10/08/2020

Vertices and vectors are represented as $[x, y, z, w]^T$ (4x1 matrix)
Transformations are represented as 4x4 matrices.

Applying transformations to vectors and vertices
Hence, [4x4]*[4x1] → transformed [4x1]

The CTM (M*P) it is a 4x4 matrix
The OpenGL functions from CG13 (used S2)

(1) glTranslate, (2) glRotate, (3) glScale

And the OpenGL projection transformations

(4) glOrtho and (5) glFrustum

Generate a [4x4] matrix
(maybe as simple as the identity matrix)

And this 4x4 matrix is multiplied by M (1, 2, and 3) or by P (4, 5)
[4x4]*[4x4] → 4x4 new P, New M –> New CTM
The CTM [4x4] is applied to each vertex on the scene before rendering
[4x4]*[4x1] for each vertex and this yields a new [4x1].

Translation

See previous lectures.

Rotation (of vertices).
Slide 11

Given a vertex at location (x, y) where the vector from origin to (x, y)
Is at an angle of ϕ with the X axis.
We rotate the vertex by an angle of θ and obtain a new vertex at location (x', y')
The vector from the origin to (x', y') is at an angle of $(\phi + \theta)$.

The distance to the origin is fixed (R). The angle WRT to the X axis is changed from ϕ to $\phi + \theta$
x, y are given by $r \times cos(\phi), r \times sin(\phi)$
x' y'; are $r \times cos(\phi + \theta), r \times sin(\phi + \theta)$

After the rotation, the new coordinates of the point (x, y)

Slide 12
Are

x'=x cos θ – y sin θ

y' = x sin θ + y cos θ

Note that the zZ value is not changed here z'= z.
Rotation on the plan z=d (d can be 0 and z does not change)
Referred to as rotation about z.

Eventually, rotation about Z can be expressed as
p'=$\mathbf{R}_z$(q)**p**

R_z is the rotation matrix about Z.

Slide 13 shows Rz(thetha)

$$\begin{matrix} \cos\theta & -\sin(\theta) & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \quad * \quad \begin{matrix} X \\ Y \\ z \\ 1 \end{matrix} \quad = \quad \begin{matrix} x' \\ y' \\ z' \\ 1 \end{matrix} \quad = \quad \begin{matrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ Z \\ 1 \end{matrix}$$

We started with a normal Vertex ended out with a new normal vertex.

This can be expressed via matrix multiplication

Result in slide 12, producing the matrix of slide 13.

Exercise
Perform $R_y(\theta)$ by $[x, y, z, 1]^T$
Look at slide 14

Submit to TRACS attendance (to be opened promptly)
by 5:40.

Inverse rotation by (θ) – get the matrix with $(-\theta)$.

OGL glRotate(θ, u, v, w) is rotating about the vector
[u, v, w, 0] by θ.
Example
glRothate(45, 0, 1, 0) 45 degrees about Y.

Produce the inverse matrix about Z
Replace θ by $-\theta$

cos $(-\theta)$ = cos (θ)
sin (-θ) = - *sin* (θ)

Scaling in 15

For inverse scaling by S_x, S_y, S_z, use: $1/S_x$, $1/S_y$, $1/S_z$

We can always find the inverse matrix using Linear Algebra
But this will take way more time

Slide 16 special cases of scaling

Different scaling factors in different axes
might change a square to rectangle
Circle to ellipse.

Inverses in slide 17.

Slide 18
We can concatenate for example for smooth transformation.
Additionally one may want to use double-buffers

Note that in general A*B ≠ B *A

Order in slide 19.

We use [4x1] column to represent vectors and vertices.
[4x4]*[4x1] → [4x1]   (post multiply)

Alternatively, use row elements [1x4]*[4x4] → [1X4] (slide 19)

Due to the post multiplication we get an artifact to be considered.
Additionally, the fact that rotation and scaling are "around" the origin generates artifacts.
The origin is the fixed-point WRT the two transformation.

how to rotate (scale) about an arbitrary point

Slide 20 shows rotation about an arbitrary vector
This can be done by glRotate()

CG-12-21 rotation about an arbitrary fixed point
Slide 21 has an issue due to post multiply
Hence, we need to translate the point to origin, rotate, and translate back

CG13-10 Show how to do this in OGL
Due to the post multiple
Inverse translation, rotation, and translation

The Midterm will include material from CG12 and CG13 with emphasis on
The OGL side.

CG12
Need to know the basics excluding slides 20 and above

CG13 up to slide 19

CG11 ignore:
3, 4 6, 10 7 15 and above

CG 10 ignore 18, 20 and above