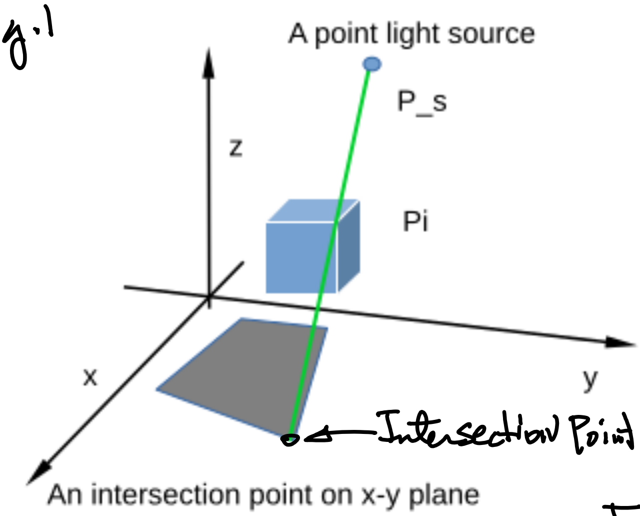
Nov.15 (Monday).

Example: 3D Shadow Computation.

From Eqn(2) pp.50.

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

Fig.1



Intersection Point. $\vec{P_i'}$

Ray Equation: pp.46.

$$\vec{R}=\vec{P_i}+\lambda(\vec{P_s}-\vec{P_i}) \quad \cdots (3)$$

The Intersection Point on X_w-Y_w plane, $\vec{P_i'}$ is a common pt shared by Ray Eqn (3) and plane eqn(2).

$$\begin{cases} \vec{R}=\vec{P_i}+\lambda(\vec{P_s}-\vec{P_i}) & \cdots (4\text{-}1) \\ \vec{n}\cdot(\vec{v}-\vec{a})=0 & \cdots (4\text{-}2) \end{cases}$$

From Eqn(4-2),

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

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

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

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

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

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

Note: λ together with the Ray Equation(3), will give the intersection point.

Example: Given a Single point light Source $\vec{P_s}(-20,110,200)$, $\vec{P_i}(100,100,110)$ Find intersection Point to plot Shadow.

$\underline{\underline{\text{Sol}}}$ From Eqn(5), find λ.

Since

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

$\vec{P_i}=(100,100,110)$

Hence,

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

$$=\frac{(n_x,n_y,n_z)(a_x,a_y,a_z)-(n_x,n_y,n_z)\cdot(x_i,y_i,z_i)}{(n_x,n_y,n_z)(x_s-x_i,y_s-y_i,z_s-z_i)}$$

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

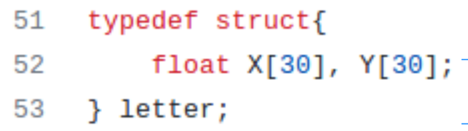
Note: Define Letter(s) for my initial for Linear Decoration. Latter with Furth Discussion.

$n_x = 0, n_y = 0, n_z = 1.$

Therefore, the above equation Becomes

$$= \frac{0 + 0 + 0 \quad - (0 + 0 + 1 \cdot z_i)}{0 + 0 + 1 \cdot (z_s - z_i)}$$

$$= -\frac{z_i}{z_s - z_i}$$ from the given condition



```c
typedef struct{
    float X[30], Y[30];
} letter;
```

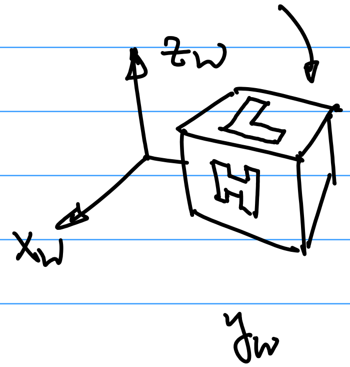
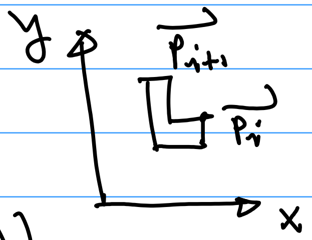


Fig. 2.

$z_i = 110, z_s = 200,$

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

Define 2D Patterns



2D Pattern to Decorate 3D plane.

Substitute λ into the Ray Equation (3)

$$\vec{R} = \vec{P}_i + \lambda(\vec{P}_s - \vec{P}_i)$$

$$= (100, 100, 110) + \lambda\left((-20, 110, 200) - (100, 100, 110)\right)$$

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

$$= \left(100 + \frac{11}{9} \times 130, 100 - \frac{11}{9} \times 10, 110 - \frac{11}{9} \times 90\right)$$

$$= \left(100 + \frac{11 \times 130}{9}, 100 - \frac{11 \times 10}{9}, 110 - 110\right)$$

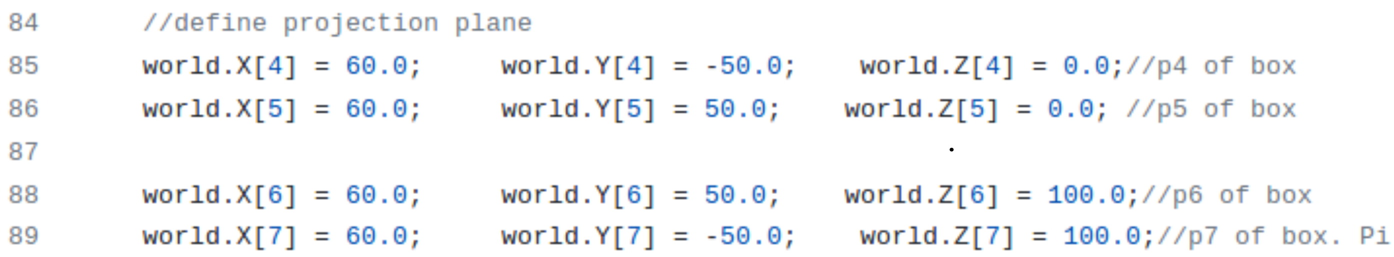
$$= \left(100 + \frac{11 \times 130}{9}, 100 - \frac{11 \times 10}{9}, \underline{\underline{0}}\right)$$ On $X_w - Y_w$ plane, so z must Be 0!

C/C++ Implementation.

2018F-116-11diffuse20181114.cpp

2018F-117-12dda.cpp

CmpE240 (II) SJ



```
    //define projection plane
    world.X[4] = 60.0;     world.Y[4] = -50.0;    world.Z[4] = 0.0;//p4 of box
    world.X[5] = 60.0;     world.Y[5] = 50.0;    world.Z[5] = 0.0; //p5 of box

    world.X[6] = 60.0;     world.Y[6] = 50.0;    world.Z[6] = 100.0;//p6 of box
    world.X[7] = 60.0;     world.Y[7] = -50.0;    world.Z[7] = 100.0;//p7 of box. Pi
```

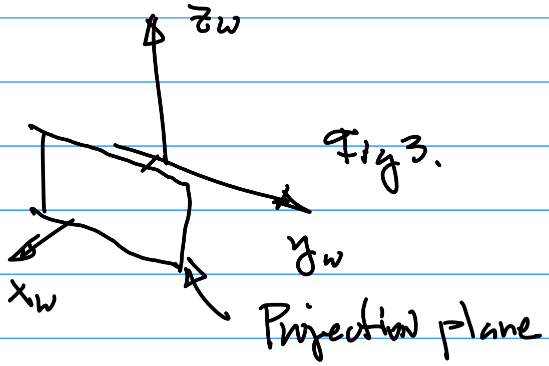
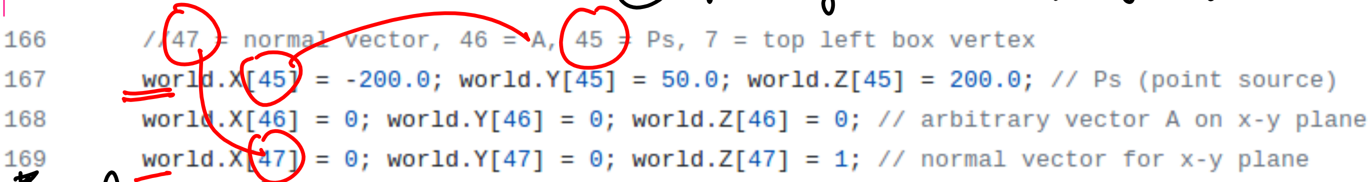


Fig 3.

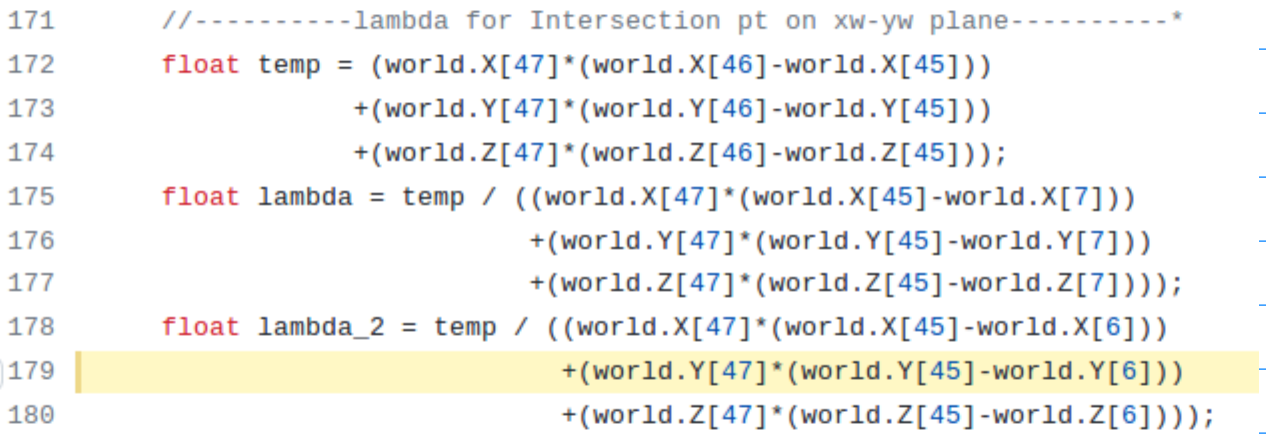
② Point Light Source $\overrightarrow{P_s}(x_s, y_s, z_s)$



```
    //47 = normal vector, 46 = A, 45 = Ps, 7 = top left box vertex
    world.X[45] = -200.0; world.Y[45] = 50.0; world.Z[45] = 200.0; // Ps (point source)
    world.X[46] = 0; world.Y[46] = 0; world.Z[46] = 0; // arbitrary vector A on x-y plane
    world.X[47] = 0; world.Y[47] = 0; world.Z[47] = 1; // normal vector for x-y plane
```

① normal vector $\vec{n}$ (0,0,1);

λ Computation is implemented Below,



```
    //----------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])));
```

$$\lambda = \frac{n_x \cdot a_x + n_y \cdot a_y + n_z \cdot a_z - (n_x x_i + n_y y_i + n_z z_i)}{n_x(x_s - x_i) + n_y(y_s - y_i) + n_z \cdot (z_s - z_i)} \quad \text{OR}$$

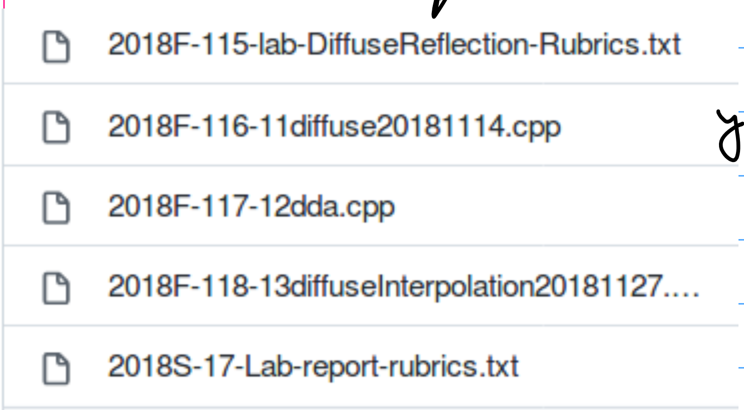
$$= \frac{n_x (a_x - x_i) + n_y (a_y - y_i) + n_z (a_z - z_i)}{n_x (x_s - x_i) + n\,(y_s - y_i) + n_z \cdot (z_s - z_i)}$$

CmpE240 (II)

Nov.17 (Wed)

3D G.E. Design On Diffuse Reflection

Ref: from class github.



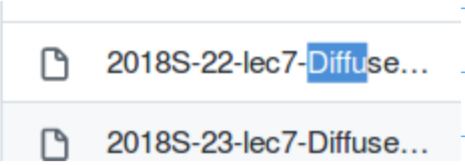
2018F-115-lab-DiffuseReflection-Rubrics.txt

2018F-116-11diffuse20181114.cpp

2018F-117-12dda.cpp

2018F-118-13diffuseInterpolation20181127....

2018S-17-Lab-report-rubrics.txt



2018S-22-lec7-Diffuse...

2018S-23-lec7-Diffuse...

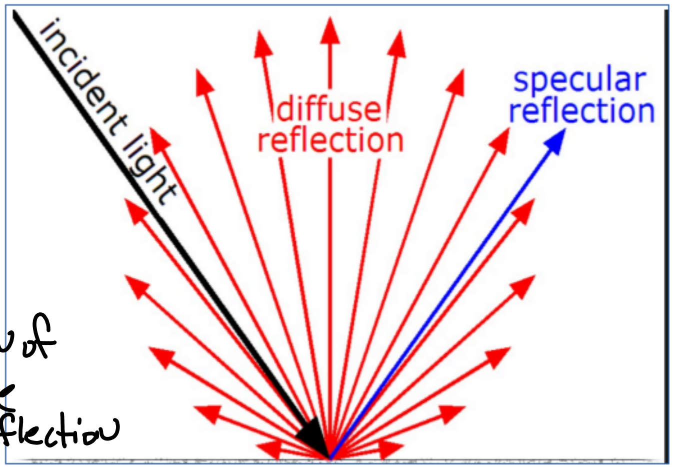


Illustration of Diffuse Reflection

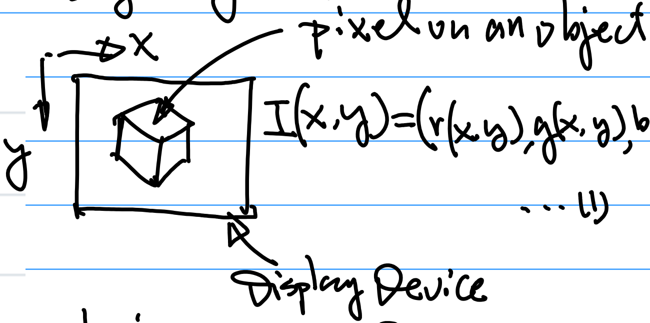
① https://en.wikipedia.org/wiki/Diffuse_reflection

Chapter 12 ② My Book in-progress

Lighting Models with Emphasis on Diffuse Reflection

In the lighting model formulation, very often you will see 3 different type of lighting models as shown in the following Figure:

Example: Generate Realistic Looking Graphics, Simulate/Formulate Lighting model.



$I(x,y)=(r(x,y),g(x,y),b(x,y))$ ... (1)

Color image defined By 3 Primitive colors, r—red, g—green, b—blue

Each primitive color is represented as 8 bit value.

$r(x,y) \in [0,255]$, $g(x,y) \in [0,255]$, and $b(x,y) \in [0,255]$

3 Type of Light Contributors to generate the color

$$I(x,y)=\underset{\text{Diffuse Reflection}}{I_1(x,y)}+\underset{\text{Specular Reflection}}{I_2(x,y)}+\underset{\text{Ambient Light}}{I_3(x,y)} \quad \dots (2)$$

Note: Specular Reflection is the reflection which generates high light, it is a function of "Eye" e.g. Virtual Camera location.

Ambient Light (Reflection), are those Coming from indirect light Source(s).

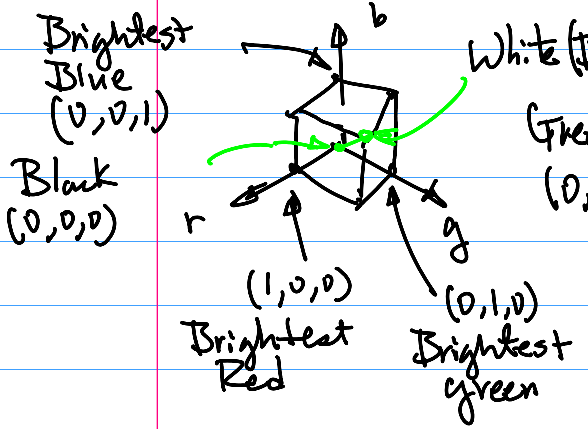
Example: color Intensity generated by indirect Light when viewing the object(s), for example, underneath a table.

Math Description: A constant.

Diffuse Reflection.

A Reflection Reflects in-coming light uniformally in all different directions

3D Vector Color Space



Green Line from (0,0,0) to (1,1,1)

Color is grey. Intensity Increases.

About color of An Object: Characteristic of the Object. It depends on reflectivity of the object itself.

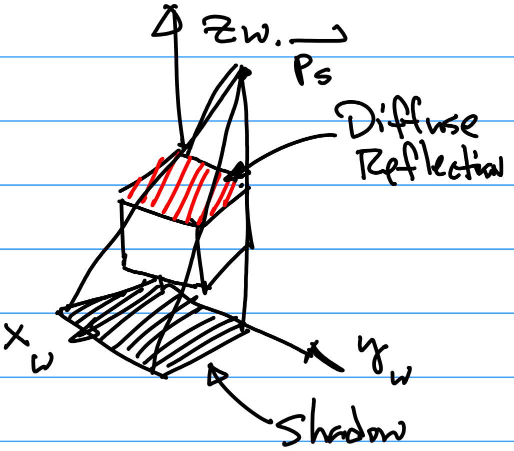
Nov. 22nd (monday)

Note: Last Project Diffuse Reflection + Decoration Algorithm → 3D Graphics Processing Engine

2021F-109b-project-DiffuseReflection-Rubrics-…

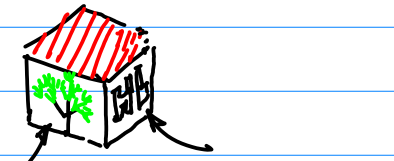
1. Count 20 pts., Due Dec 8th (Wed) 11:59 pm. (No Late submission) Submission on CANVAS.

2. (1) Solid Cube



(2) Diffuse Reflection on the top Surface, with 1 primitive color, Red

(3) Tree On One Surface



Self Design

2-Letter Font On the Surface

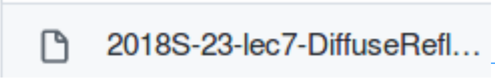
Your Initial

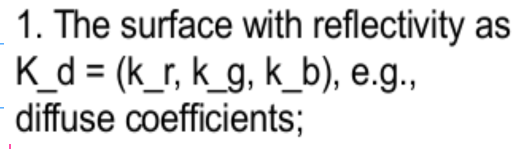
3. please follow the requirements Posted On-Line

(1) IEEE Style Report 5 pages.

(2) Exported Project; (3) photo of the Implementation; (4) 5 Second Video Clips

Diffuse Reflection Formulation.


2018S-23-lec7-DiffuseRefl...


1. The surface with reflectivity as K_d = (k_r, k_g, k_b), e.g., diffuse coefficients;

K coefficient of Reflectivity.

K_d : "d" for diffuse Reflection.

$\vec{K}_d(K_r, K_g, K_b)$

Reflectivity for Red

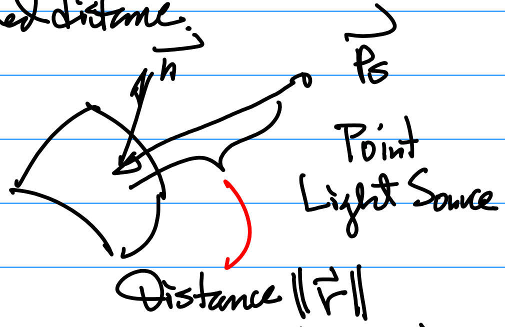
~ for green

~ for Blue

$0 \le K_r \le 1$, $0 \le K_g \le 1$, $0 \le K_b \le 1$ ... (1)

If $K_r = K_g = 0$, $K_b = 1$, then we have "blue" color (Highest — Full Reflection of Blue).

2. Light (color) Intensity from $\vec{P}_s$ to a surface is formulated as a decay function to the Squared distance.

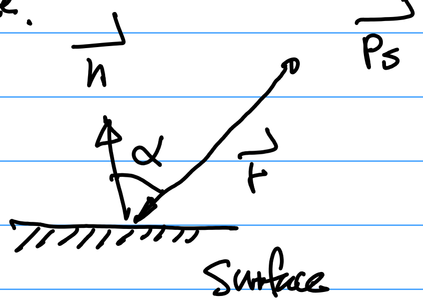


$\vec{r}$ Ray Equation from $\vec{P}_s$ to $\vec{P}_i$ on the surface.

Intensity (color)

$$\frac{1}{\left(\sqrt{x_r^2+y_r^2+z_r^2}\right)^2} = \frac{1}{\|\vec{r}\|^2}$$

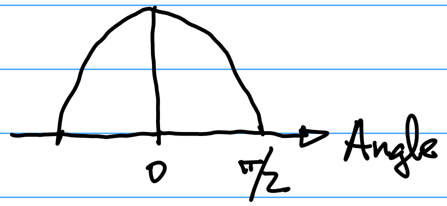
3. Intensity (Color) is a function of the angle of the incoming point light Source.



$$I_{diff} = (\text{Reflectivity})\left(\frac{1}{\|\vec{r}\|_2^2}\right)\left(\text{Angle of the incoming light}\right) \quad \ldots (2)$$

at a particular Point on a Surface

Cos (Angle of the incoming Light) :



Mathematically :

$$\vec{r}\cdot\vec{n} = \|\vec{r}\|\|\vec{n}\| \cos\alpha \quad \cdots (3)$$

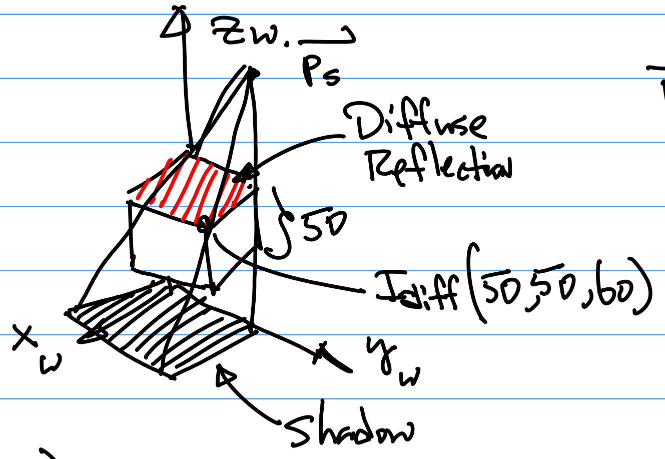
$$\cos\alpha = \frac{\vec{r}\cdot\vec{n}}{\|\vec{r}\|\|\vec{n}\|} \quad \ldots (3\text{-}b)$$

Hence:

$$\boxed{\vec{I}_{diff} = K_d \cdot \frac{1}{\|\vec{r}\|^2} \cdot \cos\alpha = K_d \cdot \frac{1}{\|\vec{r}\|^2} \cdot \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|}}$$

... (4)

Example: Given Conditions



$\vec{P}_s(40,60,120)$. Cube is floating by $z=10$.

Find $\vec{I}_{diff}(50,50,60) = ?$

Sol: First, $K_d = (K_r, K_g, K_b) = (0.8,0,0)$

Then, the distance from $\vec{P}_s$ to $\vec{P}_i(50,50,60)$

$$\|\vec{r}\|^2 = (x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2$$
$$= (40-50)^2 + (60-50)^2 + (120-60)^2$$
$$= 10^2 + 10^2 + 60^2$$

$$\cos\alpha = \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|}.$$

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

$(50,50,60)-(40,60,120)$
$= (10,-10,-60)$

$$\vec{r} = \vec{P}_i + \underline{\underline{\lambda}}(\vec{P}_s - \vec{P}_i)$$

! for this calculation

$$= (50,50,60) + \lambda\left((40,60,120) - (50,50,60)\right)$$
$$= (50,50,60) + \lambda(-10,10,60)$$

Let $\lambda = 1$. $\vec{r}\big|_{\lambda=1} = (50,50,60) + (-10,10,60)$

$$= (40,60,120)$$

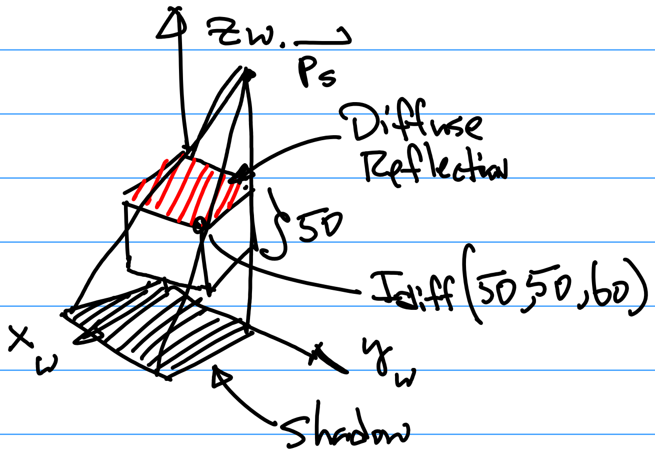
$$\vec{n}\cdot\vec{r} = (0,0,1)\cdot(40,60,120) = \cancel{0\times40} + \cancel{0\times60} + 1\times120 = 120$$

$$\frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} = \frac{120}{1\cdot\sqrt{40^2+60^2+120^2}} = \frac{120}{\sqrt{40^2+60^2+120^2}}$$

Therefore

$$\vec{I}_{diff}(50,50,60) = (0.8,0,0)\,\frac{1}{10^2+10^2+60^2}\cdot\frac{120}{\sqrt{40^2+60^2+120^2}}$$

red: $I_{diff,r} = \dfrac{0.8\times120}{(10^2+10^2+60^2)\times(\sqrt{40^2+60^2+120^2})}$

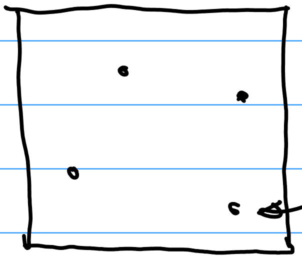


Note: The Computation is carried out in X_w-Y_w-Z_w.

You can Compute diffuse reflection for the rest 3 vertices

$\vec{P}(50,0,60)$, $\vec{P}(0,50,60)$, $\vec{P}(0,0,60)$

Note: To plot these 4 point with the color, use Transformation Pipeline, !!!

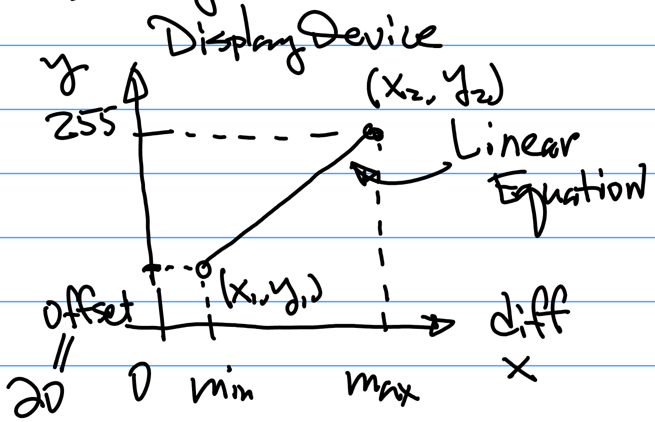


$\vec{P_i}''$ its diffuse reflection is Computed in X_w-Y_w-Z_w.

Their Locations are Computed by Transformation PipeLine !!!

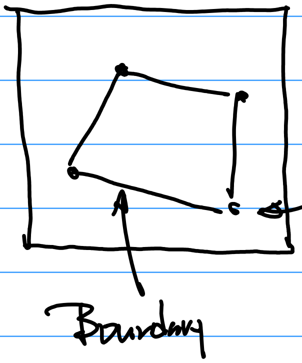
Important! To make the Diff. Reflection Result Visible, use

Scaling as follows



$\frac{x-x_2}{x_2-x_1} = \frac{y-y_2}{y_2-y_1}$ → find the Eqn.

Example: Compute Boundary Color



$\vec{P_i}''$ its diffuse reflection is Computed in X_w-Y_w-Z_w.

Two Steps Process.

After Perspective Projection, we have to deal with Finite Resolution of the display device.

D.D.A

(Digital Differential Algorithm)