Scan Conversion (fixed function)

OCS
modeling transformation

WCS
viewing transformation

VCS
projection transformation

CCS

NDCS

DCS

OCS - object coordinate system
WCS - world coordinate system
VCS - viewing coordinate system
CCS - clipping coordinate system
NDCS - normalized device coordinate system
DCS - device coordinate system

Implicit, Explicit, and Parametric equations for defining geometry
Lines and Curves

**Explicit**
- line
- circle
- plane
- sphere

Lines and Curves

**Parametric**
- line
- circle
- plane
Lines and Curves

Implicit

line

circle

Polygons

Interactive graphics uses polygons

simple convex  simple concave  non-simple (self-intersection)
In practice we use triangles

- why?
  - simple convex polygons
    - trivial to break into triangles
  - concave or non-simple polygons
    - more effort to break into triangles

What is Scan Conversion?
(a.k.a. Rasterization)
Modern Rasterization

*Define a triangle as follows:*

**Scaled Implicit Line Equation**
**Edge Equations: Code**

```c
findBoundingBox(&xmin, &xmax, &ymin, &ymax);
setupEdges (&a0,&b0,&c0,&a1,&b1,&c1,&a2,&b2,&c2);

for (int y = yMin; y <= yMax; y++) {
    for (int x = xMin; x <= xMax; x++) {
        float e0 = a0*x + b0*y + c0;
        float e1 = a1*x + b1*y + c1;
        float e2 = a2*x + b2*y + c2;
        if (e0 > 0 && e1 > 0 && e2 > 0)
            Image[x][y] = TriangleColor;
    }
}
```

**Interpolation During Scan Conversion**

- Interpolate values from vertices to interior pixels:
  - $z$ depth values
  - $r,g,b$ colour components
  - $u,v$ texture coordinates
  - $N_x,N_y,N_z$ surface normals

- Equivalent ways of viewing this (for triangles)
  - plane equation
  - barycentric coordinates
  - bilinear interpolation
Plane Equation

- \( v = Ax + By + C \)

Barycentric Coordinates

- **weighted combination of vertices**

\[
P = \alpha \cdot P_1 + \beta \cdot P_2 + \gamma \cdot P_3
\]

\[
\alpha + \beta + \gamma = 1
\]

\[
0 \leq \alpha, \beta, \gamma \leq 1
\]

To interpolate a scalar quantity, \( v \), whose values are known at the vertices:

\[
v = \alpha \cdot v_1 + \beta \cdot v_2 + \gamma \cdot v_3
\]
Interpreting Barycentric Coordinates

• (1) coordinate system using edges as basis vectors
• (2) fractional distances
• (3) fractional areas
• (4) fractional weights
Interpolation: Screen vs World Space

Perspective-correct interpolation

\[ v = \frac{\alpha \cdot v_1 / h_1 + \beta \cdot v_2 / h_2 + \gamma \cdot v_3 / h_3}{\alpha / h_1 + \beta / h_2 + \gamma / h_3} \]

\[ v = \frac{\text{Barycentric}(v_1, v_2, v_3)}{\text{Barycentric}(1, 1, 1)} \]