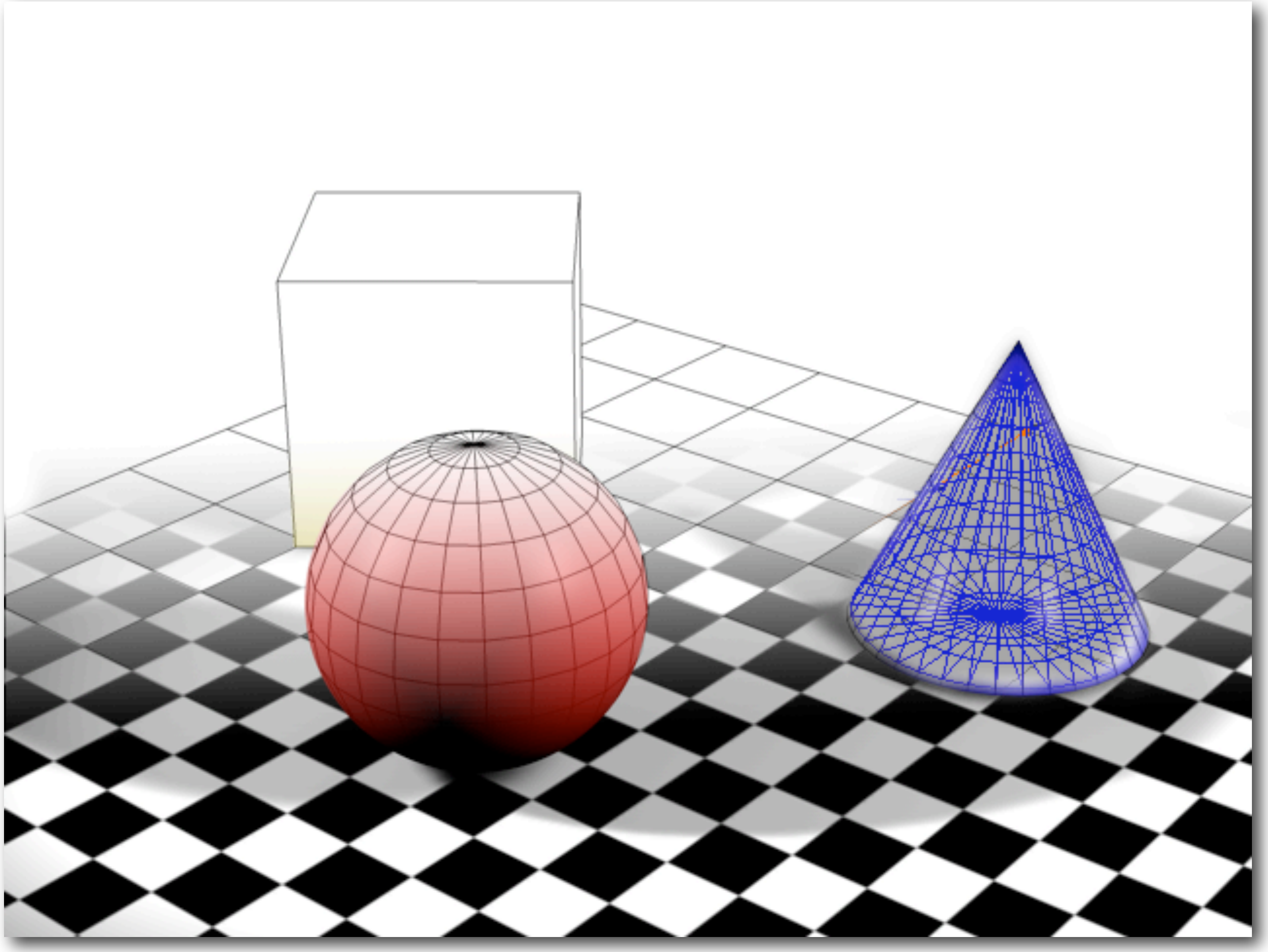


Monash University • Clayton's School of Information Technology

CSE3313 Computer Graphics

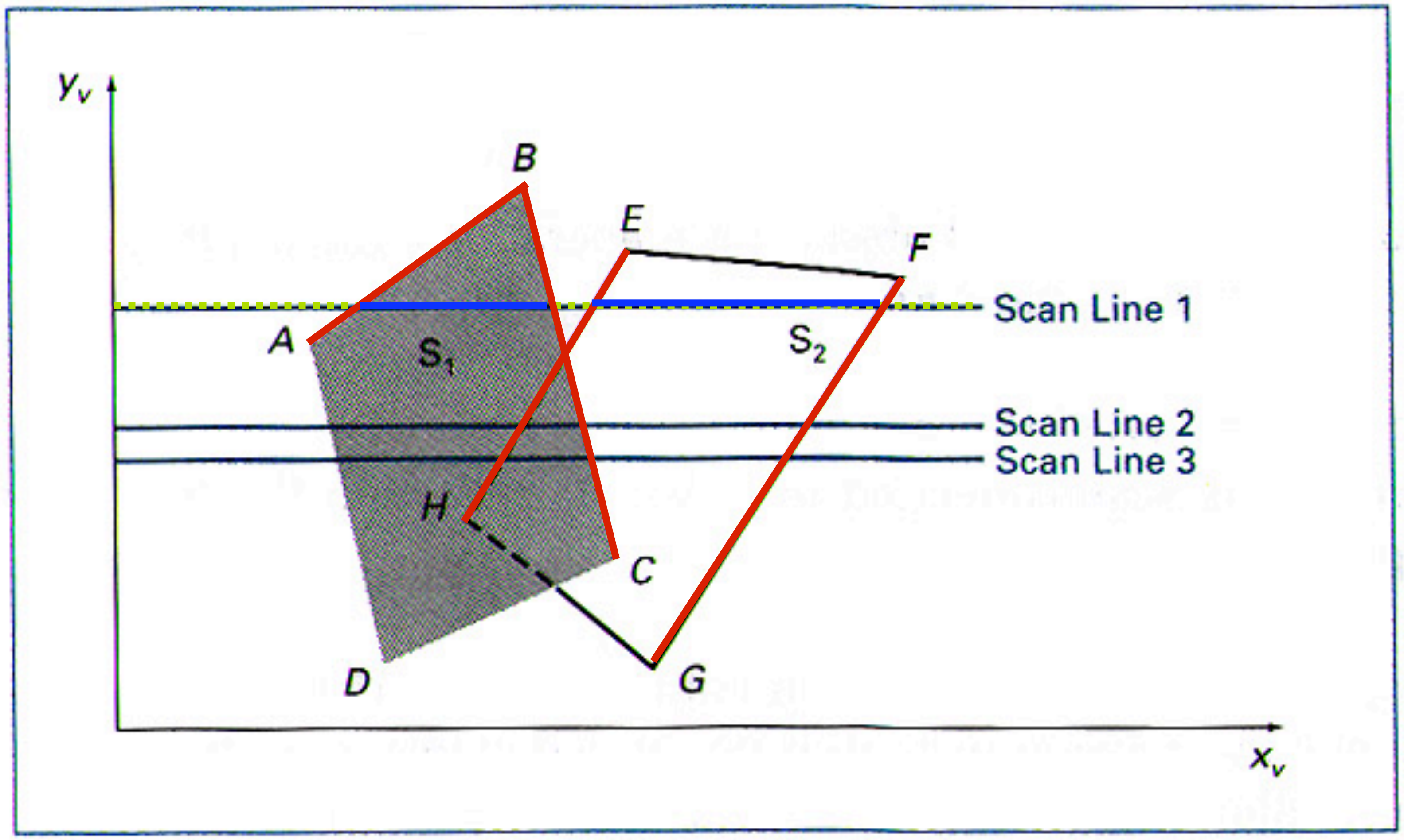
Lecture 21: Hidden Surface Removal (continued)

Scanline Algorithms

- This image space method for removing hidden surfaces is an extension of the scan-line algorithm for filling polygon interiors. It generalises the scanline algorithm by considering that multiple surfaces may potentially be visible at a given pixel.

- For each scan line we continue to maintain an active edge list of edges that cross the current scan line, sorted in increasing x. In addition, for each surface we define a flag that shows if that surface is visible depending on our position along the scan line.

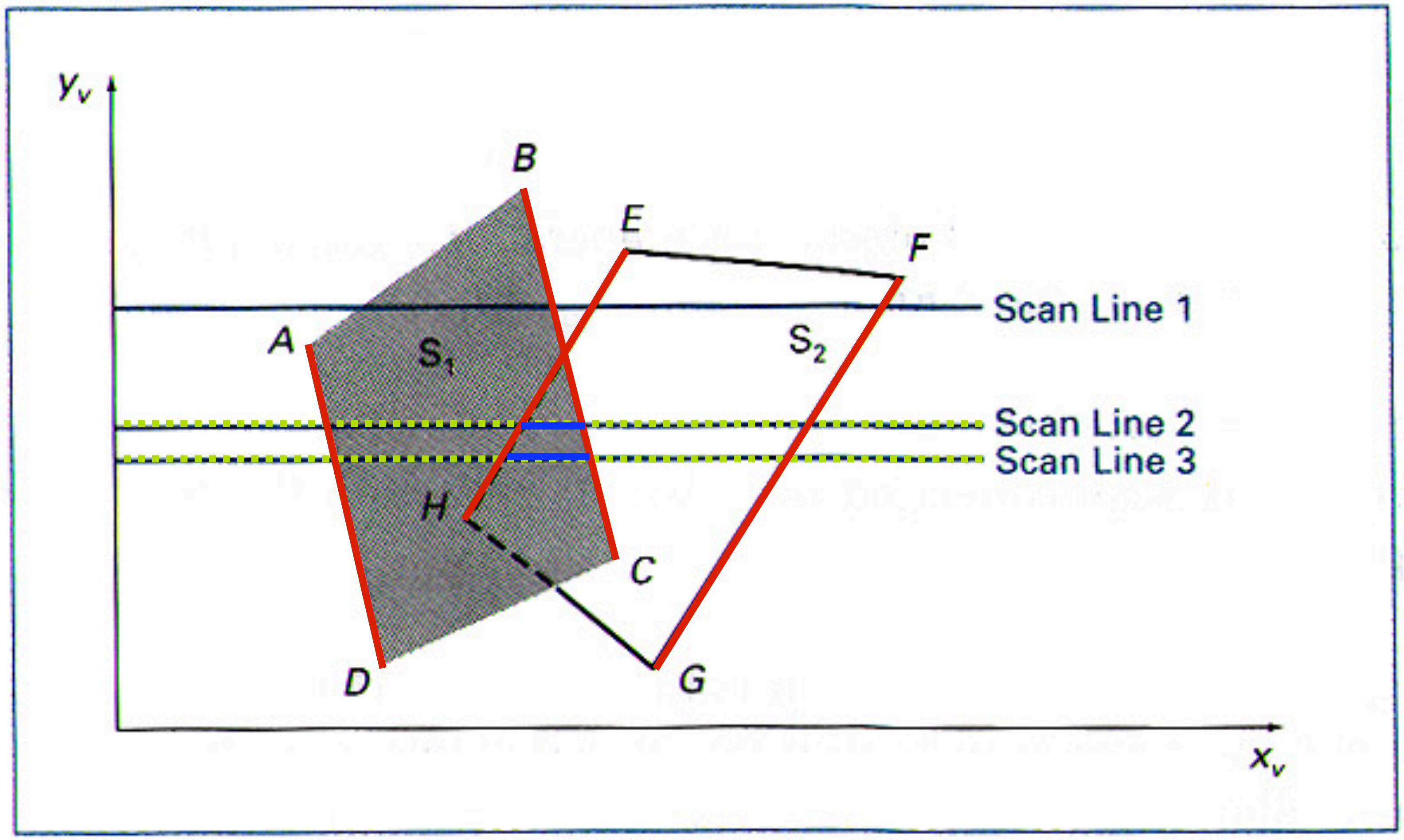
- As the scan line crosses the edge for a particular polygon we need to determine which polygon is visible.



Scan lines crossing the projection of two surfaces, S_1 and S_2, in the view plane. Dashed lines indicate the boundaries of hidden surfaces.

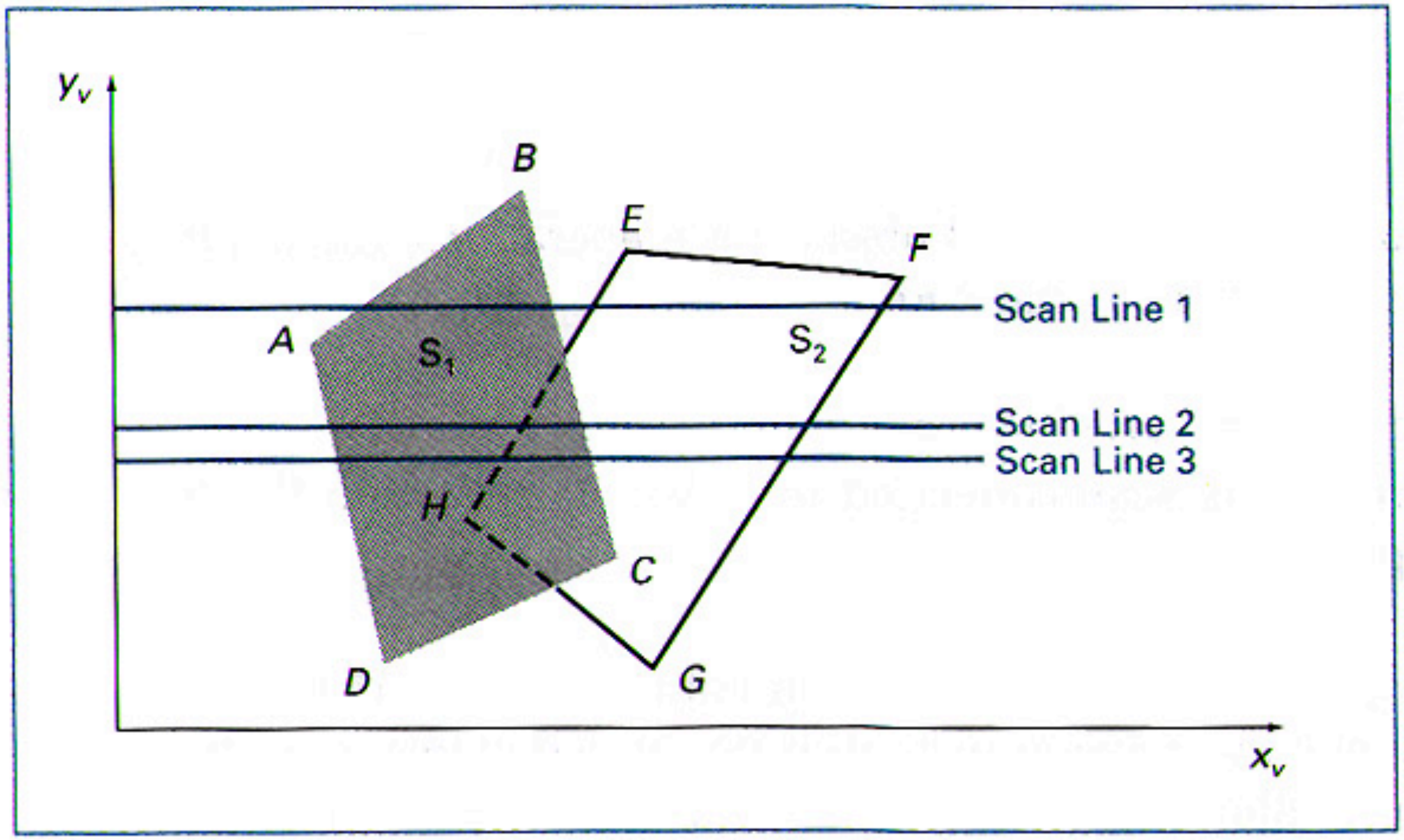
- Active edges for scan 1: AB, BC, EH, FG. Between AB and BC the flag for S1 is on and can be directly rendered. Same for EH and FG for S2.

Scanline Algorithms (cont.)



Scan lines crossing the projection of two surfaces, S_1 and S_2, in the view plane. Dashed lines indicate the boundaries of hidden surfaces.

- For scan lines 2 & 3, active edges are: AD, EH, BC, and FG. From AD to EH, S1 is on. Between EH and BC both S1 and S2 are on so depth calculations must be made using plane equations for each surface.

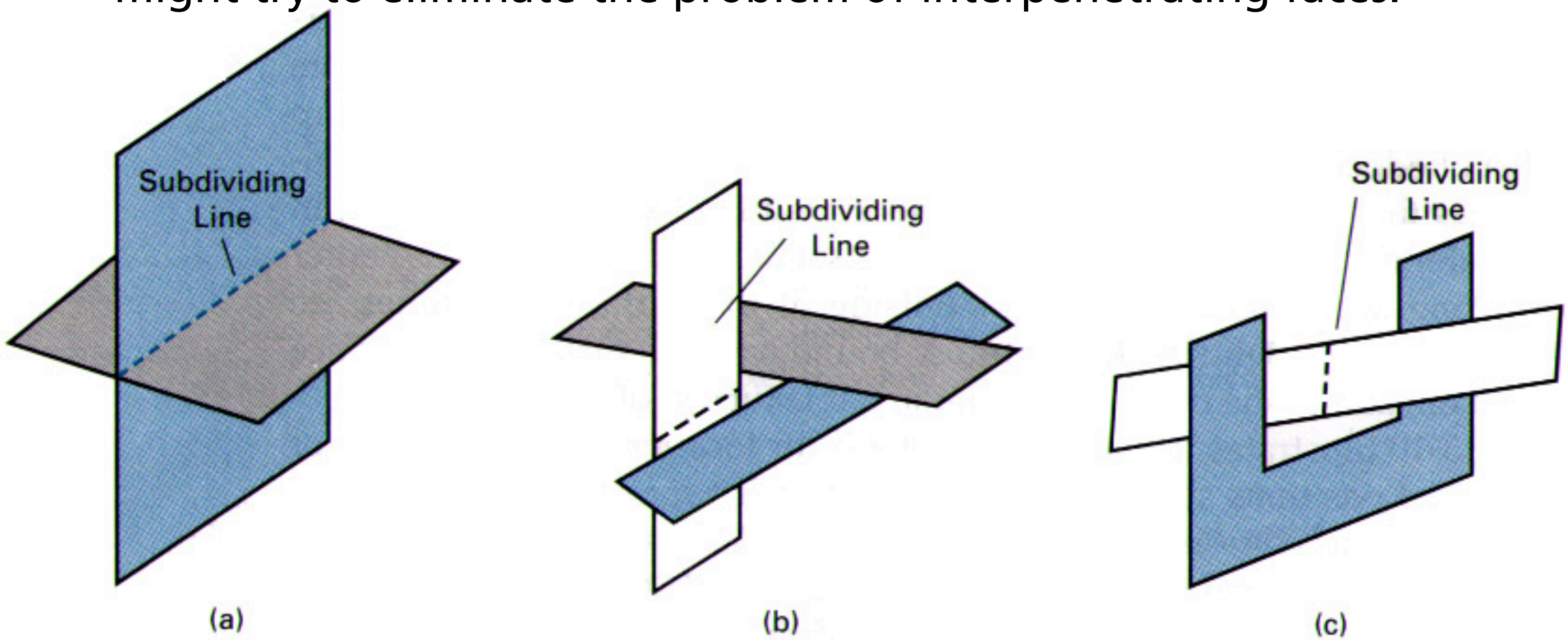


Scan lines crossing the projection of two surfaces, S_1 and S_2, in the view plane. Dashed lines indicate the boundaries of hidden surfaces.

- Scan line 3 has the same active edge list as scan 2. Since no changes have occurred in line intersections, we do not need to recompute the depth calculations between EH and BC. (Scan-line Coherence).

Scanline Algorithms (cont.)

- If we do not allow for interpenetrating faces, a polygon will continue to be visible for a particular scan-line until either
  a. The face finishes, or
  b. Another polygon first intersects with the scan line.
- By dividing intersecting polygons along the line of intersection we might try to eliminate the problem of interpenetrating faces.



Intersecting and cyclically overlapping surfaces that alternately obscure one another.

Comparison of Hidden Surface Methods

- Most hidden surface removal algorithms are related to sorting algorithms. This is most clearly seen in the z buffer method.

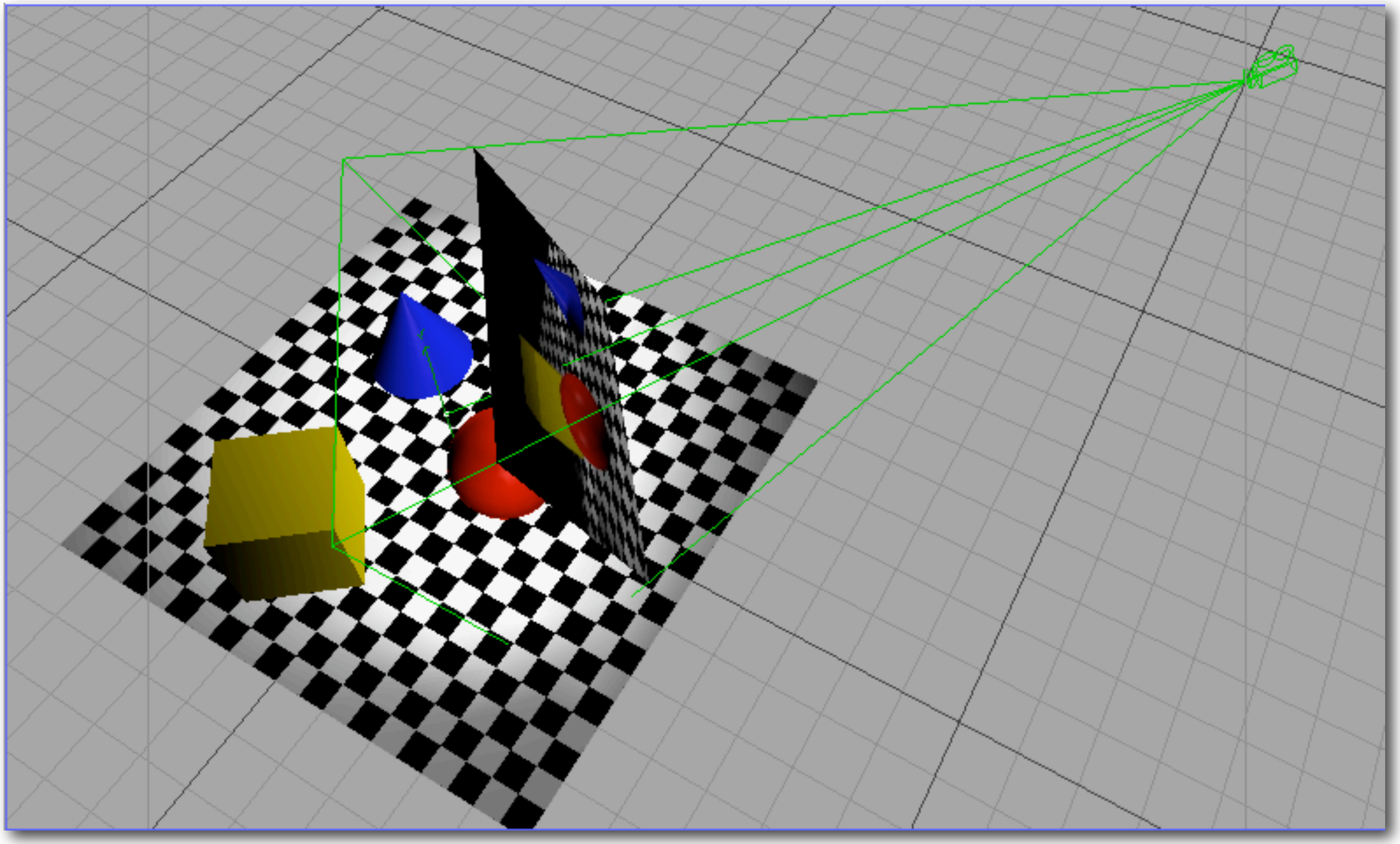
- For all points that are projected onto the window in the view plane we want to sort them in order of:

  1. maximum y to minimum y – picture drawn from top to bottom;
  2. minimum x to maximum x – picture drawn from left to right;
  3. minimum z to maximum z – ordered from nearest to furthest.

- For any point on the window (x,y), only the point closest to the point of view (minimum z value), will be displayed.

HSR Comparison (cont.)

- In sorting there is no best sorting method.
- *Quicksort* has good average case behaviour.
- *Heapsort* has good worst case behaviour.
- *Quicksort* is O(n log n) average case behaviour, while Bubblesort has $O(n^2)$. However, if the data is nearly sorted, Bubblesort might be more efficient.
- E.g. 3 7 9 8 10 13 12 17
- Such a sequence is partially ordered and might be sorted in one or two passes by a Bubblesort algorithm.
- A similar feature for hidden surface algorithms is *coherence*.
- Reference: *A Characterization of Ten Hidden-Surface Elimination Algorithms*, I.E. Sutherland, R.F. Sproul and R.A. Schumaker, *ACM Computing Surveys*, Vol 6., No. 1, March 1974, pp 1-56.



Rendering 3D Polygonal Objects

1. Polygons representing an object are extracted from the database and transformed into the world coordinate system using linear transformations such as translation and scaling.

2. A scene constructed in this way is transformed into a coordinate system based on a view point and direction.

3. The polygons are then subjected to a visibility test. This is called 'backface elimination' or 'culling' and removes those polygons that face away from the viewer. Typically half the polygons in an object are removed by this test (for convex objects), and these will not have to be dealt with by a general hidden surface removal algorithm.

4. Unculled polygons are clipped against a three dimensional view volume.

5. Clipped polygons are then projected onto a view plane or image plane.

6. Projected polygons are then shaded by an incremental shading algorithm. This algorithm runs three processes in parallel. First the polygon is rasterised, or those pixels that the edges of the polygon contain are determined (usually in scan line order). Second, a depth for each pixel is evaluated and a hidden surface calculation is performed. Third, the polygon is shaded. Both the second and the third processes use geometric information from three-dimensional world or view space, but overall the process is driven from, or controlled by, screen space. By this we mean that the algorithm processes pixels consecutively.

Rendering 3D Polygonal Objects (cont.)