Lecture 11-12 3-D Geometric Transformations and Viewing (Part I)

Paramate Horkaew

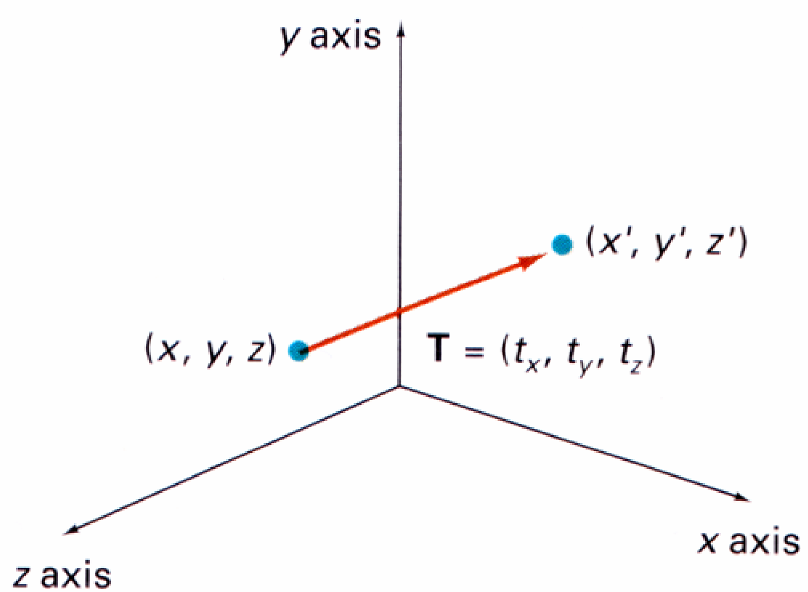
School of Computer Engineering, Institute of Engineering
Suranaree University of Technology

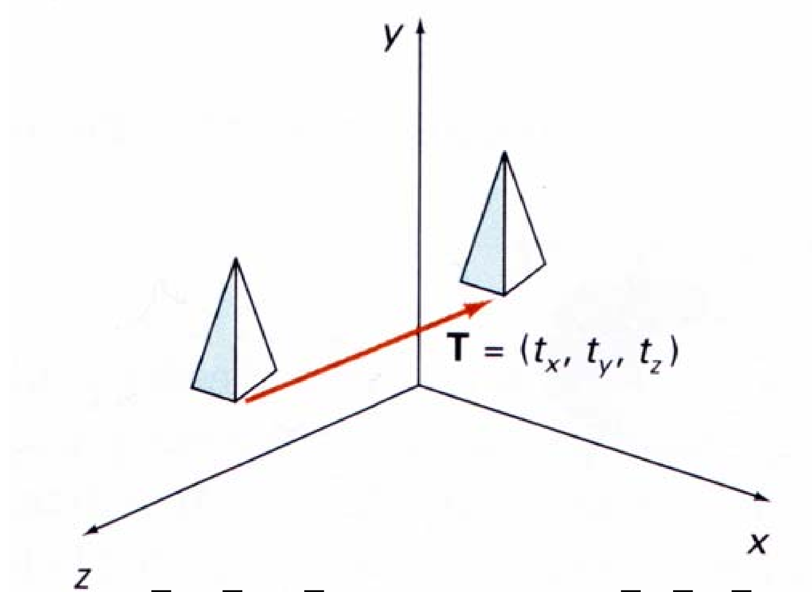
Previous Outline

- 3D Geometric Transformation
  - Basic Translations and Rotations
  - General Rotation about Arbitrary Axis
- 3D Viewing
  - Viewing Pipeline and Viewing Coordinates
  - Coordinates Transformation
- Projections
  - Parallel Projections (Orthogonal and Oblique)
  - Depth Cueing
  - Perspective Projections
- Viewing Volumes
- General Projections

3D Translation

เช่นเดียวกับกรณี 2 มิติ การเลื่อนในพิกัด 3 มิติใดๆ จะมีผลเท่ากันทั้งกับ จุด (รูปด้านซ้าย) และ วัตถุทั้งชิ้น (รูปด้านขวา)





$$x' = x + t_x$$

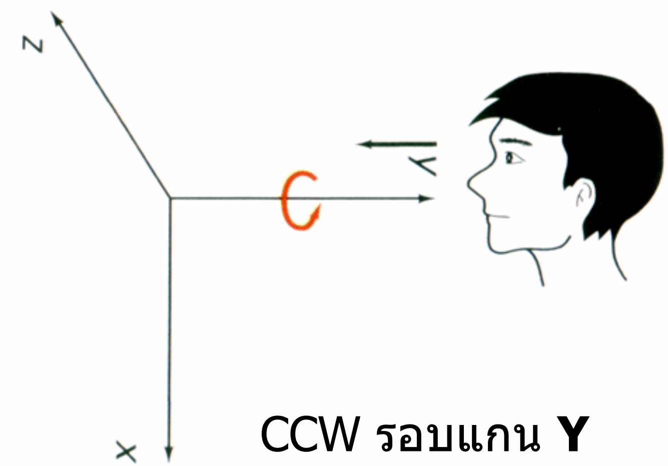
$$y' = y + t_y$$

$$z' = z + t_z$$

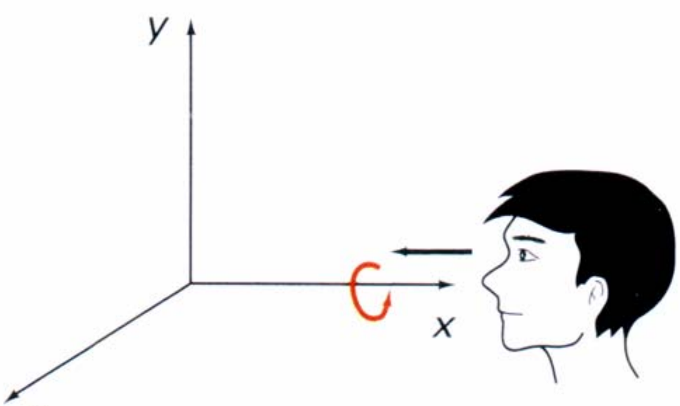
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Rotations

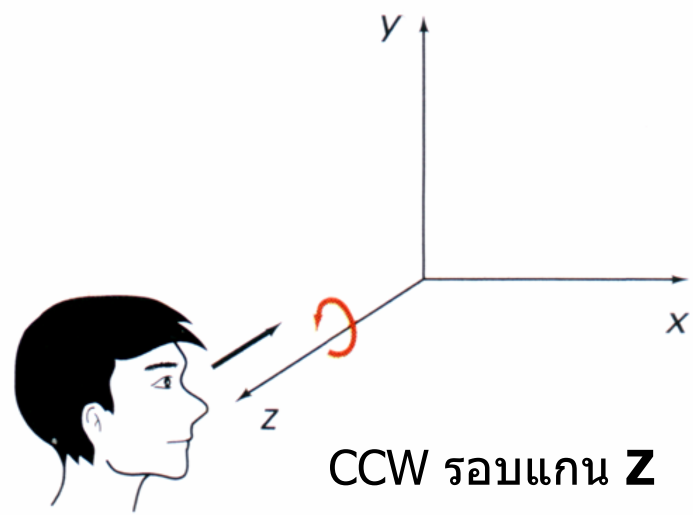
สำหรับการหมุนในพิกัด 3 มิติใด จะต้องกำหนด แกนอ้างอิง สำหรับการหมุนโดยที่ถ้าเรามองจาก ฟากที่เป็นบวกของแกนนั้น ไปยังจุดกำเนิด <u>การหมุนที่มีมุมเป็นบวก</u> จะหมายถึง <u>การหมุนทวนเข็มนาฬิกา</u> (CCW) รอบแกนนั้น ดังรูป



CCW รอบแกน **Y**



CCW รอบแกน **X**



CCW รอบแกน **Z**

3D Rotations about Z, X, Y Axes

Matrix การหมุนรอบแกน Z สามารถสร้างได้จากขยาย Matrix การหมุนใน 2 มิติ โดยเพิ่มแถวที่เป็นของแกน Z

Rotation about Z Axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about X Axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about Y Axis

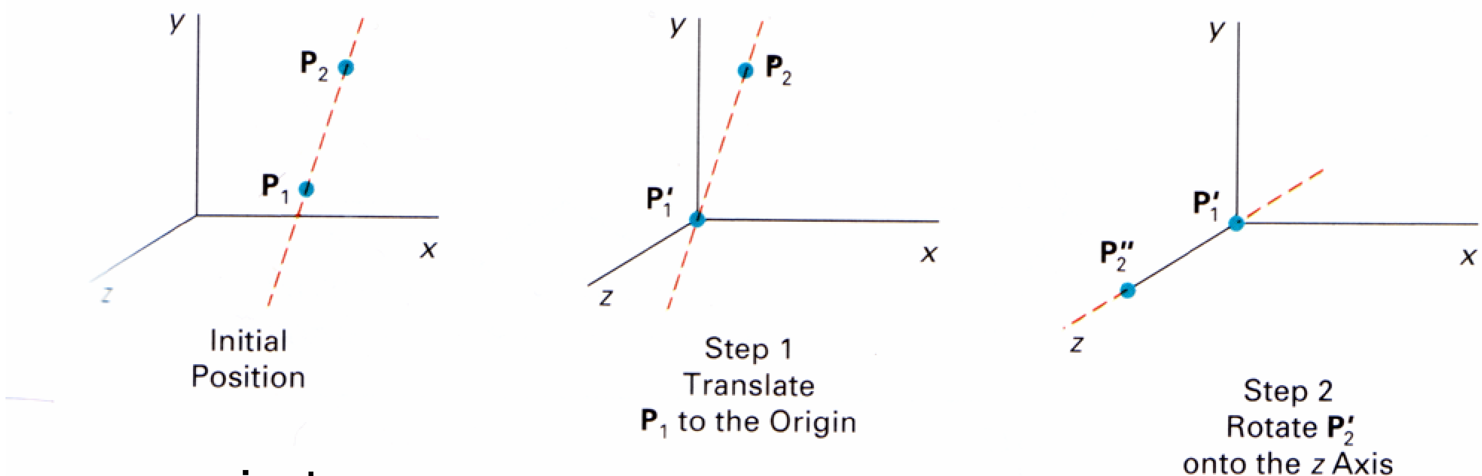
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

สำหรับการหมุนรอบแกน X และ Y ทำได้โดยหมุนเปลี่ยนพิกัดแบบวนรอบ

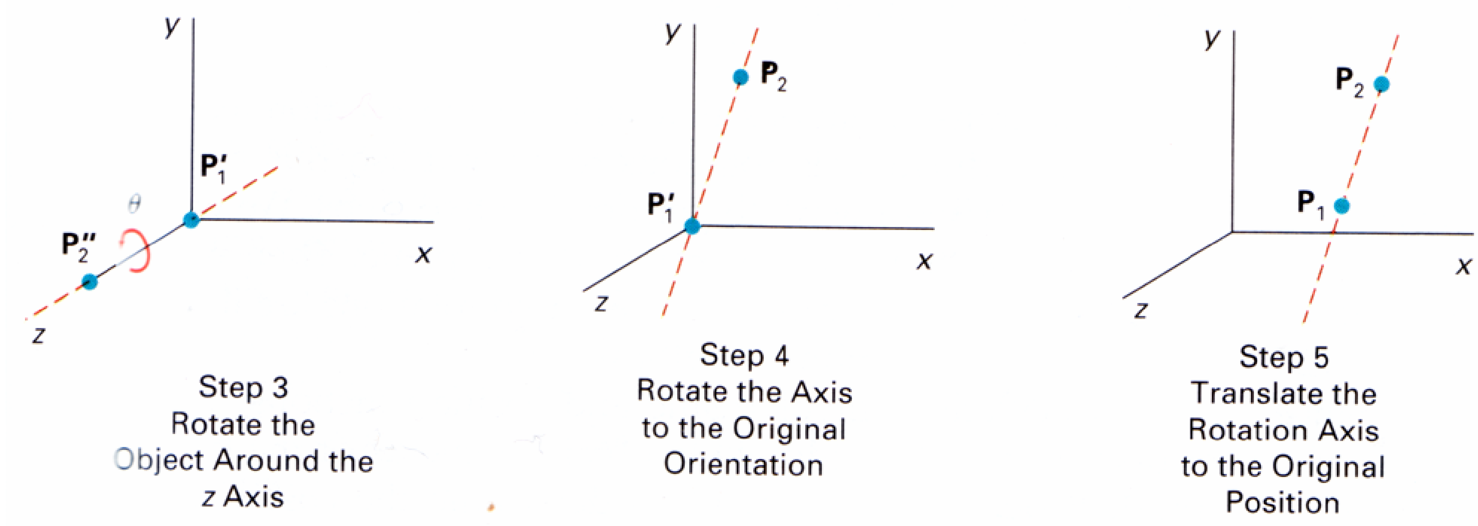
$$X \rightarrow Y \rightarrow Z$$

Suranaree University
of Technology

General 3D Rotations



ดูขั้นตอนบน projector



Lecture 12 3-D Viewing (Part II)

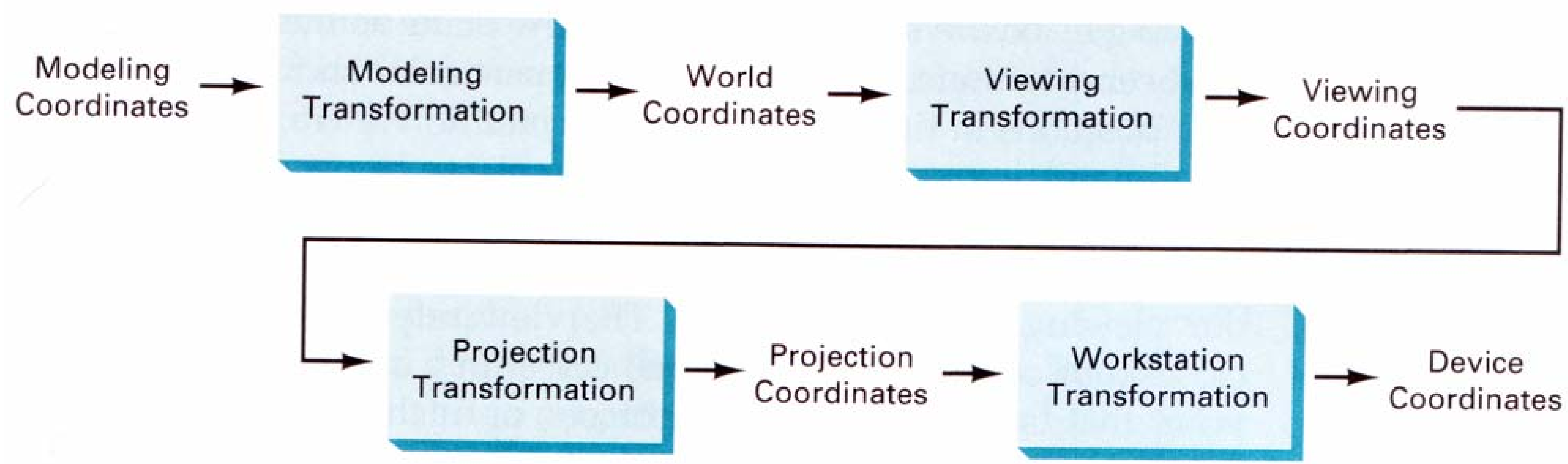
Paramate Horkaew

School of Computer Engineering, Institute of Engineering
Suranaree University of Technology

3D Viewing Pipeline

การแสดงผลใน 3D เปรียบเทียบได้กับ การถ่ายภาพด้วยกล้อง

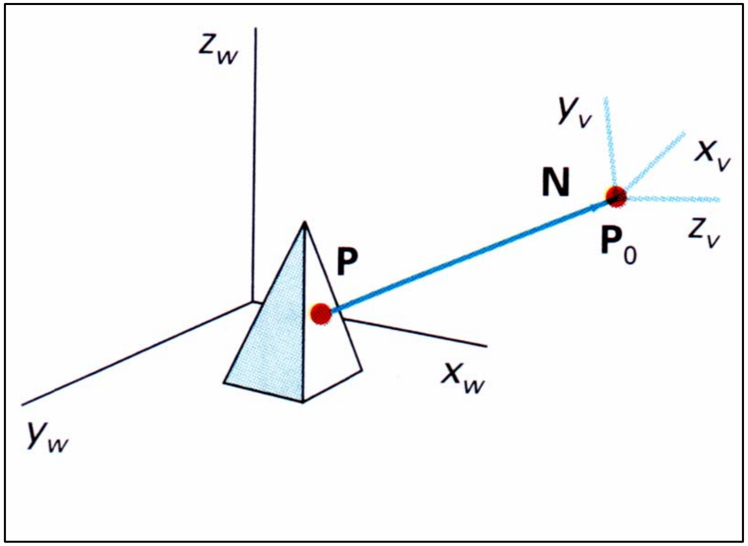
- จัดตำแหน่งกล้องถ่ายภาพ ที่ตำแหน่งในปริภูมิ และ ปรับมุมเอียงของกล้องตามต้องการ (เช่น ระบุทิศทางด้านบน มุมที่ช่องมองทำกับระนาบ ฯลฯ)
- เล็งกล้องไปยังฉากที่ต้องการ แล้วเก็บภาพ ซึ่งจะปรากฏอยู่ภายในช่องสี่เหลี่ยมที่เรียกว่า บริเวณรับแสง (Aperture)
- ภาพที่ได้จึงไปปรากฏบนฟิล์มหรืออุปกรณ์รับแสงแบบ CCD



Viewing Coordinates

การถ่ายภาพด้วยกล้อง จึงกำหนดโดย องค์ประกอบ 3 ประการ ได้แก่ 1) ตำแหน่งของกล้องเทียบกับฉาก, 2) การวางตัวของกล้อง และ 3) ขนาดของช่องรับแสง

ในการแสดงผลแบบ 3D ก็เช่นกัน จำเป็นต้องกำหนด องค์ประกอบดังกล่าวตามรูป



โดยที่ ตำแหน่ง และ การวางตัวของกล้อง กำหนดด้วยตำแหน่ง P0 และ แกนหลักของ View Reference/ Viewing Coordinates System (x_v, y_v, z_v)

ระนาบของช่องรับแสง กำหนดโดย ระนาบ (x_v, y_v) ซึ่งตั้งฉากกับแกน (z_v) ทั้งนี้ทิศด้านบนของภาพที่ปรากฏจะขนานกับ (y_v)

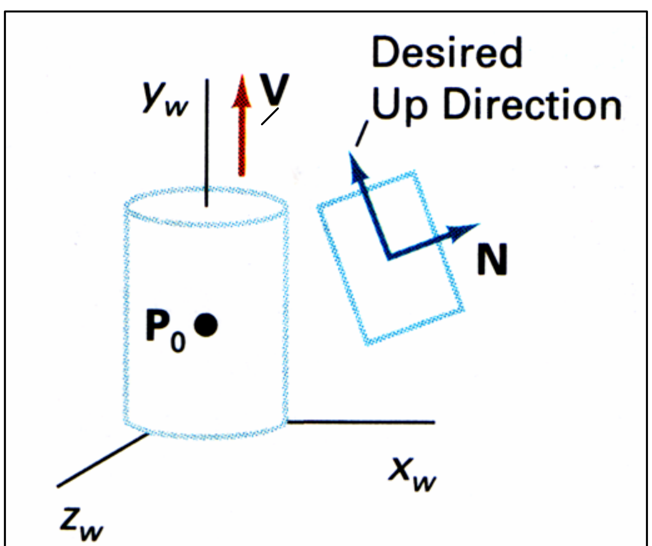
ดังนั้นการแสดงผลใน 3D คือ "การคำนวณหา Transformation ที่แปลงพิกัดจุดในปริภูมิกายภาพ (x_w, y_w, z_w) ให้เป็นพิกัดบนฉาก (x_v, y_v, z_v) นั่นเอง"

Suranaree University of Technology

Defining Viewing Plane

ขั้นตอนถัดไป คือนิยามระนาบการมอง (ระนาบของฉากรับภาพ x_v, y_v) ซึ่งกำหนด โดย View Up Vector (**V**) และ View Right Vector (**U**) ที่ขนานกับ vector y_v, x_v ตามลำดับ และ มีขนาด **1 หน่วย** โดยมีขั้นตอนต่อไปนี้

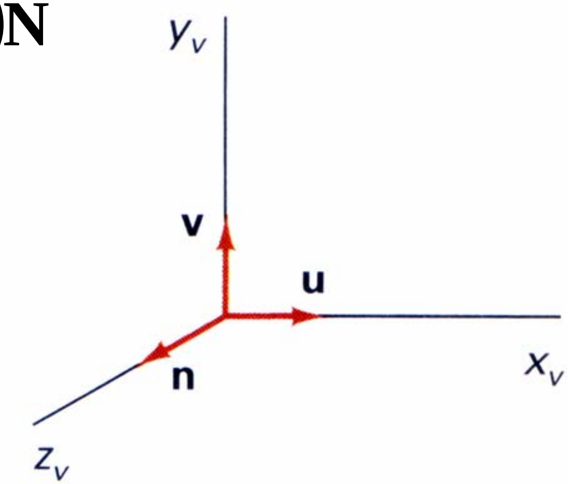
ประมาณ vector **V'** ในทิศทางที่ต้องการให้เป็นชี้ขึ้นของภาพ โดยที่ต้อง ไม่ขนาน กับ **N** แล้วคำนวณ vector **V** ที่ตั้งฉากกับ **N** แล้วจึง คำนวณ **U** ที่ตั้งฉากกับ vector **N** และ **V** และ ขนานกับ vector x_v



$$\mathbf{V}'' = \mathbf{V}' - (\mathbf{V}' \cdot \mathbf{N})\mathbf{N}$$

$$\mathbf{V} = \frac{\mathbf{V}''}{\|\mathbf{V}''\|}$$

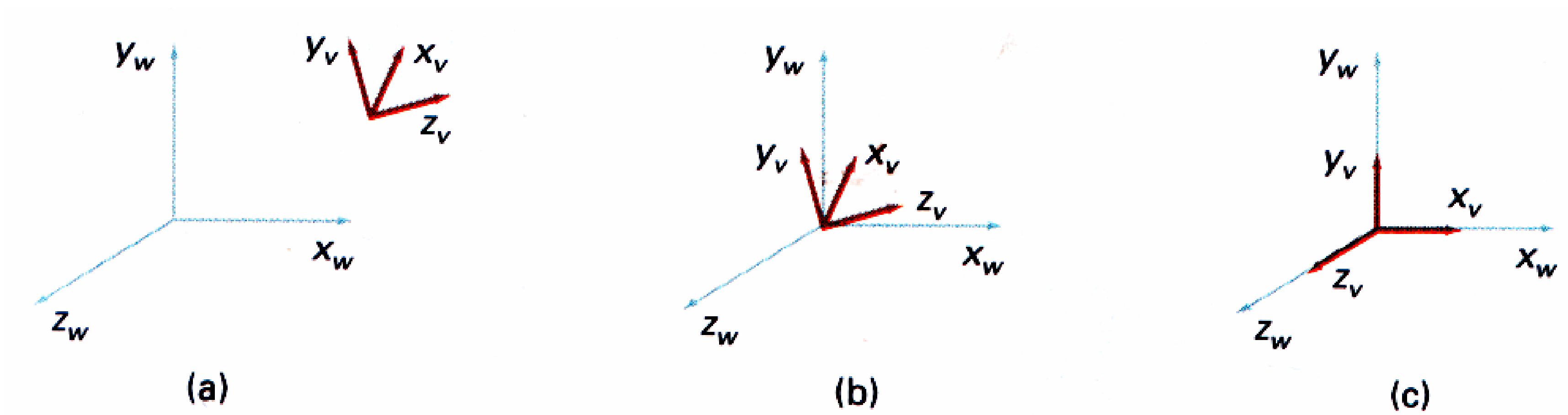
$$\mathbf{U} = \frac{\mathbf{V} \times \mathbf{N}}{\|\mathbf{V} \times \mathbf{N}\|}$$



Suranaree University of Technology

Coordinates Transformation

เมื่อกำหนด ปริภูมิกายภาพ (w) และ ปริภูมิการมอง (v) ดังกล่าวแล้ว เราสามารถหาพิกัด ของวัตถุที่นิยามจาก (w) ซึ่งมองเห็นโดยผู้สังเกตจาก กรอบ (v) ได้ โดยทำการคำนวณลำดับ Transformation พื้นฐาน แบบ 3D ดังแสดงตามแผนผัง



1) เลื่อนจุดกำเนิดของกรอบการมอง (v) ไปยังจุดกำเนิดของกรอบ (w) รูป (b)

2) ทำการหมุนแกน ให้ $(x, y, z)_v$ ไปซ้อนทับกับ $(x, y, z)_w$ รูป (c)

Computing T•R Matrix

1) เลื่อนจุดกำเนิดของกรอบการมอง (v) $\mathbf{P}_0 = [x_0 \; y_0 \; z_0]^T$ ไปยังจุดกำเนิด ของกรอบ (w)
2) ทำการหมุนแกน ให้ $(x, y, z)_v$ ไปซ้อนทับกับ $(x, y, z)_w$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow \mathbf{R} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_x & v_x & 0 \\ n_x & n_x & n_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

พิกัด P_v ที่ผู้สังเกตมองวัตถุ ในกรอบ P_w นิยามโดย (ค่า z ในกรอบ v บอก ความลึก ของวัตถุเทียบกับผู้สังเกต)

$$\mathbf{P}_v = \mathbf{M}_{WC,VC} \cdot \mathbf{P}_w = (\mathbf{R} \cdot \mathbf{T}) \cdot \mathbf{P}_w$$

โดยที่

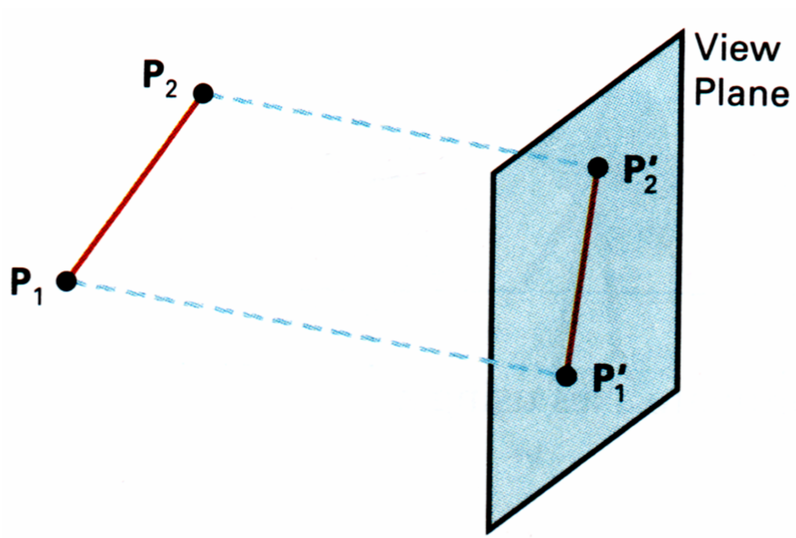
$$\mathbf{U} = [u_x \quad u_y \quad u_z]^T$$

$$\mathbf{V} = [v_x \quad v_y \quad v_z]^T$$

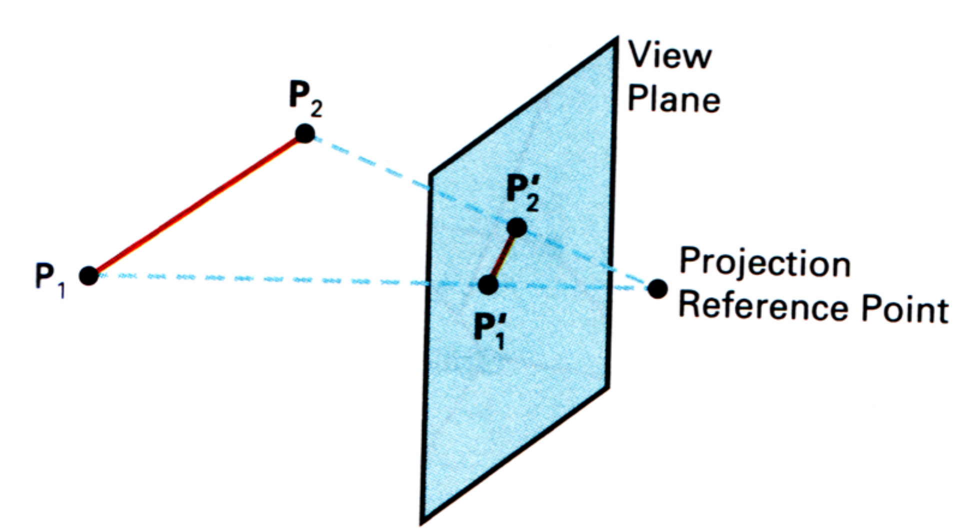
$$\mathbf{N} = [n_x \quad n_y \quad n_z]^T$$

Projections

จากขั้นตอนที่แล้ว เราสามารถแสดงพิกัดของวัตถุในกรอบ w (world) ในรูปของพิกัด ในกรอบ v ได้ โดยที่ ค่าพิกัด z ที่คำนวณได้บอกถึง **ความลึก** ของวัตถุเทียบกับผู้สังเกต ซึ่ง **ภาพฉาย** ของวัตถุ (กำหนดด้วยพิกัด P_v) ที่ปรากฏบนฉากรับภาพ จะหาได้ 2 วิธี (เป็นฟังก์ชันของ z) ได้แก่



Parallel Projection
เส้นฉายขนานกันทุกจุด จากวัตถุถึงฉาก
แบ่งเป็น **Orthographic** และ **Oblique**

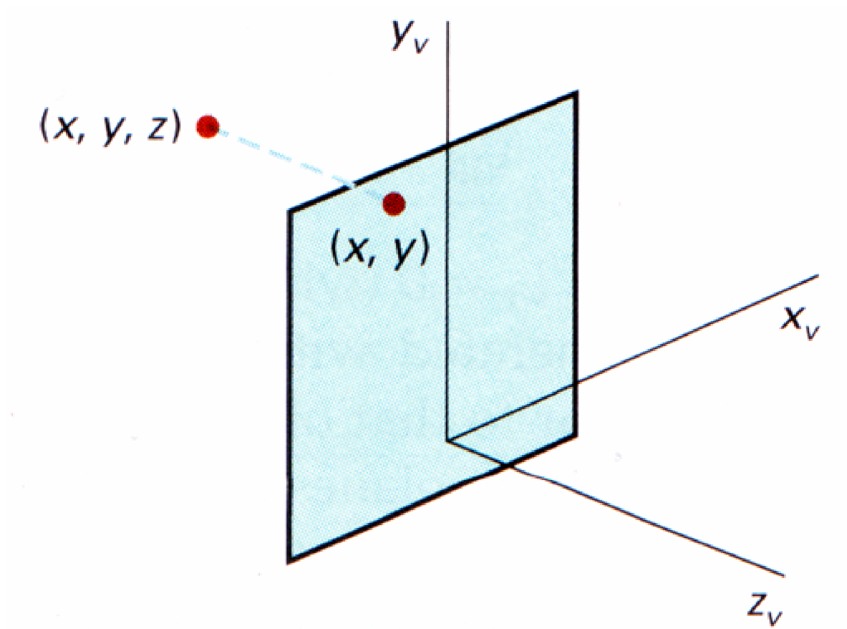


Perspective Projection (เหมือนจริง)
เส้นฉายลู่เข้าไปหาจุดอ้างอิงที่หลังฉาก

Suranaree University of Technology

Orthographic Projections

การฉายวัตถุลงบนฉากแบบ Orthographic Projection นั้น เราสามารถแปลงพิกัด วัตถุที่อยู่ในกรอบ v เป็นพิกัด ในระนาบได้อย่างตรงไปตรงมา



$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

หรือ $x_p = x_v \quad y_p = y_v$

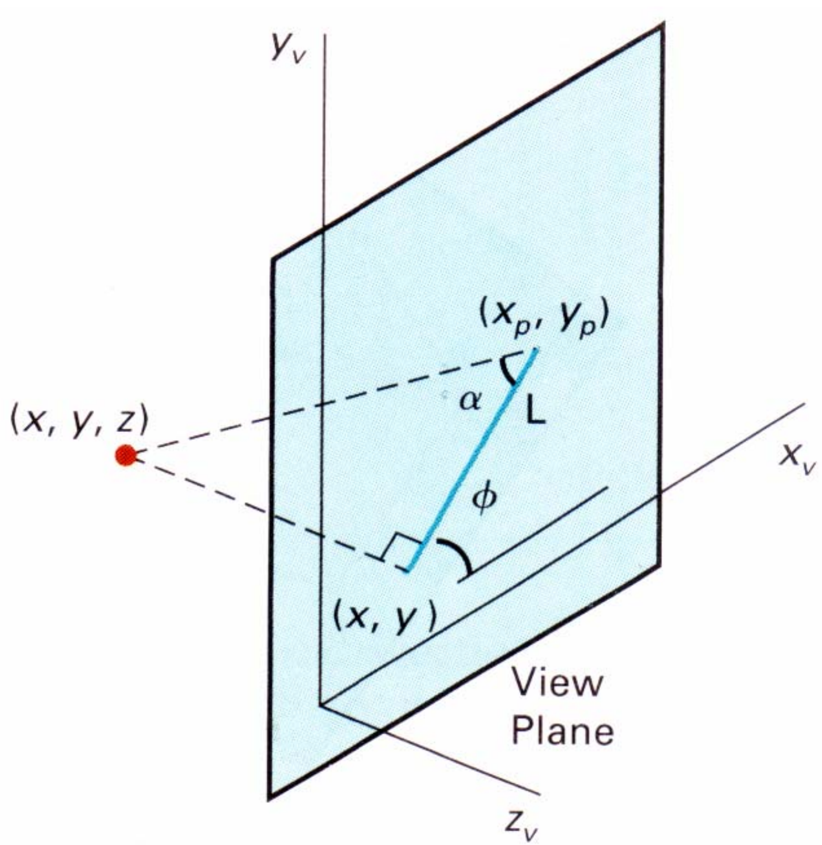
$$z_p = 0$$

ในที่นี้ค่า z_v อาจจะใช้สำหรับวัดความลึก โดยที่ ถ้าวัตถุอยู่ลึกมาก (z ขนาดใหญ่) ภาพที่ปรากฏก็จะมีสีจางลง (หรือมัว) กว่าภาพที่อยู่ใกล้กว่า (z ขนาดเล็ก)

Suranaree University of Technology

Oblique Projections

การฉายวัตถุลงบนฉากแบบ Oblique Projection นั้น คือการฉายเงา ที่ขนานกัน จากทุกๆ จุดบนวัตถุ แต่เส้นฉาย **ไม่ตั้งฉาก** กับระนาบการมอง โดยนิยามด้วย ตัวแปร 2 ตัวได้แก่ α และ ϕ โดยที่



α คือมุมที่ ทำระหว่างเส้นตรงที่เชื่อมจุด (x, y, z) และ $(x_p, y_p, 0)$ กับเส้นตรงเชื่อมจุด $(x_p, y_p, 0)$ กับ (x, y, 0)

ϕ คือมุมที่เส้นตรงเชื่อม $(x_p, y_p, 0)$ กับ (x, y, 0) ซึ่งมีความยาว = L ทำกับ แกน x_v

ดังนั้น

$$x_p = x + L\cos\phi$$

$$y_p = y + L\sin\phi$$

$$L = \frac{z}{\tan\alpha}$$

Oblique Projections Matrix

จากระบบสมการ สำหรับการแปลงพิกัด ในกรอบ (v) เป็นพิกัดบนฉากรับภาพ จัดรูปใหม่โดยให้ $L_1 = 1/(\tan \alpha)$ จะได้ว่า

$$x_p = x + \frac{z}{\tan \alpha} \cdot \cos \phi$$

$$y_p = y + \frac{z}{\tan \alpha} \cdot \sin \phi$$

$$\Downarrow$$

$$x_p = x + L_1 \cdot z \cdot \cos \phi$$

$$y_p = y + L_1 \cdot z \cdot \sin \phi$$

หรือเขียนในรูป Matrix ได้ดังนี้

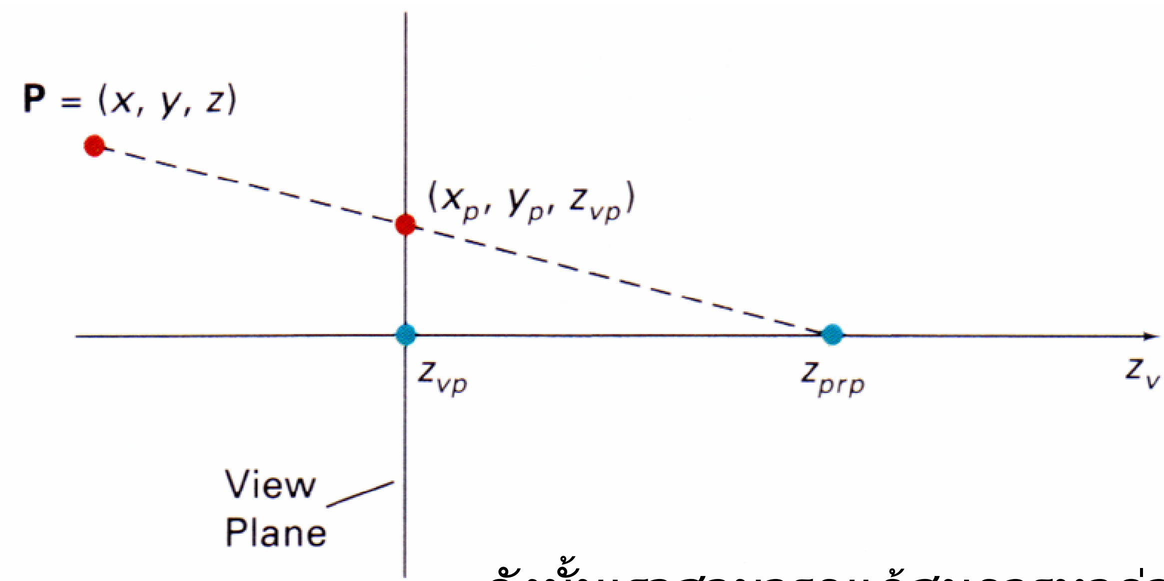
$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L_1 \cos \phi & 0 \\ 0 & 1 & L_1 \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

เมื่อ $L_1 = 0$ จะกลายเป็นการฉาย Orthographic หรือ อีกนัยหนึ่ง คือ มุม α มีค่าเท่ากับ 90 องศา

Perspective Projections

สำหรับการฉายภาพแบบ Perspective จะกำหนดเวกเตอร์ฉาย จากวัตถุ P ซึ่งมีพิกัด (x, y, z) ไปยัง จุดอ้างอิงบนแกน z_v (0, 0, z_{prp}) จุดที่เวกเตอร์นี้ ตัดกับฉากรับภาพ จะเป็นภาพของ จุด P (x_p, y_p, z_p) ดังรูป

จุดใดๆ ที่อยู่บนเวกเตอร์นี้ จะอยู่ในรูปของสมการเส้นตรงนั่นคือ



$$x' = x - ux$$

$$y' = y - uy$$

$$z' = z - u\left(z - z_{prp}\right)$$

ดังนั้นเราสามารถแก้สมการหาค่า u สำหรับพิกัด (x′, y′) บนฉากรับภาพ (x_p, y_p, z_p) โดยกำหนดให้ $z' = z_{vp}$

Perspective Projection Matrix

ดังนั้นที่ $z' = z_{vp}$ จะได้ว่า $u = (z_{vp} - z)/ (z_{prp} - z)$ แทนไปในสมการ

$$x_p = x - \left(\frac{z_{vp} - z}{z_{prp} - z}\right)x = \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)\cdot x = \frac{d_p}{z_{prp} - z}\cdot x$$

$$y_p = y - \left(\frac{z_{vp} - z}{z_{prp} - z}\right)y = \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)\cdot y = \frac{d_p}{z_{prp} - z}\cdot y$$

$$d_p = z_{prp} - z_{vp}$$

หรือในรูป Matrix (Homogeneous Coordinates)

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}\left(z_{prp}/d_p\right) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

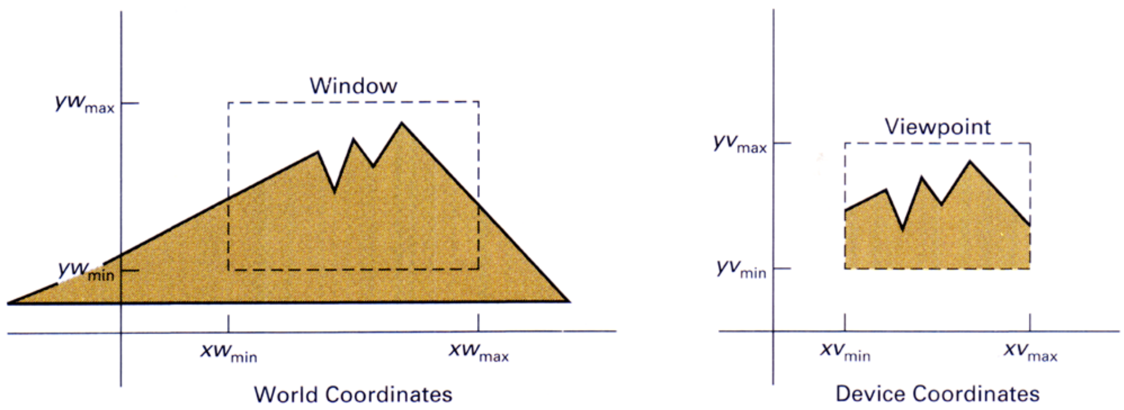
$$h = \frac{z_{prp} - z}{d_p}$$

$$x_p = x_h / h$$

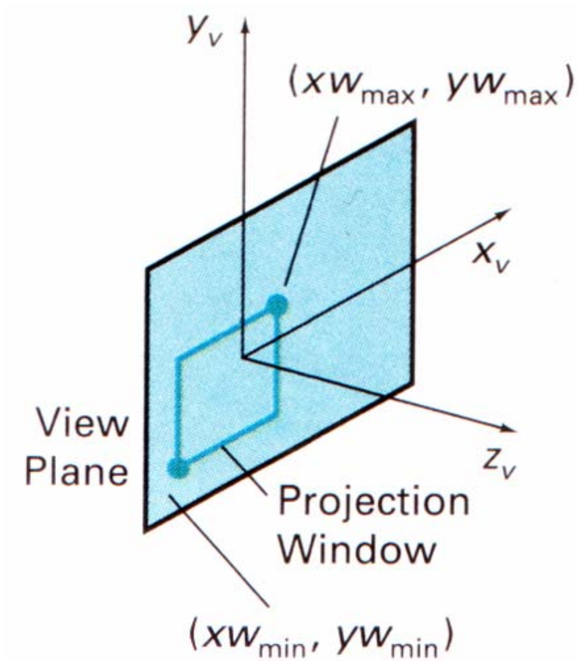
$$y_p = y_h / h$$

View Volumes

เช่นเดียวกับ ในกรณี 2 มิติ บริเวณที่ผู้สังเกต สามารถมองเห็น บนฉากรับภาพ จะถูกจำกัดด้วยบริเวณซึ่งกำหนดโดย Window ใน Word Coordinate ดังรูปซ้ายมือ



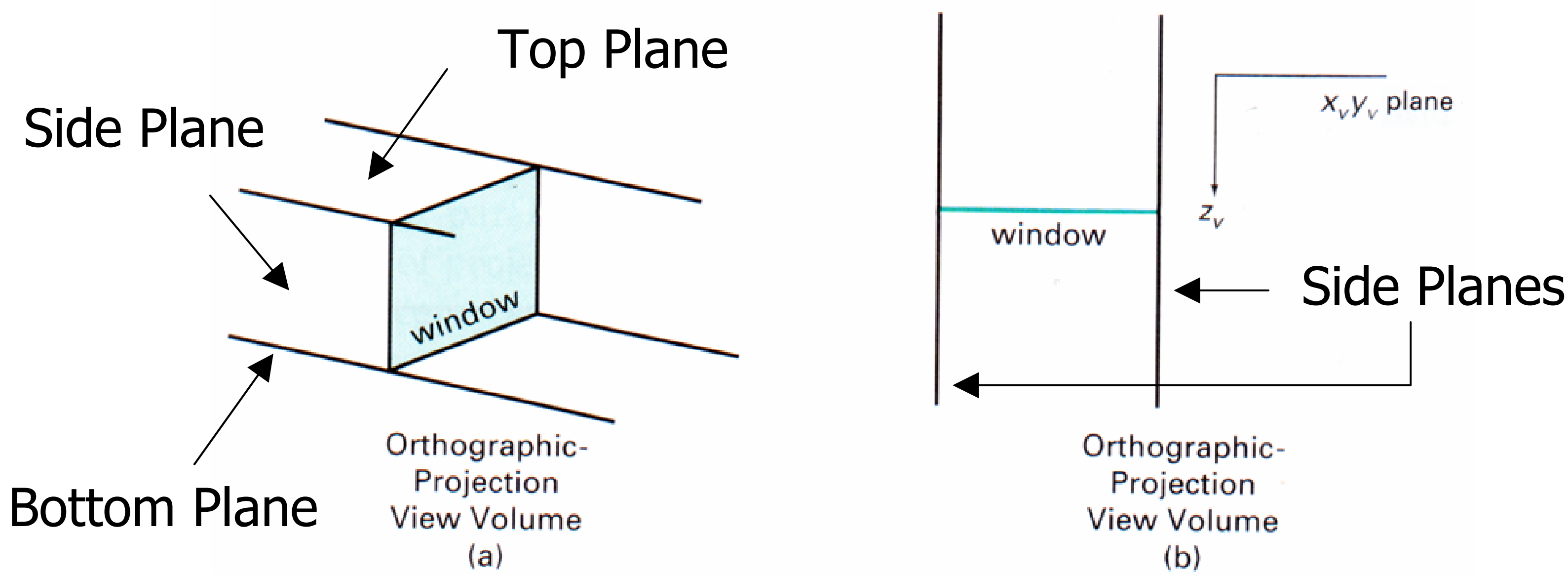
ในกรณี 3 มิติพิกัดของจุดยอดมุม ของ Window ใน Word Coordinate จะถูกแปลงให้อยู่ในรูปของพิกัดในกรอบ ของ Viewing Coordinate แล้วจึงฉายลงบน ฉากรับภาพ ดังรูปขวามือ



<u>วัตถุที่อยู่ภายในบริเวณ **View Volume** เท่านั้นที่จะไปปรากฏบนฉากรับภาพ</u>

Parallel Projection Volume

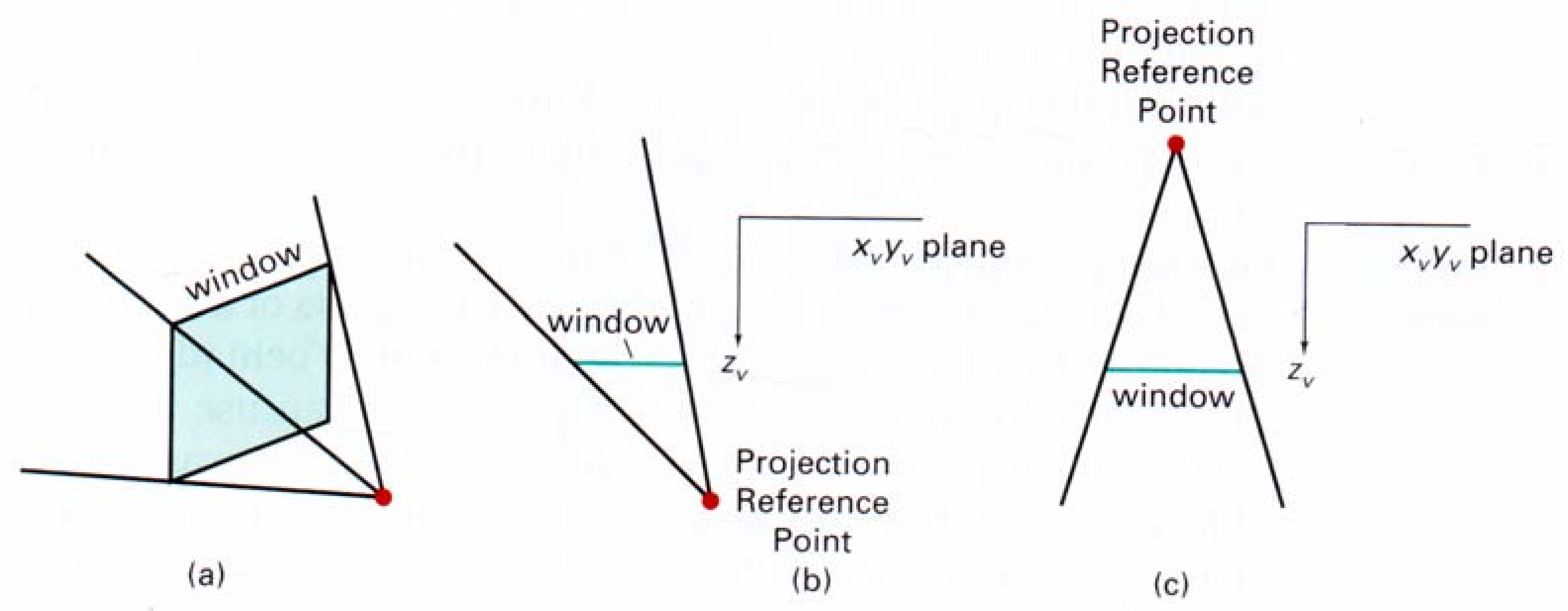
ในกรณีของ การฉายภาพแบบขนาน ขนาดของ View Volume จะขึ้นอยู่กับ ขนาดของกรอบที่กำหนด ใน World Coordinates



รูปทรงของของ Viewing Volume จะเป็น **ท่อปลายเปิด** รูปทรงสี่เหลี่ยมมุมฉาก ใน 3 มิติ ซึ่งมีความยาวเป็นอนันต์ และ แต่ละด้าน (ศรชี้) จะเป็น **ระนาบ** ที่ขนานกัน และ ตัดผ่าน เส้นกรอบ ของ window

Perspective Projection Volume

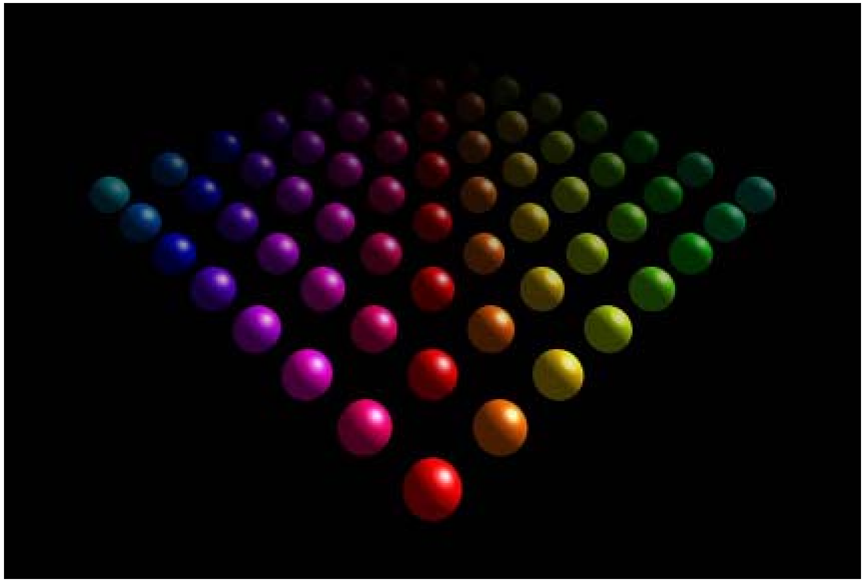
สำหรับกรณีของ การฉายภาพแบบ Perspective นั้น ขนาดของ View Volume จะขึ้นอยู่กับ ขนาดของกรอบที่กำหนด ใน World Coordinates เช่นกัน

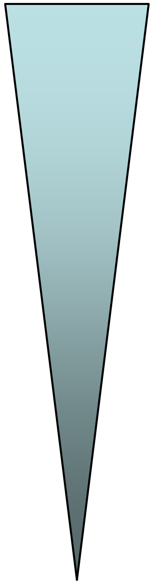


อย่างไรก็ดี รูปทรงของ Viewing Volume ในกรณีนี้ จะเป็นรูป Pyramid ฐานสี่เหลี่ยมที่อยู่ที่ระยะอนันต์ และ ยอดของ Pyramid จะอยู่ที่จุดอ้างอิง (Projection Reference Point)

Depth Cueing (Z-Buffer)

การวิเคราะห์ความลึกของวัตถุ เทียบกับฉาก (หรือผู้สังเกต) นี้ ทำให้ภาพที่ปรากฏมีความเหมือนจริงมากขึ้น ดังตัวอย่างต่อไปนี้

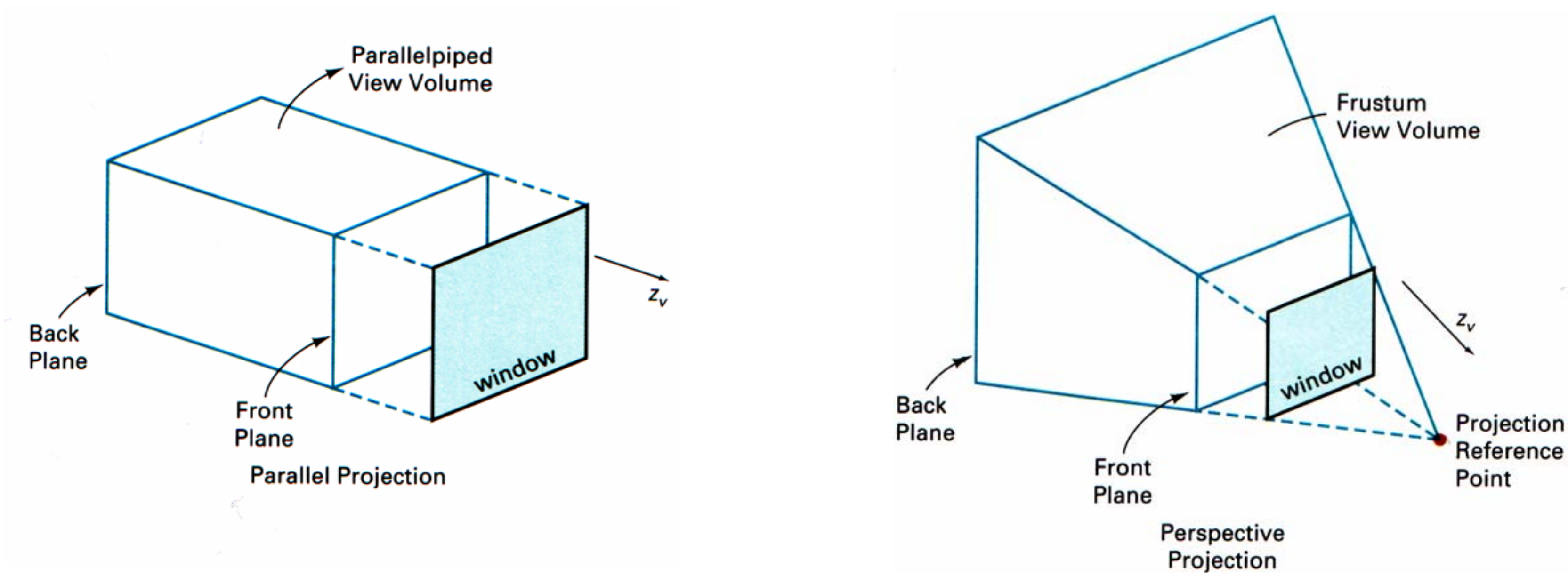




สำหรับการแสดงผลด้วย Polygon พิกัด z จะเรียงอยู่ใน array (Z-Buffer) ตามค่าของ z_v โดยที่ วัตถุที่อยู่ไกลกว่า จะถูกบังด้วยวัตถุที่อยู่ใกล้กว่า เป็นต้น

Front and Back Planes

เนื่องจาก วัตถุที่อยู่ไกลจาก หรือ ใกล้กับ ผู้สังเกตมากๆ (จนไม่สามารถมองเห็นได้) จะไม่ปรากฏบนจอภาพ ดังนั้น เราอาจใช้เทคนิคการพิจารณา Depth Cueing เพื่อกำหนดพิสัยของ View Volume โดยนิยาม Back (Far) Plane และ Front (Near) Plane จาก พิกัด z_v (พิกัด z_w ที่แปลงให้อยู่ใน View Coordinates)



View Volume ที่ได้จะเป็น กล่องสี่เหลี่ยม สำหรับการฉายแบบขนาน และ เป็น Frustum (ปิรามิดหัวตัด) สำหรับการฉายแบบ Perspective

Suranaree University of Technology

Projection Summary

ลักษณะจำเพาะของการฉายภาพแต่ละแบบสรุปได้ดังนี้

สำหรับการฉายแบบ Orthographic ตำแหน่งของ View Volume เทียบกับ ฉากรับภาพเทียบ จะไม่มีผลต่อ ภาพของวัตถุ นอกจากตัดส่วนที่เกินออกไป

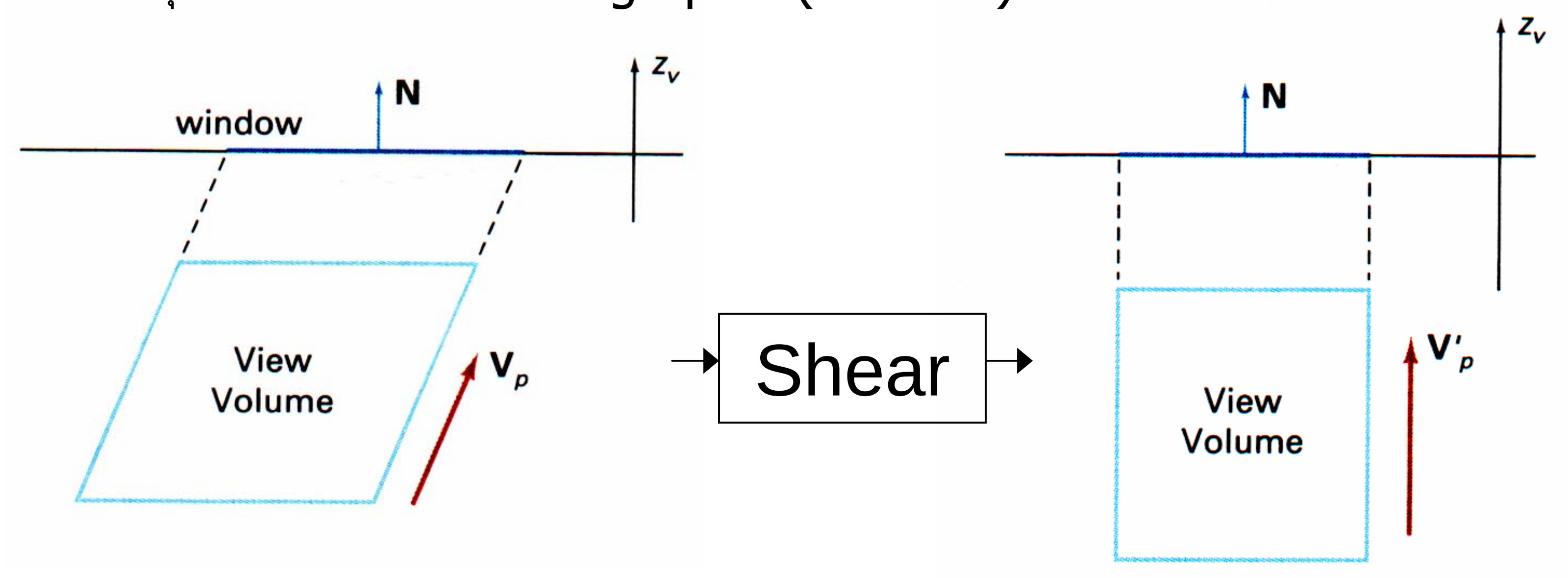
สำหรับการฉายแบบ Oblique Project ตำแหน่งดังกล่าวจะมีผลต่อภาพ โดยที่ถ้า View Volume เบนจากฉากรับภาพมาก เงาของวัตถุจะเอนมาก

สำหรับการฉายแบบ Perspective ขนาดของภาพวัตถุจะมีขนาด เล็กลงเมื่อวัตถุอยู่ไกลจากฉากรับภาพมาก และ มีขนาดโตเมื่อวัตถุอยู่ใกล้กับฉาก

ระยะของจุดอ้างอิง (Focus) ของ Perspective เทียบกับฉากรับภาพจะมีผลต่อความแตกต่างของขนาด ระหว่างวัตถุที่อยู่ใกล้และไกล จากฉาก นั่นคือ สำหรับวัตถุ 2 ชิ้น ที่อยู่ห่างจากฉากเป็นระยะ D1 และ D2 ถ้าจุดนี้อยู่ใกล้ *ขนาดของเงาวัตถุจะต่างกันมาก* แต่ถ้าจุดนี้อยู่ไกล *ขนาดของเงาวัตถุจะต่างกันน้อย* ถ้าจุดนี้อยู่ที่ระยะอนันต์ *ขนาดของวัตถุจะไม่ต่างกัน* นั่นคือ กลายเป็นการฉายแบบขนาน

General Parallel Projections

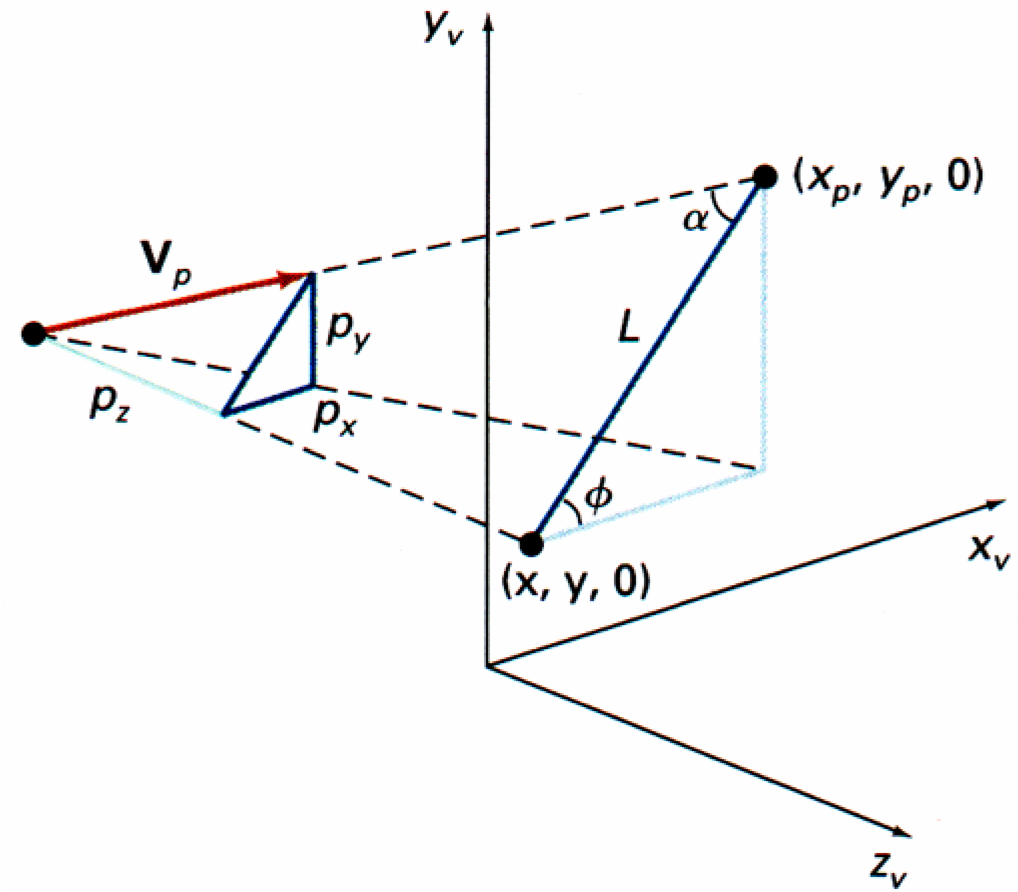
เราทราบแล้วว่า การฉายแบบ Orthographic Projection $(x_p, y_p) = (x_v, y_v)$ ดังนั้น เราสามารถหา Matrix การแปลงกรณีทั่วไป สำหรับการฉายแบบ Oblique และ Orthographic ได้ โดยที่ สำหรับการฉาย แบบ Oblique เราจะแปลง View Volume รูปสี่เหลี่ยมด้านขนาน (ด้านขวา) โดยใช้การเฉือน (Shear) ให้เป็นรูปสี่เหลี่ยมมุมฉาก เหมือน Orthographic (ด้านซ้าย)



รูปแบบการฉายจะกำหนดด้วย vector $\mathbf{V}_p$ ดังนั้นถ้า $\mathbf{V}_p$ ไม่ตั้งฉากกับ N (Oblique) เราจะแปลงด้วยการเฉือนก่อน ให้เป็น $\mathbf{V}'_p$ ซึ่งทำให้ $(x_p, y_p) = (x'_v, y'_v)$

General Parallel Projections

ดังนั้น Matrix M ที่ใช้แปลง (x_v, y_v, z_v) เป็น (x_p, y_p, z_p) เมื่อกำหนด vector $\mathbf{V}_p = [p_x, p_y, p_z]^T$ ให้ กับเมื่อกำหนด มุม alpha, phi ให้มีค่าเท่ากัน

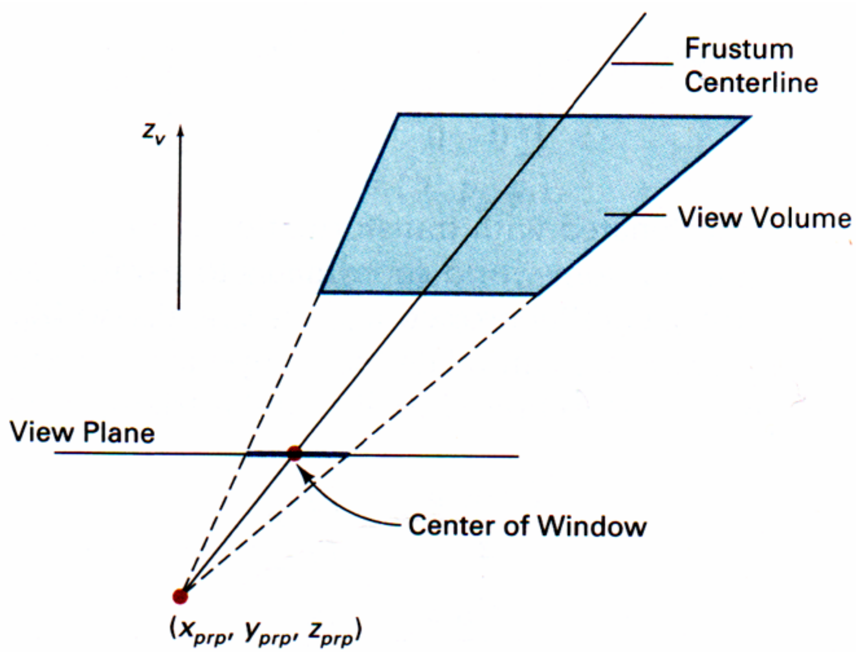


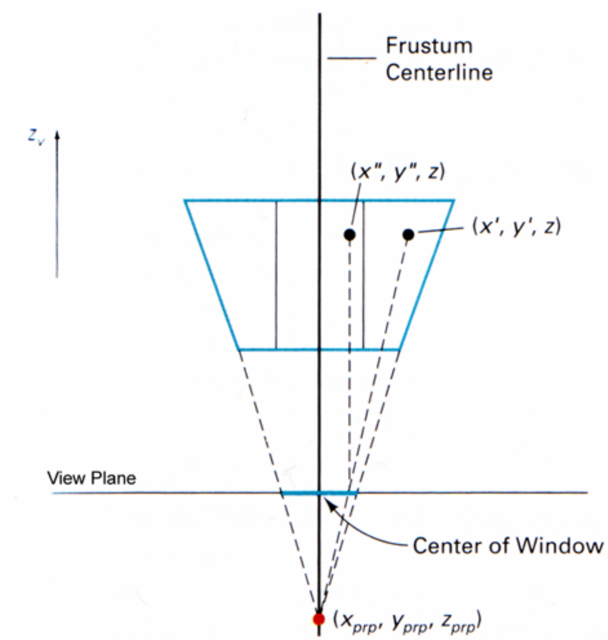
$$\mathbf{M}_p = \begin{bmatrix} 1 & 0 & \dfrac{\cos\phi}{\tan\alpha} & 0 \\ 0 & 1 & \dfrac{\sin\phi}{\tan\alpha} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -p_x/p_z & 0 \\ 0 & 1 & -p_y/p_z & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

General Perspective Projections

ในทำนองเดียวกัน การฉายโดยทั่วไปสำหรับกรณีการฉายแบบ Perspective จะกำหนดด้วย vector ที่นิยามแกนกลางของ Pyramid หัวตัด (Frustum) ภาพที่ปรากฏบนฉาก จะเป็นเงาของวัตถุที่อยู่ในบริเวณที่แรเงา (ดังรูปด้านซ้าย)





การเบี่ยงเบนของ Frustum กำหนดโดย $(x_{prp}, y_{prp}, z_{prp})$ โดยที่ x_{prp}, y_{prp} ไม่เท่ากับ 0 ซึ่งต้องหา Matrix เฉือน **M** ที่ทำให้ จุดกึ่งกลางฉากใหม่ = x_{prp}, y_{prp}

General Perspective Projections

Matrix การเฉือน

$$\mathbf{M}_{shear} = \begin{bmatrix} 1 & 0 & a & -az_{prp} \\ 0 & 1 & b & -bz_{prp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ดังนั้น

$$x' = x + a \cdot \left(z - z_{prp}\right)$$
$$y' = y + b \cdot \left(z - z_{prp}\right)$$

เมื่อ

$$a = -\frac{x_{prp} - 0.5 \cdot \left(xw_{\max} - xw_{\min}\right)}{z_{prp}} \qquad b = -\frac{y_{prp} - 0.5 \cdot \left(yw_{\max} - yw_{\min}\right)}{z_{prp}}$$

หมายความว่า

- ถ้าจุดกึ่งกลางของ Frustum อยู่ห่างจาก จุดอ้างอิงมาก ก็เฉือนมาก
- ถ้าจุดอ้างอิงอยู่ใกล้กับฉากรับภาพมาก ก็เฉือนมาก เช่นกัน

Standard Perspective Projection

ดังนั้นที่ $z' = z_{vp}$ จะได้ว่า $u = (z_{vp} - z)/(z_{prp} - z)$ แทนไปในสมการ

$$\left.\begin{aligned} x_p &= x - \left(\frac{z_{vp} - z}{z_{prp} - z}\right)x = \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)\cdot x = \frac{d_p}{z_{prp} - z}\cdot x \\ y_p &= y - \left(\frac{z_{vp} - z}{z_{prp} - z}\right)y = \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)\cdot y = \frac{d_p}{z_{prp} - z}\cdot y \end{aligned}\right\} d_p = z_{prp} - z_{vp}$$

หรือในรูป Matrix (Homogeneous Coordinates)

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}\left(z_{prp}/d_p\right) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \boxed{\begin{aligned} h &= \frac{z_{prp} - z}{d_p} \\ x_p &= x_h/h \\ y_p &= y_h/h \end{aligned}}$$

General Perspective Projections

ขั้นตอนถัดไป คือทำการแปลงพิกัดของ $(x, y, z)'_v$ ที่ได้ไปเป็น $(x, y, z)''_p$ โดยพิสูจน์เช่นเดียว กันกับกรณีมาตรฐาน

$$\left(x''_p - x_{prp}\right) = \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)\left(x'_v - x_{prp}\right)$$

$$\left(y''_p - y_{prp}\right) = \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)\left(y'_v - y_{prp}\right)$$

$$x''_p = x_{prp} + \left(\frac{z_{prp}}{z_{prp} - z}\right)\left(x'_v - x_{prp}\right)$$

$$y''_p = y_{prp} + \left(\frac{z_{prp}}{z_{prp} - z}\right)\left(y'_v - y_{prp}\right)$$

โดยที่ถ้า $z_{vp} = 0$ เราจะได้ Matrix ที่ทำการแปลงพิกัดในกรอบ v เป็นพิกัดบนฉากคือ

$$\mathbf{M}_{scale} = \begin{bmatrix} 1 & 0 & -x_{prp}/z_{prp} & 0 \\ 0 & 1 & -y_{prp}/z_{prp} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/z_{prp} & 1 \end{bmatrix}$$

$$M'_{perspective} = M_{scale} \bullet M_{shear}$$

(*อย่าลืมแปลงพิกัด Homogeneous ด้วย*)

Conclusions

- 3D Geometric Transformation
  - Basic Translations and Rotations
  - General Rotation about Arbitrary Axis
- 3D Viewing
  - Viewing Pipeline and Viewing Coordinates
  - Coordinates Transformation
- Projections
  - Parallel Projections (Orthogonal and Oblique)
  - Depth Cueing
  - Perspective Projections
- **Viewing Volumes**
- **General Projections**