

Geometric Objects - Spaces and Matrix

514780
2017년 가을학기
9/21/2017
단국대학교 박경신

Spaces

- 벡터 공간 (Vector space)
 - 실수와 같은 스칼라 (scalars)와 벡터 (vectors)를 가지고 있다.
 - Scalars: α, β, δ
 - Vectors: u, v, w
- 어파인 공간 (Affine space)
 - 벡터공간에 점(points)이 추가된다.
 - Points: P, Q, R
- 유클리드 공간 (Euclidean space)
 - 거리 개념 (Concept of distance)이 추가된다.

Scalars, Points, Vectors

- 3 basic types needed to describe the geometric objects and their relations
- Scalars: α, β, δ
- Points: P, Q, R
- Vectors: u, v, w
- Vector space
 - scalars & vectors
- Affine space
 - Extension of the vector space that includes a point

Scalars

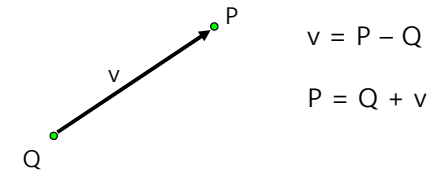
- 덧셈, 곱셈에 대하여 교환법칙, 결합법칙, 분배법칙 성립
 - $\alpha + \beta = \beta + \alpha$
 - $\alpha \cdot \beta = \beta \cdot \alpha$
 - $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$
 - $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$
 - $\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$
- 덧셈 항등원(identity) 0과 곱셈 항등원 1
 - $\alpha + 0 = 0 + \alpha = \alpha$
 - $\alpha \cdot 1 = 1 \cdot \alpha = \alpha$
- 덧셈의 역(inverse)과 곱셈의 역
 - $\alpha + (-\alpha) = 0$
 - $\alpha \cdot \alpha^{-1} = 1$

Vectors

- 벡터는 크기(magnitude 혹은 length)와 방향(direction)을 가지고 있다.
- 속도(velocity)나 힘(force)과 같은 물리량은 벡터이다.
- 컴퓨터그래픽스에서 쓰이는 방향성 선분(directed line segments)은 벡터이다.
- 벡터는 공간 내에 고정된 위치(position)를 갖지 않는다.

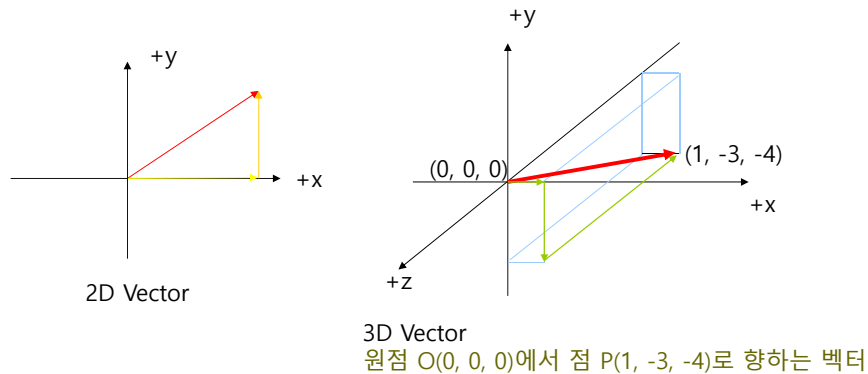
Points

- 점(points)은 공간에서의 위치(position)을 갖는다.
- 점과 벡터와의 연산
 - 점-점 뺄셈(point-point subtraction)은 벡터를 만든다.
 - 점-벡터 덧셈(point-vector addition)은 점을 만든다.

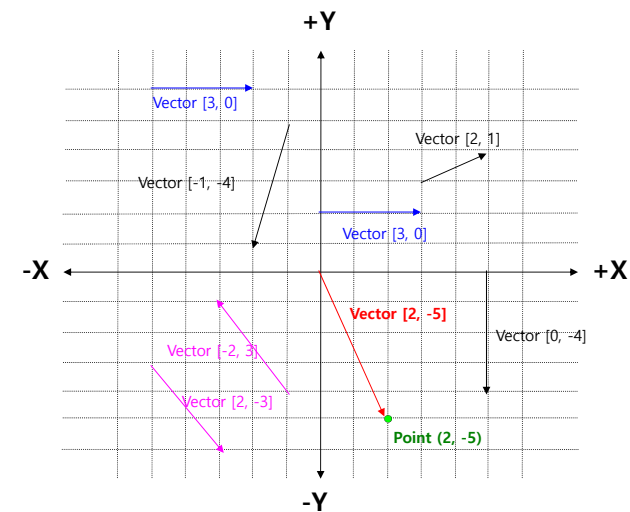


Specifying Vectors

- 2D Vector: (x, y)
- 3D Vector: (x, y, z)



Examples of 2D vectors

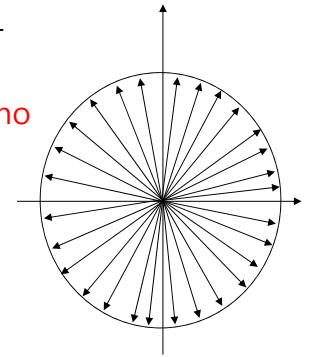


Vector Operations

- zero vector
- vector negation
- vector/scalar multiply
- add & subtract two vectors
- vector magnitude (length)
- normalized vector
- distance formula
- vector product
 - dot product
 - cross product

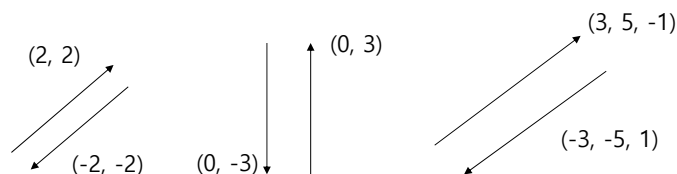
The Zero Vector

- 3차원 영벡터 (zero vector)는 $(0, 0, 0)$ 이다.
- 영벡터는 0 크기 (zero magnitude)를 가지고 있다.
- 영벡터는 방향을 가지고 있지 않다 (no direction).



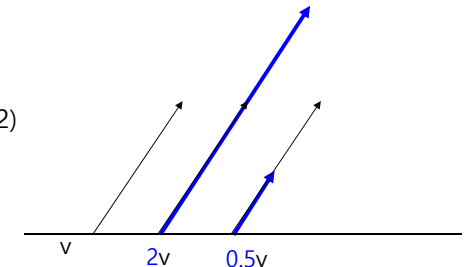
Negating a Vector

- 모든 벡터 $\mathbf{v}$ 는 음수벡터 $-\mathbf{v}$ 를 가지고 있다: $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- 음수벡터
 - $-(a_1, a_2, a_3, \dots, a_n) = (-a_1, -a_2, -a_3, \dots, -a_n)$
- 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
 - $\alpha * (x, y, z) = (\alpha x, \alpha y, \alpha z)$
- 벡터 스칼라 나누기
 - $1/\alpha * (x, y, z) = (x/\alpha, y/\alpha, z/\alpha)$
- 예제:
 - $2 * (4, 5, 6) = (8, 10, 12)$
 - $1/2 * (4, 5, 6) = (2, 2.5, 3)$
 - $-3 * (-5, 0, 0.4) = (15, 0, -1.2)$
 - $3\mathbf{u} + \mathbf{v} = (3\mathbf{u}) + \mathbf{v}$



Vector Addition and Subtraction

□ 벡터 더하기 (Addition)

- 수미연결규칙 (head-to-tail axiom)으로 정의

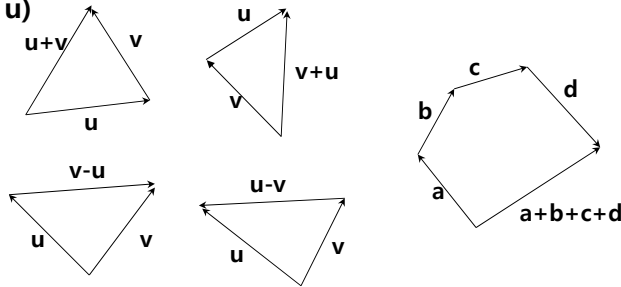
$$(x_1, y_1, z_1) + (x_2, y_2, z_2) = (x_1+x_2, y_1+y_2, z_1+z_2)$$

$$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$$

□ 벡터 빼기 (Subtraction)

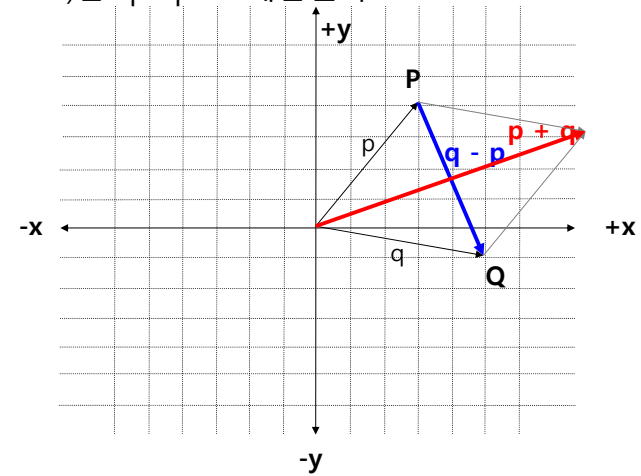
$$(x_1, y_1, z_1) - (x_2, y_2, z_2) = (x_1-x_2, y_1-y_2, z_1-z_2)$$

$$\mathbf{u} - \mathbf{v} = -(\mathbf{v} - \mathbf{u})$$



Vector Addition and Subtraction

- 점(Point) P 에서 점(Point) Q 의 변위벡터 (Displacement vector)는 $\mathbf{q} - \mathbf{p}$ 로 계산된다.



Vector Magnitude (Length)

- 벡터의 크기 (magnitude or length):

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

$$\|(5, -4, 7)\| = \sqrt{5^2 + (-4)^2 + 7^2}$$

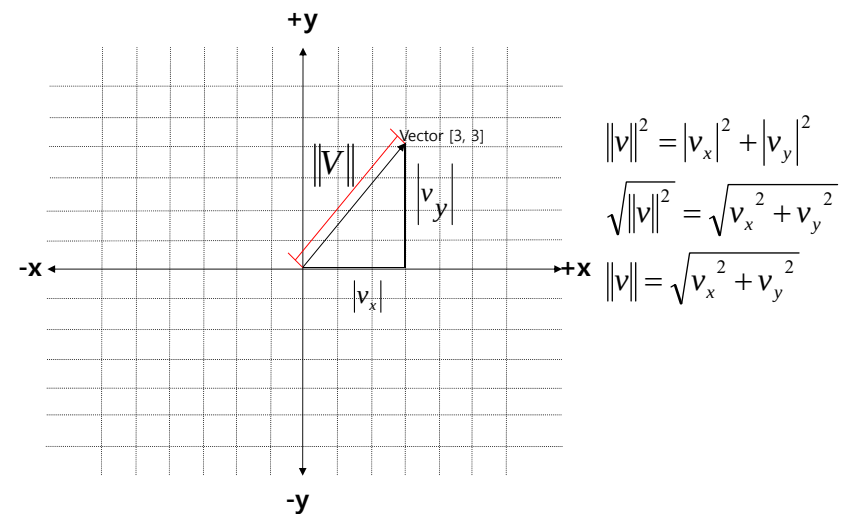
$$= \sqrt{25 + 16 + 49}$$

$$= \sqrt{90}$$

$$= 3\sqrt{10}$$

$$\approx 9.4868$$

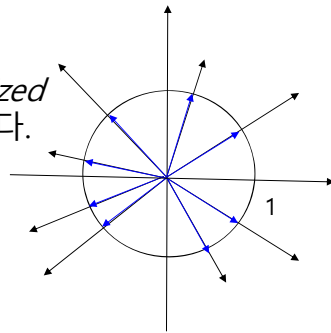
Vector Magnitude



Normalized Vectors

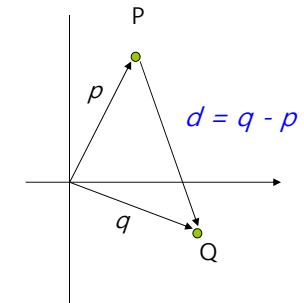
- 벡터의 크기에 상관없이 벡터의 방향만을 필요로 할 때가 있다.
- 단위벡터 (Unit vector)는 벡터의 크기 (magnitude)가 1이다.
- 단위벡터의 다른 이름은 *normalized vectors* 혹은 *normal*이라고 부른다.
- 벡터의 정규화 ("Normalizing" a vector):

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



Distance

- 두 점 P, Q 간의 거리(distance)는 다음과 같이 계산된다.
 - 벡터 p
 - 벡터 q
 - 변위벡터 $d = q - p$
 - 벡터 d 의 길이를 구한다.
 - $distance(P, Q) = \|d\| = \|q - p\|$



Vector Dot Product

- 두 벡터간의 내적 (Dot product): $u \cdot v$
 $(u_1, u_2, u_3, \dots, u_n) \cdot (v_1, v_2, v_3, \dots, v_n) =$
 $u_1v_1 + u_2v_2 + \dots + u_{n-1}v_{n-1} + u_nv_n$

or

$$u \cdot v = \sum_{i=1}^n u_i v_i$$

$$u \cdot u = \|u\|^2$$

- 예제:

$$(4, 6) \cdot (-3, 7) = 4 \cdot (-3) + 6 \cdot 7 = 30$$

$$(3, -2, 7) \cdot (0, 4, -1) = 3 \cdot 0 + (-2) \cdot 4 + 7 \cdot (-1) = -15$$

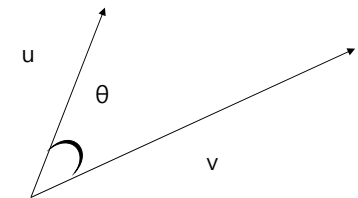
Vector Dot Product

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

$$u \cdot v = \|u\| \|v\| \cos \theta$$

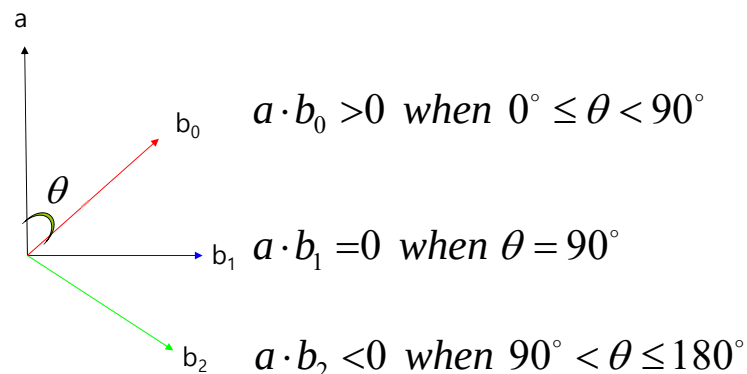
$$\theta = \arccos\left(\frac{u \cdot v}{\|u\| \|v\|}\right)$$

$$\theta = \arccos(u \cdot v), \text{ where } u, v \text{ are unit vectors}$$



Dot Product as Measurement of Angle

□ 다음은 내적의 특성을 정리한 것이다.



Projecting One Vector onto Another

□ 두 개의 벡터, w, v 가 주어지면, 그 중 하나의 벡터 w 를 다른 벡터 v 에 평행한 성분과 직교하는 성분으로 나눌 수 있다.

$$w = w_{\text{par}} + w_{\text{per}}$$

$$w = \alpha v + u$$

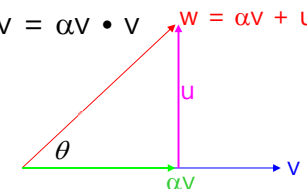
u 는 v 에 직교해야 하므로, $u \cdot v = 0$

$$w \cdot v = (\alpha v + u) \cdot v = \alpha v \cdot v + u \cdot v = \alpha v \cdot v$$

$$\alpha = \frac{w \cdot v}{v \cdot v}$$

$$u = w - \alpha v = w - \frac{w \cdot v}{v \cdot v} v = w - \frac{w \cdot v}{\|v\|^2} v$$

$$\alpha v = w - u = w - w + \frac{w \cdot v}{v \cdot v} v = \frac{w \cdot v}{v \cdot v} v = \frac{w \cdot v}{\|v\|^2} v$$

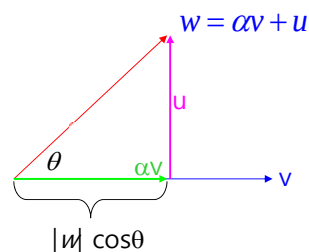


Projecting One Vector onto Another

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

$$w_{\text{per}} = u = w - (w \cdot v)v$$

$$w_{\text{par}} = \alpha v = (w \cdot v)v$$



$$\cos \theta = \frac{\|\alpha v\|}{\|w\|} \Rightarrow \|\alpha v\| = \|w\| \cos \theta$$

$$\sin \theta = \frac{\|u\|}{\|w\|} \Rightarrow \|u\| = \|w\| \sin \theta$$

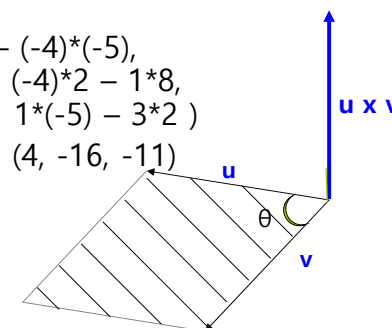
Vector Cross Product

□ 벡터의 외적 (Cross product): $u \times v$

$$(x_1, y_1, z_1) \times (x_2, y_2, z_2) = (y_1 z_2 - z_1 y_2, z_1 x_2 - x_1 z_2, x_1 y_2 - y_1 x_2)$$

□ 예제:

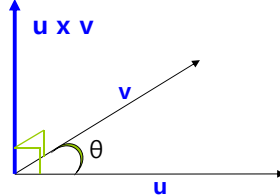
$$(1, 3, -4) \times (2, -5, 8) = (3 \cdot 8 - (-4) \cdot (-5), (-4) \cdot 2 - 1 \cdot 8, 1 \cdot (-5) - 3 \cdot 2) = (4, -16, -11)$$



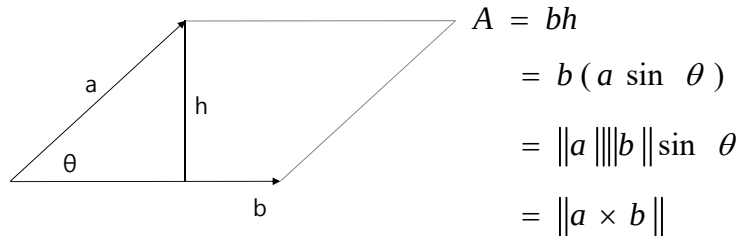
Vector Cross Product

- 두 벡터 간의 외적 ($\mathbf{u} \times \mathbf{v}$)의 크기는 각 벡터의 크기와 두 벡터간 각도의 사인(sine)의 곱이다.

$$\|\mathbf{u} \times \mathbf{v}\| = \|\mathbf{u}\| \|\mathbf{v}\| \sin \theta$$

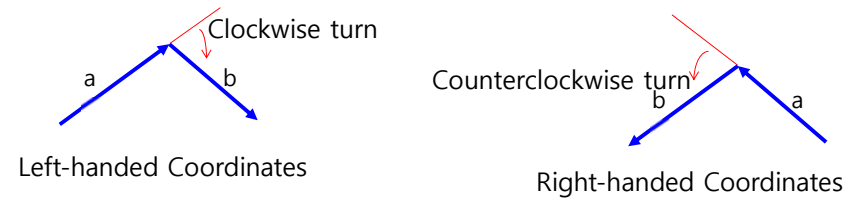


- 평행사변형 (parallelogram)의 영역(area)은 bh 로 계산된다.



Vector Cross Product

- 왼손 좌표계에서는 벡터 $\mathbf{u}, \mathbf{v}$ 가 시계방향(clockwise turn)으로 움직일 때 $\mathbf{u} \times \mathbf{v}$ 는 우리 향하는 방향을 가리키고, 반시계방향 (counterclockwise turn)으로 움직일 때, $\mathbf{u} \times \mathbf{v}$ 가 우리로부터 멀어지는 방향을 가리킨다.
- 오른손 좌표계에서는 벡터 $\mathbf{u}, \mathbf{v}$ 가 반시계방향으로 움직일 때 $\mathbf{u} \times \mathbf{v}$ 는 우리를 향하는 방향을 가리키고, 시계방향으로 움직일 때 $\mathbf{u} \times \mathbf{v}$ 는 우리로부터 멀어지는 방향을 가리킨다.



Linear Algebra Identities

Identity	Comments
$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$	벡터 덧셈 교환법칙
$\mathbf{u} - \mathbf{v} = \mathbf{u} + (-\mathbf{v})$	벡터 뺄셈
$(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$	벡터 덧셈 결합법칙
$\alpha(\beta\mathbf{u}) = (\alpha\beta)\mathbf{u}$	스칼라-벡터 곱셈 결합법칙
$\alpha(\mathbf{u} + \mathbf{v}) = \alpha\mathbf{u} + \alpha\mathbf{v}$ $(\alpha + \beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$	스칼라-벡터 분배법칙
$\ \alpha\mathbf{v}\ = \alpha \ \mathbf{v}\ $	스칼라의 곱
$\ \mathbf{v}\ \geq 0$	벡터의 크기는 양수 (nonnegative)
$\ \mathbf{u}\ ^2 + \ \mathbf{v}\ ^2 = \ \mathbf{u} + \mathbf{v}\ ^2$	피타고리안 법칙 (Pythagorean theorem)
$\ \mathbf{u}\ + \ \mathbf{v}\ \geq \ \mathbf{u} + \mathbf{v}\ $	벡터 덧셈 삼각법칙 (Triangle rule)
$\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$	내적(dot product) 교환법칙
$\ \mathbf{v}\ = \sqrt{\mathbf{v} \cdot \mathbf{v}}$	내적(dot product)을 이용한 벡터의 크기 정의

Linear Algebra Identities

Identity	Comments
$\alpha(\mathbf{u} \cdot \mathbf{v}) = (\alpha\mathbf{u}) \cdot \mathbf{v} = \mathbf{u} \cdot (\alpha\mathbf{v})$	벡터의 내적과 스칼라 곱 결합법칙
$\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$	벡터 덧셈/뺄셈과 내적 분배법칙
$\mathbf{u} \times \mathbf{u} = \mathbf{0}$	벡터 자신의 외적 (cross product)은 0
$\mathbf{u} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{u})$	벡터의 외적은 반교환법칙 (anti-commutative)
$\mathbf{u} \times \mathbf{v} = (-\mathbf{u}) \times (-\mathbf{v})$	벡터의 외적은 각 벡터의 역에 외적과 같다
$\alpha(\mathbf{u} \times \mathbf{v}) = (\alpha\mathbf{u}) \times \mathbf{v} = \mathbf{u} \times (\alpha\mathbf{v})$	벡터의 외적과 스칼라 곱 결합법칙
$\mathbf{u} \times (\mathbf{v} + \mathbf{w}) = (\mathbf{u} \times \mathbf{v}) + (\mathbf{u} \times \mathbf{w})$	두 벡터의 덧셈과 다른 벡터와의 외적은 분배법칙을 성립
$\mathbf{u} \cdot (\mathbf{u} \times \mathbf{v}) = 0$	Dot product of any vector with cross product of that vector & another vector is 0

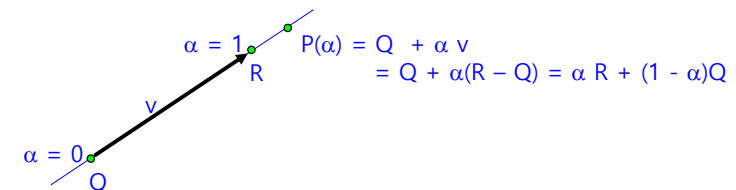
Geometric Objects

- Line
 - 2 points
- Plane
 - 3 points
- 3D objects
 - Defined by a set of triangles
 - Simple convex flat polygons
 - hollow

Lines

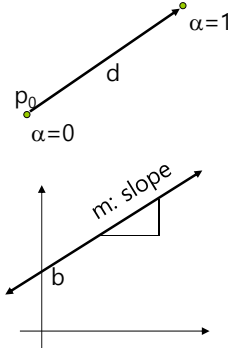
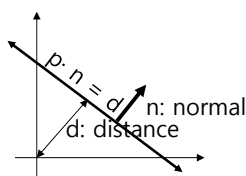
- 직선 (line)은 점과 벡터의 덧셈 (또는 두 점의 뺄셈)이다.
- 직선의 매개변수 형식 (parametric form): $P(\alpha) = P_0 + \alpha v$
 - P_0 는 임의의 점, v 는 임의의 벡터
 - 매개 변수 α 를 변경함으로써 직선 위에 점들이 생성
- $v = R - Q$

$$P = Q + \alpha v = Q + \alpha(R - Q) = \alpha R + (1 - \alpha)Q$$
- $P = \alpha_1 R + \alpha_2 Q$ where $\alpha_1 + \alpha_2 = 1$



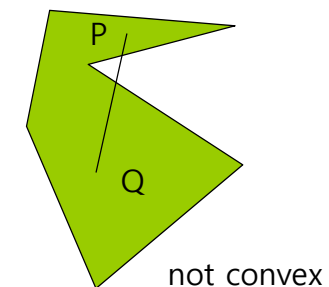
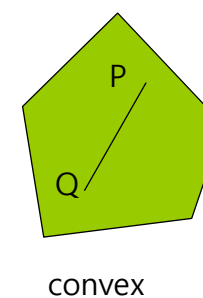
Lines, Rays, Line Segments

- 직선 (line)은 양방향으로 무한히 길다
- 선분 (line segment)는 두 점 (two endpoints) 사이의 선 조각이다. $0 \leq \alpha \leq 1$
- 방사선 (ray)는 한 점으로부터 한 방향으로 무한히 길다. $\alpha \geq 0$
- 직선 (line)의 정의:
 - $p(\alpha) = p_0 + \alpha d$ (parametric)
 - $y = mx + b$ (explicit)
 - $ax + by = d$ (implicit)
 - $p \cdot n = d$



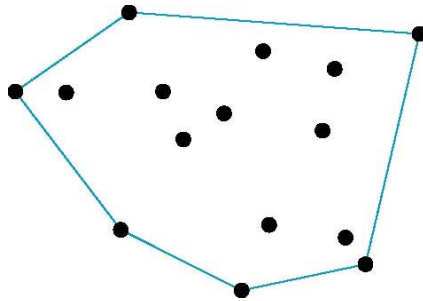
Convexity

- 볼록한(convex) 객체는 객체 내에 임의의 두 점을 연결하는 선분 위에 놓인 임의의 점이 객체 내에 있다.



Convex Hull

- 컨벡스 헐(convex hull)이란 점들의 집합 $P_1, P_2, \dots, P_n$ 을 포함하는 가장 작은 볼록 객체이다.
- 주어진 점들을 축소 포장 ("shrink wrapping")함으로 얻는다.



Affine Sums

- N개의 점 $P_1, P_2, \dots, P_n$ 에 의해 정의되는 객체를 포함하도록 하는 아핀 합은

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_n P_n$$

$$\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$$

- 추가적인 제한 $\alpha_i \geq 0, i=1, 2, \dots, n$ 하에서 n개의 점의 아핀 합에 의하여 만들어진 점들의 집합은 컨벡스 헐 (convex hull)이라고 불린다.
- 컨벡스 헐이 $\{P_1, P_2, \dots, P_n\}$ 에서 점들의 쌍을 연결하는 모든 선분을 포함하는 것을 확인할 수 있다.

Linear/Affine Combination of Vectors

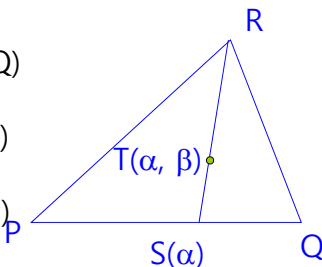
- 벡터의 선형조합 (linear combination of m vectors)
 - m개의 벡터 $v_1, v_2, \dots, v_m$
 - $w = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_m v_m$ where $\alpha_1, \alpha_2, \dots, \alpha_m$ are scalars
- 만약 벡터의 선형조합의 스칼라 값들 $\alpha_1, \alpha_2, \dots, \alpha_m$ 의 합이 1이면 어파인 조합(affine combination)이 된다.
 - $\alpha_1 + \alpha_2 + \dots + \alpha_m = 1$

Convex Combination

- 추가적인 제한, 즉 $\alpha_i \geq 0, i=1, 2, \dots, m$ 하에서 m개의 점의 아핀 합에 의하여 만들어진 점들의 집합은 컨벡스 헐 (convex hull)이라고 불린다.
- 따라서, 아래의 조건을 만족하는 벡터의 선형조합은 컨벡스(convex)이다.
 - $\alpha_1 + \alpha_2 + \dots + \alpha_m = 1$
 - and
 - $\alpha_i \geq 0$ for $i=1, 2, \dots, m$
 - α_i 는 0과 1 사이에 존재
- Convexity
 - Convex hull

Plane

- 평면은 매개변수형 직선을 확장함으로써 정의할 수 있다.
- 3 점 P, Q, R이 있다고 가정하자.
- P와 Q를 잇는 선분 (line segment PQ)
 - $S(\alpha) = \alpha P + (1 - \alpha)Q$
- S와 R을 잇는 선분 (line segment SR)
 - $T(\beta) = \beta S + (1 - \beta)R$
- P, Q, R에 의해 결정되는 평면 (plane)
 - $T(\alpha, \beta) = \beta(\alpha P + (1 - \alpha)Q) + (1 - \beta)R$
 $= P + \beta(1 - \alpha)(Q - P) + (1 - \beta)(R - P)$
 - $0 \leq \alpha, \beta \leq 1$ 인 $T(\alpha, \beta)$ 의 모든 점은 P, Q, R에 의해 형성되는 삼각형 내에 위치한다.

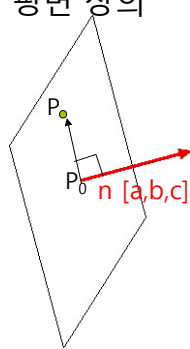


Plane

- 하나의 점 (point) P_0 와 두 개의 평행하지 않는 벡터 (two nonparallel vectors) u, v 에 의해 결정되는 평면의 식
 - $T(\alpha, \beta) = P_0 + \alpha u + \beta v$
 - $P - P_0 = \alpha u + \beta v$ (P는 평면의 한 점)
- u, v 의 외적 n 을 이용하면 평면의 방정식은 다음과 같다.
 - $n \cdot (P - P_0) = 0$ (where $n = u \times v$ and n is a normal vector)

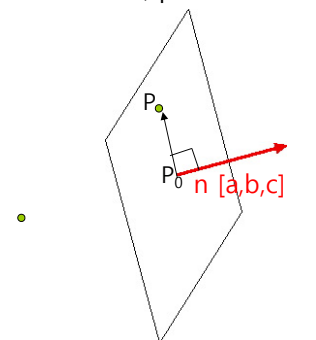
Plane

- 평면은 하나의 법선 벡터 (normal vector) n 과 평면 상의 점 p_0 으로 표현된다.
 - Plane (n, d) where $n(a, b, c)$
 - $ax + by + cz + d = 0$
 - $n \cdot p + d = 0$
 $d = -n \cdot p$
- 평면 위의 점 p 에 대해, $n \cdot (p - p_0) = 0$
- 만약 평면의 법선 벡터 n 이 단위 길이라면, $n \cdot p + d$ 로 평면에서 점 p 까지의 부호를 가진 가장 짧은 거리 (the shortest signed distance)를 얻을 수 있다: $d = -n \cdot p$



Relationship between Point and Plane

- 점 p 과 평면 (n, d) 의 공간 관계
 - 만약 $n \cdot p + d = 0$ 라면, p 는 평면에 있다.
 - 만약 $n \cdot p + d > 0$ 라면, p 는 평면의 바깥쪽에 있다.
 - 만약 $n \cdot p + d < 0$ 라면, p 는 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

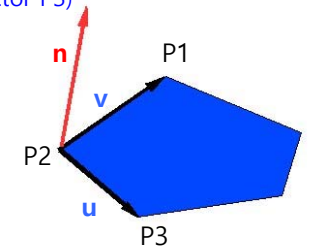
$$\frac{1}{\|n\|} (n, d) = \left(\frac{n}{\|n\|}, \frac{d}{\|n\|} \right)$$

Computing a Normal from 3 Points in Plane

□ 폴리곤의 정점 자료에서 법선을 구하라.

- 폴리곤의 법선벡터 (normal)는 2개의 two non-collinear edges를 계산한다. (assuming that no two adjacent edges will be collinear)
- 그리고, 외적 (cross product)을 구한 후 정규화(normalize) 한다.

```
void computeNormal(vector P1, vector P2, vector P3)
{
    vector u, v, n, y(0, 1, 0);
    u = P3 - P2;
    v = P1 - P2;
    n = cross(u, v);
    if (n.length() == 0)
        return y;
    else
        return n.normalize();
}
```



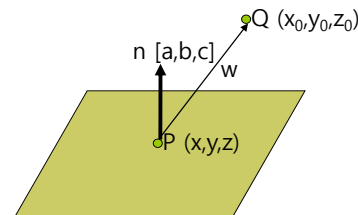
Computing a Distance from Point to Plane

□ 공간에 한 평면 (n, d)과 평면 밖의 한 점 Q와 가장 가까운 거리를 구하라.

- 평면의 법선 벡터(normal vector)가 n 이고, D는 평면 상의 한 점 P와 점 Q간의 거리

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

$$D = \frac{|n \cdot w|}{\|n\|} = \frac{|a(x_0 - x) + b(y_0 - y) + c(z_0 - z)|}{\sqrt{a^2 + b^2 + c^2}} = \frac{ax_0 + by_0 + cz_0 + d}{\sqrt{a^2 + b^2 + c^2}}$$

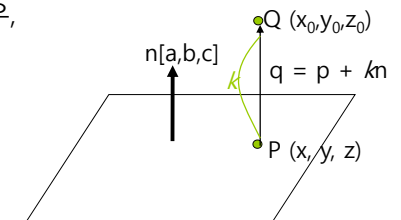


Projecting w onto n: $w_{\parallel} = n \frac{w \cdot n}{\|n\|^2}$ & $\|w_{\parallel}\| = \frac{|w \cdot n|}{\|n\|}$

Closest Point on the Plane

□ 공간에 하나의 점 Q에서 가장 가까운 평면 (n, d)상의 점 P를 구하라

- $p = q - kn$ (k는 점 Q에서 plane과의 the shortest signed distance)
- n이 단위벡터(unit vector)인 경우,
 $k = n \cdot q + d$
 $p = q - (n \cdot q + d)n$



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 $ax + by + cz + d = 0$
Distance(q, plane) = $n \cdot q + d$ (n is a unit vector)

Intersection of Ray and Plane

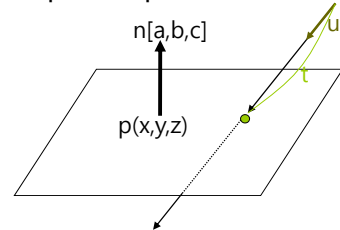
□ 광선 (ray) $\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{u}$ & 평면 (plane) $\mathbf{p} \cdot \mathbf{n} + d = 0$ $\mathbf{p}_0$

□ 광선/평면의 교차점:

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

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

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



□ 만약 광선이 평면과 평행하다면, denominator $\mathbf{u} \cdot \mathbf{n} = 0$ 따라서 광선은 평면과 교차하지 않는다.

□ 만약 t 값이 범위 $[0, \infty)$ 내에 있지 않으면, 광선은 평면과 교차하지 않는다.

$$\mathbf{p} \left(\frac{-(\mathbf{p}_0 \cdot \mathbf{n} + d)}{\mathbf{u} \cdot \mathbf{n}} \right) = \mathbf{p}_0 + \frac{-(\mathbf{p}_0 \cdot \mathbf{n} + d)}{\mathbf{u} \cdot \mathbf{n}} \mathbf{u}$$

Matrix

□ 다음과 같이 사각형 형태로 표기한 숫자 배열을 행렬 M ($r \times c$ matrix) 라고 한다.

■ 가로로 배열된 행렬을 행 (row)

■ 세로로 배열된 행렬을 열 (column)

■ M_{ij} 는 행 i 와 열 j 에 있는 원소 (element)

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \left. \begin{array}{l} \text{ } \\ \text{ } \\ \text{ } \end{array} \right\} r(2) \text{ rows}$$

$\underbrace{\hspace{10em}}_{c(2) \text{ columns}}$

Matrix

2x5
matrix

$$\begin{pmatrix} 2 & -4 & 7 & 7/8 & 8 \\ -3 & 4 & 3/8 & 0 & 1 \end{pmatrix}$$

$$m_{12} = -4$$

m_{ij} 는 M 행렬의 행 i 와 열 j 원소

4x3
matrix

$$\begin{pmatrix} 4 & 0 & 12 \\ -5 & 4 & 3 \\ 12 & 3/8 & -1 \\ 1/2 & 18 & 0 \end{pmatrix}$$

$$m_{42} = 18$$

Square Matrix

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

Nondiagonal elements

Diagonal elements

□ $n \times n$ 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_n 로 표시한다.
- 주 대각선이 전부 1이고, 나머지 원소는 0을 값으로 갖는 $n \times n$ 정방행렬이다.
- $MI = IM = M$

Vectors as Matrices

- n-dimension 벡터는 $1 \times n$ 행렬 또는 $n \times 1$ 행렬로 표현
 - $1 \times n$ 행렬은 행 벡터 (또는 행 행렬이라고도 부름)
 - $n \times 1$ 행렬은 열 벡터 (또는 열 행렬이라고도 부름)

$$A = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix} \quad A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \end{bmatrix}$$

Transpose Matrix

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

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

Transposing Matrix

$$\begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix}^T = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}^T = \begin{bmatrix} x & y & z \end{bmatrix}$$

Matrix Scalar Multiplication

- 행렬의 스칼라 곱 (Multiplying a matrix **M** with a scalar $\alpha = \alpha \mathbf{M}$)

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

Two Matrices Addition

- Matrix C는 A ($r \times c$ matrix)와 B ($r \times c$ matrix)의 덧셈은 $r \times c$ matrix이다.
- 각 원소 c_{ij} 는 A의 ij^{th} 원소와 B의 ij^{th} 원소의 합이다.
- $c_{ij} = a_{ij} + b_{ij}$

$$\begin{pmatrix} 1 & 3 & 6 \\ 10 & 0 & -5 \\ 4 & 7 & 2 \end{pmatrix}_{r \times c} + \begin{pmatrix} 3 & 7 & 1 \\ 6 & 4 & 9 \\ 8 & -9 & 4 \end{pmatrix}_{r \times c} = \begin{pmatrix} 4 & 10 & 7 \\ 16 & 4 & 4 \\ 12 & -2 & 6 \end{pmatrix}_{r \times c}$$

1+3

Two Matrices Multiplication

- Matrix C는 A (rxn matrix)와 B (nxc matrix)의 곱 $rx c$ 이다.
- 각 원소 c_{ij} 는 A의 i^{th} row와 B의 j^{th} column의 vector dot product이다.
- $c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$

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

3+18+48

must match columns in result rows in result

Multiplying Two Matrices

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \end{pmatrix}$$

$c_{24} = a_{21}m_{14} + a_{22}m_{24}$

Matrix Operation

- $MI = IM = M$ (I는 단위행렬)
- $A + B = B + A$: 행렬-행렬 덧셈의 교환법칙
- $A + (B + C) = (A + B) + C$: 행렬-행렬 덧셈의 결합법칙
- $AB \neq BA$: 행렬-행렬 곱의 교환법칙은 성립되지 않는다.
- $(AB)C = A(BC)$: 행렬-행렬 곱의 결합법칙
- $ABCDEF = (((AB)C)D)E)F = A(((BC)D)E)F = (AB)(CD)(EF)$
- $\alpha(AB) = (\alpha A)B = A(\alpha B)$: 스칼라-행렬 곱
- $\alpha(\beta A) = (\alpha\beta)A$: 스칼라-행렬 곱
- $(\alpha A)B = \alpha(AB)$
- $(AB)^T = B^T A^T$
- $(M_1 M_2 M_3 \dots M_{n-1} M_n)^T = M_n^T M_{n-1}^T \dots M_3^T M_2^T M_1^T$

Matrix Determinant

- 행렬식의 값 (The determinant of a square matrix M)은 $|M|$ 혹은 "**det M**" 으로 표시된다.
- Non-square matrix의 determinant 는 정의되지 않는다.

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

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

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* (수반행렬) of M , denoted "**adj M**"은 여인자 행렬의 전치행렬 (The transpose of the matrix of cofactors)

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

Cofactor of a Square Matrix & Computing Determinant using Cofactor

- 여인자 (Cofactor of M)는 정방행렬 M 의 i 행과 j 열을 제거하여 만든 소행렬(Minor of M)의 행렬식 (the signed determinant)
- $C_{ij} = (-1)^{i+j} |M^{(ij)}|$
- 코팩터 (cofactor)를 사용한 $n \times n$ determinant 계산:

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

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

Minor of a Matrix

- 소행렬 (Minor)는 정방행렬 M 의 i 행과 j 열을 제거하여 만든 부분행렬 $M^{(ij)}$

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

Determinant, Cofactor, Inverse Matrix

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

$$\det M = m_{11}m_{22} - m_{12}m_{21}$$

$$C = \begin{pmatrix} m_{22} & -m_{21} \\ -m_{12} & m_{11} \end{pmatrix}$$

$$\text{adj}M = \begin{pmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{pmatrix}$$

$$M^{-1} = \frac{1}{\det M} \begin{pmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{pmatrix}$$

Determinant, Cofactor, Inverse Matrix

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

$$\det M = m_{11}(m_{22}m_{33} - m_{23}m_{32}) - m_{12}(m_{21}m_{33} - m_{23}m_{31}) + m_{13}(m_{21}m_{32} - m_{22}m_{31})$$

$$C = \begin{pmatrix} (m_{22}m_{33} - m_{23}m_{32}) & -(m_{21}m_{33} - m_{23}m_{31}) & (m_{21}m_{32} - m_{22}m_{31}) \\ -(m_{12}m_{33} - m_{13}m_{32}) & (m_{11}m_{33} - m_{13}m_{31}) & -(m_{11}m_{32} - m_{12}m_{31}) \\ (m_{12}m_{23} - m_{13}m_{22}) & -(m_{11}m_{23} - m_{13}m_{21}) & (m_{11}m_{22} - m_{12}m_{21}) \end{pmatrix}$$

$$\text{adj}M = \begin{pmatrix} (m_{22}m_{33} - m_{23}m_{32}) & -(m_{12}m_{33} - m_{13}m_{32}) & (m_{12}m_{23} - m_{13}m_{22}) \\ -(m_{21}m_{33} - m_{23}m_{31}) & (m_{11}m_{33} - m_{13}m_{31}) & -(m_{11}m_{23} - m_{13}m_{21}) \\ (m_{21}m_{32} - m_{22}m_{31}) & -(m_{11}m_{32} - m_{12}m_{31}) & (m_{11}m_{22} - m_{12}m_{21}) \end{pmatrix}$$

$$M^{-1} = \frac{\text{adj}M}{\det M}$$

Multiplying a Vector and a Matrix

$$\begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} p_x & p_y & p_z \\ q_x & q_y & q_z \\ r_x & r_y & r_z \end{pmatrix} = \begin{pmatrix} xp_x + yq_x + zr_x & xp_y + yq_y + zr_y & xp_z + yq_z + zr_z \end{pmatrix} = xp + yq + zr$$

- 벡터-행렬의 곱을 이용하여 좌표계 변환 (coordinate space transformation)을 표현할 수 있다.

$$uM = v \quad // \quad \text{행렬 } M \text{이 벡터 } u \text{를 벡터 } v \text{로 변환}$$

Multiplying a Vector and a Matrix

- OpenGL (Column-Major Order)에서 벡터-행렬의 곱
 $\mathbf{v} = \mathbf{M} * \mathbf{u}$ // 행렬 M이 벡터 u를 벡터 v로 변환

$$\mathbf{v} = \mathbf{M} * \mathbf{u}$$

$$\begin{pmatrix} x m_{11} + y m_{12} + z m_{13} \\ x m_{21} + y m_{22} + z m_{23} \\ x m_{31} + y m_{32} + z m_{33} \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$