Discussion 02

Triangles & Transforms

Computer Graphics and Imaging
UC Berkeley CS 184

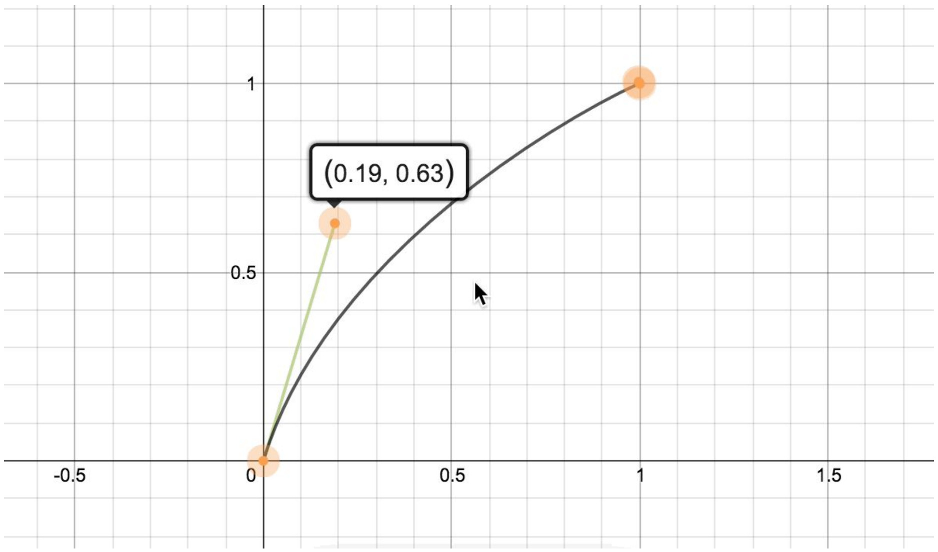
Week 2 Announcements

- Check out the course website for OH times, resources, and more!
- We start tallying Participation Credit this week :) By being here, you can earn +1 point (out of 4).
- The Discussion Attendance form is in the upper navigation of our website.
- **Homework 0** should be done! You still have until tomorrow night if you have any unforeseen circumstances.
- We want to make sure everyone is set-up for the semester :D

Vector Graphics

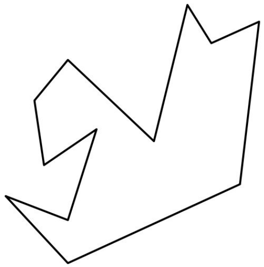
Vector Graphics

- Vector graphics are composed of geometric shapes: points, lines, curves, and polygons!



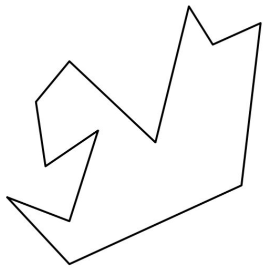
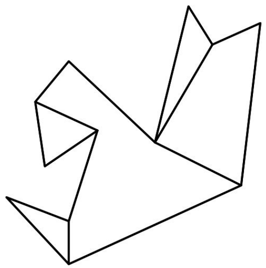
Triangulation

- A polygon is defined in <u>math</u>.
- A computer screen is composed of <u>pixels</u>.
- How do we go from math → pixels?



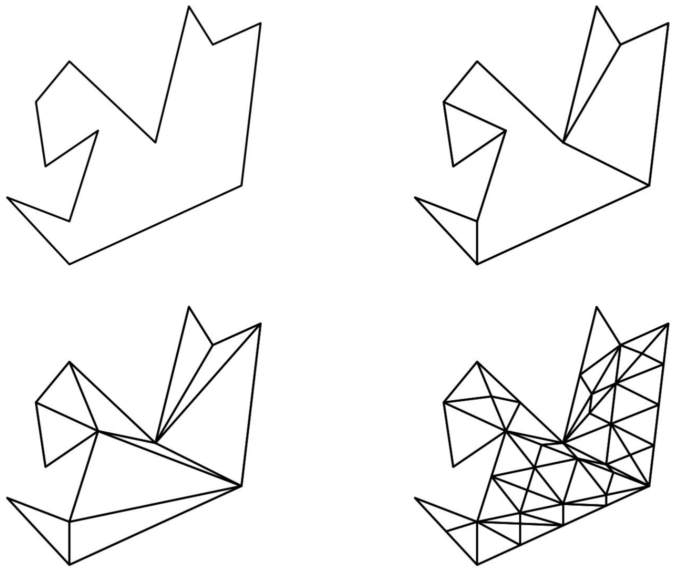
Triangulation

- A polygon is defined in math.
- A computer screen is composed of pixels.
- How do we go from math ➜ pixels?

Triangulation

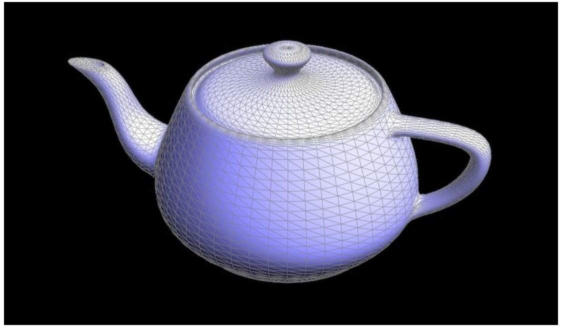
- A polygon is defined in <u>math</u>.
- A computer screen is composed of <u>pixels</u>.
- How do we go from math ➔ pixels?



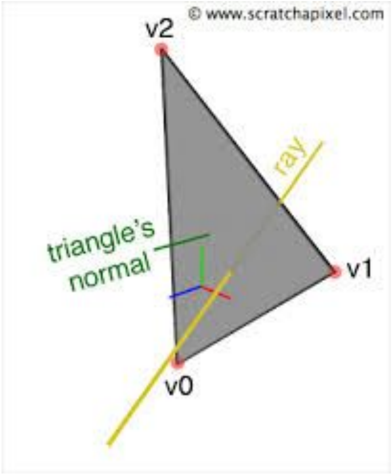
- Using triangulation algorithms, our polygon can be translated from math ➔ triangles!

Triangles are everywhere!

Rasterization
HW 1



Geometric modeling
HW 2, HW 4



Ray tracing
HW 3

Why we use triangles in graphics

- Rasterization
  - Graphics hardware is optimized for triangles.
- Geometric modeling
  - 2D and 3D shapes can be modeled as a collection of triangles.
  - Detecting collisions is easy for triangles.
- Ray tracing
  - Triangles are planar in 3D space.
  - GPUs are optimized to work with triangles.

Vectors & Triangles

Linear Algebra Review

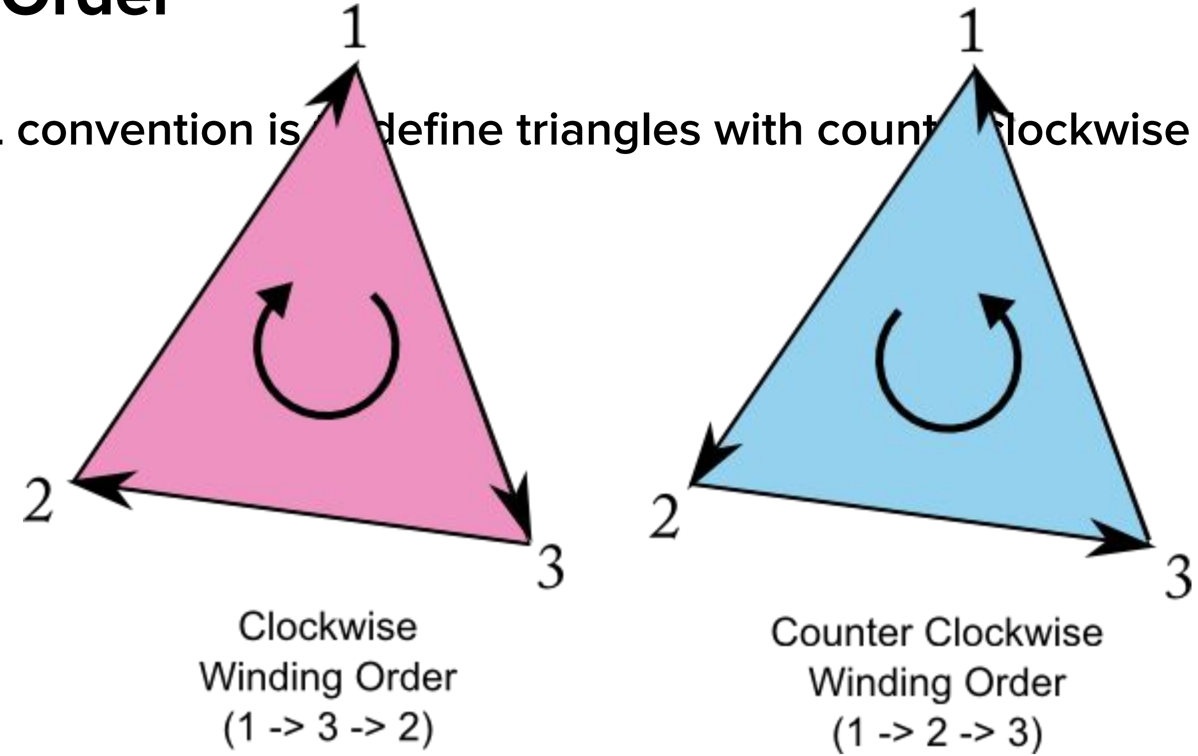
- Dot product
- Matrix-vector multiplication
- Cross product

Winding Order

- OpenGL convention is to define triangles with counterclockwise winding order.

Winding Order

- OpenGL convention is to define triangles with counter-clockwise winding order.



1. Given two vectors, $\mathbf{a} = \begin{pmatrix} 10 \\ 7 \\ 0 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} -2 \\ -11 \\ 0 \end{pmatrix}$, calculate $\mathbf{a} \times \mathbf{b}$. Next, calculate $\mathbf{b} \times \mathbf{a}$.

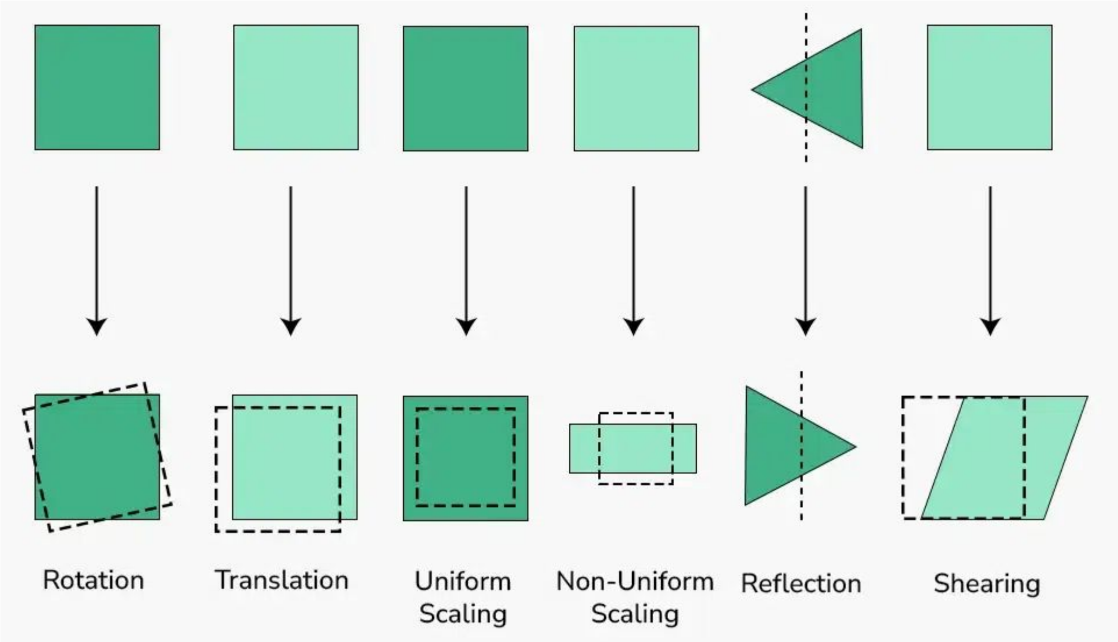
2. Draw the triangle given by points $(2, -1)$, $(12, 6)$, $(10, -5)$. What is the *winding order* of the triangle? In other words, are the points given in clockwise or counter-clockwise order?

3. What are the three vectors defined by the edges of this triangle? Assume the triangle is lying in the xy-plane. Select two of the three vectors. Calculate their cross product.

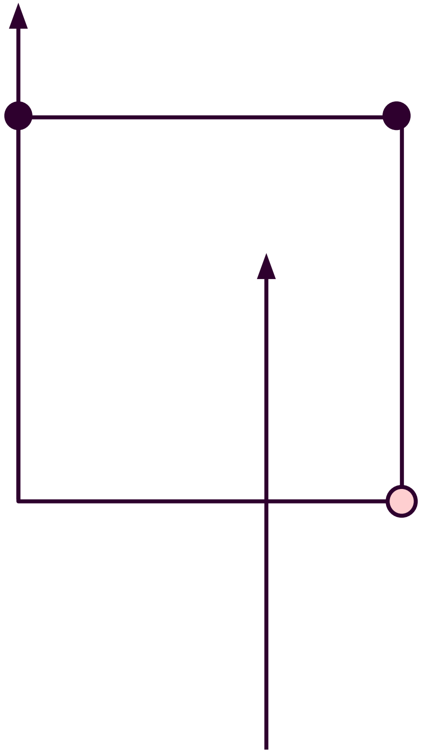
4. In general, given triangle vertices $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, how can winding order be determined?

Transformation Matrices

Transformations

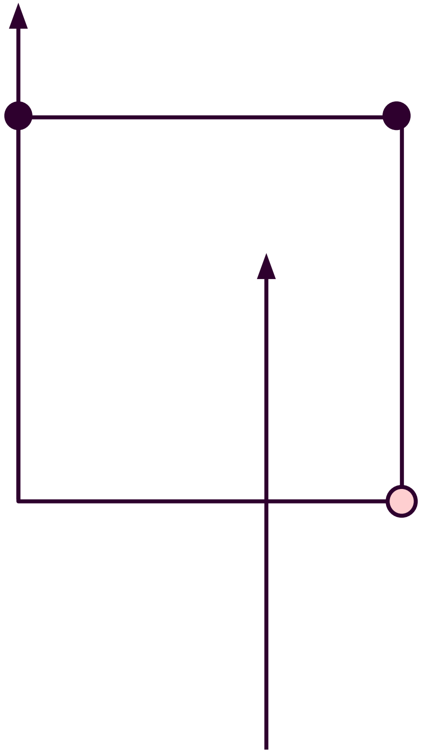


Matrix-Vector Multiplication



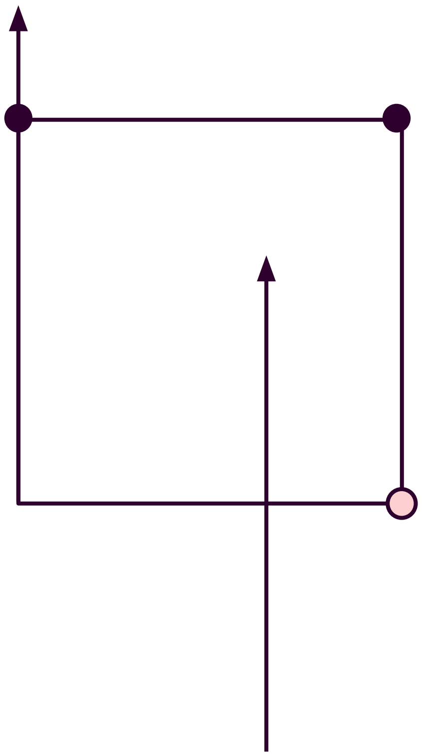
$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \\ \end{bmatrix}$$

Matrix-Vector Multiplication



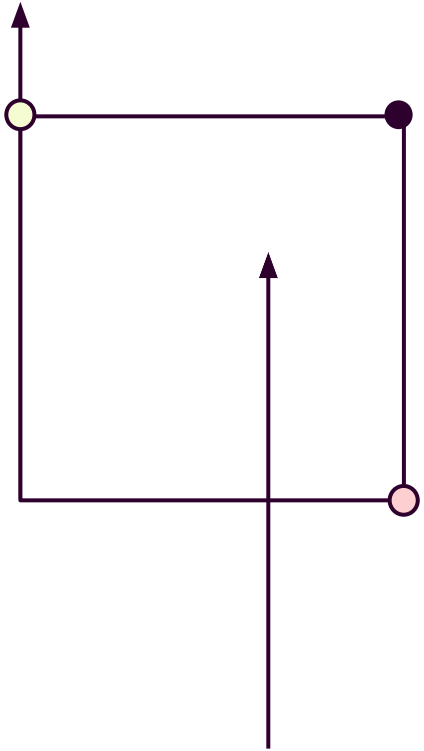
$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \end{bmatrix}$$

Matrix-Vector Multiplication



$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

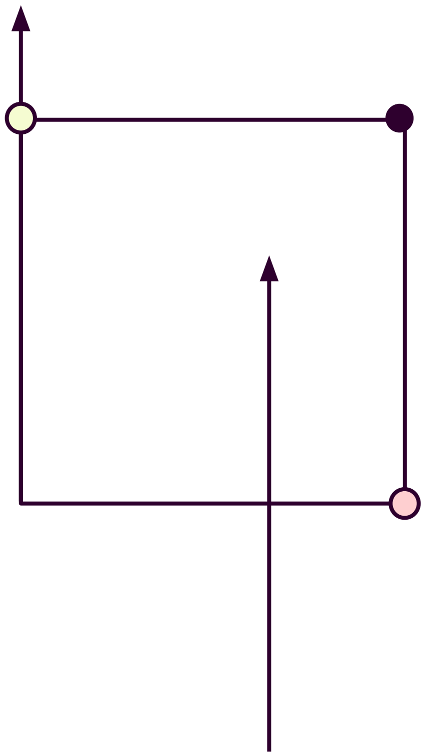
Matrix-Vector Multiplication



$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\theta \\ \end{bmatrix}$$

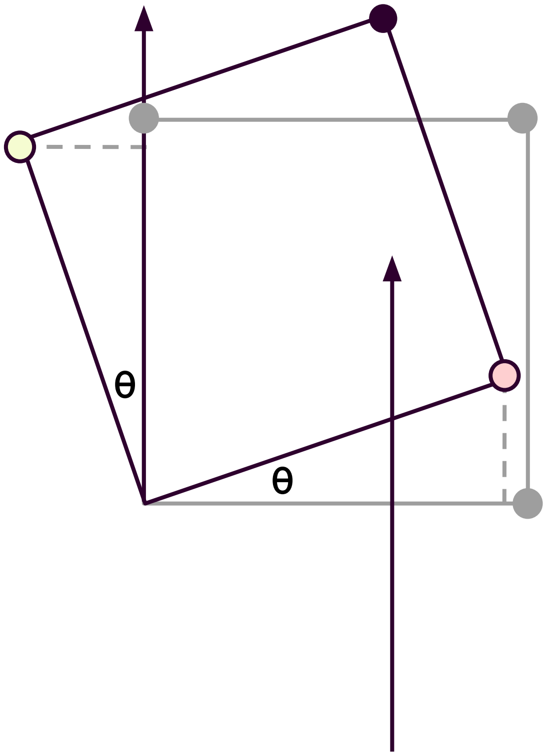
Matrix-Vector Multiplication



$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

Matrix-Vector Multiplication



$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

Homogeneous Coordinates

- Represent a point in n-dimensional space using $n + 1$ coordinates.

$(x, y)^T \rightarrow (x, y, w)^T$

$(x, y, z)^T \rightarrow (x, y, z, w)^T$

- w = 1 to represent a <u>point</u>.
- w = 0 to represent a <u>vector</u>.

Homogeneous Coordinates

- Represent a point in *n*-dimensional space using *n + 1* coordinates.

$(x, y)^T \rightarrow (x, y, w)^T$

$(x, y, z)^T \rightarrow (x, y, z, w)^T$

- w = 1 to represent a <u>point</u>.
- w = 0 to represent a <u>vector</u>.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- Homogeneous coordinates to represent <u>translations</u> as matrices!
- To shift <u>right</u> by t_x, shift <u>up</u> by t_y:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

Transformation Matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s_x} & 0 & 0 \\ 0 & \boldsymbol{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh_x} & 0 \\ \boldsymbol{sh_y} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Shear

Transformation Matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x\mathbf{u} + y\mathbf{v} + \mathbf{o} \\ 1 \end{bmatrix}$$

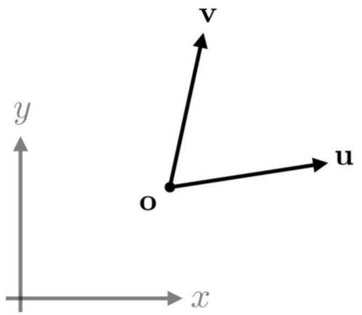
Coordinate System Change

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh}_x & 0 \\ \boldsymbol{sh}_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Shear



1. Isometric transformations preserve the distances between every pair of points on an object. Which transformations are isometric? Which transformations aren't?

1. Isometric transformations preserve the distances between every pair of points on an object. Which transformations are isometric? Which transformations aren't?

Solution: Rotations, translations, and reflections (and their combinations) all fulfill this property. They are what we call isometries or rigid transformations (as if we were manipulating an object that cannot be bent or broken).

Scaling and shearing are examples of transformations that are not isometric.

2. Using homogenous coordinates, define a point, $\mathbf{p} = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix}$, a vector, $\mathbf{v} = \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix}$, and a translation transformation, $\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$. What is the result of applying $\mathbf{T}$ to $\mathbf{p}$?

2. Using homogenous coordinates, define a point, $\mathbf{p} = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix}$, a vector, $\mathbf{v} = \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix}$, and a translation transformation, $\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$. What is the result of applying $\mathbf{T}$ to $\mathbf{p}$?

Solution:

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 + t_x \\ 5 + t_y \\ 1 \end{pmatrix}$$

Point $\mathbf{p}$ is shifted t_x to the right and t_y up.

3. What is the result of applying $\mathbf{T}$ to $\mathbf{v}$? Justify this result.

3. What is the result of applying **T** to **v**? Justify this result.

NOTHING :)

Solution:

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix}$$

A vector represents a direction, not a point in space, so it should be unaffected by translation transformations. (Translating a *direction* should not change anything about the *direction*.)

4. Write the transformation matrix for a 2D object that is reflected across the y-axis, translated up by 1 unit and right by 3 units, then rotated around the origin by 90° counterclockwise.

4. Write the transformation matrix for a 2D object that is reflected across the y-axis, translated up by 1 unit and right by 3 units, then rotated around the origin by 90° counterclockwise.

Solution:

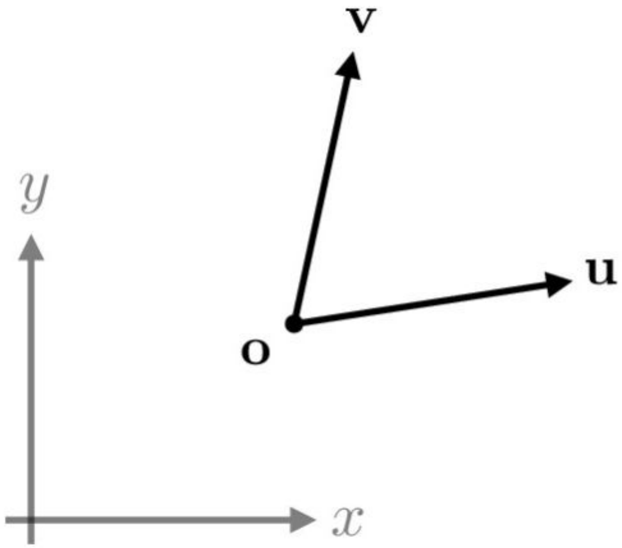
An important thing to remember here is that the transformation matrices are multiplied right to left. Keeping the order correct is important!

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

Coordinate Spaces

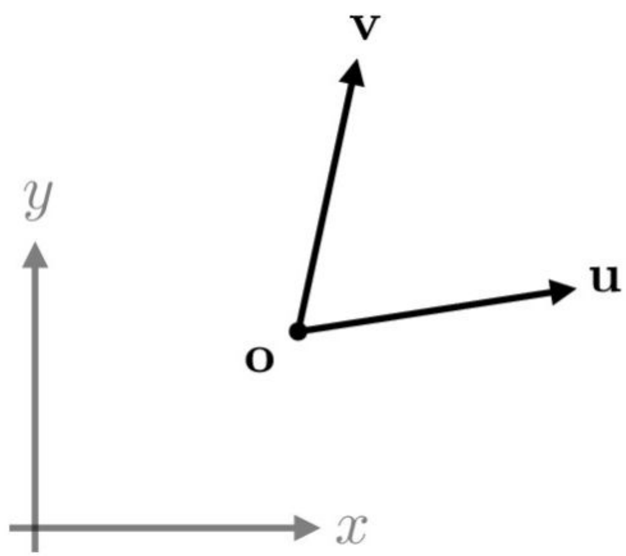
How do we specify coordinate spaces?

- In original coordinate space, specify:
  - Origin, **o**.
  - Two unit vectors, **u** and **v**.



Transform from (o, u, v) frame to (x, y) frame

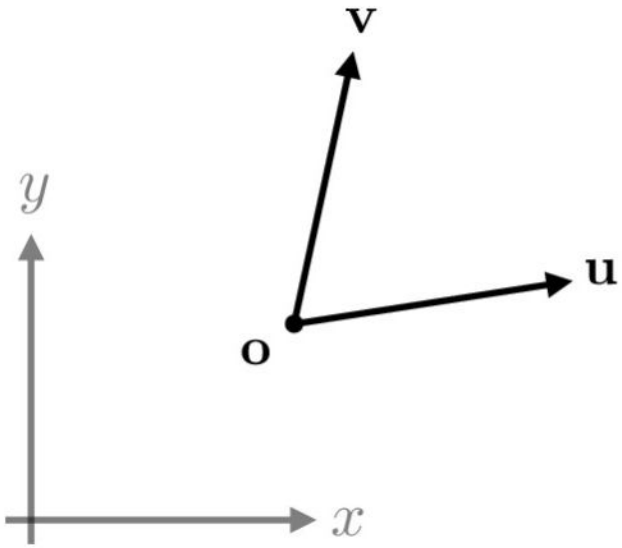
Transformation matrix: $\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix}$



Transform from (o, u, v) frame to (x, y) frame

Transformation matrix: $\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix}$

Check: where does point (1, 0) go?

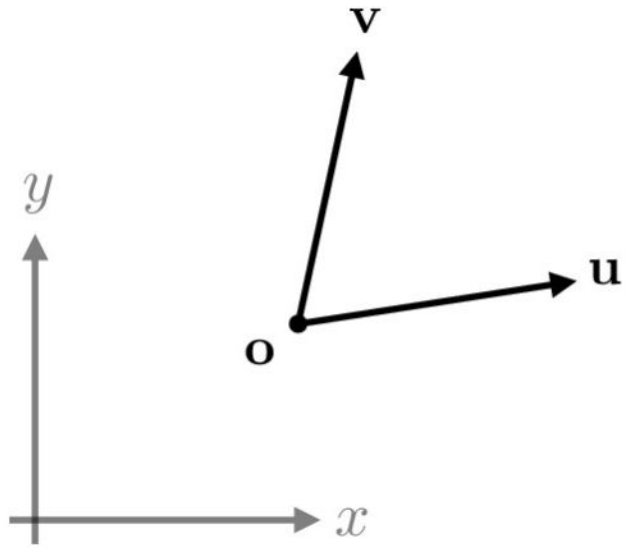


Transform from (o, u, v) frame to (x, y) frame

Transformation matrix: $\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix}$

Check: where does point (1, 0) go?

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} + \mathbf{o} \\ 1 \end{bmatrix}$$



Transform from (o, u, v) frame to (x, y) frame

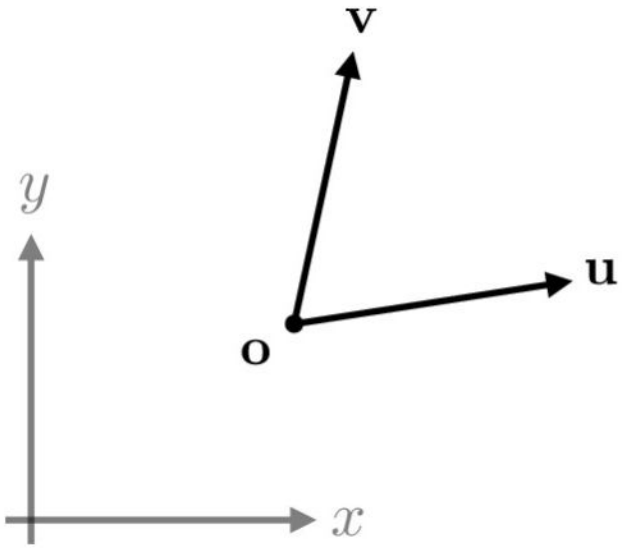
Transformation matrix: $\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix}$

Check: where does point (1, 0) go?

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} + \mathbf{o} \\ 1 \end{bmatrix}$$

Check: where does point (0, 1) go?

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{v} + \mathbf{o} \\ 1 \end{bmatrix}$$

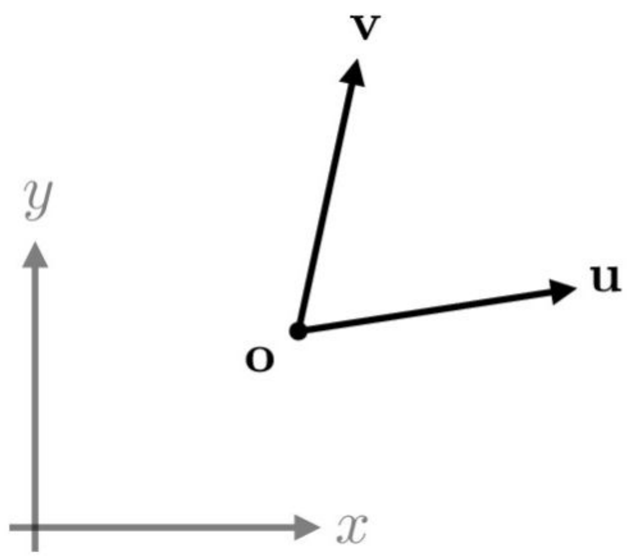


Coordinate System Transformation

- Transform from (u, v, o) frame to (x, y) frame using:

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} + \mathbf{o} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{v} + \mathbf{o} \\ 1 \end{bmatrix}$$

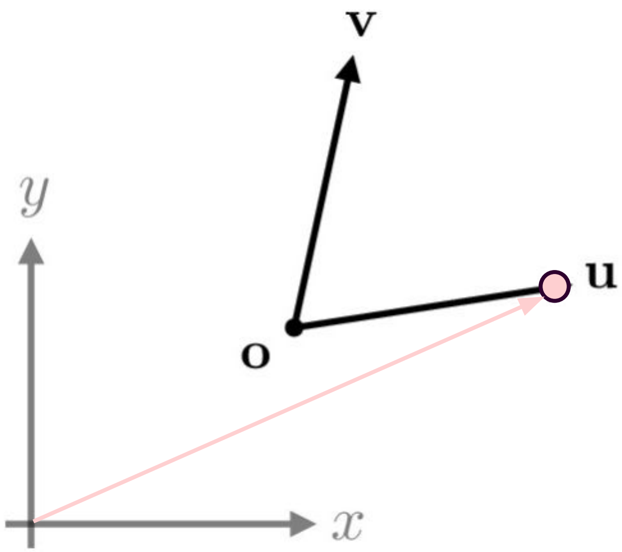


Coordinate System Transformation

- Transform from (u, v, o) frame to (x, y) frame using:

$$\boxed{\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} + \mathbf{o} \\ 1 \end{bmatrix}}$$

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{v} + \mathbf{o} \\ 1 \end{bmatrix}$$

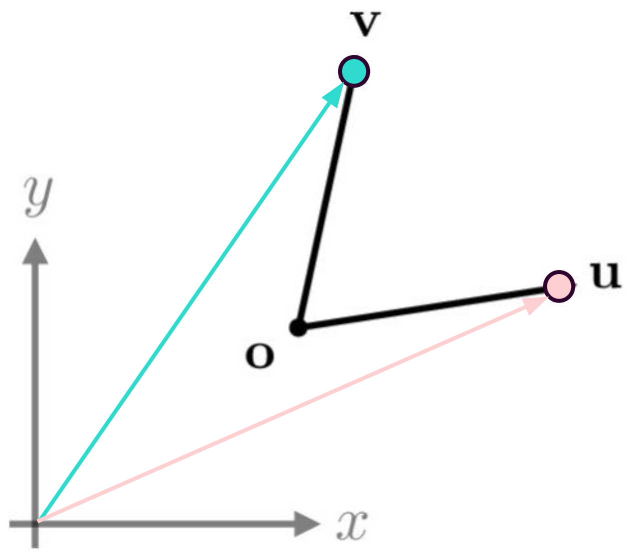


Coordinate System Transformation

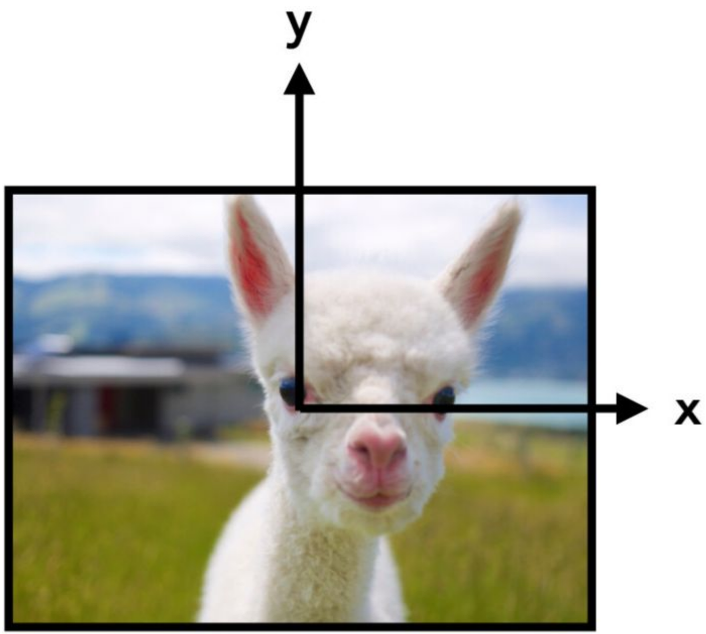
- Transform from (u, v, o) frame to (x, y) frame using:

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} + \mathbf{o} \\ 1 \end{bmatrix}$$

$$\boxed{\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{o} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{v} + \mathbf{o} \\ 1 \end{bmatrix}}$$



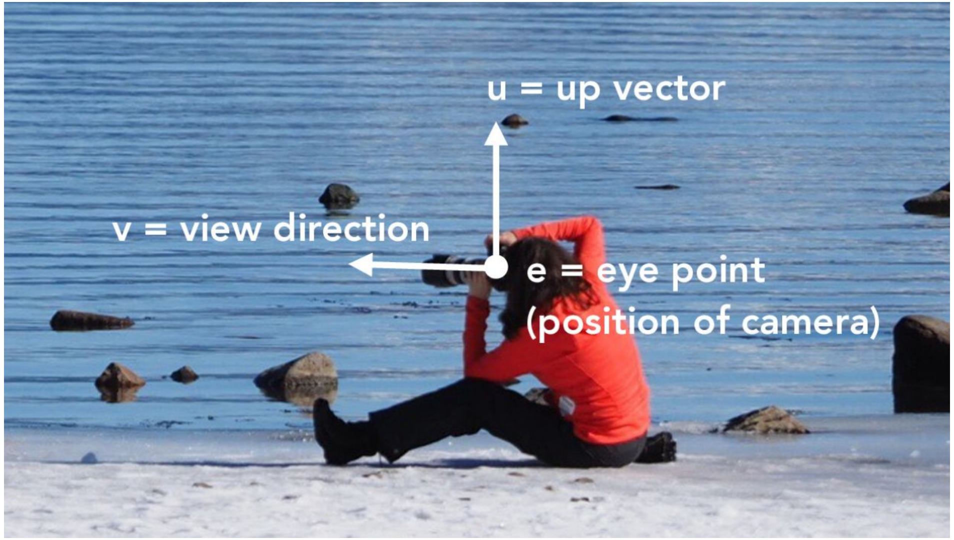
Camera coordinates



Resulting image
(z-axis pointing away from scene)

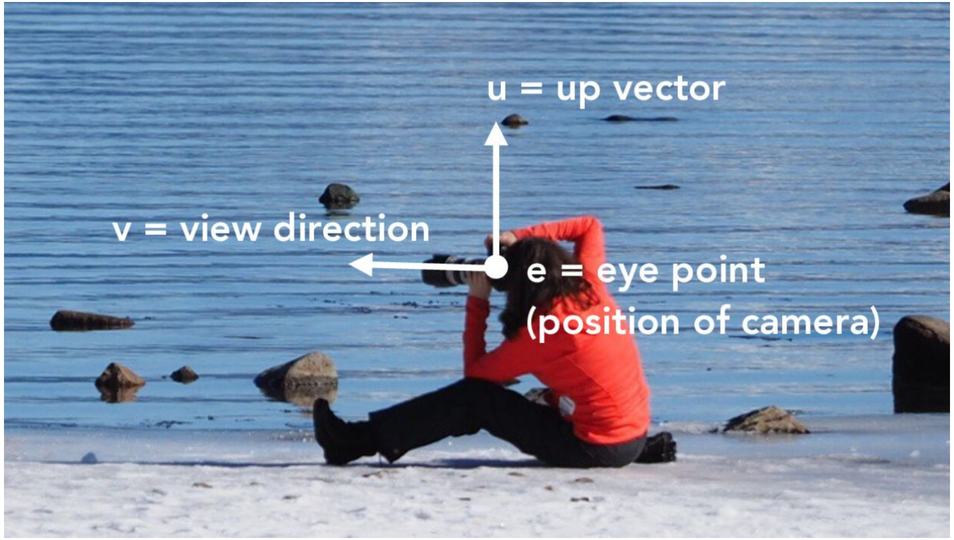
Given an object's camera coordinates, how do we find its world coordinates?

Camera has **e**, **u**, **v** in world coordinates



Given an object's camera coordinates, how do we find its world coordinates?

Camera has **e**, **u**, **v** in world coordinates



Transformation matrix

$$\begin{pmatrix} r_x & u_x & -v_x & e_x \\ r_y & u_y & -v_y & e_y \\ r_z & u_z & -v_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In world space coordinates, a teapot sits at $(0, 0, 0)$. Rebecca holds her camera at position $(1, 0, 2)$ — the eye point, $\mathbf{e}$, of the camera.

1. Rebecca points her camera at the teapot, intending to take a picture. What is the view direction, $\mathbf{v}$, of her camera?

In world space coordinates, a teapot sits at $(0, 0, 0)$. Rebecca holds her camera at position $(1, 0, 2)$ — the eye point, $\mathbf{e}$, of the camera.

1. Rebecca points her camera at the teapot, intending to take a picture. What is the view direction, $\mathbf{v}$, of her camera?

In world space coordinates, a teapot sits at $(0, 0, 0)$. Rebecca holds her camera at position $(1, 0, 2)$ — the eye point, **e**, of the camera.

2. In world space coordinates, in what direction does the positive z-axis of Rebecca's "standard" camera space point?

In world space coordinates, a teapot sits at $(0, 0, 0)$. Rebecca holds her camera at position $(1, 0, 2)$ — the eye point, **e**, of the camera.

3. The up vector, **u**, points in the direction of the positive y-axis of "standard" camera space. In world space coordinates, $\mathbf{u} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$. Calculate the right vector, **r**, which points in the direction of the positive x-axis of "standard" camera space.

In world space coordinates, a teapot sits at $(0, 0, 0)$. Rebecca holds her camera at position $(1, 0, 2)$ — the eye point, **e**, of the camera.

4. Write the matrix that transforms coordinates from "standard" camera space to world space. The inverse of this matrix is the *Look-At* matrix. What coordinate system transformation does the *Look-At* matrix perform?

Let’s Take Attendance.

- The attendance form is in the header of the CS 184 course website: cs184.eecs.berkeley.edu/su25/.
- Be sure to select <u>Week 2</u> and input your TA’s <u>secret word</u> 😉
- Any feedback? Let us know!