## Ch14/15: Projection

Taxonomy of projections
- Parallel
- Perspective

Parallel projection:
– Center of projection is at infinity
– **Direction of projection (DOP)** is the **same** for all points (parallel)

Orthographic (DOP, VPN, coordinate axis are parallel)
Axonometric (DOP, VPN are parallel, coordinate axis is not)
Oblique (DOP and VPN are not parallel, VPN is parallel to coordinate axis)

Orthographic
Pros: accurate measurement, all views are same scale
Cons: perspective foreshortening, not realistic

**Simple orthographic transformation**

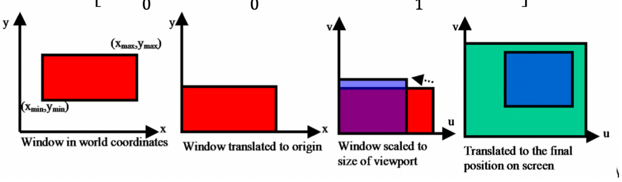
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ (front view)}$$

**View volume window to viewport transformation**

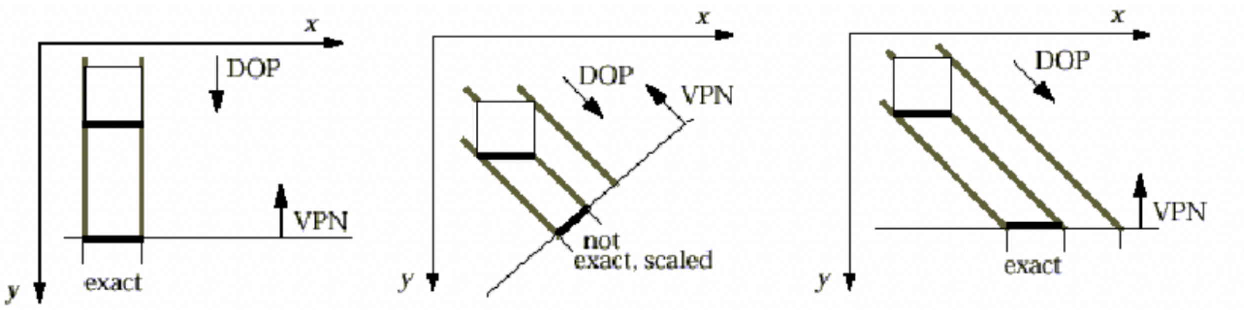
$$M_{W2V} = T(u_{min}, v_{min}) \cdot S\left(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}}\right) \cdot T(-x_{min}, -y_{min})$$

$$= \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & 0 \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & -x_{min} \cdot \frac{u_{max}-u_{min}}{x_{max}-x_{min}} + u_{min} \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & -y_{min} \cdot \frac{v_{max}-v_{min}}{y_{max}-y_{min}} + v_{min} \\ 0 & 0 & 1 \end{bmatrix}$$




Parallel projections summary



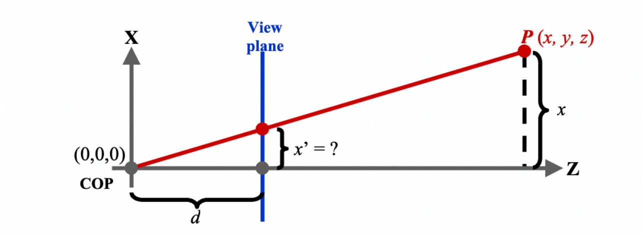
| Multiview Orthographic | Axonometric | Oblique |
| --- | --- | --- |
| VPN is parallel to a principal coordinate axis | VPN is not parallel to a principal coordinate axis | VPN is parallel to a principal coordinate axis |
| DOP is parallel to VPN | DOP is parallel to VPN | DOP is not parallel to VPN |
| Single face, exact measurements | Adjacent faces, none exact, uniformly foreshortened (function as angle between) | Adjacent faces, one exact, others uniformly foreshortened |

Perspective projection

Categories: 1–point perspective, 2–point, 3–point

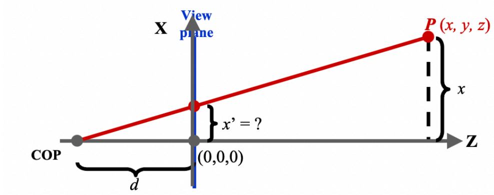
**Perspective projection matrix**

$$P_{perspective} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$$




**Perspective projection: origin in view plane**

$$P_{perspective} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix}$$



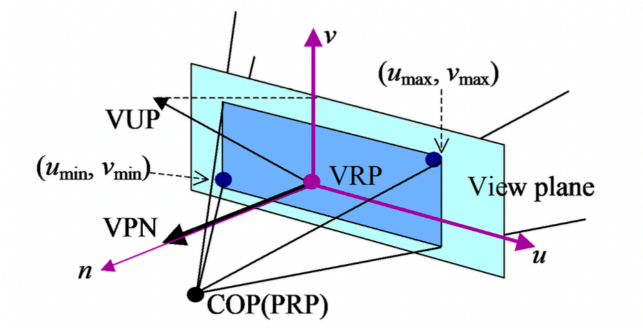

Perspective v.s. Parallel

| | Perspective projection | Parallel |
| --- | --- | --- |
| + | Size varies inversely with distance — looks **realistic** | Good for **exact measurement** |
| | | Parallel lines remain **parallel** |
| – | **Distance and angles** are not preserved | **Angles** are not preserve |
| | **Parallel lines** do not remain parallel | **Less realistic** looking |

## View Reference Coordinate System (VRC)

VUP: view reference point

VPN: view plane normal




$$n = \frac{VPN}{|VPN|}, u = \frac{VUP \times VPN}{|VUP \times VPN|}, \text{v} = n \times u$$

## Transform world coordinate in VRC

1. Translation of the coordinate system to the origin in homogeneous matrix form:

$$T = \begin{bmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Using the unit vectors of the coordinate axes, the resulting rotation matrix is:

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Combination for the single transformation matrix (parallel)

$$M = T \cdot R = \begin{bmatrix} u_x & u_y & u_z & -(u_x \cdot VRP_x + u_y \cdot VRP_y + u_z \cdot VRP_z) \\ v_x & v_y & v_z & -(v_x \cdot VRP_x + v_y \cdot VRP_y + v_z \cdot VRP_z) \\ n_x & n_y & n_z & -(n_x \cdot VRP_x + n_y \cdot VRP_y + n_z \cdot VRP_z) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Parallel or Perspective projection:

$$P_{perspective} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \text{ or } P'_{perspective} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix} \text{ or } P''_{perspective} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Combination for the single transformation matrix:

$M = P \cdot R \cdot T$ (perspective projection with origin at d distance from the view plane)

$M' = P' \cdot R \cdot T$ (perspective projection with origin in the view plane)

$M'' = P'' \cdot R \cdot T$ (parallel projection)

Question:

a) This question is about projection.

**[16 marks]**

i) Consider **Figure 3**. Give the perspective projection transformation matrix for point P(x,y,z) projecting to the view plane at z=2 if the centre of projection is the origin.

**(6 marks)**

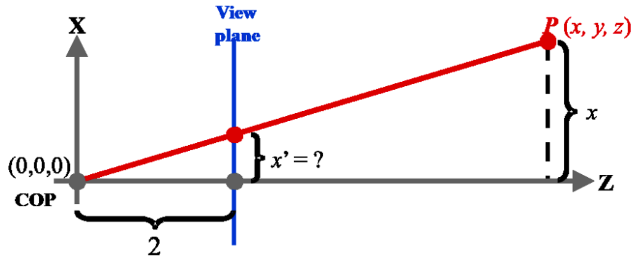



**Figure 3.**

ii) In the world coordinate system, if the view plane normal is [4,3,0], the view-up vector is [0,0,1], the view reference point is [1,2,3], find the transformation matrix for parallel

**(10 marks)**

(1) Assign the $d$ as the distance between COP to the view plane

$$\frac{x'}{d} = \frac{x}{z} \qquad \frac{y'}{d} = \frac{y}{z} \qquad w = \frac{z}{d}. \longrightarrow \begin{cases} x' = \frac{x}{w} \\ y' = \frac{y}{w} \\ z' = \frac{z}{w} \\ w = \frac{z}{d} \end{cases}$$

$$x' = \frac{x}{z} \cdot d = \frac{x}{\frac{z}{d}} \qquad y' = \frac{y}{\frac{z}{d}}$$

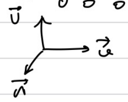
$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{z}{d} \end{bmatrix}$$

$$P_{perspective} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$$

(2) $\vec{VPN} = [4, 3. 0] \qquad VRP = [1, 2, 3]$

$\vec{VUP} = [0, 0, 1]$

① translate to origin : $T = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$




② Rotation matrix ( R)

$$\vec{n} = \frac{\vec{VPN}}{|\vec{VPN}|} = [\tfrac{4}{5}, \tfrac{3}{5}, 0]$$

$$\vec{u} = \frac{\vec{VUP} \times \vec{VPN}}{|\vec{VUP} \times \vec{VPN}|} \qquad \vec{VUP} \times \vec{VPN} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 0 & 1 \\ 4 & 3 & 0 \end{vmatrix} = \vec{i} \begin{vmatrix} 0 & 1 \\ 3 & 0 \end{vmatrix} - \vec{j} \begin{vmatrix} 0 & 1 \\ 4 & 0 \end{vmatrix} + \vec{k} \begin{vmatrix} 0 & 0 \\ 4 & 3 \end{vmatrix}$$

$$= -3\vec{i} + 4\vec{j} + 0 \cdot \vec{k} = [-3, 4, 0]$$

$$\vec{u} = [-\tfrac{3}{5}, \tfrac{4}{5}, 0] , \quad \vec{v} = \vec{n} \times \vec{u} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{4}{5} & \frac{3}{5} & 0 \\ -\frac{3}{5} & \frac{4}{5} & 0 \end{vmatrix} = 0 + 0 + \vec{k} \begin{vmatrix} \frac{4}{5} & \frac{3}{5} \\ -\frac{3}{5} & \frac{4}{5} \end{vmatrix} = \vec{k} = [0, 0, 1]$$

$$R = \begin{bmatrix} -\frac{3}{5} & \frac{4}{5} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{4}{5} & \frac{3}{5} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

③ parallel projection matrix : $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$M = P \cdot R \cdot T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{3}{5} & \frac{4}{5} & 0 & -1 \\ 0 & 0 & 1 & -3 \\ \frac{4}{5} & \frac{3}{5} & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\frac{3}{5} & \frac{4}{5} & 0 & -1 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question:

a) This question is about projection.

**[16 marks]**

i) Give the transformation matrix for a 2D window-to-viewport transformation. Use the variables given in **Figure 3**.

**(5 marks)**

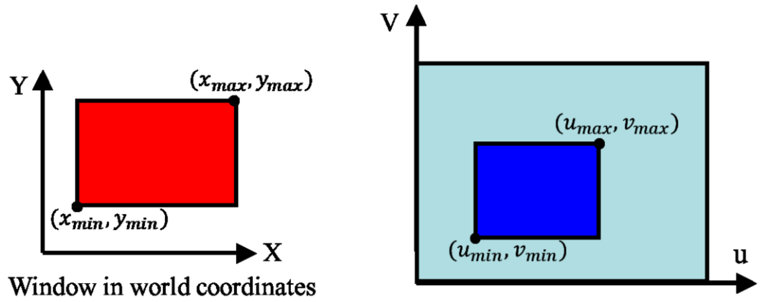



**Figure 3.**

ii) In the world coordinate system, if the view plane normal is [2,2,1], the view-up vector is [0,1,1], the view reference point is [5,6,7], and the distance from the centre of projection to the view plane is 5, find the transformation matrix for perspective projection.

**(11 marks)**

(1) $M = T(u_{min}, v_{min}) \cdot S\left(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}}\right) \cdot T(-x_{min}, -y_{min})$

T represents translation, S represents scaling

$$T(-x_{min}, -y_{min}) = \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

$$S\left(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}}\right) = \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & 0 \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(u_{min}, v_{min}) = \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & 0 \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & -x_{min} \cdot \left(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}\right) \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & -y_{min} \cdot \left(\frac{v_{max}-v_{min}}{y_{max}-y_{min}}\right) \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & u_{min} - x_{min} \cdot \left(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}\right) \\ 0 & \frac{v_{max}-u_{min}}{y_{max}-y_{min}} & v_{min} - y_{min} \cdot \left(\frac{v_{max}-v_{min}}{y_{max}-y_{min}}\right) \\ 0 & 0 & 1 \end{bmatrix}$$

(2) $M = P \cdot R \cdot T$

The translation matrix to the view reference point

$$T = \begin{bmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

the 3-axes vectors are:

$$\vec{n} = \frac{\overrightarrow{VPN}}{|\overrightarrow{VPN}|} = \frac{[2\ 2\ 1]}{\sqrt{2^2+2^2+1^2}} = \frac{1}{3}[2\ 2\ 1] = \left[\tfrac{2}{3}\ \tfrac{2}{3}\ \tfrac{1}{3}\right]$$

$$\vec{u} = \frac{\overrightarrow{VUP} \times \overrightarrow{VPN}}{|\overrightarrow{VUP} \times \overrightarrow{VPN}|}, \quad \overrightarrow{VUP} \times \overrightarrow{VPN} = \begin{vmatrix} i & j & k \\ 0 & 1 & 1 \\ 2 & 2 & 1 \end{vmatrix}$$

$$= \vec{i} \cdot \begin{vmatrix} 1 & 1 \\ 2 & 1 \end{vmatrix} - \vec{j} \begin{vmatrix} 0 & 1 \\ 2 & 1 \end{vmatrix} + \vec{k} \begin{vmatrix} 0 & 1 \\ 2 & 2 \end{vmatrix}$$

$$= -\vec{i} + 2\vec{j} - 2\vec{k} = [-1\ 2\ -2]$$

$$\vec{u} = \frac{[-1\ 2\ -2]}{\sqrt{(-1)^2+2^2+(-2)^2}} = \left[-\tfrac{1}{3}\ \tfrac{2}{3}\ -\tfrac{2}{3}\right]$$

$$\vec{v} = \vec{n} \times \vec{u} = \begin{vmatrix} i & j & k \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{2}{3} \end{vmatrix} = \vec{i} \begin{vmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{2}{3} \end{vmatrix} - \vec{j} \begin{vmatrix} \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & -\frac{2}{3} \end{vmatrix} + \vec{k} \begin{vmatrix} \frac{2}{3} & \frac{2}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{vmatrix}$$

$$= -\frac{6}{9}\vec{i} - \vec{j} \cdot \left(-\frac{3}{9}\right) + \vec{k} \cdot \left(\frac{6}{9}\right) = -\frac{2}{3}\vec{i} + \frac{1}{3}\vec{j} + \frac{2}{3}\vec{k}$$

$$= \left[-\tfrac{2}{3}\ \tfrac{1}{3}\ \tfrac{2}{3}\right]$$

The rotation matrix is:

$$R = \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} & -\frac{2}{3} & 0 \\ -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} & 0 \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The perspective matrix is:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{5} & 0 \end{bmatrix}$$

The composite transformation matrix is: $(d=5)$

$$M = P \cdot R \cdot T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{5} & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} & -\frac{2}{3} & 0 \\ -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} & 0 \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{5} & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} & -\frac{2}{3} & \frac{7}{3} \\ -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} & -\frac{10}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{29}{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} & -\frac{2}{3} & \frac{7}{3} \\ -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} & -\frac{10}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{29}{3} \\ \frac{2}{15} & \frac{2}{15} & \frac{1}{15} & -\frac{29}{15} \end{bmatrix}$$

## Ch16: Rasterization

Faster Approaches

1. Use **integer** calculation
2. Avoid **divides and multiplications**
3. Use **incremental computations**
4. Use **spatial coherence**

Digital Differential Analyzer (DDA):

x = x + 1 | y = y + round(dy/dx)

### Bresenham's Midpoint Line Algorithm

- Initial value $2(dy)-(dx)$
- Choose $\begin{cases} E & \text{if } d \le 0 \\ NE & \text{otherwise} \end{cases}$
- Case E: $d \leftarrow d + \Delta_E$, where $\Delta_E = 2(dy)$
- Case NE: $d \leftarrow d + \Delta_{NE}$,

  where $\Delta_{NE} = 2\{(dy)-(dx)\}$

- Note, all deltas are constants

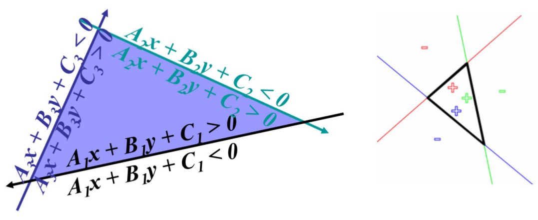
Triangle rasterization issues

- Shape issue
- Position issue
- Shared edge issue

### Rasterising triangles

1. Put vertices in a right order so that the inside is always in positive half-plane
2. Test if the point makes the three line equations of the three edges positive
3. If all positive, it is inside. If not, it is outside

- For an edge of 2 vertices, take the third vertex as in the positive half-space, a triangle can be defined as the intersection of three positive half-spaces




在判断的是否为positive half-plane时可以用三角形的第三个顶点代入直线公式，看是否大于0

## Filling Polygons

Parity test: if odd, set pixels; if even, do not set pixels

Active edge table (AET): consists all the intersection points of edges with the current scanline

- intersection points are **sorted by increasing x** from left to right
- Next scanline, AET is **updated**

Filling Polygon Algorithm

1. Sort all edges by their **minimum y coordinate**
2. Starting at the bottom, add **edges with Ymin = 0** to AET
3. For each scanline:
    1. **Sort** edges in AET by **x intersection**
    2. Walk from left to right, set pixels by **parity test**
    3. **Increment** scanline
    4. **Retire edges** with Ymax < Y, **add edges** with Ymin < Y
    5. **Recalculate intersections**
4. Stop if Y > Ymax for last edges

Question:

b) This question is about rasterisation. Use the line equation for a line from (x0, y0) to (x1, y1) to derive the mid-point algorithm for line generation.

**[9 marks]**

(b)

The line equation is: $\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$

Implicit:

$F(x,y) = (y_1 - y_0)(x - x_0) - (x_1 - x_0)(y - y_0)$ ⇒ 顺序不能反

$F(x+1, y) - F(x,y) = y_1 - y_0$

$F(x+1, y+1) - F(x,y) = (y_1 - y_0) - (x_1 - x_0)$

$F(x+1, y+\frac{1}{2}) - F(x,y) = (y_1 - y_0) - \frac{1}{2}(x_1 - x_0)$

$2F(x,y) = 0$, $dy = y_1 - y_0$, $dx = x_1 - x_0$

$d(x_0, y_0) = 2dy - dx$ ⇒ initial value

$d(x+1, y) = 2dy$ ⇒ move east (E)

$d(x+1, y+1) = 2dy - 2dx$ ⇒ move north-east (NE)

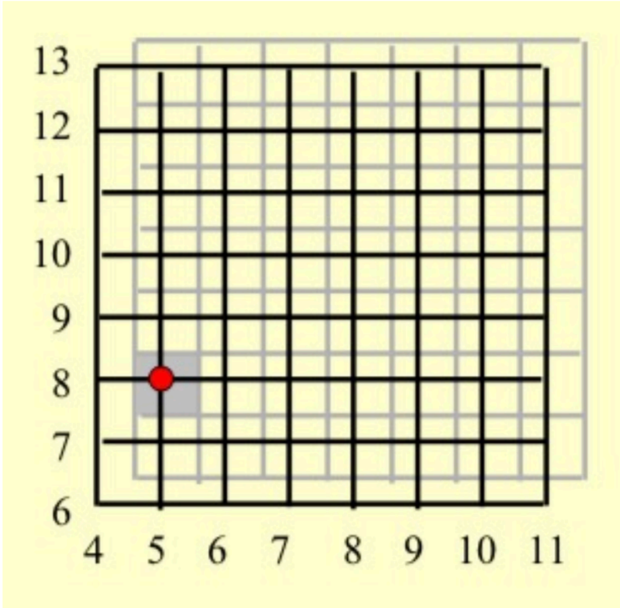
Question:

- Example

- Line end points:

$$(x_0, y_0) = (5,8); \quad (x_1, y_1) = (9,11)$$

- Deltas: $dx = 4;\ dy = 3$




$dy = 11 - 8 = 3$ , $dx = 9 - 5 = 4$

$d(x_0, y_0) = 2dy - dx = 6 - 4 = 2$

$d(x+1, y) = 2dy = 6$ (E)

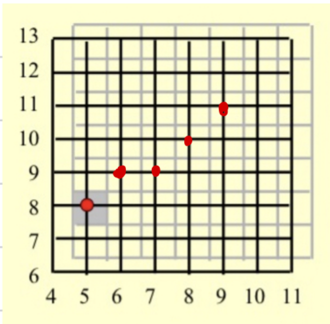
$d(x+1, y+1) = 2(dy - dx) = 2(3 - 4) = -2$ (NE)

① $d(x_0, y_0) > 0 \Rightarrow$ move NE

② $d = 2 - 2 = 0 \Rightarrow$ move E

③ $d = 0 + 6 = 6 \Rightarrow$ move NE

④ $d = 6 - 2 = 4 \Rightarrow$ move NE



Question:

b) This question is about rasterisation.

**[9 marks]**

i) For a line from (x0, y0) to (x1, y1), give the implicit function as the line equation F(x,y)=0. Give the normal vector of the line and show which half-plane makes F(x,y)>0 and which half-plane makes F(x,y)<0.

**(5 marks)**

ii) Give a method to test if a point is inside or outside a triangle.

**(4 marks)**

(1)

$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{y - y_0}{x - x_0}$$

$F(x,y) = (y_1 - y_0)(x - x_0) - (x_1 - x_0)(y - y_0) = 0$

$Ax + By + C = 0$ normal vector is $[A, B]$

normal vector: $[y_1 - y_0, -(x_1 - x_0)]$

In the normal vector positive side is positive:

(Assume $(x', y')$ in the half plane)

$< (y_1 - y_0), -(x_1 - x_0) > \cdot < (y' - y_0), (x' - x_0) > \; > 0$

In the normal vector negative side is negative:

$< (y_1 - y_0), -(x_1 - x_0) > \cdot < (y' - y_0), (x' - x_0) > \; < 0$

(2) First, ordering the vertices of the triangles so that the inside of the triangle is the positive half-planes for all three edges of triangles. Second, test the point that if it makes the three line equations positive. Third, if they are all positive, it is in the triangle, if not, then it is negative.

Question: the algorithm for filling polygon

1. Sort all the edges by their minimum y coordinate
2. Starting at the bottom point, add edges with Ymin = 0 to the AET
3. For each scanline:
    1. Sort edges in AET by their x intersection
    2. Walk from left to right, set pixels by using parity test
    3. Increment scanline
    4. Retire edges with **Ymax < Y**, add edges with **Ymin < Y**
    5. **Recalculate intersections**
4. Stop when Y > Ymax for the last edges