5CM507 Graphics

Lecture A02a Mathematics

Dr Youbing Zhao

September 25, 2025

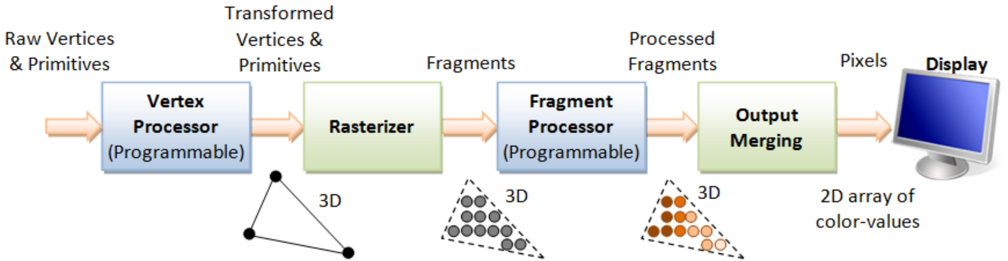
Last Week

An Introduction to

- Computer Graphics (3D -> 2D)
- Graphics Pipeline (vertex shader and fragment shader)
- Module Logistics

Note: this module is not teaching how to use a 3d modelling software, but teach you the basic rendering, lighting, animation, and physics mechanisms of game engines. It requires progrqamming.

The Classical Rasterisation Pipeline



3D Graphics Rendering Pipeline: Output of one stage is fed as input of the next stage. A vertex has attributes such as (x, y, z) position, color (RGB or RGBA), vertex-normal (n_x, n_y, n_z), and texture. A primitive is made up of one or more vertices. The rasterizer raster-scans each primitive to produce a set of grid-aligned fragments, by interpolating the vertices.

[Prof. Chua Hock Chuan, NTU]

Overview

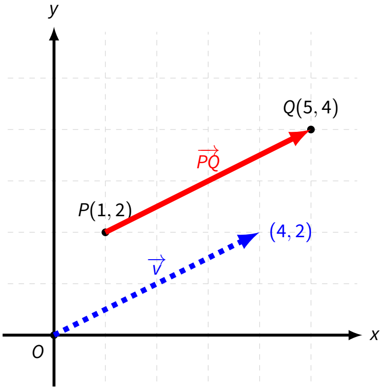
- **Vectors** and dot product
- Matrix and transforms

Vectors

Vectors

Representation

- Properties
  - Direction
  - Length(Magnitude)
  - No absolute starting point
- Examples include
  - Force
  - Velocity
  - Represented by the start and end points $\overrightarrow{PQ}$
  - Notation $\overrightarrow{v}$ or in bold $\boldsymbol{v}$: e.g. $\boldsymbol{v} = Q - P = [4, 2]$

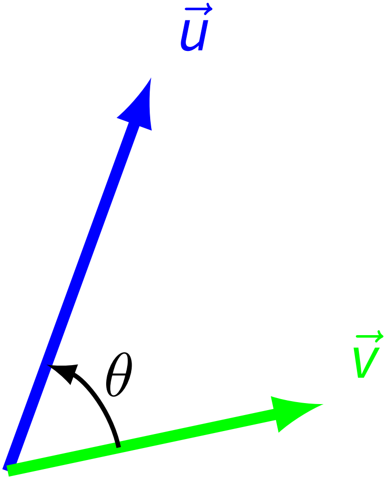


Vector Multiplication

- Scalar product (easy)
  - Every vector can be multiplied by a number
- Dot product (most important)
- Cross product (needed)

Dot Product

- 2D: $\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y$
- 3D: $\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y + u_z v_z$
- General: $\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i$
- Vector length: $\vec{u} \cdot \vec{u} = \|\vec{u}\|^2$
- A dot product returns a scalar



Dot Product

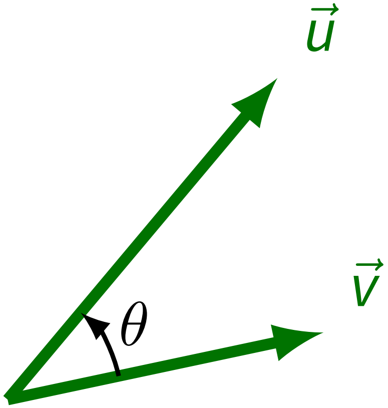
Angle between two vectors

If two vectors have an angle of θ

- $\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos(\theta)$

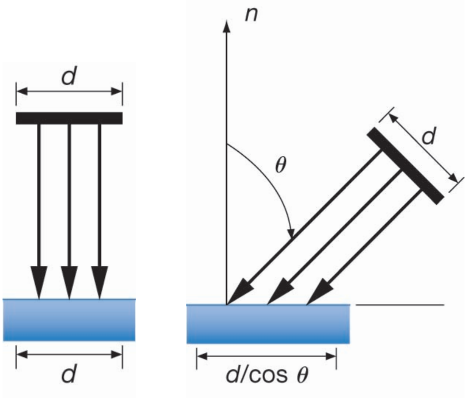
Calculate the angle between two vectors

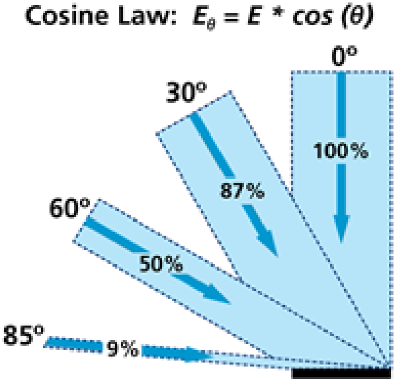
- $$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$
- 1 if two vectors are in the same direction ($\theta = 0$)
- 0 if two vectors are perpendicular ($\theta = 90$)



Dot Product in Lighting Models

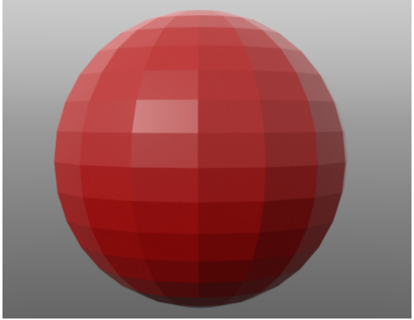
$$E_\theta = E_0 \cos\theta = E_0(\vec{\hat{l}} \cdot \vec{\hat{n}})$$



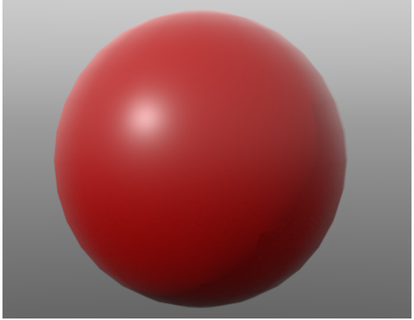


Light intensities Imply the Object Shape

Under a given directed light source, object surfaces with different normals will receive and reflect light differently, allowing us to perceive the shape of the 3D object.



Flat shading



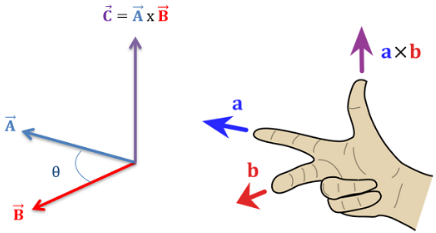
Smooth shading

3D Cross Product

- Given two non-parallel vectors, *A* and *B*
- $\mathrm{A} \times \mathrm{B}$ calculates third vector *C* that is orthogonal to *A* and *B*
- $\mathrm{A} \times \mathrm{B} = (\mathrm{a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x})$

$$A \times B = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$

these details not needed for programming

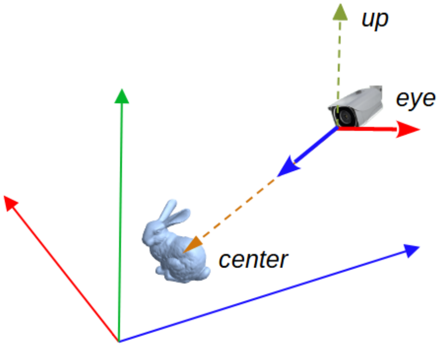


Right hand rule

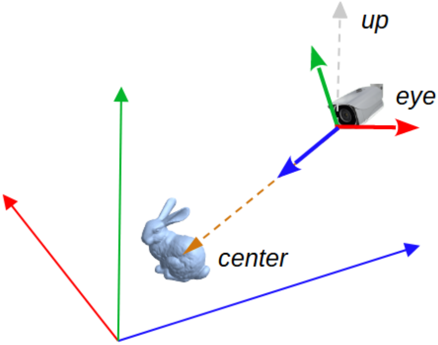
Using Cross Product to Set up the Camera Frame

mat4 glm::lookAt(vec3 eye, vec3 center, vec3 up);

$\vec{Z}$: $Center - Eye$



X (left) Axis of the camera frame: $\vec{\mathbf{x}}$: $\vec{up} \times \vec{Z}$



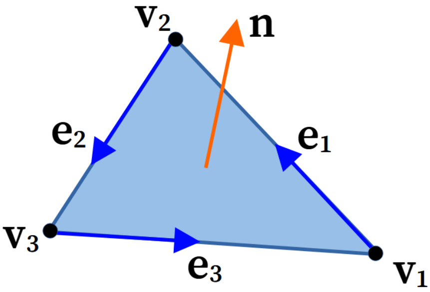
Y (up) Axis of the camera frame: $\vec{\mathbf{y}} = \mathbf{z} \times \mathbf{x}$

Normals of Triangles and Vertices in a Triangle Mesh

for Lighting and Shading

$$\vec{e_1} = \frac{v_2 - v_1}{\|v_2 - v_1\|}, \vec{e_2} = \frac{v_3 - v_2}{\|v_3 - v_2\|}, \vec{e_3} = \frac{v_1 - v_3}{\|v_1 - v_3\|}$$

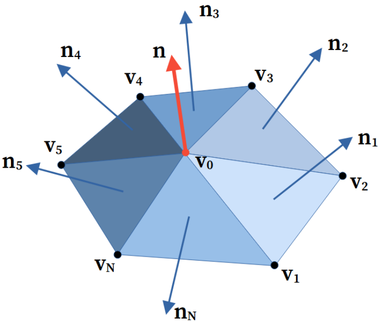
Triangle normal: $\vec{n} = \vec{e_1} \times \vec{e_2}$



Vertex Normal: The angle or area-weighted average of the normals of neighboring triangles.

$$\vec{n} = \frac{\sum_{i=1}^{N} w_i \vec{n_i}}{\left\|\sum_{i=1}^{N} w_i \vec{n_i}\right\|}$$

Notice : $Area_{123} = \frac{1}{2}|\overrightarrow{v_1 v_2} \times \overrightarrow{v_1 v_3}|$



Matrices

Matrix Multiplication

- $[C] = [A][B]$
- Sum over rows & columns
- Recall: matrix multiplication is
  - Associative $ABC = A(BC)$
  - Not commutative $AB \neq BA$

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + a_{14}b_{41}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$$

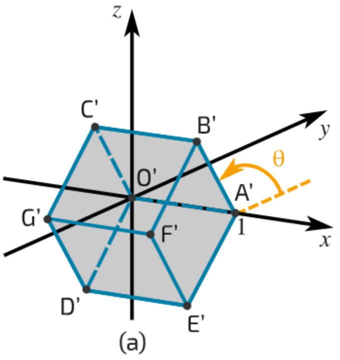
2 x 4   4 x 3   2 x 3

$$c_{22} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} + a_{24}b_{42}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$$

Linear Transformation as Matrices

$$x' = x$$
$$y' = \cos(\theta)y - \sin(\theta)z$$
$$z' = \sin(\theta)y + \cos(\theta)z$$
$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$
$$P' = MP = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



Rotation by an angle θ around the x axis

Questions?