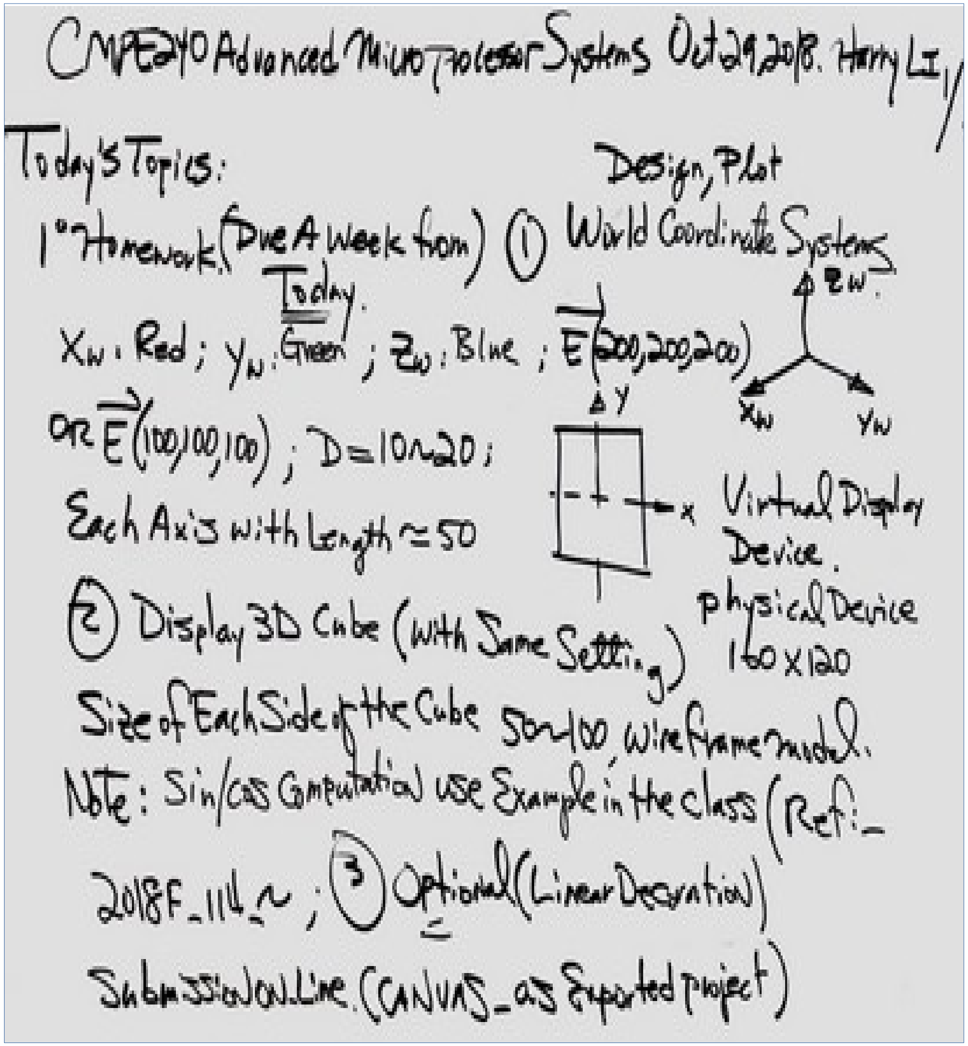
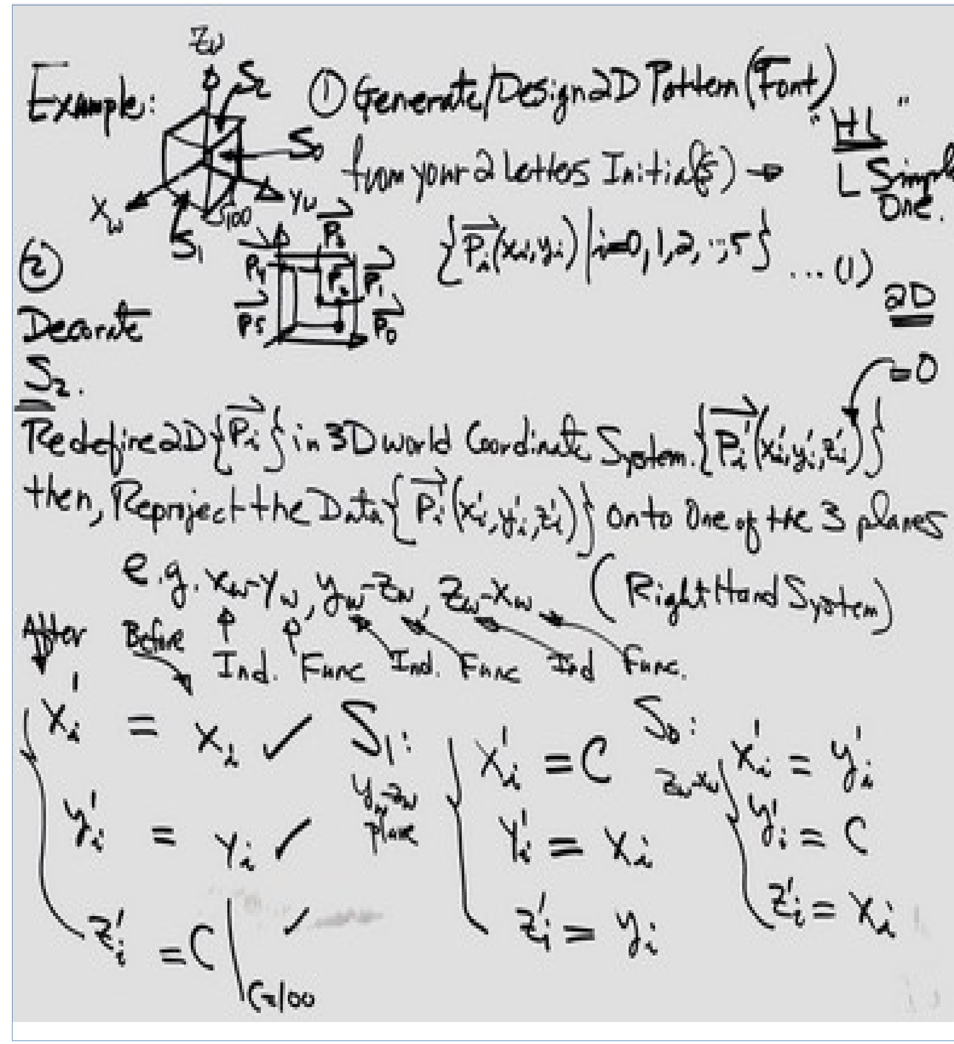
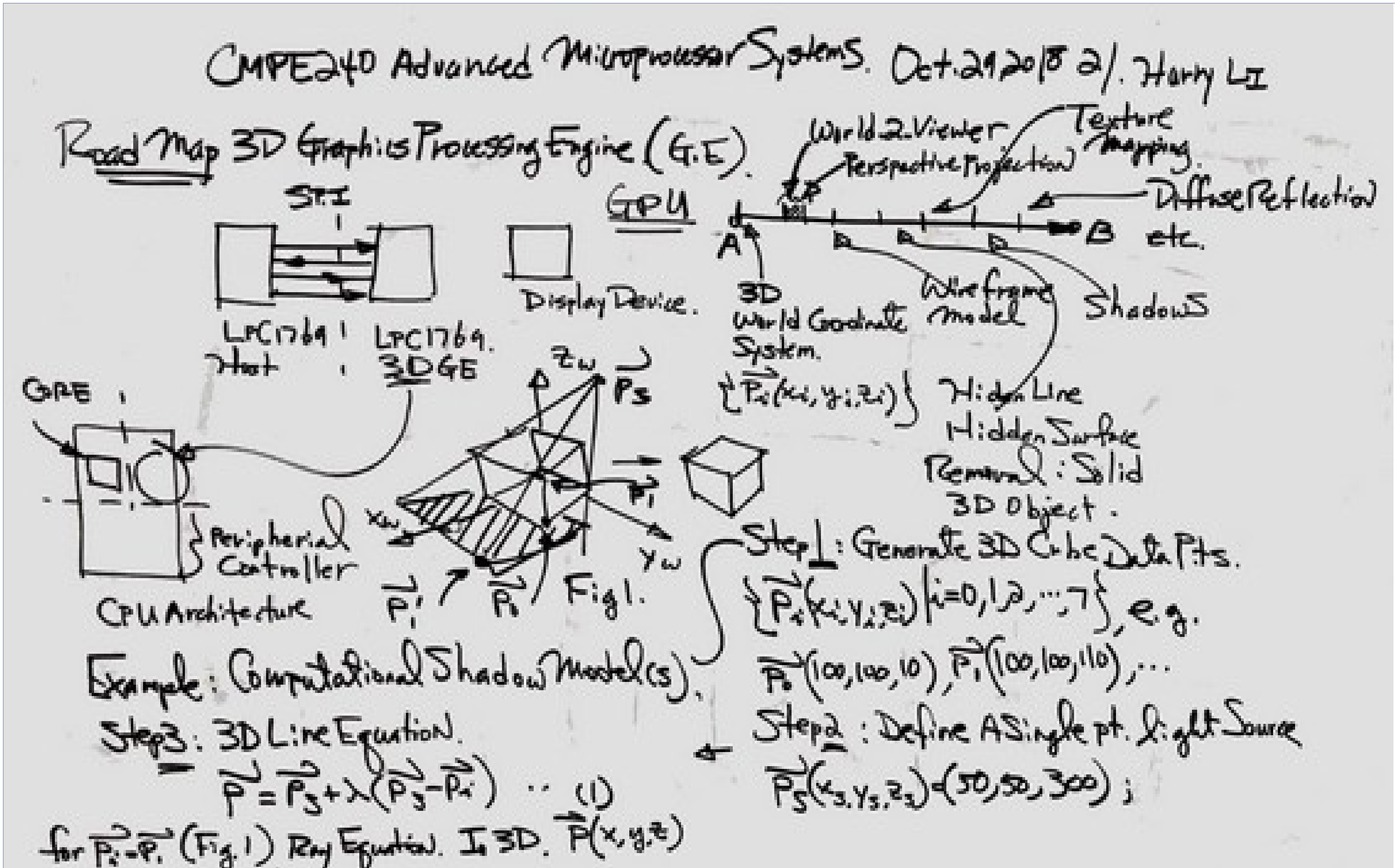
10-29-2018 3D Graphics Engine: Shade Computation





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



CMPE240 Advanced Microprocessor Systems. Oct. 29 2018 2/. Harry Li

Road Map 3D Graphics Processing Engine (G.E).

SPI

GPU

World 2 Viewer Perspective Projection

Texture mapping

Diffuse Reflection etc.

A

B

Display Device.

LPC1769 Host

LPC1769 3DGE

GPE

Peripheral Controller

CPU Architecture

3D World Coordinate System.

$\{\vec{P_i}(x_i, y_i, z_i)\}$

Wireframe model

Shadows

Hidden Line Hidden Surface Removal: Solid 3D Object.

z_w, $\vec{P_s}$, $\vec{P_1}$, x_w, y_w, $\vec{P_i}$, $\vec{P_0}$

Fig 1.

Step 1: Generate 3D Cube Data Pts.

$\{\vec{P_i}(x_i, y_i, z_i) \mid i = 0, 1, 2, \ldots, 7\}$, e.g.

$\vec{P_0}(100, 100, 10)$, $\vec{P_1}(100, 100, 110)$, ...

Example: Computational Shadow Model(s).

Step 2: Define A Single pt. light Source

$\vec{P_s}(x_s, y_s, z_s) = (50, 50, 300)$;

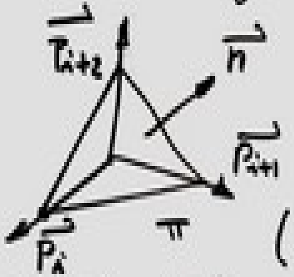
Step 3: 3D Line Equation.

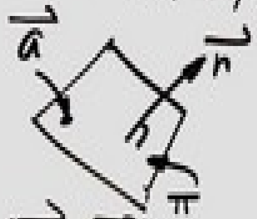
$$\vec{P} = \vec{P_s} + \lambda(\vec{P_s} - \vec{P_i}) \quad \cdots (1)$$

for $\vec{P_i} = \vec{P_1}$ (Fig. 1) Ray Equation. In 3D, $\vec{P}(x, y, z)$

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

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





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

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

$$\vec{n} \cdot (\vec{a} - \vec{v}) = 0 \quad \cdots (3)$$ where $\vec{a} = (a_x, a_y, a_z)$

Any Arbitrary Known pt. On the plane.

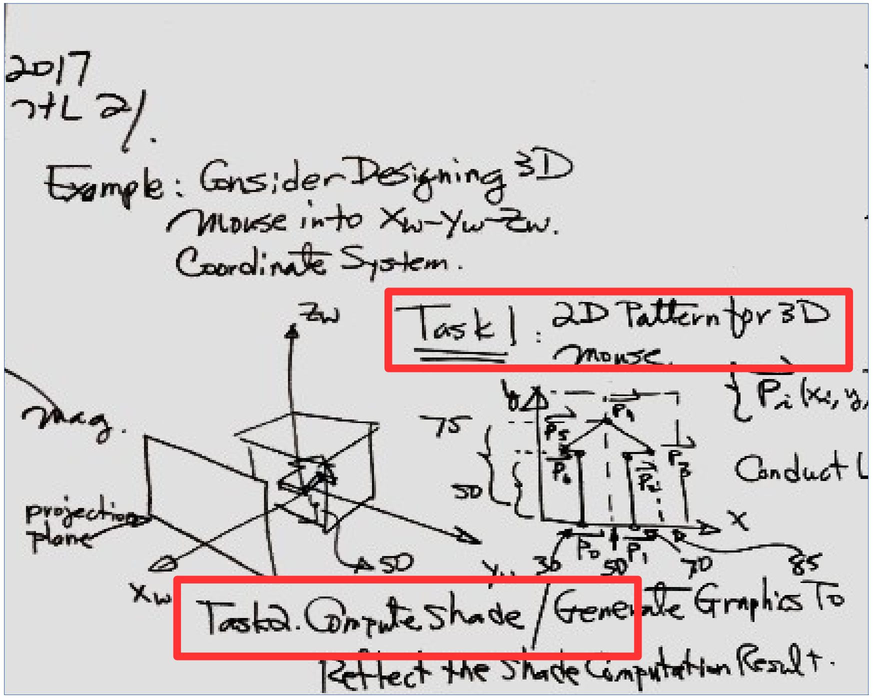
Step 5. Intersect pt.

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

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

Harry Li, Ph.D

2017 Design of 3D Virtual Display



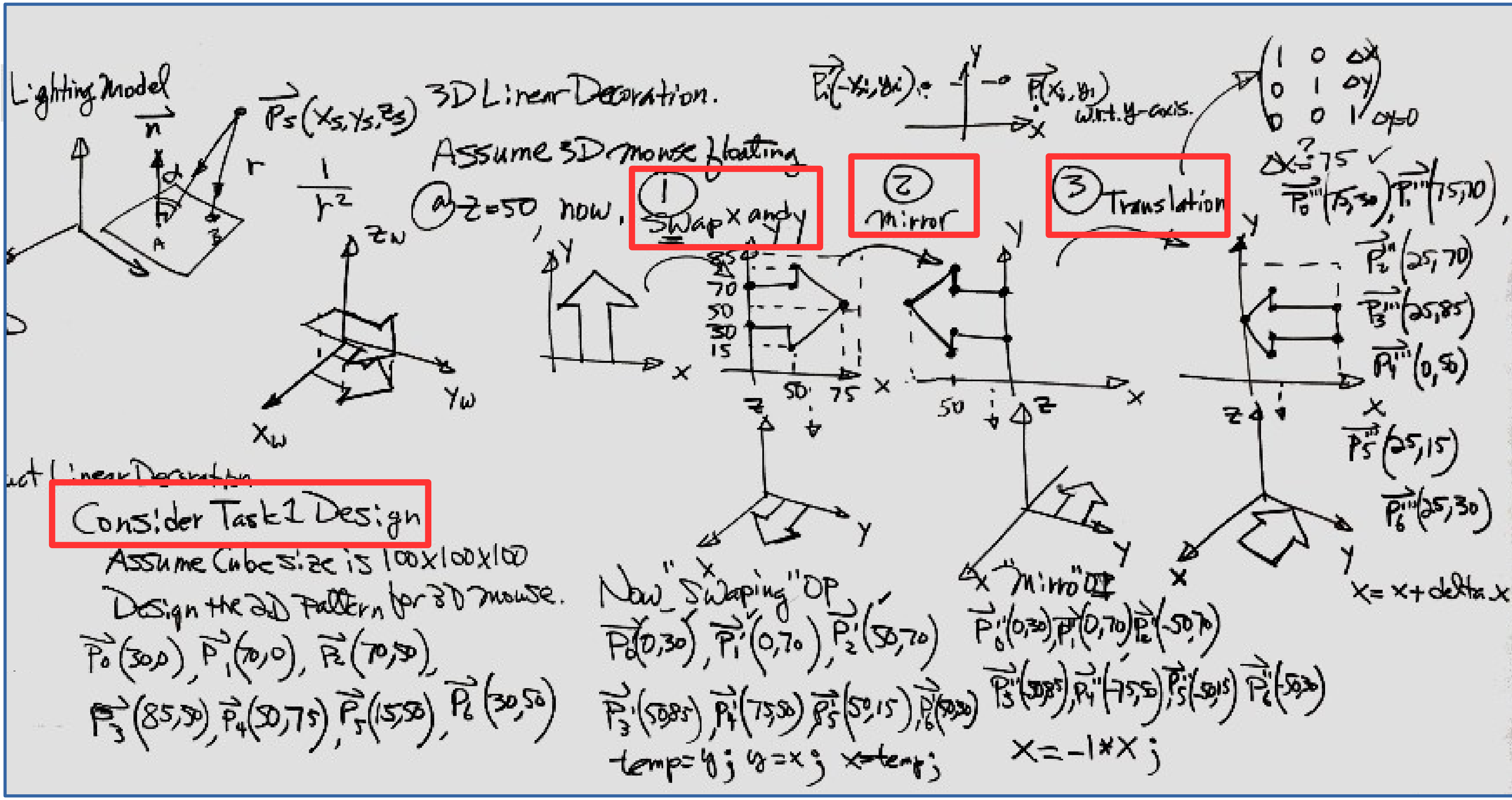
Three tasks:

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



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

$$\left\{\vec{P_i}(x_i, y_i, z_i) \;\middle|\; \begin{array}{l} i=0,1,2,\cdots,6 \\ z_i=50 \end{array}\right\}$$

$\vec{P_0}(75,30,50)$

$\vec{P_1}(75,70,50)$, $\vec{P_2}(25,70,50)$, $\vec{P_3}=(25,85,50)$

$\vec{P_4}(0,50,50)$, $\vec{P_5}(25,15,50)$, $\vec{P_6}(25,30,50)$

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)

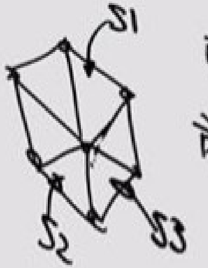
$S2: \{\vec{P_i'}(x_i, y_i, z_i-5) \mid i=0,1,2,\cdots,6\}$

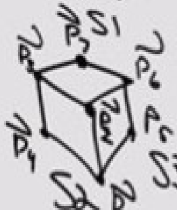
"Wireframe" → Solid Object

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)



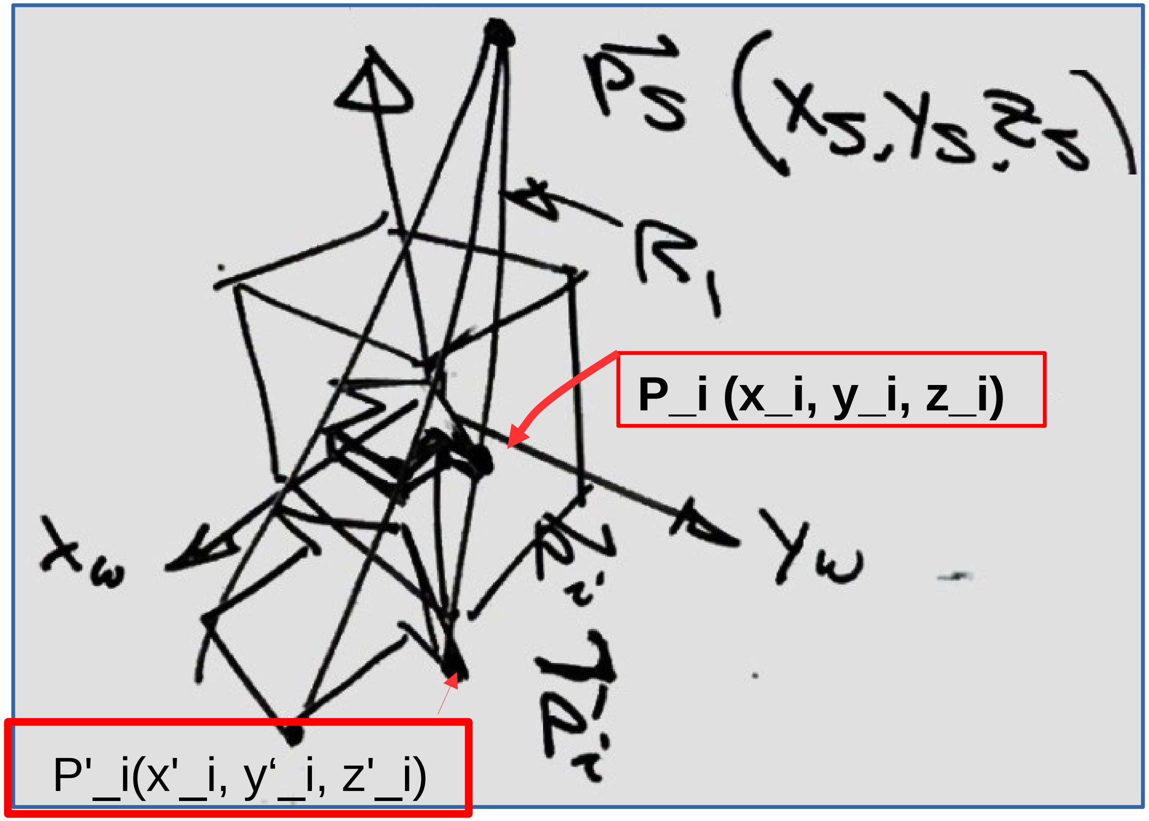


$S_1: \vec{P_2}-\vec{P_6}-\vec{P_7}-\vec{P_3}$ C.C.

$S_2: \vec{P_1}-\vec{P_2}-\vec{P_3}-\vec{P_4}$. C.C

$S_3: \vec{P_1}-\vec{P_5}-\vec{P_6}-\vec{P_2}$ 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}

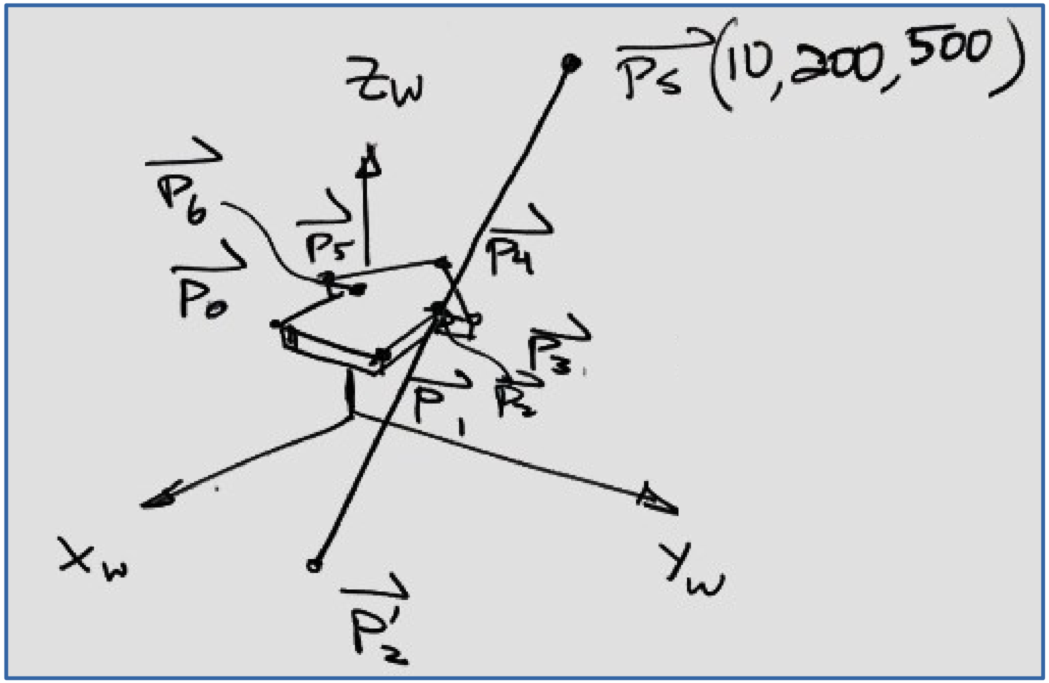
e.g.,

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)

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

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

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

And the known point vertex a is:

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

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,

$$\vec{n} * \left(\vec{P_S} + \lambda(\vec{P_S} - \vec{P_i})\right) = 0 \qquad (5)$$

Or

$$\vec{n} * \vec{P_S} + \lambda \vec{n} * (\vec{P_S} - \vec{P_i}) = 0 \qquad (5.1)$$

Solve for lamda,

$$\lambda = - \frac{\vec{n} * \vec{P_S}}{\vec{n} * (\vec{P_S} - \vec{P_i})} \qquad (6)$$

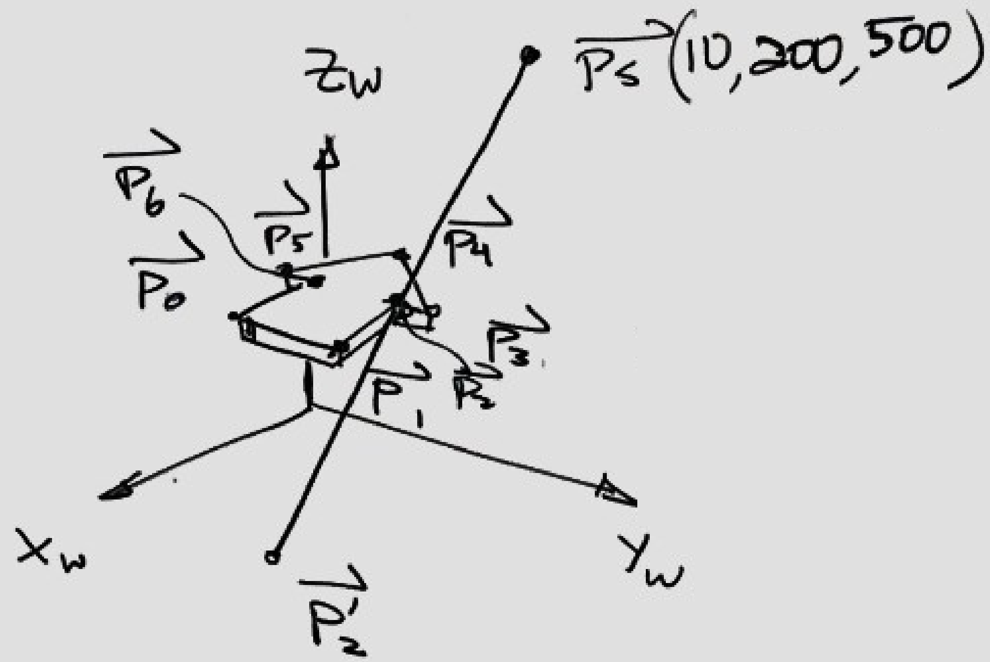
e.g.

$$\lambda = \frac{-(n_x, n_y, n_z) * (x_S, y_S, z_S)}{(n_x, n_y, n_z) * (x_S - x_i, y_S - y_i, z_S - z_i)} \qquad (6.1)$$

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