2007학년도 1학기

3D 그래픽스를 위한 기초이론

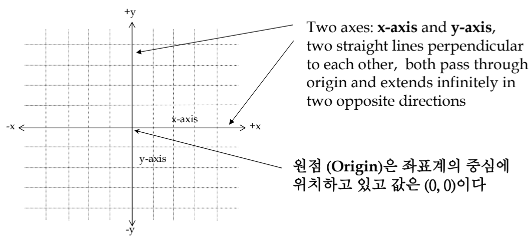
305890
2007년 봄학기
3/14/2007
박경신

Outline

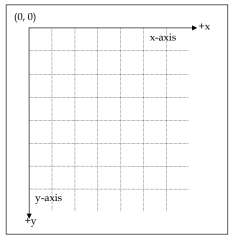
- Coordinate Systems (좌표계)
- Vector (벡터)
- Matrix (행렬)
- Rendering Pipeline (렌더링 파이프라인)
- Quaternion (사원수)
- Lighting (조명모델)

2D Cartesian Coordinate Systems

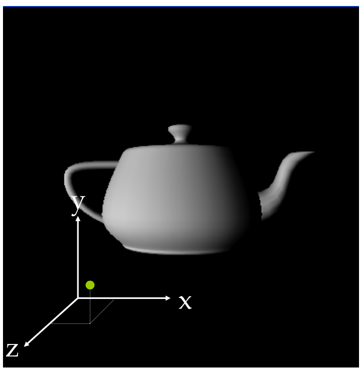
- Cartesian Coordination Systems



Screen Coordinate System

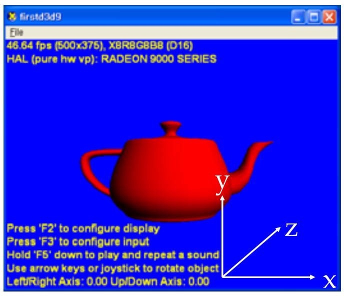


3D Coordinate Systems



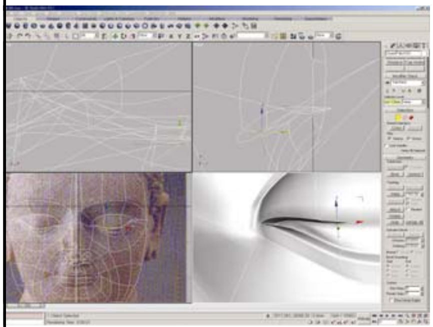
- OpenGL coordinate system is right-handed: x to the right, y up and z coming out of the screen.

3D Coordinate Systems



- Direct3D coordinate system is left-handed: x to the right, y up and z forward.

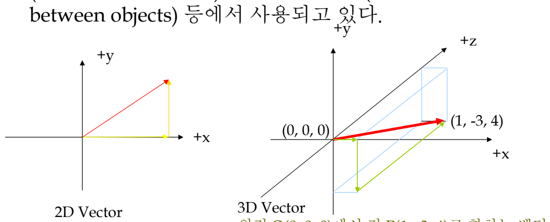
3D Coordinate Systems



- 3D Studio Max 좌표계는 x+ 오른쪽, y+ 스크린으로 튀어나오는 방향, z+ 위로.

Vector

- 벡터는 크기(magnitude 혹은 length)와 방향 (direction)이 있다
- 벡터는 조명의 방향 (light source directions), 표면의 방향 (surface orientations), 물체간의 거리 (relative distance between objects) 등에서 사용되고 있다.



원점 O(0, 0, 0)에서 점 P(1, -3, 4)로 향하는 벡터

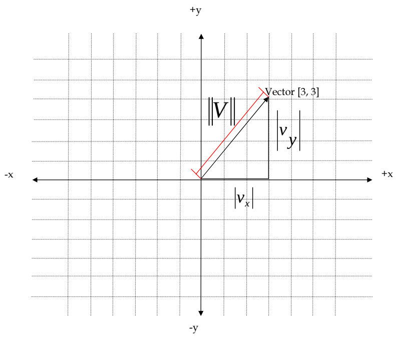
Vector Magnitude (Length)

- 벡터의 크기 (magnitude or length):

Examples: $\|v\| = \sqrt{v_1^2 + v_2^2 + \ldots + v_{n-1}^2 + v_n^2}$

$$\|(5,-4,7)\| = \sqrt{5^2 + (-4)^2 + 7^2}$$
$$= \sqrt{25 + 16 + 49}$$
$$= \sqrt{90}$$
$$= 3\sqrt{10}$$
$$\approx 9.4868$$

Vector Magnitude

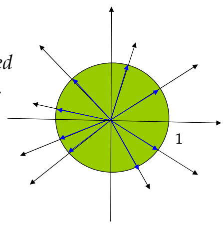


$$\|v\|^2 = |v_x|^2 + |v_y|^2$$
$$\sqrt{\|v\|^2} = \sqrt{v_x^2 + v_y^2}$$
$$\|v\| = \sqrt{v_x^2 + v_y^2}$$

Normalized Vectors

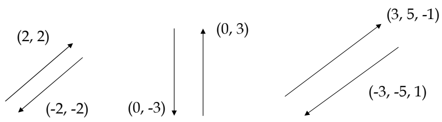
- 벡터의 크기에 상관없이 벡터의 방향만을 필요로 할 때가 있다,
- 단위벡터 (Unit vector)는 벡터의 크기 (magnitude)가 1이다.
- 단위벡터의 다른 이름은 *normalized vectors* 혹은 *normal*이라고 부른다.
- 벡터의 정규화 ("Normalizing" a vector):

$$v_{norm} = \frac{v}{\|v\|}, v \neq 0$$



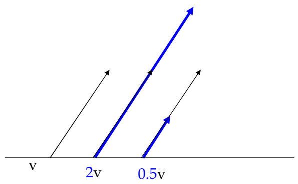
Negating a Vector

- 모든 벡터 **v**는 음수벡터 **-v**를 가지고 있다: **v + (-v) = 0**
- 음수벡터
  - $-(a_1, a_2, a_3, \ldots, a_n) = (-a_1, -a_2, -a_3, \ldots, -a_n)$
- 2D, 3D, 4D 벡터의 음수(negation)
  - $-(x, y) = (-x, -y)$
  - $-(x, y, z) = (-x, -y, -z)$
  - $-(x, y, z, w) = (-x, -y, -z, -w)$



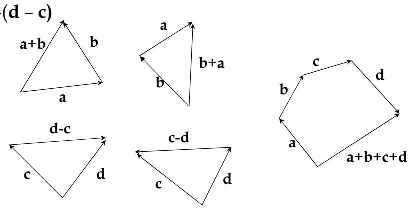
Vector Scalar Multiplication

- 벡터 스칼라 곱
  $k * (x, y, z) = (kx, ky, kz)$
- 벡터 스칼라 나누기
  $1/k * (x, y, z) = (x/k, y/k, z/k)$
- 예제:
  2 * (4, 5, 6) = (8, 10, 12)
  ½ * (4, 5, 6) = (2, 2.5, 3)
  -3 * (-5, 0, 0.4) = (15, 0, -1.2)
  3**a** + **b** = (3**a**) + **b**



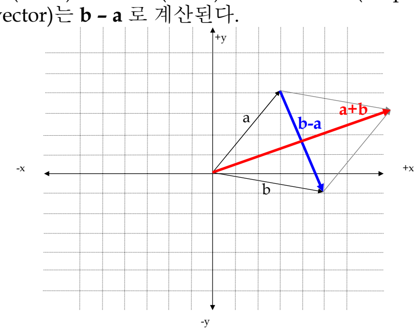
Vector Addition and Subtraction

- 벡터 더하기 (Addition)
  $(x_1, y_1, z_1) + (x_2, y_2, z_2) = (x_1+x_2, y_1+y_2, z_1+z_2)$
  a + b = b + a
- 벡터 빼기 (Subtraction)
  $(x_1, y_1, z_1) - (x_2, y_2, z_2) = (x_1-x_2, y_1-y_2, z_1-z_2)$
  c - d = -(d – c)



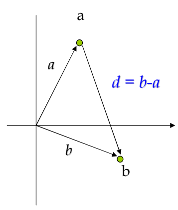
Vector Addition and Subtraction

- 점(Point) **a** 에서 점(Point) **b** 의 변위벡터 (Displacement vector)는 **b – a** 로 계산된다.



Distance

- 두 점 a, b 간의 거리(distance)는 다음과 같이 계산된다.
  - 벡터 *a*
  - 벡터 *b*
  - 변위벡터 $d = b - a$
  - 벡터 *d의 길이를 구한다.*
  - distance(a,b)= | *d* | = | *b-a* |



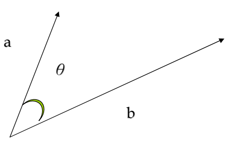
Vector Dot Product

- 두 벡터간의 내적 (Dot product): **a • b**
  $(a_1, a_2, a_3, \ldots, a_n) \cdot (b_1, b_2, b_3, \ldots, b_n) = a_1b_1 + a_2b_2 + \ldots + a_{n-1}b_{n-1} + a_nb_n$
  or
  $$a \cdot b = \sum_{i=1}^{n} a_i b_i$$
  $$a \cdot a = \|a\|^2$$
- 예제:
  (4, 6) · (-3, 7) = 4*-3 + 6*7 = 30
  (3, -2, 7) · (0, 4, -1) = 3*0 + -2*4 + 7*-1 = -15

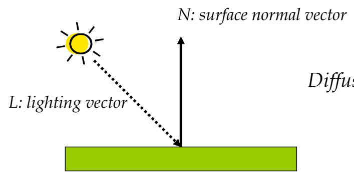
Vector Dot Product

- 두 벡터간의 내적이 벡터 크기 배율을 가진 벡터 간 각도의 코사인(cosine)이다.

$$a \cdot b = \|a\|\|b\|\cos\theta$$
$$\theta = \mathrm{acos}\left(\frac{a \cdot b}{\|a\|\|b\|}\right)$$
$$\theta = \mathrm{acos}(a \cdot b), \textit{where } a, b \textit{ are unit vectors}$$



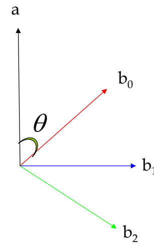
Lambertian Lighting Model



Diffuse = N • (-L)

Dot Product as Measurement of Angle

- 다음은 내적의 특성을 정리한 것이다.



$$a \cdot b_0 > 0 \ \ when \ \ 0^\circ \le \theta < 90^\circ$$

$$a \cdot b_1 = 0 \ \ when \ \ \theta = 90^\circ$$

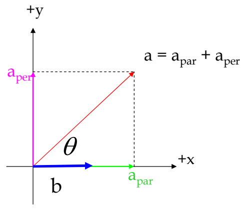
$$a \cdot b_2 < 0 \ \ when \ \ 90^\circ < \theta \le 180^\circ$$

Projecting One Vector onto Another

- Given two vector **a** and **b**, projecting **a** onto **b**.
  a_{par}, a_{per} are parallel and perpendicular to **b**, **a** = $a_{par} + a_{per}$

$$\frac{b}{\|b\|} = \frac{a_{par}}{\|a_{par}\|}, \quad a_{par} = b\frac{\|a_{par}\|}{\|b\|}$$

$$\cos\theta = \frac{\|a_{par}\|}{\|a\|}, \quad \cos\theta\|a\| = \|a_{par}\|$$



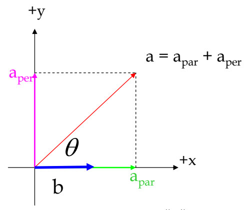
Projecting One Vector onto Another

$$a_{par} = b\frac{\|a\|\cos\theta}{\|b\|} = b\frac{\|a\|\|b\|\cos\theta}{\|b\|^2} = b\frac{a \cdot b}{\|b\|^2}$$

$$a = a_{per} + a_{par}$$

$$a_{per} = a - a_{par}$$

$$a_{per} = a - b\frac{a \cdot b}{\|b\|^2}$$



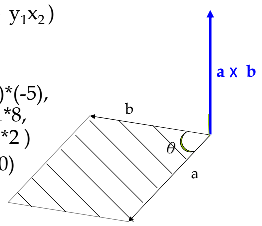
If b is a unit vector, then $\|b\| = 1$

$$a_{par} = b(a \cdot b)$$

$$a_{per} = a - b(a \cdot b)$$

Vector Cross Product

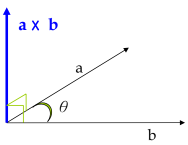
- 벡터의 외적 (Cross product): **a x b**
  (x_1, y_1, z_1) **x** (x_2, y_2, z_2) = ($y_1z_2 - z_1y_2$,
  $z_1x_2 - x_1z_2$,
  $x_1y_2 - y_1x_2$)
- 예제:
  (1, 3, -4) **x** (2, -5, 8) = (3*8 – (-4)*(-5),
  (-4)*2 – 1*8,
  1*(-5) – 3*2)
  = (4, -16, -10)



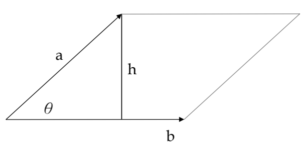
Vector Cross Product

- 두 벡터간의 외적 (**a x b**)의 크기는 각 벡터의 크기와 두 벡터간 각도의 사인(sine)의 곱이다:

$$\|a \times b\| = \|a\|\|b\|\sin\theta$$



- 평행사변형 (parallogram)의 영역(area)은 *bh*로 계산된다:



$$A = bh = b(a\sin\theta) = \|a\|\|b\|\sin\theta = \|a \times b\|$$

Matrix

- 다음과 같이 사각형 형태로 표기한 숫자 배열을 행렬 M (*r* x *c* matrix) 라고 한다.
  - 가로로 배열된 행렬을 **행 (row)**
  - 세로로 배열된 행렬을 **열 (column)**
  - Mij는 행 i 와 열 j에 있는 **원소 (element)**

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{33} & m_{33} \end{pmatrix} \Big\} r(3) \text{ rows}$$

c(3) columns

Square Matrix

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{33} & m_{33} \end{pmatrix}$$

Nondiagonal elements

Diagonal elements

- n x n 행렬을 n 차 정방행렬 (square matrix)라 한다. e.g. 2x2, 3x3, 4x4
- Diagonal elements vs. Non-diagonal elements

Identity Matrix

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- 항등행렬 혹은 단위행렬 (Identity Matrix)은 I 또는 I_n로 표시한다.
  - 주대각선이 전부 1이고, 나머지 원소는 0을 값으로 갖는 *nxn* 정방행렬이다.
- M I = I M = M

Transpose Matrix

- M (rxc matrix)의 전치행렬(Transpose of M)은 M^T으로 표시하며 cxr matrix 로 변환된다.
  - $M^T_{ij} = M_{ji}$
  - $(M^T)^T = M$
  - $D^T = D$ for any diagonal matrix D.

$$\begin{pmatrix} a & m & c \\ d & e & f \\ g & h & i \end{pmatrix}^T = \begin{pmatrix} a & d & g \\ m & e & h \\ c & f & i \end{pmatrix}$$

Matrix Scalar Multiplication

- 행렬의 스칼라 곱 (Multiplying a matrix **M** with a scalar ***k*** = ***k*M)**

$$k\mathbf{M} = k\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{33} & m_{33} \end{pmatrix} = \begin{pmatrix} km_{11} & km_{12} & km_{13} \\ km_{21} & km_{22} & km_{23} \\ km_{31} & km_{33} & km_{33} \end{pmatrix}$$

Two Matrices Multiplication

- Matrix **C** is the **rxc** product **AB** of the **rxn** matrix **A** with the **nxc** matrix **B**.
- Each element c_{ij} is equal to the vector dot product of row *i* of **A** with column *j* of **B**.
- $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

3+18+48

$$\begin{pmatrix} 1 & 3 & 6 \\ 10 & 0 & -5 \\ 4 & 7 & 2 \end{pmatrix} * \begin{pmatrix} 3 & 7 & 1 \\ 6 & 4 & 9 \\ 8 & -9 & 4 \end{pmatrix} = \begin{pmatrix} 69 & -35 & 52 \\ -10 & 115 & -10 \\ 70 & 38 & 75 \end{pmatrix}$$

r x n — *n x c* — *r x c*

must match — *columns in result* — *rows in result*

Matrix Determinant

- 행렬식의 값 (The determinant of a square matrix M)은 **|M|** 혹은 **“det M”** 으로 표시된다.
- The determinant of a non-square matrix is undefined.

$$|M| = \begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix} = m_{11}m_{22} - m_{12}m_{21}$$

$$|M| = \begin{vmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{vmatrix} = m_{11}(m_{22}m_{33} - m_{23}m_{32}) + m_{12}(m_{23}m_{31} - m_{21}m_{33}) + m_{13}(m_{21}m_{32} - m_{22}m_{31})$$

Inverse Matrix

- 정방행렬 M (square matrix)의 역행렬(Inverse of M)은 M^{-1}으로 표시한다.
- $M^{-1} = \frac{adjM}{|M|}$
- $(M^{-1})^{-1} = M$
- $M(M^{-1}) = M^{-1}M = I$
- The determinant of a non-singular matrix (i.e, invertible) is nonzero.
- The *adjoint* of M, denoted **“adj M”** is the transpose of the matrix of cofactors.

$$adjM = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}^T$$

Cofactor of a Square Matrix & Computing Determinant using Cofactor

- The *cofactor* of a square matrix M at a given row and column is the signed determinant of the corresponding minor of M.
- $C_{ij} = (-1)^{ij} \, |M^{\{ij\}}|$
- Compute an n x n determinant using cofactor:

$$|M| = \sum_{j=1}^{n} m_{ij}c_{ij} = \sum_{j=1}^{n} m_{ij}(-1)^{i+j}|M^{\{ij\}}|$$

$$|M| = \begin{vmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{vmatrix} = m_{11}\begin{vmatrix} m_{22} & m_{23} & m_{24} \\ m_{32} & m_{33} & m_{34} \\ m_{42} & m_{43} & m_{44} \end{vmatrix} - m_{12}|M^{\{12\}}| + m_{13}|M^{\{13\}}| - m_{14}|M^{\{14\}}|$$

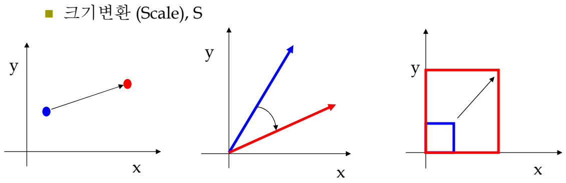
Minor of a Matrix

- The submatrix $M^{\{ij\}}$ is known as a minor of M, obtained by deleting row *i* and column *j* from M.

$$M = \begin{pmatrix} -4 & -3 & 3 \\ 0 & 2 & -2 \\ 1 & 4 & -1 \end{pmatrix} \quad M^{\{12\}} = \begin{pmatrix} 0 & -2 \\ 1 & -1 \end{pmatrix}$$

Transformation

- 기하변환 (geometric transformation)이란 점들(points)을 한곳에서 다른 곳으로 옮겨주는 함수를 의미한다.
- 2D transformation
  - 이동변환 (Translation), T
  - 회전변환 (Rotation), R
  - 크기변환 (Scale), S



2D Transformation Matrix

$$T(dx,dy) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{pmatrix}$$

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

$$S(sx, sy) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Inverse 2D Transformation Matrix

$$T^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -dx & -dy & 1 \end{pmatrix}$$

$$R^{-1} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S^{-1} = \begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3D Translation Matrix

- 주어진 위치로 이동하는 행렬을 말한다. 사실 크기의 행렬만으로도 3차원 공간을 표현하는 것이 가능하지만 이동행렬 때문에 크기의 행렬이 되었다.

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{pmatrix}$$

3D Rotation Matrix

$$R_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_Y = \begin{pmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_Z = \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

회전 행렬은 오일러 각(euler-angle) 방식과 각 축(angle-axis) 방식 등의 다양한 표현 방법이 존재하지만, 일반적으로는 뒤에서 배울 사원수(quaternion)를 많이 사용한다.

3D Scale Matrix

- 크기변환을 표현하는 행렬이다. sx의 값이 1보다 작으면 x축 방향으로 축소, 1보다 크면 x축 방향으로 확대를 나타낸다. sy, sz도 마찬가지 이다.

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformation Matrix

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} p_x & q_x & r_x & 0 \\ p_y & q_y & r_y & 0 \\ p_z & q_z & r_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

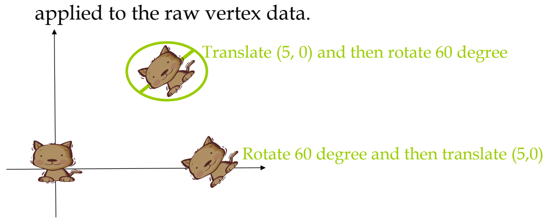
$$= \begin{pmatrix} xp_x + yq_x + zr_x & xp_y + yq_y + zr_y & xp_z + yq_z + zr_z \end{pmatrix}$$

$$= \mathbf{xp + yq + zr}$$

- Multiplying a vector by the matrix performs a coordinate space transformation.
- If **aM = b**, we say that **M** transformed **a** to **b**.

Transformation Order Matters!

- Transformations are not, in general, commutative.
- Applying the same transformations in different orders can produce different results.
- Multiple transformations are effectively applied in reverse order, i.e., the last transformation is the first on applied to the raw vertex data.

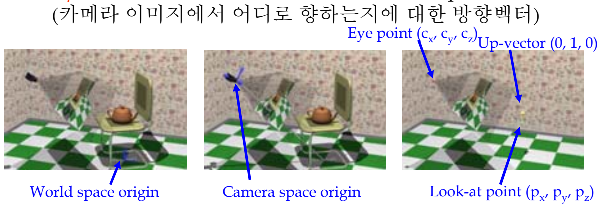


Modeling Transformation

- 모델 좌표계 (Model Coordinate System)을 3차원의 절대 좌표계인 월드 좌표계 (World Coordinate System)로 변환
- 이동(translation), 회전(rotation), 크기 변환(scaling) 행렬이 대표적인 예이다.
- Twist, Shear, Mirror등의 변환도 있다.

Viewing Transformation

- 월드좌표계 (WCS)에서 눈좌표계 (ECS)로 변환
- *Eye Point* : 카메라의 원점 (월드 좌표계)
- *Look-At* : 카메라가 쳐다보고 있는 위치 (카메라 이미지의 중심이 되는 위치)
- *Up-Vector* : 월드좌표계에서 카메라가 보는 up 벡터 (카메라 이미지에서 어디로 향하는지에 대한 방향벡터)



Viewing Transformation

- Camera coordinates
  - n (z-axis) = normalize(p – c)
  - v (x-axis) = normalize(up x n)
  - u (y-axis) = n x v

- Viewing transformation matrix = world * translate T(-c) * rotate coordinates (n, v, u)

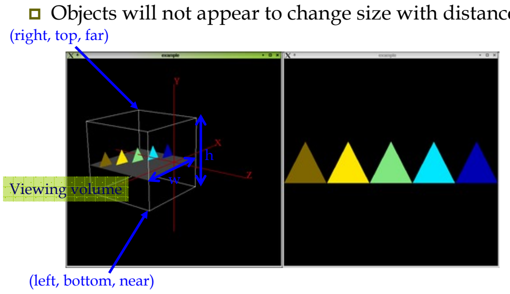
$$\begin{pmatrix} v_x & u_x & n_x & 0 \\ v_y & u_y & n_y & 0 \\ v_z & u_z & n_z & 0 \\ -(c\bullet v) & -(c\bullet u) & -(c\bullet n) & 1 \end{pmatrix}$$

Projection Transformation

- 눈좌표계 (ECS)를 절단좌표계 (CC)로 변환하고 정규디바이스좌표계 (NDCS)로 변환
- 직교투영 (Orthographic Projection)
- 원근투영 (Perspective Projection)

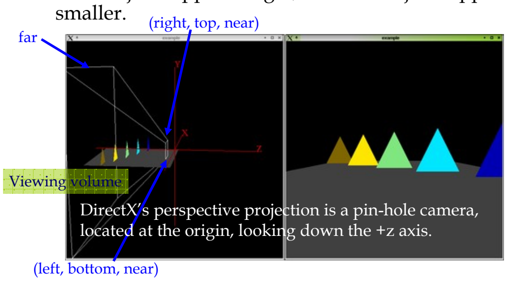
Orthographic Projection

- 직교투영 (Orthographic projection) projects rectilinear box onto display.
- Objects will not appear to change size with distance.



Perspective Projection

- 원근투영 (Perspective projection) projects *frustum (truncated pyramid)* onto display.
- Near objects appear larger, distance objects appear smaller.



Orthographic Projection Transformation

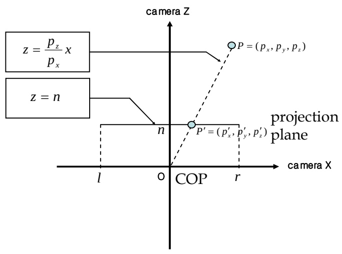
$$\begin{pmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & \frac{1}{f-n} & 1 \\ 0 & 0 & \frac{-n}{f-n} & 0 \end{pmatrix}$$

w = right – left
h = top – bottom
f = far
n = near

Perspective Projection Transformation

- Projection plane in front of the center of projection

$$\begin{pmatrix} \frac{2n}{w} & 0 & 0 & 0 \\ 0 & \frac{2n}{h} & 0 & 0 \\ 0 & 0 & \frac{f}{f-n} & 1 \\ 0 & 0 & \frac{-fn}{f-n} & 0 \end{pmatrix}$$



w = right – left
h = top – bottom
f = far
n = near

Clipping Transformation

$$\begin{pmatrix} \frac{2}{C_w} & 0 & 0 & 0 \\ 0 & \frac{2}{C_h} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_{max}-Z_{min}} & 0 \\ 1-2\frac{C_x}{C_w} & 1-2\frac{C_y}{C_w} & \frac{-Z_{min}}{Z_{max}-Z_{min}} & 1 \end{pmatrix}$$

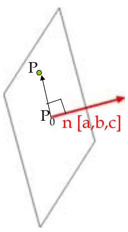
Viewport Transformation

- 정규디바이스좌표계 (Normalized Device Coordinate System 혹은 Screen Coordinate System)에서 윈도우 좌표계(Window Coordinate System)로 변환

$$\begin{pmatrix} \frac{dwWidth}{2} & 0 & 0 & 0 \\ 0 & -\frac{dwHeight}{2} & 0 & 0 \\ 0 & 0 & dwMaxZ - dwMinZ & 0 \\ dwX + \frac{dwWidth}{2} & dwY + \frac{dwHeight}{2} & dwMinZ & 1 \end{pmatrix}$$

Plane

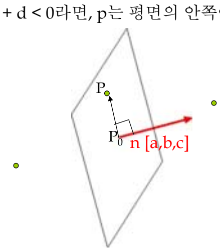
- 평면은 하나의 법선 벡터 (normal vector) n과 평면 상의 점 p_0으로 표현된다:
  - n = [(a, b, c), d]
  - ax + by + cz + d = 0
  - n•p + d = 0
  - d = -n•p



- 평면 위의 점 p에 대해, n•(p - p_0) =0
- 만약 평면의 법선 벡터 n이 단위 길이라면, n•p + d로 평면에서 점 p까지의 부호를 가진 가장 짧은 거리 (the shortest signed distance)를 얻을 수 있다: d = -n•p

Relationship between Point and Plane

- 점 p과 평면 (n, d)의 공간 관계
  - 만약 n•p + d = 0라면, p는 평면에 있다.
  - 만약 n•p + d > 0라면, p는 평면의 바깥쪽에 있다.
  - 만약 n•p + d < 0라면, p는 평면의 안쪽에 있다.



Plane Normalization

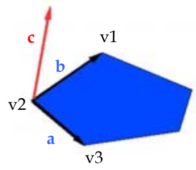
- 평면의 정규화 (normalization)
  - 평면의 법선 벡터 (normal)를 정규화
  - 법선 벡터의 길이가 상수 d에 영향을 주기 때문에, d도 역시 정규화

$$\frac{1}{\|\mathrm{n}\|}(\mathrm{n},\mathrm{d})=\left(\frac{n}{\|n\|},\frac{d}{\|n\|}\right)$$

Computing a Normal from 3 Points in Plane

- Given the vertex data for a polygon,
  - you can compute a normal to the polygon by taking two non-collinear edges (assuming that no two adjacent edges will be collinear)
  - Compute the cross product, and normalize the result

```
computeNormal(v1, v2, v3)
{
    a = v3 – v2
    b = v1 – v2
    c = cross(a, b)
    if (length(c)==0)
        return [0, 1, 0]
    else
        return c.normalize()
}
```



Computing a Distance from Point to Plane

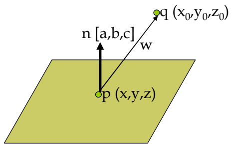
- Given a plane and a point, q, that is not in the plane,
  - Assume n is a normal vector of the plane and D is the distance from p to q, then

$$w=[x_0-x, y_0-y, z_0-z]$$

$$D=\frac{|n\bullet w|}{\|n\|}$$

$$=\frac{|a(x_0-x)+b(y_0-y)+c(z_0-z)|}{\sqrt{a^2+b^2+c^2}}$$

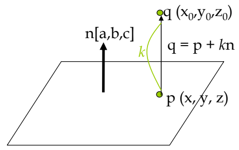
$$=\frac{ax_0+by_0+cz_0+d}{\sqrt{a^2+b^2+c^2}}$$



Projecting w onto n: $w_{\|}=n\frac{w\cdot n}{\|n\|^2}$ & $\|w_{\|}\|=\frac{|w\cdot n|}{\|n\|}$

Closest Point on the Plane

- 공간에 하나의 점 q를 가지고 있고, 점 q에서 가장 가까운 평면 (n, d)상의 점 p를 구하라
  - p = q – kn (k는 q에서 plane과의 the shortest signed distance)
  - n이 단위벡터(unit vector)인 경우, k = n•q + d
  - p = q – (n•q + d)n



$$\mathrm{Distance(q, plane)}=\frac{ax_0+by_0+cz_0+d}{\sqrt{a^2+b^2+c^2}}$$

$$where\ q(x_0,y_0,z_0)\ \ and\ \ Plane\ ax+by+cz+d=0$$

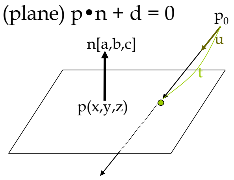
Intersection of Ray and Plane

- 광선 (ray) $\mathbf{p}(t)=\mathbf{p}_0+t\mathbf{u}$ & 평면 (plane) p•n + d = 0
- 광선/평면의 교차점:

$$(\mathrm{p}_0+t\mathrm{u})\cdot\mathrm{n}+d=0$$

$$t\mathrm{u}\cdot\mathrm{n}=-d-\mathrm{p}_0\cdot\mathrm{n}$$

$$t=\frac{-(\mathrm{p}_0\cdot\mathrm{n}+d)}{\mathrm{u}\cdot\mathrm{n}}$$



- 만약 광선이 평면과 평행하다면, denominator **u**•**n**=0 따라서 광선은 평면과 교차하지 않는다.
- 만약 t 값이 범위 [0, ∞)내에 있지 않으면, 광선은 평면과 교차하지 않는다.
- $$\mathrm{p}\left(\frac{-(\mathrm{p}_0\cdot\mathrm{n}+d)}{\mathrm{u}\cdot\mathrm{n}}\right)=\mathrm{p}_0+\frac{-(\mathrm{p}_0\cdot\mathrm{n}+d)}{\mathrm{u}\cdot\mathrm{n}}\mathrm{u}$$