게임 수학 – 강의 4

동명대학교 게임공학과
강영민

벡터의 곱

- 세 종류의 곱
  - 아직 이해하기 힘든 부분은 그냥 넘어 가자
  - 스칼라 곱scalar product
    - 내적inner product, 점곱dot product

$$s = \mathbf{u} \cdot \mathbf{v} = \mathbf{u}^{\mathrm{T}}\mathbf{v}$$

  - 벡터곱vector product
    - 가위곱cross product, 외적outer product (이 수업에서는 외적이라 부르지 않음)

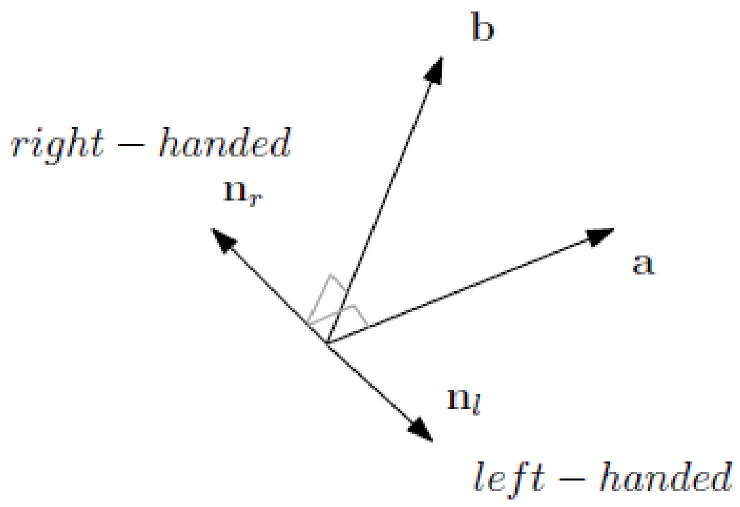
$$\mathbf{w} = \mathbf{u} \times \mathbf{v}$$

  - 텐서곱tensor product
    - 외적outer product (외적이라는 이름은 서로 다른 곱에 중복되어 사용)
    - 이 수업에서 외적은 텐서곱만을 의미

$$\mathbf{A} = \mathbf{u}\mathbf{v}^{\mathrm{T}}$$

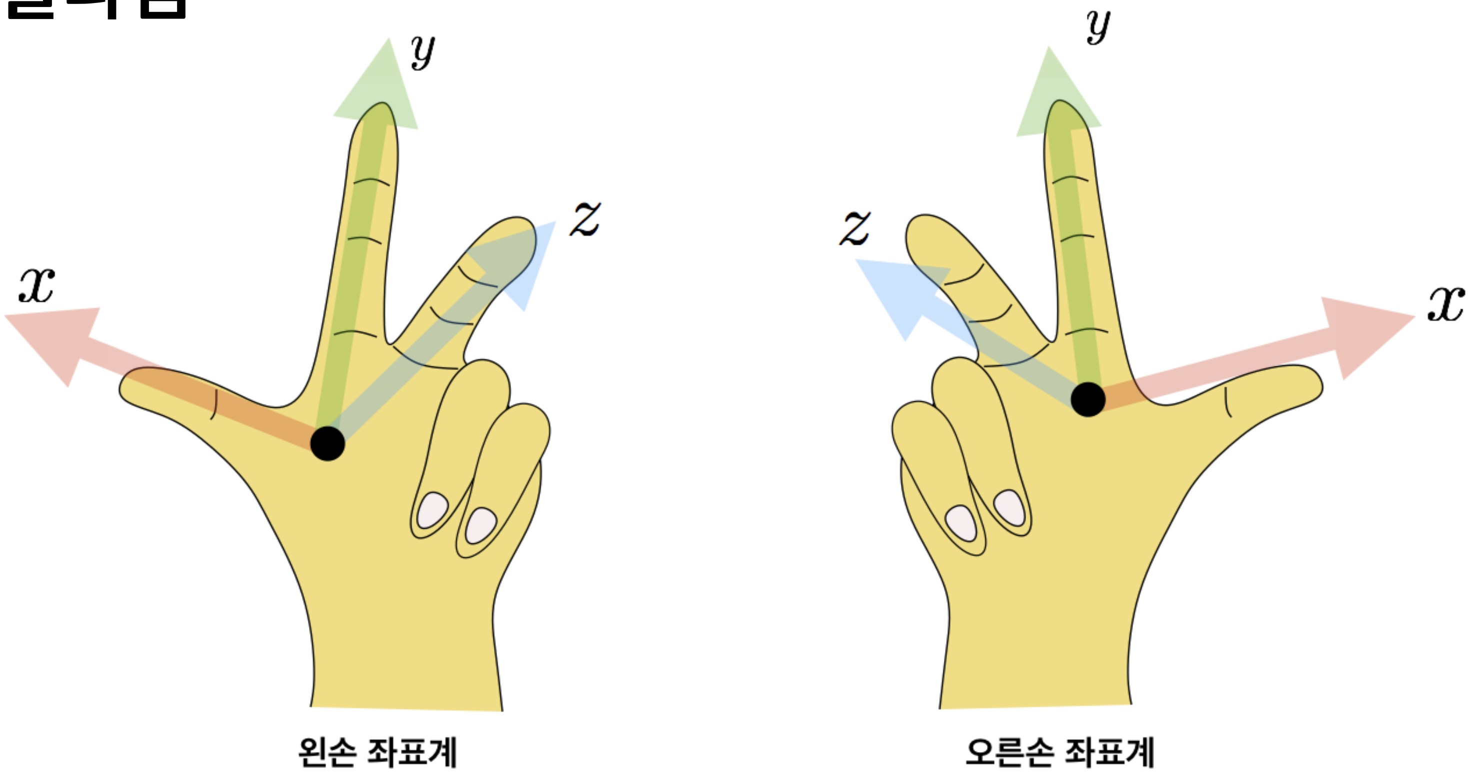
가위곱의 기하적 의미

- 벡터의 외적(cross product)
  - 벡터 곱(vector product): 두 벡터를 피연산자로 하는 이항연산으로 그 결과가 벡터
  - 벡터를 곱해 행렬을 얻는 외적(outer product)과 용어의 혼동이 있음. 여기서는 결과가 벡터인 곱
- 표현
  - 두 벡터 $\mathbf{a}$와 $\mathbf{b}$의 외적은 $\mathbf{a} \times \mathbf{b}$로 표현
  - 그 결과는 벡터이므로 $k\mathbf{n}$ ($\mathbf{n}$은 $\mathbf{a}$와 $\mathbf{b}$에 동시에 수직인 단위벡터)
  - 동시에 수직인 벡터는 두 개가 존재. 좌표계에 의해 결정됨.



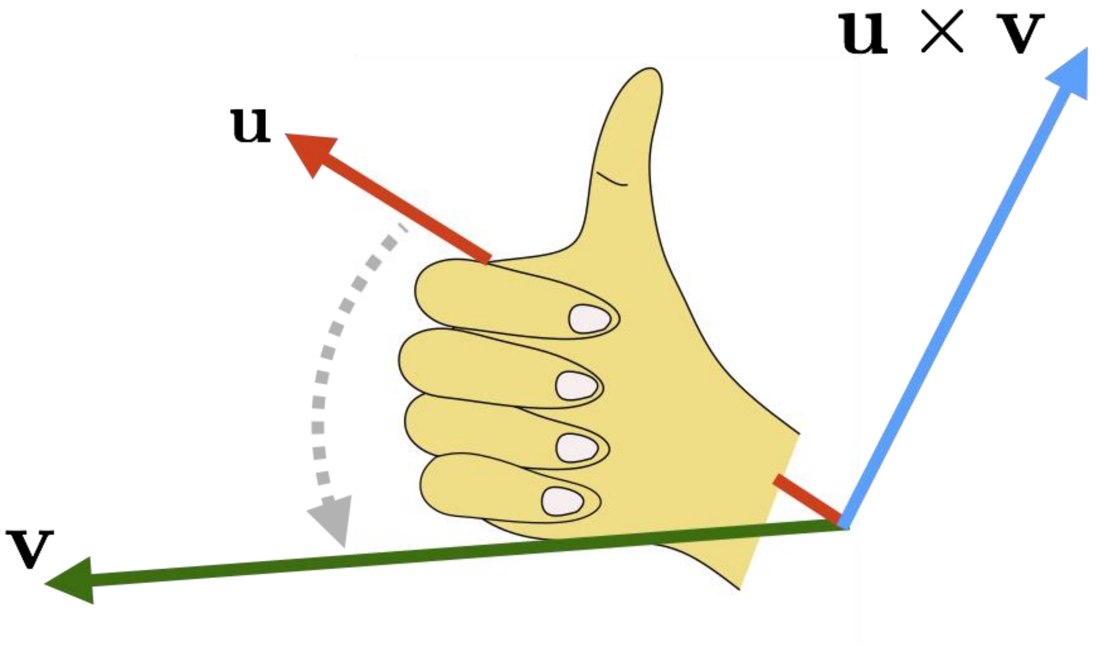
가위곱의 방향

- 좌표계에 따라 달라짐



가위곱 방향

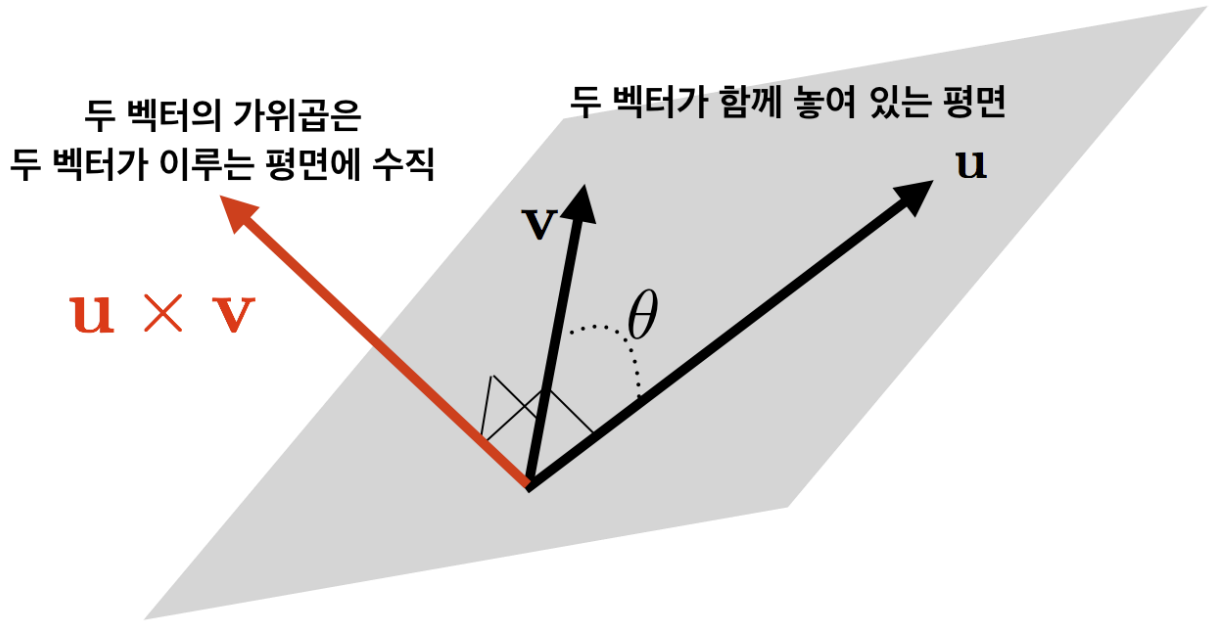
- 오른손 법칙
  - 가위곱의 피연산자가 되는 벡터 둘에 대해 피연산자로 먼저 나타나는 벡터에서 두 번째 피연산자 벡터 쪽으로 오른손 손바닥으로 감싸쥐었을 때에 펼쳐진 엄지가 가리키는 방



가위곱의 기하적 의미

- 오른손 좌표계를 사용할 때
  - 방향은 그러한데, 그 크기는 얼마일까?

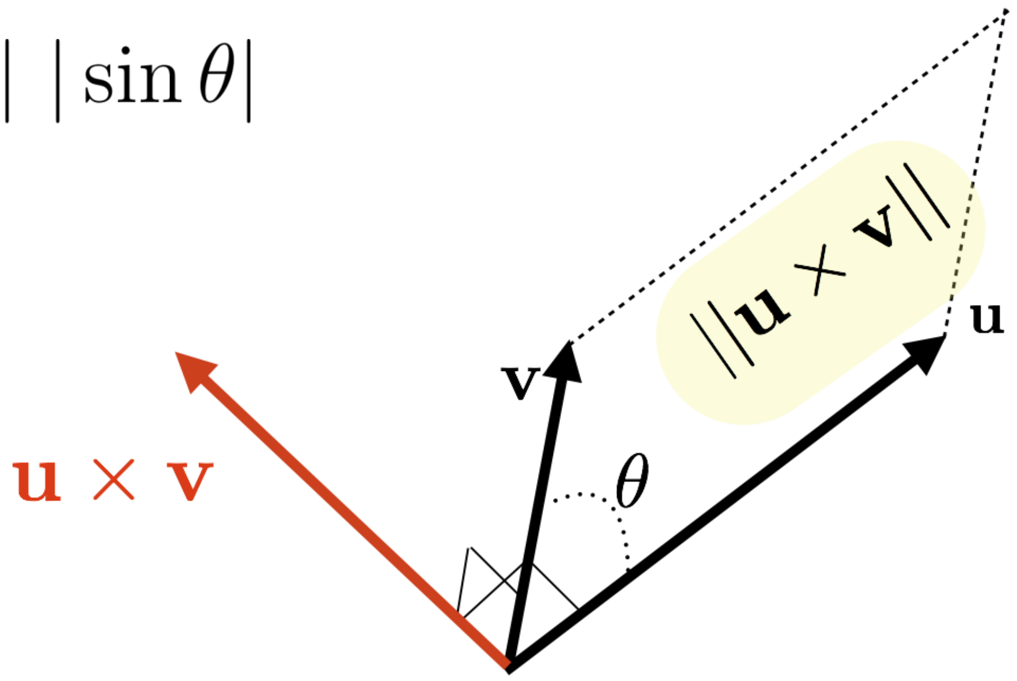
$$||\mathbf{u} \times \mathbf{v}||$$



가위곱 결과 벡터의 크기

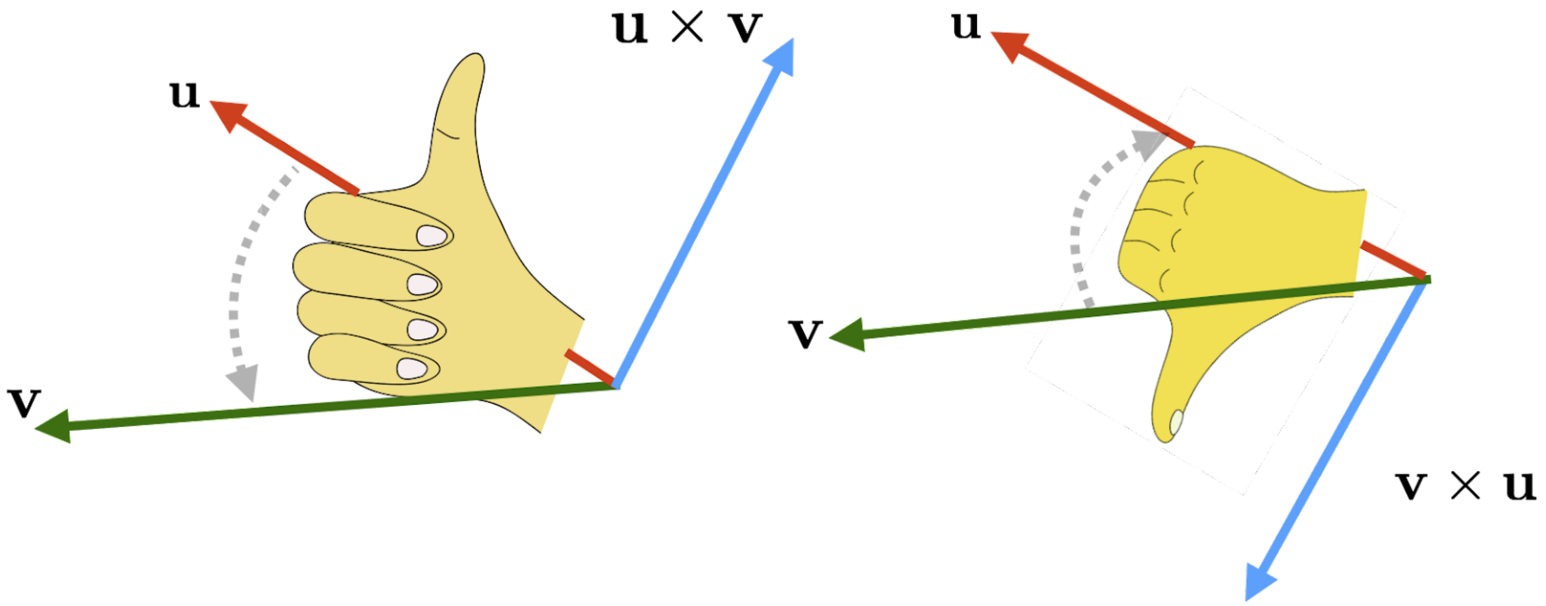
- 두 벡터가 만들어내는 사이 공간의 크기
  - 두 벡터의 합을 계산할 때 그려보는 평행사변형의 넓이

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



가위곱

- 교환법칙이 성립하지 않는다



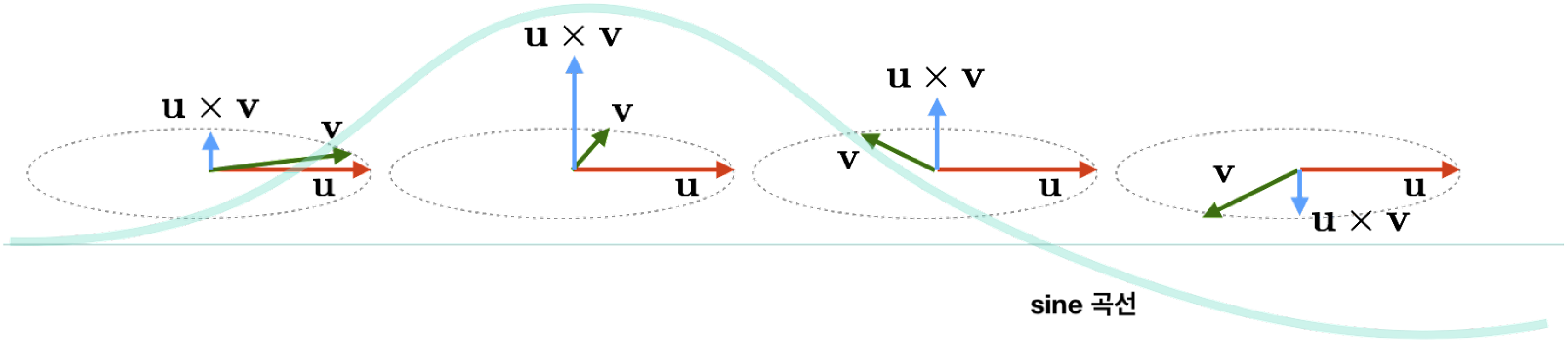
가위곱의 성질

- u, v 사이의 각도에 따라

θ	점곱	가위곱
0 (같은 방향)	$\mathbf{u} \cdot \mathbf{v} = \|\|\mathbf{u}\|\| \; \|\|\mathbf{v}\|\|$	$\mathbf{u} \times \mathbf{v} = \mathbf{0}$
$\pi/2$ (직교)	$\mathbf{u} \cdot \mathbf{v} = 0$	$\|\|\mathbf{u} \times \mathbf{v}\|\| = \|\|\mathbf{u}\|\| \; \|\|\mathbf{v}\|\|$
언제나	$\mathbf{u} \cdot \mathbf{u} = \|\|\mathbf{u}\|\|^2$ $\mathbf{v} \cdot \mathbf{v} = \|\|\mathbf{v}\|\|^2$	$\mathbf{u} \times \mathbf{u} = \mathbf{0}$ $\mathbf{v} \times \mathbf{v} = \mathbf{0}$

가위곱의 성질

- u, v 사이의 각도에 따라 그 크기가 변화



가위곱의 계산

- 계산법

$$\mathbf{u} = (u_1, u_2, u_3) \qquad \mathbf{v} = (v_1, v_2, v_3)$$

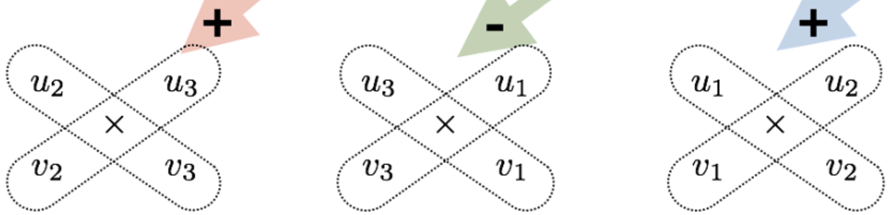
$$\mathbf{u} \times \mathbf{v} = (u_2 v_3 - u_3 v_2, u_3 v_1 - u_1 v_3, u_1 v_2 - u_2 v_1)$$

가위곱의 계산

- 계산법

$$\begin{array}{lccccc} \mathbf{u} & (\,u_1 & u_2 & u_3\,) & u_1 & u_2 \\ \mathbf{v} & (\,v_1 & v_2 & v_3\,) & v_1 & v_2 \end{array}$$

$\mathbf{u} \times \mathbf{v}$



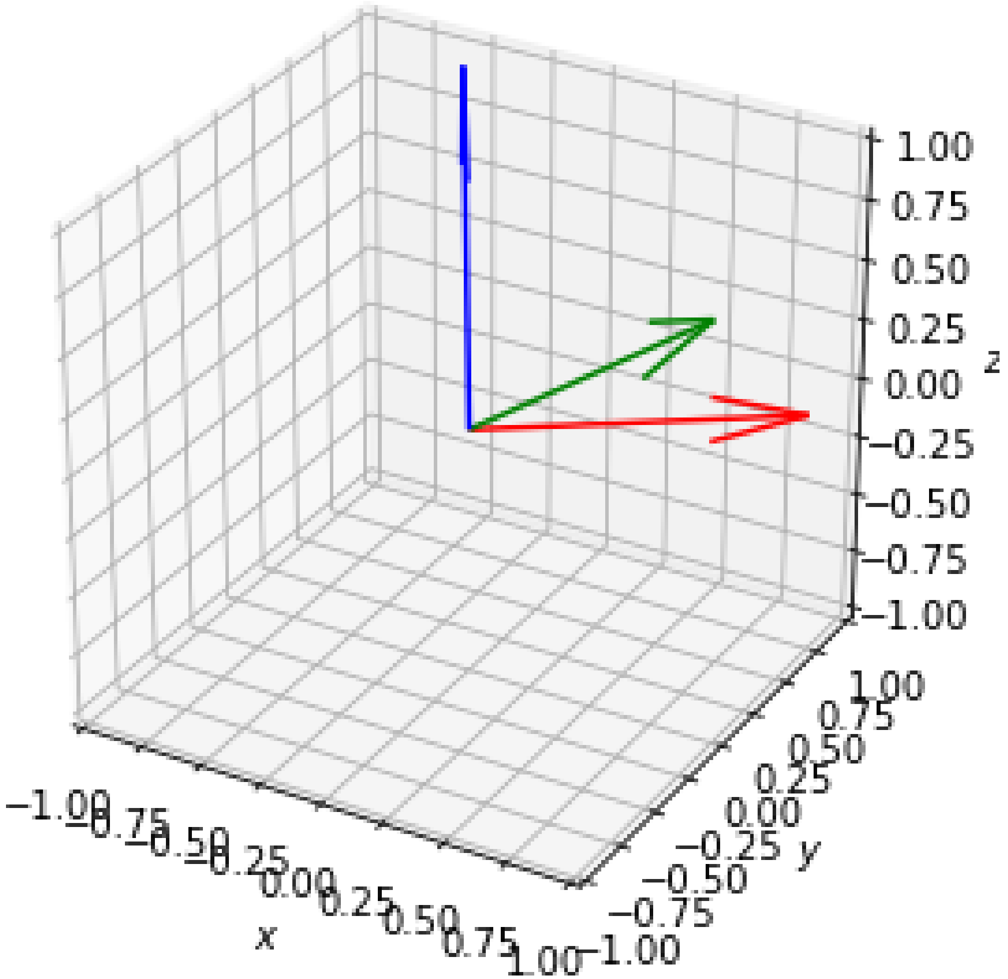
$$\left(u_2v_3 - u_3v_2 \qquad -(u_3v_1 - u_1v_3) \qquad u_1v_2 - u_2v_1 \right)$$

$$\left(u_2v_3 - u_3v_2 \qquad u_1v_3 - u_3v_1 \qquad u_1v_2 - u_2v_1 \right)$$

컴퓨터로 계산해 보기

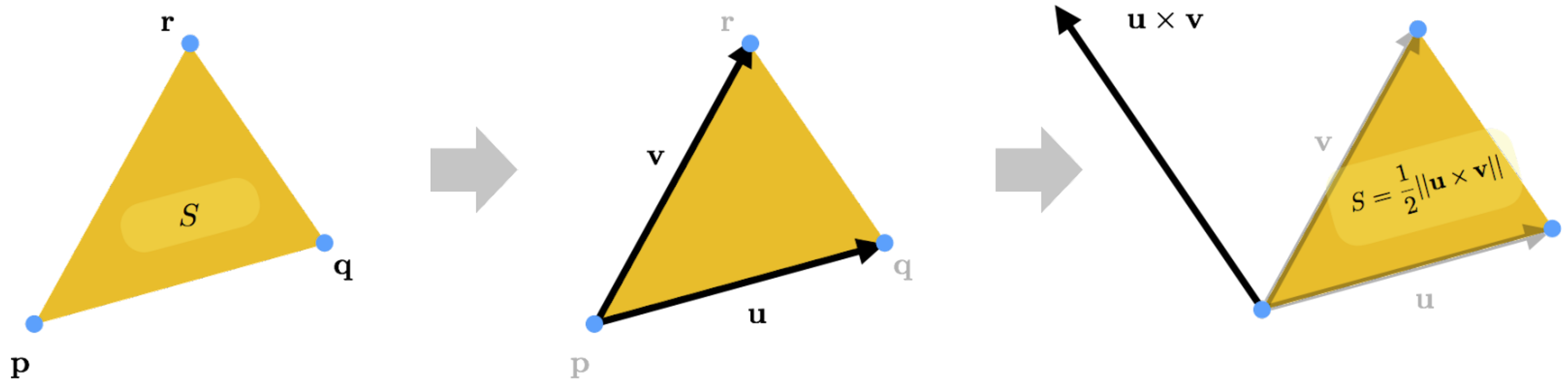
```
u = np.array([1.0, 0.7, 0.0])
v = np.array([0.2, 1.5, −.2])

e0 = u[1]*v[2] − u[2]*v[1]      # 가위곱의 첫 원소
e1 = u[0]*v[2] − u[0]*v[2]      # 가위곱의 두번째 원소
e2 = u[0]*v[1] − u[1]*v[0]      # 가위곱의 세번째 원소
uxv = np.array([e0, e1, e2])    # 가위곱으로 얻는 벡터
```



가위곱의 응용

- 삼각형의 면적 계산

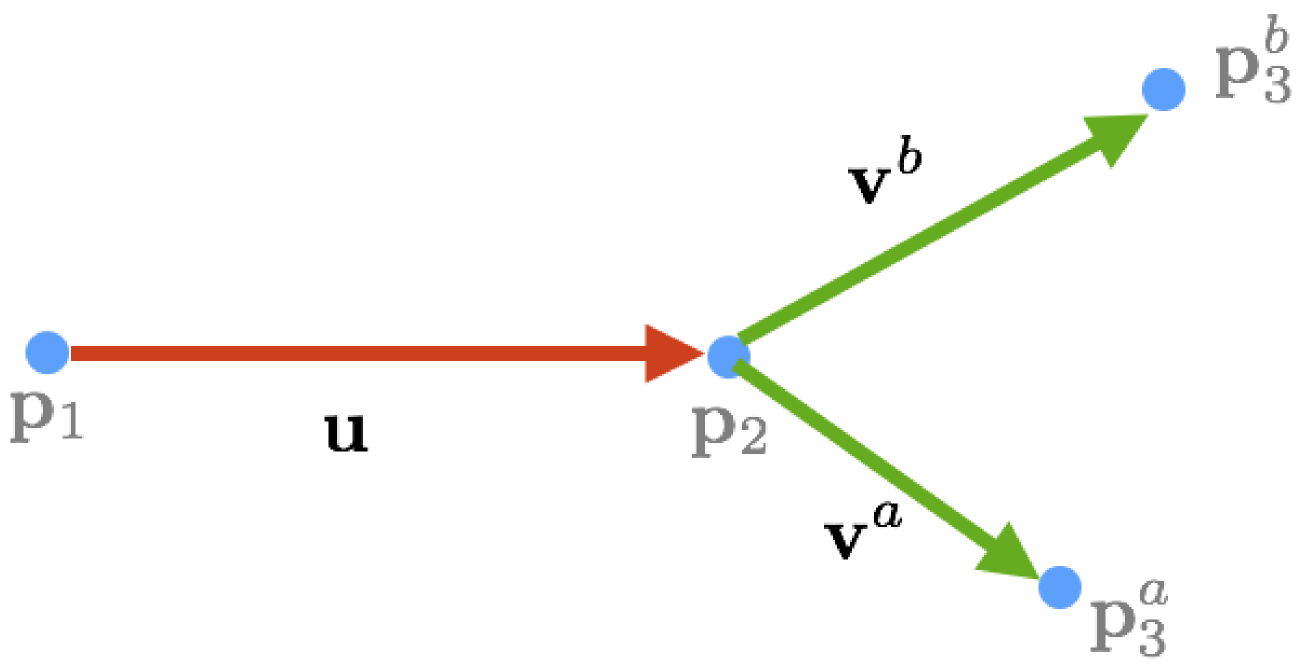


2차원 공간의 삼각형

- 가위곱은 언제나 z축 방향 (+/−)
  - $u = (u_x, u_y) \rightarrow u = (u_x, u_y, 0)$
  - $v = (v_x, v_y) \rightarrow v = (v_x, v_y, 0)$
  - $u \times v = (\ 0, 0, u_x v_y - u_y v_x\ )$ ← 언제나 z축 성분만 존재
- z축 성분의 값을 signed area라 부름
  - 부호에 따라 두 벡터가 시계/반시계 방향 어느쪽으로 배치되는지 확인 가능
  - 이 성분의 값이 두 벡터가 만드는 삼각형의 크기

진행 방향 체크

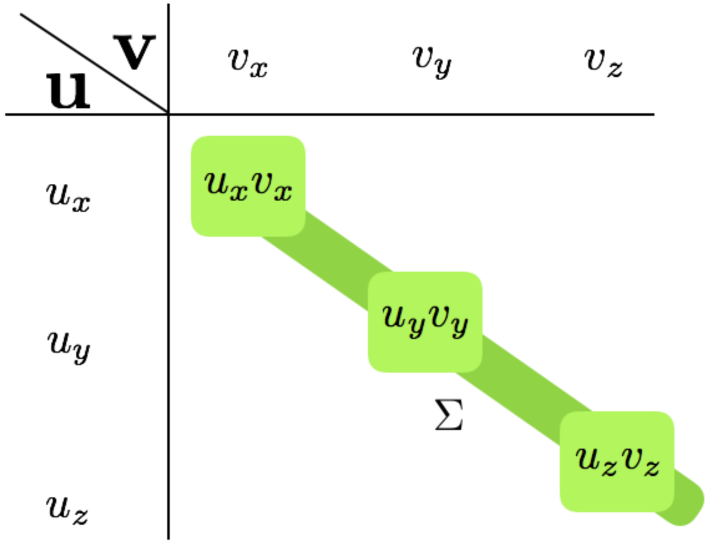
- 벡터가 왼쪽으로 휘는지 오른쪽으로 휘는지 검사



점곱과의 비교

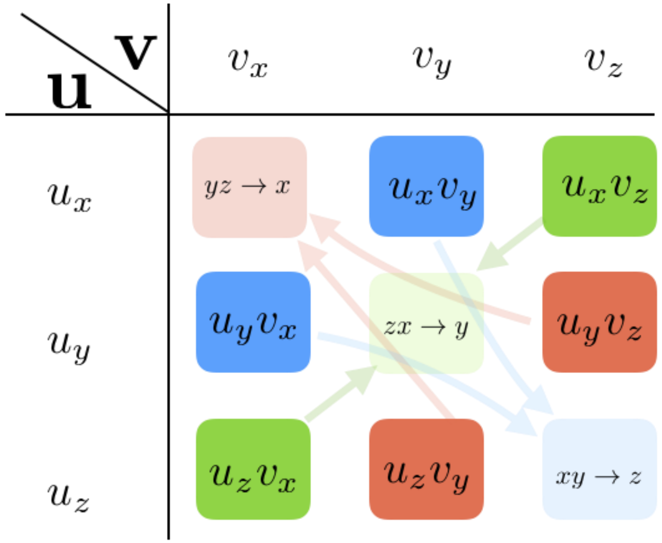
- 점곱의 크기: 얼마나 비슷하나
- 가위곱의 크기: 얼마나 어긋났나

얼마나 비슷한가



$$\mathbf{u} \cdot \mathbf{v} = u_xv_x + u_yv_y + u_zv_z$$

얼마나 어긋났나

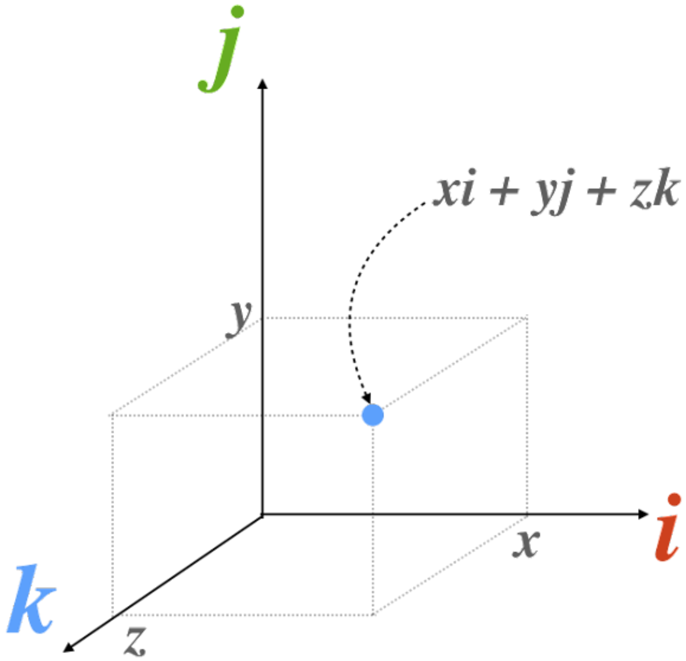


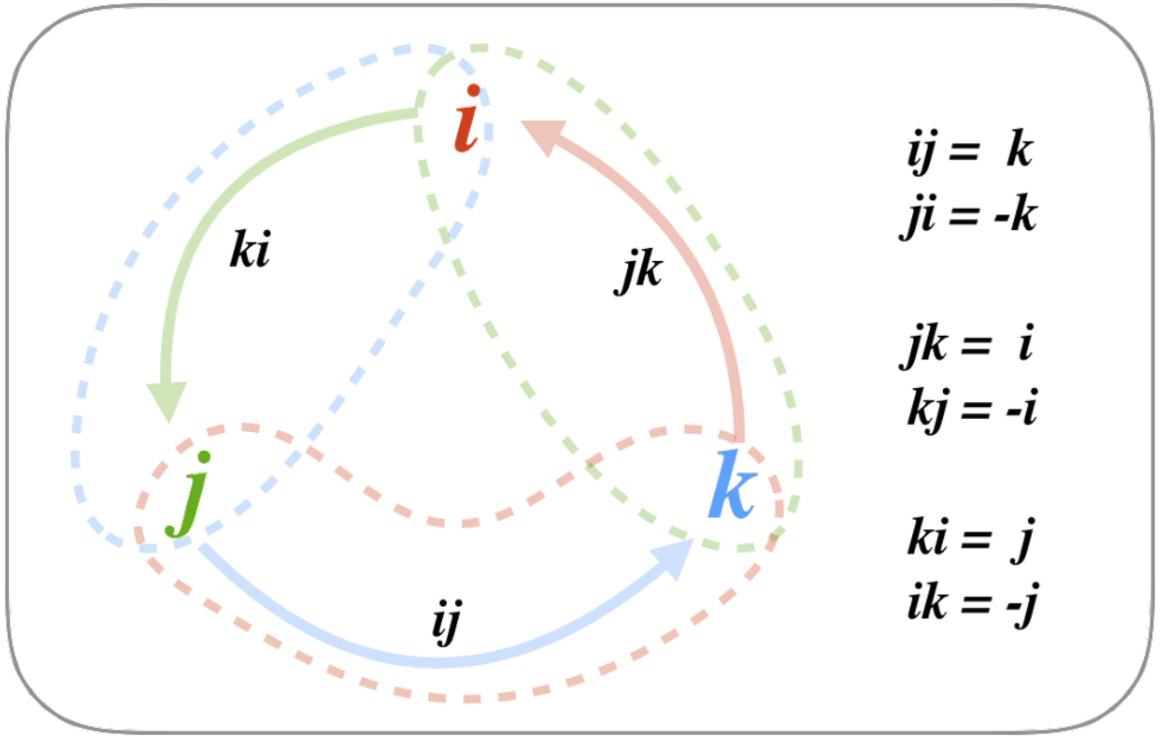
$$\mathbf{u} \times \mathbf{v} = (u_yv_z - u_zv_y, \quad u_zv_x - u_xv_z, \quad u_xv_y - u_yv_x)$$

복소수의 곱으로 이해하는 점곱과 가위곱

- 벡터를 복소수로 표현하기

$$(x, y, z) = xi + yj + zk$$





복소수의 곱으로 이해하는 점곱과 가위곱

$$\begin{aligned}
(u_x i + u_y j + u_z k)(v_x i + v_y j + v_z k) &= u_x v_x i^2 + u_x v_y ij + u_x v_z ik \\
&\quad + u_y v_x ji + u_y v_y j^2 + u_y v_z jk \\
&\quad + u_z v_x ki + u_z v_y kj + u_z v_z k^2 \\
&= -(u_x v_x + u_y v_y + u_z v_z) \\
&\quad + u_x v_y ij + u_x v_z ik \\
&\quad + u_y v_x ji + u_y v_z jk \\
&\quad + u_z v_x ki + u_z v_y kj \\
&= -(u_x v_x + u_y v_y + u_z v_z) \\
&\quad + (u_y v_z - u_z v_y) i \\
&\quad + (u_z v_x - u_x v_z) j \\
&\quad + (u_x v_y - u_y v_x) k
\end{aligned}$$

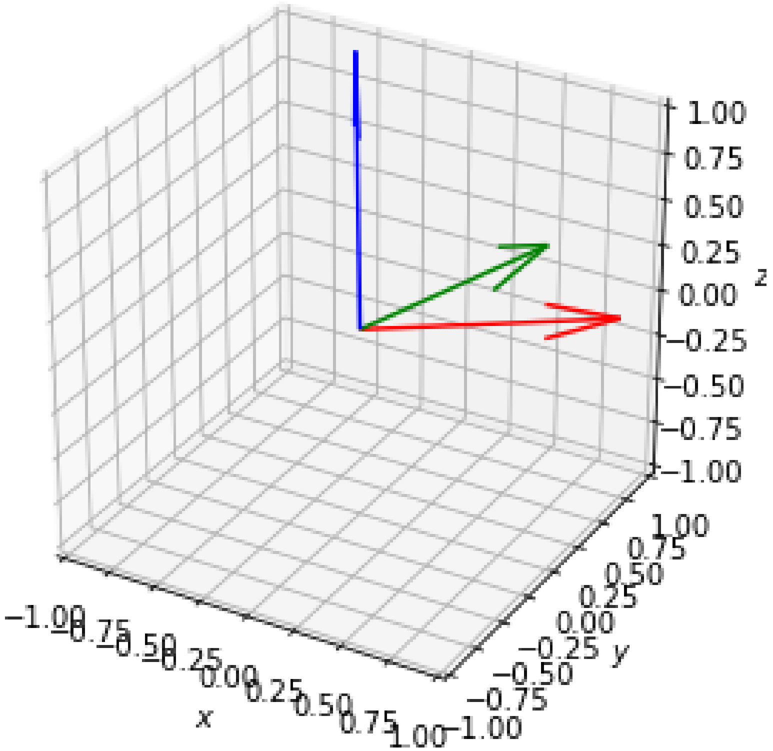
가위곱의 연산 규칙

$$\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$$
$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \times \mathbf{b}) \times \mathbf{c}$$
$$(\mathbf{a} + \mathbf{b}) \times \mathbf{c} = (\mathbf{a} \times \mathbf{c}) + (\mathbf{b} \times \mathbf{c})$$
$$k(\mathbf{a} \times \mathbf{b}) = k\mathbf{a} \times \mathbf{b} = \mathbf{a} \times k\mathbf{b}$$
$$\mathbf{a} \times \vec{0} = \vec{0} \times \mathbf{a} = \vec{0}$$
$$\mathbf{a} \times \mathbf{a} = \vec{0}$$
$$\mathbf{a} \cdot (\mathbf{a} \times \mathbf{b}) = \vec{0}$$
$$\mathbf{b} \cdot (\mathbf{a} \times \mathbf{b}) = \vec{0}$$

넘파이를 이용한 가위곱 계산

```
u = np.array([1.0, 0.7, 0.0])
v = np.array([0.2, 1.5, −.2])


uxv = np.cross(u, v)
```



문제

- 어떤 평면 위에 세 점을 안다: 파란 점 (p1, p2, p3)
- 평면에서 떨어진 어떤 점이 있다: 붉은 점 (p4)
- 문제: 이 붉은 점과 평면 사이의 거리는?

```
u = p2-p1
v = p3-p1
w = p4-p1

uxv = u.cross(v)
n = uxv  / np.linalg.norm(uxv)
d = n.dot(w)
```

Why?