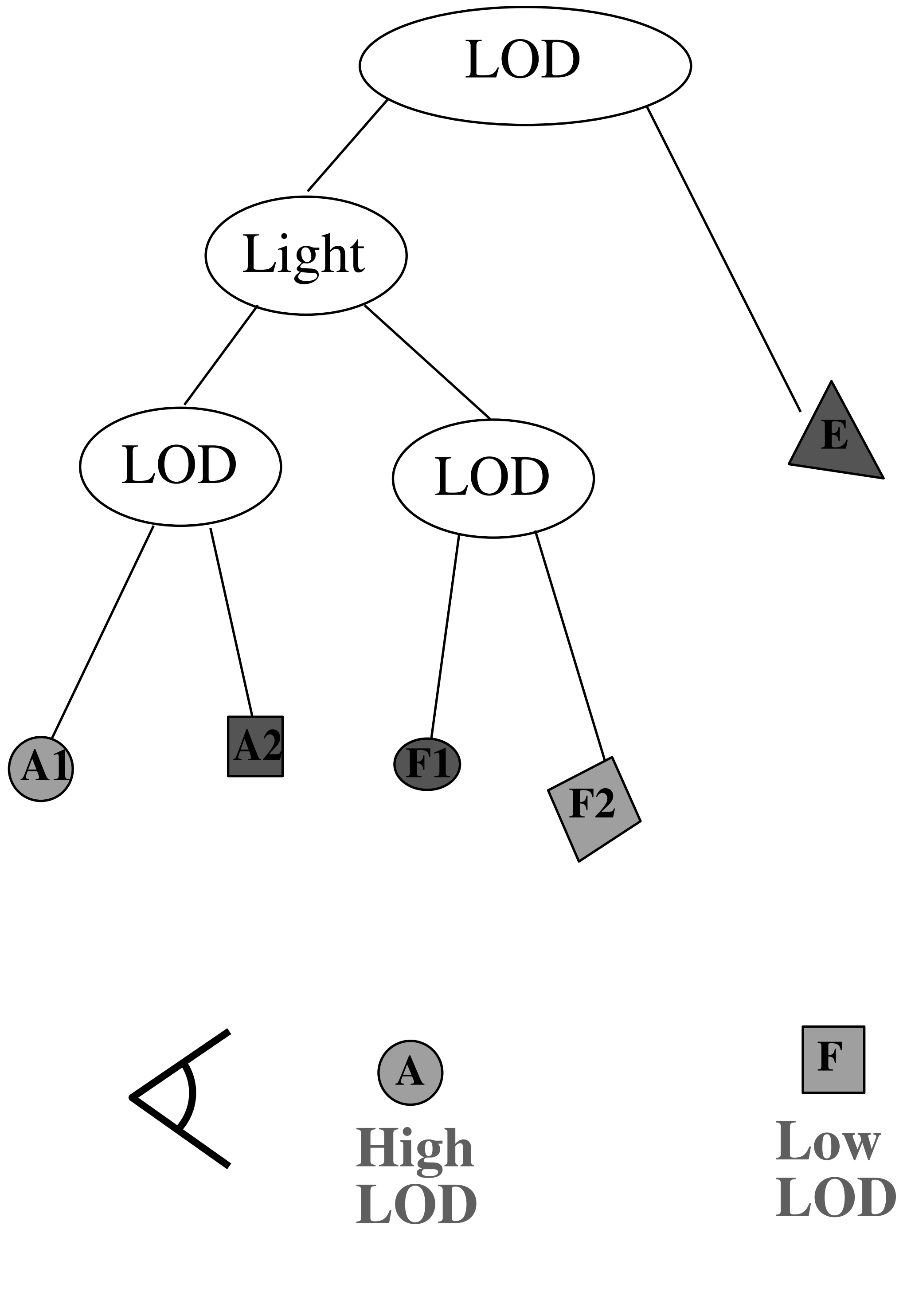
- **Is there a bounding volume construction algorithm for which we can prove worst–case bounds on goodness?**
  - Graph theory?

Open Question #2

- **How can we choose which primitives should be the leaves of a bounding volume hierarchy?**



Levels of Detail



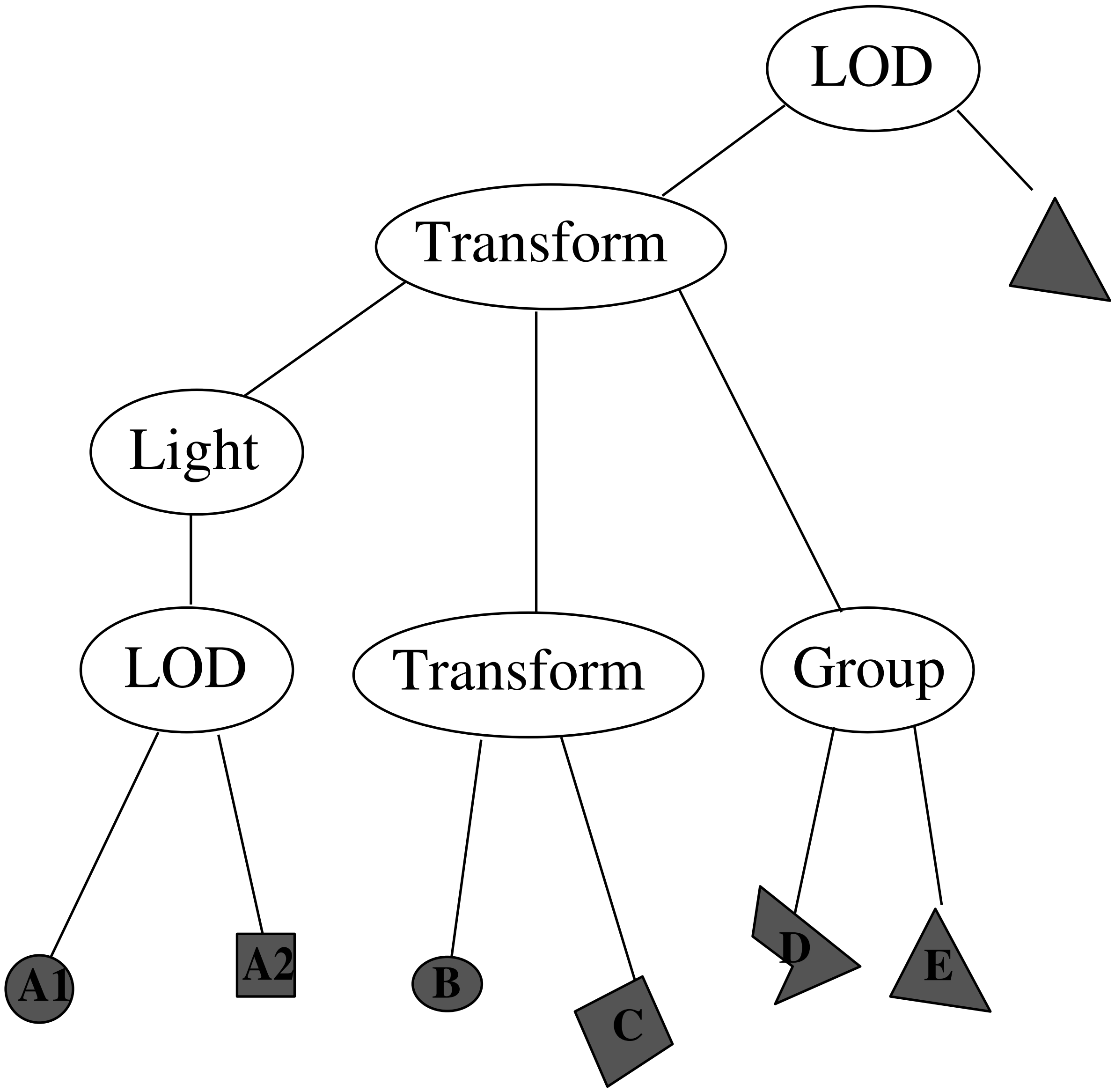
Each LOD node
has distance ranges

LOD Construction

- **Mesh Simplification**
  – Vertex decimation (Schroeder92)
  – Edge contraction (Hoppe,Garland97)
  – Vertex Clustering (Rossignac93)
  – Simplification envelopes (Cohen96)
  – Volumetic Methods (He95)
- **Methods for ‘polygon soup?’**
  – Rossignac93
- **Methods that take into account surface properties?**

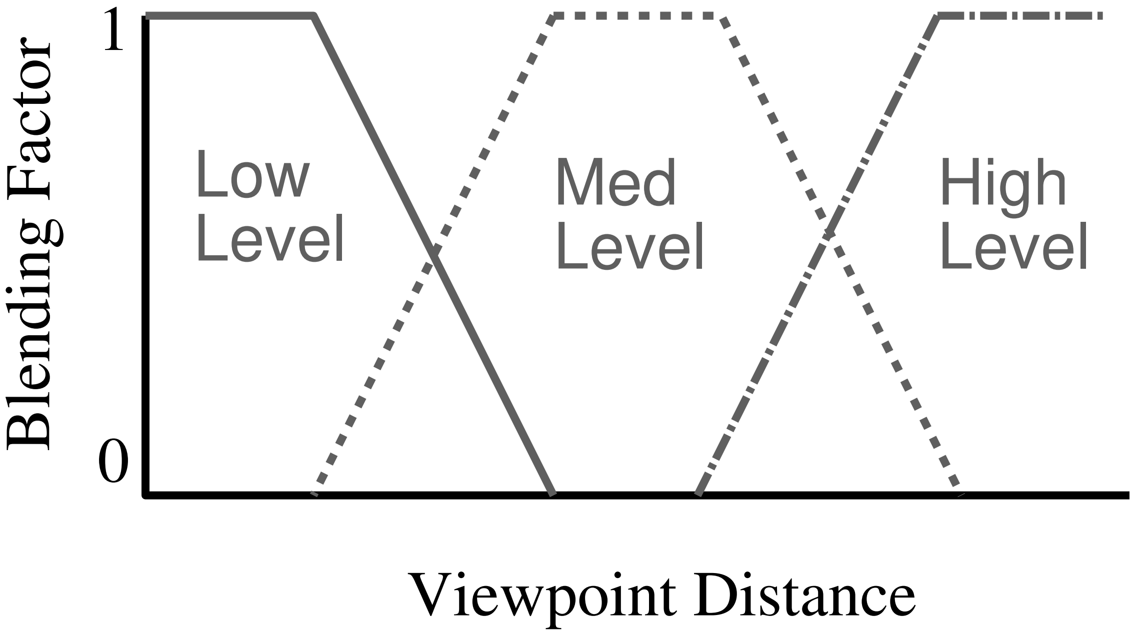
Open Question #3

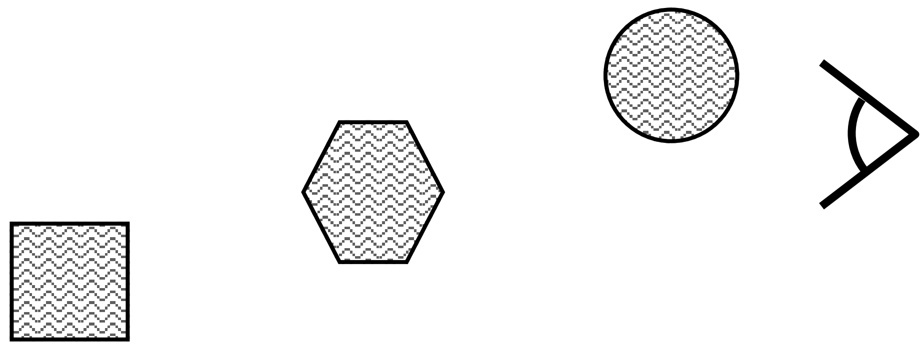
- How can we choose which nodes of the scene graph should have levels of detail?



Open Question #4

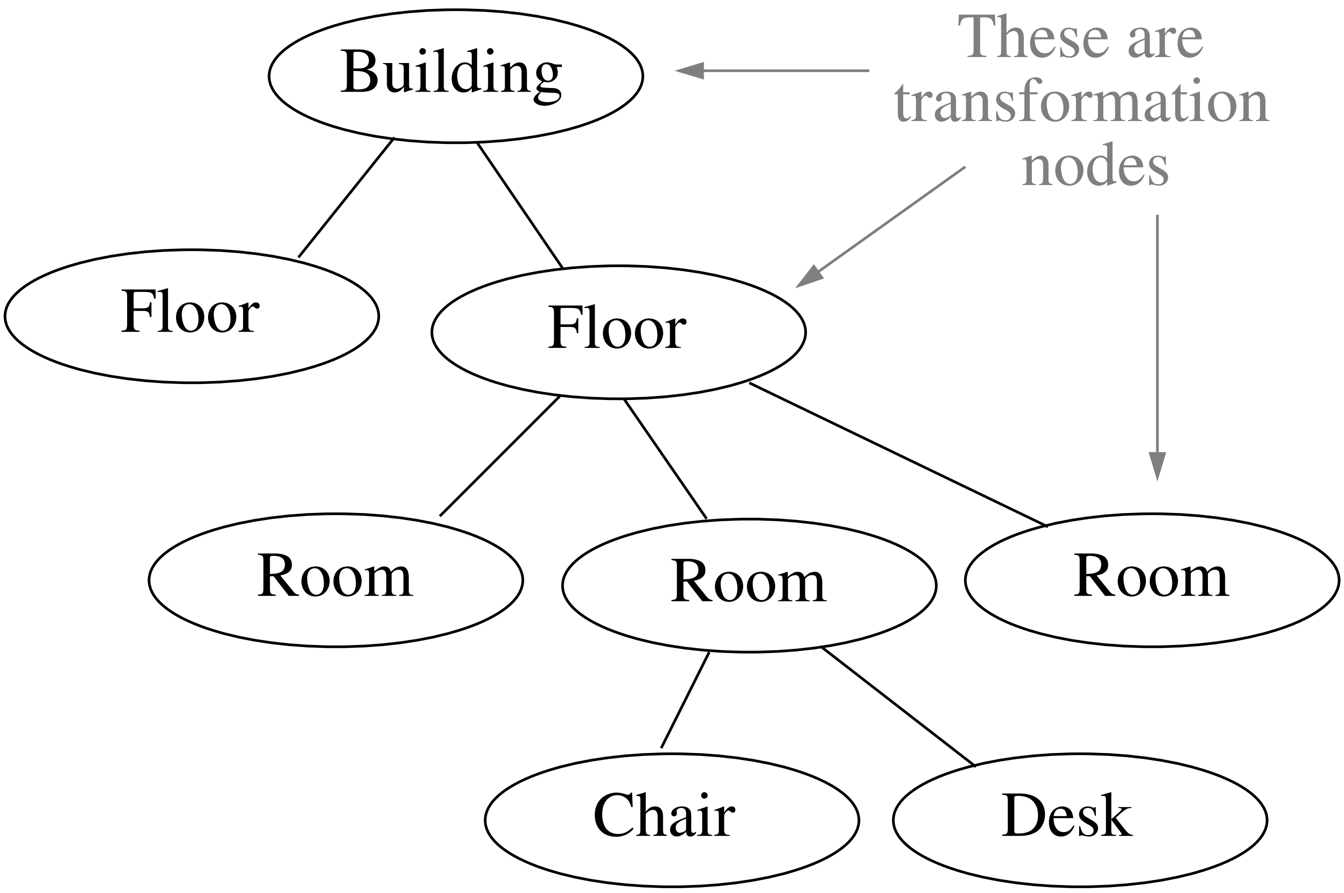
- **Can we optimize LOD ranges?**
  – Fade distances





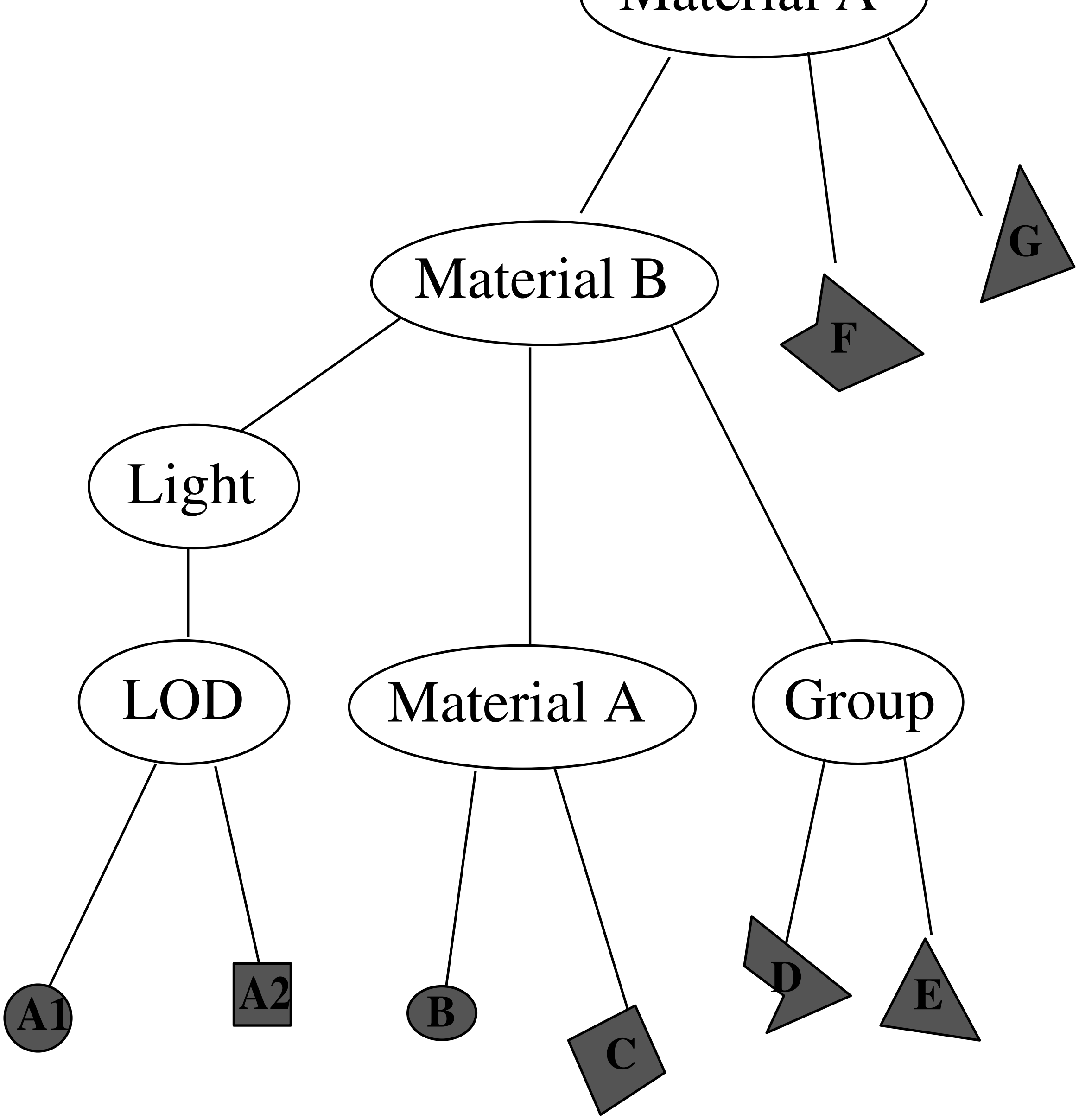
Attribute Management

- **Minimize transformations**
  - Each transformation is expensive during rendering, intersection, etc.
  - Need automatic algorithms to collapse/adjust transform hierarchy.



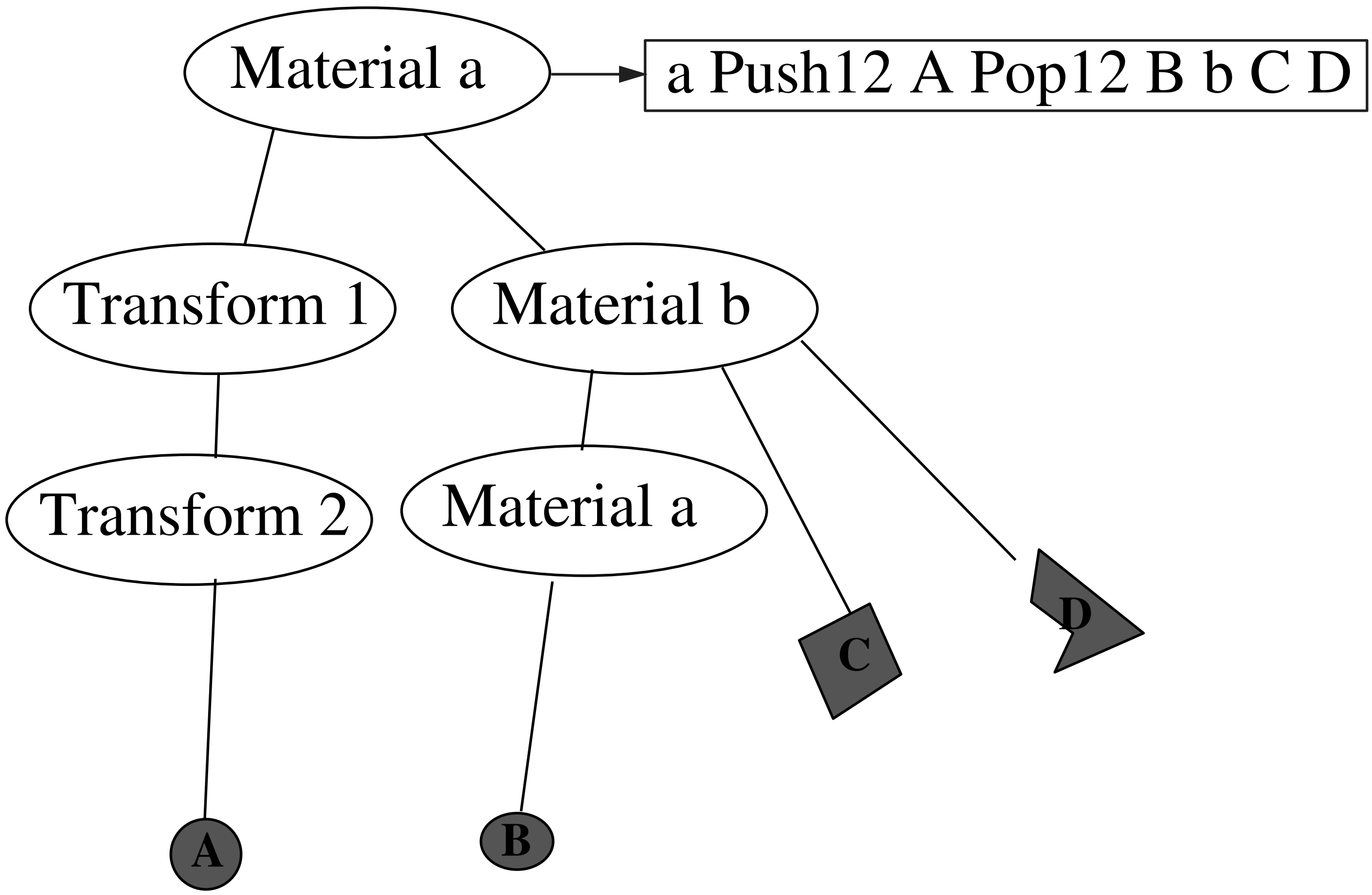
Attribute Management

- **Minimize attribute changes**
  – Each state change is expensive during rendering



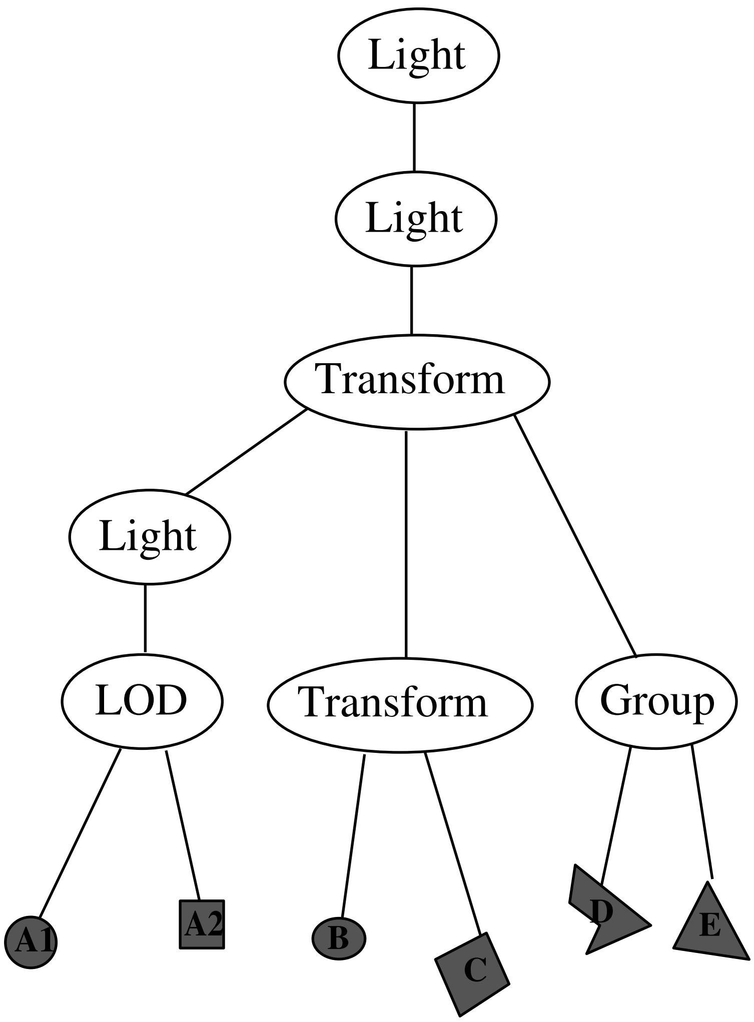
Automatic Attr Mgmt

- **Amortized Optimization** **(Gossweiler95)**
  - Convert each node into a render list
  - Peephole optimization
  - Combine transformations



Open Question #6

- **Optimize light scoping?**
  - OpenGL supports only 8 lights
  - What if there are 200 lights?
  - The modeler must ‘scope’ the lights in the scene graph?



Summary

- **Scene Graph Overview**
  - Primitives
  - Operations
  - Considerations
- **Scene Graph Optimization**
  - Function
  - Bounding volumes
  - Levels of Detail
  - Attributes
  - Lights
- **Thought Exercise**

Thought Exercise

- **Due February 17**
  - One or two paragraphs

- **Discuss methods to handle optimization of scene graphs with multiple competing goals.**

  Can we optimize all these factors in a single scene graph hierarchy?
  - Function
  - Bounding volumes
  - Levels of Detail
  - Attributes