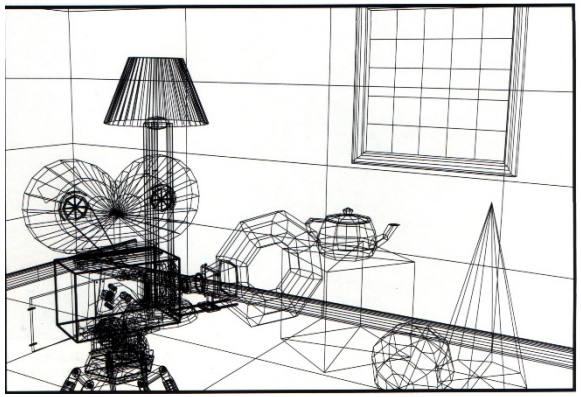
Image Synthesis
Evaluation of Visibility

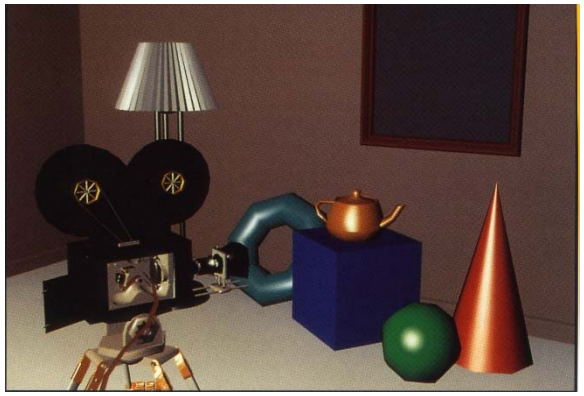
Graphics Systems /

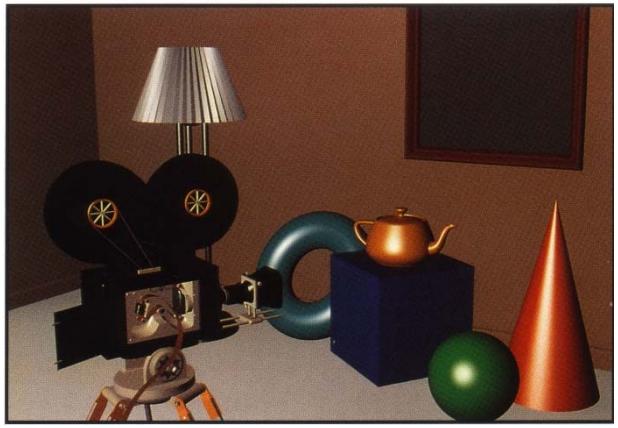
Computer Graphics and Interfaces

Image Synthesis

The **image synthesis** (English ***rendering***) Is to create images with a high degree of realism from the description of the objects it contains (geometry and interaction with light), the light sources and the position of the observer.







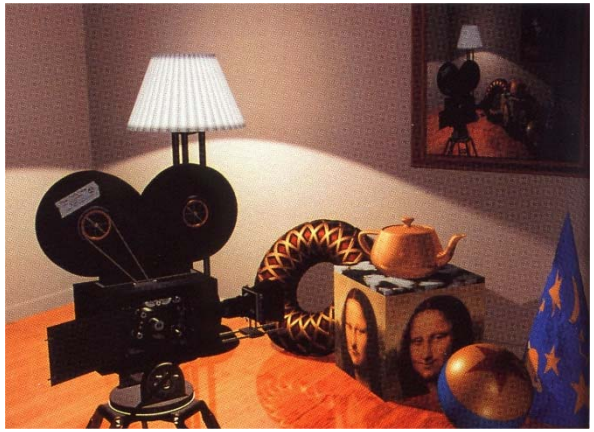


Image Synthesis

Index

Image Synthesis
- Evaluation of Visibility -

Objective: from a set of 3D objects, determine which lines or surfaces of the objects are visible, both from the center of projection (perspective projection) or along the projection direction (for parallel projection), so that only lines or visible surfaces are shown.

Two possible approaches:

1. Algorithms in the **IMAGE AREA**:

For each image pixel, determine which object is visible

```
for (Each pixel in the image)
  determine the object that is the closest to the observer,
      given the projection ray;
  draw the pixel with the appropriate color;
```

2. Algorithms in the **OBJECT AREA:**

Compare objects between themselves in order to select the visible part of each

```
  for (Each object in the "world")
      determine the parts of the object not obstructed by him or
            other objects;
      draw the visible parts in the appropriate color;
```

Image Synthesis
- Evaluation of Visibility -

Backface culling

Image Synthesis - Evaluation of Visibility
Algorithms in the object space

Algorithms in object space to determine visible lines:

By Volume: Algorithm of Roberts

By Edge: Algorithm ofAppel, Loutrel, Galimberti and Montanari

In these algorithms all edges are tested to produce a list of visible segments of all edges.

By Volume: It is assumed that an edge may be hidden by the volume of an object.

By Edge: The test is performed edge against edge noting that the visibility of an edge uses coherence, which allows to determine the invisibility of an edge from the invisibility of other edge that has a common vertex.

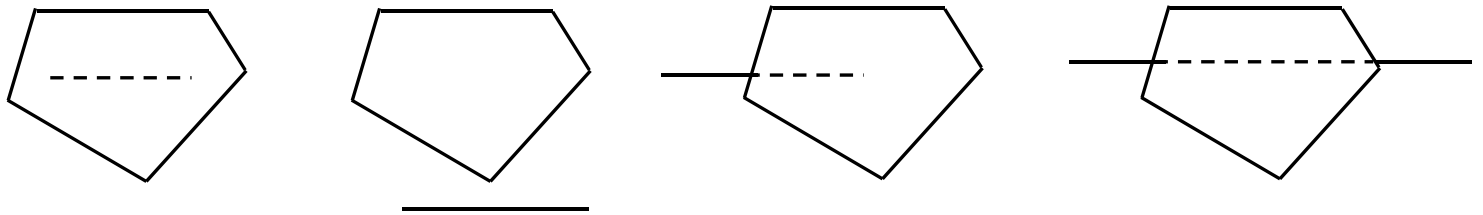
Edge Coherence: one edge changes its visibility only where it crosses (behind) a visible edge.

Image Synthesis
- Evaluation of Visibility – Algorithms in the object space

By Volume - Algorithm Roberts (1963)

Requirement: Each edge must belong to one face of a convex **polyhedron**. Concave polyhedra must be broken into several convex in order to apply the algorithm.

1. Remove all back faces of objects *(Backface culling)* and corresponding edges
2. Compare the remaining edges against each **volume** (Polyhedron) of the scene; 4 of this test situations may occur:



- Edge completely hidden by volume.
- Edge does not hide
- A part of the edge is not hidden
- Two parts of the edge are not hidden

Image Synthesis
- Evaluation of Visibility - Algorithms in the object space

By Edge - Algorithm Appel, Loutrel, Galimberti and Montanari (1967/9, 1970)

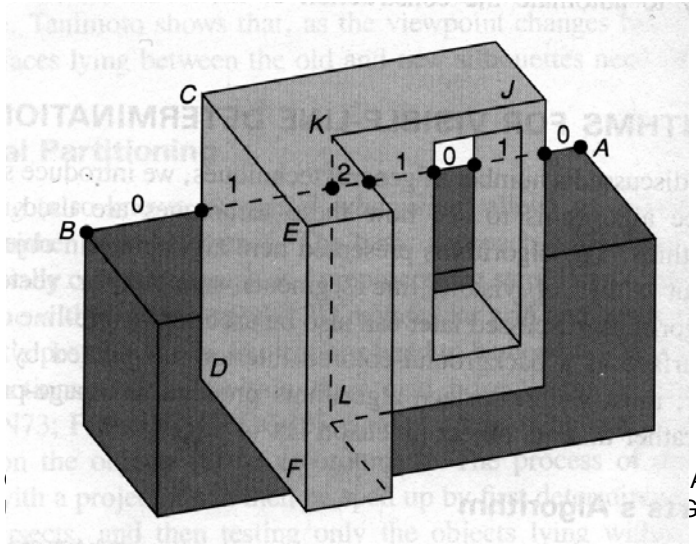
Unlike alg. Roberts, it works at the polygons level.

1. Determine the faces oriented to the observer *(Backface culling).*
2. Calculate "***Quantitative Invisibility***" of a vertex for each object.

   "*Quantitative Invisibility*" **IQ** of one point: the number of polygons between the observer and the point.

   When passing behind an edge of a polygon, their **IQ** is incremented by 1, and when it is no longer occluded, is decremented by 1.

- When it reaches the end vertex of an edge, the value of **IQ** is copied to the initial value of **QI** of the edges starting in that vertex.



<u>*Contour line*</u>: Is defined as an edge shared by a polygon *back-facing* with another front-facing, or polygon *front-facing* isolated.

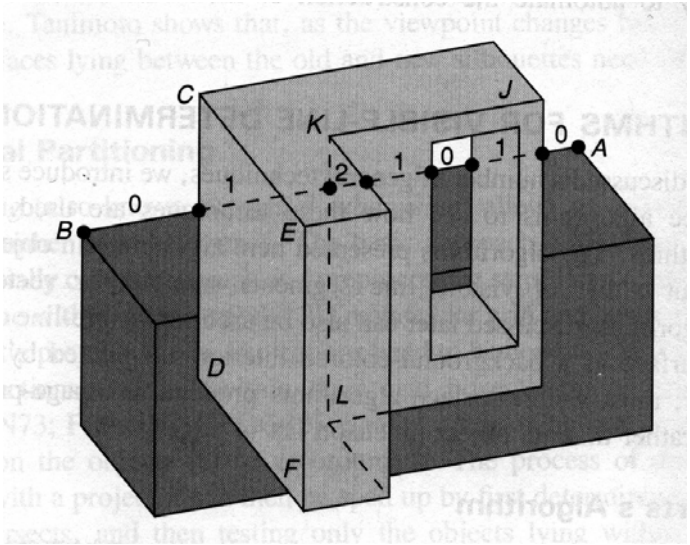
<u>*Contour lines*</u>: AB, CD, DF, KL...

EC, EF, JK are not (because they are shared by polygons *front-facing*)

Image Synthesis
- Evaluation of Visibility - Algorithms in the object space

The **IQ** changes only when the edge passes behind a *contour line*.

In the figure the points indicate the intersections of the projection of the edge AB with the projections of *contour lines*.



In the end, only the segments with value **IQ** equal to **zero** are visible.

Image Synthesis
- Evaluation of Visibility - Algorithms in the object space

Determining visible faces: **Atherton & Weiller (1977)**

Requirement: Requires the application of a sophisticated algorithm for *clipping* polygons, capable of perform *clipping* of concave polygons with holes against any other.

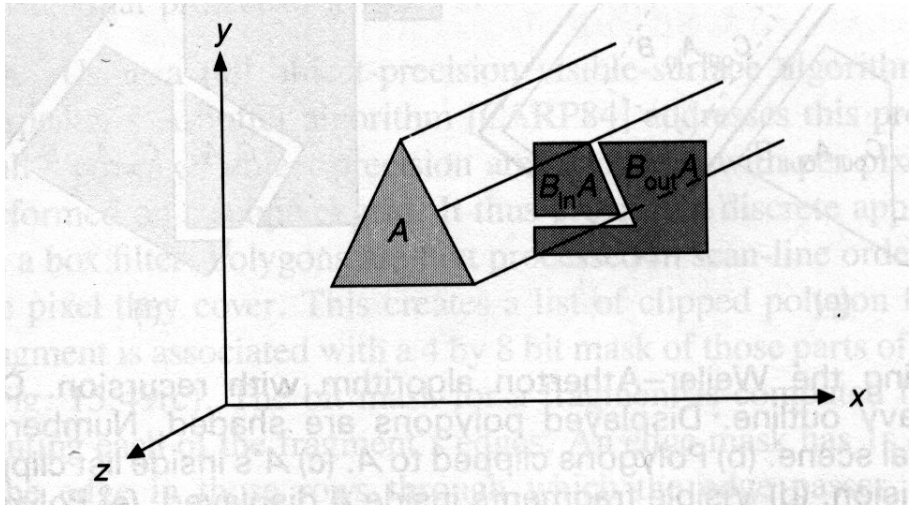


Image Synthesis
- Evaluation of Visibility - Algorithms in the object space -

Atherton & Weiller

Procedure:

1. Sort polygons by coordinated **Z**; the polygon that is the nearest to the observer is selected as the cutting window.

2. The polygon is used to perform *clipping* of the others, resulting **two lists** containing the polygons that are **interior (invisible)** and **exterior (visible)** to the *clipping* region.

3. **Interior polygons** are marked as invisible.

4. Advance to the next polygon in exterior list and go to step 2.

Image Synthesis
- Evaluation of Visibility - Algorithms in the object space-

Atherton & Weiller

In the figure the values indicate the Z coordinate of each vertex.

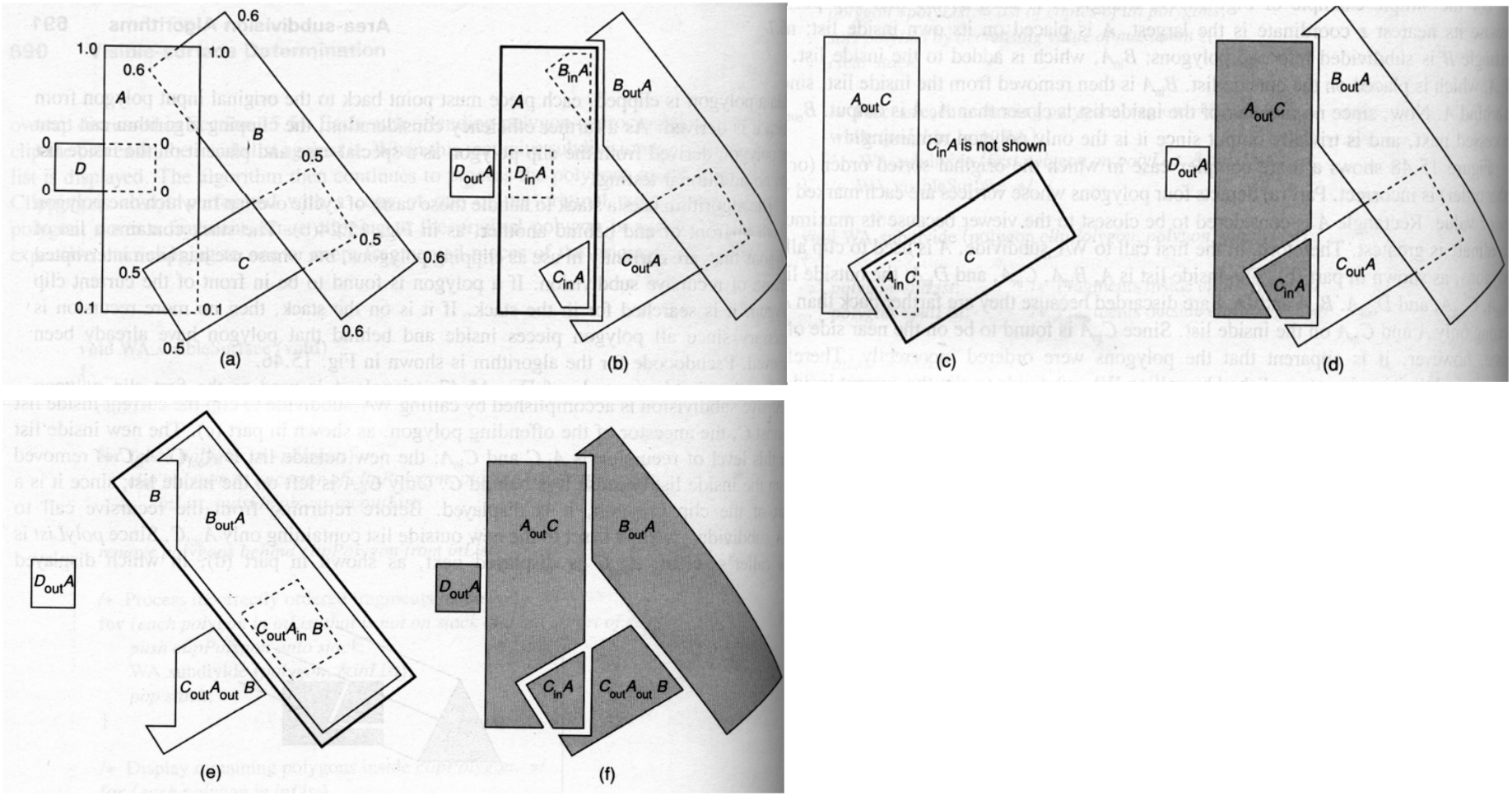


Image Synthesis
- Evaluation of Visibility - Algorithms in the image space

1. Oriented area: **Algorithm Warnock** (1969)
2. Oriented to the scan Line: **Scan Line/Watkins**
3. Oriented pixel: **Z-buffer, Ray Casting**

Algorithm of Warnock

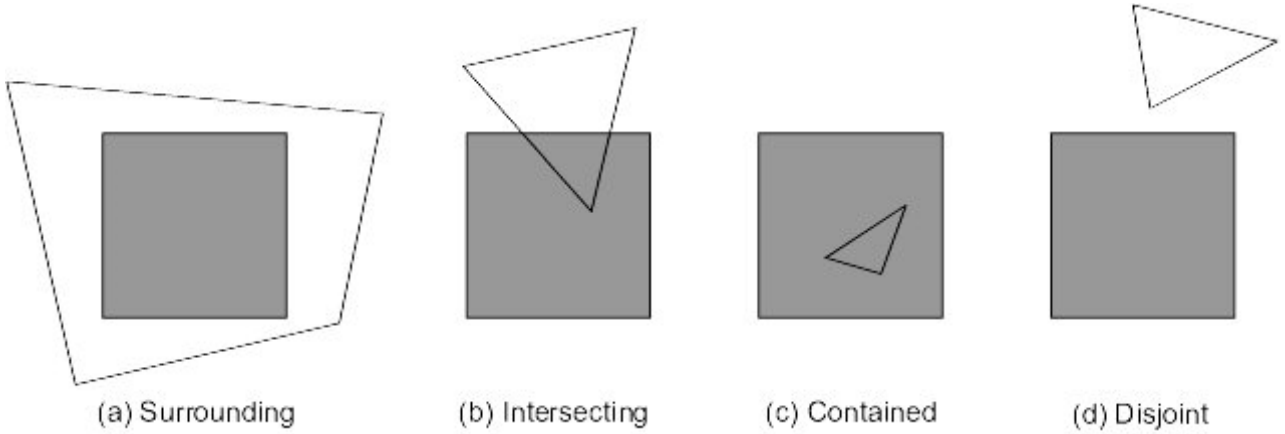
- The algorithm divides the projected image into rectangular areas.
- If one area is considered coherent,
  - Area is painted with contained polygons
  - **OTHERWISE**, the area is divided into smaller areas; the procedure is applied recursively.
- The smaller the rectangular areas, the fewer the polygons that are overlapping inside them; it is easier to decide which polygon is to be drawn.

The algorithm uses the concept of coherence area: A group of adjacent pixels is usually covered by the same visible surface.

Image Synthesis - Evaluation of Visibility
Algorithms in the image space: alg. Warnock

Procedure Alg. Warnock:

- Division of the area into 4 equal blocks. At each stage of division, the projection of each polygon will be in one of the four situations for each field:



- **a)-** Only one polygon covers the entire area, without any other → Paint the area with the polygon color.
- **b, c)-** Only a polygon that intersects or is fully in inside → Fill the area with the background color and then paint the part of the polygon which is inside this area.
- **d)-** All polygons are outside the area → Paint the area with the background color.

Image Synthesis - Evaluation of Visibility Algorithms in the image space : alg. Warnock

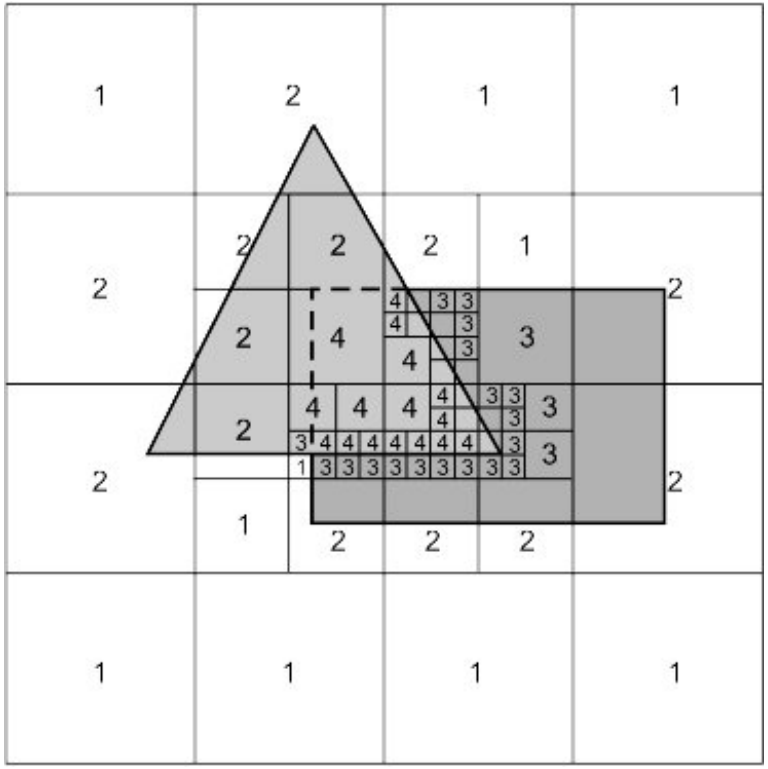


Image Synthesis - Evaluation of Visibility
Algorithms in the image space : scan line

Scan Line Algorithm (Alg. de Watkins)

The image is created line by line (scan lines), similar to the algorithm used to fill 2D regions, called **the list of active edges algorithm**.

Concepts used:

- **Vertical coherence**: the set of visible objects in a certain scan line, differs only a little from the one in the previous line.
- **Edge coherence**: One edge changes its visibility only when it crosses another visible edge or when it penetrates one face.

Data structures:

AEL - Active Edges List

ET - (new) Edges Table

PT - Polygons Table

Image Synthesis - Evaluation of Visibility
Algorithms in the image space : scan line

(Note: descriptors with more information than given in the book, in red)

Edges Table (ET): keeps track of all edges whose projection in the view plane is not horizontal. The table entries are ordered by the smallest value of Y, and initially contain:

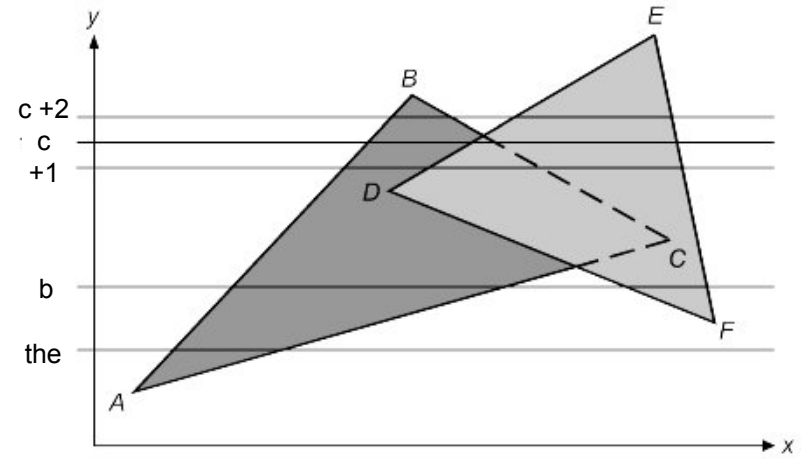
1. Coordinates (X, Z) of the lower Y vertex
2. Edge height (Y1-Y0) or, alternatively, Ymax
3. Increments $\Delta X/\Delta Y=\partial X/\partial Y$, $\Delta Z/\Delta Y=\partial Z/\partial Y$, used to update the X and Z in the transition to the next scan line
4. Identification of polygon(s) shared by the edge

Polygons Table (PT): information of all the polygons, each containing:

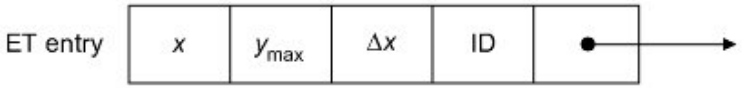
1. Coefficients of the plane equation (at least $\Delta Z/\Delta X =\partial Z/\partial X$)
2. Color information
3. Coordinate Z, to be updated in each pixel
4. Flag *in-out*, initialized as *False*, is used to control whether the procedure is inside or outside the polygon

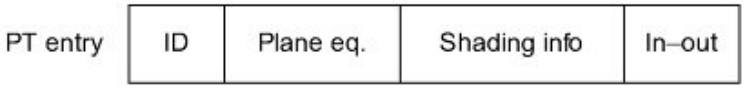
Active Edge List (AEL): information about which edges are active in the current scan line.

Image Synthesis - Evaluation of Visibility Algorithms in the image space: scan line



When checking the overlapping polygons, as in line ***c***, More than one polygon have their flag ***in-out*** the ***true***. Using equation of each polygon plane, each coordinate **Z** is determined to decide which one is closest to the observer.





AET contents

Scan line	Entries			
a	*AB*	*AC*		
b	*AB*	*AC*	*FD*	*FE*
c, c+1	*AB*	*DE*	*CB*	*FE*
c+2	*AB*	*CB*	*DE*	*FE*

Photo Synthesis - Evaluation of Visibility Algorithms in the image space: scan line

- Incremental update of an edge when moving to the next scan line:

$$\left.\begin{aligned} x &= x + \Delta y.\left(\frac{\partial x}{\partial y}\right) = x + \left(\frac{\partial x}{\partial y}\right) \\ z &= z + \Delta y.\left(\frac{\partial z}{\partial y}\right) = z + \left(\frac{\partial z}{\partial y}\right) \end{aligned}\right\}, \text{ since } \Delta y = 1$$

- Initialization of the depth of a polygon when *in-out* changes to TRUE:

$$z_{poly} = z_{edge}$$

- Update z_{poly} after Δx pixels:

$$z = z + \Delta x.\left(\frac{\partial z}{\partial x}\right)$$

Image Synthesis - Evaluation of Visibility
Algorithms in the image space: Z-Buffer

Z-Buffer Algorithm

One of the simplest algorithms implemented either in hardware or in software. Does not require any pre-processing of sorting nor performs comparison object-object.

Requirements: two memory buffers

- *Frame buffer*: contains the final image, pixel by pixel
- *Depth buffer / Z-Buffer*: contains Z values, pixel by pixel

Procedure:

1. Fill the *Z-Buffer* with zeroes (equivalent to the max Z) and the *frame buffer* with the background color (background).
2. Scan each polygon (scan-convert, pixel by pixel), in any order.
3. If the point (x, y) of the current polygon is closer to the observer than the current point of the Z-Buffer, then the current replaces the previous point.

Image Synthesis - Evaluation of Visibility Algorithms in the image space: Z-Buffer

Z-Buffer Algorithm

```
/* initialize buffers*/
for(y=0; y<YMAX; y++)
     for(x=0; x<XMAX; x++)
     {    ImBuffer(x,y) = BACKGROUND_COLOR;
          ZBuffer(x,y) = 0;
     }
for (each polygon)
{    for (each pixel in the polygon projection)
     {    /* calculates Z for (x,y) on the polygon*/
          pz = polygon_Z(x,y);
          if (pz > ZBuffer(x,y))
          {    ZBbuffer(x,y) = pz;
               ImBuffer(x,y) = polygon_color();
          }
     }
 }
```

Image Synthesis - Evaluation of Visibility Algorithms in the image space: Z-Buffer

Optimization in the procedure:

The *z* value in the point *(x+1, y)* in the polygon can be obtained from the value *z* in *(x, y)* if the polygon is flat/plane.

Taking *Ax+By+Cz+D=0*:

$$z = \frac{-D - Ax - By}{C}$$

If on *(x, y)* the equation has the value z_1 then in *(x+1, Y)* the value of *z* can be calculated:

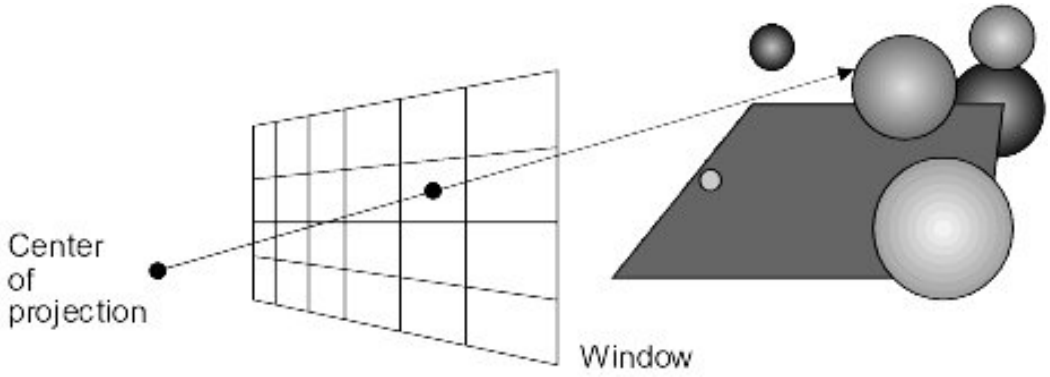
$$z = z_1 + \Delta x . \frac{\partial z}{\partial x} = z_1 + 1 . \frac{\partial z}{\partial x}$$

$$z = z_1 - \frac{A}{C}$$

Image Synthesis - Evaluation of Visibility
Algorithms in the image space: Ray-casting

Ray-casting algorithm

The visible surface at each pixel of the image is determined by tracing a ray of light from the imaginary center of projection (observer), passing through the center of the pixel to the 3D scene. The color at each pixel is defined by the color of the closest object at the intersection point.




```
Set Projection Center and window in the view plane
for (Each line of the image)
       for (Each pixel in the line)
       {      Define the ray from the center of projection, through that pixel
              for (Each object in the scene)
              {      if ((Object is Intersected) and (is closer than the previous object))
                         register the intersection and the reference to the object
              }
              assign the pixel the color of the closest intersected object
       }
```

Image Synthesis - Evaluation of Visibility
Alg type priority list

Type algorithms Priorities List

- Alg Newel Newel & Sancha
- Binary Space-Partitioning Trees

Objective: Determining the order of visibility for objects (polygons), ensuring that the image will be created correctly if the objects are drawn by a certain order:

1. Paint, first, the furthest faces to the observer
2. As other, closer faces, are being painted, they occlude the above ones.

(Algorithm "Painter")

Image Synthesis - Evaluation of Visibility
Alg type priority list -**Alg Newel Newel & Sancha**

Alg Newel Newel & Sancha (Depth-sort algorithm)

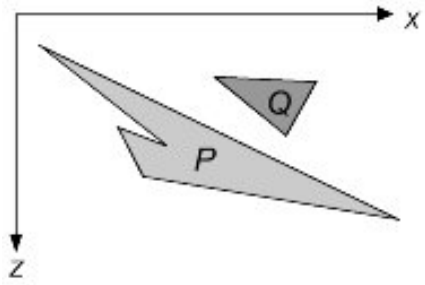
Procedure: paint the polygons in order of decreasing distance to the observer.

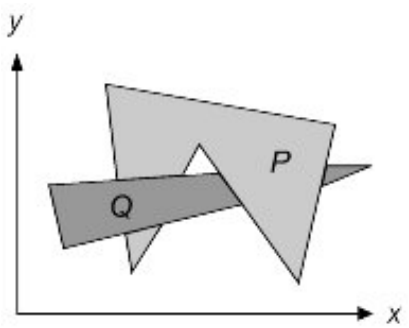
For this 3 steps are performed:

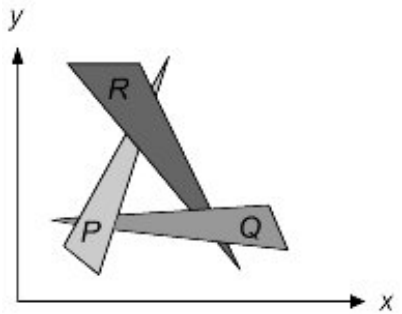
1. Sort the polygons by order of increasing ***z***
2. Resolve any ambiguity in the ordering, namely if there are overlapping polygons on coordinate ***z***. It may be necessary to split polygons.
3. Paint the polygons in order of farthest to nearest.

Image Synthesis - Evaluation of Visibility
Alg priority list type - -**Alg Newel Newel & Sancha**

Kind of ambiguities that can arise in the ordering of polygons:

a) 

b) 

c) 

Preprocessing:

Sort polygons by the coordinate ***Z of the farthest vertex***

Processing:

For the last polygon **P** in the list, check if there is any polygon **Q** whose greatest ***Z*** is further away than the lowest ***Z*** of **P** and that is being occluded by **P**. If not, then **P** can be drawn.