

การสร้างภาพโดยคอมพิวเตอร์

ในบทนี้จะกล่าวถึง ทฤษฎีที่ใช้ในการนำข้อมูลของแบบจำลอง ที่มีอยู่ในหน่วยความจำ ของระบบคอมพิวเตอร์ มาสร้างเป็นภาพ 2 มิติ และ 3 มิติ บนจอภาพรวมทั้งวิธีการกำจัด เส้น และพื้นผิวที่ถูบบังค้ำ โดยวิธีของChan S. Park(9)และวิธีของFoley J.D.และA. Van Dam(10)

4.1 ทฤษฎีในการสร้างภาพ 2 มิติ

ทฤษฎีในการสร้างภาพ 2 มิติ นี้จะประกอบไปกับการสร้างภาพ 2 มิติ ของแบบจำลอง บนจอภาพ ซึ่งนอกจากการสร้างตามปกติแล้ว บางครั้งเราต้องการขยายภาพให้ใหญ่ขึ้น ต้องการ หมุนภาพไปรอบแกนที่กำหนดอีกด้วย ดังจะกล่าวในรายละเอียดต่อไป

4.1.1 การเปลี่ยนแปลงของจุด (Transformation of Points) ในระบบ 2 มิติ  
ถ้าเรามีจุด P ซึ่งมีค่าพิกัดเป็น (x,y) แล้วจุด P ถูกเคลื่อนย้ายไปเป็น P' = (x',y') ความสัมพันธ์ระหว่างจุดทั้งสอง คือ

$$P' = PT = (x,y) \begin{bmatrix} A & B \\ C & D \end{bmatrix} = [(Ax + Cy), (Bx + Dy)] \quad (4.1)$$

โดย  $T = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$  เรียกว่า Transformation Matrix

ถ้า B = C = 0 และ A = D = 1 Matrix T จะเรียกว่า Identity Matrix

$$(x',y') = (x,y) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = (x,y) \quad (4.2)$$

ลักษณะ ดังสมการที่ 4.2 นั้น เรียกว่า Identity Transformation ซึ่งแสดงว่า จุด P ไม่มีการเปลี่ยนแปลง

ถ้าให้  $D = 1$  และ  $B = C = 0$  ผลที่ได้ คือ

$$(x', y') = (x, y) \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} = (Ax, y) \quad (4.3)$$

ลักษณะดังสมการที่ 4.3 เรียกว่าการเปลี่ยนแปลงขนาด (Scaling) ในทิศทางของแกน  $x$  ดังแสดงในรูป 4.1 (ก)

ถ้าให้  $A = 1$  และ  $B = C = 0$  ผลที่ได้ คือ

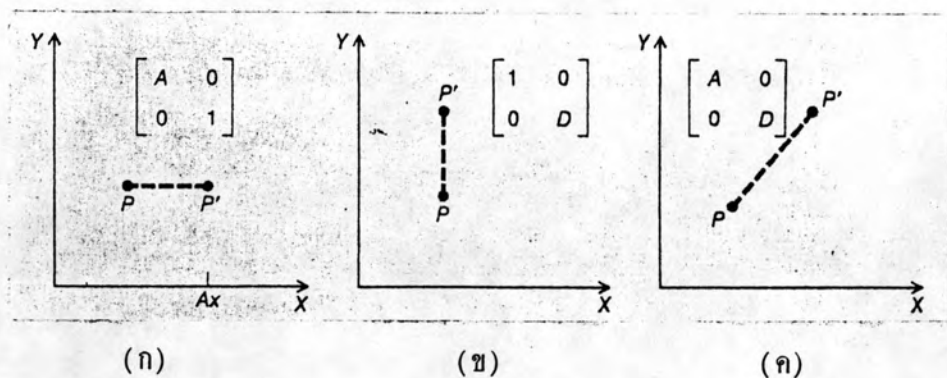
$$(x', y') = (x, y) \begin{bmatrix} 1 & 0 \\ 0 & D \end{bmatrix} = (x, Dy) \quad (4.4)$$

ลักษณะดังสมการที่ 4.4 เรียกว่า การเปลี่ยนแปลงขนาด (Scaling) ในทิศทางของแกน  $y$  ดังแสดงในรูป 4.1 (ข)

ถ้าให้  $B = C = 0$  ผลที่ได้ คือ

$$(x', y') = (x, y) \begin{bmatrix} A & 0 \\ 0 & D \end{bmatrix} = (Ax, Dy) \quad (4.5)$$

ลักษณะดังสมการที่ 4.5 เรียกว่า การเปลี่ยนแปลงขนาด (Scaling) ในทิศทางของแกน  $x$  และ  $y$  ดังแสดงในรูป 4.1 (ค)



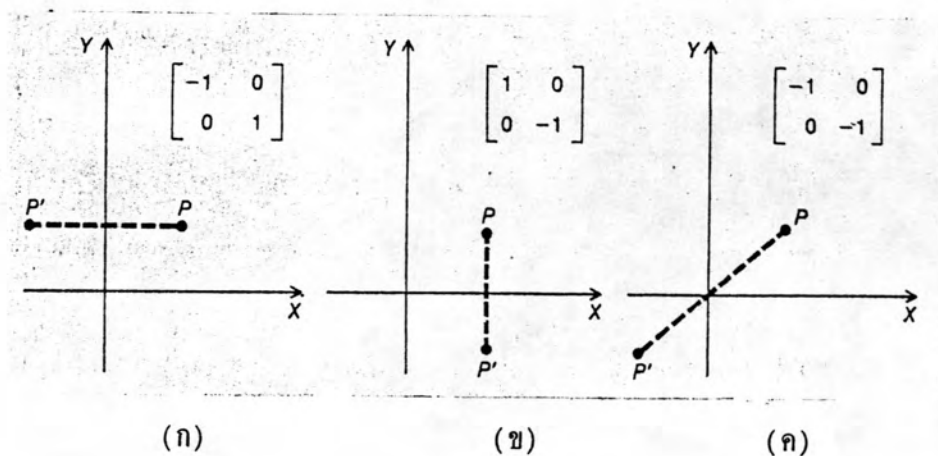
รูปที่ 4.1 แสดงการเปลี่ยนแปลงของจุดโดย Scaling

ถ้าให้ค่า  $A = 1, B = C = 0$  และ  $D = 1$  ผลที่ได้ คือ

$$(x', y') = (x, y) \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = (-x, y) \quad (4.6)$$

ลักษณะคัมสมการ 4.6 นี้ เรียกว่า การสะท้อน (Reflection) ผ่านแกน  $Y$  ดังแสดงในรูป 4.2 (ก)

ถ้าให้ค่า  $A = 1, B = C = 0$  และ  $D = -1$  จะเป็นการสะท้อน (Reflection) ผ่านแกน  $X$  ถ้าให้ค่า  $A = D = -1$  และ  $B = C = 0$  จะเป็นการสะท้อน (Reflection) ผ่านจุดกำเนิด (Origin) ดังแสดงในรูป 4.2 (ข) และ 4.2 (ค) ตามลำดับ



รูปที่ 4.2 แสดงการเปลี่ยนแปลงของจุดโดย Reflection

ถ้าให้ค่า  $A = D = 1$  และ  $C = 0$  ผลที่ได้ คือ

$$(x', y') = (x, y) \begin{bmatrix} 1 & B \\ 0 & 1 \end{bmatrix} = (x, Bx + y) \quad (4.7)$$

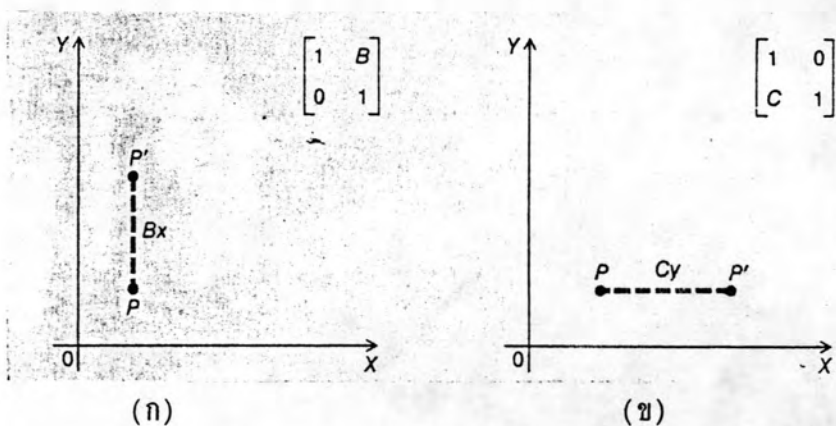
เรียกสมการ 4.7 ว่าเป็นการเฉือน (Shear) ในทิศทางแกน  $Y$  ดังแสดงในรูป 4.3 (ก)

ถ้าให้  $A = D = 1$  และ  $B = 0$  ผลที่ได้ คือ

$$(x', y') = (x, y) \begin{bmatrix} 1 & 0 \\ C & 1 \end{bmatrix} = (x + Cy, y) \quad (4.8)$$

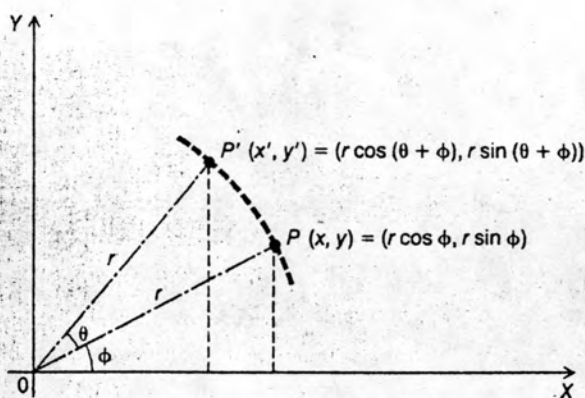
เรียกสมการ 4.8 ว่าเป็นการเฉือน (Shear) ในทิศทางแกน  $x$  ดังแสดงในรูป

4.3 (ข)



รูปที่ 4.3 แสดงการเปลี่ยนแปลงของจุดโดย Shear

ในกรณีที่มีการหมุนรอบจุดกำเนิด (Origin) ให้พิจารณา รูปที่ 4.4 ซึ่งจุด  $P(x, y)$  หมุนรอบจุด  $o$  ในทิศทางทวนเข็มนาฬิกาไปเป็นมุม  $\phi$



รูปที่ 4.4 แสดงการหมุนของจุดในระบบพิกัด 2 มิติ

จากรูปที่ 4.4 เราได้  $x = r \cos \phi$  และ  $y = r \sin \phi$  และได้

$$\begin{aligned} x' &= r \cos (\theta + \phi) \\ &= r \cos \phi \cos \theta - r \sin \phi \sin \theta \\ &= x \cos \theta - y \sin \theta \end{aligned}$$

$$\begin{aligned} y' &= r \sin (\theta + \phi) \\ &= r \cos \phi \sin \theta + r \sin \phi \cos \theta \\ &= x \sin \theta + y \cos \theta \end{aligned}$$

นั่นคือ  $(x', y') = (x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta)$

หรือ  $(x', y') = (x, y) \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$  (4.9)

ในกรณีที่ จุด P หมุนตามเข็มนาฬิกา เราใช้ค่า  $-\theta$  แทน  $\theta$

จะได้  $(x', y') = (x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$

หรือ  $(x', y') = (x, y) \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$  (4.10)

4.1.2 การเปลี่ยนแปลงของเส้นและวัตถุ (Transformation of Lines and Objects) การเปลี่ยนแปลงของเส้นเราใช้วิธีเปลี่ยนแปลงจุดที่ปลายทั้งสองข้างของเส้นก่อน แล้วลากเส้นใหม่ระหว่างจุดที่ถูกเปลี่ยนแปลงแล้วนั้น ในกรณีของวัตถุซึ่งจะประกอบไปด้วย เส้นหลายๆเส้น เราจะเปลี่ยนแปลงวัตถุนั้นโดยการเปลี่ยนแปลงจุดยอด (Vertices) ซึ่งเป็นจุดที่เส้น 2 เส้น มาพบกันโดยใช้สมการ 4.1 ถ้า วัตถุนั้นมีจุดยอดอยู่  $n$  จุด การเปลี่ยนแปลง จะมีรูปแบบดังนี้

ค่าพิกัดที่ถูกเปลี่ยนแปลงแล้ว = ค่าพิกัดเดิมของวัตถุ  $\cdot$  Matrix ที่ใช้เปลี่ยนแปลงค่าพิกัด

$$\begin{bmatrix} P'_1 \\ P'_2 \\ \vdots \\ \vdots \\ \vdots \\ P'_n \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ x'_n & y'_n \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (4.11)$$

ตัวอย่างของการใช้สมการ 4.11 ให้พิจารณา รูปที่ 4.5 (ก) เป็นรูปสามเหลี่ยม  
 คาพิกัคของจุดยอดทั้งสาม คือ  $(0,0)$ ,  $(30,0)$  และ  $(15,30)$  ถ้าขยายรูปนี้โดยให้  $A = D = 2$   
 จะได้ผลดัง รูปที่ 4.5 (ข) โดยมีคาพิกัคของจุดยอดใหม่ ดังนี้

$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 30 & 0 \\ 15 & 30 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 60 & 0 \\ 30 & 60 \end{bmatrix}$$

ถ้าต้องการหมุนรูปสามเหลี่ยมดังรูปที่ 4.5 (ข) หวนเข็มนาฬิกาไป  $30^\circ$  รอบจุด  
 Origin จะได้ผลดังรูปที่ 4.5 (ค) โดยมีคาพิกัคของจุดยอดใหม่ ดังนี้

$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 30 & 0 \\ 15 & 30 \end{bmatrix} \begin{bmatrix} \cos 30^\circ & \sin 30^\circ \\ -\sin 30^\circ & \cos 30^\circ \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 \\ 30 & 0 \\ 15 & 30 \end{bmatrix} \begin{bmatrix} 0.8660 & 0.5 \\ -0.5 & 0.8660 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 \\ 52 & 30 \\ -4 & 67 \end{bmatrix}$$

ในกรณีที่ต้องการเคลื่อนย้าย (Translation) วัตถุไปบนระนาบนั้น สามารถทำได้ โดยบวกขนาดของระยะทางที่จะเคลื่อนย้ายวัตถุนั้น เข้ากับค่าพิกัดของจุดยอดเดิมของวัตถุ ดังนี้

ค่าพิกัดที่ถูกเคลื่อนย้ายแล้ว = ค่าพิกัดเริ่มต้น + ขนาดที่จะเคลื่อนย้าย

$$\begin{bmatrix} P'_1 \\ P'_2 \\ \vdots \\ P'_n \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} + \begin{bmatrix} T_x & T_y \\ T_x & T_y \\ \vdots & \vdots \\ T_x & T_y \end{bmatrix}$$

โดยที่  $T_x$  และ  $T_y$  คือ ขนาดของระยะทางที่จะเคลื่อนย้ายวัตถุนั้นไปในแนวแกน  $x$  และ  $y$  ตามลำดับ ตัวอย่างเช่นถ้าเราจะเคลื่อนย้ายรูป 4.5 (ค) ไปทางขวา 100 หน่วย และเคลื่อนขึ้นไปข้างบน 50 หน่วย ( $T_x = 100$ ,  $T_y = 50$ ) เราจะได้ผลดัง รูปที่ 4.5 (ง) และมีค่าพิกัดของจุดยอดใหม่ ดังนี้

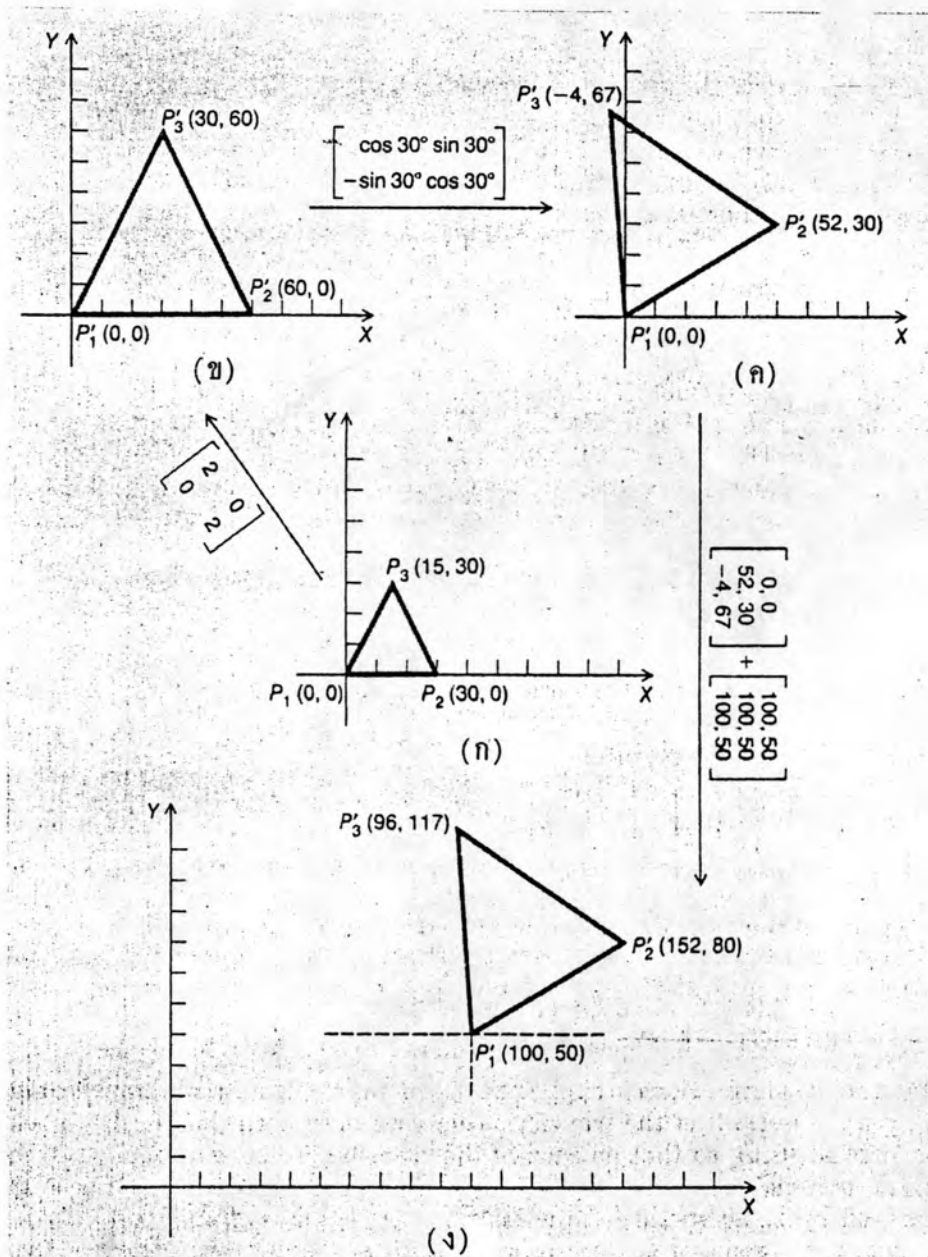
$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 52 & 30 \\ -4 & 67 \end{bmatrix} + \begin{bmatrix} 100 & 50 \\ 100 & 50 \\ 100 & 50 \end{bmatrix} = \begin{bmatrix} 100 & 50 \\ 152 & 80 \\ 96 & 117 \end{bmatrix}$$

วิธีการต่างๆ ในการเปลี่ยนแปลง (Transformation) ดังกล่าวมาแล้วนั้น ยังมีข้อจำกัดอยู่ 2 ประการ คือ

ประการแรก การหมุน (Rotation) กระทำไ้รอบจุด Origin (0,0) เท่านั้นและ การสะท้อนก็ทำไ้รอบแกน x,y และ Origin เท่านั้น

ประการที่สอง รูปแบบการ Transform มีทั้งคูณและบวก ทำให้ไม่สะดวกในการใช้ กับคอมพิวเตอร์

ดังนั้น ระบบพิกัดจึงต้องปรับปรุงใหม่ดังจะกล่าวในหัวข้อ 4.1.3 ต่อไป



รูปที่ 4.5 แสดงการเปลี่ยนแปลงรูปสามเหลี่ยมใน 2 มิติ โดย (ก) เป็นรูปเริ่มต้น (ข) Scaling (ก) Rotation และ (ง) Translation



### 4.1.3 ระบบพิกัดซึ่งเข้ากันได้ (Homogeneous Coordinate Systems)

ในการเปลี่ยนแปลง (Transformation) จุด เส้น และ วัตถุ ดังที่กล่าวมาแล้วนั้น ไม่เหมาะสมที่จะใช้กับระบบคอมพิวเตอร์ ดังนั้น จึงขยาย Transformation Matrix (T) จาก  $2 \times 2$  Matrix เป็น  $3 \times 3$  Matrix ดังนี้

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \longrightarrow \begin{bmatrix} A & B & 0 \\ C & D & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

และเพิ่มค่าในการเคลื่อนย้าย (Translation) ลงไปในแถวที่ 3 ซึ่งจะเปลี่ยนจาก  $(0 \ 0 \ 1)$  เป็น  $(L \ M \ 1)$  โดย L เป็นค่าการเคลื่อนย้ายในแนวระดับ และ M เป็นค่าการเคลื่อนย้ายในแนวตั้ง ดังนั้น Transformation Matrix จะเป็น

$$T = \begin{bmatrix} A & B & 0 \\ C & D & 0 \\ L & M & 1 \end{bmatrix} \quad (4.16)$$

ซึ่งคอลัมน์ที่ 3 ของ T จะช่วยให้เราสามารถรวม Translation Scaling และ Rotation เข้าด้วยกันได้โดยวิธีคูณ ดังนั้นการเปลี่ยนแปลงค่าพิกัดของจุด  $(x, y)$  จะมีรูปแบบใหม่ ดังนี้

$$(x', y', 1) = (x, y, 1) T$$

$$\text{หรือ} \quad x' = Ax + Cy + L$$

$$y' = Bx + Dy + M$$

ระบบพิกัดเช่นนี้จะสะดวกต่อการใช้มากดังจะแสดงต่อไป

ถ้าต้องการ Translation เพียงอย่างเดียว Matrix ในสมการที่ 4.16 จะเป็น

ดังนี้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L & M & 1 \end{bmatrix} \quad (4.17)$$

สมมุติว่ามีรูปสามเหลี่ยมที่มีจุดยอดอยู่ที่ (15,30) (60,30) และ (30,60) ต้องการ Translate รูปสามเหลี่ยมนี้ไปเป็นระยะ (20,50) จะได้ผลดังนี้

$$P' = \begin{bmatrix} 15 & 30 & 1 \\ 60 & 30 & 1 \\ 30 & 60 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 20 & 50 & 1 \end{bmatrix} = \begin{bmatrix} 35 & 80 & 1 \\ 80 & 80 & 1 \\ 50 & 110 & 1 \end{bmatrix}$$

ถ้าเราต้องการ Translate  $P'$  ไปอีก (30,10) จะเป็น

$$P'' = P' \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 30 & 10 & 1 \end{bmatrix} = P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 20 + 30 & 50 + 10 & 1 \end{bmatrix}$$

นั่น คือ เราสรุปได้ว่าถ้าต้องการ Translate หลาย ๆ ครั้งแล้วจะได้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L_1 & M_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L_2 & M_2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L_1 + L_2 & M_1 + M_2 & 1 \end{bmatrix} \quad (4.18)$$

ในทำนองเดียวกันจากสมการ 4.5 ซึ่งเป็น Scaling จะเปลี่ยนขนาดของ Matrix เป็น  $3 \times 3$  ดังนี้

$$\begin{bmatrix} A & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.19)$$

และถ้ามีการ Scaling หลายครั้ง เราสามารถเอา Transformation Matrix ของแต่ละครั้งมาคูณกันก่อนแล้วนำผลสุดท้ายไปใช้ได้เหมือนกับการ Translate ดังนี้

$$\begin{bmatrix} A_1 & 0 & 0 \\ 0 & D_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_2 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} A_1 A_2 & 0 & 0 \\ 0 & D_1 D_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$

สำหรับการ Rotation ก็จะได้ รูปแบบของ Transformation Matrix ใหม่ ดังนี้

$$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.21)$$

ถ้ามีการ Rotation หลายครั้ง จะสรุปผลของ Transformation Matrix ได้ เหมือนกับทั้ง Translation และ Scaling ดังนี้

$$\begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

ถ้าเราพิจารณา Translation Matrices ดังต่อไปนี้

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -L & -M & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L & M & 1 \end{bmatrix}$$

จะเห็นว่า Matrices  $T_1$  และ  $T_2$  จะเคลื่อนย้ายจุดใดๆ ไปในทิศตรงกันข้ามกัน ถ้าเราเคลื่อนย้ายจุดตาม Matrices ทั้งสองนี้ จะได้ผลสุดท้ายว่าจุดนั้นจะกลับมาที่เดิม ซึ่งผลคูณของ  $T_1$  และ  $T_2$  ก็ให้ผลสอดคล้องกับที่กล่าวมาแล้ว คือได้ผลคูณเป็น Identity Transformation ดังนี้

$$T_1 T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -L & -M & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L & M & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.23)$$

เราเรียก  $T_2$  ว่าเป็น inverse ของ Matrix  $T_1$  ( $T_1^{-1}$ ) ซึ่ง Matrix ทุกตัวจะมีค่า Inverse ดังเช่นถ้ามีการหมุนตามเข็มนาฬิกาไป  $30^\circ$  จะได้

$$R(30^\circ) = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ & 0 \\ \sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

และเมื่อเราหมุนกลับมา  $30^\circ$  จะได้

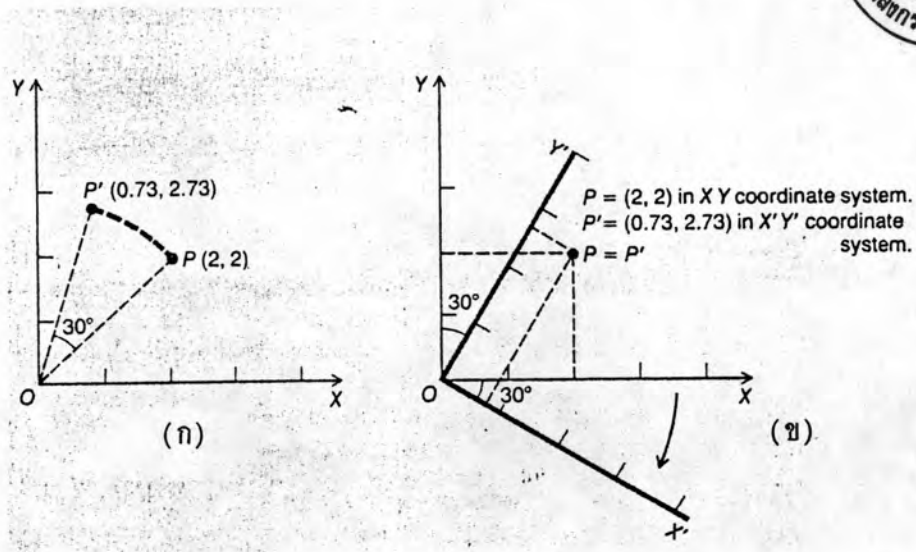
$$R(-30^\circ) = \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ซึ่ง  $R(-30^\circ)$  ก็คือ  $R(30^\circ)^{-1}$  นั่นเองเพราะทำให้จุดที่ถูกหมุนไปนั้นกลับมาอยู่ที่เดิม เราสามารถพิสูจน์ได้ในทางคณิตศาสตร์ว่า

$$R(30^\circ) \cdot R(-30^\circ) = I$$

$$\begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

จากที่กล่าวมาทั้งหมดแล้วนั้น เป็นการเคลื่อนย้ายจุด ภายในระบบพิกัดเดียวกัน แต่บางครั้งเราจำเป็นต้องเปลี่ยนระบบพิกัดด้วย ซึ่งในกรณีนี้ให้พิจารณา รูปที่ 4.6 (ก) แสดงการเคลื่อนย้ายจุด P ไปเป็น P' โดยหมุนทวนเข็มนาฬิกาในขณะที่ รูป 4.6 (ข) นั้น จุด P อยู่กับที่แต่หมุนแกน X และ Y ตามเข็มนาฬิกาจะเห็นได้ว่า จุด P' ในรูป 4.6 (ก) และจุด P' ในรูป 4.6 (ข) มีค่าพิกัดเมื่อเทียบกับแกน X' และ Y' เท่ากัน



รูปที่ 4.6 แสดงความสัมพันธ์ระหว่างการเคลื่อนย้ายจุดและการเคลื่อนย้ายระบบพิกัด

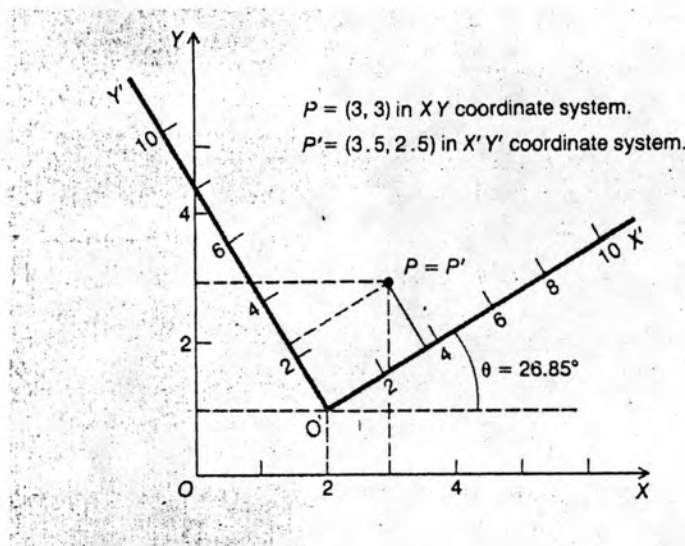
ดังนั้น เราสรุปได้ว่า Matrix ที่ใช้เคลื่อนย้ายระบบพิกัด ก็คือ Inverse ของ Matrix ที่ใช้เคลื่อนย้ายจุดนั้นในระบบพิกัดเดิม ตัวอย่างเช่น ใน รูปที่ 4.7 ซึ่งเป็นการเคลื่อนย้ายจุด  $P(3,3)$  ในระบบพิกัด  $X, Y$  ไปเป็นจุด  $P'(3.5, 2.5)$  ในระบบพิกัด  $X', Y'$  เราสามารถทำได้โดยมีขั้นตอน ดังนี้

ขั้นตอนที่ 1 เคลื่อนย้ายจุด Origin จาก  $O$  ไปเป็น  $O'$

ขั้นตอนที่ 2 ขยาย (Scaling) ค่าพิกัดของทั้งสองแกน

ขั้นตอนที่ 3 หมุนแกน  $X', Y'$  ทวนเข็มนาฬิกา ซึ่งก็คือหมุน จุด  $P$  ตามเข็มนาฬิกา

นั่นเอง



รูปที่ 4.7 แสดงการเคลื่อนย้ายจุดไปสู่ระบบพิกัดใหม่

Matrix  $T$  ที่ใช้เคลื่อนย้ายจุด  $P$  ไปเป็น  $P'$  จะเป็นดังนี้

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \cos \theta & -2 \sin \theta & 0 \\ 2 \sin \theta & 2 \cos \theta & 0 \\ -4 \cos \theta - 2 \sin \theta & 4 \sin \theta - 2 \cos \theta & 1 \end{bmatrix} \quad (4.24)$$

ดังนั้นสรุปได้ว่า

$$P' = (3, 3, 1) \cdot T = (2 \cos \theta + 4 \sin \theta, -2 \sin \theta + 4 \cos \theta, 1)$$

จากสมการข้างบนนี้ ถ้าเรารู้ค่ามุม  $\theta$  ก็จะหาค่าพิกัดของจุด  $P'$  ในระบบพิกัด  $x' y'$  ได้ แต่ด้วยวิธีนี้ ระบุว่าต้องการให้จุด  $P'$  มีค่าพิกัดเป็น  $(2.5, 2.5)$  ดังนั้นเราจะหาค่ามุม  $\theta$  ที่ต้องการได้ โดย

$$\text{ให้ } 2 \cos \theta + 4 \sin \theta = 3.5$$

$$\text{และ } -2 \sin \theta + 4 \cos \theta = 2.5$$

$$\text{แก้สมการจะได้ } \theta = 26.85^\circ$$

ในทางกลับกันกับที่กล่าวมาแล้ว ถ้าต้องการเปลี่ยนระบบพิกัดจาก  $x' y'$  มาเป็น  $x y$  จะทำได้โดยใช้ Inverse ของสมการ 4.24 และถ้า Matrix  $T$  ประกอบไปด้วย Matrices ย่อย  $n$  ตัว เราเขียนได้ ดังนี้

$$T = T_1 \cdot T_2 \cdot \dots \cdot T_{n-1} \cdot T_n$$

ดังนั้น Inverse ของ  $T$  จะเป็น

$$T^{-1} = T_n^{-1} \cdot T_{n-1}^{-1} \cdot \dots \cdot T_2^{-1} \cdot T_1^{-1} \quad (4.25)$$

$$T^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/2 \cos \theta & 1/2 \sin \theta & 0 \\ -1/2 \sin \theta & 1/2 \cos \theta & 0 \\ 2 & 1 & 1 \end{bmatrix}$$

เราสามารถหาจุด P ได้โดยคูณ P' ด้วย  $T^{-1}$

$$P = P' \cdot T^{-1}$$

$$= (1.75 \cos \theta - 1.25 \sin \theta + 2, 1.75 \sin \theta + 1.25 \cos \theta + 1, 1)$$

$$= (3, 3, 1)$$

วิธีการเคลื่อนย้ายระบบพิกัดนี้ มีประโยชน์มากในเมื่อมีวัตถุหลายชิ้นอยู่ในระบบพิกัด และเราต้องการ รวมวัตถุทั้งหมดมาอยู่ในระบบพิกัดเดียวกัน

4.1.4 ลำดับการเคลื่อนย้ายในระบบ 2 มิติ (Sequential 2 - D Transformations) จากที่กล่าวมาก่อนแล้วนั้น การหมุนของจุดใด ๆ เป็นการหมุนรอบจุด Origin ถ้าเราต้องการหมุนรอบจุดใด ๆ  $Q (q_1, q_2)$  เราสามารถทำได้ 3 ขั้นตอน ดังนี้

ขั้นตอนที่ 1 เคลื่อนย้ายจุด  $Q$  มายัง Origin

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -q_1 & -q_2 & 1 \end{bmatrix}$$

ขั้นตอนที่ 2 หมุนจุดที่ต้องการรอบจุด  $Q$

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



ขั้นตอนที่ 3 เคลื่อนย้าย จุด  $Q$  กลับไปยังตำแหน่งเดิม

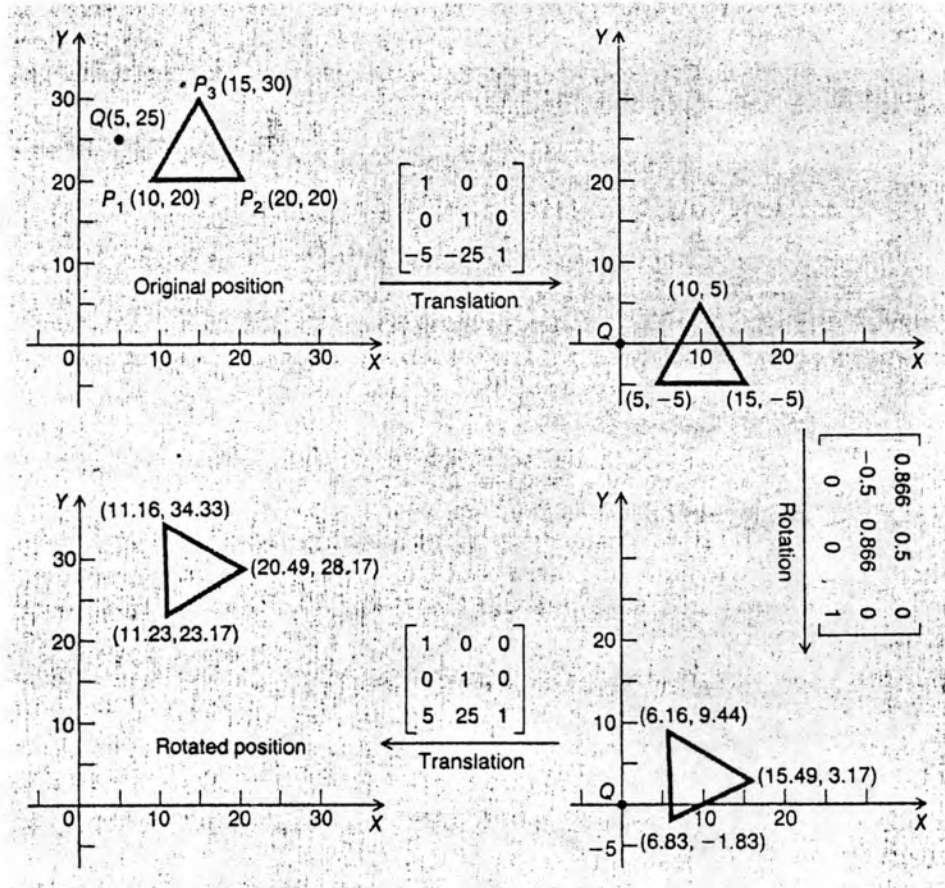
$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ q_1 & q_2 & 1 \end{bmatrix}$$

ดังนั้นจะสรุป Matrices ทั้งหมดที่ใช้ในการเคลื่อนย้ายเป็น

$$T = T_1 R(\theta) T_2$$

$$= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ q_1(1 - \cos \theta) + q_2 \sin \theta & q_2(1 - \cos \theta) - q_1 \sin \theta & 1 \end{bmatrix} \quad (4.26)$$

ตัวอย่างของการหมุนรอบจุดใดตา แสดงใน รูปที่ 4.8



รูปที่ 4.8 แสดงตัวอย่างการหมุนรอบจุดใดๆ ที่ไม่ใช่ Origin

ในกรณีที่ต้องการสะท้อน (Reflection) รอบแกนใดๆ เช่น รอบแกนที่มีสมการเป็น  $y = a + bx$  เราสามารถทำได้ดังขั้นตอนต่อไปนี้

ขั้นตอนที่ 1 เคลื่อนย้าย จุด Origin ไปที่  $(0, a)$

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -a & 1 \end{bmatrix}$$

ขั้นตอนที่ 2 หมุนแกนของ  $y = a + bx$  ตามเข็มนาฬิกา จนไปทับกับแกน  $x$  มุม  $\theta$  ที่หมุนไปนั้นหาได้โดยง่ายเพราะ ค่าความลาดชันของแกน  $y = a + bx$  คือ  $\tan \theta = b$  ดังนั้นจะได้  $\cos \theta = 1/r$  และ  $\sin \theta = b/r$  โดยที่  $r = (b^2 + 1)^{1/2}$

$$R(-\theta) = \begin{bmatrix} 1/r & -b/r & 0 \\ b/r & 1/r & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 3 สะท้อนวัตถุรอบแกน  $x$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 4 Inverse ของขั้นตอนที่ 2

$$R(\theta) = \begin{bmatrix} 1/r & b/r & 0 \\ -b/r & 1/r & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 5 Inverse ของขั้นตอนที่ 1

$$T_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & a & 1 \end{bmatrix}$$

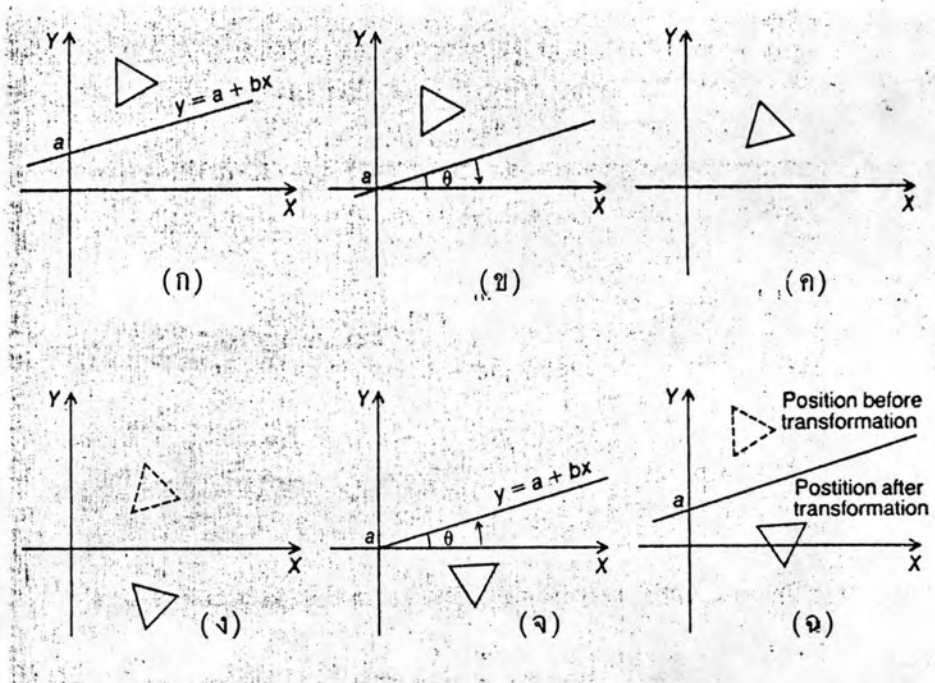
ขั้นตอนที่ 4 และ 5 นั้น เป็นการปรับค่าพิกัดของวัตถุให้กลับไปสู่ระบบพิกัดตอนเริ่มต้น

เราสามารถสรุป ขั้นตอนทั้ง 5 มาเป็น Matrix T ได้ดังนี้

$$T = T_1 R(-\theta) T_2 R(\theta) T_1^{-1}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -a & 1 \end{bmatrix} \begin{bmatrix} \cos 2\theta & \sin 2\theta & 0 \\ \sin 2\theta & -\cos 2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & a & 1 \end{bmatrix} \quad (4.29)$$

ขั้นตอนของการสะท้อนรอบแกนโคตา แสดงใน รูปที่ 4.9



รูปที่ 4.9 แสดงขั้นตอนการสะท้อนรอบแกนโคตา : (ก) เริ่มต้น (ข) ขั้นตอนที่ 1  
(ค) ขั้นตอนที่ 2 (ง) ขั้นตอนที่ 3 (จ) ขั้นตอนที่ 4 และ (ฉ) ขั้นตอนที่ 5

4.1.5 การสร้างขอบเขตของภาพในระบบ 2 มิติ (Viewport Planning of 2-D Graphics) จากที่เรากล่าวถึงการเปลี่ยนแปลงของจุดในลักษณะต่าง ๆ มาแล้วนั้น ค่าพิกัดที่ใช้ ถือว่าอยู่ในเนื้อที่ที่เป็นนามธรรม (Abstract Space) โดยค่าพิกัดนี้ เรียกว่า World Coordinate System ( หรือ Standard Coordinate System) ดังนั้นในหัวข้อนี้เราจะเปลี่ยนค่าพิกัดนี้ให้ไปอยู่ในระบบพิกัดของจอภาพ (Screen Display Coordinate System) ซึ่งมีขั้นตอนการเปลี่ยน 3 ขั้นตอน คือ การกำหนดขอบเขตของภาพ (Window Clipping), การกำหนดว่าจะนำภาพที่กำหนดขอบเขตแล้วไปแสดงที่ตำแหน่งใดบนจอภาพ (Window to Viewport Mapping) และขั้นตอนสุดท้ายคือการสร้างค่าพิกัดของจอภาพ (Generating Device Dependent Physical Screen Coordinates) ซึ่งรายละเอียดของแต่ละขั้นตอนจะแสดงดังต่อไปนี้

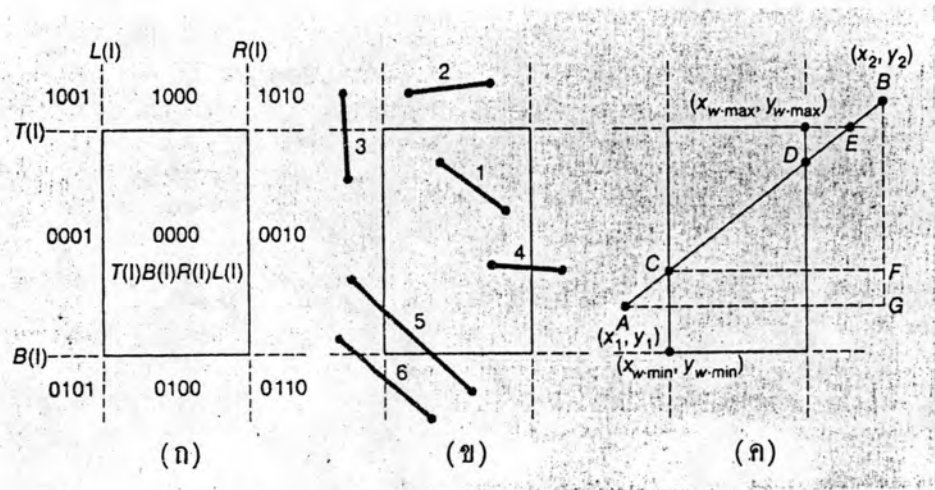
Window Clipping ในการนำภาพของวัตถุไปแสดงบนจอภาพนั้น เราอาจไม่สามารถแสดงภาพทั้งหมดได้ จึงต้องมีการกำหนดขอบเขตของภาพเสียก่อน ซึ่งแบ่งเป็นการกำหนดจุด (Clipping Points) และการกำหนดเส้น (Clipping Lines)

Clipping Points สามารถทำได้โดยง่าย โดยการเปรียบเทียบค่าพิกัดของจุดบนภาพกับค่าพิกัดที่กำหนดให้เป็นขอบเขต ถ้ากำหนดให้ขอบเขตของค่าพิกัดในแกน  $x$  เป็น  $x_{w.min}$  และ  $x_{w.max}$  ให้ขอบเขตของค่าพิกัดในแกน  $y$  เป็น  $y_{w.min}$  และ  $y_{w.max}$  ดังนั้นถ้าจุดที่มีค่าพิกัดเป็น  $(x,y)$  จะอยู่ในขอบเขตที่กำหนดต้องอยู่ในเงื่อนไข ดังนี้

$$x_{w.min} \leq x \leq x_{w.max}$$

$$y_{w.min} \leq y \leq y_{w.max}$$

สำหรับ Clipping Lines นั้น จะมีปัญหายุ่งยากขึ้น ทั้งนี้เพราะจะต้องตรวจสอบว่า เส้นนั้นอยู่ในขอบเขตที่กำหนดทั้งเส้นหรืออยู่ในขอบเขตเป็นบางส่วนหรืออยู่นอกขอบเขตทั้งเส้น ซึ่งจะใช้วิธีการดังต่อไปนี้



รูปที่ 4.10 แสดงการ Clipping Lines

พิจารณา รูปที่ 4.10 (ก) เส้นกำหนดขอบเขต 4 เส้น จะกำหนดให้เกิดเขตต่างๆ 9 เขต ตัวเลข 4 ตัวของแต่ละเขตจะเป็นตัวบอกสถานะของจุดที่อยู่ในเขตนั้นว่า อยู่สูงกว่า ต่ำกว่า ซ้าย หรือ ขวา ของขอบเขตที่กำหนดไว้ โดยกำหนดตัวแปร ไว้ดังนี้

$T(I) =$  จุด I อยู่สูงกว่าขอบเขตที่กำหนด

$B(I) =$  จุด I อยู่ต่ำกว่าขอบเขตที่กำหนด

$R(I) =$  จุด I อยู่ขวามือของขอบเขตที่กำหนด

$L(I) =$  จุด I อยู่ซ้ายมือของขอบเขตที่กำหนด

ถ้าตัวแปรแต่ละตัวเป็นจริงจะมีค่าเป็น 1 ถ้าไม่จริงจะมีค่าเป็น 0 ดังแสดงใน รูปที่ 4.10 (ก) การตรวจสอบว่าเส้นใดๆ จะอยู่ในขอบเขตที่กำหนดหรือไม่ แบ่งพิจารณาได้ 3 ประเภท ดังนี้

ประเภทที่ 1 ถ้าจุดปลายทั้งสองของเส้นอยู่ในขอบเขตที่กำหนด (ซึ่งหมายความว่า เส้นนั้นทั้งเส้นจะถูกนำไปแสดงบนจอภาพต่อไป) เช่น เส้น 1 ใน รูป 4.10 (ข) ซึ่งกรณีนี้จะเป็น

ไปตามสมการ 4.31

$$T(1) + B(1) + R(1) + L(1) + T(2) + B(2) + R(2) + L(2) = 0 \quad (4.31)$$

โดยตัวเลข 1 และ 2 ในวงเล็บหมายถึงจุดปลายทั้งสองของเส้น

ประเภทที่ 2 ถ้าจุดปลายทั้งสองของเส้นและทุกส่วนของเส้นอยู่นอกขอบเขตที่กำหนด เช่น เส้น 2 และ 3 ในรูปที่ 4.10 (ข) จะมีผลดังแสดงในสมการ 4.32

$$[T(1)T(2)] + [B(1)B(2)] + [R(1)R(2)] + [L(1)L(2)] \neq 0 \quad (4.32)$$

ประเภทที่ 3 กรณีนี้จะยุ่งยากกว่าที่กล่าวมาแล้ว ให้พิจารณาเส้นที่ 4 5 และ 6 ในรูปที่ 4.10 (ข) ซึ่งมีผลดังแสดงในสมการ 4.33

$$[T(1)T(2)] + [B(1)B(2)] + [R(1)R(2)] + [L(1)L(2)] = 0 \quad (4.33)$$

ในกรณีเช่นสมการ 4.33 เราต้องคำนวณหาจุดตัดระหว่างเส้นใด ๆ นั้นกับเส้นกำหนดขอบเขต เพื่อกำจัดส่วนที่อยู่นอกขอบเขตไป และนำส่วนที่อยู่ในขอบเขตไปแสดงบนจอภาพต่อไป ตัวอย่างของการหาจุดตัดให้พิจารณาเส้น AB ในรูป 4.10 (ค) ถ้าต้องการหาจุดตัด ซึ่งมีค่าพิกัดเป็น  $(x_{w.min}, y)$  จากการพิจารณารูปสามเหลี่ยมมุมฉาก ABG และ CBF เห็นว่าเป็นสามเหลี่ยมคล้ายกัน จะได้

$$\frac{BG}{AG} = \frac{BF}{CF}$$

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y_2 - y}{x_2 - x_{w.min}} \quad (4.34)$$

$$y = y_2 - \frac{y_2 - y_1}{x_2 - x_1} (x_2 - x_{w.min}) \quad (4.35)$$

เมื่อรู้ค่าพิกัดของจุด C แล้ว เส้น AC จะถูกกำจัดออกไปส่วนของเส้น CB ที่เหลือ เมื่อตรวจสอบแล้ว ยังเป็นประเภทที่ 3 ตามสมการ 4.33 อยู่จึงต้องคำนวณหาจุด ตัด D

แล้วกำจัดส่วน DB ออกไป ผลสุดท้ายคือเส้น CD ซึ่งจะมีผลตามสมการ 4.31 ดังนั้น เส้น CD จึงอยู่ในขอบเขตที่กำหนดทั้งเส้นและจะถูกส่งไปแสดงบนจอภาพต่อไป

Window to Viewport Mapping หลังจากเรากำหนดขอบเขตของภาพที่เราสนใจ (Window clipping) แล้ว ขั้นตอนต่อไปคือกำหนดตำแหน่งบนจอภาพที่จะนำภาพนั้นไปแสดง ซึ่งคือการเปลี่ยนจาก World Coordinate System ไปเป็น Viewport Coordinate System

พิจารณา รูปที่ 4.11 เราต้องการเปลี่ยนจุด  $(x_w, y_w)$  ไปเป็น  $(x_v, y_v)$  โดยใช้ Matrices ในการเคลื่อนย้ายจุดดังนี้

$$[x_v, y_v, 1] = [x_w, y_w, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_{w.min} & -y_{w.min} & 1 \end{bmatrix} \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_{v.min} & y_{v.min} & 1 \end{bmatrix}$$

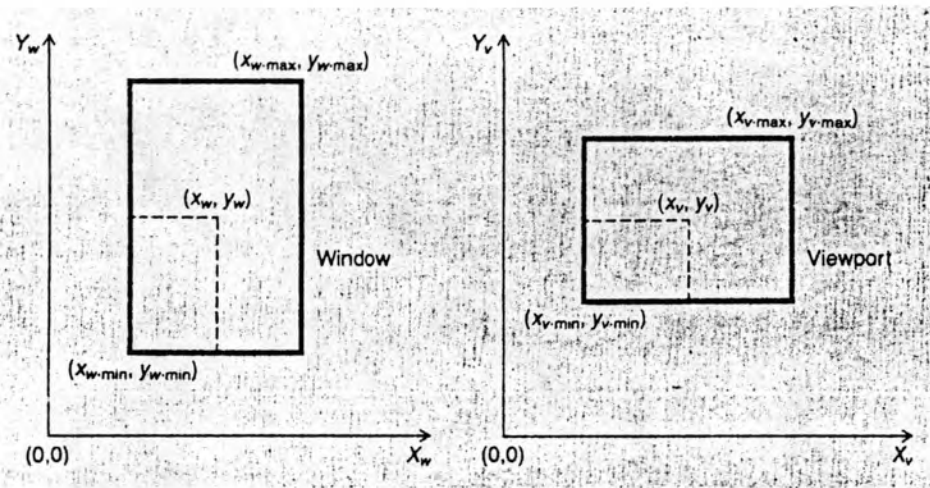
$$= [s_1(x_w - x_{w.min}) + x_{v.min}, s_2(y_w - y_{w.min}) + y_{v.min}, 1] \quad (4.36)$$

$$\text{โดย } s_1 = \frac{x_{v.max} - x_{v.min}}{x_{w.max} - x_{w.min}} \quad (4.37)$$

$$s_2 = \frac{y_{v.max} - y_{v.min}}{y_{w.max} - y_{w.min}} \quad (4.38)$$

ถ้า  $s_1$  ไม่เท่ากับ  $s_2$  เราจะได้รูปที่บิดเบี้ยวผิดส่วน





รูปที่ 4.11 Window to Viewport Mapping

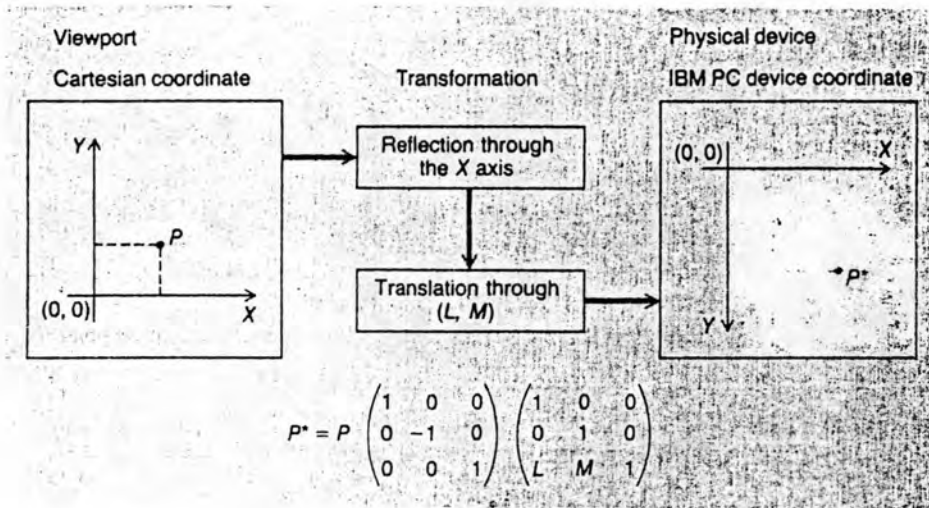
Generating Physical Device Coordinates เนื่องจากจอภาพคอมพิวเตอร์ ถูกผลิตมาโดยหลายบริษัท ดังนั้น หลังจากที่ได้ Viewport Coordinates ของวัตถุแล้ว จึงต้อง เปลี่ยนไปเป็น Screen Coordinates หรืออาจเรียกว่า Device Coordinates (ค่าพิกัดของจอภาพ) เช่น จอภาพของบริษัทหนึ่ง มีทิศทางของแกน  $y$  พุ่งลงล่าง และจุด Origin ก็อยู่มุมซ้ายบนของจอภาพ ดังนั้นต้องเคลื่อนย้ายระบบพิกัด โดยการสะท้อน (Reflection) ค่าพิกัดของวัตถุรอบแกน  $x$  แล้วเคลื่อนย้ายวัตถุไปตามแนวราบและแนวตั้ง ดังแสดงในสมการที่ 4.39 และ รูปที่ 4.12

$$[x_s, y_s, 1] = [x_v, y_v, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L & M & 1 \end{bmatrix}$$

$$(x_s, y_s, 1) = (x_v + L, M - y_v, 1) \quad (4.39)$$

$$\text{หรือ } x_s = S_1(x_w - x_{w.min}) + x_{v.min} + L \quad (4.40)$$

$$y_s = S_2(y_{w.min} - y_w) + y_{v.min} + M \quad (4.41)$$



รูปที่ 4.12 แสดงการเปลี่ยน Viewport Coordinate ไปเป็น Device Coordinate

สมการที่ 4.40 (ค่าของ  $x_s$ ) ที่ใ้ค้ันยั้งไม้ใช้สมการสุดท้ายที่ใ้ค้ต้องปรับเนื่องจากรายละเอียดของเครื่องมือแต่ละประเภท ค้ันนี้

$$x_s = SCF [S_1(x_w - x_{w.min}) + x_{v.min} + L]$$

SCF = Screen Scaling Adjustment Factor

ตัวอย่างเช่น เครื่องคอมพิวเตอร์ มีส่วนของการสร้างภาพ (Graphic Mode) ประกอบด้วยตะแกรง (Grid) ขนาด 310 × 200 หน่วย และมีจอภาพที่มีความยาวทางแนวราบ 240 มม. และความยาวทางแนวตั้ง 180 มม. ค้ันนี้เราสามารถหาใ้ค้ว่า 1 หน่วย จาก Graphic Mode จะเทียบเท่าใ้ค้กี่ มม. บนจอภาพทั้งใ้ค้แนวราบและแนวตั้ง ค้ันนี้

$$H = \frac{\text{ความยาวทางแนวราบของจอภาพ}}{\text{จำนวนหน่วยทางแนวราบของ Graphic Mode}} = \frac{240 \text{ มม.}}{320}$$

$$V = \frac{\text{ความยาวทางแนวตั้งของจอภาพ}}{\text{จำนวนหน่วยทางแนวราบของ Graphic Mode}} = \frac{180 \text{ มม.}}{200}$$

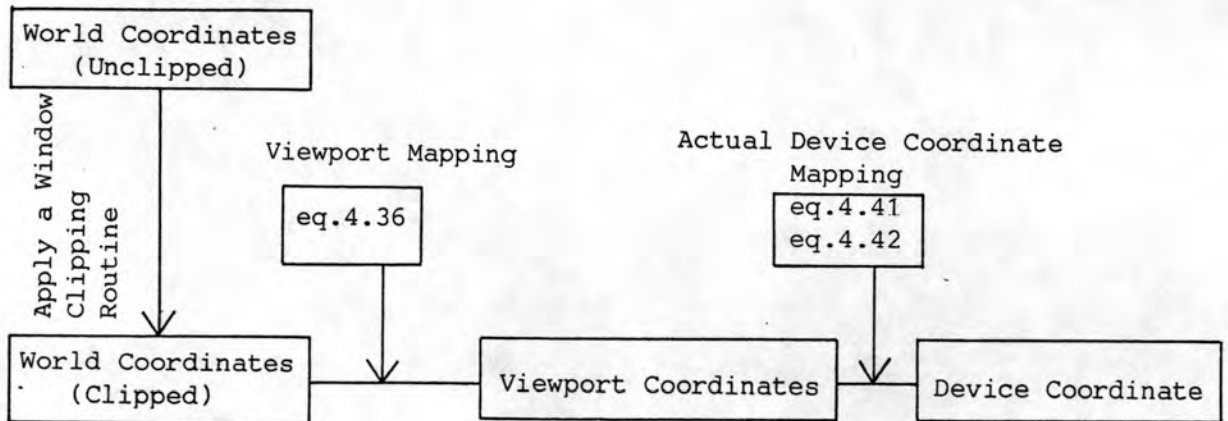
จะเห็นใ้ค้ว่า 1 หน่วย ทางแนวราบและแนวตั้งมีความยาวไม่เท่ากัน ถ้าเราไม่ปรับ

ค่า  $x_s$  ให้เทียบเท่ากับ  $y_s$  แล้ว ภาพที่ปรากฏบนจอภาพจะผิดส่วน เช่น วงกลมจะเบี้ยวไป ดังนั้น เราจะได้

$$SCF = \frac{V}{H} = 1.2$$

เราสรุปขั้นตอนการ นำภาพที่เก็บไว้ในหน่วยความจำของเครื่องคอมพิวเตอร์ซึ่งเป็น World Coordinate System มาแสดงบนจอภาพซึ่งเป็น Device Coordinate System ได้ ดังแสดงในแผนผังตามรูปที่ 4.13

Objects Defined in World Coordinates



รูปที่ 4.13 แสดงการเปลี่ยน World Coordinates ไปเป็น Device Coordinates

## 4.2 ทฤษฎีในการสร้างภาพ 3 มิติ

ในหัวข้อนี้ จะแสดงถึงวิธีการสร้างภาพ 3 มิติ บนจอภาพคอมพิวเตอร์ แต่เนื่องจาก จอภาพคอมพิวเตอร์ เป็น 2 มิติ ดังนั้นเราจึงต้องฉายภาพ (Perspective Projection) 3 มิติ ลงไปบนระนาบ 2 มิติ เพื่อสามารถแสดงภาพนั้นบนจอภาพได้ ซึ่งวิธีการที่ใช้จะกล่าวถึง รายละเอียดอีกต่อไป

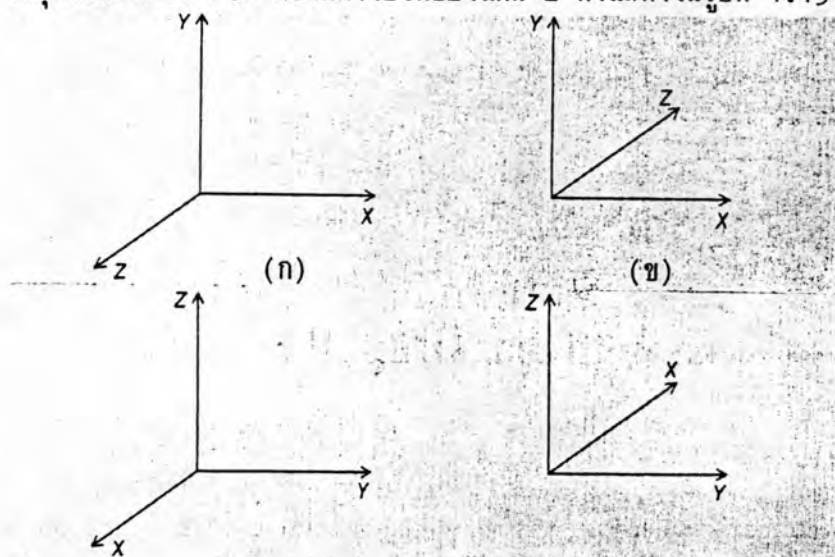
4.2.1 ระบบพิกัด (Coordinate Systems) ในระบบพิกัด 3 มิติ นี้จะเพิ่มแกน  $z$  ขึ้นมา ดังนั้น ในระบบนี้จะมีแกน  $x, y$  และ  $z$  ตั้งฉากซึ่งกันและกัน และระบบพิกัด 3 มิติ นี้จะแบ่งได้เป็น 2 ประเภท คือระบบขวามือ (Right - Handed System) และระบบซ้ายมือ

(Left - Handed System) ดังแสดงในรูปที่ 4.14 สำหรับ World Coordinate System จะใช้ระบบขวามือ

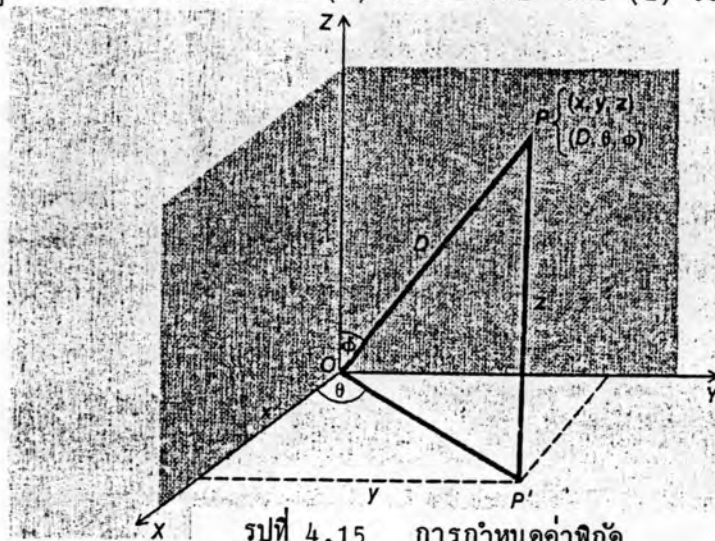
การกำหนดค่าพิกัดของจุดมีอยู่ 2 ระบบ คือ

ระบบพิกัดสี่เหลี่ยมมุมฉาก (Rectangular Coordinate System) จะบอกค่า  $(x,y,z)$  ซึ่งเป็นระยะตามแนวแกน  $x,y,z$  ดังแสดงในรูปที่ 4.15

และระบบพิกัดทรงกลม (Spherical Coordinate System) จะบอกค่า  $(D,\theta,\phi)$  โดย  $D$  เป็นระยะจากจุด  $P$  ที่จะบอกค่าพิกัดถึงจุด  $O$  หรือ Origin,  $\theta$  เป็นมุมที่วัดในทิศทวนเข็มนาฬิกา จากแกน  $x$  ถึง เส้น  $OP'$  ( $OP'$  คือ ภาพของเส้น  $OP$  บนระนาบ  $xy$ ) และ  $\phi$  คือมุมระหว่างเส้น  $OP$  กับทิศทางบวกของแกน  $z$  ดังแสดงในรูปที่ 4.15



รูปที่ 4.14 ระบบพิกัด : (ก) ระบบขวามือ และ (ข) ระบบซ้ายมือ



รูปที่ 4.15 การกำหนดค่าพิกัด

เราสามารถหาความสัมพันธ์ระหว่าง 2 ระบบพิกัด ได้ดังนี้

$$x = D \sin \varnothing \cos \Theta$$

$$y = D \sin \varnothing \sin \Theta$$

$$z = D \cos \varnothing \quad (4.42)$$

และ  $D = \sqrt{x^2 + y^2 + z^2}$

$$\Theta = \tan^{-1} (y/x)$$

$$\varnothing = \cos^{-1} (z/D)$$

#### 4.2.2 Matrix ที่ใช้ในการเปลี่ยนแปลงจุด (Transformation Matrix)

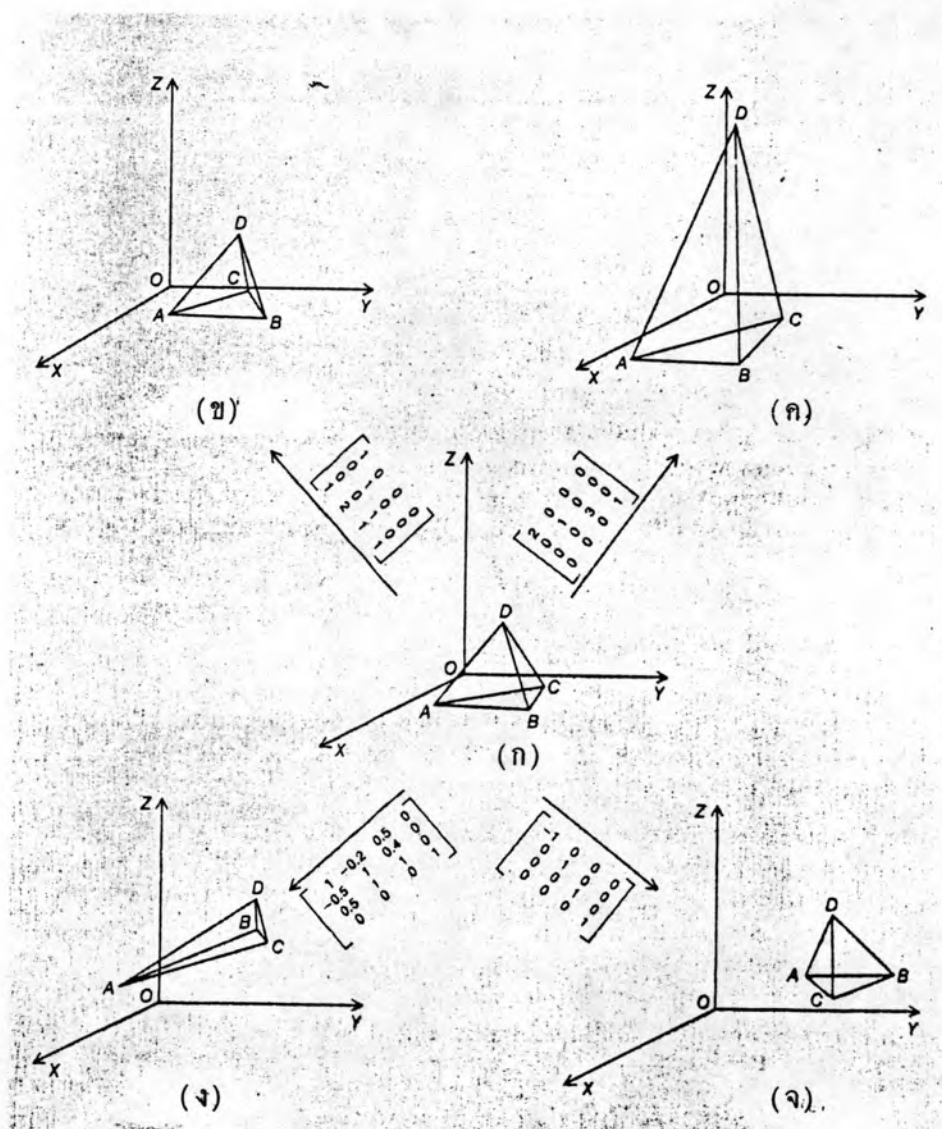
เช่นเดียวกับในระบบ 2 มิติ ที่กล่าวมาแล้ว ระบบ 3 มิติ ก็ใช้ Homogeneous Coordinate ดังนั้น Transformation Matrix จะเป็นดังนี้

$$\left[ \begin{array}{ccc|c} A & B & C & 0 \\ D & E & F & 0 \\ G & H & I & 0 \\ \hline L & M & N & 1 \end{array} \right] \quad (4.43)$$

เราสามารถแบ่ง Matrix ข้างบนนี้เป็น Matrices ย่อยๆ ได้ดังนี้

$$\left[ \begin{array}{c|c} \text{I} & \text{III} \\ (3 \times 3) & 4 \\ \hline \text{II} & \times \\ (1 \times 3) & 1 \end{array} \right] \quad (4.44)$$

Matrix ย่อย I ใช้ในการเปลี่ยนขนาด (Scaling) การเงื่อน (Shearing) การสะท้อน (Reflection) และการหมุน (Rotation) Matrix ย่อย II ใช้ในการเคลื่อนย้าย (Translation) Matrix ย่อย III เป็นตัวช่วยให้การเปลี่ยนแปลงจุด (Transformation) ทุกวิธีทำได้โดยการคูณ Matrix เพียงอย่างเดียว ตัวอย่างของ Transformation แสดงไว้ใน รูปที่ 4.16



รูปที่ 4.16 Transformations ของรูป 3 มิติ : (ก) เริ่มต้น (ข) Translation (ค) Scaling (ง) Shearing และ (จ) Reflection

รูปที่ 4.16 (ข) เป็นการ Translation จุด  $(x,y,z)$  ไปยังจุดใหม่  $(x',y',z')$  ไปเป็นระยะ  $(L,M,N)$  โดย  $L, M$  และ  $N$  คือระยะที่เคลื่อนจุดไปตามแกน  $x, y$  และ  $z$  ตามลำดับตาม รูปที่ 4.16 (ข) นี้  $L, M$  และ  $N$  มีค่า  $1, 2$  และ  $1$  หน่วย รูปแบบของ Transformation คือ

$$[x', y', z', 1] = [x, y, z, 1] \cdot \begin{bmatrix} 1 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \\ \hline L & M & N & | & 1 \end{bmatrix} \quad (4.45)$$

รูปที่ 4.16 (ค) เป็นการ Scaling ให้  $x'$  เป็น 2 เท่าของ  $x, y'$  เท่ากับ  $y$  และ  $z'$  เป็น 3 เท่าของ  $z$  นั่น คือ  $A, E$  และ  $I$  มีค่า  $2, 1$  และ  $3$  ตามลำดับ และมีรูปแบบของ Transformation ดังนี้

$$[x', y', z', 1] = [x, y, z, 1] \cdot \begin{bmatrix} A & 0 & 0 & | & 0 \\ 0 & E & 0 & | & 0 \\ 0 & 0 & I & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (4.46)$$

รูปที่ 4.16 (ง) เป็นการ Shearing ซึ่งได้แก่การที่เราให้ค่าของตัวเลขในแนวเส้นทะแยงมุมจากบนซ้ายมาล่างขวาของ Matrix ย่อย  $I$  มีค่าเป็น 1 นอกนั้นเป็นตัวเลขใดๆ จะมีรูปแบบของ Transformation ดังนี้

$$[x', y', z', 1] = [x, y, z, 1] \cdot \begin{bmatrix} 1 & B & C & | & 0 \\ D & 1 & F & | & 0 \\ G & H & 1 & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (4.47)$$

รูปที่ 4.16 (จ) เป็นการ Reflection ผ่านระนาบ  $Y Z$  นั้น คือค่าพิคักของแกน  $X$  จะถูกเปลี่ยนให้มีเครื่องหมายเป็นตรงกันข้ามกับของเดิมทุกจุดการ Reflection มีอยู่ 3 แบบ คือ

Reflection ผ่านระนาบ  $X Y$  จะมี Transformation Matrix เป็น

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.48)$$

Reflection ผ่านระนาบ  $Y Z$  จะมี Transformation Matrix เป็น

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.49)$$

Reflection ผ่านระนาบ  $X Z$  จะมี Transformation Matrix เป็น

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.50)$$

สำหรับ Rotation นั้นเราอาศัยสมการ 4.9 ซึ่งเป็นการหมุนในระบบ 2 มิติ และเป็นการหมุนรอบแกน  $Z$  ดังนั้นค่าพิคักตามแนวแกน  $Z$  จะมีค่าเท่าเดิม นั่นคือ ค่าในแถวที่ 3 และคอลัมน์ที่ 3 ของ Transformation Matrix เป็น 0 ยกเว้นในแนวเส้นทะแยงมุมจะเป็น 1 ดังนี้



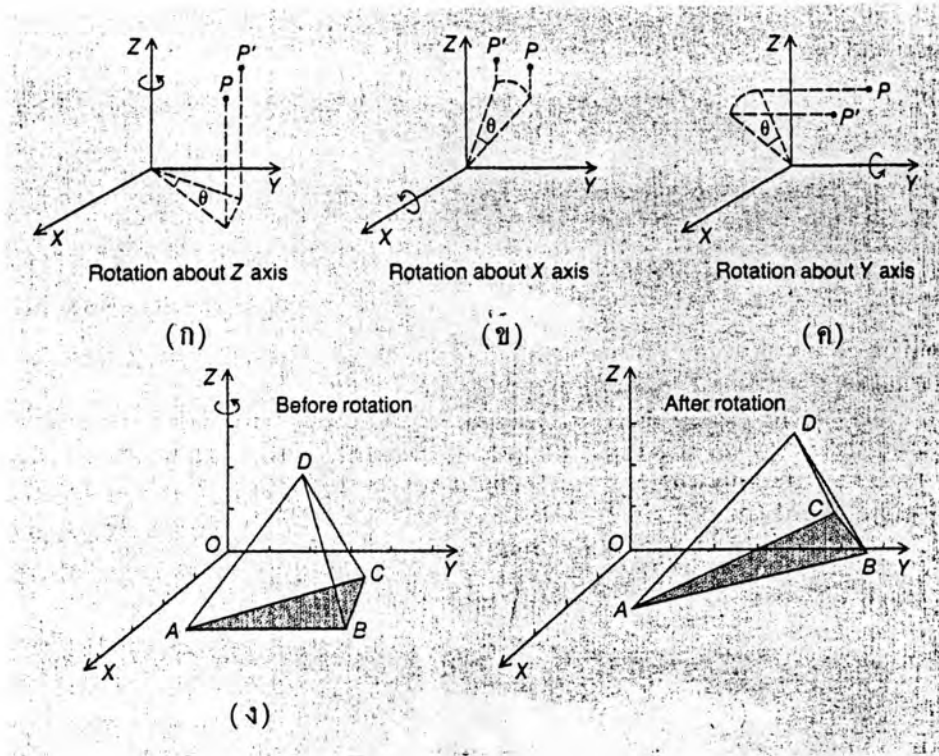
$$R(\theta)_z = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.51)$$

มุม  $\theta$  เป็นมุมที่วัดในทิศทางทวนเข็มนาฬิกา โดยมองจากจุดบนแกนบวกของ  $z$  เข้าไปยัง Origin ดังแสดงในรูปที่ 4.17 (ก)

สำหรับ Rotation รอบแกน  $X$  และ  $Y$  จะมี Transformation Matrices ดังสมการ 4.52 และ 4.53 และดังแสดงในรูปที่ 4.17 (ข) และ 4.17 (ค)

$$R(\theta)_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.52)$$

$$R(\theta)_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.53)$$



รูปที่ 4.17 Rotation ในระบบ 3 มิติ : (ก) Rotation รอบแกน Z (ข) Rotation รอบแกน X (ค) Rotation รอบแกน Y และ (ง) แสดง Rotation ของวัตถุรอบแกน Z

รูปที่ 4.17 (ง) เป็นตัวอย่าง Rotation ของวัตถุรูปปริมาตรรอบแกน z ไปเป็นมุม  $30^\circ$  ในทิศทางทวนเข็มนาฬิกา เราใช้สมการ 4.51 หาค่าพิกัดใหม่ของจุดบนปริมาตร ได้ดังนี้

$$\begin{matrix} A' \\ B' \\ C' \\ D' \end{matrix} = \begin{bmatrix} 3 & 1 & 0 & 1 \\ 3 & 5 & 0 & 1 \\ 1 & 4 & 0 & 1 \\ 2 & 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & 0.5 & 0 & 0 \\ -0.5 & 0.866 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2.1 & 2.37 & 0 & 1 \\ 0.1 & 5.83 & 0 & 1 \\ -1.13 & 3.96 & 0 & 1 \\ 0.232 & 3.60 & 3 & 1 \end{bmatrix}$$

ถ้าเราหมุนปริมาตรนี้ ในทิศตามเข็มนาฬิกาไป  $30^\circ$  รอบแกน z หรือให้  $\theta = -30^\circ$  จากสมการ 5.51 เราจะได้ Transformation Matrix ดังนี้

$$R(-30^\circ) = \begin{bmatrix} 0.866 & -0.5 & 0 & 0 \\ 0.5 & 0.866 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ซึ่งถ้าเราเอา  $R(30^\circ)$  คูณกับ  $R(-30^\circ)$  จะได้ผลเป็น I หรือ Identity Matrix แสดงว่า  $R(-30^\circ)$  เป็น  $R(30^\circ)^{-1}$  (Inverse ของ  $R(30^\circ)$ ) ดังนั้นเราสามารถสรุปได้ว่า

$$R(\theta)_z^{-1} = R(-\theta)_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.54)$$

$$R(\theta)_x^{-1} = R(-\theta)_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.55)$$

$$R(\theta)_y^{-1} = R(-\theta)_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.56)$$

สิ่งสำคัญที่ควรระวังในเรื่อง Rotation ก็คือ ลำดับของการคูณมีความสำคัญต่อผลสุดท้ายมาก เช่น การหมุนรอบแกน z เป็นมุม  $\theta_1$  แล้วจึงหมุนรอบแกน x เป็นมุม  $\theta_2$  จะได้ผลสุดท้ายไม่เหมือนกับการหมุนรอบแกน x เป็นมุม  $\theta_2$  ก่อน แล้วจึงหมุนรอบแกน z เป็นมุม  $\theta_1$  ซึ่งพิสูจน์ได้ดังนี้

ถ้าหมุนรอบแกน z เป็นมุม  $\theta_1$  ก่อนแล้วจึงหมุนรอบแกน x เป็นมุม  $\theta_2$  จะได้ Transformation Matrix ดังนี้

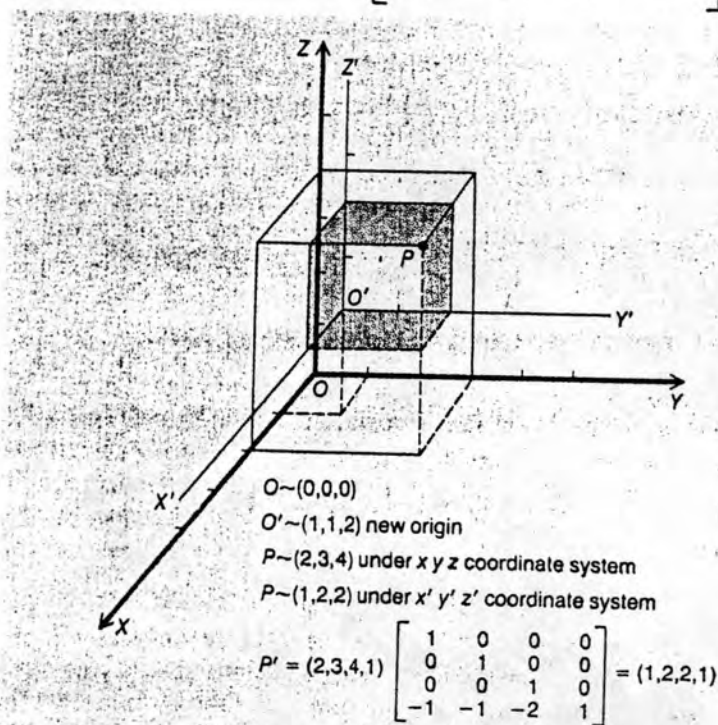
$$R(\theta_1)_z \cdot R(\theta_2)_x = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 \cos \theta_2 & \sin \theta_1 \sin \theta_2 & 0 \\ -\sin \theta_1 & \cos \theta_1 \cos \theta_2 & \cos \theta_1 \sin \theta_2 & 0 \\ 0 & -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

แต่ถ้าหมุนรอบแกน x ก่อน แล้วจึงหมุนรอบแกน z ผลจะเป็น

$$R(\theta_2)_x \cdot R(\theta_1)_z = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\cos \theta_2 \sin \theta_1 & \cos \theta_2 \cos \theta_1 & \sin \theta_2 & 0 \\ \sin \theta_2 \sin \theta_1 & -\sin \theta_2 \cos \theta_1 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เช่นเดียวกับในระบบ 2 มิติ ระบบ 3 มิติ ก็มีความจำเป็นต้องเปลี่ยนระบบพิกัดในบางครั้ง (Transforming a Coordinate System) ให้พิจารณารูปที่ 4.18 แสดงถึงจุด P มีค่าพิกัด (2,3,4) ในระบบพิกัด x y z และมีค่าพิกัด (1,1,2) ในระบบพิกัด x' y' z' การเคลื่อนย้ายระบบพิกัดจาก x y z ไปเป็น x' y' z' ทำโดยการ Translation จุด Origin o ไปเป็น o'

$$P' = P.T = (2,3,4,1) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & -1 & -2 & 1 \end{bmatrix} = (1,2,2,1)$$



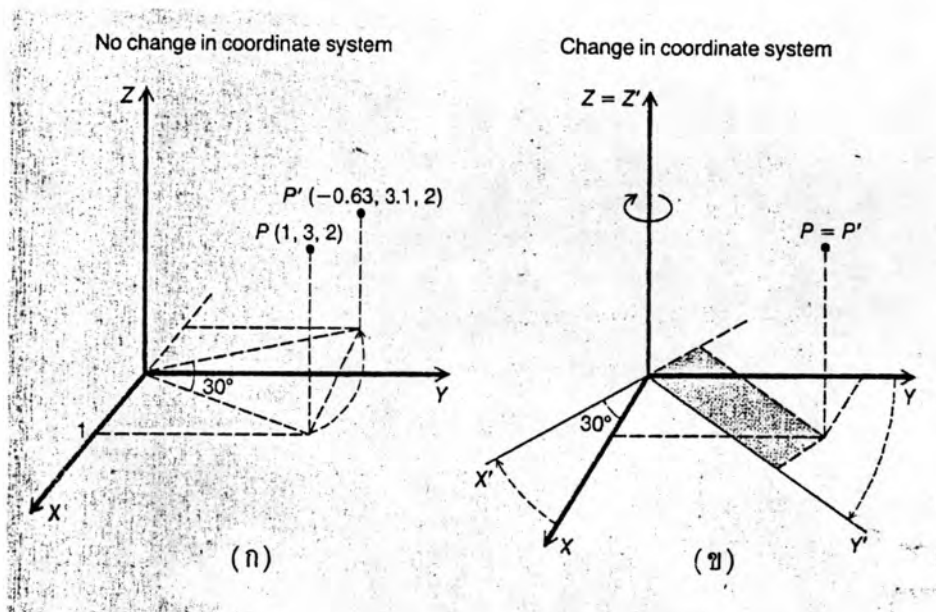
รูปที่ 4.18 การเปลี่ยนระบบพิกัด

ลองพิจารณา รูปที่ 4.19 (ก) ซึ่งเป็นการหมุน จุด P ไป 30° ทวนเข็มนาฬิกา รอบแกน z เทียบกับรูปที่ 4.19 (ข) ซึ่งเป็นการหมุนระบบพิกัดไป 30° ตามเข็มนาฬิกา เราจะเห็นว่า จุด P' เมื่อเทียบกับระบบพิกัด x y z ก็คือ จุด P เมื่อเทียบกับระบบพิกัด x' y' z' นั่นเอง ดังนั้น Transformation Matrix ที่ใช้ในการเปลี่ยนระบบพิกัดก็คือ Inverse ของ Transformation Matrix ที่ใช้ในการ Transformation จุดภายในระบบพิกัดเดิมซึ่ง

Transformation Matrix ที่ใช้ในรูปที่ 4.19 จะเป็นดังนี้

$$P' = P \cdot R(30^\circ)_Z = (1, 3, 2, 1) \begin{bmatrix} \cos 30^\circ & \sin 30^\circ & 0 & 0 \\ -\sin 30^\circ & \cos 30^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (-0.63, 3.1, 2, 1)$$

ผลข้างบนนี้ คือ ค่าพิกัดของจุด  $P'$  เมื่อเทียบกับระบบพิกัด  $X Y Z$  และจะมีค่าพิกัดเท่ากับของจุด  $P$  เมื่อเทียบกับระบบพิกัด  $X' Y' Z'$



รูปที่ 4.19 การเปลี่ยนระบบพิกัด : (ก) หมุนจุด  $P$  รอบแกน  $Z$  และ (ข) หมุนแกน  $X$  และ  $Y$  รอบแกน  $Z$

ปัญหาของการหมุนประการสำคัญ ก็คือการหมุนรอบแกนใดที่ไม่ใช่แกน  $X, Y$  และ  $Z$  ดังเช่นในรูปที่ 4.20 (ก) เราต้องการหมุนจุด  $P$  รอบแกน  $OJ$  โดยจุด  $Q (q_1, q_2, q_3)$  เป็นจุดที่ใช้เป็นศูนย์กลางการหมุนบนแกน  $OJ$

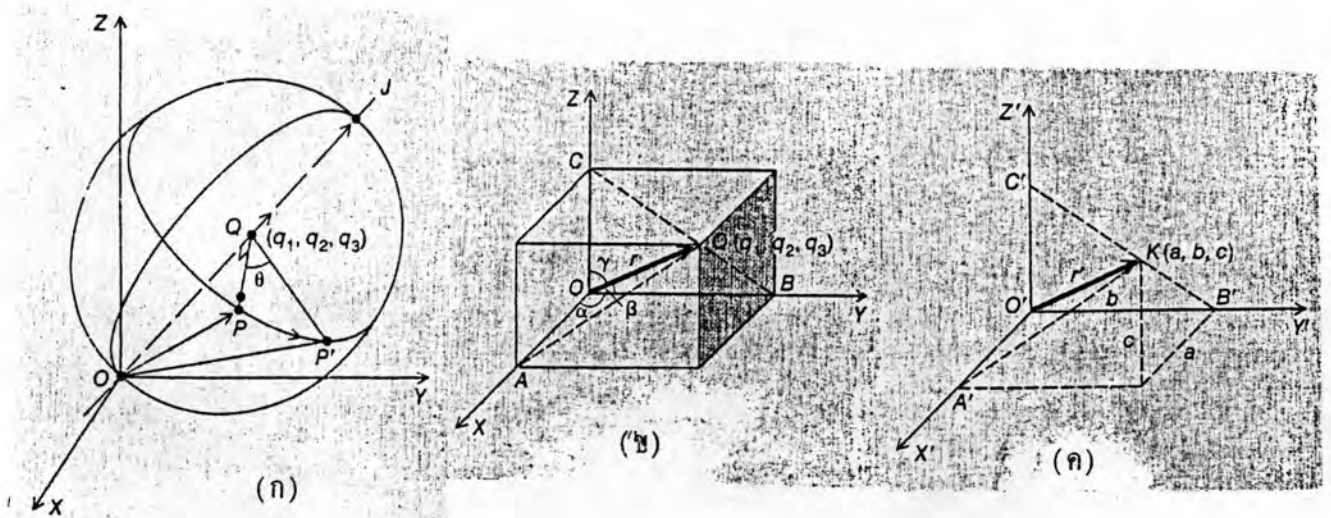
เพื่อช่วยในการคำนวณหา Transformation Matrix ง่ายขึ้นให้เรามาทำความเข้าใจ

กับ Direction Cosines ก่อน Direction Cosines คือการบอกทิศทางของแกนที่ใช้หมุน โดยบอกเป็นค่า Cosines ของมุมระหว่างแกน  $OQ$  กับแกน  $x$   $y$  และ  $z$  ( $\cos \alpha$   $\cos \beta$  และ  $\cos \gamma$  ให้อยู่รูปที่ 4.20 (ข)) เนื่องจากสามเหลี่ยม  $OAPQ$  เป็นสามเหลี่ยมมุมฉาก ซึ่งมีมุม  $A$  เป็นมุมฉาก และเนื่องจาก  $r = (q_1^2 + q_2^2 + q_3^2)^{\frac{1}{2}}$  ดังนั้นเราจะได้ Direction Cosines ดังนี้

$$\begin{aligned} \cos \alpha &= \frac{q_1}{r} = \frac{q_1}{(q_1^2 + q_2^2 + q_3^2)^{\frac{1}{2}}} \\ \cos \beta &= \frac{q_2}{r} = \frac{q_2}{(q_1^2 + q_2^2 + q_3^2)^{\frac{1}{2}}} \\ \cos \gamma &= \frac{q_3}{r} = \frac{q_3}{(q_1^2 + q_2^2 + q_3^2)^{\frac{1}{2}}} \end{aligned} \tag{4.57}$$

จากสมการข้างบนเราสามารถสรุปได้ว่า

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = \frac{q_1^2 + q_2^2 + q_3^2}{r^2} = 1 \tag{4.58}$$



รูปที่ 4.20 แสดงการหมุนรอบแกนใด ๆ ในระบบ 3 มิติ และ Direction Cosines (ก) การหมุนจุด  $P$  รอบแกน  $OQ$  (ข) Direction Cosines และ (ค) Direction Cosines

ในรูป 4.20 (ค) แสดงถึงเวกเตอร์ ตัวหนึ่งในระบบพิกัด  $x' y' z'$  ถ้าเราเลือกกำหนดจุด  $A'$ ,  $B'$  และ  $C'$  ให้ได้ค่า  $a = \cos \alpha$ ,  $b = \cos \beta$  และ  $c = \cos \gamma$  โดย  $\alpha, \beta$  และ  $\gamma$  เป็นค่าของมุม ตามรูป 4.20 (ข) ดังนั้นความยาว  $O'K$  คือ

$$r' = (a^2 + b^2 + c^2)^{\frac{1}{2}} = (\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma)^{\frac{1}{2}} = 1$$

สามเหลี่ยม  $O' A' K'$  เป็นสามเหลี่ยมมุมฉาก ซึ่งมีมุม  $A'$  เป็นมุมฉาก ดังนั้น  $\cos \alpha' = \cos \alpha / r' = \cos \alpha / 1 = \cos \alpha$  ในทำนองเดียวกันเราจะได้  $\cos \beta' = \cos \beta$  และ  $\cos \gamma' = \cos \gamma$  ดังนั้น  $O'Q$  และ  $O'K$  จะมี Direction Cosines เหมือนกัน นั่น คือ เส้น  $O'K$  ที่เราสร้างขึ้นใหม่มีทิศทางเดียวกับ เส้น  $O'Q$

เมื่อเรารู้ Direction Cosines ของแกน  $O'J$  เราสามารถหมุนจุด  $P$  เป็นมุม  $\theta$  รอบแกน  $O'J$  ได้ โดยทำเป็น 4 ขั้นตอน ดังต่อไปนี้

ขั้นตอนที่ 1 เคลื่อนย้ายวัตถุมาสู่ระบบพิกัดใหม่นั้น คือ  $Q$  จะเป็น Origin ใหม่ โดยใช้ Matrix ดังนี้

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\alpha_1 & -\alpha_2 & -\alpha_3 & 1 \end{bmatrix}$$

ขั้นตอนที่ 2 ปรับแกน  $O'J$  ให้มาทับกับแกน  $z'$  ซึ่งต้องหมุน 2 ครั้ง คือ หมุนรอบ  $x'$  และหมุนรอบแกน  $y'$  การที่เราใช้รูป 4.20 (ค) ในการคำนวณแทนรูป 4.20 (ข) เพราะ 2 รูปนี้ มี Direction Cosines เหมือนกันและค่าของ  $r'$  เป็น 1 หน่วย จะสะดวกในการคำนวณมากกว่า เมื่อเราฉายภาพ (Project) ของ  $O'K$  ลงไปบนระนาบ  $y' z'$  ได้ความยาวบนระนาบ  $y' z'$  เท่า  $v$  ดังรูปที่ 4.21 (ก) ถ้าเราหมุน  $O'K$  ไปเป็นมุม  $\phi_1$  ทวนเข็มนาฬิกา รอบแกน  $x'$  เส้น  $O'K$  จะไปอยู่บนระนาบ  $x' z'$  และเราจะได้

$$\cos \phi_1 = \frac{c}{v}$$

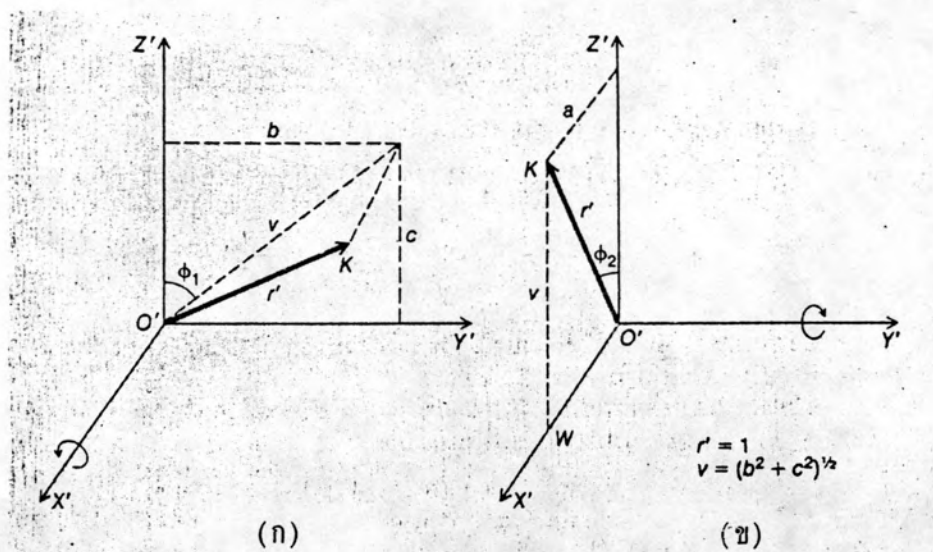


$$\sin \phi_1 = \frac{b}{v}$$

โดย  $v = (b^2 + c^2)^{1/2}$  และใช้สมการ 4.52 จะได้ Transformation Matrix เป็นดังนี้

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/v & b/v & 0 \\ 0 & -b/v & c/v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

หลังจากที่หมุนแล้วจะได้ผลดังแสดงในรูปที่ 4.21 (ข)



รูปที่ 4.21 การปรับจุด  $K$  ให้ไปอยู่บนแกน  $Z'$  : (ก) หมุนรอบแกน  $X'$  และ (ข) หมุนรอบแกน  $Y'$

จากรูปที่ 4.21 (ข) จะได้

$$O'K = [(KW)^2 + a^2]^{\frac{1}{2}}$$

$$r' = [(KW)^2 + a^2]^{\frac{1}{2}}$$

$$(a^2 + b^2 + c^2)^{\frac{1}{2}} = [(KW)^2 + a^2]^{\frac{1}{2}}$$

$$b^2 + c^2 = KW^2$$

$$KW^2 = v^2$$

$$KW = v$$

เนื่องจาก  $\phi_2$  เป็นมุมตามเข็มนาฬิกา ดังนั้น

$$\cos(-\phi_2) = \cos(\phi_2) = v/r' = v$$

$$\sin(-\phi_2) = -\sin(\phi_2) = -a/r' = -a$$

และใช้สมการ 4.53 จะได้ Transformation Matrix ดังนี้

$$R_2 = \begin{bmatrix} v & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ดังนั้น เส้น  $O'K$  จะถูกหมุนไปอยู่บนแกน  $Z'$

ขั้นตอนที่ 3 หมุนจุด  $P$  รอบแกน  $Z'$  ไปเป็นมุม  $\theta$  ตามต้องการ จะได้

Transformation Matrix ดังนี้

$$R_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 4 เปลี่ยนค่าพิกัดของวัตถุให้กลับไปยังระบบพิกัดในตอนเริ่มต้น โดยใช้ Inverse Matrices ของขั้นตอนที่ 2 และขั้นตอนที่ 1 เราจะได้ Transformation Matrix ที่สมบูรณ์ดังนี้

$$T = T_1 \cdot R_1 \cdot R_2 \cdot R_z(\theta) \cdot R_2^{-1} \cdot R_1^{-1} \cdot T_1^{-1}$$

$$T = \begin{bmatrix} A & B & C & 0 \\ D & E & F & 0 \\ G & H & I & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.59)$$

$$\text{โดย } A = a^2 + (1-a^2) \cos \theta$$

$$B = ab(1-\cos \theta) + c \sin \theta$$

$$C = ac(1-\cos \theta) - b \sin \theta$$

$$D = ab(1-\cos \theta) - c \sin \theta$$

$$E = b^2 + (1-b^2) \cos \theta$$

$$F = bc(1-\cos \theta) + a \sin \theta$$

$$G = ac(1-\cos \theta) + b \sin \theta$$

$$H = bc(1-\cos \theta) - a \sin \theta$$

$$I = c^2 + (1+c^2) \cos \theta$$

$$a = \cos \alpha = q_1/r$$

$$b = \cos \beta = q_2/r$$

$$c = \cos \gamma = q_3/r$$

$$r = (q_1^2 + q_2^2 + q_3^2)^{1/2}$$

เราสรุปการหมุนของจุด P ได้ว่า

$$P' = [x, y, z, 1].T \quad (4.60)$$

ตัวอย่างการคำนวณ เช่น จุด P (x,y,z) = (1,2,3), Q (q<sub>1</sub>,q<sub>2</sub>,q<sub>3</sub>) = (4,8,5) และ  $\theta = 30^\circ$  ทวนเข็มนาฬิกา จะได้

$$r = (4^2 + 8^2 + 5^2)^{1/2} = \sqrt{105}$$

$$\cos \alpha = \frac{4}{\sqrt{105}} = 0.39036 = a$$

$$\cos \beta = \frac{8}{\sqrt{105}} = 0.78072 = b$$

$$\cos \gamma = \frac{5}{\sqrt{105}} = 0.48795 = c$$

และ  $\cos 30^\circ = 0.866, \sin 30^\circ = 0.5$  แทนค่าลงในสมการ 4.59 ได้

$$T = \begin{bmatrix} 0.886 & 0.285 & -0.365 & 0 \\ -0.203 & 0.948 & 0.246 & 0 \\ 0.416 & -0.144 & 0.898 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ดังนั้น  $P' = (1, 2, 3, 1).T = (1.73, 1.75, 2.83, 1)$

ถ้าเราทดลองใช้สมการ 4.59 ในการหมุนวัตถุรอบจุด Origin เดิม คือ จุด O นั้น คือ ให้  $Q = (0, 0, 0)$  และ  $O' = O$  ดังนั้นจะได้

$$\cos \alpha = \cos 90^\circ = 0 = a$$

$$\cos \beta = \cos 90^\circ = 0 = b$$

$$\cos \gamma = \cos 0^\circ = 1 = c$$

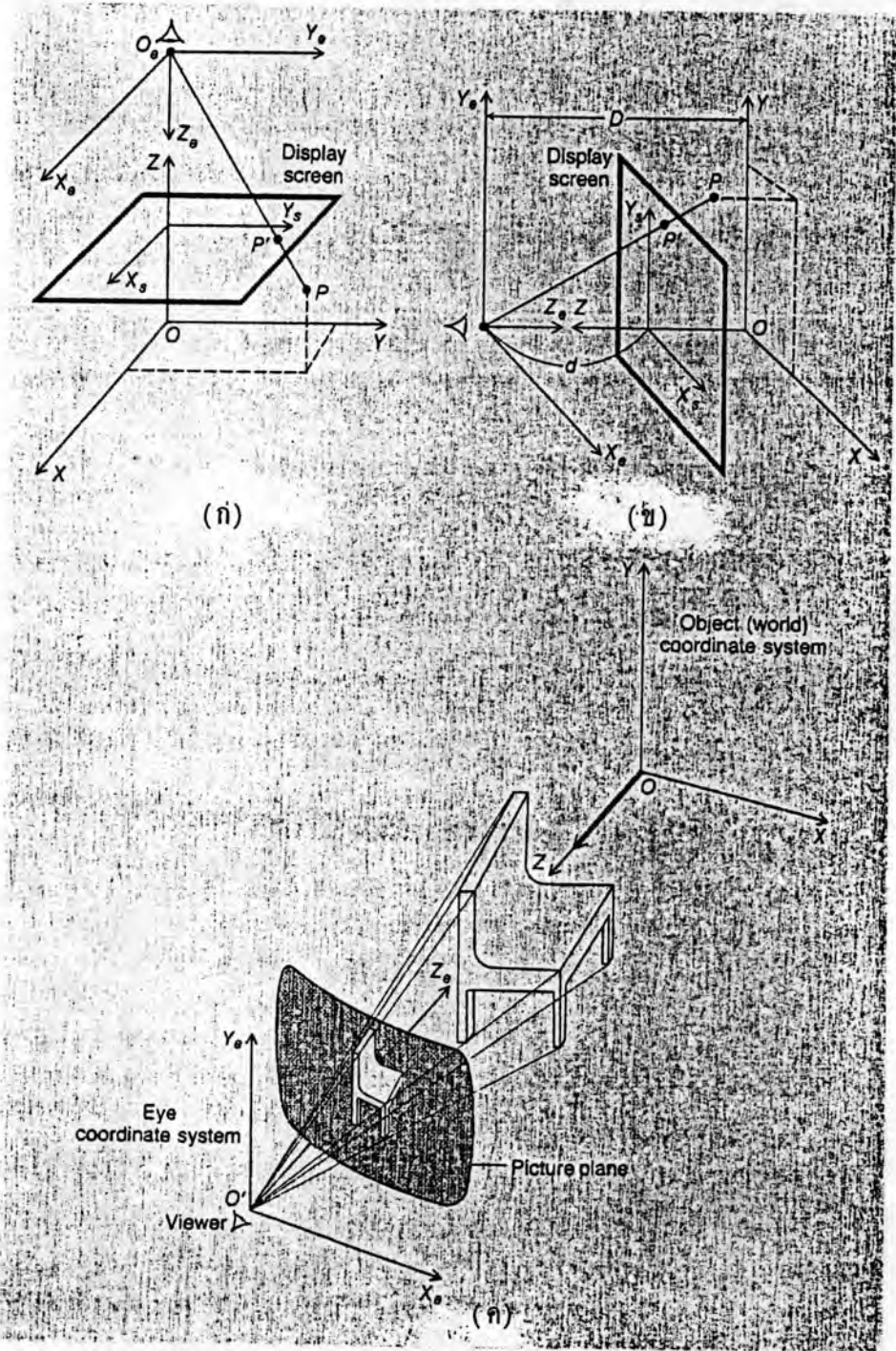
แทนค่าลงในสมการ 4.59 ได้

$$T = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ซึ่งเหมือนกับสมการ 4.51

4.2.3 การสร้างภาพ 3 มิติ (Viewing in Three Dimensions) ในหัวข้อนี้ จะอธิบายถึงวิธีการนำภาพ 3 มิติ ไปแสดงบนจอภาพคอมพิวเตอร์ โดยการฉายภาพ 3 มิติ (Perspective Projection) ลงไปบนระนาบ 2 มิติ ซึ่ง Perspective Projection มีอยู่ 2 แบบ คือ Central Projection และ Arbitrary Projection ดังจะอธิบาย รายละเอียดของแต่ละแบบต่อไป การฉายภาพ 3 มิติ ไปบนระนาบ 2 มิตินี้ ต้องเปลี่ยนระบบพิกัด เดิม (World Coordinate System) ไปเป็นระบบพิกัดของสายตา (Eye Coordinate System) และใช้ Viewing Transformation หาค่าแทนของวัตถุบนจอภาพ

4.2.3.1 Central Projection Projection แบบนี้ คือ จุดศูนย์กลางของการฉายภาพ (Center of Projection หรือ Viewpoint) จะอยู่บนแกนหนึ่งของ World Coordinate System และอยู่บนแกนหนึ่งของ Eye Coordinate System ด้วย ซึ่งเราให้เป็นแกน z ดังนั้น แนวสายตาจะทับกับแกน z และจอภาพจะขนานกับระนาบ x y เราสร้างภาพบนจอภาพได้โดยฉาย (Project) จุดของวัตถุลงบนจอภาพ ดังแสดงในรูปที่ 4.22



รูปที่ 4.22 Central Projection ของวัตถุบนระนาบ 2 มิติ : (ก) Central Projection  
 (ข) Central Projection และ (ค) การฉายภาพของวัตถุ

เราให้ Eye Coordinate System เป็นระบบพิกัดซ้ายมือ (Left-Handed Coordinate System) ค่าพิกัดของ Eye Coordinate System จะระบุเป็น  $(x_e, y_e, z_e)$  โดยมีทิศทางในทางบวกของแกน  $x_e$ ,  $y_e$  และ  $z_e$  ดังแสดงในรูปที่ 4.22 แต่ World Coordinate System ยังคงเป็นระบบพิกัดขวามือ (Right-Handed Coordinate System) เหมือนเดิม จากรูป 4.22 แกน  $z$  เป็นแกนของแนวสายตา ดังนั้นระนาบ  $x_e$ ,  $y_e$  จะขนานกับระนาบ  $x$ ,  $y$  เป็นผลให้มุม  $\theta$  และ  $\phi$  เป็น 0 นั่นคือ ค่าพิกัดของ Viewpoint ในระบบพิกัดทรงกลมเป็น  $(D, 0, 0)$

Viewing Transformation ที่ใช้ในการเปลี่ยน World Coordinate System ของวัตถุไปเป็น Eye Coordinate System กระทำได้ 4 ขั้นตอนดังนี้

ขั้นตอนที่ 1 เปลี่ยน Origin ไปอยู่ที่ Viewpoint

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -D & 1 \end{bmatrix}$$

ขั้นตอนที่ 2 เนื่องจาก Eye Coordinate System เป็นระบบพิกัดซ้ายมือ จึงต้องกลับทิศทางของแกน  $z_e$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 3 รวม Transformation Matrices ทั้ง 2 ที่หามาแล้วเข้าด้วยกัน

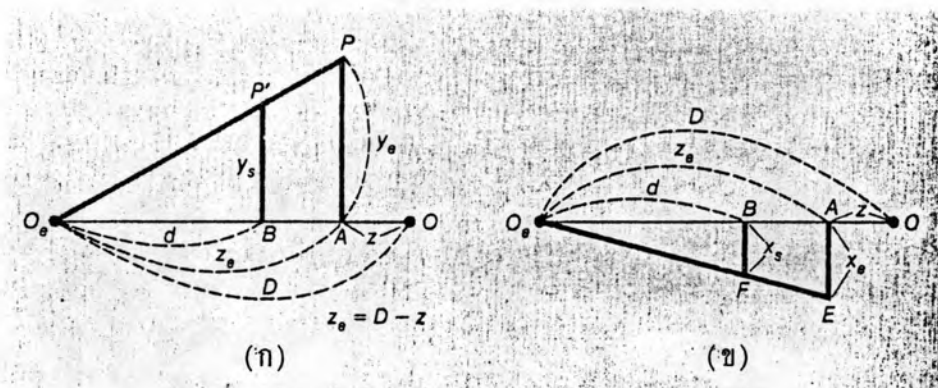
$$T = T_1 \cdot T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & D & 1 \end{bmatrix}$$

ขั้นตอนที่ 4 ทาค่าพิกัดใน Eye Coordinate System

$$(x_e, y_e, z_e, 1) = (x, y, z, 1) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & D & 1 \end{bmatrix} \quad (4.61)$$

$$= (x, y, D-z, 1)$$

หลังจากได้ค่าพิกัด  $(x_e, y_e, z_e)$  ของ Eye Coordinate System แล้ว จึงเปลี่ยนไปเป็นค่าพิกัด  $(x_s, y_s)$  บนจอภาพ ซึ่งกระทำโดยพิจารณาระนาบ  $y_e z_e$  ในรูป 4.23 (ก) และระนาบ  $x_e z_e$  ในรูป 4.23 (ข)



รูปที่ 4.23 การเปลี่ยน Eye Coordinate System ไป Screen Coordinate System  
(ก) ระนาบ  $y_e z_e$  และ (ข) ระนาบ  $x_e z_e$



พิจารณารูป 4.23 (ก) จะเห็นว่าสามเหลี่ยม  $A O_e P$  คล้ายกันกับสามเหลี่ยม  $B O_e P'$  เราจะได้

$$\frac{y_s}{d} = \frac{y_e}{z_e} \longrightarrow y_s = d \frac{y_e}{z_e} = d \frac{y}{D-z} \quad (4.62)$$

และจากรูป 4.23 (ข) สามเหลี่ยม  $A O_e E$  คล้ายกันกับสามเหลี่ยม  $B O_e F$  จะได้

$$\frac{x_s}{d} = \frac{x_e}{z_e} \longrightarrow x_s = d \frac{x_e}{z_e} = d \frac{x}{D-z} \quad (4.63)$$

มีข้อสังเกตสำหรับสมการ 4.62 และ 4.63 ดังนี้

- เพราะ  $y_s$  และ  $x_s$  เป็นสัดส่วนโดยตรงกับ  $d$  ดังนั้นถ้าเคลื่อนจอภาพให้เข้าใกล้ Viewpoint จะทำให้ภาพของวัตถุบนจอภาพเล็กลง

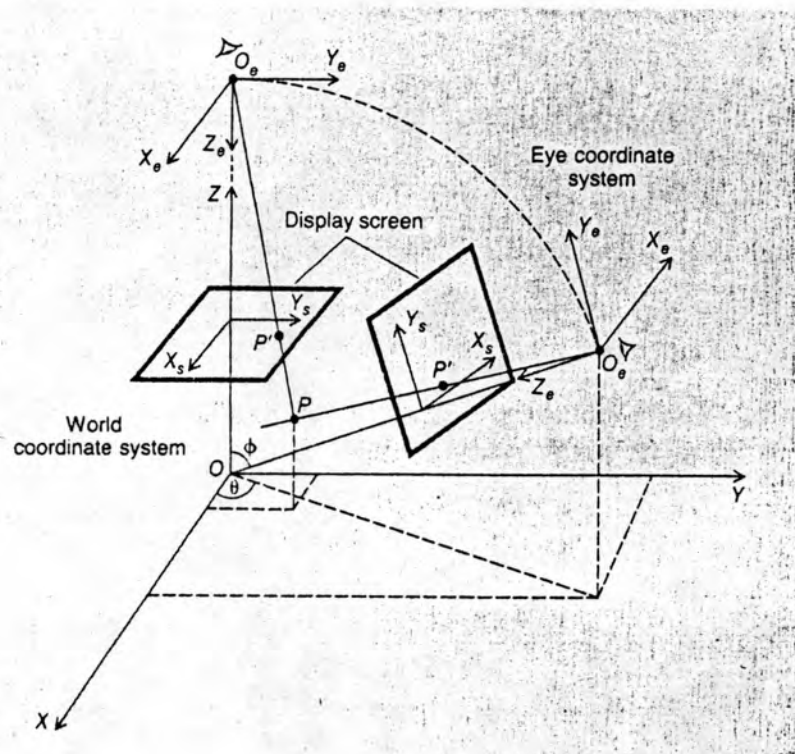
- ถ้าให้  $d$  คงที่แล้วเพิ่มระยะ  $D$  คือ เลื่อน Viewpoint ให้ห่างออกจากวัตถุจะทำให้ภาพของวัตถุบนจอภาพเล็กลง

- Central Projection ไม่ได้ใช้ Viewing Transformation ที่ยุ่งยากเลย เพราะ Viewpoint อยู่บนแกนของ world Coordinate System

- ถ้าเราเคลื่อนจอภาพให้ไปทับกับระนาบ  $x y$  ทำให้  $D = d$  และเลื่อน Viewpoint ห่างออกไปจน  $D \longrightarrow \infty$  เราจะได้  $x_s = x$  และ  $y_s = y$  ลักษณะเช่นนี้เรียกว่า Orthographic Projection ซึ่งทำให้เส้นขนานของวัตถุ 3 มิติ จะยังคงขนานกันในภาพบนจอ

4.2.3.2 Arbitrary Projection กรณีนี้ เราให้จอภาพอยู่ห่างจาก Viewpoint เป็นระยะ  $d$  เหมือนเดิม และจอภาพตั้งฉากกับเส้นที่เชื่อมตอระหว่าง Viewpoint กับ Origin Viewpoint จะเป็นจุดใดๆ ไม่จำเป็นต้องอยู่บนแกนของ world Coordinate System แนวของสายตา (แกน  $z_e$ ) จะลากจาก viewpoint ไปยัง Origin ของ world Coordinate System โดยแกน  $x_e$  จะมีทิศบวกไปทางขวาและแกน  $y_e$  มีทิศบวกไปข้างบน รูปที่ 4.24 แสดงความแตกต่างระหว่าง Central Projection และ Arbitrary Projection ในการหา Screen Coordinates  $(x_s, y_s)$  นั้นต้องเปลี่ยน World Coordinates  $(x, y, z)$

ให้เป็น Eye Coordinates ( $x_e, y_e, z_e$ ) ก่อน จึงจะเปลี่ยนมาเป็น Screen Coordinates ได้



รูปที่ 4.24 เปรียบเทียบ Central Projection และ Arbitrary Projection

Viewing Transformation ที่ใช้เปลี่ยนค่า World Coordinates ไปเป็น Eye Coordinates ได้จากการ Transform ระบบพิกัดโดยให้จุด P ของวัตถุอยู่ที่เดิม ทั้งนี้ต้องระลึกไว้เสมอว่า Transformation Matrix ในการเปลี่ยนระบบพิกัดจะเป็น Inverse ของการเปลี่ยนตำแหน่งของจุด ดังเคยกล่าวมาก่อนแล้ว เมื่อพิจารณา Viewpoint  $O_e$  ในรูปที่ 4.24 จะเห็นว่าค่าพิกัดตามระบบทรงกลมเป็น  $(D, \theta, \phi)$  จากสมการ 4.42 เราได้

$$x = D \sin \theta \cos \phi$$

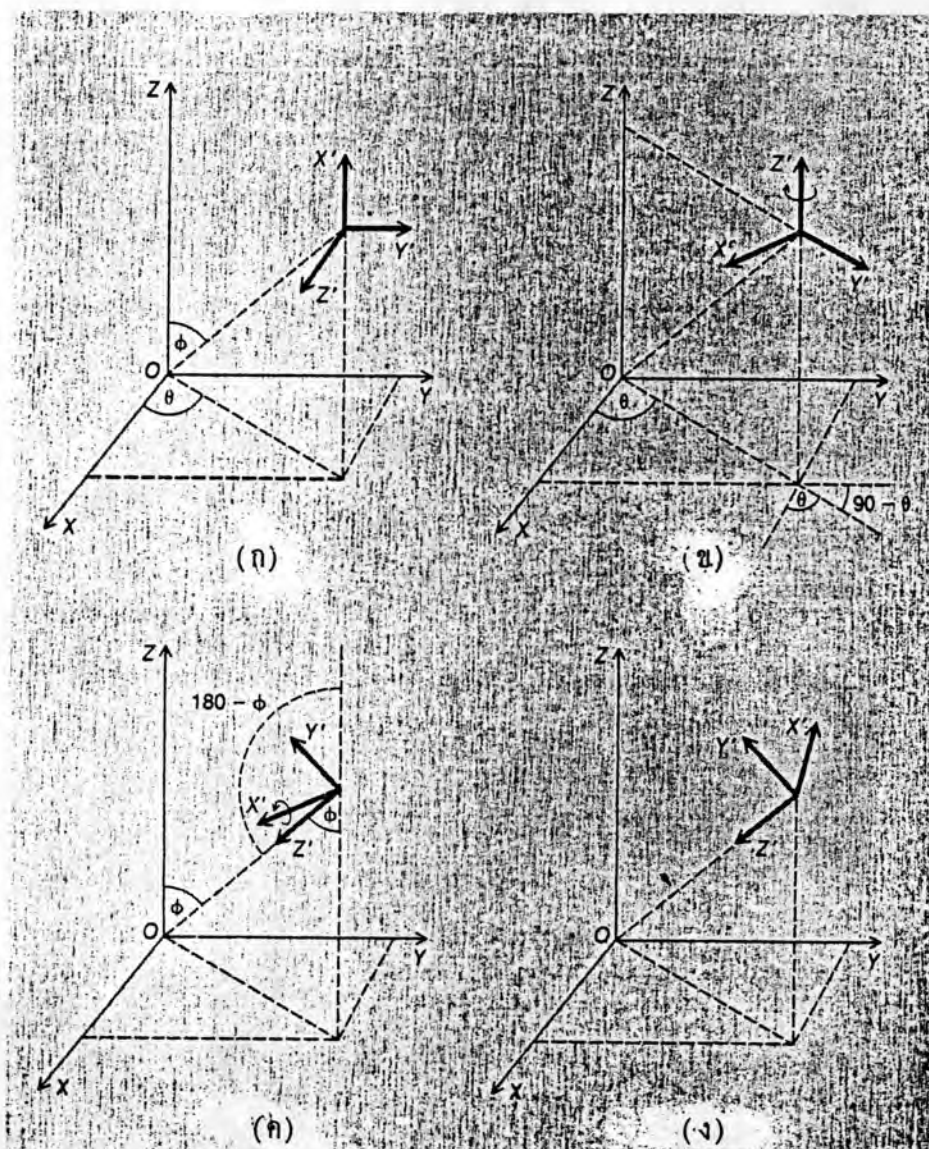
$$y = D \sin \theta \sin \phi$$

$$z = D \cos \theta$$

เราสร้าง Viewing Transformation เป็นขั้นตอน ดังนี้

ขั้นตอนที่ 1 ย้ายจุด Origin ไปอยู่ที่  $O_e$  ดังรูป 4.25 (ก)

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -D \cdot \sin \phi \cos \theta & -D \cdot \sin \phi \sin \theta & -D \cdot \cos \phi & 1 \end{bmatrix}$$



รูปที่ 4.25 ขั้นตอนในการเปลี่ยน World Coordinates ไปเป็น Eye Coordinates

ขั้นตอนที่ 2 หมุนระบบพิกัดตามเข็มนาฬิการอบแกน  $z'$  ไปเป็นมุม  $90^\circ - \theta$  ผลของการหมุนจะทำให้ แกน  $y'$  มีทิศทางไปตัดกับแกน  $z$  ดังรูปที่ 4.25 (ข) Transformation Matrix จะเป็น Inverse ของการหมุนจุดใดๆ รอบแกน  $z'$  ในทิศทางเข็มนาฬิกาเป็นมุม  $90^\circ - \theta$  จะได้  $\sin(90^\circ - \theta) = \cos \theta$  และ  $\cos(90^\circ - \theta) = \sin \theta$

$$T_2 = \begin{bmatrix} \sin \theta & \cos \theta & 0 & 0 \\ -\cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 3 หมุนระบบพิกัด รอบแกน  $x'$  ในทิศทางทวนเข็มนาฬิกาไปเป็นมุม  $180^\circ - \phi$  ผลที่ได้คือแกน  $z'$  จะมีทิศทางพุ่งเข้าสู่อุบัติของ World Coordinate System ดังรูปที่ 4.25 (ค) Transformation Matrix จะเป็น Inverse ของการหมุนจุดใดๆ รอบแกน  $x'$  ในทิศทางตามเข็มนาฬิกา เป็นมุม  $180^\circ - \phi$  จะได้  $\sin(180^\circ - \phi) = \sin \phi$  และ  $\cos(180^\circ - \phi) = -\cos \phi$

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & -\cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ขั้นตอนที่ 4 เปลี่ยนระบบพิกัดเป็นระบบพิกัดซ้ายมือ ซึ่งเราเห็นจากรูป 4.25 (ค) ว่าแกน  $x'$  มีทิศทางไม่ถูกต้อง คือ เป็นระบบพิกัดขวามืออยู่

$$T_4 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เราสรุปขั้นตอนทั้ง 4 เป็น Viewing Transformation ดังนี้

$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4$$

$$T = \begin{bmatrix} -\sin \theta & -\cos \theta \cos \phi & -\cos \theta \sin \phi & 0 \\ \cos \theta & -\sin \theta \cos \phi & -\sin \theta \sin \phi & 0 \\ 0 & \sin \phi & -\cos \phi & 0 \\ 0 & 0 & D & 1 \end{bmatrix} \quad (4.64)$$

ดังนั้นเราหา Eye Coordinates ได้จาก World Coordinates ดังนี้

$$(x_e, y_e, z_e, 1) = (x, y, z, 1) \cdot T$$

$$\text{หรือ } x_e = -x \sin \theta + y \cos \theta$$

$$y_e = -x \cos \theta \cos \phi - y \sin \theta \cos \phi + z \sin \phi \quad (4.65)$$

$$z_e = -x \cos \theta \sin \phi - y \sin \theta \sin \phi - z \cos \phi + D$$

ทำนองเดียวกับสมการ 4.61 และ 4.62 จะได้

$$x_s = d \left( \frac{x_e}{z_e} \right), \quad y_s = d \left( \frac{y_e}{z_e} \right) \quad (4.66)$$

และเราสามารถใส่สมการ 4.63 และ 4.64 ใน Central Projection โดย  $\theta = 270^\circ$  และ  $\phi = 0^\circ$  ดังนั้นเราจะได้สมการ 4.63 เหมือนสมการ 4.61 ดังนี้

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & D & 1 \end{bmatrix}$$

เราสามารถเปลี่ยนค่าของ  $x_s$  และ  $y_s$  ให้เป็นตัวเลขที่ไม่มีหน่วยได้โดยหารค่าทั้งสองด้วย ความยาวจากศูนย์กลางของจอภาพ ถึงขอบ (ในกรณีจุด Origin อยู่ที่ศูนย์กลางของจอภาพ) จะได้ Screen Coordinate ใหม่ ดังนี้

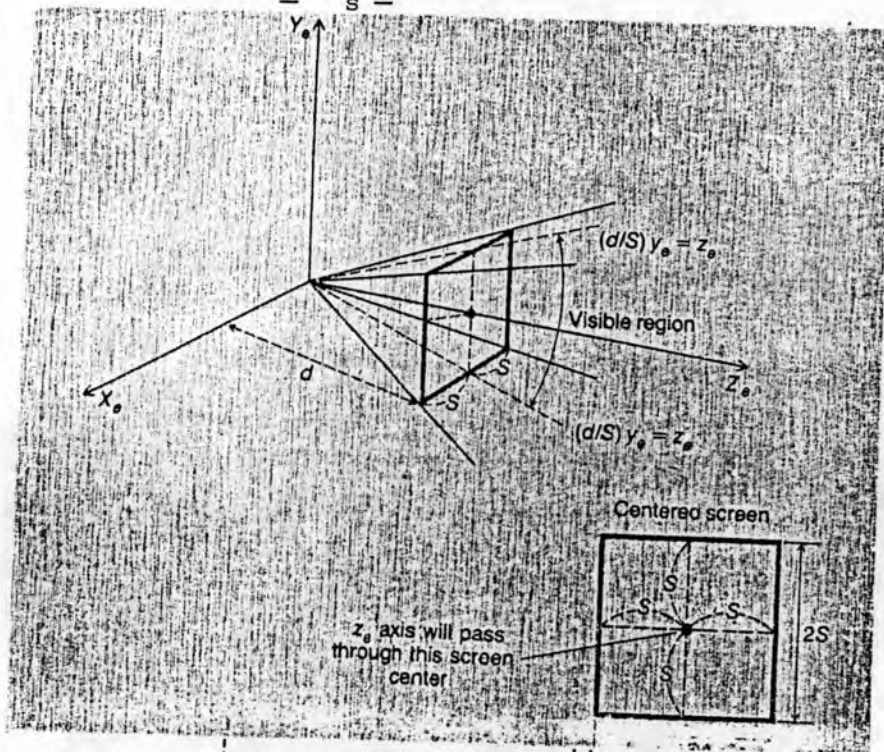
$$x_s = \left( \frac{d}{S} \right) \frac{x_e}{z_e} \quad (4.67)$$

$$y_s = \left( \frac{d}{S} \right) \frac{y_e}{z_e} \quad (4.68)$$

ดังแสดงในรูปที่ 4.26 ถ้าเราให้ viewpoint อยู่ที่เดิม  $d/S$  จะเป็นตัวควบคุมภาพบนจอในลักษณะคล้ายกับความยาวโฟกัสของเลนส์กล้องถ่ายรูป ถ้า  $d/S$  น้อย ความยาวโฟกัสจะน้อย ภาพที่ปรากฏบนจอจะเหมือนถ่ายด้วยเลนส์มุมกว้าง แต่ถ้า  $d/S$  มาก ภาพจะเหมือนถ่ายด้วยเลนส์มุมแคบ (Telephoto) เราสรุปได้ว่า ภาพที่จะปรากฏบนจอได้ ต้องมีค่าพิกัดอยู่ในขอบเขตดังนี้

$$-1 \leq x_s \leq 1$$

$$-1 \leq y_s \leq 1$$



รูปที่ 4.26 แสดง Origin อยู่กึ่งกลางจอภาพ

4.2.4 ความลึกของภาพ 3 มิติ (Perspective Depth) หัวข้อนี้จะกล่าวถึงการหาความลึกของภาพ 3 มิติ ซึ่งจะต้อนนำไปใช้ในการกำหนดขอบเขตของภาพ (หัวข้อ 4.2.5) และการกำจัดเส้นที่ถูบัง (หัวข้อ 4.3)

ในขั้นแรกเราต้องขยาย Screen Coordinate System ให้เป็น 3 มิติ ก่อน คือ เปลี่ยนจาก  $(x_s, y_s)$  เป็น  $(x_s, y_s, z_s)$  ในขั้นที่สอง เราถือว่าเส้นตรงใน Eye Coordinate System ยังคงเป็นเส้นตรงใน Screen Coordinate System อยู่คือหลังจากเปลี่ยนให้จุดปลายทั้งสองของเส้นตรงมาอยู่ใน Screen Coordinate System แล้วลากเส้นตรงต่อจุดปลายทั้งสองได้เลย นั่นคือระนาบตา ใน Eye Coordinate System สามารถถูกเปลี่ยนไปอยู่ใน Screen Coordinate System ได้โดยอาศัยสมการของระนาบ ดังนี้

$$\text{จาก } Ax_e + By_e + Cz_e + D = 0$$

$$A'x_s + B'y_s + C'z_s + D' = 0 \quad (4.69)$$

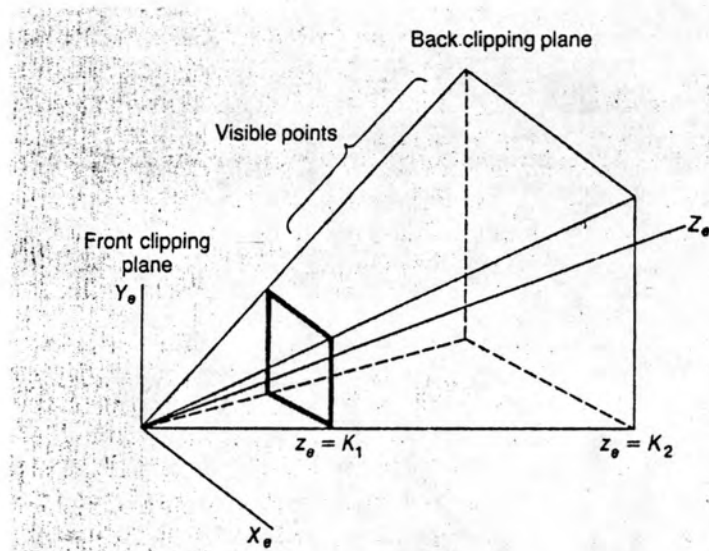
แทนค่า  $x_s$  และ  $y_s$  จากสมการ 4.67 และ 4.68 ลงในสมการ 4.69

$$\text{ได้ } z_s = \alpha - \frac{\beta}{z_e} \quad (4.70)$$

$$\text{โดย } \alpha = D'/C' \quad \beta = (d/C'S)(A'x_e + B'y_e)$$

วิธีการหาค่า  $z_s$  ต่อไปนั้นเราอาศัยหลักการของ Newman และ Sproull(11) ในการหาค่า  $\alpha$  และ  $\beta$  ดังนี้

- กำหนดขอบเขตของภาพโดยให้  $K_1 \leq z_e \leq K_2$  จะได้รูปปริามิคใน Eye Coordinate System ดังแสดงในรูป 4.27



รูปที่ 4.27 รูปปิรามิดใน Eye Coordinate System จุดที่จะมองเห็นจะต้องมีค่าที่กัก

$$K_1 \leq z_e \leq K_2$$

- ให้  $z_e = K_1$  เป็นค่าที่น้อยที่สุดของ  $z_s$  และให้  $z_e = K_2$  เป็นค่าที่มากที่สุดของ  $z_s$   $z_s$  มีค่าตั้งแต่ 0 ถึง 1

- เพราะ  $z_s = 0$   $z_e = k_1$  และ  $z_s = 1$   $z_e = k_2$  แทนค่าลงในสมการ 4.70 จะได้

$$0 = \alpha - \frac{\beta}{K_1}$$

$$1 = \alpha - \frac{\beta}{K_2}$$

แก้สมการทั้งสองจะได้

$$\alpha = \frac{K_2}{K_2 - K_1} \quad \beta = \frac{K_1 K_2}{K_2 - K_1}$$

- เราหา  $z_s$  โดยแทนค่า  $\alpha$  และ  $\beta$  ลงในสมการ 4.70



$$z_s = \frac{K_2 (z_e - K_1)}{(K_2 - K_1) z_e}, \quad 0 \leq z_s \leq 1 \quad (4.71)$$

- เมื่อเราเปรียบเทียบจุดสองจุด จุดที่มีค่า  $z_s$  น้อยกว่าจะเป็นจุดที่อยู่ใกล้

Viewpoint มากกว่า

4.2.5 การสร้างขอบเขตของภาพในระบบ 3 มิติ (Viewport Planning for 3 - D Graphics) การกำหนดขอบเขตของภาพ 3 มิติ ก็คือการกำหนดจำนวนข้อมูลที่จะนำมาสร้างภาพบนจอ ดังเช่นในรูปที่ 4.27 เรากำหนดระนาบกำหนดขอบเขตด้านหน้า (Front Clipping Plane) และระนาบกำหนดขอบเขตด้านหลัง (Back Clipping Plane) ระนาบทั้งสองนี้จะขนานกับระนาบของจอภาพและตั้งฉากกับแนวของสายตา ( $z_e$ )

Clipping Points ทำได้โดยเปรียบเทียบจุดใด ๆ ตามสมการต่อไปนี้

$$-1 \leq x_s \leq 1 \rightarrow -z_e \leq \left(\frac{d}{S}\right) x_e \leq z_e \rightarrow -w \leq x_e \leq w \quad (4.72)$$

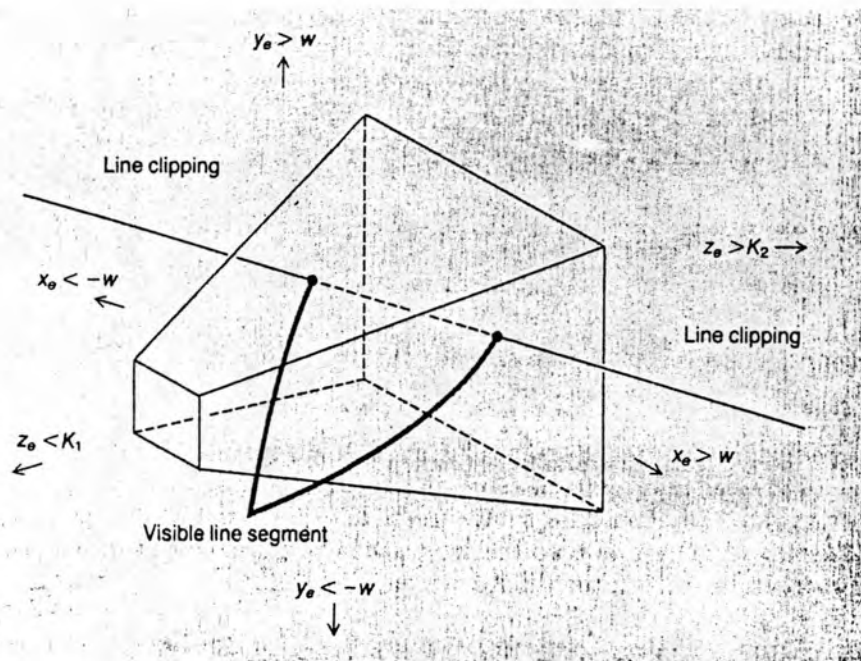
$$-1 \leq y_s \leq 1 \rightarrow -z_e \leq \left(\frac{d}{S}\right) y_e \leq z_e \rightarrow -w \leq y_e \leq w \quad (4.73)$$

$$0 \leq z_s \leq 1 \rightarrow K_1 \leq z_e \leq K_2 \quad (4.74)$$

โดยที่  $w = \left(\frac{S}{d}\right) z_e$  และ  $w > 0$

ถ้าจุดใด นำค่าพิกัดใน Eye Coordinate System มาทดสอบแล้วผ่านสมการ 4.72 ถึง 4.74 จึงเปลี่ยนค่าพิกัดนั้นไปเป็น Screen Coordinate System ได้ เพราะแสดงว่าเป็นจุดที่สามารถมองเห็นบนจอภาพได้

สำหรับ Clipping Lines นั้น จะมีปัญหายุ่งยากขึ้น ให้พิจารณารูปที่ 4.28 ซึ่งเป็น การแสดงปริมาตรของภาพ (View Volume) ซึ่งถูกกำหนดโดยสมการ 4.72 ถึง 4.74



รูปที่ 4.28 แสดง View Volume

เราจะกำหนดตัวแปรในการสร้างขอบเขตของภาพในทำนองเดียวกับที่ใช้ในระบบ 2 มิติ (หัวข้อ 4.1.5) ดังนี้

$$\begin{aligned}
 T(I) &= \text{จุดอยู่เหนือ View Volume } y_e > w \\
 B(I) &= \text{จุดอยู่ใต้ View Volume } y_e < -w \\
 R(I) &= \text{จุดอยู่ขวามือของ View Volume } x_e > w \\
 L(I) &= \text{จุดอยู่ซ้ายมือของ View Volume } x_e < -w \\
 BH(I) &= \text{จุดอยู่หลัง View Volume } z_e > K_2 \\
 F(I) &= \text{จุดอยู่หน้า View Volume } z_e < K_1
 \end{aligned}$$

ถ้าตัวแปรดังกล่าวข้างบนเป็นจริง ถือว่ามีค่าเป็น 1 ถ้าไม่จริงถือว่ามีค่าเป็น 0 เราแบ่งพิจารณาได้ 3 ประเภท ดังนี้

ประเภทที่ 1 ถ้าพิจารณาจุดปลายทั้งสองของเส้น แล้วได้ผลดังนี้

$$T(1) + B(1) + R(1) + L(1) + BH(1) + F(1) \\ + T(2) + B(2) + R(2) + L(2) + BH(2) + F(2) = 0 \quad (4.75)$$

แสดงว่า เส้นนี้ทั้งเส้นอยู่ใน View Volume

ประเภทที่ 2 ถ้าเส้นทั้งเส้นไม่อยู่ใน View Volume เลขจะไดผล ดังนี้

$$T(1).T(2) + R(1).R(2) + L(1).L(2) \\ + B(1).B(2) + BH(1).BH(2) + F(1).F(2) \neq 0 \quad (4.76)$$

ประเภทที่ 3 คือ ประเภทที่ไม่เป็นไปตามสมการ 4.75 และ 4.76 ทั้งหมด  
ประเภทนี้เราต้องคำนวณหาจุดตัดระหว่างเส้นกับระนาบของ View Volume โดยการสร้างสมการ  
ดังนี้ ถ้า  $P_1(x_{e1}, y_{e1}, z_{e1})$  และ  $P_2(x_{e2}, y_{e2}, z_{e2})$  เป็นจุดปลายทั้งสองของเส้นตรงเรา  
เขียนสมการได้เป็น

$$x_e = (x_{e2} - x_{e1}) \cdot t + x_{e1} \\ y_e = (y_{e2} - y_{e1}) \cdot t + y_{e1} \\ z_e = (z_{e2} - z_{e1}) \cdot t + z_{e1} \quad (4.77)$$

โดย  $0 \leq t \leq 1$  และ  $(x_e, y_e, z_e)$  คือ ค่าพิกัดของจุดใดๆ บนเส้นตรงนั้น จะเห็น  
ได้ว่าเมื่อ  $t = 0$  จะได้  $(x_e, y_e, z_e) = (x_{e1}, y_{e1}, z_{e1})$  และเมื่อ  $t = 1$  จะได้

$$(x_e, y_e, z_e) = (x_{e2}, y_{e2}, z_{e2})$$

ถ้าเราต้องการหาจุดตัดระหว่างเส้นกับระนาบบนของ View Volume ก็แทนค่า  
 $y_e = w$  ลงในสมการ 4.77 ได้

$$(y_{e2} - y_{e1}) \cdot t + y_{e1} = w \\ = (s/d) z_e \\ = (s/d) [(z_{e2} - z_{e1}) \cdot t + z_{e1}]$$

$$t = \frac{(s/d) z_{e1} - y_{e1}}{(y_{e2} - y_{e1}) - (s/d) (z_{e2} - z_{e1})} \quad (4.78)$$

ถ้าค่า  $t$  ที่ได้ ไม่อยู่ในช่วง 0 ถึง 1 แสดงว่า เส้นนั้นไม่ได้ตัดระนาบบน แต่ถ้าอยู่ในช่วง 0 ถึง 1 ให้แทนค่า  $t$  ลงในสมการ 4.77 ก็จะทราบจุดตัดนั้น แล้วส่วนของเส้นที่อยู่ นอก View Volume ก็จะถูกกำจัดออก สำหรับการหาจุดตัดของเส้นกับระนาบอื่นๆ ก็ทำในลักษณะเดียวกัน

ตัวอย่าง เช่น ในการพิจารณาเส้นตรงที่มีปลายทั้งสองใน Eye Coordinate System เป็น  $P_1 (x_{e1}, y_{e1}, z_{e1}) = (1, 6, 2)$  และ  $P_2 (x_{e2}, y_{e2}, z_{e2}) = (4, -2, 5)$  ให้  $s = d$  และ  $4 \leq z_e \leq 10$  สำหรับจุด  $P_1$  เราได้  $w = (s/d) z_{e1} = 2$  และจุด  $P_2$  ได้  $w = (s/d) z_{e2} = 5$  ดังนั้นค่าตัวแปรต่างๆ จะเป็นดังนี้

$$T(1) = 1 \quad T(2) = 0$$

$$B(1) = 0 \quad B(2) = 0$$

$$R(1) = 0 \quad R(2) = 0$$

$$L(1) = 0 \quad L(2) = 0$$

$$BH(1) = 0 \quad BH(2) = 0$$

$$F(1) = 1 \quad F(2) = 0$$

จากค่าของตัวแปร เราสรุปได้ว่า  $P_1$  อยู่นอก View Volume แต่  $P_2$  อยู่ใน View Volume เราเริ่มต้นโดยใช้สมการ 4.78 เพื่อตรวจสอบว่าเส้นตรงนี้จะตัดระนาบบนหรือไม่ ซึ่งได้  $t = 4/11$  อยู่ระหว่าง 0 ถึง 1 นำค่า  $t$  ไปแทนในสมการ 4.77 ได้ค่า

$$x_e = (4-1) (4/11) + 1 = 23/11$$

$$y_e = (-2-6) (4/11) + 6 = 34/11$$

$$z_e = (5-2) (4/11) + 2 = 34/11$$

เรายังสรุปว่าจุดที่เส้น  $P_1P_2$  ตัด View Volume ไม่ได้เพราะค่า  $z_e$  ที่น้อยที่สุด คือ 4 มากกว่า  $34/11$  แสดงว่า จุดตัดนี้ยังไม่ถูกต้องจึงต้องคำนวณหาค่า  $t$  สำหรับระนาบอื่นๆ อีกใดค่างนี้

| ระนาบ | $t$      | มีจุดตัดกับระนาบหรือไม่   |
|-------|----------|---------------------------|
| บน    | $4/11$   | ไม่ ( $z_e = 34/11 < 4$ ) |
| กลาง  | $-3/5$   | ไม่                       |
| ขวา   | $\infty$ | ไม่                       |
| ซ้าย  | $-1/6$   | ไม่                       |
| หลัง  | $3$      | ไม่                       |
| หน้า  | $2/3$    | มี ( $z_e = 4$ )          |

ดังนั้นเราคำนวณหาค่าพิกัดของจุดตัดได้

$$x_e = (4 -) (2/3) + 1 = 3$$

$$y_e = (-2 - 6) (2/3) + 6 = 2/3$$

$$z_e = (5 - 2) (2/3) + 2 = 4$$

จุด  $(3, 2/3, 4)$  ก็คือจุดซึ่งเส้น  $P_1P_2$  ตัดกับระนาบด้านหน้าของ View Volume

หลังจากเราได้ขอบเขตของภาพแล้วการนำภาพไปแสดงบนจอภาพก็ทำในทำนองเดียวกับภาพ 2 มิติ เช่น ถ้าจอภาพมีแกน  $y$  ทิศทางพุ่งลงและจุด Origin อยู่มุมซ้ายบนของจอภาพ เราจะได้

$$x_s = S C F [ (d/S) (x_e/z_e) V_x + L ] \quad (4.79)$$

$$y_s = -(d/S) (y_e/z_e) V_y + M \quad (4.80)$$

โดยที่ Viewport Coordinate System มี Origin อยู่ที่ระยะ  $L$  และ  $M$  ของ Origin ของ Graphic Mode  $2 V_x$  และ  $2 V_y$  คือหน่วยความกว้างและความสูงของ

Graphic Mode ที่ใช้แสดงภาพ

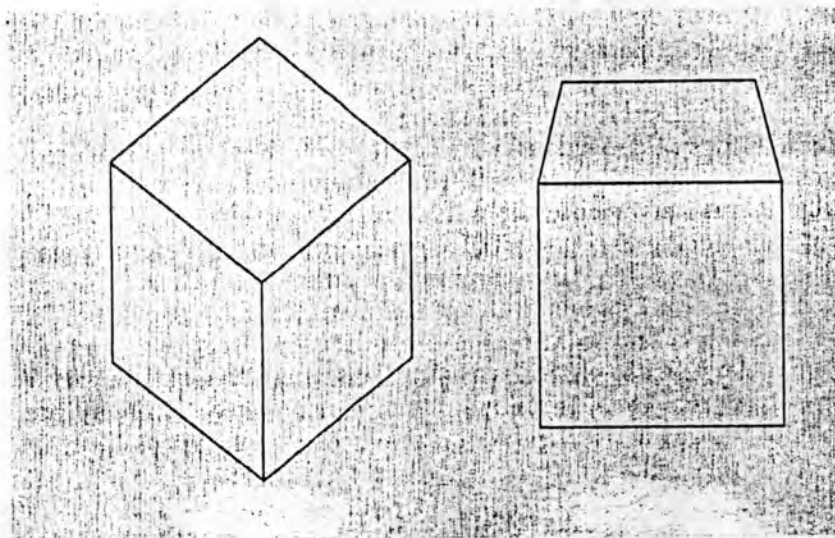
#### 4.3 การกำจัดเส้นและพื้นผิวที่ถูกบัง

ในการสร้างภาพบนจอคอมพิวเตอร์ บางครั้งเราต้องการกำจัดเส้นและพื้นผิวที่ถูกบัง (Hidden Lines and Surfaces) เราต้องหาวิธีตรวจสอบว่าเส้นและพื้นผิวแต่ละเส้นนั้นสามารถเห็นได้ทั้งหมด เห็นเป็นบางส่วน หรือไม่สามารถเห็นได้เลย เพื่อกำจัดส่วนที่มองไม่เห็น (หรือถูกบัง) นี้ออกไปก่อนสร้างภาพบนจอ

สมมติฐานประการแรก ที่ใช้ในหัวข้อนี้ คือ พื้นผิวใดๆ ของวัตถุประมาณได้ว่าเป็นรูปหลายเหลี่ยม (Polygons) สมมติฐานประการที่สอง คือ รูปหลายเหลี่ยมนั้นราบเรียบ (Flat) นั่น คือ จุดที่มุมของรูปหลายเหลี่ยมจะอยู่บนระนาบเดียวกันทุกจุด

##### 4.3.1 การมองเห็นได้ของวัตถุชิ้นเดียว (The Visibility of Single Object)

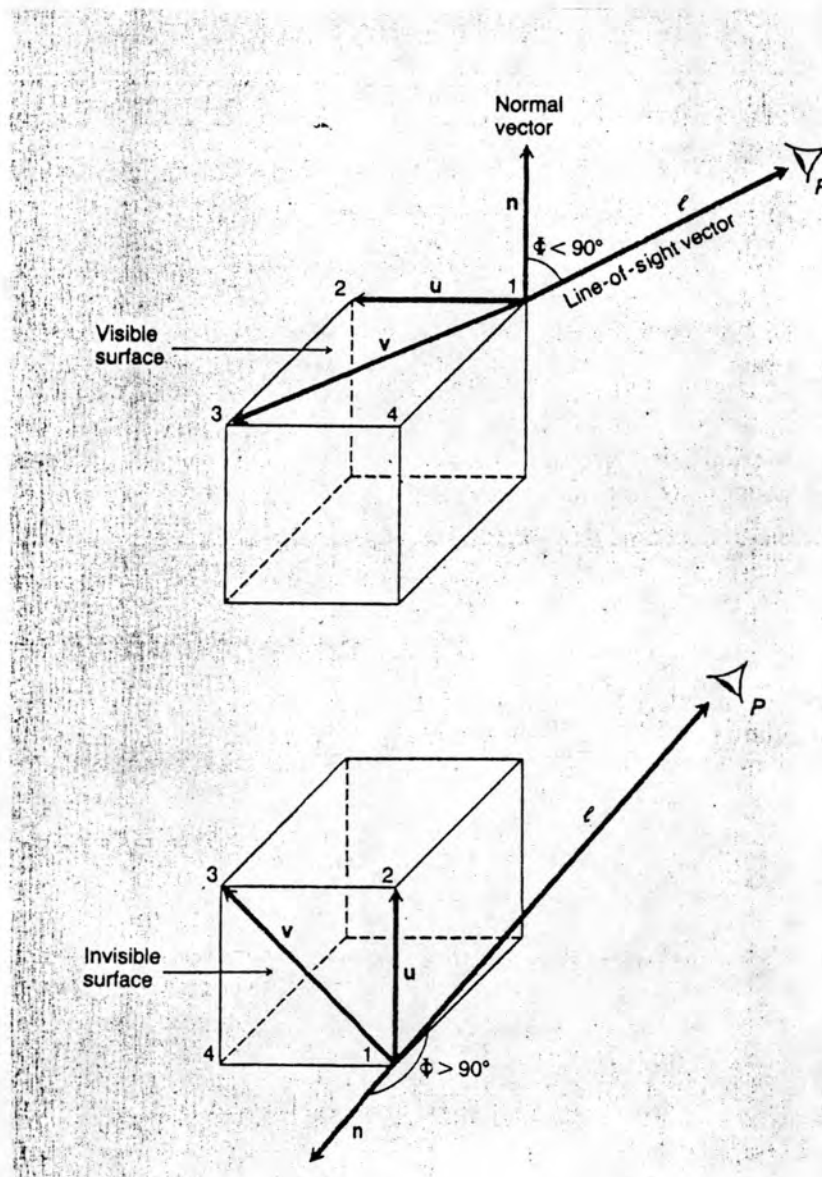
ให้พิจารณารูปลูกบาศก์ ที่มี 8 มุม 12 ขอบและ 6 ด้าน ดังรูปที่ 4.29



รูปที่ 4.29 แสดงรูปสี่เหลี่ยมลูกบาศก์ซึ่งมองจากมุมต่าง ๆ

จะเห็นว่าความบางมุมเราจะมองเห็น 2 ด้าน บางมุมจะเห็น 3 ด้าน ดังนั้นเราต้องหาวิธีแยกแยะ เส้นต่างๆ ว่าที่มุมมองนั้น เส้นใดจะเห็นได้และเส้นใดจะไม่เห็น

การทดสอบการมองเห็นได้ (Visibility Test) ทำโดยสร้างเวกเตอร์ 2 ตัว บนแต่ละพื้นผิว คือ เวกเตอร์ที่ตั้งฉากกับพื้นผิว (Surface Normal Vector) และเวกเตอร์ตามแนวสายตา (Line of Sight Vector) โดยที่ Surface Normal Vector ( $n$ ) เป็นเวกเตอร์ที่ตั้งฉากกับพื้นผิวตรงจุดซึ่งอยู่นอกสุดของพื้นผิวและ Line of Sight Vector ( $l$ ) คือ เส้นที่ผ่านจุดมองและจุดเริ่มต้นของ Surface Normal Vector ดังแสดงในรูปที่ 4.30



รูปที่ 4.30 Normal และ Line of Sight Vector

มุม  $\phi$  คือ มุมระหว่าง  $n$  และ  $l$  ถ้ามุมนี้มีค่าระหว่าง  $0^\circ$  ถึง  $90^\circ$  แสดงว่าพื้นผิวชั้นมองเห็นได้ ถ้ามุมนี้มีค่าเกิน  $90^\circ$  พื้นผิวชั้นจะมองไม่เห็น

การคำนวณหา Normal Vector นั้น ในขั้นแรกต้องกำหนดหมายเลขประจำมุม บนพื้นผิวก่อน จะเริ่มต้นที่จุดใดก็ได้ แต่ต้องวนไปในทิศทางเข็มนาฬิกา (โดยมองจากด้านนอกของวัตถุเข้าไป) ดังแสดงในรูปที่ 4.30 จากนั้นกำหนด Vector  $u$  จากจุดที่ 1 ไป 2 และ Vector  $v$  จากจุดที่ 1 ไป 3 Cross Product ( $u \times v$ ) จะตั้งฉากกับพื้นผิวและมีทิศทางพุ่งออก ดังนี้

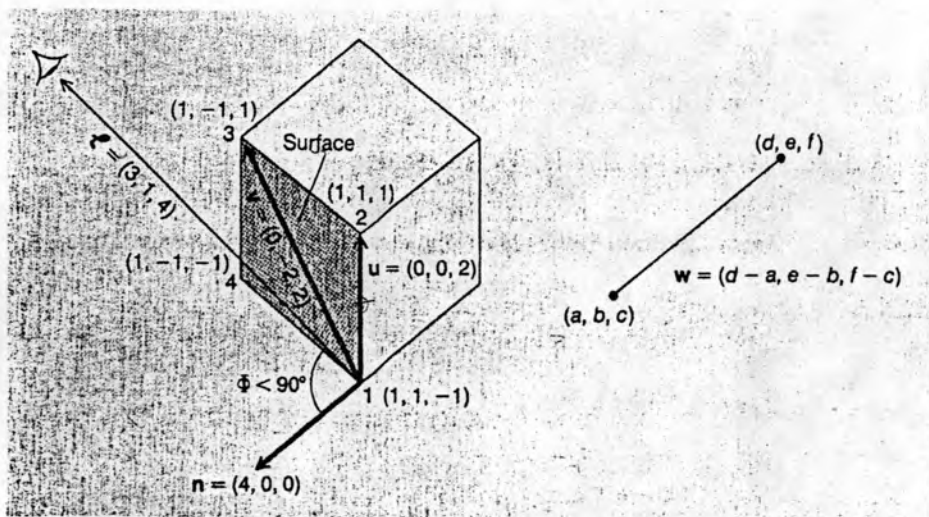
$$N = u \times v$$

$$= (bf - ce, cd - af, ae - bd) \tag{4.81}$$

โดย  $u = (a,b,c)$  และ  $v = (d,e,f)$

เช่น ถ้าเรากำหนดค่าพิกัดของจุดในรูปที่ 4.31 ดังนี้

| จุดที่ | ค่าพิกัด (x,y,z) |
|--------|------------------|
| 1      | (1, 1, -1)       |
| 2      | (1, 1, 1)        |
| 3      | (1, -1, 1)       |
| 4      | (1, -1, -1)      |



รูปที่ 4.31 แสดง Visibility Test



$$u = (1, 1, 1) - (1, 1, -1) = (0, 0, 2)$$

$$v = (1, -1, 1) - (1, 1, -1) = (0, -2, 2)$$

$$n = u \times v = (4, 0, 0)$$

จากนั้นเรากำหนดหา Line of Sight Vector โดยลากเส้นจากจุดมอง  
 $P(D, \theta, \phi)$  ที่กำหนดไปยังจุดที่ 1 และเนื่องจาก  $P(D, \theta, \phi) = P(D \sin \phi \cos \theta, D \sin \phi \sin \theta, D \cos \phi)$  ดังนั้นเรากำหนดค่า  $l$  ได้

$$l = (D \sin \phi \cos \theta, D \sin \phi \sin \theta, D \cos \phi) - (x_1, y_1, z_1) \quad (4.82)$$

ถ้าเราให้ค่าพิกัดของจุดมองเป็น  $(5.4, 27, 56) = (4, 2, 3)$  เราจะได้

$$l = (4, 2, 3) - (1, 1, -1) = (3, 1, 4)$$

เมื่อได้ค่าของ  $n$  และ  $l$  แล้วสามารถทำ Visibility Test ได้โดยหา  
 Dot Product ของ  $n$  และ  $l$  ดังนี้

$$n \cdot l = (n_1, n_2, n_3) \cdot (l_1, l_2, l_3) = n_1 l_1 + n_2 l_2 + n_3 l_3 \quad (4.83)$$

จากหลักของพีชคณิตเรทราบคุณสมบัติของ Dot Product ในสมการ 4.83 เป็น  
 ดังนี้

$$n \cdot l = |n| \cdot |l| \cdot \cos \phi$$

$$\phi = \arccos \left[ \frac{n \cdot l}{|n| \cdot |l|} \right] \quad (4.84)$$

โดย  $|n|$  = ความยาวของ  $n$

$$|l| = \text{ความยาวของ } l$$

$$\phi = \text{มุมระหว่าง } n \text{ และ } l$$

สำหรับพื้นผิวที่จะมองเห็นได้  $\phi$  จะมีค่าระหว่าง  $0^\circ$  ถึง  $90^\circ$  และ  $n \cdot l \geq 0$  สำหรับ

พื้นผิวที่มองไม่เห็น  $\phi$  จะมีค่าระหว่าง  $90^\circ$  ถึง  $180^\circ$  และ  $n \cdot 1 < 0$

ดังนั้นตัวอย่างตามรูปที่ 4.31 จะได้

$$n \cdot 1 = (4, 0, 0) \cdot (3, 1, 4) = 12 > 0$$

นั่นคือพื้นผิวที่มองเห็นได้จากจุดมองที่  $(4, 2, 3)$

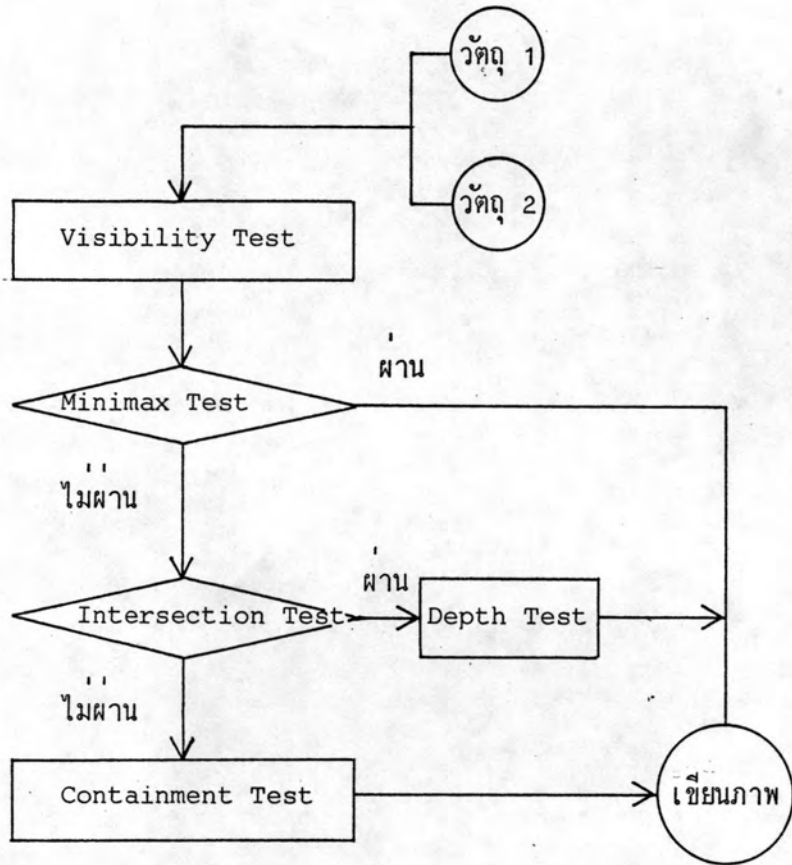
4.3.2 การมองเห็นได้ของวัตถุหลายชิ้น (The Visibility of Several Objects) ในกรณีที่เราต้องการกำจัดเส้นและพื้นผิวที่ถูกบังของวัตถุที่อยู่รวมกันหลายชิ้น ต้องทำตามขั้นตอนดังนี้

ขั้นตอนที่ 1 ทำ Visibility Test ของวัตถุแต่ละชิ้นโดยวิธีดังแสดงในหัวข้อ 4.3.1 เพื่อหาว่าวัตถุแต่ละชิ้นมีด้านใดบ้างที่มองเห็นได้

ขั้นตอนที่ 2 เปรียบเทียบด้านที่มองเห็นได้ทุกด้านของวัตถุทุกชิ้นว่า ขอบของด้านติดกันหรือไม่ ด้านหนึ่งของวัตถุชิ้นหนึ่งอาจบังด้านของวัตถุอีกชิ้นจนมิด และบางครั้งด้านของวัตถุ 2 ชิ้นที่มาเปรียบเทียบกับกันอาจไม่ซ้อนกันเลยก็ได้สำหรับขั้นตอนนี้ จะทำการทดสอบว่า เป็นไปได้หรือไม่ว่าด้านที่นำมาเปรียบเทียบกับกันนั้นไม่ใดซ้อนกันเลย การทดสอบนี้เรียกว่า Minimax Test

ขั้นตอนที่ 3 ถ้า Minimax Test แล้วไม่ผ่านแสดงว่าเป็นไปได้ที่ด้าน 2 ด้านจะซ้อนกัน ให้ทำการทดสอบว่าด้าน 2 ด้านของวัตถุ 2 ชิ้น ที่เรากำลังเปรียบเทียบอยู่นั้นขอบของด้านหนึ่งจะไปตัดขอบของอีกด้านหนึ่งหรือไม่ การทดสอบนี้เรียกว่า Intersection Test ถ้าไม่ตัดกันก็มีทางเป็นไปได้อีก 2 ทาง ทางแรก คือ ด้านหนึ่งบังอีกด้านหนึ่งจนมิด ทางที่สอง คือ ด้านทั้ง 2 นั้นไม่ใดซ้อนกันเลย (ถึงแม้จะทำ Minimax Test มาแล้วแต่บางครั้งก็มีกรณีที่ด้าน 2 ด้านไม่ใดซ้อนกัน แต่ผ่าน Minimax Test มาได้) จึงต้องมาทดสอบว่าด้านหนึ่งจะบังอีกด้านหนึ่งจนมิดหรือไม่ การทดสอบนี้เรียกว่า Containment Test ในกรณีที่ Intersection Test ให้ผลมาว่ามี การตัดกัน ต้องมาทำการทดสอบว่าส่วนใดบ้างที่จะถูกบัง ซึ่งเรียกว่า Depth Test

ขั้นตอนทั้ง 3 นี้ สรุปได้ดังแสดงในรูปที่ 4.32



รูปที่ 4.32 ขั้นตอนการทำ Hidden Lines and Surfaces

4.3.2.1 Minimax Test พิจารณา รูปที่ 4.33 รูปหลายเหลี่ยม A และ B คือภาพฉาย (Projections) ของวัตถุ 3 มิติ 2 ชิ้น โดยแต่ละรูปเราต้องหาค่าพิกัดที่มากที่สุด และน้อยที่สุด ทั้งแกน x และแกน y แล้วเปรียบเทียบตามเงื่อนไข 4 เงื่อนไข ดังต่อไปนี้

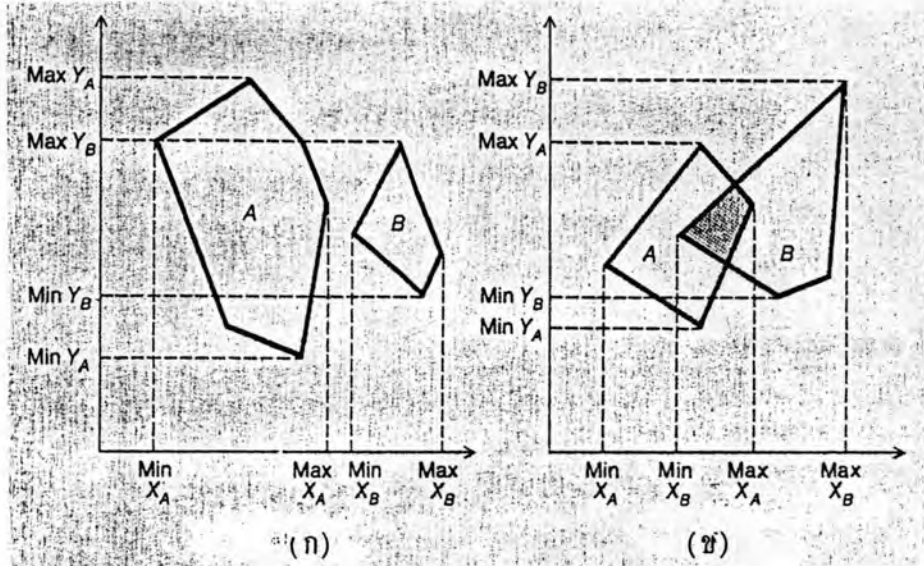
$$\max x_A < \min x_B$$

$$\max x_B < \min x_A$$

$$\max y_A < \min y_B$$

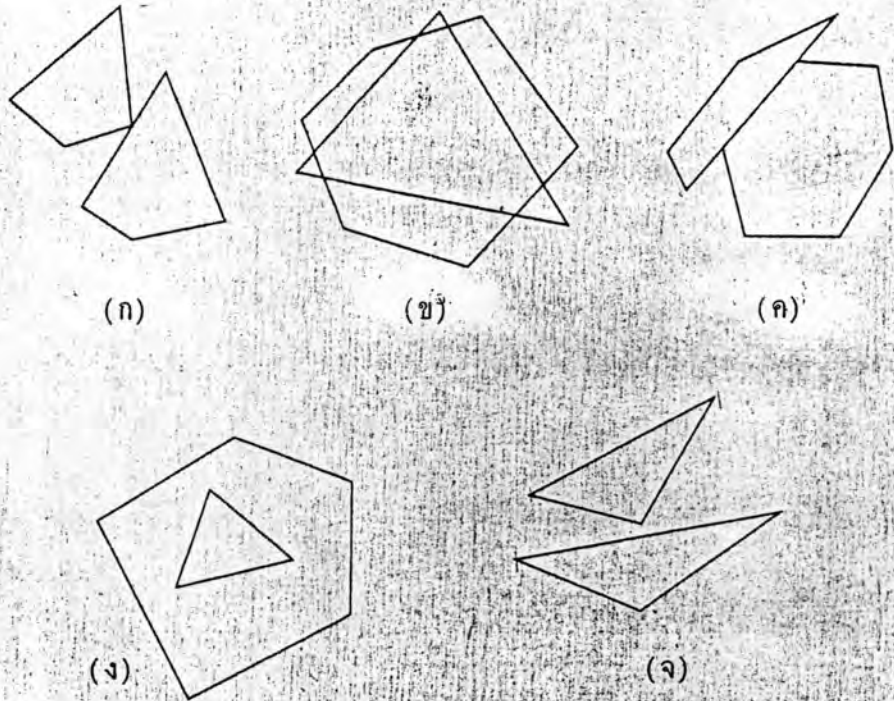
$$\max y_B < \min y_A$$

ถ้าผลการเปรียบเทียบแสดงให้เห็นว่า เพียง 1 หรือ เกิน 1 ในเงื่อนไขทั้ง 4 เป็นจริง แสดงว่ารูปทั้ง 2 ไม่ซ้อนกัน เช่น ในรูปที่ 4.33 นั้น  $\max x_A < \min x_B$  เป็นจริง รูป A และ รูป B จึงไม่ซ้อนกัน



รูปที่ 4.33 Minimax Test : (ก) ไม่ซ้อนกัน และ (ข) เป็นไปได้ที่จะซ้อนกัน

ในรูปที่ 4.34 แสดงถึงรูปที่นำมาทำ Minimax Test และทั้ง 5 รูปผลการทดสอบจะแสดงว่าเป็นไปได้ที่จะซ้อนกัน ทั้งๆ ที่ รูป 4.34 (จ) ไม่ได้ซ้อนกันเลย



รูปที่ 4.34 (ก) ตัด 1 จุด (ข) ตัดหลายจุด (ค) จำนวนจุดที่ตัดกันเป็นอนันต์ (ง) บังกัน และ (จ) ไม่ซ้อนกัน

### 4.3.2.2 Intersection Test พิจารณาสมการเส้นตรงของเส้น

L และ L'

$$L : Ax + By + C = 0$$

$$L' : Dx + Ey + F = 0$$

ความสัมพันธ์ของเส้นทั้ง 2 แบ่งเป็น 3 ประเภท คือ

ประเภทที่ 1 เส้นทั้งสองตัดกันที่จุด  $(x_i, y_i)$

$$x_i = \frac{BF - CE}{AE - BD}, \quad y_i = \frac{CD - AF}{AE - BD}$$

ประเภทที่ 2 เส้นทั้งสองขนานกัน ซึ่งหมายความว่าเส้นทั้งสองไม่มีโอกาสตัดกันเลย ซึ่งกรณีนี้จะได้อะไร  $AE - BD = 0$

ประเภทที่ 3 เส้นทั้งสองทับซ้อนกันโดยตลอด กรณีนี้จะได้ว่า  $A/D = B/E = C/F$

พิจารณารูปที่ 4.35 เส้น G และ G' เป็นส่วนของเส้น L และ L' ตามลำดับโดยเส้น G มีปลายทั้งสองอยู่ที่  $(x_1, y_1)$  และ  $(x_2, y_2)$  ส่วนเส้น G' มีปลายทั้งสองอยู่ที่  $(x'_1, y'_1)$  และ  $(x'_2, y'_2)$  เปรียบเทียบค่าพิกัดทั้งหมดตามเงื่อนไข 8 เงื่อนไข ดังต่อไปนี้

$$x_i < \min(x_1, x_2) \quad x_i > \max(x_1, x_2) \quad x_i < \min(x'_1, x'_2) \quad x_i > \max(x'_1, x'_2)$$

$$y_i < \min(y_1, y_2) \quad y_i > \max(y_1, y_2) \quad y_i < \min(y'_1, y'_2) \quad y_i > \max(y'_1, y'_2)$$

ถ้าเงื่อนไขเพียง 1 หรือมากกว่า 1 ในเงื่อนไขทั้ง 8 เป็นจริงแสดงว่าเส้น G และ G' ไม่ตัดกัน

ที่กล่าวมาแล้วเป็นทฤษฎีของ Intersection Test ในการทดสอบจริงๆ เราจะทำดังนี้ สมมติว่า คานหนึ่งของรูปหลายเหลี่ยม A มีจุดปลายทั้งสองอยู่ที่  $(x_1, y_1)$  และ  $(x_2, y_2)$  สมการของเส้นนี้ คือ

$$y = y_1 + m(x - x_1) \quad (4.85)$$

$$mx - y + (y_1 - mx_1) = 0 \quad (4.86)$$

$$\text{โดย } m = (y_2 - y_1) / (x_2 - x_1)$$

และค่านึงของรูปหลายเหลี่ยม B มีจุดปลายทั้งสองอยู่ที่  $(x'_1, y'_1)$  และ  $(x'_2, y'_2)$   
สมการของเส้นนี้ คือ

$$y' = y'_1 + m'(x' - x'_1) \quad (4.87)$$

$$m'x' - y' + (y'_1 - m'x'_1) = 0 \quad (4.88)$$

$$\text{โดย } m' = (y'_2 - y'_1) / (x'_2 - x'_1)$$

ในการหาจุดตัดของเส้นทั้งสองทำเป็นขั้นตอน ดังนี้

ขั้นตอนที่ 1 ถ้า  $m' - m = 0$  แสดงว่า 2 เส้นนี้ขนานกันให้ข้ามไปขั้นตอนที่ 5

ขั้นตอนที่ 2 ถ้า  $m/m' = 1 = (y_1 - mx_1) / (y'_1 - m'x'_1)$  แสดงว่าเส้น  
ทั้งสองนี้ทับกันให้ข้ามไปขั้นตอนที่ 5

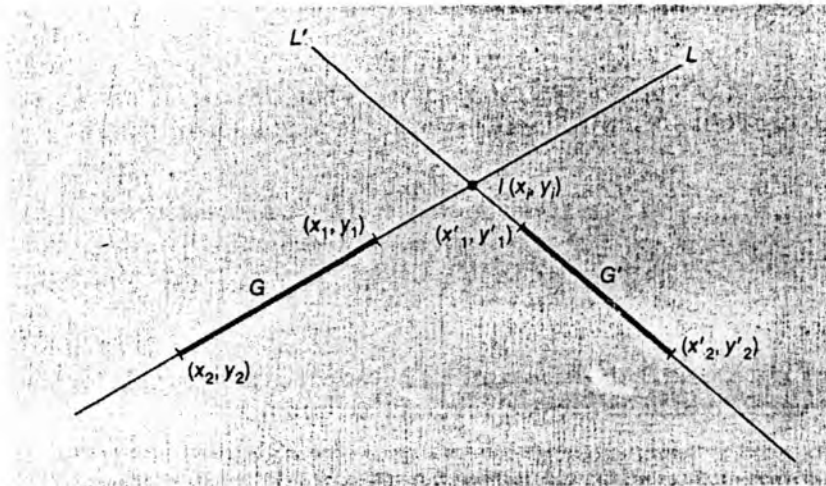
ขั้นตอนที่ 3 คำนวณค่าพิภคของจุดตัด  $(x_i, y_i)$  ได้โดยแก้สมการ 4.85 และ  
4.87 จะได้

$$x_i = \frac{(y_1 - y'_1) - (m x_1 - m' x'_1)}{m' - m}$$

$$y_i = \frac{(y_1 - m x_1) m' - (y'_1 - m' x'_1) m}{m' - m}$$

ขั้นตอนที่ 4 เปรียบเทียบตาม 8 เงื่อนไขที่กล่าวมาแล้ว

ขั้นตอนที่ 5 เปลี่ยนค่านึงของรูปหลายเหลี่ยม B แล้วเริ่มขั้นตอนที่ 1 ใหม่



รูปที่ 4.35 Intersection Test

4.3.2.3 Containment Test หรือการทดสอบว่า รูปหลายเหลี่ยมสองรูปที่เราตรวจสอบนั้น รูปหนึ่งจะบังอีกรูปหนึ่งหรือไม่ ซึ่งมีวิธีที่นิยมมีอยู่ 2 วิธีดังจะอธิบายรายละเอียดต่อไปนี้

วิธีแรก คือ การคำนวณผลรวมของมุม พิจารณารูปที่ 4.36 จุด  $P_c$  เป็นจุดมุมจุดหนึ่งของรูปหลายเหลี่ยม A จุด  $P_j = (x_j, y_j)$  เป็นจุดมุมที่  $j$  ของรูปหลายเหลี่ยม B โดย  $j = 1, 2, \dots, n$  และ  $P_n = P_1$  เราจะเรียงลำดับเลขในทิศทางเข็มนาฬิกาให้  $P_c, P_j$  เป็นเส้นที่เชื่อมต่อกจุด  $P_c$  กับจุด  $P_j$  มุม  $\alpha_j$  เป็นมุมระหว่างเส้น  $P_c P_j$  กับเส้น  $P_c P_{j+1}$  โดยวัดจาก  $P_c P_j$  ซึ่งถ้าเป็นการวัดตามเข็มนาฬิกาจะถือว่าเป็นมุมลบ แต่ถ้าวัดทวนเข็มนาฬิกาจะถือว่าเป็นมุมบวก เรากำหนดมุม  $\alpha_j$  ให้ครบทุกค่าของ  $j$  แล้วพิจารณา ดังนี้

ถ้า  $\sum \alpha_j = 0^\circ$  แสดงว่าจุด  $P_c$  อยู่นอกรูป B

ถ้า  $\sum \alpha_j = 360^\circ$  แสดงว่าจุด  $P_c$  อยู่ในรูป B

ตัวอย่างเช่น จุดที่มุมทุกมุมของรูป B และจุดที่จะทดสอบของรูป A มีค่าพิกัดดังแสดงในรูปที่ 4.36 เรากำหนด ดังนี้

$$P_c P_1 = (30, 80) - (60, 40) = (-30, 40)$$

$$P_t P_2 = (40, 50) - (60, 40) = (-20, 10)$$

$$|P_t P_1| = \sqrt{(-30)^2 + (40)^2} = 50$$

$$|P_t P_2| = \sqrt{(-20)^2 + (10)^2} = 22.36$$



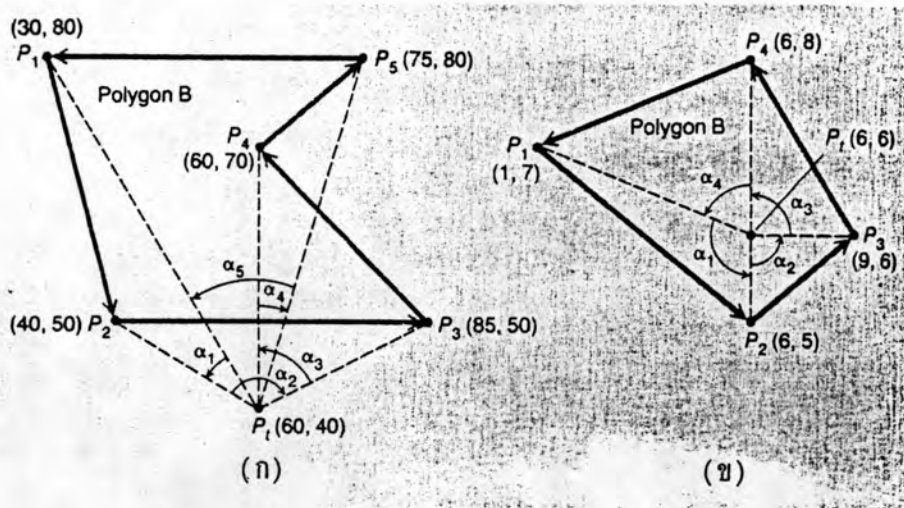
ดังนั้นจะหา Dot Product ของเวกเตอร์ได้

$$P_t P_1 \cdot P_t P_2 = (-30, 40) \cdot (-20, 10) = 1000$$

จากสมการ 4.84 จะได้

$$\alpha_1 = \cos^{-1} \frac{1000}{(50)(22.36)} = 26.56^\circ$$

การคำนวณหามุมจะทำในทำนองเดียวกันนี้ แล้วจึงนำค่ามุมทั้งหมดมาทดสอบดังที่กล่าวมาแล้ว ต้องทำการทดสอบทุกจุดมุมของรูป A ถ้ารูป A อยู่ในรูป B จุดที่มุมทุกจุดของรูป A ต้องอยู่ในรูป B ให้ระวังไว้ว่าถ้าผลการทดสอบบอกว่า จุดที่มุมทุกจุดของรูป A อยู่นอกรูป B อย่างรีบสรุปว่า รูป A และ B ไม่ทับซ้อนกันเลย ทั้งนี้อาจเป็นกรณีที่ รูป B อยู่ในรูป A ก็ได้ ต้องทดสอบใหม่โดยเปลี่ยนเป็นทดสอบว่าจุดที่มุมของรูป B อยู่ในรูป A หรือไม่



รูปที่ 4.36 Containment Test โดยการคำนวณผลรวมของมุม



วิธีที่สองของ Containment Test คือ วิธีครึ่งของระนาบ (Half-Plane Method) ให้พิจารณารูปที่ 4.36 (ก) เส้นที่ต่อจุด  $P$  และ  $Q$  จะแบ่งระนาบ  $x y$  ออกเป็นสองด้าน คือชายและขวา ในรูปที่ 4.36 (ข) เส้นต่อจุด  $P_j$  กับจุด  $P_{j+1}$  เป็นด้านหนึ่งของรูปหลายเหลี่ยม จุด  $P_t$  เป็นจุดที่ต้องการทดสอบ สร้างเวกเตอร์  $v$  และ  $u$  ดังแสดงในรูป หากค่าของ  $u \times v$  แล้วเปลี่ยนผลให้อยู่ในรูปของ  $K (0,0,1)$  จากนั้นจึงพิจารณาผลดังนี้

ถ้าค่า  $K$  เป็นลบแสดงว่าจุด  $P_t$  อยู่ขวามือของเส้น  $P_j P_{j+1}$  นั่น คือ จุด  $P_t$  จะอยู่นอกรูปหลายเหลี่ยม

ถ้าค่า  $K$  เป็นบวกแสดงว่าจุด  $P_t$  อยู่ซ้ายมือของเส้น  $P_j P_{j+1}$  นั่น คือ จุด  $P_t$  จะอยู่ในรูปหลายเหลี่ยม

จากทฤษฎีที่กล่าวมาแล้ว เราแสดงในรูปทางคณิตศาสตร์ได้ ดังนี้

ให้  $P_j = (a,b)$   $P_{j+1} = (c,d)$  และ  $P_t = (e,f)$  ดังนั้น  
 $u = (e - a, f - b)$  และ  $v = (c - a, d - b)$  เนื่องจาก Cross Product  $u \times v$  นั้น  
 ต้องหาจากเวกเตอร์ 3 มิติ จึงให้ค่าพิคัดในแกน  $z$  ของ  $u$  และ  $v$  เป็น ศูนย์ ดังนี้

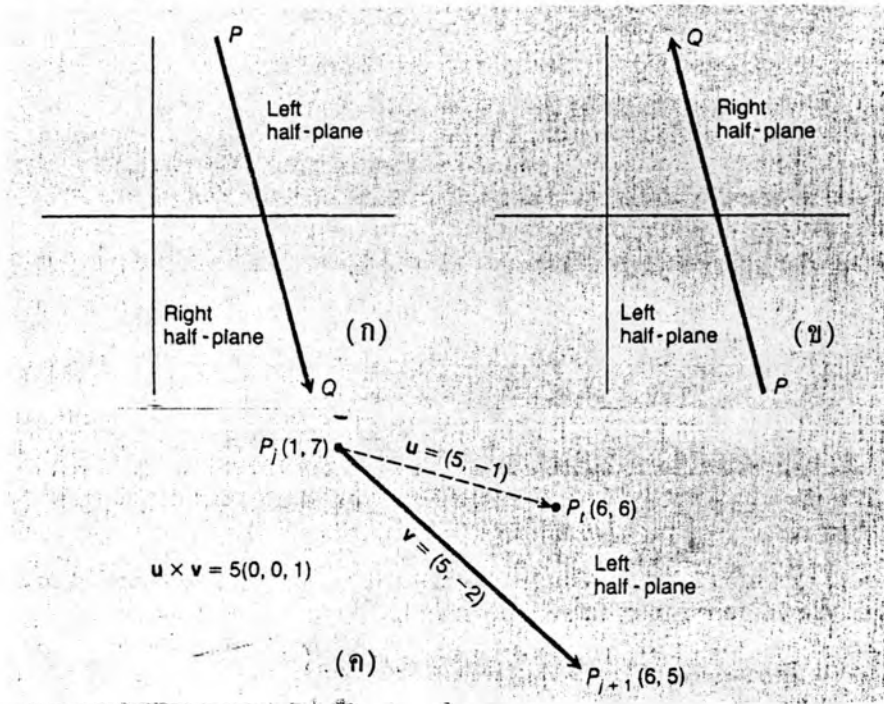
$$u = (e - a, f - b, 0) \text{ และ } v = (c - a, d - b, 0)$$

$$u \times v = (0, 0, (e-a)(d-b) - (f-b)(c-a))$$

สรุปได้ว่าถ้า  $(e-a)(d-b) > (f-b)(c-a)$  ค่าของ  $u \times v$  จะเป็นบวก แสดงว่าจุด  $P_t$  อยู่ซ้ายมือของเส้น  $P_j P_{j+1}$  และถ้า  $(e-a)(d-b) < (f-b)(c-a)$  แสดงว่าอยู่ขวามือ การที่เราจะทดสอบว่าจุดหนึ่งจะอยู่ในรูปหลายเหลี่ยม  $B$  หรือไม่ ต้องทดสอบจุดนั้นกับทุกด้านของรูป  $B$  ถ้าจุดนั้นจะอยู่ในรูป  $B$  ผลการทดสอบต้องแสดงว่าจุดนั้นอยู่ซ้ายมือของทุกด้านของรูป  $B$  ไหระวังไว้วา การเรียงลำดับเลขที่มุมของรูป  $B$  นั้น ต้องเรียงทวนเข็มนาฬิกา

ตัวอย่างเช่น ด้าน  $P_1 P_2$  ของรูปที่ 4.36 (ข) นำมาแสดงในรูปที่ 4.37 (ข) จะ  
 ได้  $u \times v = (5, -2, 0) \times (5, -1, 0) = (0, 0, 5) = 5 (0, 0, 1)$  แสดงว่าจุด  $P_t$  อยู่

ซ้ายมือของคาน  $P_1 P_2$



รูปที่ 4.37 Containment Test โดยวิธีครึ่งของระนาบ

4.3.2.4 Depth Test จากสมการของระนาบใด ๆ คือ

$$Ax + By + Cz + D = 0$$

ถ้าเราทราบค่าพิกัดของจุดมุมของระนาบนั้น 3 จุด เราสามารถสร้างเป็นเวกเตอร์  $u$  และ  $v$  ได้ ดังนี้

$$u = P_1P_2 = (x_2, y_2, z_2) - (x_1, y_1, z_1) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

$$v = P_1P_3 = (x_3, y_3, z_3) - (x_1, y_1, z_1) = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$$

จากหลักทางคณิตศาสตร์ Cross Product  $u \times v$  จะให้ค่าสัมประสิทธิ์ของสมการของระนาบ นั้น คือ  $(A, B, C) = u \times v$  ดังนั้นเราสามารถได้ว่า

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$

$$D = -Ax_1 - By_1 - Cz_1$$

เมื่อใดก็ตามที่ประสิทธิผลของสมการของรูปสองรูปที่จะนำมาเปรียบเทียบกันแล้วขั้นต่อไป จะหาว่าที่จุดตัดในระนาบ  $x-y$  ของรูปทั้งสอง (อาจตัดกันหลายจุดแต่ใช้เพียงจุดเดียวก็พอ) นั้น รูปใดจะอยู่ใกล้จุดมองมากกว่า โดยแทนค่าพิกัดของจุดตัด  $(x,y)$  ลงในสมการของแต่ละรูป จะหาค่า  $z$  ของแต่ละรูปได้ รูปใดที่มีค่า  $z$  มากกว่า แสดงว่ารูปนั้นจะถูกบัง

#### 4.4 สรุป

ในบทที่ 4 นี้ ได้กล่าวถึงทฤษฎีที่ใช้ในการสร้างภาพ 2 มิติ ซึ่งในงานด้านวิศวกรรมการทาง เราจะใช้สร้างภาพของรูปตัดขวาง (Cross Section) รูปแสดงระดับดินเดิม (Profile) ฯลฯ และได้กล่าวถึงทฤษฎีที่ใช้ในการสร้างภาพ 3 มิติ ซึ่ง เราสามารถใช้สร้างภาพเหมือนของจริงของโครงการเพื่อตรวจสอบความถูกต้องและเหมาะสม นอกจากนี้ยังใช้ช่วยในการแนะนำโครงการต่อผู้บริหารและสาธารณะชนอีกด้วย สำหรับการกำจัดเส้นและพื้นผิวที่ถูกบังเป็นการช่วยให้ภาพ 3 มิติที่สร้างขึ้นนั้นดูสมจริงมากขึ้น สรุปได้ว่าทฤษฎีที่กล่าวในบทนี้ เป็นหลักการพื้นฐานที่ใช้ในการแสดงผลของงานด้านวิศวกรรมการทาง โดยภาพต่อไป