

CT4201/EC4215: 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

Homogeneous 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

- $$M = \begin{bmatrix} m_{11} & m_{12} & x_\delta \\ m_{21} & m_{22} & y_\delta \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Coordinates

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

- e.g., 2D translation

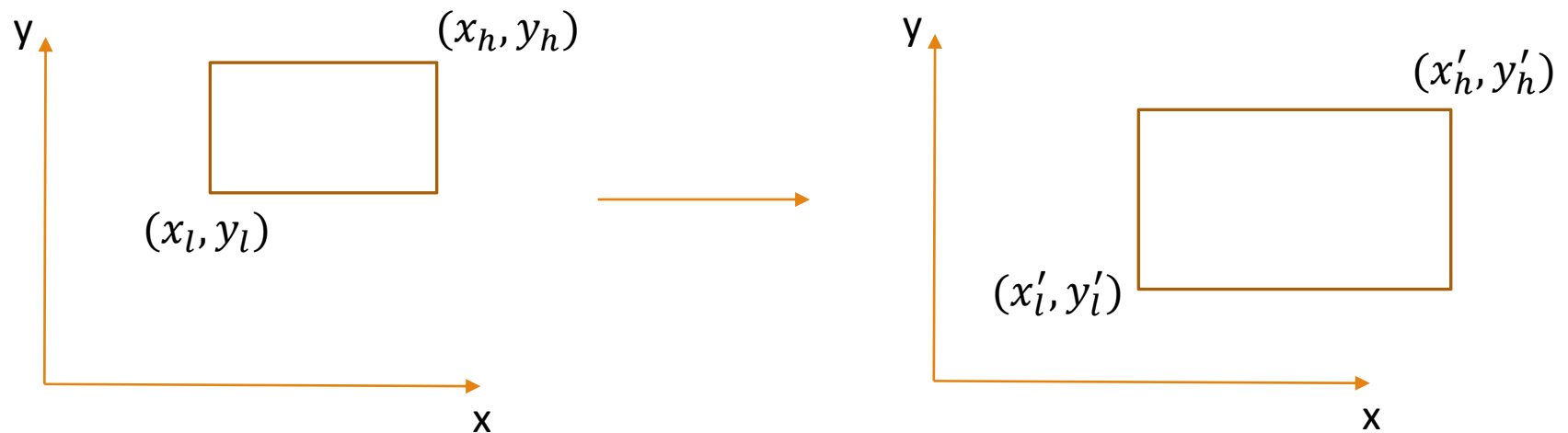
- $$\begin{bmatrix} 1 & 0 & x_\delta \\ 0 & 1 & y_\delta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + x_\delta \\ y + y_\delta \\ 1 \end{bmatrix}$$

- e.g., rotation after 2D translation

- $$M = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & x_\delta \\ 0 & 1 & y_\delta \\ 0 & 0 & 1 \end{bmatrix}$$

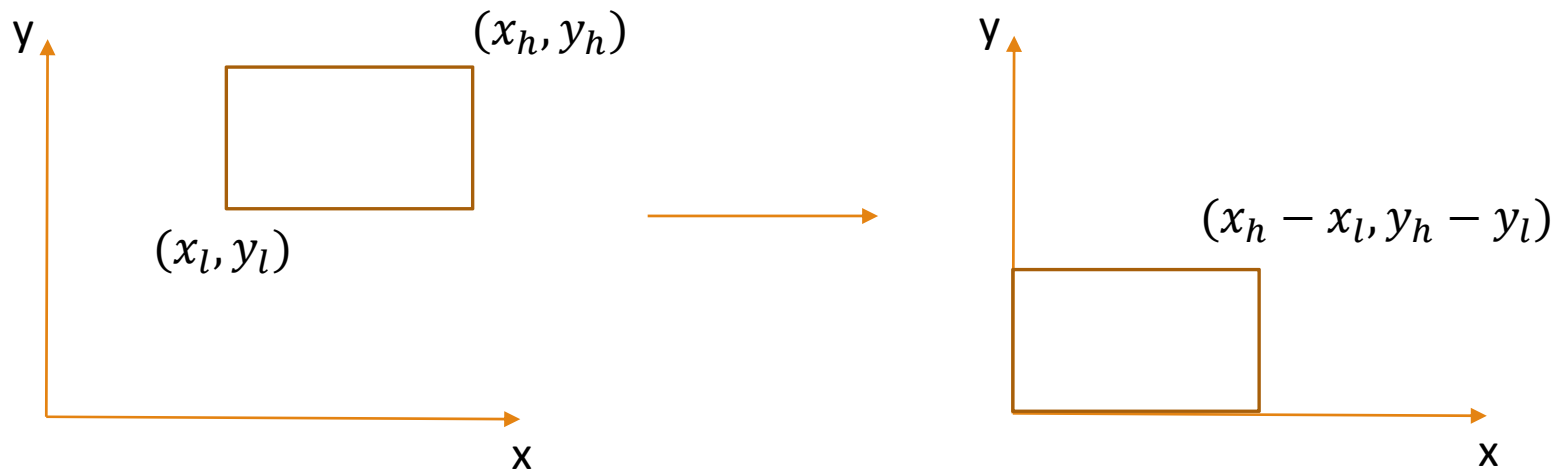
Examples

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



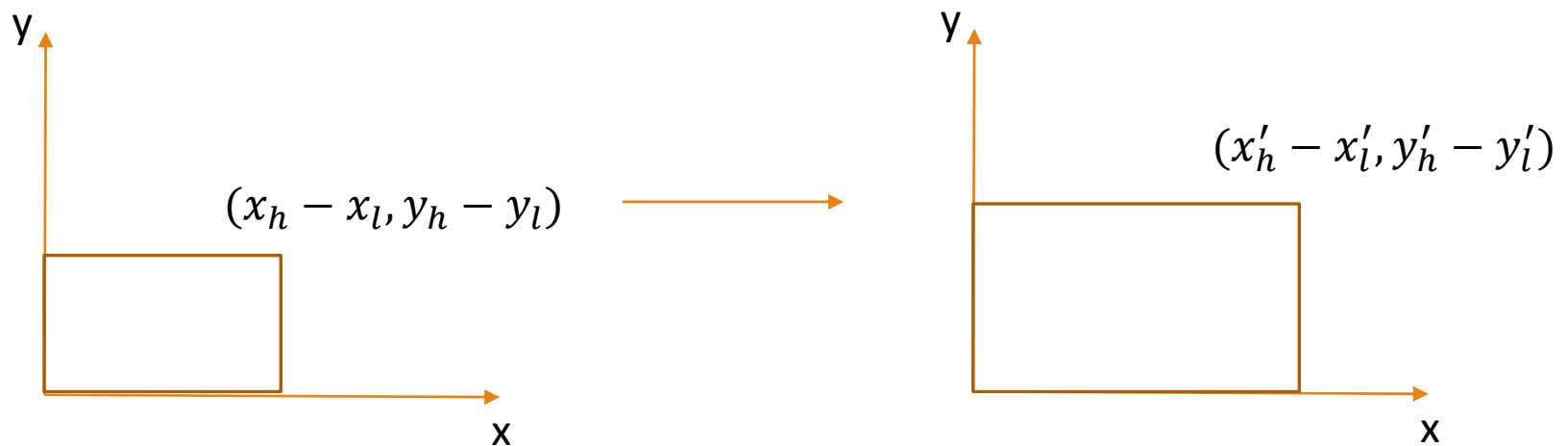
Examples

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



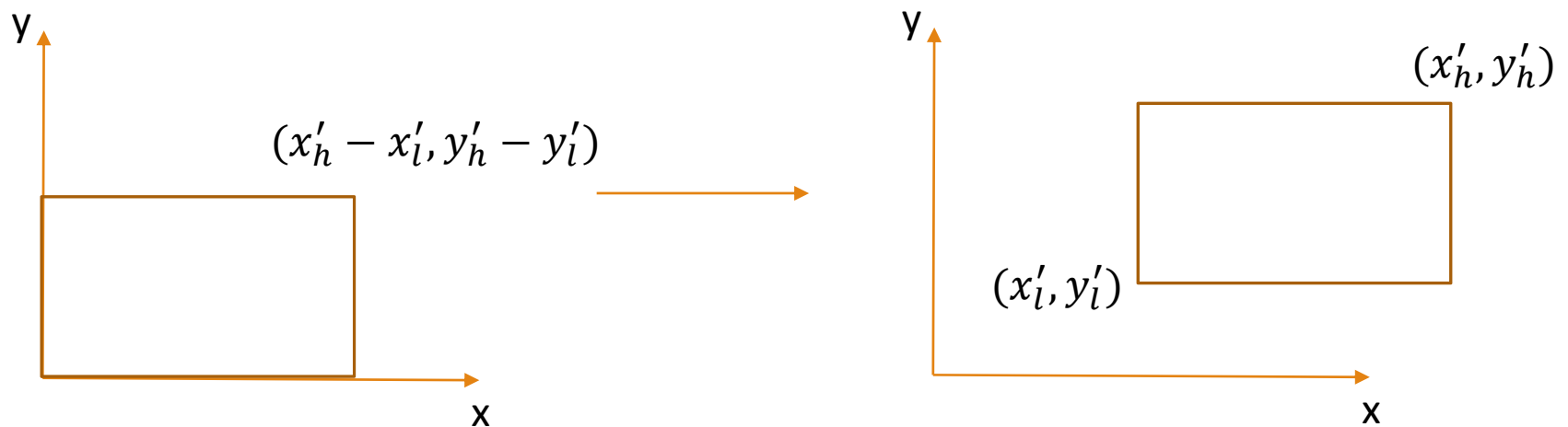
Examples

- 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'_i, y'_i)



Examples

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

$$\circ = \begin{bmatrix} 1 & 0 & x'_l \\ 0 & 1 & y'_l \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x'_h - x'_l}{x_h - x_l} & 0 & 0 \\ 0 & \frac{y'_h - y'_l}{y_h - y_l} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_l \\ 0 & 1 & -y_l \\ 0 & 0 & 1 \end{bmatrix}$$

$$\circ = \begin{bmatrix} \frac{x'_h - x'_l}{x_h - x_l} & 0 & \frac{x'_l x_h - x'_h x_l}{x_h - x_l} \\ 0 & \frac{y'_h - y'_l}{y_h - y_l} & \frac{y'_l y_h - y'_h y_l}{y_h - y_l} \\ 0 & 0 & 1 \end{bmatrix}$$

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

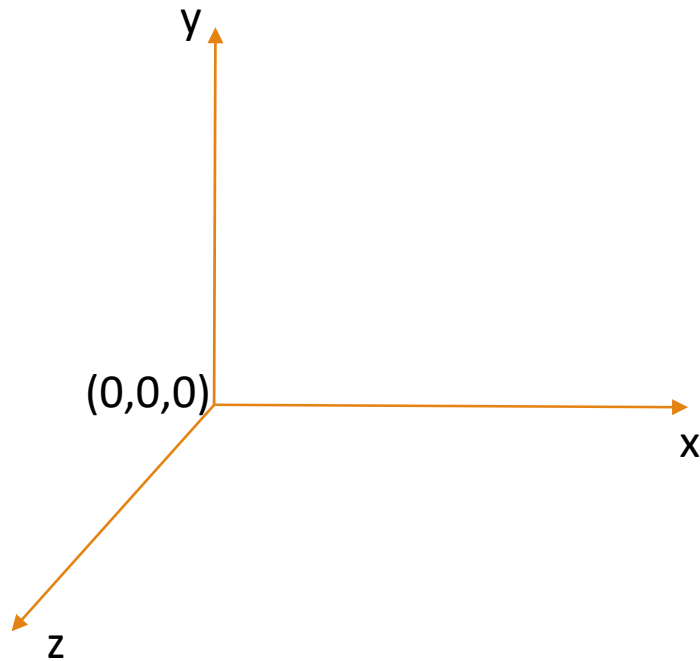
- Homogenous coordinates are common for graphics applications. Why?
- A naïve way of implementing translations is to move the positions directly without forming a matrix

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

- $rotate - z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

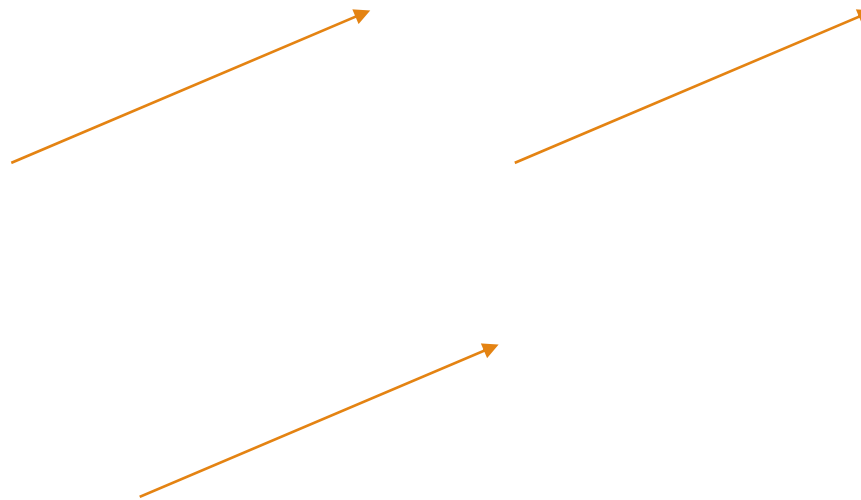
3D Transformation (rotation)

- 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

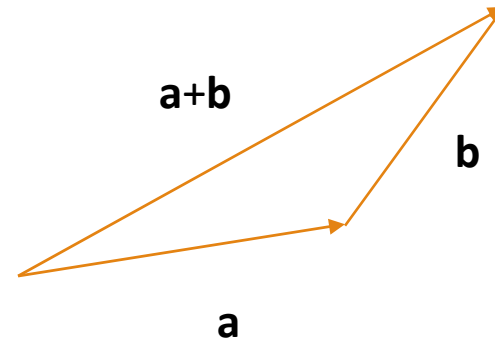


Background: Vector

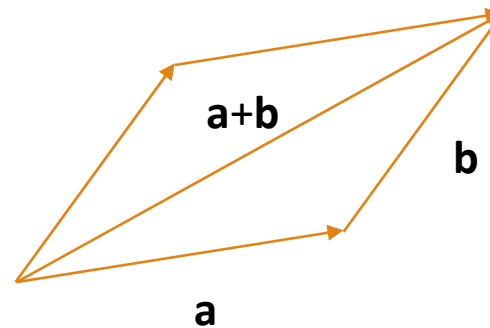
- A vector describes a length and a direction
- Notations: **a** (bold character)
 - Other ways? e.g., \vec{a}
- Length of a vector
 - $\| \mathbf{a} \|$
- Unit vector
 - A vector **a** if $\| \mathbf{a} \| = 1$
- Zero vector
 - A vector **a** if $\| \mathbf{a} \| = 0$
- Two vectors are equal if and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

Background: Vector

- Vector addition: *Parallelogram rule*

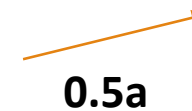
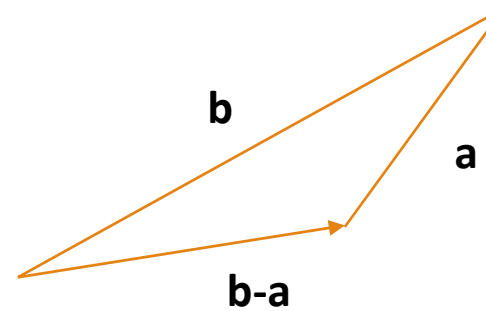
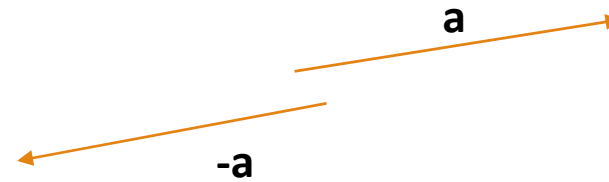


- Properties
 - $a + b = b + a$ (commutative)



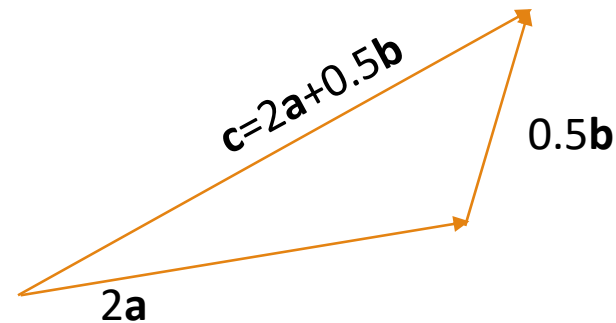
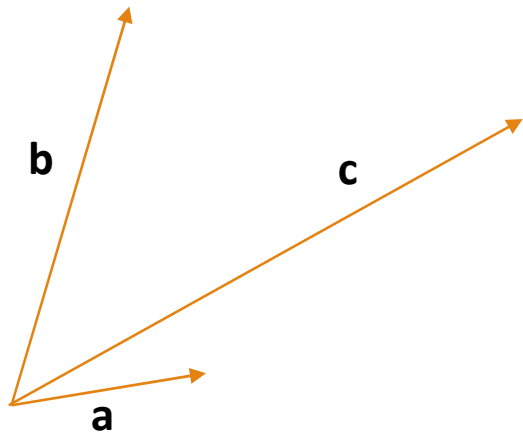
Background: Vector

- Unary minus of a vector a
 - $-a$
- Subtraction
 - $b - a \equiv -a + b$
- Scale
 - ka



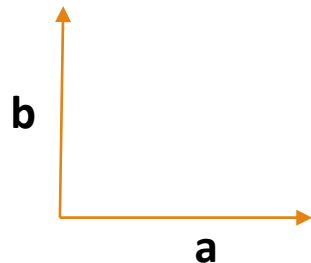
Background: Basis Vector

- 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., $\mathbf{c} = a_c\mathbf{a} + b_c\mathbf{b}$*
 - Weights are unique



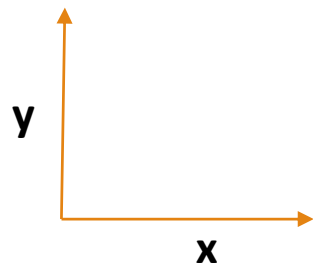
Background: Basis Vector

- 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

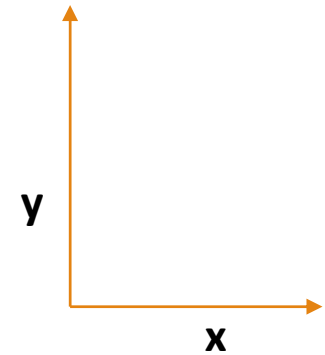


Background: Basis Vector

- 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., $\mathbf{a} = x_a\mathbf{x} + y_a\mathbf{y}$
 - x_a and y_a are Cartesian coordinates of the 2D vector \mathbf{a}

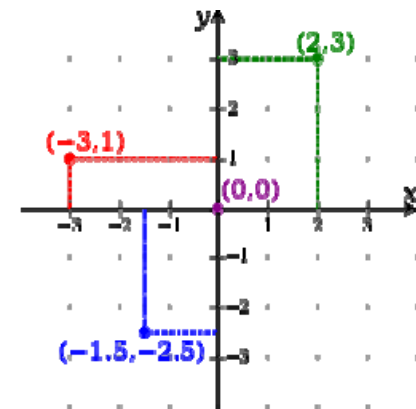


Image from wikipedia

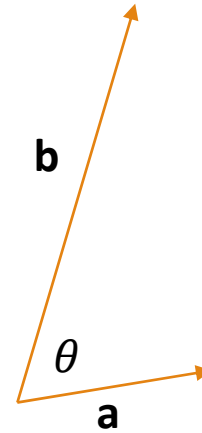
Background: Cartesian Coordinate System

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

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

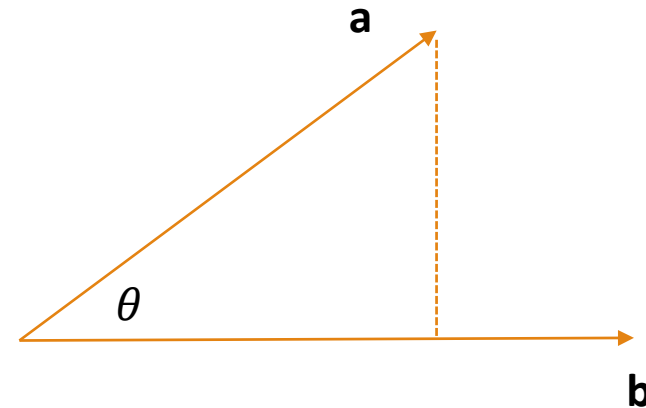
Background: Dot Product

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



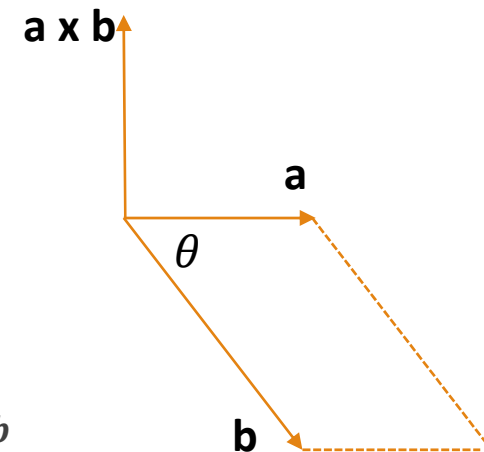
Background: Dot Product

- Vector multiplications
 - Dot product (scalar product)
 - $\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos\theta$
 - Usage: ($\mathbf{a} \rightarrow \mathbf{b}$) projection of a vector to another one
 - $\mathbf{a} \rightarrow \mathbf{b} = \|\mathbf{a}\| \cos\theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|}$
 - Note: this is the length of the projected vector onto \mathbf{b}
 - Dot product in Cartesian coordinates
 - Properties: $\mathbf{x} \cdot \mathbf{x} = \mathbf{y} \cdot \mathbf{y} = 1$ and $\mathbf{x} \cdot \mathbf{y} = 0$
 - $\mathbf{a} \cdot \mathbf{b} = (x_a \mathbf{x} + y_a \mathbf{y}) \cdot (x_b \mathbf{x} + y_b \mathbf{y})$
 - $= x_a x_b (\mathbf{x} \cdot \mathbf{x}) + x_a y_b (\mathbf{x} \cdot \mathbf{y}) + x_b y_a (\mathbf{y} \cdot \mathbf{x}) + y_a y_b (\mathbf{y} \cdot \mathbf{y})$
 - $= x_a x_b + y_a y_b$
 - In 3D,
 - $\mathbf{a} \cdot \mathbf{b} = x_a x_b + y_a y_b + z_a z_b$



Background: Cross Product

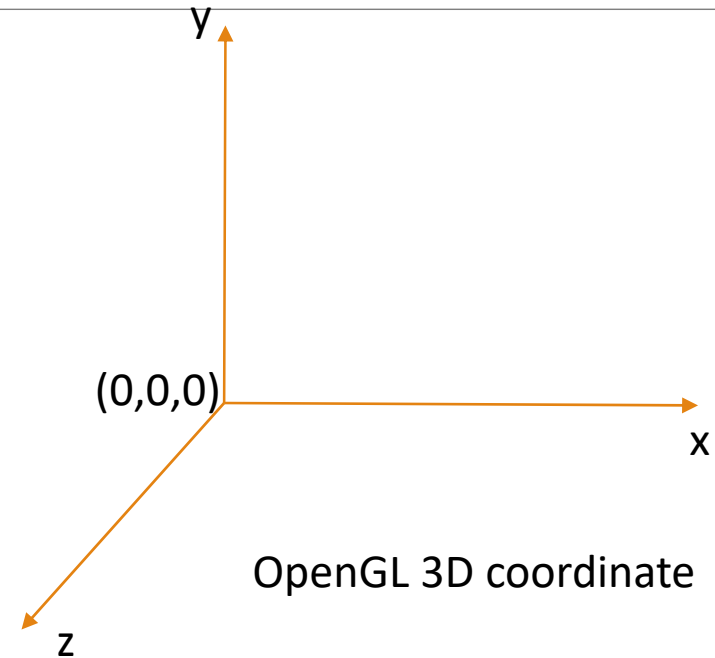
- Vector multiplications
 - Cross products (used only for 3D vectors)
 - $\mathbf{a} \times \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \sin\theta \mathbf{n}$
 - \mathbf{n} : unit vector that is perpendicular to \mathbf{a} and \mathbf{b}
 - Return a 3D vector that is perpendicular to the two arguments
 - Two possible directions of the resulting vector
 - Length of the resulting vector
 - $\|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\| \|\mathbf{b}\| \sin\theta$
 - Equal to the area of the parallelogram formed by the two vectors \mathbf{a} and \mathbf{b}
 - Three Cartesian unit vectors
 - $\mathbf{x} = (1,0,0)$
 - $\mathbf{y} = (0,1,0)$
 - $\mathbf{z} = (0,0,1)$



Background: Cross Product

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

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



Background: Cross Product

- Properties
 - $\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$
 - $\mathbf{a} \times (k\mathbf{b}) = k(\mathbf{a} \times \mathbf{b})$
 - $\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})$
- Cross product in Cartesian coordinates (based on $\mathbf{x} \times \mathbf{x} = \mathbf{0}$)
 - $\mathbf{a} \times \mathbf{b} = (x_a\mathbf{x} + y_a\mathbf{y} + z_a\mathbf{z}) \times (x_b\mathbf{x} + y_b\mathbf{y} + z_b\mathbf{z})$
 - $= x_ax_b\mathbf{x} \times \mathbf{x} + x_ay_b\mathbf{x} \times \mathbf{y} + x_az_b\mathbf{x} \times \mathbf{z} + y_ax_b\mathbf{y} \times \mathbf{x} + y_ay_b\mathbf{y} \times \mathbf{y} + y_az_b\mathbf{y} \times \mathbf{z} + z_ax_b\mathbf{z} \times \mathbf{x} + z_ay_b\mathbf{z} \times \mathbf{y} + z_az_b\mathbf{z} \times \mathbf{z}$
 - $= (y_az_b - z_ay_b)\mathbf{x} + (z_ax_b - x_az_b)\mathbf{y} + (x_ay_b - y_ax_b)\mathbf{z}$
- In coordinate form,
 - $\mathbf{a} \times \mathbf{b} = (y_az_b - z_ay_b, z_ax_b - x_az_b, x_ay_b - y_ax_b)$

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

- $R^T R = R R^T = I$

- $R^{-1} = R^T$