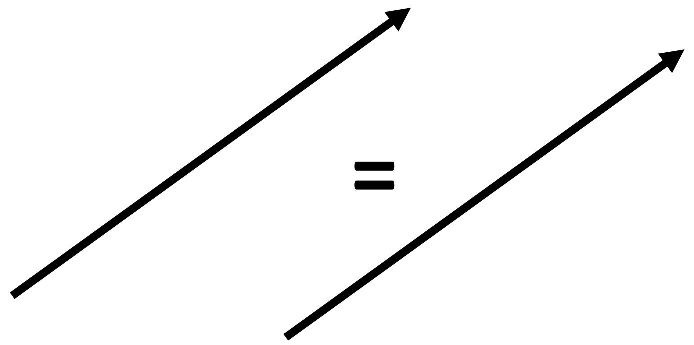
COMPUTER GRAPHICS

Review of Basic Math

Dr. Zhonggui Chen
School of Informatics, Xiamen University
http://graphics.xmu.edu.cn

Vectors

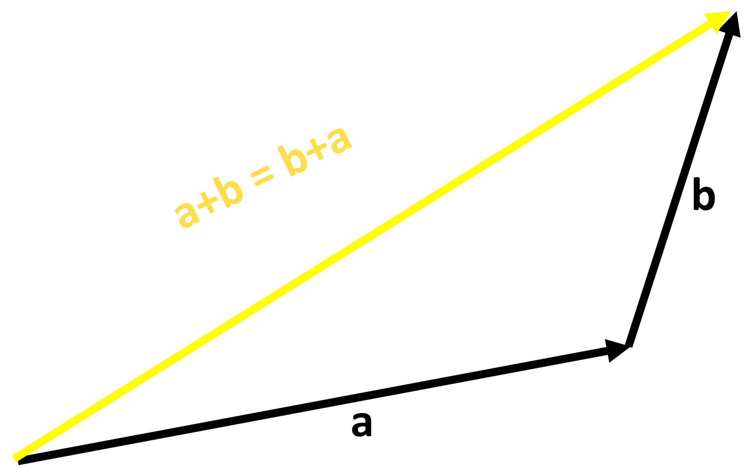


- Length and direction. Absolute position not important

  Usually written as $\vec{a}$ or in *italics*. Magnitude written as $\|\vec{a}\|$

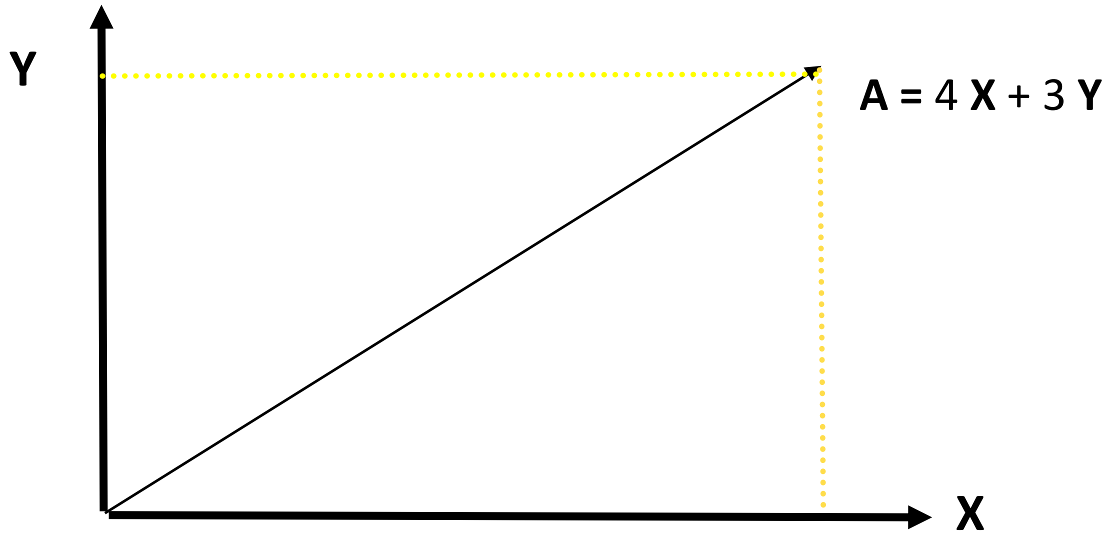
- Use to store offsets, displacements, locations
  - But strictly speaking, positions are not vectors and cannot be added: a location implicitly involves an origin, while an offset does not.

Vector Addition



- Geometrically: Parallelogram rule
- In Cartesian coordinates (next), simply add coords

Cartesian Coordinates



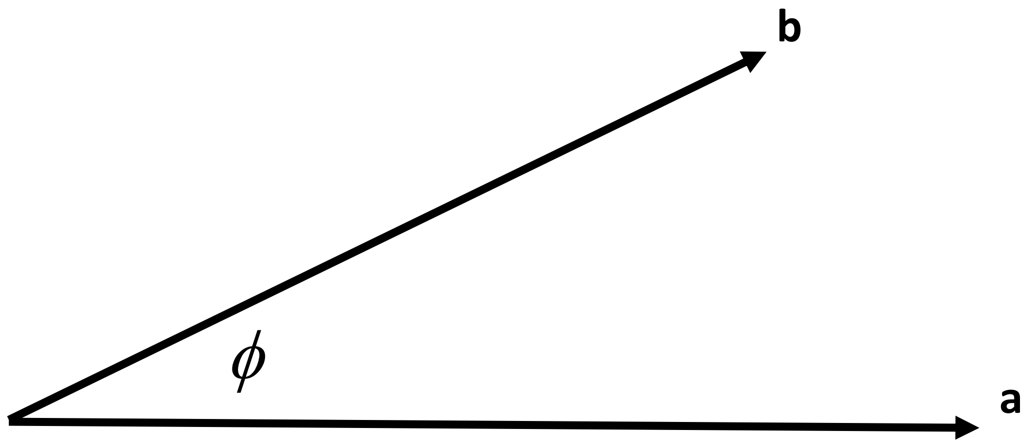
- X and Y can be any (usually orthogonal ***unit***) vectors

$$A = \begin{pmatrix} x \\ y \end{pmatrix} \qquad A^T = \begin{pmatrix} x & y \end{pmatrix} \qquad \|A\| = \sqrt{x^2 + y^2}$$

Vector Multiplication

- *Dot product*
- Cross product
- Orthonormal bases and coordinate frames

Dot (scalar) product

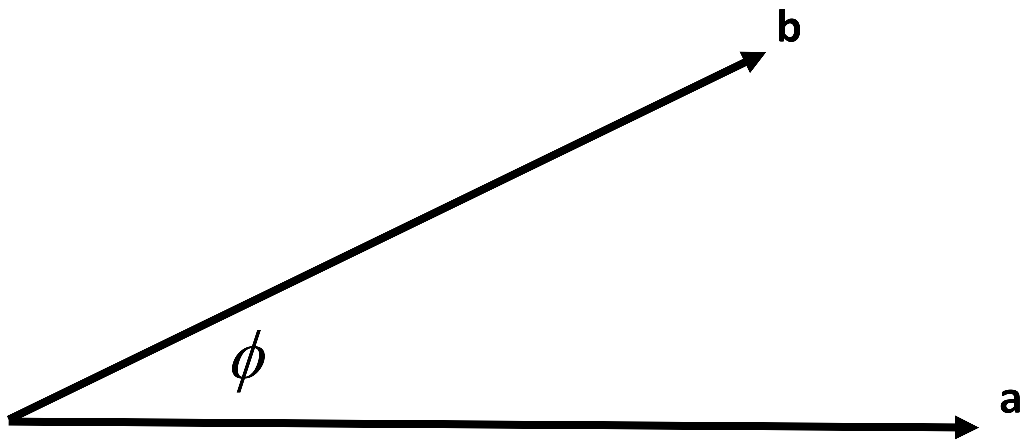


$$\mathrm{a}\cdot b = b\cdot a$$

$$\mathrm{a}\cdot(b+c) = \mathrm{a}\cdot b + a\cdot c$$

$$(ka)\cdot b = a\cdot(kb) = k(a\cdot b)$$

Dot (scalar) product

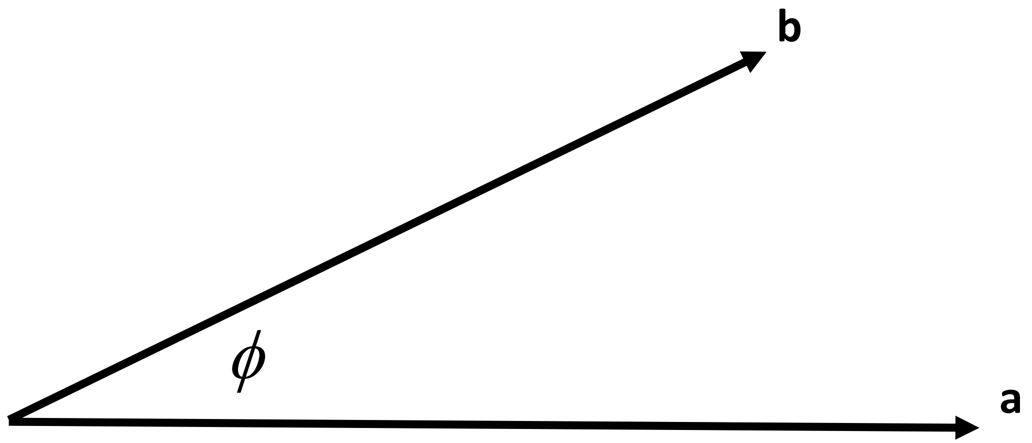


$$\mathrm{a}\cdot b = b\cdot a = ?$$

$$\mathrm{a}\cdot(b+c) = \mathrm{a}\cdot b + a\cdot c$$

$$(ka)\cdot b = a\cdot(kb) = k(a\cdot b)$$

Dot (scalar) product



$$\mathrm{a}\cdot b = b\cdot a = ?$$

$$\mathrm{a}\cdot(b+c) = \mathrm{a}\cdot b + a\cdot c$$

$$(ka)\cdot b = a\cdot(kb) = k(a\cdot b)$$

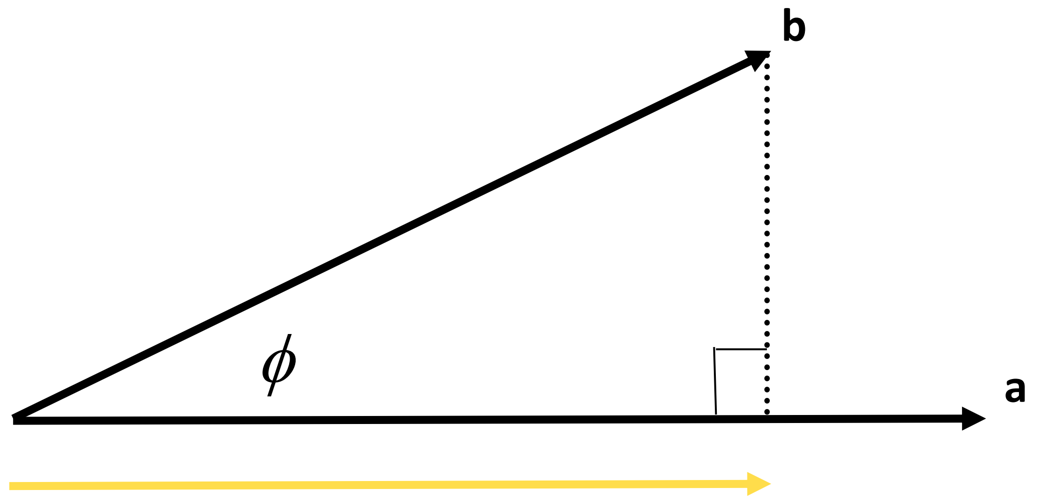
$$\mathrm{a}\cdot b = \|a\|\|b\|\cos\phi$$

$$\phi = \arccos\left(\frac{a\cdot b}{\|a\|\|b\|}\right)$$

Dot product: some applications in CG

- Find angle between two vectors (e.g. cosine of angle between light source and surface for shading)
- Finding projection of one vector on another (e.g. coordinates of point in arbitrary coordinate system)
- Advantage: can be computed easily in Cartesian components

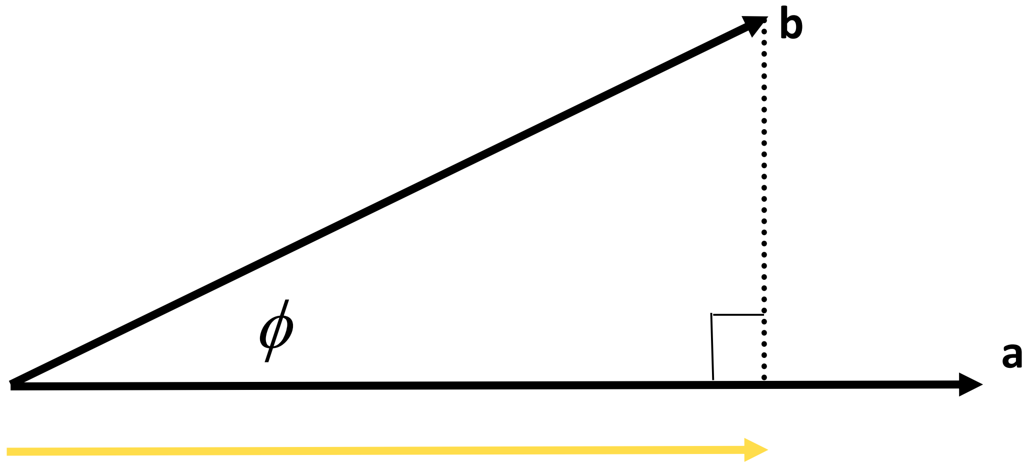
Projections (of b on a)



$$\|b \to a\| = ?$$

$$b \to a = ?$$

Projections (of b on a)



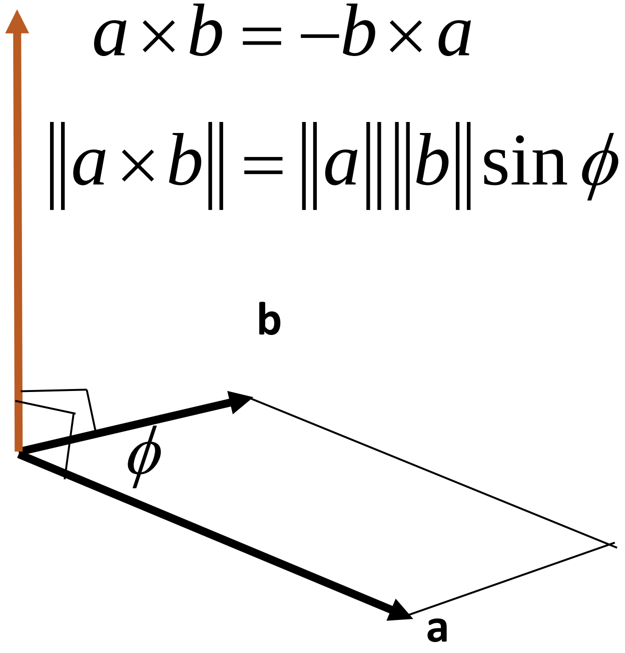
$$\|b \to a\| = ?$$

$$\|b \to a\| = \|b\| \cos\phi = \frac{a \cdot b}{\|a\|}$$

$$b \to a = ?$$

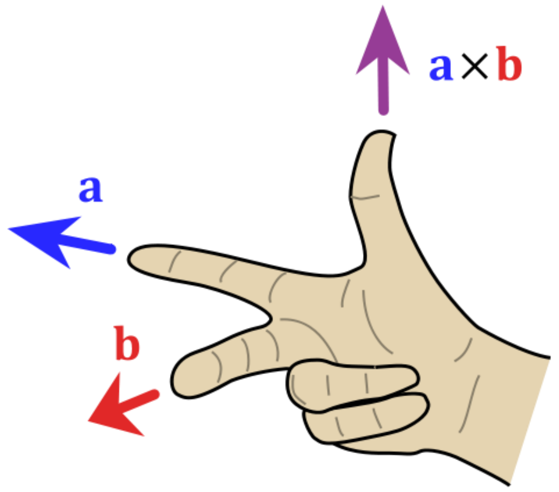
$$b \to a = \|b \to a\| \frac{a}{\|a\|} = \frac{a \cdot b}{\|a\|^2} a$$

Cross (vector) product



$$a \times b = -b \times a$$

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



- Cross product orthogonal to two initial vectors
- Direction determined by right-hand rule

Cross product: Properties

$$x \times y = +z$$
$$y \times x = -z$$
$$y \times z = +x$$
$$z \times y = -x$$
$$z \times x = +y$$
$$x \times z = -y$$

$$a \times b = -b \times a$$
$$a \times a = 0$$
$$a \times (b + c) = a \times b + a \times c$$
$$a \times (kb) = k(a \times b)$$

Cross product: Cartesian formula

$$a = (x_a, y_a, z_a),\ b = (x_b, y_b, z_b)$$

$$a \times b = \begin{vmatrix} X & Y & Z \\ x_a & y_a & z_a \\ x_b & y_b & z_b \end{vmatrix} = \begin{pmatrix} y_a z_b - y_b z_a \\ z_a x_b - x_a z_b \\ x_a y_b - y_a x_b \end{pmatrix}$$

$$a \times b = A^* b = \begin{pmatrix} 0 & -z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix}$$

Dual matrix of vector *a*

Exercises

Exercise 1. Compute the direction of the top of the camera for the viewing transformation: gluLookAt(0.0, 5.0, 5.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0)

Exercise 2. What is the viewing transformation equivalent to the following sequence of modeling transformations:

```
glRotatef(45.0, 0.0, 1.0, 0.0);
glTranslatef(0.0, 0.0, -5.0);
```

Exercise 3. A programmer writes the following at the top of his drawing routine:

```
glRotatef(45.0, 0.0, 1.0, 0.0);
gluLookAt(10.0, 0.0, 10.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
```

However, a transformation preceding the gluLookAt() statement in the drawing routine is bad practice – gluLookAt() should be the first transformation. So, combine the above two statements into one gluLookAt() having the same effect.

Exercises

Exercise . (Programming) Enhance \Chapter10\OBJModelViewer\OBJmodelViewer.cpp with translation and scaling commands whose parameters, determined from the bounding box of the object specified by the OBJ file, cause that object to be centralized in the viewing frustum.

No need to submit the solutions!