

# IMAGE SPACE METHODS

*Amrita Kaur, Assistant Professor*

*Sushma Jain, Associate Professor*

*Anupam Garg, Assistant Professor*

*Santarpal Singh, Assistant Professor*



THAPAR INSTITUTE  
OF ENGINEERING & TECHNOLOGY  
(Deemed to be University)

# Outline

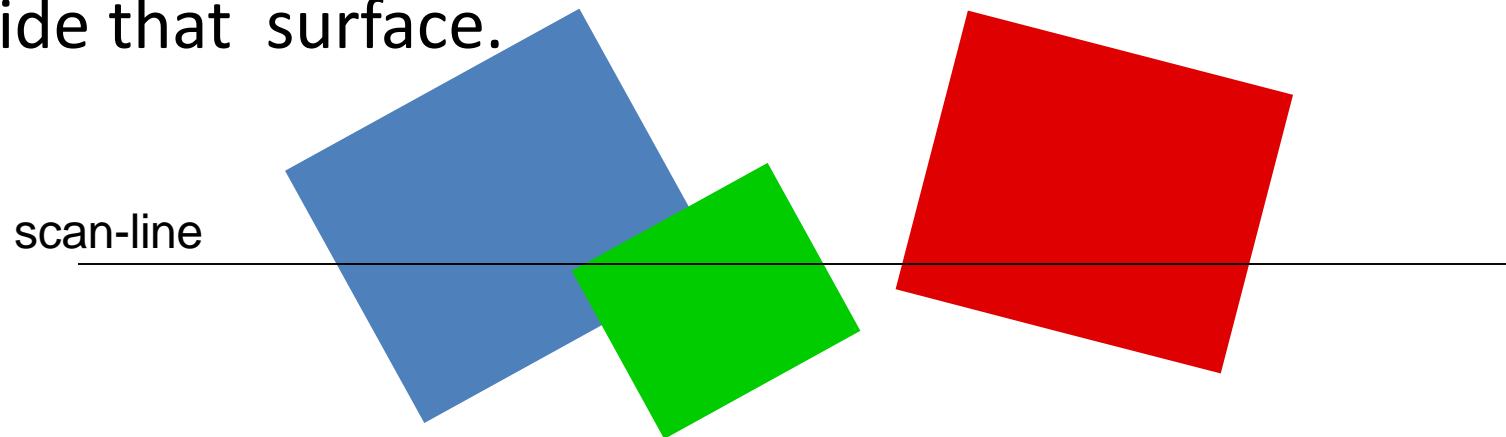
- Scan-line method
- Area-Subdivision method
- Area-Subdivision stopping conditions

# Scan-Line Method

Unlike z-buffer or A-buffer, scan-line method has depth info only for a single scan-line.

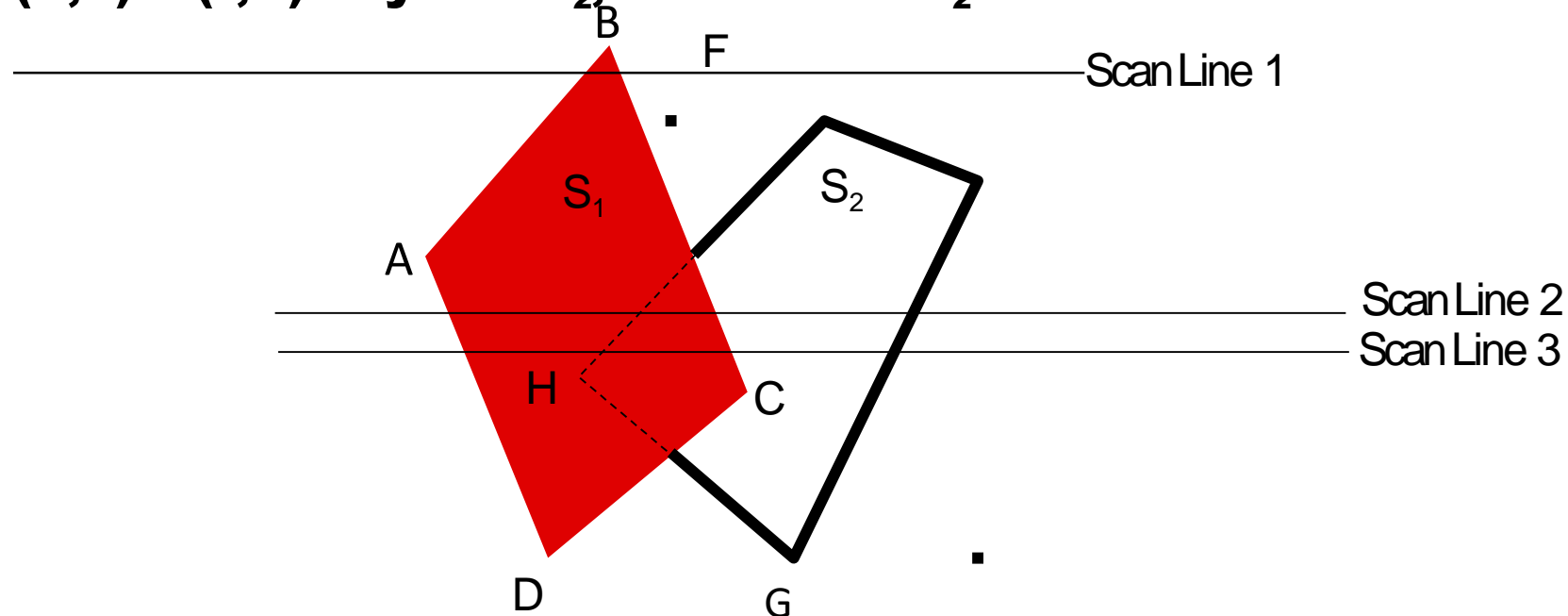
In order to require one scan-line of depth values, we must group and process all polygons intersecting a given scan-line at the same time before process the next scan-line

Build table of edges of all polygons in scene. Maintain active-edge-table as we visit each scan-line in scene. AET now contains edges for all polygons at that scanline. Must maintain flag for each surface to determine whether pixel on scan-line is inside that surface.



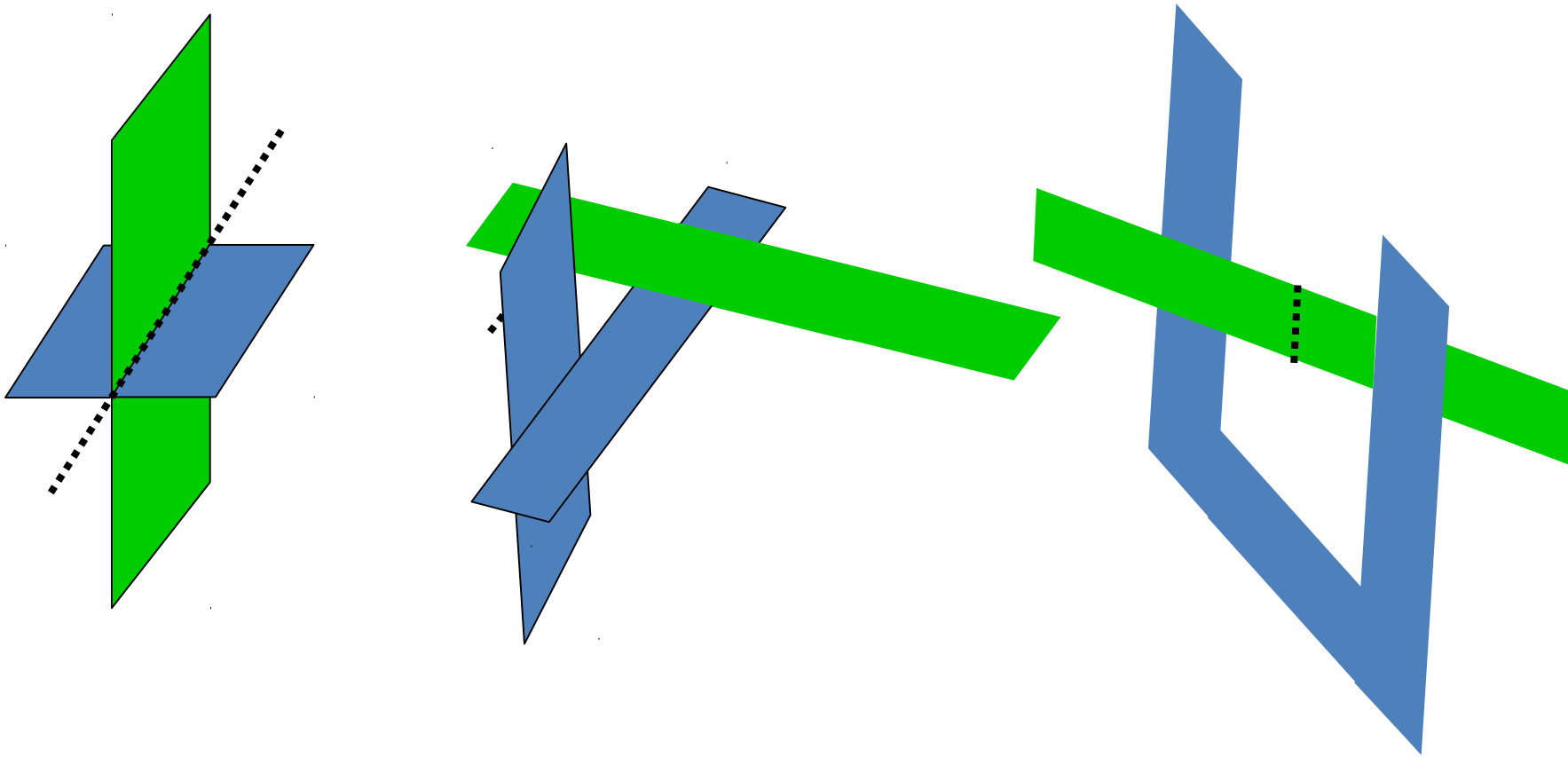
# Scan-Line Method Basic Example

- Scan Line 1:
  - (A,B) to (B,C) only inside  $S_1$ , so color from  $S_1$
  - (E,H) to (F,G) only inside  $S_2$ , so color from  $S_2$
- Scan Line 2:
  - (A,D) to (E,H) only inside  $S_1$ , so color from  $S_1$
  - (E,H) to (B,C) inside  $S_1$  and  $S_2$ , so compute & test depth In this example we color from  $S_1$
  - **(B,C) to (F,G) only inside  $S_2$ , so color from  $S_2$**



# Scan-Line Method Generalization

- This basic approach fails when surfaces cut-through each other or overlap. To generalize we must divide surfaces to eliminate overlaps



# Scan Line Method

- Extension of the scan-line algorithm for filling polygon interiors
- For all polygons intersecting each scan line
- Processed from left to right
- Depth calculations for each overlapping surface
- The intensity of the nearest position is entered into the refresh buffer

# Tables for the various surfaces

- Edge table

- Coordinate endpoints for each line

- Slope of each line

- Pointers into the polygon table

- Identify the surfaces bounded by each line

- Polygon table

- Coefficients of the plane equation for each surface

- Intensity information for the surfaces

- Pointers into the edge table

# Active List and Flag

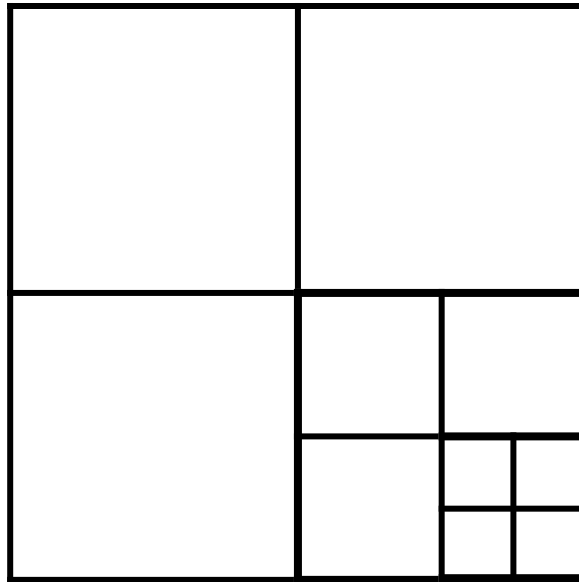
- Active list
  - Contain only edges across the current scan line
  - Sorted in order of increasing  $x$
- Flag for each surface
  - Indicate whether inside or outside of the surface
- At the leftmost boundary of a surface
  - The surface flag is turned on
- At the rightmost boundary of a surface
  - The surface flag is turned off



# Area-Subdivision Method

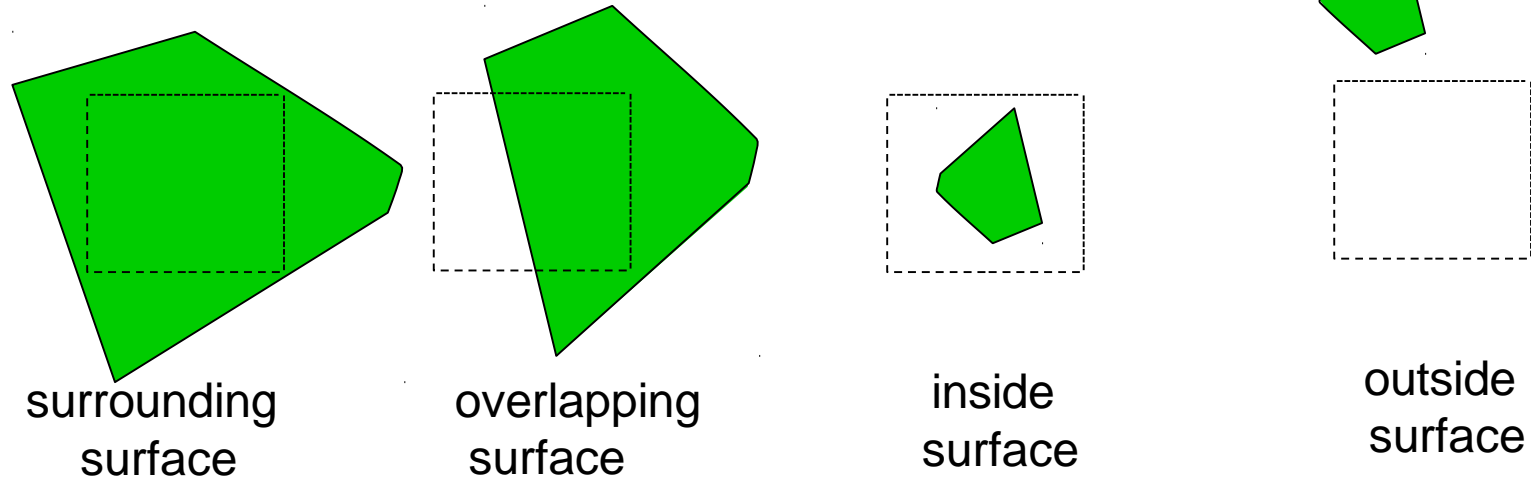
Recursively subdivide viewplane into quadrants until:

- rectangle contains part of 1 projected surface
- rectangle contains part of no surface
- rectangle is size of pixel



# Area-Subdivision Method

- We need tests that can quickly determine tell if current area is part of one surface or if further subdivision is needed.
- Four cases for relation between surface and rectangular area:



# Area-Subdivision: Stopping Conditions

- Recursive subdivision can stop when either:
  - 1) a rectangle has all surfaces outside
  - 2) a rectangle has exactly one inside, overlapping, or surrounding surface
  - 3) a rectangle has one surrounding surface and the surface occludes all other surfaces in the area
- For efficiency:
  - compare rectangle to projected surface bounding rectangle first. Only perform exact intersection test if necessary. If single bounding rect. intersects rectangle, test for exact intersection and color the frame buffer for the intersection of surface and rectangle.

THANKING YOU