Visibility Computations: II

- **Visible Surface Determination**
- **Visibility Culling**

Visible Surface Determination

- Area-Subdivision Algorithms
- z-buffer Algorithm
- List Priority Algorithms
- BSP (Binary Space Partitioning Tree)
- Scan-line Algorithms

Area Partitioning Algorithms

- Divide-and-conquer strategy: spatial partitioning in the projection plane
- An area of the projection plane is examined
- If all polygons visible in that area can be easily decided, they are displayed
- Otherwise, it is recursively decided
- Exploits area coherence

Warnock Algorithm: Recursive Subdivision

At each stage, the projection of each polygon has one of four relationships to the area of interest

- **Surrounding polygons** completely contain the area of interest
- **Intersecting polygons** intersect the area of interest
- **Contained polygons** are completely inside the area
- **Disjoint polygons** are completely outside the area

Warnock Algorithm: Easy Rejection Decision

- All the polygons are disjoint from the area
- There is only one intersecting or contained polygon
- There is a single surrounding polygon
- More than one polygon is intersecting, contained in, or surrounding the are, but one is a surrounding polygon that in front of all the other polygons.

Dinesh Manocha,

Warnock Algorithm: Terminating Criteria

- Stop subdividing until the display resolution is reached
- If none of the easy rejection decision can be made at a pixel, compute the depth of all relevant polygons at the center of pixel-size area
- The polygon with the closest Z coordinate defines the shading of the area
- For anti-aliasing, several further levels of subdivision may be used

Dinesh Manocha,

Warnock Algorithm: Complexity & Implementation

- Complexity of Warnock algorithm
- Implementation issues:
  - Exact intersection of a polygon with a window
  - Computing whether a surrounding polygon is in front of all the polygon
  - Robustness and accuracy issues

Dinesh Manocha,

Weiler-Atherton Algorithm: Overview

- Eliminates the image-precision operations to terminate
- Subdivides the screen area against polygon boundaries as opposed to rectangle boundaries
- Uses a concave polygon clipping algorithm

Dinesh Manocha

Weiler-Atherton Algorithm

- Compute the closest polygon (P) along the Z axis
- Clip all the polygons w.r.t. P
- Generate two lists containing the pieces inside and outside P
- All polygons on the inside list that are behind P are discarded
- Recursively apply the algorithms to polygon on the inside list that are in front of P
- Display the inside list and apply the algorithm to process the polygons on the outside list

Dinesh Manocha,

z-buffer Algorithm

- Store a z-buffer along with the frame buffer. A z-value is stored for each pixel
- Each z-value is initialized to back of the clipping plane. The frame buffer is initialized to the background color
- Polygons are scan-converted in arbitrary order
- If the polygon being scan-converted at a pixel is no farther from the viewer than is the point whose color and depth are currently in the buffers, then the new point's color and depth replace the old values

Dinesh Manocha

z-buffer Algorithm: Advantages

- Simplicity
- Polygons can be rendered in any order
- The primitives need not be polygons (need to compute depth and shading values at each pixel)
- The time taken by the visible-surface calculation tends to be independent of the number of polygons on average or is the complexity linear???
- Can be extended to handle CSG objects (later)
- The z-buffer can be saved with the image and used later to merge in other objects whose z can be computed

Dinesh Manocha,

z-buffer Algorithm: Disadvantages

- High memory requirements
- The required z-precision is maintained at a higher resolution as compared to x,y precision
- Aliasing problems (due to finite-precision)
- Rendering almost co-linear edges or co-planar polygons (visible artifacts)
- Issues in implementing transparency and translucency effects (due to arbitrary order of rendering)

Dinesh Manocha,

List Priority Algorithms

- Determine a visibility ordering for objects ensuring that a correct image results if the objects are rendered in that order
- Use a combination of object precision and image precision operations
- Object precision: depth comparisons and object splitting
- Image precision: scan conversion
- The list of sorted objects is created with object precision
- Two examples: Depth sort algorithm and BSP's

Dinesh Manocha,

Depth Sort Algorithm

To paint the polygons into the frame buffer in order of decreasing distance from the viewpoint

- Sort the polygons according to the farthest Z coordinate
- Resolve any ambiguities when the polygons' z extents overlap and split them if necessary
- Scan convert each polygon in ascending order of farthest z coordinate (back-to-front)

A simplified version where an explicit priority is associated with each polygon: **painter's algorithm**

Dinesh Manocha,

Depth Sort Algorithm: Ambiguities

Let the polygon at the far end of the list be P. It is tested again each polygon Q whose z extent overlaps with that of P. Perform 5 tests:

- Do the polygons' x extents not overlap
- Do the polygons' y extend not overlap
- Is P entirely on the opposite side of Q's plane from the viewpoint
- Is Q entirely on the same side of P's plane as the viewpoint
- Do the projection onto (x,y) plane not overlap

Dinesh Manocha

Depth Sort Algorithm: Ambiguities

- If all the five tests fail, repeat them by reversing P and Q. If the tests fail again, either P or Q must be split by the plane of the other. The algorithm is recursively applied to the split pieces
- All these tests are conservative
- What is its complexity?

Dinesh Manocha,

Binary Space Partitioning Algorithms

- For static scenes with moving viewpoints
- Uses general priority characteristics to pre-compute, off-line, the priority lists
- At run time computes a back-to-front priority of all the polygons in the scene

Dinesh Manocha

Binary Space Partitioning: Schumacker Algorithm

- Allows only convex polygons in the scene
- Grouped into clusters that are linearly separable
- For any viewpoint, the cluster priority is pre-computed
- The simplest cluster is a polygon. They can be complex polygonal or non-polygonal surface or volumes
- Within certain types of clusters, the priority of individual polygons is independent of the viewpoint

Dinesh Manocha

BSP Trees: Fuchs et al.

For a given viewpoint a polygon is correctly rendered if all the polygons on its side away from the viewpoint are rendered first; then the polygon is rendered; and finally all the polygons on the side nearer the viewpoint are rendered

Dinesh Manocha

Constructing the BSP Trees

- It recursively subdivides the space into two half spaces
- Uses one of the polygons in the scene as the separating or dividing plane
- Other polygons in the scene that are entirely on one side of the separating plane are placed in the appropriate half space
- Polygons that intersect the separating plane are split along the separating plane
- Each half space is recursively divided until there is a single polygon in each half space

Dinesh Manocha,

BSP Trees

- It is not unique. Depends on the choice of initial polygon and subsequent half spaces
- It is advantageous if separating planes create the minimum number of polygon splits
- Fuchs et al.' 83 show that testing only 5 or 6 polygons as potential separating planes give a near optimal result
- Many other applications besides visiblity culling: collision detection, boolean set operations and model representation

Dinesh Manocha

List Priority (LP) vs z-buffer

- LP algorithms do not need to check for z at each pixel
- LP display polygons in a back-to-front order
- Shading calculations are performed more than once at each pixel
- LP algorithms introduce new edges due to the subdivision process

Dinesh Manocha