CT5510: Computer Graphics

Transformation

BOCHANG MOON

2D Translation

- Transformations such as rotation and scale can be represented using a matrix M
  - $e.g., \; M = SR$
  - $x' = m_{11}x + m_{12}y$
  - $y' = m_{21}x + m_{22}y$
- How about translation?
  - $x' = x + x_\delta$
  - $y' = y + y_\delta$
  - No way to express this using a 2 x 2 matrix

Homogenous Coordinates

- Affine transformation
  - Preserve points, straight lines, and planes after a transformation
  - e.g., scale, rotation, translation, reflect, shear
- Represent the point (x, y) by a 3D vector $[x, y, 1]^t$
  - Add an extra dimension
- Use the following matrix form to implement affine transformations
- $M = \begin{bmatrix} m_{11} & m_{12} & x_\delta \\ m_{21} & m_{22} & y_\delta \\ 0 & 0 & 1 \end{bmatrix}$

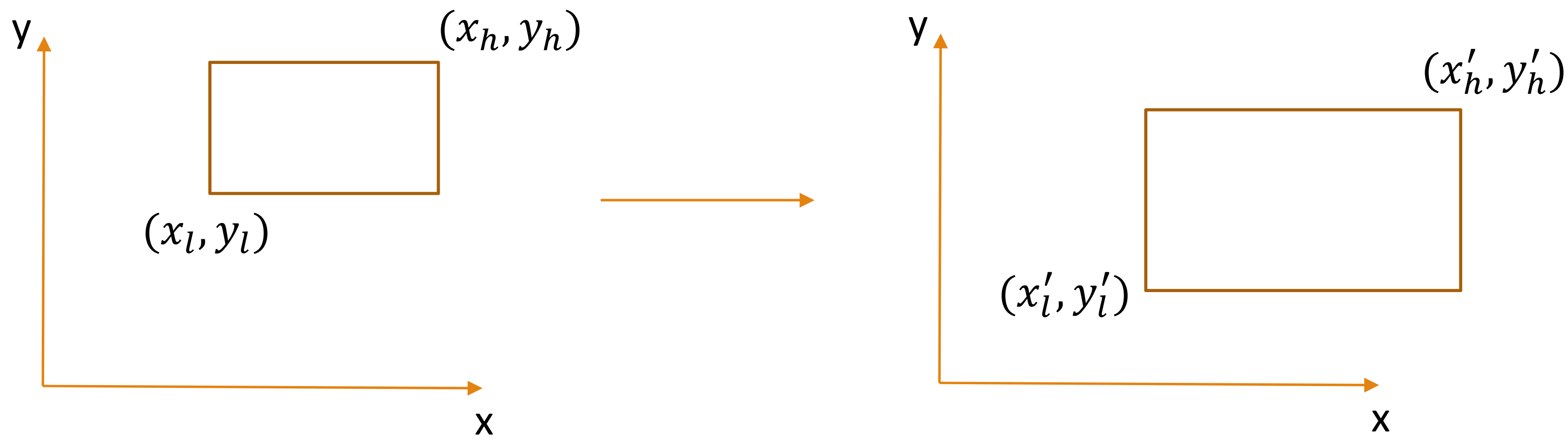
Homogenous Coordinates

- Compactly represent multiple affine transformations (including translations) with a matrix

- e.g., 2D translation
  - $$\begin{bmatrix} 1 & 0 & x_\delta \\ 0 & 1 & y_\delta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_\delta \\ y + y_\delta \\ 1 \end{bmatrix}$$
- e.g., rotation after 2D translation
  - $$M = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & x_\delta \\ 0 & 1 & y_\delta \\ 0 & 0 & 1 \end{bmatrix}$$

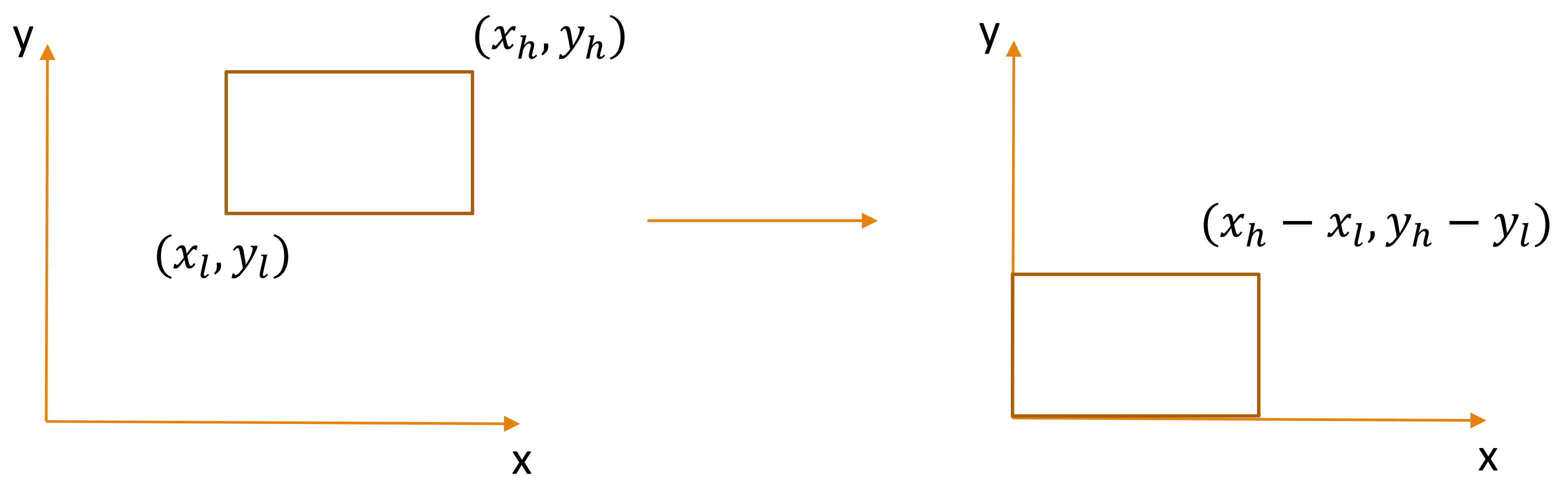
Examples

- Problem specification: move a 2D rectangle into a new position



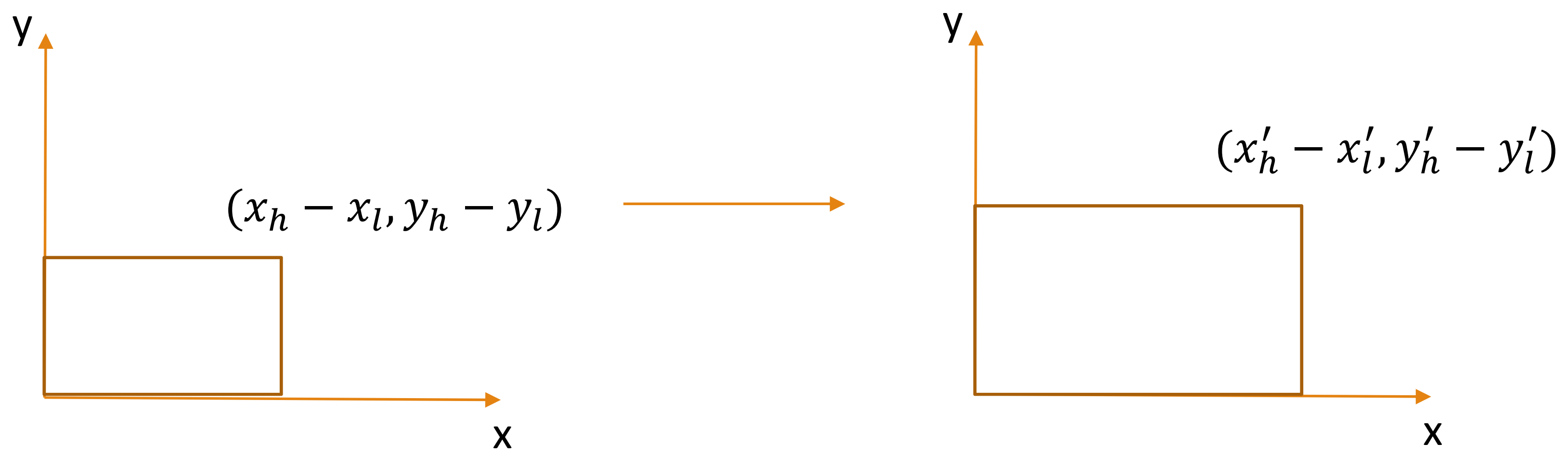
Examples

- Problem specification: move a 2D rectangle into a new position
  - Step1. translate: move the point (x_l, y_l) to the origin



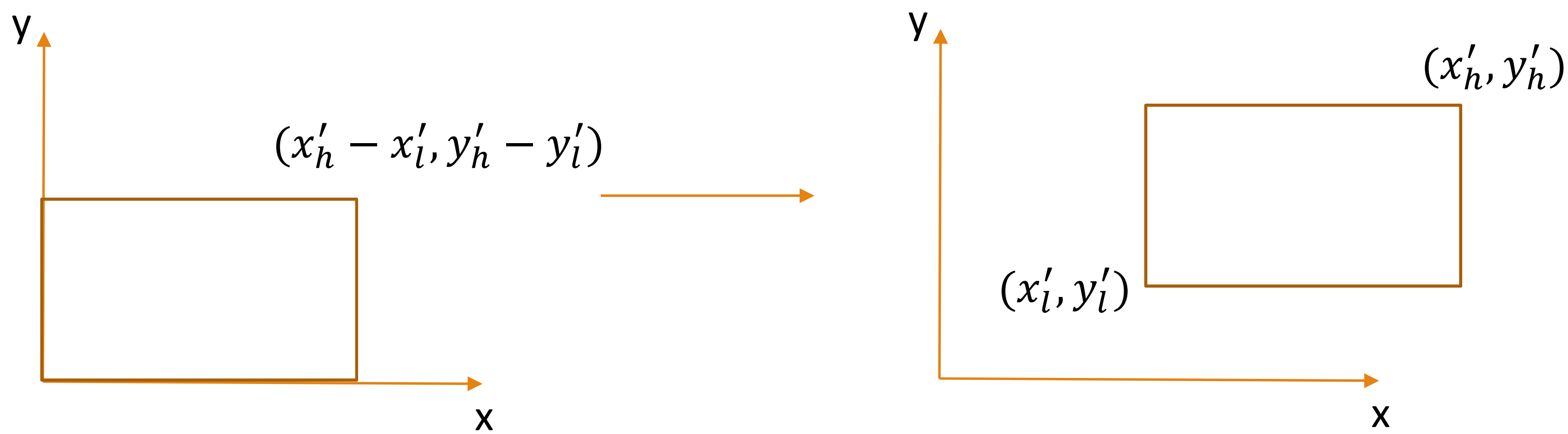
Examples

- Problem specification: move a 2D rectangle into a new position
  - Step2. scale: resize the rectangle to be the same size of the target.



Examples

- Problem specification: move a 2D rectangle into a new position
  - Step3. translate: move the origin to point (x_l', y_l')



Examples

- Problem specification: move a 2D rectangle into a new position
  - Target = translate(x'_l, y'_l) $scale\left(\frac{x'_h - x'_l}{x_h - x_l}, \frac{y'_h - y'_l}{y_h - y_l}\right)$ $translate(-x_l, -y_l)$
  - $= \begin{bmatrix} 1 & 0 & x'_l \\ 0 & 1 & y'_l \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x'_h - x'_l}{x_h - x_l} & 0 & 0 \\ 0 & \frac{y'_h - y'_l}{y_h - y_l} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_l \\ 0 & 1 & -y_l \\ 0 & 0 & 1 \end{bmatrix}$
  - $= \begin{bmatrix} \frac{x'_h - x'_l}{x_h - x_l} & 0 & \frac{x'_l x_h - x'_h x_l}{x_h - x_l} \\ 0 & \frac{y'_h - y'_l}{y_h - y_l} & \frac{y'_l y_h - y'_h y_l}{y_h - y_l} \\ 0 & 0 & 1 \end{bmatrix}$

Rigid-body Transformation

- A transformation that preserves distances between every pair of points
  - Are composed only of translations and rotations
  - i.e., no stretching or shrinking of the objects

Discussion

- Homogenous coordinates are common for graphics applications. Why?

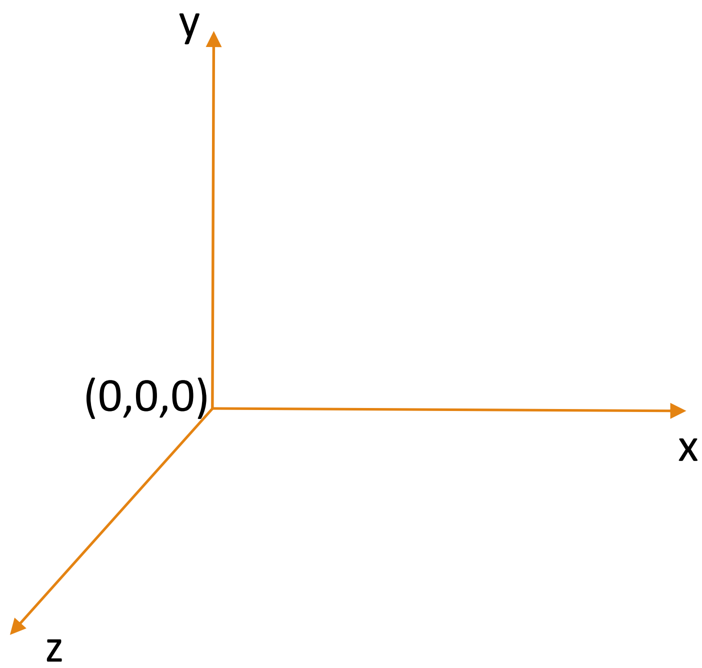
- A naïve way of implementing translations is to move the positions directly without forming a matrix

3D Transformation

- Extension of 2D transformation
- Why 3D transformation?
  - You virtual world is a 3D world.
  - 3D transformations are fundamental units to form your virtual scene.

OpenGL 3D coordinate

- Right-hand Coordinate System (RHS)
  - Counter-clockwise



3D Transformation (scale)

- $scale(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$

- Representation with homogeneous coordinates
  - $\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

3D Transformation (translation)

- $translate(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

3D Transformation (rotation)

- $rotate - z(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $rotate - y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}$
- $rotate - x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}$
- Representation with homogeneous coordinates
- $rotate - z(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

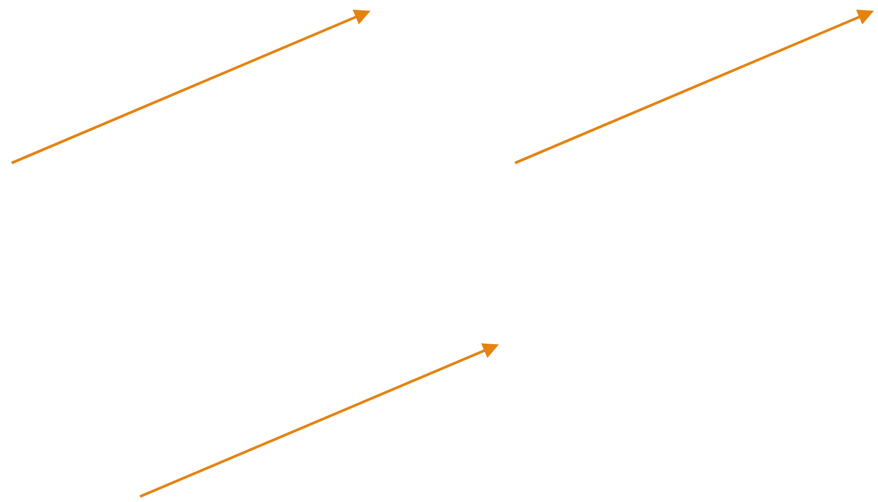
3D Transformation (rotation)

- 2D, 3D rotations are *orthogonal* matrices.
  - Rows of the matrix are mutually orthogonal unit vectors

- Need some background on vectors…

Background: Vector

- A vector describes a length and a direction
  - Commonly drawn by an arrow

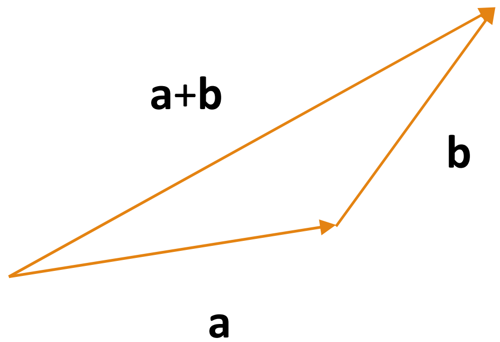


Background: Vector

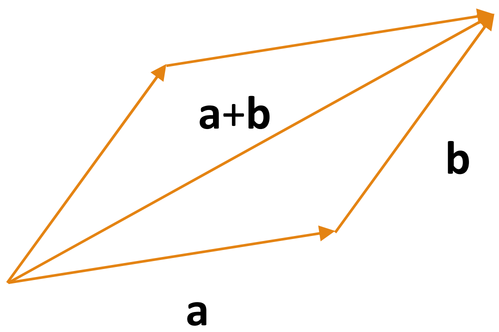
- A vector describes a length and a direction
- Notations: **a (bold character)**
  - Other ways? e.g., $\vec{a}$
- Length of a vector
  - ‖ **a** ‖
- Unit vector
  - A vector **a** if ‖ **a** ‖ $= 1$
- Zero vector
  - A vector **a** if ‖ **a** ‖ $= 0$
- Two vectors are equal if and only if they have the same length and direction.

Background: Vector

- Vector addition: *Parallelogram rule*

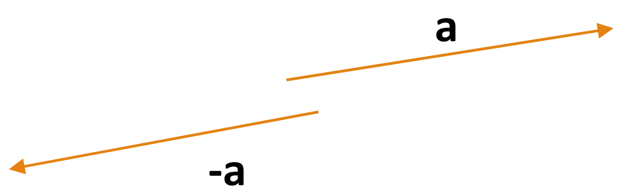


- Properties
  - $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b} + \boldsymbol{a}$ (commutative)

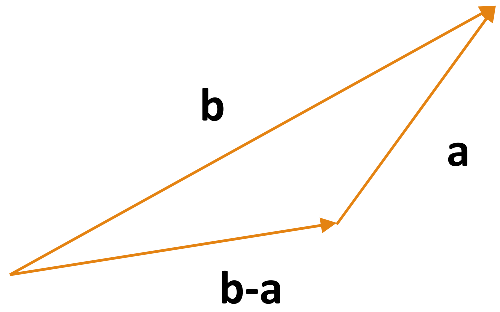


Background: Vector

- Unary minus of a vector **a**
  - $-\boldsymbol{a}$



- Subtraction
  - $\boldsymbol{b}-\boldsymbol{a}\equiv-\boldsymbol{a}+\boldsymbol{b}$

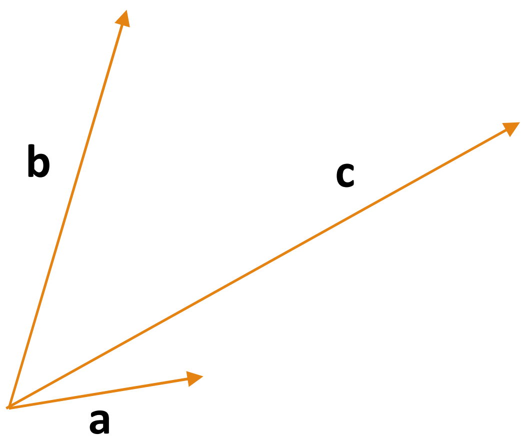


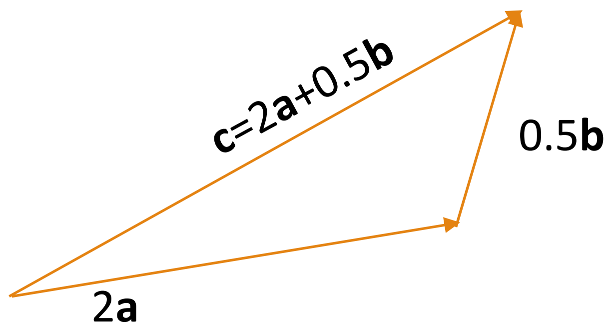
- *Scale*
  - $k\,\boldsymbol{a}$

Background: Basis Vector

- A 2D vector can be written as a combination of any two nonzero vectors which are not parallel (*linear independence)*
  - *Two linearly independent vectors (basis vectors) form a 2D basis*

- $e.g., \boldsymbol{c} = a_c\boldsymbol{a} + b_c\boldsymbol{b}$
  - Weights are unique

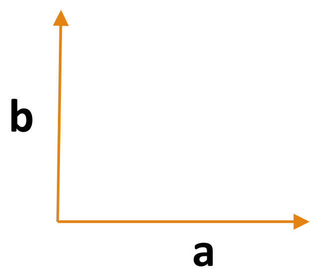




Background: Basis Vector

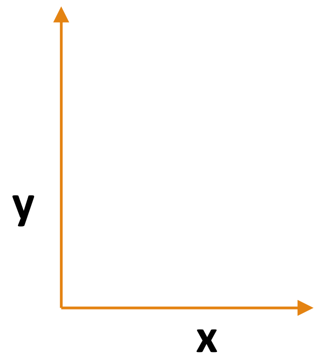
- A 2D vector can be written as a combination of any two nonzero vectors which are not parallel (*linear independence)*
  - *Two linearly independent vectors (basis vectors) form a 2D basis*

- Orthogonality
  - Two vectors are orthogonal if they are at right angles to each other

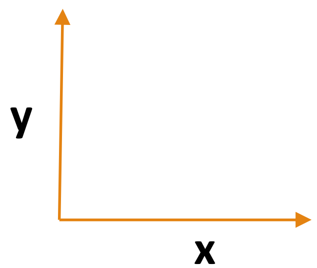


Background: Basis Vector

- A 2D vector can be written as a combination of any two nonzero vectors which are not parallel (*linear independence)*
  - *Two linearly independent vectors (basis vectors) form a 2D basis*



- *Special cases*
  - Two vectors are *orthogonal* if they are at right angles to each other
  - Two vectors are *orthonormal* if they are orthogonal and unit vectors



$$\| \mathbf{x} \| = \| \mathbf{y} \| = 1$$

  - Note: the special vectors can be used to represent all other vectors in *a Cartesian coordinate system*
    - *A coordinate system that specifies each point uniquely*

Background: Cartesian Coordinate System

- *Special cases*
  - Two vectors are *orthogonal* if they are at right angles to each other
  - Two vectors are *orthonormal* if they are orthogonal and unit vectors

  - Note: the special *orthonormal* vector can be used to represent all other vectors in *a Cartesian coordinate system*
    - *A coordinate system that specifies each point uniquely in a plane or 3D space by a pair of numerical components*
    - $e.g., \ \boldsymbol{a} = x_a\boldsymbol{x} + y_a\boldsymbol{y}$
    - x_a *and* y_a *are Cartesian coordinates of the 2D vector* $\boldsymbol{a}$

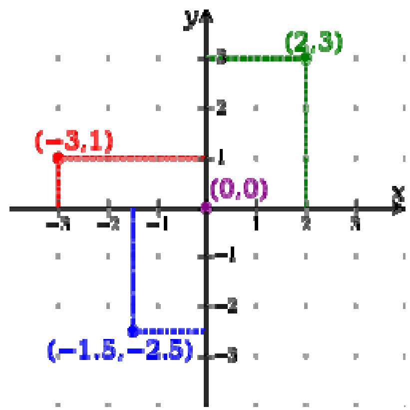


Image from wikipedia

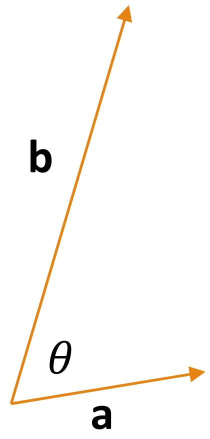
Background: Cartesian Coordinate System

- *Properties of a Cartesian coordinate system*
  - $\| \boldsymbol{a} \| = \sqrt{x_a^2 + y_a^2}$
  - $\boldsymbol{a} = \begin{bmatrix} x_a \\ y_a \end{bmatrix}$
  - $\boldsymbol{a}^T = [x_a \;\; y_a]$

- *3D case*
  - $\boldsymbol{a} = x_a\boldsymbol{x} + y_a\boldsymbol{y} + z_a\boldsymbol{z}$
  - ***x, y, z*** *are orthonormal*

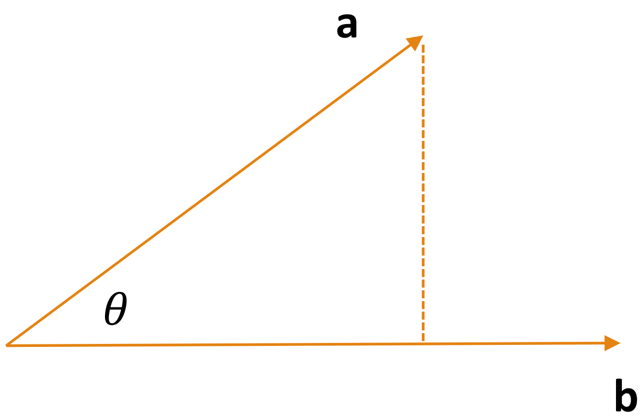
Background: Dot Product

- Vector multiplications
  - Dot product (scalar product)
    - $\boldsymbol{a} \cdot \boldsymbol{b} = \| \boldsymbol{a} \| \| \boldsymbol{b} \| \cos\theta$
    - Properties
      - $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \mathbf{a}$ (commutative)
      - $\boldsymbol{a} \cdot (\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \cdot \boldsymbol{c}$ (distributive)
      - $(k\boldsymbol{a}) \cdot \boldsymbol{b} = \boldsymbol{a} \cdot (k\boldsymbol{b}) = k\boldsymbol{a} \cdot \boldsymbol{b}$ (scalar multiplication)
    - *Orthogonal*
      - $\boldsymbol{a} \cdot \boldsymbol{b} = \| \boldsymbol{a} \| \| \boldsymbol{b} \| \cos\theta = 0$
      - Two non-zero vectors **a** and **b** are *orthogonal* if and only if $\boldsymbol{a} \cdot \boldsymbol{b} = 0$



Background: Dot Product

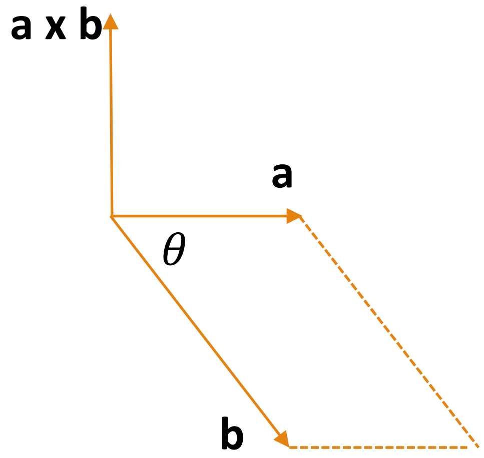
- Vector multiplications
  - Dot product (scalar product)
    - $\boldsymbol{a} \cdot \boldsymbol{b} = \| \boldsymbol{a} \| \| \boldsymbol{b} \| \cos\theta$
    - Usage: $(\boldsymbol{a} \rightarrow \boldsymbol{b})$ projection of a vector to another one
    - $a \rightarrow b = \| \boldsymbol{a} \| \cos\theta = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{b}\|}$
    - Note: this is the length of the projected vector onto **b**



  - Dot product in Cartesian coordinates
    - Properties: $\boldsymbol{x} \cdot \boldsymbol{x} = \boldsymbol{y} \cdot \boldsymbol{y} = 1$ and $\boldsymbol{x} \cdot \boldsymbol{y} = 0$
    - $\boldsymbol{a} \cdot \boldsymbol{b} = (x_a \boldsymbol{x} + y_a \boldsymbol{y}) \cdot (x_b \boldsymbol{x} + y_b \boldsymbol{y})$
    - $= x_a x_b (\boldsymbol{x} \cdot \boldsymbol{x}) + x_a y_b (\boldsymbol{x} \cdot \boldsymbol{y}) + x_b y_a (\boldsymbol{y} \cdot \boldsymbol{x}) + y_a y_b (\boldsymbol{y} \cdot \boldsymbol{y})$
    - $= x_a x_b + y_a y_b$
    - In 3D,
      - $\boldsymbol{a} \cdot \boldsymbol{b} = x_a x_b + y_a y_b + z_a z_b$

Background: Cross Product

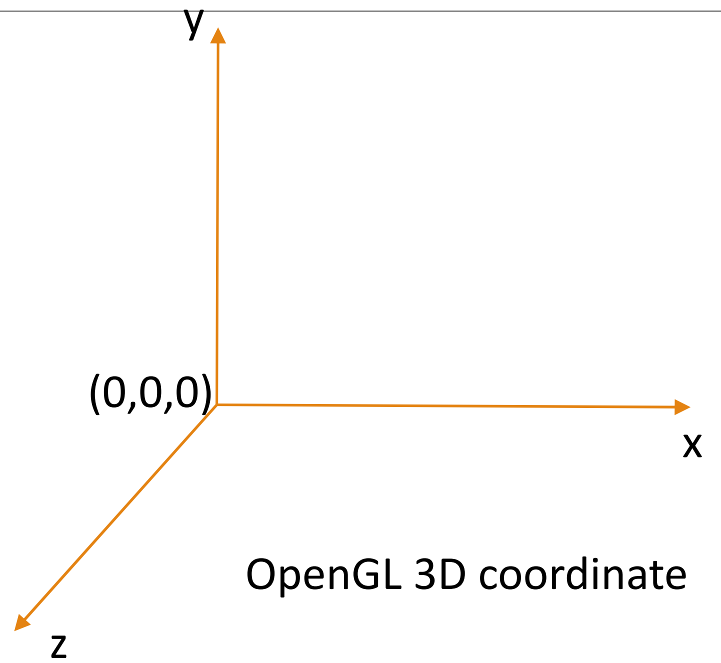
- Vector multiplications
  - Cross products (used only for 3D vectors)
    - $\boldsymbol{a} \times \boldsymbol{b} = \| a \| \| b \| \sin\theta$ **n**
    - n: unit vector that is perpendicular to $\boldsymbol{a}$ and $\boldsymbol{b}$
    - Return a 3D vector that is perpendicular to the two arguments
      - Two possible directions of the resulting vector



  - Length of the resulting vector
    - $\| \boldsymbol{a} \times \boldsymbol{b} \| = \| \boldsymbol{a} \| \| \boldsymbol{b} \| \sin\theta$
    - Equal to the area of the parallelogram formed by the two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
  - Three Cartesian unit vectors
    - $\boldsymbol{x} = (1,0,0)$
    - $\boldsymbol{y} = (0,1,0)$
    - $\boldsymbol{z} = (0,0,1)$

Background: Cross Product

- Cross products of the unit vectors
  - $\boldsymbol{x} \times \boldsymbol{y} = \boldsymbol{z}$
  - $\boldsymbol{y} \times \boldsymbol{x} = -\boldsymbol{z}$
  - $\boldsymbol{y} \times \boldsymbol{z} = \boldsymbol{x}$
  - $\boldsymbol{z} \times \boldsymbol{y} = -\boldsymbol{x}$
  - $\boldsymbol{z} \times \boldsymbol{x} = \boldsymbol{y}$
  - $\boldsymbol{x} \times \boldsymbol{z} = -\boldsymbol{y}$



OpenGL 3D coordinate

- Note:
  - We set a convention that $x \times y$ should be in the plus or minus z direction
  - $\boldsymbol{x} \times \boldsymbol{y} \neq \boldsymbol{y} \times \boldsymbol{x}$

Background: Cross Product

- Properties
  - $\boldsymbol{a} \times (\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a} \times \boldsymbol{b} + \boldsymbol{a} \times \boldsymbol{c}$
  - $\boldsymbol{a} \times (k\boldsymbol{b}) = k(\boldsymbol{a} \times \boldsymbol{b})$
  - $\boldsymbol{a} \times \boldsymbol{b} = -(\boldsymbol{b} \times \boldsymbol{a})$

- Cross product in Cartesian coordinates (based on $\boldsymbol{x} \times \boldsymbol{x} = \boldsymbol{0}$)
  - $\boldsymbol{a} \times \boldsymbol{b} = (x_a\boldsymbol{x} + y_a\boldsymbol{y} + z_a\boldsymbol{z}) \times (x_b\boldsymbol{x} + y_b\boldsymbol{y} + z_b\boldsymbol{z})$
  - $= x_a x_b \boldsymbol{x} \times \boldsymbol{x} + x_a y_b \boldsymbol{x} \times \boldsymbol{y} + x_a z_b \boldsymbol{x} \times \boldsymbol{z} + y_a x_b \boldsymbol{y} \times \boldsymbol{x} + y_a y_b \boldsymbol{y} \times \boldsymbol{y} + y_a z_b \boldsymbol{y} \times \boldsymbol{z} + z_a x_b \boldsymbol{z} \times \boldsymbol{x} + z_a y_b \boldsymbol{z} \times \boldsymbol{y} + z_a z_b \boldsymbol{z} \times \boldsymbol{z}$
  - $= (y_a z_b - z_a y_b)\boldsymbol{x} + (z_a x_b - x_a z_b)\boldsymbol{y} + (x_a y_b - y_a x_b)\boldsymbol{z}$

  - In coordinate form,
    - $\boldsymbol{a} \times \boldsymbol{b} = (y_a z_b - z_a y_b, z_a x_b - x_a z_b, x_a y_b - y_a x_b)$

3D Transformation (rotation) – cont'd

- 2D, 3D rotations are *orthogonal* matrices.
  - Rows of the matrix are mutually orthogonal unit vectors (*orthonormal*)

3D Transformation (rotation) – cont'd

- Inverse of transformation matrices
  - Scale matrix is a diagonal matrix
    - $M = \begin{bmatrix} m_{11} & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 \\ 0 & 0 & m_{33} & 0 \\ 0 & 0 & 0 & m_{44} \end{bmatrix}, M^{-1} = \begin{bmatrix} 1/m_{11} & 0 & 0 & 0 \\ 0 & 1/m_{22} & 0 & 0 \\ 0 & 0 & 1/m_{33} & 0 \\ 0 & 0 & 0 & 1/m_{44} \end{bmatrix}$
  - Rotation matrices are *orthonormal* matrices
    - A square matrix whose rows are orthogonal unit vectors
    - $R^T R = R R^T = I$
    - $R^{-1} = R^T$