

Introduction to Computer Graphics

Farhana Bandukwala, PhD

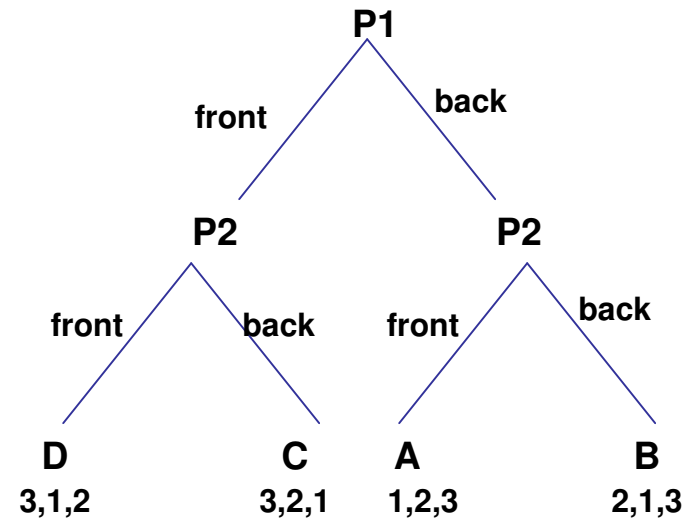
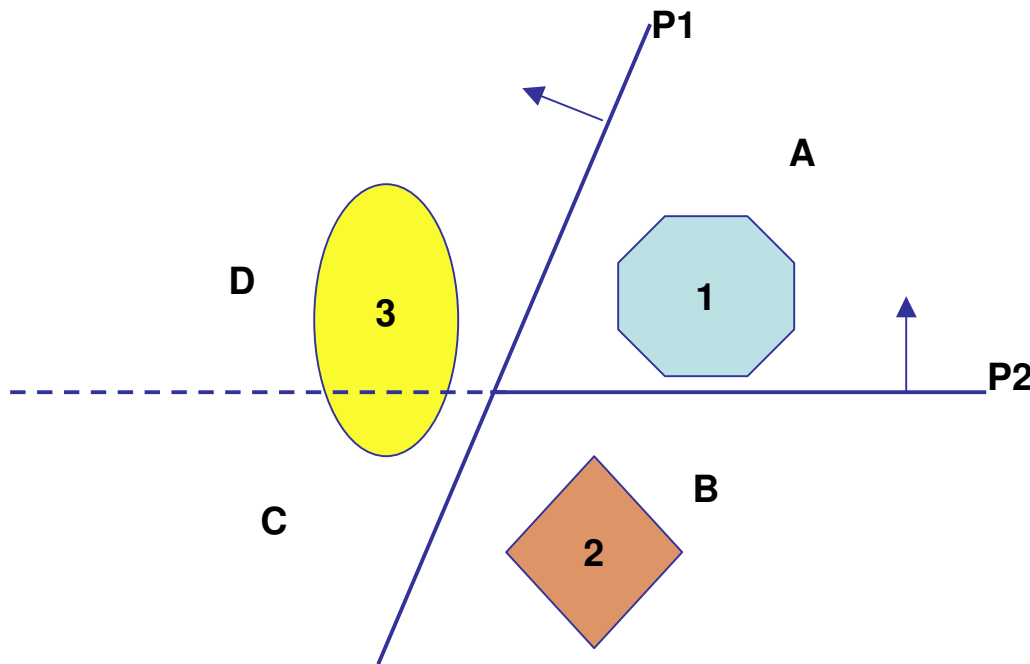
Lecture 17: Hidden surface
removal – Object Space
Algorithms

Outline

- BSP Trees
- Octrees

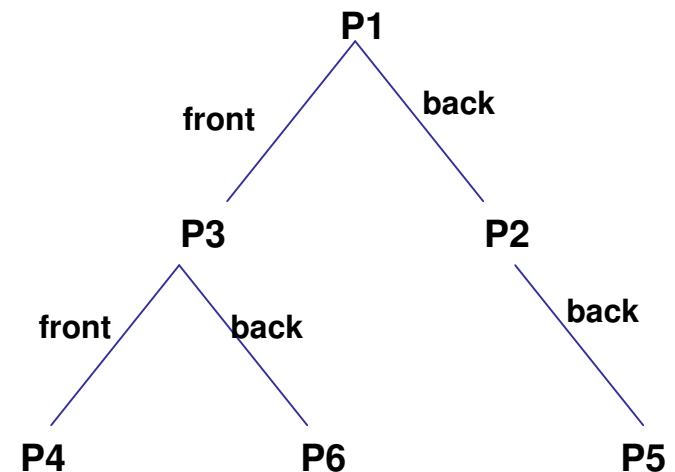
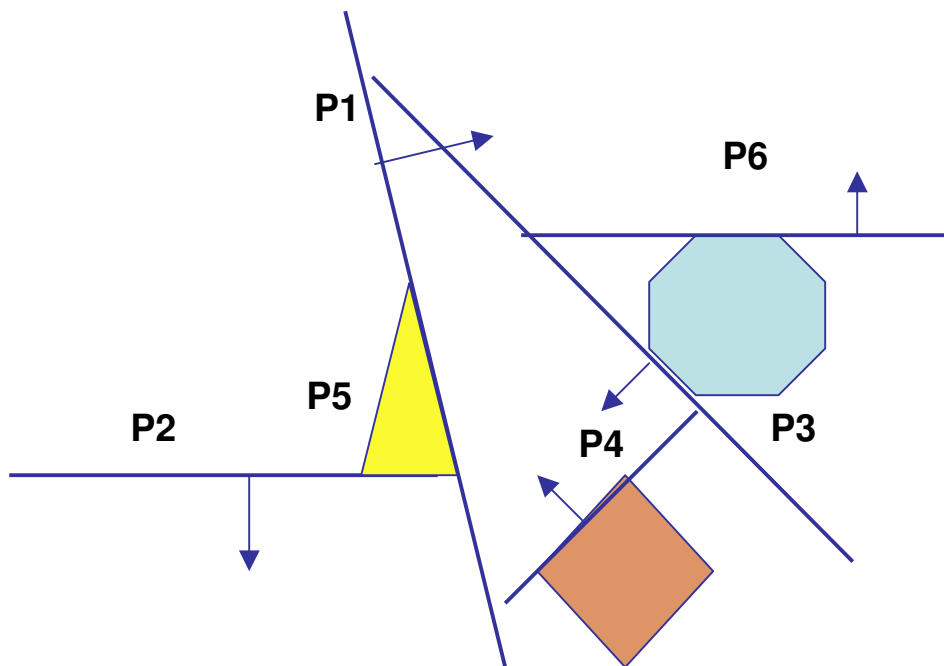
BSP Trees

- Binary space partitioning trees
- Steps:
 1. Group scene objects into clusters (clusters of polygons)
 2. Find plane, if possible that wholly separates clusters
 3. Clusters on same side of plane as viewpoint can obscure clusters on other side
 4. Each cluster can be recursively subdivided



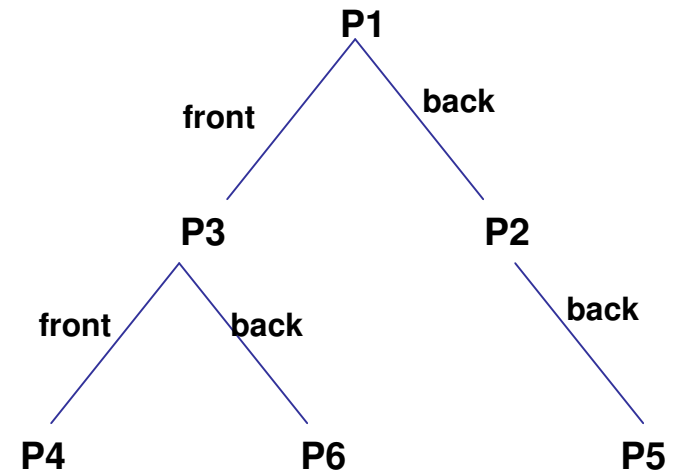
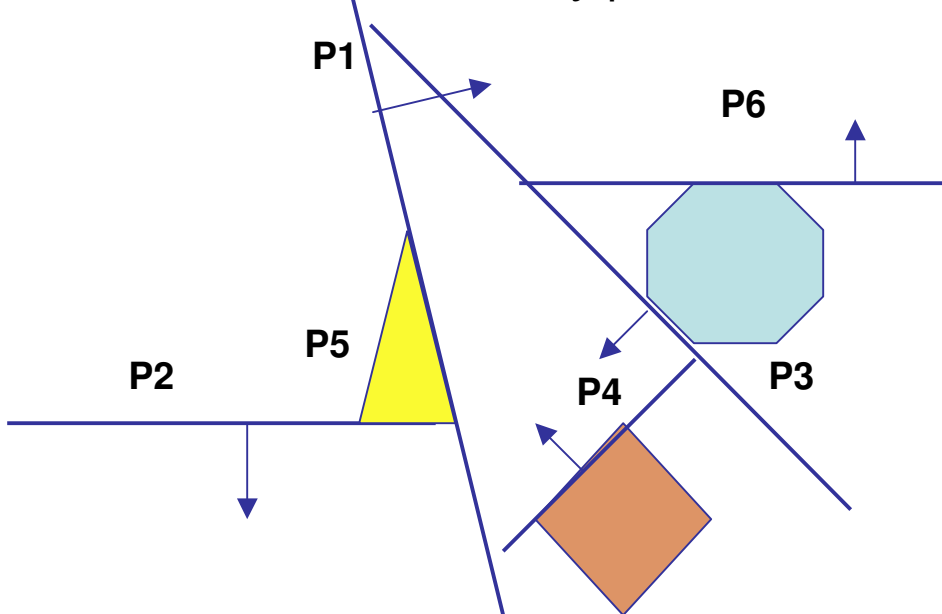
BSP Trees, contd

- Suppose the tree was built with polygons:
 1. Pick 1 polygon (arbitrary) as root
 2. Its plane is used to partition space into front/back
 3. All intersecting polygons are split into two
 4. Recursively pick polygons from sub-spaces to partition space further
 5. Algorithm terminates when each node has 1 polygon only



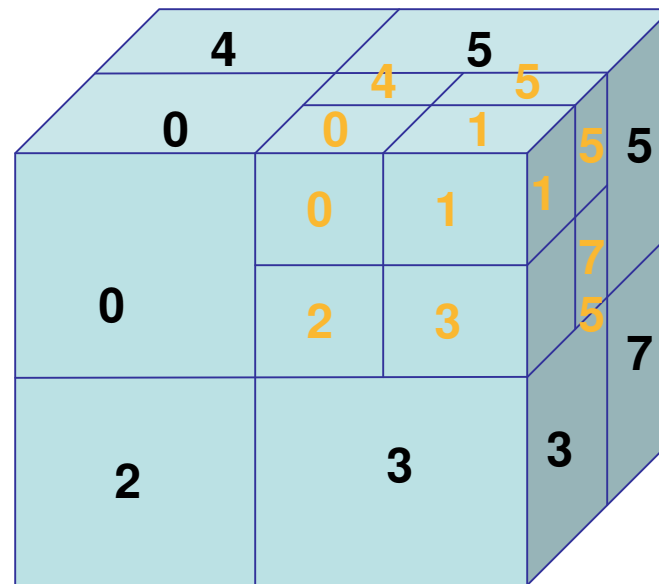
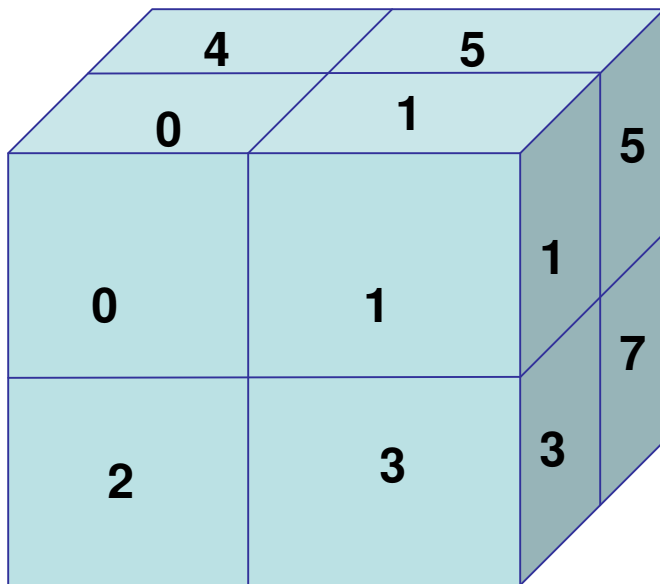
BSP Trees, contd

- Traverse tree in order:
 1. Start from root node
 2. If view point in front:
 - a) Draw polygons in back branch
 - b) Then root polygon
 - c) Then polygons in front branch
 3. If view point in back:
 - a) Draw polygons in front branch
 - b) Then root polygon
 - c) Then polygons in back branch
 4. Recursively perform front/back test and traverse child nodes



Octrees

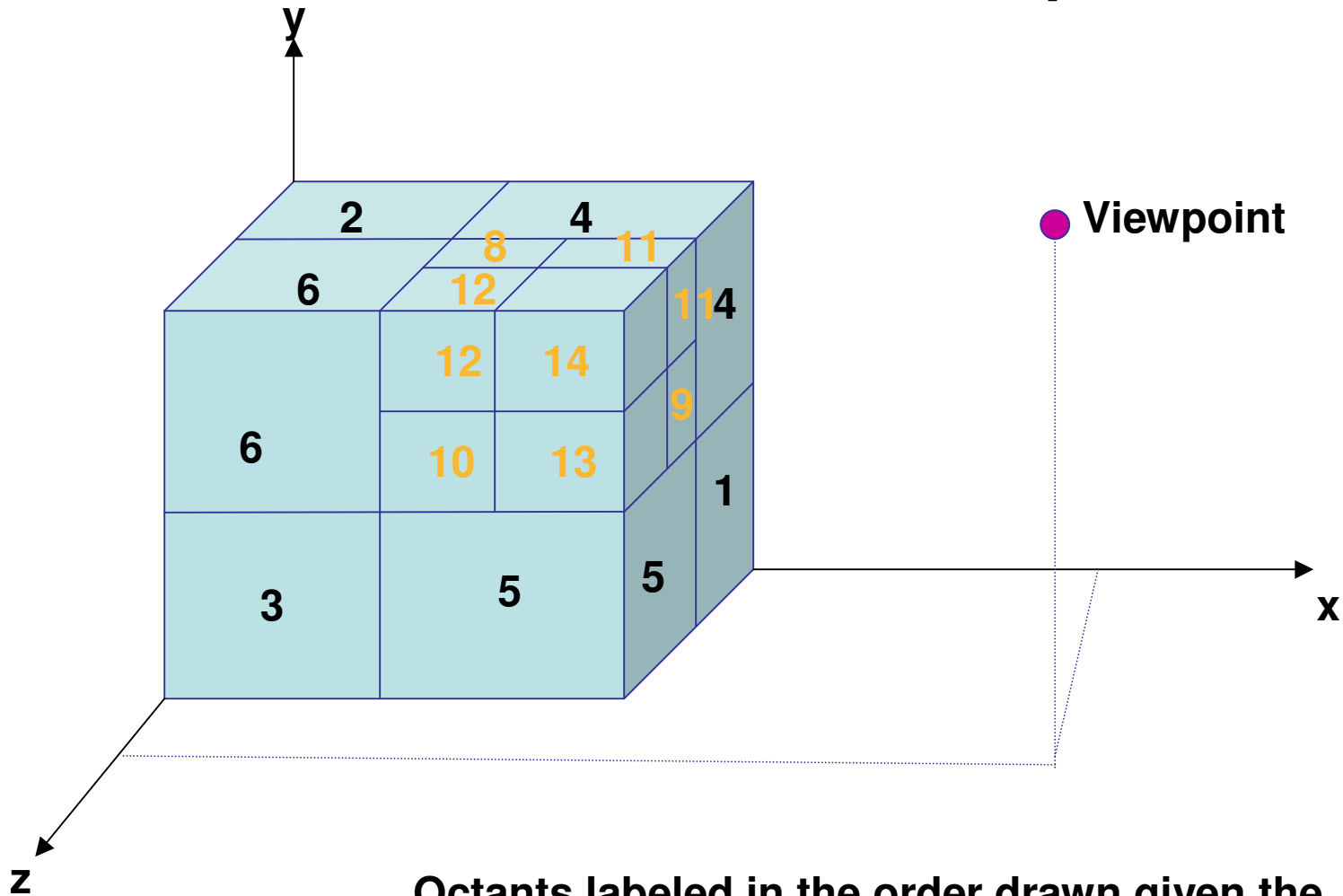
- Consider splitting volume into 8 octants
- Build tree containing polygons in each octant
- If an octant has more than minimum number of polygons, split again



Octrees, contd

- Best suited for orthographic projection
- Display polygons in octant farthest from viewer first
- Then those nodes that share a border with this octant
- Recursively display neighboring octant
- Last display closest octant

Octree, example



Octants labeled in the order drawn given the viewpoint shown here