Tutorial 3
CS3241 Computer Graphics (AY22/23)

September 5, 2022
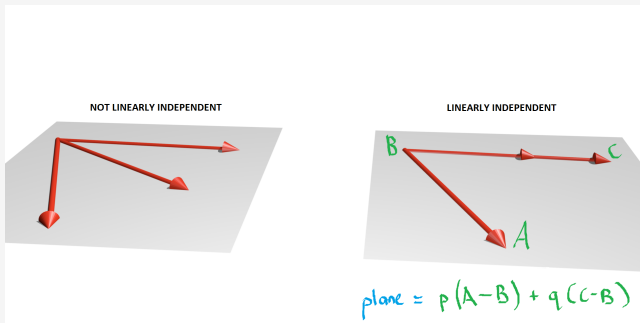
Wong Pei Xian

✉ e0389023@u.nus.edu

QUESTION 1
●○○○○○

QUESTION 2
○○○○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Question 1a

Given three points A, B, and C in 3D space, write an expression for the **normal vector** of the plane that contains the three points.

QUESTION 1
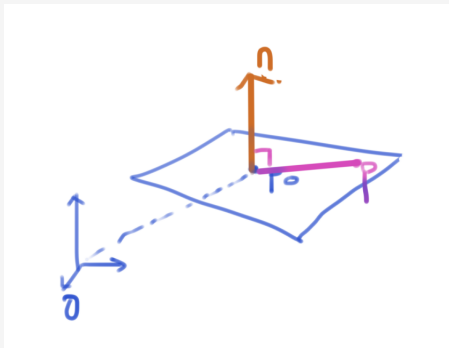○●○○○○   QUESTION 2
○○○○○   QUESTION 3
○○○   QUESTION 4
○○○○   QUESTION 5
○○   QUESTION 6
○○   QUESTION 7
○○   QUESTION 8
○○

Normal vector

The **linear combination** of any two **linearly independent** vectors forms a plane.

QUESTION 1
○○●○○○

QUESTION 2
○○○○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Question 1b

Given a point $R = [r_x \ r_y \ r_z]^T$ on a plane Π and a normal vector $n = [n_x \ n_y \ n_z]^T$ of the plane, write the implicit-form equation of the plane in the form $ax + by + cz + d = 0$.

Implicit form

$$n \cdot \left([x\,y\,z]^T - R\right) \quad = 0 \qquad (1)$$

$$n_x x + n_y y + n_z z - n \cdot R \quad = 0 \qquad (2)$$

QUESTION 1
○○○○●○
QUESTION 2
○○○○○
QUESTION 3
○○○
QUESTION 4
○○○○
QUESTION 5
○○
QUESTION 6
○○
QUESTION 7
○○
QUESTION 8
○○

Question 1c

What is the perpendicular distance of the point Q from the plane $ax + by + cz + d = 0$?

Perpendicular distance of point to plane

Let $Q = [q_x \ q_y \ q_z]^T$. Then the distance D from point to plane is

$$D = \left| \frac{aq_x + aq_y + aq_z + d}{\sqrt{a^2 + b^2 + c^2}} \right| \tag{3}$$

QUESTION 1
○○○○○○

QUESTION 2
●○○○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Question 2a

What does the homogenous coordinates $[6, 4, 2, 0.5]^T$ represent?

QUESTION 1
○○○○○○

QUESTION 2
○●○○○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Question 2a

What does the homogenous coordinates $[6, 4, 2, 0.5]^T$ represent?

$$\begin{bmatrix} x & y & z & w \end{bmatrix}$$
$$= \begin{bmatrix} x/w & y/w & z/w & 1 \end{bmatrix}$$

QUESTION 1
○○○○○○

QUESTION 2
○○●○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Question 2b

Why does OpenGL (and many other graphics APIs) use homogeneous coordinates to represent points?

Why homogenous coordinates?

1. Different representations for points aand vectors.
2. 4×4 Matrix multiplication
3. Perspective projection with matrix mult. and *perspective division*.

QUESTION 1
○○○○○○

QUESTION 2
○○○○●

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Point vector distinguishing

For **vector**; $w = 0$ e.g. $\begin{bmatrix} x & y & z & 0 \end{bmatrix}$.

For **point**; $w = 1$ e.g. $\begin{bmatrix} x & y & z & 1 \end{bmatrix}$.

> In this week's lecture, you will learn that to transform **normal vectors** instead of points, we use the matrix
>
> $$M_n = (M_t^{-1})^T$$
>
> where M_t is the upper left 3×3 submatrix.

QUESTION 1
○○○○○○

QUESTION 2
○○○○○

QUESTION 3
●○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○○

QUESTION 8
○○

Question 3

Which of the followings is the matrix that rotates objects about the fixed 3D point $[2\ 3\ 4]^T$, where the rotation axis is parallel to and in the same direction as the x-axis, and the rotation angle is θ?

A. $\mathbf{T}(-2, -3, -4) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(2, 3, 4)$

B. $\mathbf{T}(-2, -3, -4) \cdot \mathbf{R}_x(-\theta) \cdot \mathbf{T}(2, 3, 4)$

C. $\mathbf{T}(2, 3, 4) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}(-2, -3, -4)$

D. $\mathbf{T}(2, 3, 4) \cdot \mathbf{R}_x(-\theta) \cdot \mathbf{T}(-2, -3, -4)$

E. $\mathbf{T}(-2, -3, -4) \cdot \mathbf{R}_x(\theta)$

QUESTION 1
oooooo

QUESTION 2
ooooo

QUESTION 3
o●o

QUESTION 4
oooo

QUESTION 5
oo

QUESTION 6
oo

QUESTION 7
oo

QUESTION 8
oo

Things to note about transformation

Order of Transformations

- Conceptually, the transformation on the right is applied first

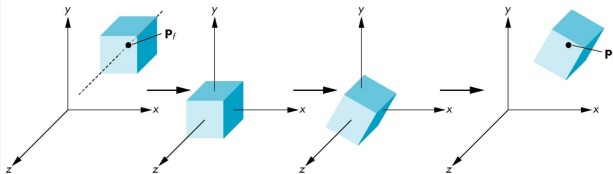$$\mathbf{p}' = \mathbf{ABCp} = \mathbf{A}(\mathbf{B}(\mathbf{Cp}))$$

Things to note about transformation

Rotation About a Fixed Point Other than the Origin

1. Move fixed point to origin

2. Rotate

3. Move fixed point back

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_f)\,\mathbf{R}(\theta)\,\mathbf{T}(-\mathbf{p}_f)$$

Ans: $T(2,3,4) \cdot R(\theta) \cdot T(-2,-3,-4)$.

Question 4

What is the inverse matrix of

$$
M = \begin{bmatrix}
s_1 r_{11} & s_1 r_{12} & s_1 r_{13} & t_1 \\
s_2 r_{22} & s_2 r_{22} & s_2 r_{23} & t_2 \\
s_3 r_{33} & s_3 r_{32} & s_3 r_{33} & t_3 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

Step 1: Decompose

You can tell there is a S, R, and T matrix multiplied together in some order to give M.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{22} & r_{22} & r_{23} & 0 \\ r_{33} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S = \begin{bmatrix} s_{11} & 0 & 0 & 0 \\ 0 & s_{22} & 0 & 0 \\ 0 & 0 & s_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question 1
○○○○○○

Question 2
○○○○○

Question 3
○○○

Question 4
○○○●○

Question 5
○○

Question 6
○○

Question 7
○○

Question 8
○○

Step 2: Order

$$M = TSR.$$

Thus $M^{-1} = R^{-1}S^{-1}T^{-1}$.

QUESTION 1
oooooo

QUESTION 2
ooooo

QUESTION 3
ooo

QUESTION 4
oooo

QUESTION 5
oo

QUESTION 6
oo

QUESTION 7
oo

QUESTION 8
oo

Step 3: Substitute and Compute

$$\mathbf{M}^{-1} = \mathbf{R}^{-1} \cdot \mathbf{S}^{-1} \cdot \mathbf{T}^{-1} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & 0 \\ r_{12} & r_{22} & r_{32} & 0 \\ r_{13} & r_{23} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/s_1 & 0 & 0 & 0 \\ 0 & 1/s_2 & 0 & 0 \\ 0 & 0 & 1/s_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -t_1 \\ 0 & 1 & 0 & -t_2 \\ 0 & 0 & 1 & -t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}^{-1} = \begin{bmatrix} r_{11}/s_1 & r_{21}/s_2 & r_{31}/s_3 & -(r_{11}t_1/s_1)-(r_{21}t_2/s_2)-(r_{31}t_3/s_3) \\ r_{12}/s_1 & r_{22}/s_2 & r_{32}/s_3 & -(r_{12}t_1/s_1)-(r_{22}t_2/s_2)-(r_{32}t_3/s_3) \\ r_{13}/s_1 & r_{23}/s_2 & r_{33}/s_3 & -(r_{13}t_1/s_1)-(r_{23}t_2/s_2)-(r_{33}t_3/s_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

QUESTION 1
000000

QUESTION 2
00000

QUESTION 3
000

QUESTION 4
0000

QUESTION 5
●○

QUESTION 6
00

QUESTION 7
00

QUESTION 8
00

Question 5

Let $M_1 = TR$.

Let $M_2 = RT$.

Is $M_1 = M_2$? Prove it.

Proof and counterexample

In general $TR \neq RT$.

i.e. $M_1 \neq M_2$. Counterexample:

$$T(1, 0, 0)R_z(\pi/2)P(1, 0, 0) \neq R_z(\pi/2)T(1, 0, 0)P(1, 0, 0)$$
$$P(1, 1, 0) \neq P(0, 2, 0)$$

Question 6

The computation of $\sin \theta$ and $\cos \theta$ is considered relatively slow for some real-time rendering applications.

However, when the angle θ is very small, we can make use of the small-angle approximation: $\sin \theta \approx \theta$, and $\cos \theta \approx 1 - \theta_{small}/2$, for better speed.

Write the 4x4 matrix for rotation about the z-axis by a very small rotation angle θ_{small}, which is given in radians.

Substitute in

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\cos\theta \approx 1 - \theta_{small}/2$$

$$\sin\theta \approx \theta_{small}$$

QUESTION 1
○○○○○○

QUESTION 2
○○○○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
●○

QUESTION 8
○○

Question 7

In OpenGL, what is the purpose of transforming vertices to window space?

QUESTION 1
○○○○○○

QUESTION 2
○○○○○

QUESTION 3
○○○

QUESTION 4
○○○○

QUESTION 5
○○

QUESTION 6
○○

QUESTION 7
○●

QUESTION 8
○○

For the rasterizer!

Question 8

For the following application program code fragment, write down the transformations that are actually applied to each vertex vi.

```
glMatrixMode( GL_MODELVIEW );
glLoadMatrixd( A );
glPushMatrix();
    glBegin( GL_POINTS ); glVertex3dv( v1 ); glEnd();
    glMultMatrixd( B );
    glPushMatrix();
        glMultMatrixd( C );
        glBegin( GL_POINTS ); glVertex3dv( v2 ); glEnd();
    glPopMatrix();
    glBegin( GL_POINTS ); glVertex3dv( v3 ); glEnd();
    glPushMatrix();
        glMultMatrixd( D );
        glPushMatrix();
            glMultMatrixd( E );
            glBegin( GL_POINTS ); glVertex3dv( v4 ); glEnd();
        glPopMatrix();
        glBegin( GL_POINTS ); glVertex3dv( v5 ); glEnd();
    glPopMatrix();
glPopMatrix();
glMultMatrixd( F );
glPushMatrix();
    glMultMatrixd( G );
glPopMatrix();
glBegin( GL_POINTS ); glVertex3dv( v6 ); glEnd();
```

Attendance taking

Thanks! Get the slides here after the tutorial.

https://trxe.github.io/cs3241-notes