CmpE240 April 4, 22 Part II-1

April 4 (Monday).

Topics: 1.30 GE focused on Shadow Computation.

Example: Given

1. x_w-y_w-z_w world coordinate Right Hand System. $\vec{P_i} + \lambda(\vec{P_s} - \vec{P_i})$

$\vec{R}$

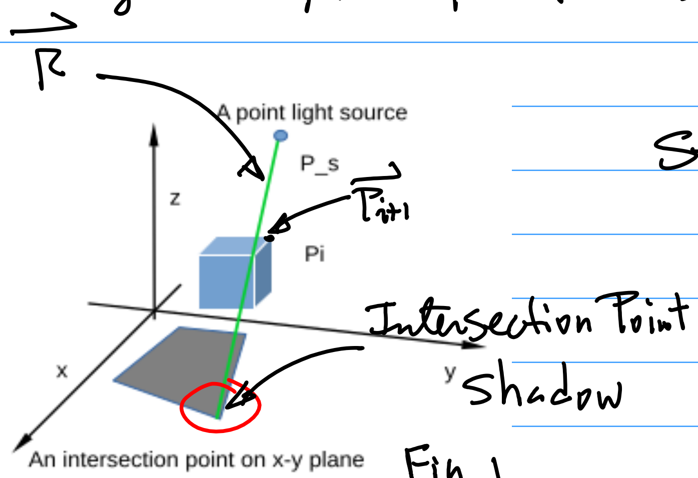


Fig. 1

2. A Given Cube $\{\vec{P_i}(x_i, y_i, z_i) \mid i = 1, 2, \cdots, N\}$

Counter Clockwise.

3. A given Point light Source $\vec{P_s}(x_s, y_s, z_s)$

4. Generate Ray Equation / Ray Cast x_w-y_w plane, Produces Shadow if Blocked by the Cube.

References:

2021F-101b-notes-cmpe240-2021-12-1.pdf

Step 1. Generate A Ray Cast / Equation

$$\vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i}) \quad \ldots (1)$$

Starting from $\vec{P_s}$, Passing through $\vec{P_i}$

$$\vec{d}(x, y) = \vec{P_s} - \vec{P_i} \quad \ldots (1b)$$

Intersection Point: $\vec{P_i}$ Between Ray $\vec{R}(x,y)$, and x_w-y_w plane.

Step 2. To find the intersection Point on x_w-y_w plane

Define a plane equation.

a. Define A Normal Vector

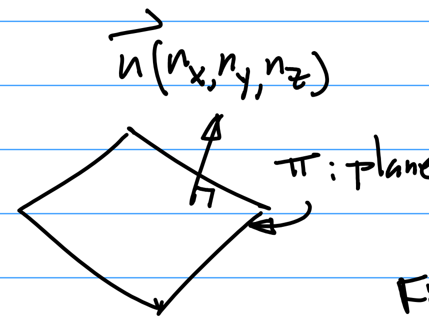


Fig. 2

use the normal vector to define the plane π.

b. Form a vector (on a plane) from $\vec{P_i}$ and $\vec{P_{i+1}}$, $(\vec{P_{i+1}} - \vec{P_i})$ a line

$$\vec{n} \cdot (\vec{P_{i+1}} - \vec{P_i}) = 0 \quad \ldots (2)$$

Take

$\vec{v}(v_x, v_y, v_z)$, $\vec{a}(a_x, a_y, a_z)$ to Replace $\vec{P_i}$, $\vec{P_{i+1}}$

Assume $\vec{a}(a_x, a_y, a_z)$ is a known vector; and $\vec{v}(v_x, v_y, v_z)$ is unknown, but an arbitrary point on the plane π.

Hence, Eqn (2) becomes

$$\vec{n} \cdot (\vec{v} - \vec{a}) = 0 \quad \ldots (2^*)$$

Now, find the intersection point defined by the ray Eqn (1). In order to that, we will need to find λ.

Since, the intersection point $\vec{P_i}$ is the common point by the ray and the plane π. we have

$$\begin{cases} \vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i}) & \ldots (3a) \\ \vec{n} \cdot (\vec{v} - \vec{a}) = 0 & \ldots (3b) \end{cases}$$

$\vec{n}(n_x, n_y, n_z)$, Normal Vector has to be known,

$\vec{a}(a_x, a_y, a_z)$ is known on π.

Starting from the plane Eqn (3b).

$$\vec{n} \cdot (\vec{v} - \vec{a}) = 0$$

where $\vec{v} = \vec{R}$, e.g.

$$\vec{n} \cdot (\vec{v} - \vec{a}) \Big|_{\vec{v} = \vec{R}} = 0 \quad \ldots (4)$$

$$\vec{n} \cdot (\vec{R} - \vec{a}) \Big|_{\vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i})} = 0$$

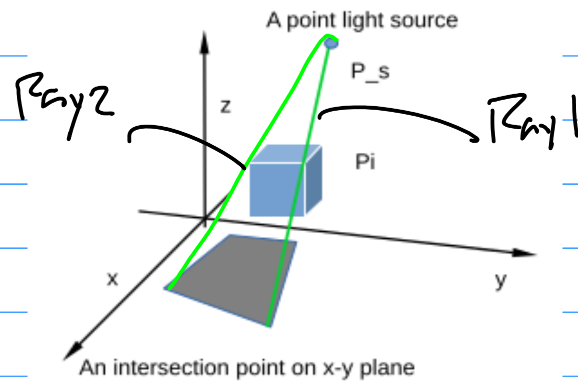
$$\vec{n} \cdot (\vec{P_i} + \lambda(\vec{P_s} - \vec{P_i}) - \vec{a}) = 0$$

$$\vec{n} \cdot \vec{P_i} + \lambda \vec{n} \cdot (\vec{P_s} - \vec{P_i}) - \vec{n} \cdot \vec{a} = 0$$

$$\lambda \vec{n} \cdot (\vec{P_s} - \vec{P_i}) = \vec{n} \cdot \vec{a} - \vec{n} \cdot \vec{P_i}$$

$$\lambda = \frac{\vec{n} \cdot \vec{a} - \vec{n} \cdot \vec{P_i}}{\vec{n} \cdot (\vec{P_s} - \vec{P_i})} = \frac{\vec{n} \cdot (\vec{a} - \vec{P_i})}{\vec{n} \cdot (\vec{P_s} - \vec{P_i})} \quad \ldots (5)$$

Note λ is <u>NOT</u> the intersection pt. it allows us to use Ray Eqn (1) to find the intersection.



use Eqn (5) to find more than one intersection

April 7th (Wed)

Example: Ref (see pp. 1. github LectureNotes, 2021F-101b-...)

Given $\vec{P_s}(-20, 110, 200)$, A vertex of a given Cube $\vec{P_i}(100, 100, 110)$. Find the intersection pt to Draw Shadow.

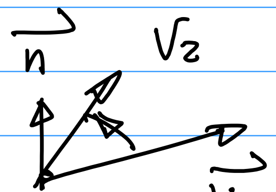
Sol First, Ray Equation, Eqn(1):

$$\vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i}) \quad \ldots (1)$$

then, plane Eqn for X_w-Y_w plane

$$\vec{n} \cdot (\vec{v} - \vec{a}) = 0$$

Find $\vec{n}$,



$$\vec{V_1} \times \vec{V_2} = \vec{n} \quad \ldots (2)$$

Let $\vec{V_1} = x_w \vec{i}$, $\vec{V_2} = y_w \vec{j}$

$$\vec{n} = x_w \vec{i} \times y_w \vec{j}$$

$$= (0, 0, 1)$$

Then, 2nd find λ for the Ray Equation, from Eqn.

$$\lambda = \frac{\vec{n} \cdot (\vec{a} - \vec{P_i})}{\vec{n} \cdot (\vec{P_s} - \vec{P_i})} \quad \ldots (5)$$

where $\vec{a} = (0, 0, 0)$, Hence,

$$\lambda = \frac{-\vec{n} \cdot \vec{P_i}}{\vec{n} \cdot (\vec{P_s} - \vec{P_i})}$$

Now, hand Calculation, also for Coding

$$\lambda = \frac{-\vec{n} \cdot \vec{P_i}}{\vec{n} \cdot (\vec{P_s} - \vec{P_i})}$$

$$= -\frac{n_x x_i + n_y y_i + n_z z_i}{n_x(x_s - x_i) + n_y(y_s - y_i) + n_z(z_s - z_i)}$$

From the given Condition $\vec{n}(0, 0, 1)$, Therefore

$$\lambda = \frac{-\cancel{n_z} z_i}{\cancel{n_z}(z_s - z_i)} = -\frac{z_i}{z_s - z_i}$$

$$= -\frac{110}{200 - 110} = -\frac{110}{90} = -11/9$$

Now, back to the Ray Equation

$$\vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i})$$

$$\vec{R} = (100,100,110) - \frac{11}{9}(-20-100,\ 110-100,\ 200-110)$$

$$= (100,100,110) - \frac{11}{9}(-120, 10, 90)$$

$$= \left(\frac{1100\times 120}{9},\ -\left(100 - \frac{110}{9}\right),\ 110-110\right)$$

$$= \left(\frac{1100\times 120}{9},\ \frac{110}{9} - 100,\ 0\right) = (246.7, 87.8, 0)$$

please finish this calculation.

Now, Coding Part. Same Code on github.

2018F-116-11diffuse20181114.cpp

Note:

a. Define Normal vector $\vec{n}$ for X_w-Y_w plane

```
world.X[47] = 0; world.Y[47] = 0; world.Z[47] = 1; // normal vector for x-y plane
```

b. Note the "typedef struct" for Defining 3D points.

```
typedef struct {
    float X[UpperBD], Y[UpperBD], Z[UpperBD];
} pworld;
```

```
    pworld  world;
    pviewer viewer;
```

$$\lambda = \frac{\vec{n}\cdot(\vec{a} - \vec{P_i})}{\vec{n}\cdot(\vec{P_s} - \vec{P_i})} \quad \cdots (5)$$

Now, λ Calculation. Eqn (5)

$$= -\frac{n_x x_i + n_y y_i + n_z z_i}{n_x(x_s - x_i) + n_y(y_s - y_i) + n_z(z_s - z_i)}$$

```
//----------lambda for Intersection pt on xw-yw plane----------*
float temp = (world.X[47]*(world.X[46]-world.X[45])
            +(world.Y[47]*(world.Y[46]-world.Y[45]))
            +(world.Z[47]*(world.Z[46]-world.Z[45]));
float lambda = temp / ((world.X[47]*(world.X[45]-world.X[7]))
                  +(world.Y[47]*(world.Y[45]-world.Y[7]))
                  +(world.Z[47]*(world.Z[45]-world.Z[7])));
float lambda_2 = temp / ((world.X[47]*(world.X[45]-world.X[6]))
                    +(world.Y[47]*(world.Y[45]-world.Y[6]))
                    +(world.Z[47]*(world.Z[45]-world.Z[6])));
```

Note, Substitute λ to Ray Equation to find the intersection point

```
//-----------ray equation to find intersection pts-----------*
world.X[48] = world.X[45] + lambda*(world.X[45] - world.X[7]); // Intersection pt p7
world.Y[48] = world.Y[45] + lambda*(world.Y[45] - world.Y[7]); // Intersection pt p7
world.Z[48] = 0.0;

world.X[49] = world.X[45] + lambda_2*(world.X[45] - world.X[6]);  //intersection pt p6
world.Y[49] = world.Y[45] + lambda_2*(world.Y[45] - world.Y[6]);  //intersection pt p6
world.Z[49] = 0.0;
```

$\vec{R} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i}) \rightarrow$

$$x = x_i + \lambda(x_s - x_i)$$
$$y = y_i + \lambda(y_s - y_i)$$
$$z = z_i + \lambda(z_s - z_i)$$

Assignment in-Class Show & Tell. Implement Intersection Computation on LPC 1769, "Show & Tell" Demo in Class. On April 11 (Monday),

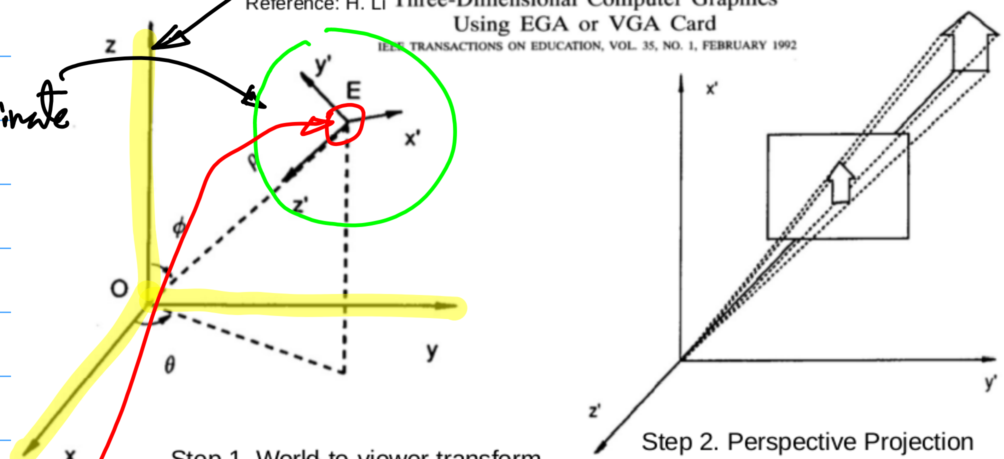
To Be Able to Display 3D Graphics on 2D Display Devices. Let's Define Transformation Pipeline. 1. Define World-Coordinate System; Right Hand System $x_w - y_w - z_w$

3D Transformation Pipeline Technique

Reference: H. Li Three-Dimensional Computer Graphics Using EGA or VGA Card
IEEE TRANSACTIONS ON EDUCATION, VOL. 35, NO. 1, FEBRUARY 1992

2. Viewer Coordinate System $x_v - y_v - z_v$

Left-Hand System



Step 1. World-to-viewer transform

Step 2. Perspective Projection

3. Virtual Camera is Located $\vec{E}$ (e_x, e_y, e_z)

$$\mathbf{T} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_p = x_e\left(\frac{D}{z_e}\right)$$
$$y_p = y_e\left(\frac{D}{z_e}\right)$$

Harry Li, Ph.D.

Example: Display Shadows on 2D Display Device.

Step1. World To Viewer Transform.

$$T=\begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots (1)$$

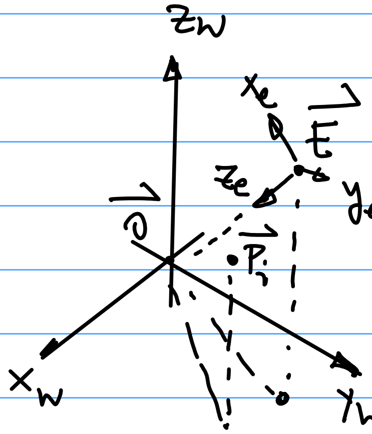


Fig.1.

Everything is defined in the World Coordinate System x_w-y_w-z_w including a Virtual Camera.

$\vec{E}(x_e, y_e, z_e)$, x_e-y_e-z_e "Viewer" Coordinate System.

Given $\vec{P_i}(x_i, y_i, z_i)$ in x_w-y_w-z_w World Coordinate, Represent this point in x_e-y_e-z_e Coordinate System.

$$\vec{P_i}: \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad \cdots (1\text{-}b)$$

$\vec{P_i}$

Assume $\vec{E}(x_e, y_e, z_e) = (200, 200, 200)$

Physical meaning of Transformation Matrix T.

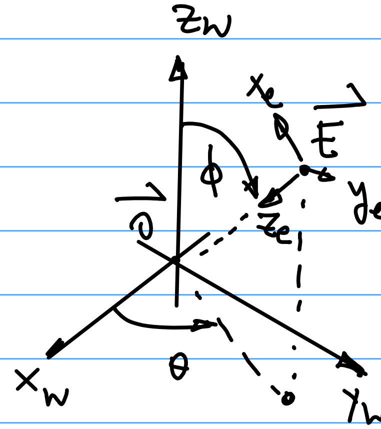


Fig.2

θ: Angle from the dash line on x_w-y_w plane w.r.t positive x_w-axis

ϕ: Angle Between z_w & z_e.

ρ (rho): $\rho = \sqrt{x_e^2 + y_e^2 + z_e} \quad \cdots (2)$

distance from $\vec{E}$ to the origin $\vec{O}$ of x_w-y_w-z_w.

Suppose $\vec{E}(200, 200, 200)$ is given, Find $\cos\theta$, $\sin\theta$, $\cos\phi$, $\sin\phi$ for T-matrix.

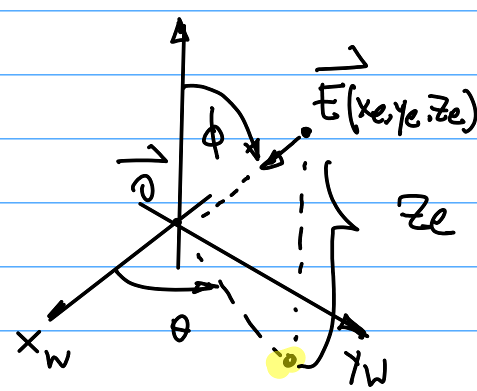
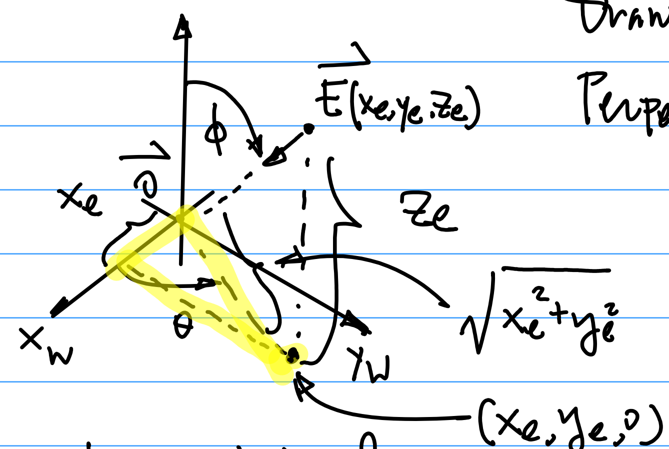


Fig.3

Draw A Line Passing through $\vec{E}$ Perpendicular to X_w-Y_w $\Rightarrow$ Form an intersection point.

Draw A Line Passing through the intersection point on X_w-Y_w plane on the plane and Perpendicular to X_w-axis.

Fig.4

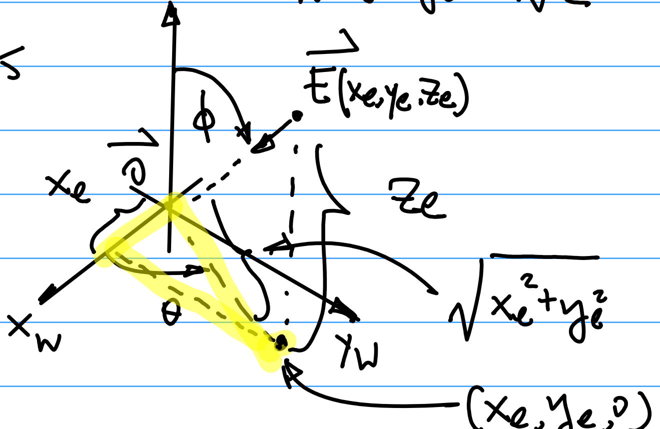


We can form a triangle on X_w-Y_w plane, as in Fig.4, hence

$$\cos\theta = \frac{x_e}{\sqrt{x_e^2+y_e^2}} = \frac{200}{200\sqrt{2}} = \frac{\sqrt{2}}{2}$$

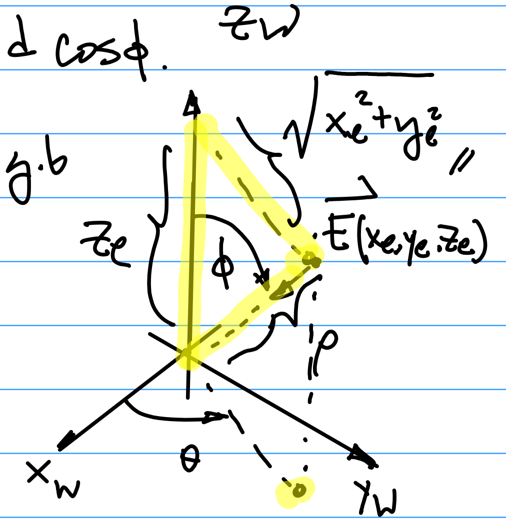
similarly, $\sin\theta = \frac{y_e}{\sqrt{x_e^2+y_e^2}} = \frac{200}{200\sqrt{2}} = \frac{\sqrt{2}}{2}$

Fig.5



Find $\cos\phi$.

Fig.6



Draw A Line Passing $\vec{E}(x_e, y_e, z_e)$ Perpendicular to Z_w-axis

$$\cos\phi = \frac{z_e}{\rho} = \frac{z_e}{\sqrt{x_e^2+y_e^2+z_e}}$$

$$= \frac{200}{200\sqrt{3}} = \sqrt{3}/3$$

$$\sin\phi = \frac{\sqrt{x_e^2+y_e^2}}{\sqrt{x_e^2+y_e^2+z_e}} = \frac{200\sqrt{2}}{200\sqrt{3}}$$

$$= \frac{\sqrt{2}\cdot\sqrt{3}}{3} = \frac{\sqrt{6}}{3}$$

Homework: Due April 18th (Monday)

1. Draw A world Coordinate System X_w-Y_w-Z_w axis, with X_w Red, Y_w Green, Z_w Blue

2. Draw a cube, size length = 100, floats 10 unit above $x_w - y_w$ plane. In other word $\vec{P_i}(100, 100, 110)$;

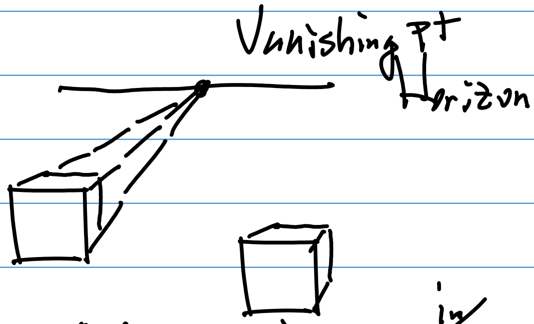
3. Draw a point light source $\vec{P_s}(-20, 110, 200)$, and RayCast to connect $\vec{P_s}$ to $\vec{P_i}$; use Green color.

4. Compute the shadow point $\vec{P_i'}$, Draw the intersection point to link $\vec{P_s} - \vec{P_i} - \vec{P_i'}$

Note: You may want to adjust the $\vec{P_s}$ position, so this $\vec{P_i'}$ (Intersection point) can be visible on your Display

Step 2. Perspective Projection

$$x_p = x_e\left(\frac{D}{z_e}\right)$$
$$y_p = y_e\left(\frac{D}{z_e}\right) \quad \ldots (3)$$



Eqn(3) project a point $\vec{P_i}(x_i, y_i, z_i)$ in $x_e - y_e - z_e$ onto a 2D Display Device, like LCD.