10-31-2018 3D Graphics Engine: Shade Computation

CMPE240 Advanced Microprocessor Systems. Oct 31 2018
Harry Li

Today's Topics: 3D G.E. (Graphics Engine) Design.
math. Formulation On "Shade" Computation. github/hualili 2018F-114-Lec5.

Example: Find the Intersection pt. ON x_w-y_w plane. From Eqn (4),

$$\begin{cases} \vec{P} = \vec{P_s} + \lambda(\vec{P_s} - \vec{P_i}) \\ \vec{n}\cdot(\vec{a} - \vec{v}) = 0 \end{cases}, \quad \vec{n}\cdot(\vec{a} - \vec{v})\Big|_{\vec{v}=\vec{P}} = 0, \quad \text{Hence,}$$

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

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

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

Okay

$$\lambda = \frac{(n_x, n_y, n_z)\cdot(x_a - x_s, y_a - y_s, z_a - z_s)}{(n_x, n_y, n_z)(x_s - x_i, y_s - y_i, z_s - z_i)}$$

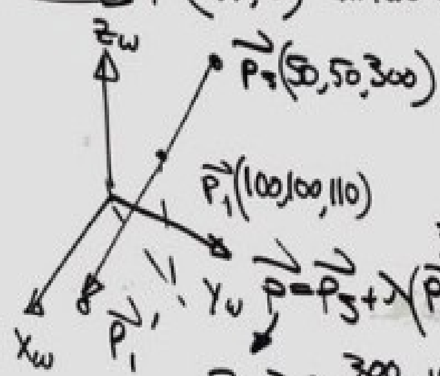
$$= \frac{n_x(x_a - x_s) + n_y(y_a - y_s) + n_z(z_a - z_s)}{n_x(x_s - x_i) + n_y(y_s - y_i) + n_z(z_s - z_i)} \quad \ldots (5^*)$$

```
float tmp = nx*(xa-xs)+ny*(ya-ys)+nz*(za-zs);
lamda = tmp /(nx*(xs-x[i])+ny*(ys-y[i])
        + nz*(zs-z[i]));
```

Note: 1° In the Project & Homework, use Top $\{\vec{P_i}\}$

2° λ is Not the intersection pt. use $\vec{P} = \vec{P_s} + \lambda(\vec{P_s} - \vec{P_i})$ to find Intersection pt.

$\vec{P}(x,y,z)$ Intersection pt. $z = 0$ $\vec{a}(0,0,0)$



From (5*), $\vec{n}(0,0,1)$

$$\lambda = \frac{0 + 0 + 1\cdot(0 - 300)}{0 + 0 + 1\cdot(300 - 110)} = -\frac{300}{190}$$

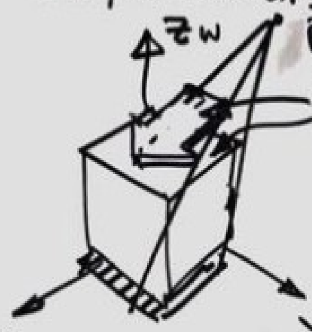
$$\vec{P} = \vec{P_s} + \lambda(\vec{P_s} - \vec{P_1}) = (50,50,300) - \frac{300}{190}(-50,-50,190)$$

$$z = 300 - \frac{300}{190}\cdot 190 = 0$$

10-31-2018 3D Graphics Engine: Shade Computation

CMPE240 Adv. Micro. Oct. 31, 2018. 2/.

Note: Bring your Laptop/LPC1769 module Board for Programming Implementation (Starting Next Lecture).

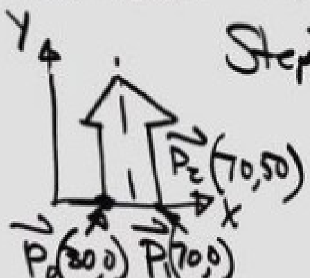


Example: Construction/Design of 3D Floating Arrow ("Red" colour).

Design Requirements: (1) Arrow points to X_w Direction (2) Arrow on top of the cube w/ 15~25. (3) Cube size: 100 x 100 x 100

Ref: 2018F-114-Lec5. Step 1: Design 2D Pattern;

Step2: "Swaping" to get the right Orientation.



$\vec{P_0}(0,30)$, $\vec{P_1}(0,70)$, $\vec{P_2}(50,70)$, $\vec{P_3}(50,85)$

$\vec{P_4}(75,50)$, $\vec{P_5}(50,15)$, $\vec{P_6}(50,30)$

Step3. Translation ("move" up to the top of the Cube)

$$Z = \underbrace{100}_{\text{Size of the Cube}} + \underbrace{10}_{\text{Elevation of the Cube}} + \underbrace{20}_{\text{On top of the Cube}} = 130$$

$\vec{P_0}(0,30,130)$, $\vec{P_1}(0,70,130)$, $\vec{P_2}(50,70,130)$,

$\vec{P_3}(50,85,130)$, $\vec{P_4}(75,50,130)$, $\vec{P_5}(50,15,13)$

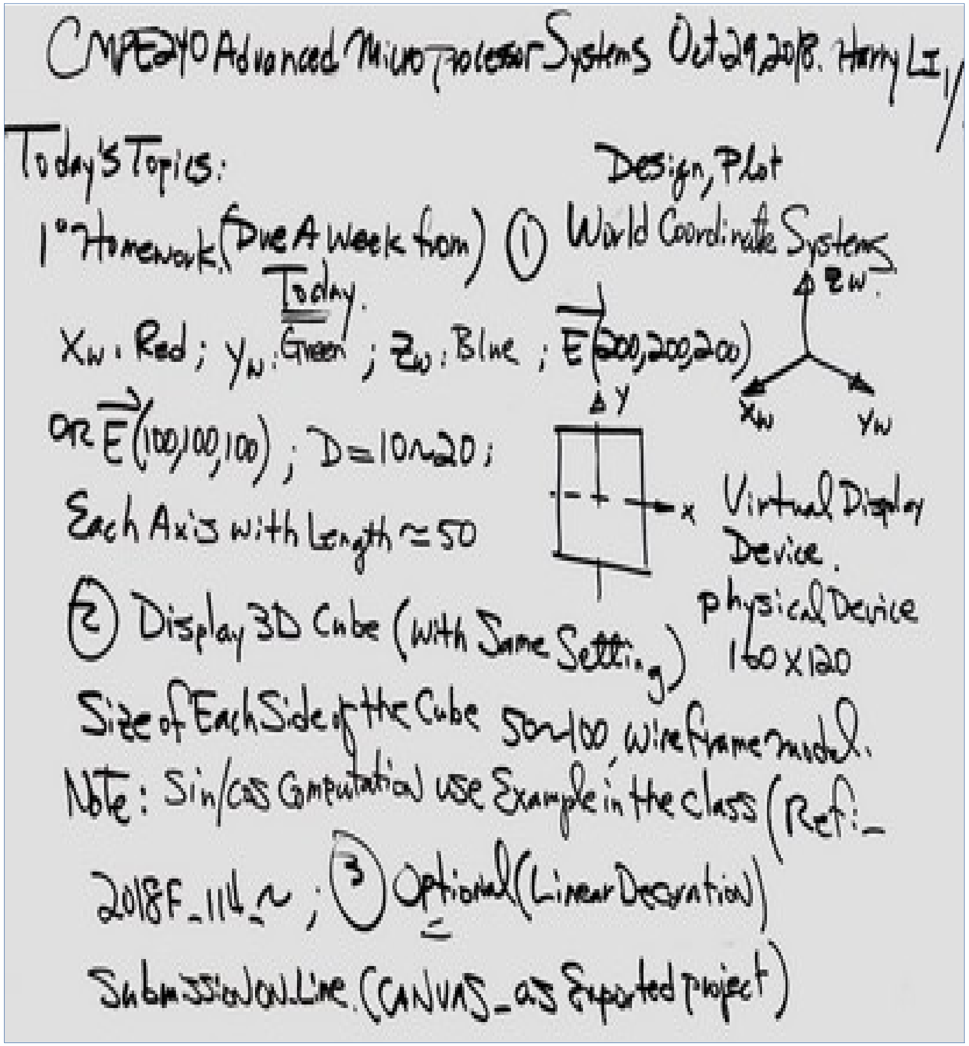
$\vec{P_6}(50,30,130)$ --- Top Plane.

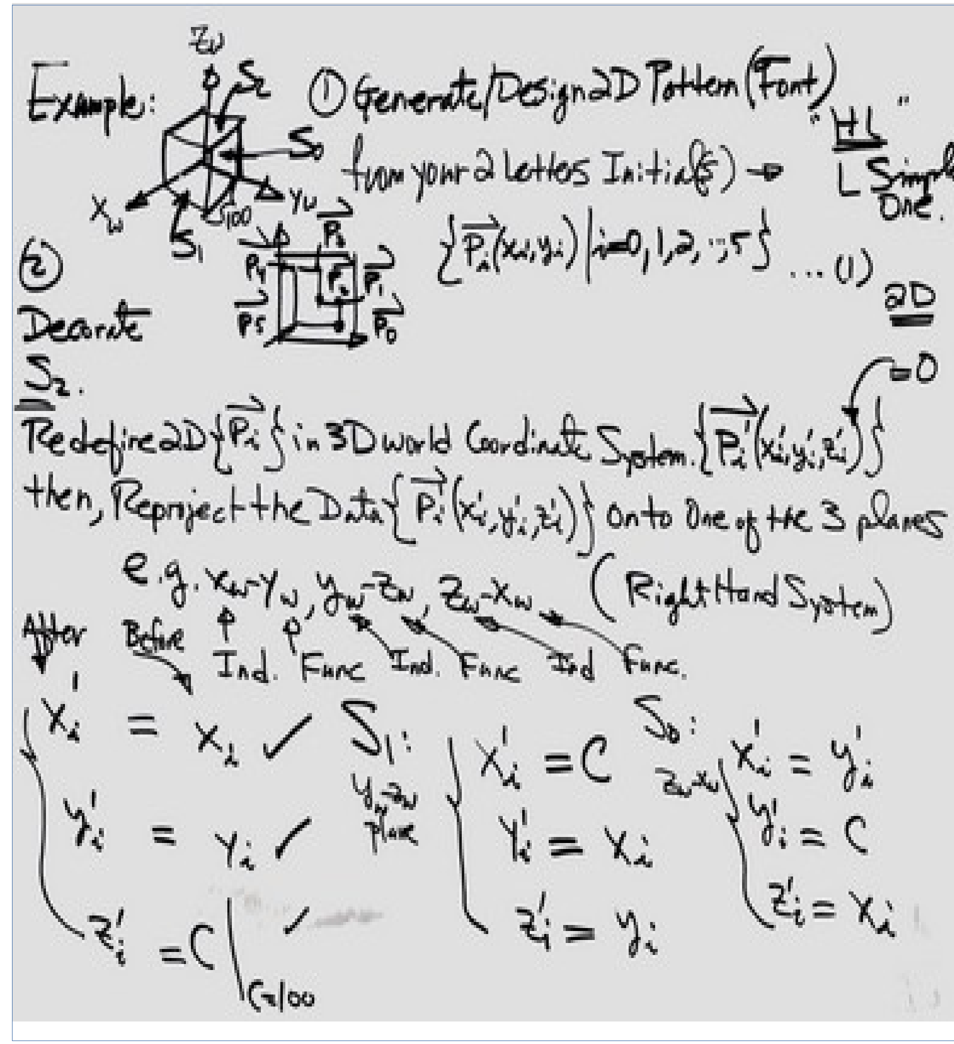
Step4. "Solid" Object → "Height" = 5 Base Plane.

$\vec{P_{0,Base}}(0,30,130-5)$, $\vec{P}(0,70,130-5)$ ...

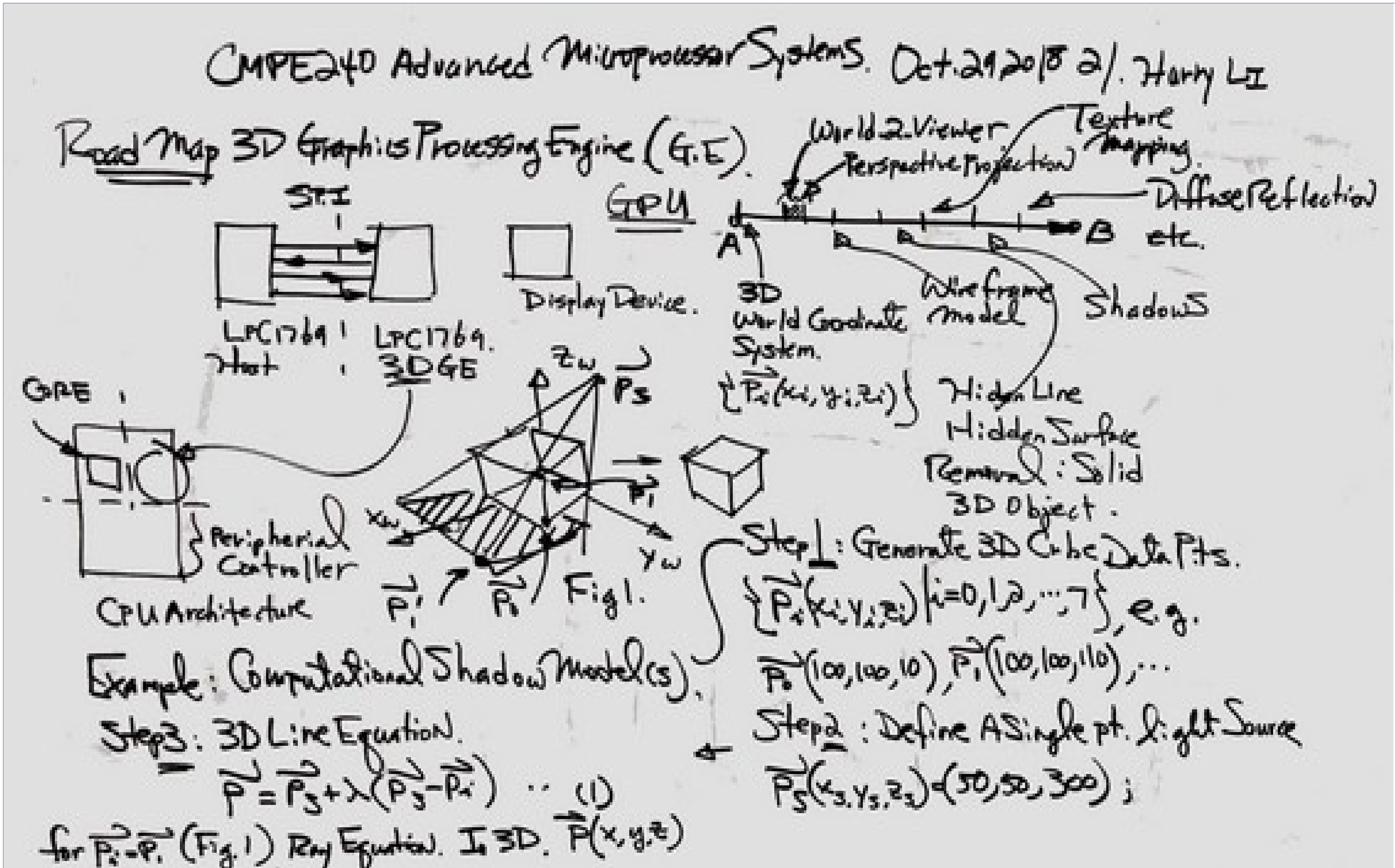
$\vec{P_6}(50,30,130-5)$

10-29-2018 3D Graphics Engine: Shade Computation





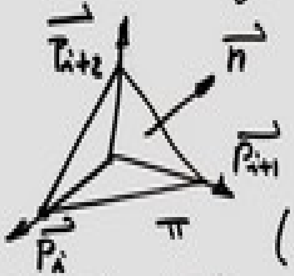
10-29-2018 3D Graphics Engine: Shade Computation



Harry Li, Ph.D

10-29-2018 3D Graphics Engine: Shade Computation

Step 4. plane Equation $\Rightarrow$ Normal Vector $\vec{n}(n_x, n_y, n_z)$



$$(\vec{P}_{i+1}-\vec{P}_i)\times(\vec{P}_{i+2}-\vec{P}_i)$$

$$= \begin{bmatrix} \vec{i} & \vec{j} & \vec{k} \\ x_{i+1}-x_i & y_{i+1}-y_i & z_{i+1}-z_i \\ x_{i+2}-x_i & y_{i+2}-y_i & z_{i+2}-z_i \end{bmatrix} \quad \dots (2)$$

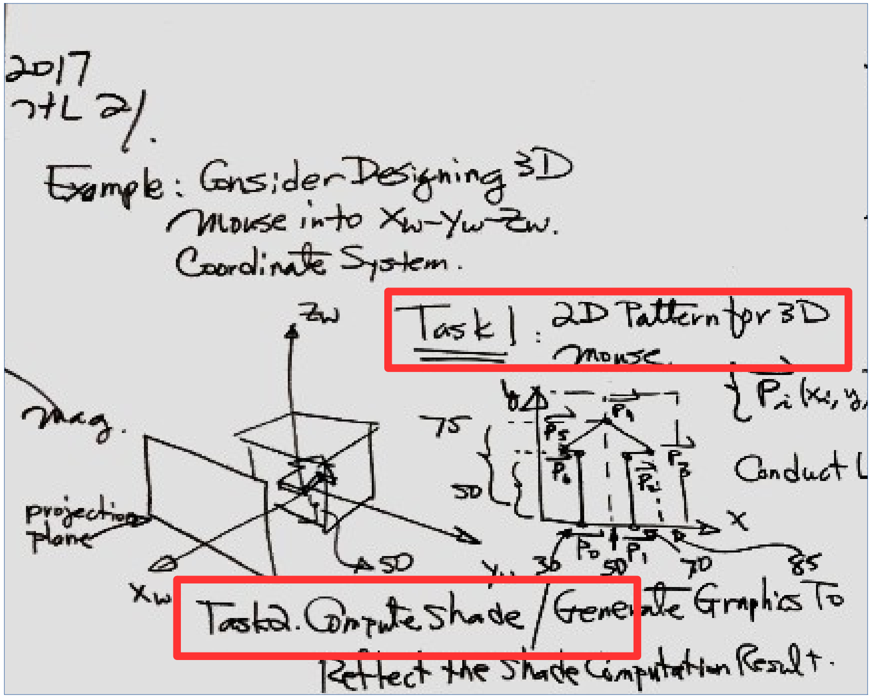
$$\vec{n}\cdot(\vec{a}-\vec{v})=0 \quad \dots (3)$$ where $\vec{a}=(a_x, a_y, a_z)$ Any Arbitrary Known pt. On the plane.

Step 5. Intersect pt.

$$\begin{cases} \vec{p}=\vec{P}_s+\lambda(\vec{P}_s-\vec{P}_i) \\ \vec{n}\cdot(\vec{a}-\vec{v})=0 \end{cases} \quad \dots (4)$$

$$\vec{n}\cdot(\vec{a}-\vec{v})\Big|_{\vec{v}=\vec{p}}=0, \quad \vec{n}\cdot(\vec{a}-\vec{p})\Big|_{\vec{p}=\vec{P}_s+\lambda(\vec{P}_s-\vec{P}_i)}=0$$

2017 Design of 3D Virtual Display



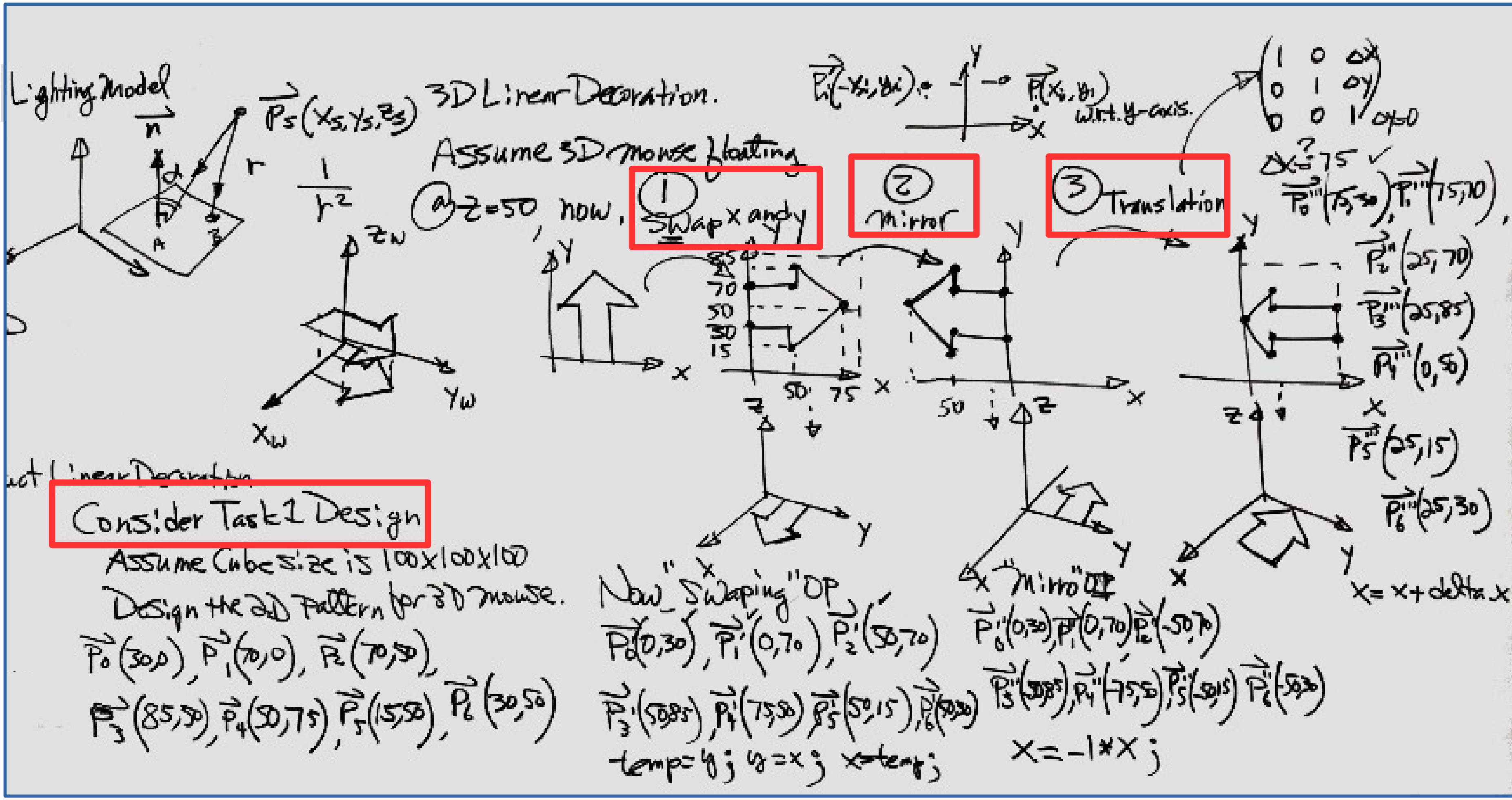
Three tasks:

1. 2D pattern for 3D mouse and perform 3D linear decoration algorithm

2. Compute shade

3. Hidden Line/Surface Removal

Design 2D Cursor Pattern then 3D Decoration



Harry Li, Ph.D

3D Decoration

CMPE163 Introduction To Computer Graphics & AR HL. 3/.

Now, with Linear Decoration, we can change $\{\vec{P}_i'''(x_i''', y_i''') \mid i=0,1,\cdots,6\}$ to 3D mouse, by adding z-dimension, such that $z_i'''=50$, Hence, we have

$$\{\vec{P}_i(x_i, y_i, z_i) \mid i=0,1,2,\ldots,6;\ z_i=50\}$$

$\vec{P}_0(75,30,50)$

$\vec{P}_1(75,70,50)$, $\vec{P}_2(25,70,50)$, $\vec{P}_3=(25,85,50)$,

$\vec{P}_4(0,50,50)$, $\vec{P}_5(25,15,50)$, $\vec{P}_6(25,30,50)$

Make the pattern with Thickness=5.

$S1: \{\vec{P}_i(x_i, y_i, z_i) \mid i=0,1,2,\cdots,6\}$

Then, (Layer Beneath S_1)

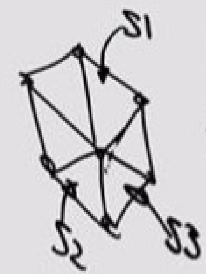
$S2: \{\vec{P}_i'(x_i, y_i, z_i-5) \mid i=0,1,2,\cdots,6\}$

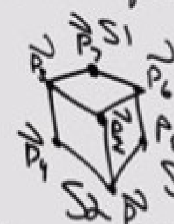
"Wireframe" → Solid Object

Hidden Line/Surface Removal.
- Simple ~ Based ON Vector Cross Product
- Z-Buffer Algorithm.

Background

1° Define Vertices of 3D Object(s) in Counter ClockWise Direction. (When Viewing the Object from the outside)



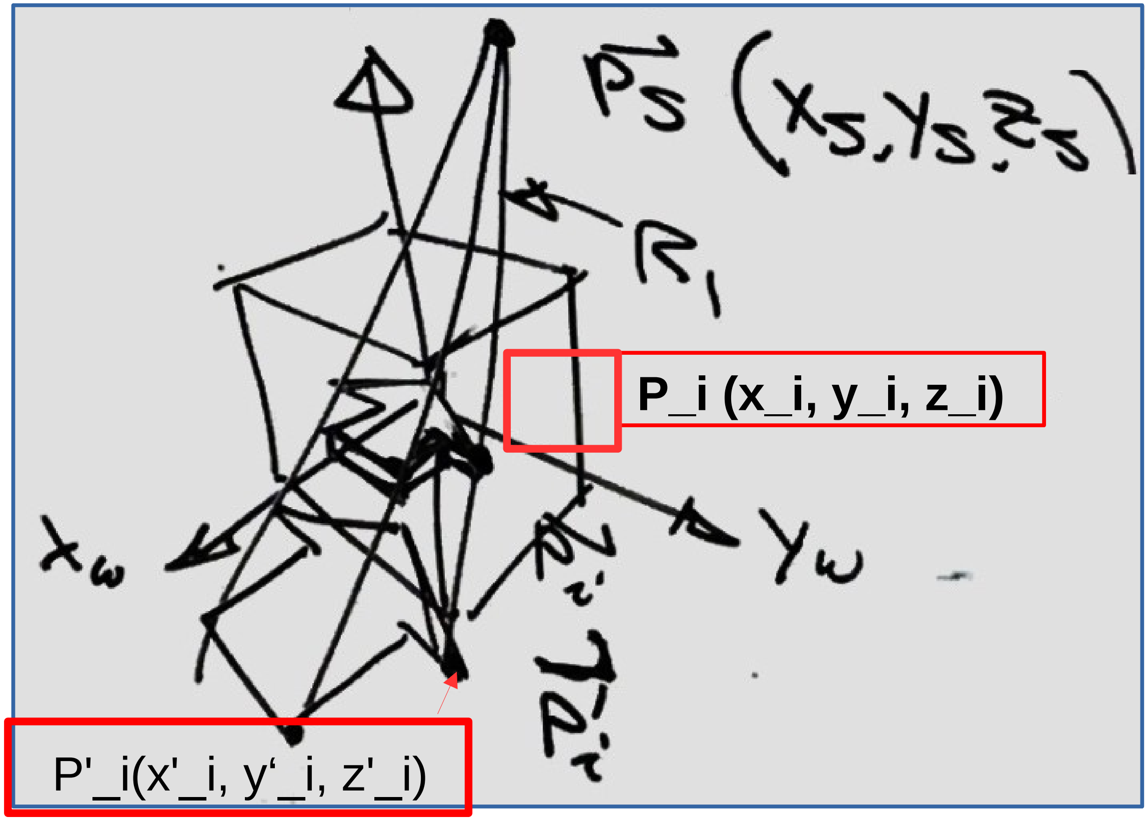


$S_1: \vec{P}_2 - \vec{P}_6 - \vec{P}_7 - \vec{P}_3$ C.C.

$S_2: \vec{P}_1 - \vec{P}_2 - \vec{P}_3 - \vec{P}_4$ CC

$S_3: \vec{P}_1 - \vec{P}_5 - \vec{P}_6 - \vec{P}_7$ C.C

Single Point Light Source



Give a single point light source P_s, and the 3D cursor as

{P_i | i = 0, 1, …, N-1}

Find the intersection points on Xw-Yw plane,

{P'_i | i = 0, 1, …, N-1}

e.g.,

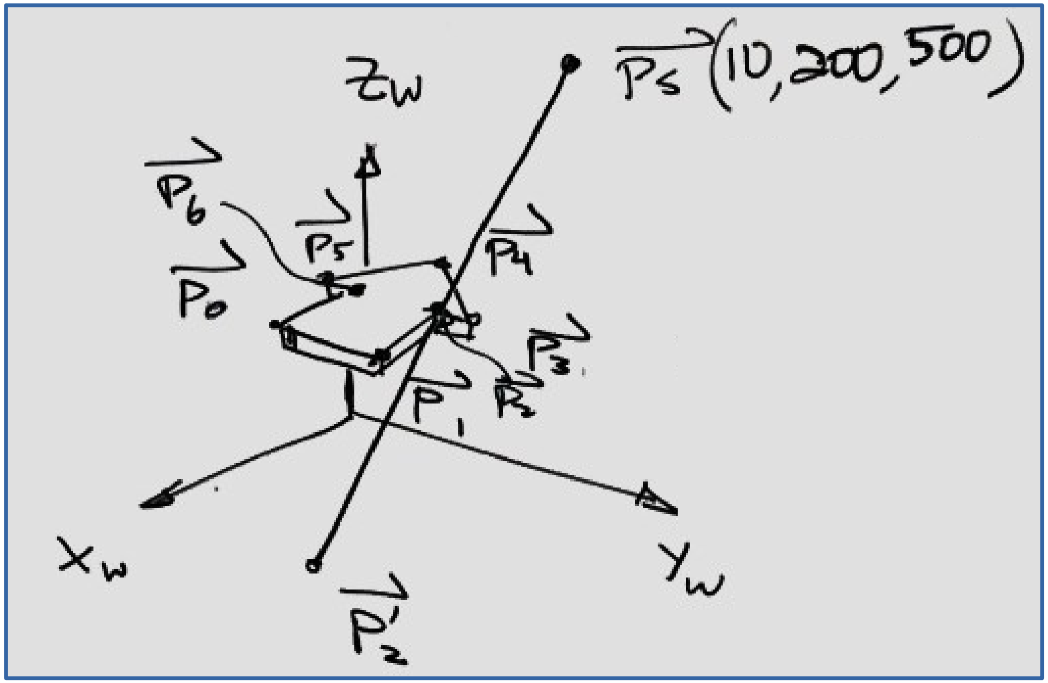
P_i (x_i, y_i, z_i) from the 3D cursor, linked to single point light source

P_s(x_s, y_s, z_s) and formed intersection point

P'_i(x'_i, y‘_i, z'_i)

Harry Li, Ph.D

Computing Shade From A Single Point Light Source (1)



Ray Equation (Line)

$$\vec{R_2} = \vec{P_S} + \lambda(\vec{P_S} - \vec{P_2}) \quad (1)$$

Plane equation below:

$$\vec{n} \cdot (\vec{r} - \vec{a}) = 0 \quad (2)$$

Where the normal vector of the Xw-Yw plane is

$$\vec{n} = (0,0,1)$$

And the known point vertex a is:

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

for $i = 0, 1, 2, \cdots, 6$, we have generalized

$$\vec{R_i}(x_i, y_i, z_i) = \vec{P_S}(x_S, y_S, z_S) + \lambda(x_S - x_i, y_S - y_i, z_S - z_i) \quad (1.1)$$

7 Ray equations for each vertex of the 3D cursor

Computing Shade From A Single Point Light Source (2)

Substitute the known condition into the plane equation, we have

$$(0,0,1) * (\vec{v} - \vec{a}) \Big|_{\vec{a}=0} = 0 \qquad (3)$$

Note vector V is the common shared point (intersection) of the ray vector, so we have

$$\vec{n} * \vec{v} \Big|_{\vec{v} = \vec{R}_i} = 0 \qquad (4)$$

e.g.

$$\vec{n} * \vec{R}_i \Big|_{\vec{R}_i = \vec{P}_s + \lambda(\vec{P}_s - \vec{P}_i)} = 0$$

Hence,

$$\vec{n} * (\vec{P}_s + \lambda(\vec{P}_s - \vec{P}_i)) = 0 \qquad (5)$$

Or

$$\vec{n} * \vec{P}_s + \lambda \vec{n} * (\vec{P}_s - \vec{P}_i) = 0 \qquad (5.1)$$

Solve for lamda,

$$\lambda = -\frac{\vec{n} * \vec{P}_s}{\vec{n} * (\vec{P}_s - \vec{P}_i)} \qquad (6)$$

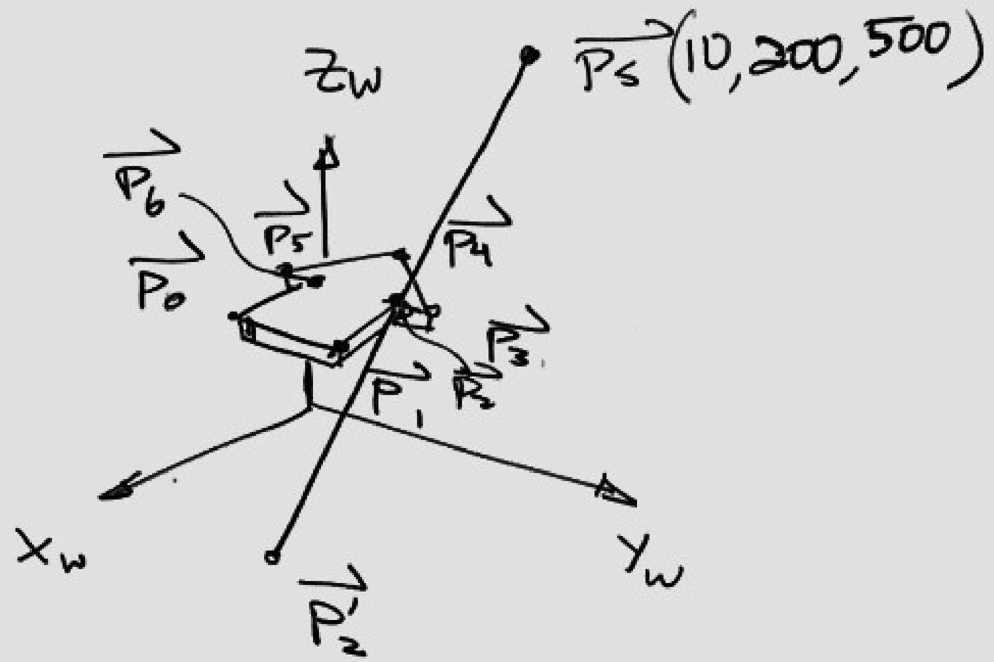
e.g.

$$\lambda = \frac{-(n_x, n_y, n_z) * (x_s, y_s, z_s)}{(n_x, n_y, n_z) * (x_s - x_i, y_s - y_i, z_s - z_i)} \qquad (6.1)$$

Computing Shade From A Single Point Light Source (3)

Example: Compute the Shade point $\vec{P_2'}$

Given $\vec{P_2}(25,70,50)$, $\vec{P_s}(10,200,500)$



Then substitute the lamda back to the ray equation to find the intersection point as follows

$$\vec{P_2'} = \vec{P_s} + \lambda(\vec{P_s} - \vec{P_2})$$

$$= (10,200,500) - \frac{10}{9}(10-25, 200-70, 500-50)$$

$$= \left(10+\frac{150}{9}, 200-\frac{1300}{9}, \underbrace{500-\frac{4500}{9}}_{0}\right)$$

From equation (6), compute lamda

$$\lambda = -\frac{\vec{n} * \vec{P_s}}{\vec{n} * (\vec{P_s} - \vec{P_2})} = -\frac{n_x \cdot x_s + n_y y_s + n_z z_s}{n_x(x_s - x_2) + n_y(y_s - y_2) + n_z(z_s - z_2)}$$

$$= -\frac{0+0+1 \cdot z_s}{0+0+1 \cdot (z_s - z_2)} = -\frac{z_s}{z_s - z_2} = -\frac{500}{500-50} = -\frac{500}{450} = -\frac{10}{9}$$

The rest of the points can be computed similarly.

Harry Li, Ph.D