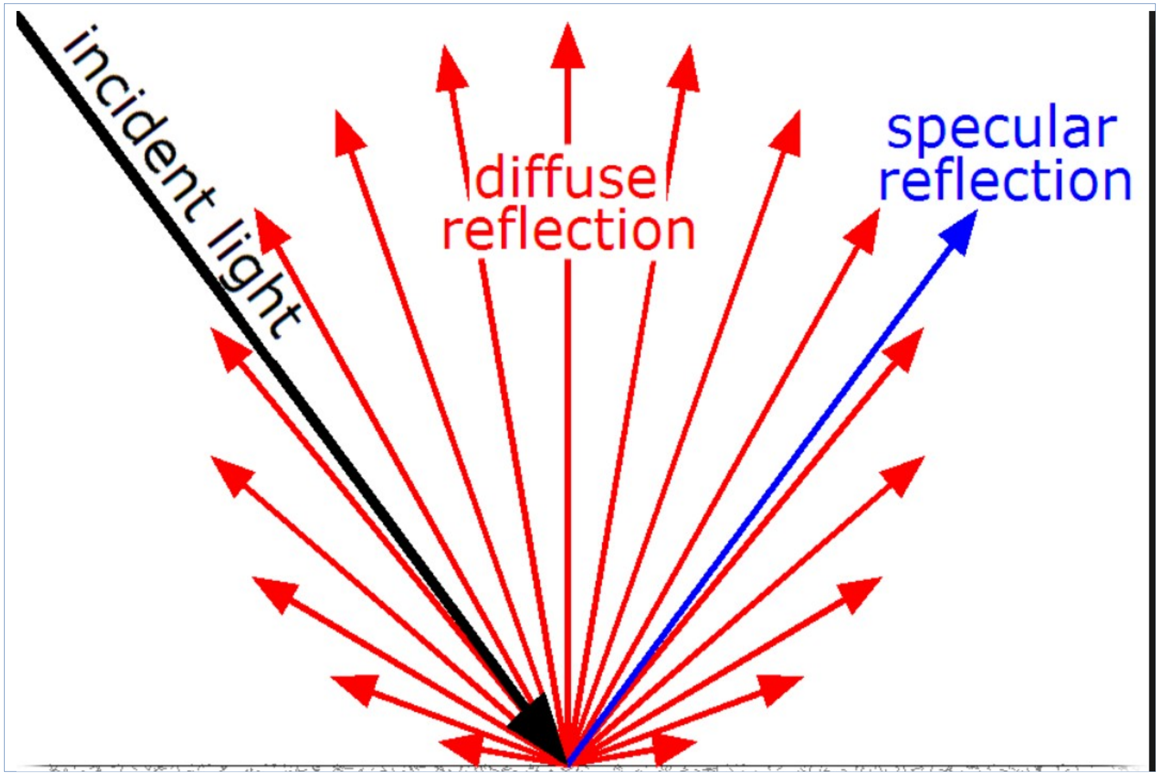
Diffuse Reflection



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

Diffuse Reflection: the reflection of light uniformly in all different directions, the surface of this reflection exhibits Lambert reflection, e.g., equal luminance when viewed from all directions.

Two Key Characteristics:

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

2. The decay of incident light is inverse proportional to its distance from the source to the surface point. e.g., 1/(r*r), where r is bing the distance from the light source to the surface.

Specular vs. diffuse reflection

Harry Li, Ph.D

Diffuse Reflection Formulation

$$\vec{I}(x,y,z) = \left(I_r(x,y,z), I_g(x,y,z), I_b(x,y,z)\right) \quad \ldots (1)$$

Object I(x,y,z) consists of r, g, b 3 primitive colors, as denoted in (1).

Light source I_s(x,y) consists of r, g, b 3 primitive colors as follows, but let's simplify it as white color, so r, g, b all equal and have the highest value (if in graphics, they are 255)

$$\vec{I_s}(x,y,z) = \left(I_r(x,y,z), I_g(x,y,z), I_b(x,y,z)\right) \quad \ldots (2)$$

Object surface consists of reflectivity, e.g., coefficient of reflection

$$\vec{K_d} = (K_r, K_g, K_b) \quad \ldots (3)$$

r_d vector in Equation (1) is a ray equation, just like I_s(x,y,z) but has no r, g, b primitive color defined in it for the matter of simplicity.

Diffuse Reflection Equation

$$\vec{I}(x,y,z) = \left(I_r(x,y,z), I_g(x,y,z), I_b(x,y,z)\right)$$

Each primitive olor of the object I(x,y,z) can be written as

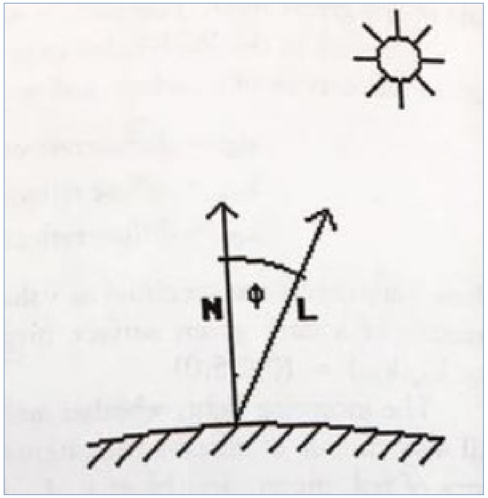
$$I_r = K_{d_r} \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \frac{1}{\|\vec{r}\|_2^2} \quad \ldots (1.1)$$

wher
e

$$\|\vec{r}\|_2^2 = x_r^2 + y_r^2 + z_r^2$$

$$I_g = K_{d_g} \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \cdot \frac{1}{\|\vec{r}\|_2^2} \quad \ldots (1.2)$$

$$I_b = K_{d_b} \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \cdot \frac{1}{\|\vec{r}\|_2^2} \quad \ldots (1.3)$$



Reference: Computer Graphics, C. K. Pokorny, C. F. Gerald, pp. 514

11-8-2018 Diffuse Reflection Equation

CMPE163 C.G. & A.R. Nov 8, 2018

1) Homework: Bring LSM303 Sensor w/ Pie Board to the class for Show & Tell;

Transformation

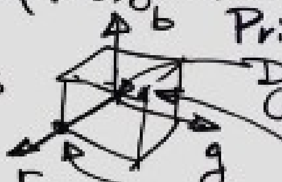
Diffuse Reflection: Background. 3D model, wireframe

0 ~ 1.0

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

Primitive Colour Texture Shadow

Pixel Graphics



Dark Black (0,0,0)

(1,1,1) Brightest White

Colour Space

Brightest Red

I(x,y)

pixel Graphics VS Vector Graphics

$\vec{I}(x,y) = \vec{I}_{dif}(x,y) + \vec{I}_{spe}(x,y) + \vec{I}_{amb}(x,y)$ ...(2).

$\vec{I}_{dif}$ Uniformly Distributed in All Directions (Not Function of "Eye"). $\vec{I}_{spe}$ function of "The Eye" Produces High Lights.

$\vec{I}_{ambient}$ generated from Indirect ("offset") Light Source.

Consider Diffuse Reflection.

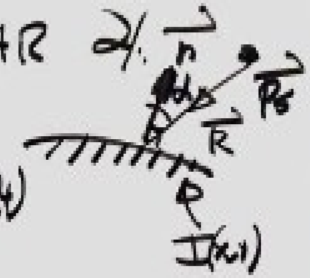
$K_d = (k_r, k_g, k_b)$ ...(3) Normalized [0, 1.0]

Reflectivity $k_r \simeq k_g \simeq k_b \simeq 1$ full Reflection

VIS Blue Red

450 nm 700 nm

CMPE163 CG&AR 2/.

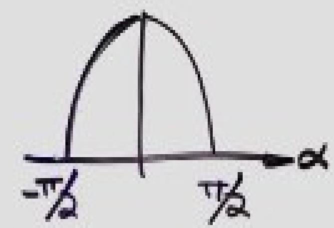


$\vec{n} \cdot \vec{r} = \|\vec{n}\| \|\vec{r}\| \cos\alpha$ ...(4)

(@ pixel location)

From $\vec{P_s}$ to the pixel location

$\cos\alpha = \dfrac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|}$ ...(4*)



Property 1: ~ Max Intensity when $\alpha = 0$ ($\vec{P_s}$ light Ray w/ $\vec{n}$)
~ Intensity = 0, when $\alpha = \pm\pi/2$

Intensity Decay follows $\cos\alpha$ function.

Property 2: Intensity Decay follows $\frac{1}{\|r\|^2}$ (Squared distance from the pt. light Source to that pixel location.

Intensity

Property 3: Colour & Intensity Combined is Computed Base Reflectivity $\vec{K_d}$ and Property 1 & 2 (@ this pixel location)

Colour

Harry Li, Ph.D

11-8-2018 Diffuse Reflection Calculation

CMPE163 CG & AR. Nov 8, 2018

3/

From Eqn (1.1) (on github) Example: PP.4.

$$I_r = K_{dr}\frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|}\frac{1}{\|r\|^2} \quad \cdots (1.1)$$

$\vec{P_s}(x_s, y_s, z_s) = (10, 200, 500)$, $\vec{P_i}(x_i, y_i, z_i)$, $i=2$.

$\vec{n}(0,0,1)$, $\vec{r} = \vec{P_s} + \lambda(\vec{P_s} - \vec{P_i}) =$

$(x_s, y_s, z_s) + \lambda(x_s - x_i, y_s - y_i, z_s - z_i)\Big|_{\lambda=-1} = \vec{P_i}$

Assume $K_{d_r} = 0.85$, therefore $[0, 255]$

$$I_r = 0.85 \cdot \frac{(0,0,1)(x_i, y_i, z_i)}{\|(0,0,1)\|\,\|(x_i,y_i,z_i)\|}\cdot\frac{1}{[(x_s-x_i)^2+(y_s-y_i)^2+(z_s-z_i)^2]}$$

$$= 0.85 \cdot \frac{z_i}{\sqrt{x_i^2+y_i^2+z_i^2}}\cdot\frac{1}{\sqrt{(x_s-x_i)^2+(y_s-y_i)^2+(z_s-z_i)^2}}$$

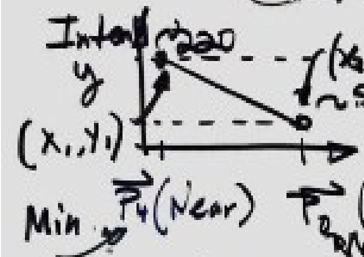
Sqrt

for Red colour

C/C++

```
#define K_d_red 0.85
I_red = K_d_red * z[i];
float tmp = sqrt(x[i]*x[i] + y[i]*y[i] + z[i]*z[i]);
float tmp2 = sqrt((xs - x[i])*(xs - x[i]) +
     (ys - y[i])*(ys - y[i]) + (zs - z[i])*(zs - z[i]));
I_red = I_red / (tmp * tmp2);
```

CMPE163 CG & AR 4/



Linear Interpolation.

$$\frac{y_2-y}{y_2-y_1} = \frac{x_2-x}{x_2-x_1} \quad \cdots (6)$$

Min $\vec{P_4}$ (Near) $\vec{P_2}$ (far) Max

$$\frac{y_2-y}{x_2-x} = \frac{y_2-y_1}{x_2-x_1}, \quad y_2 - y = \frac{y_2-y_1}{x_2-x_1}(x_2-x)$$

$$-y = -y_2 + \frac{y_2-y_1}{x_2-x_1}(x_2-x) = -y_2 - \frac{y_2-y_1}{x_2-x_1}x + \frac{y_2-y_1}{x_2-x_1}x_2$$

$$y = y_2 + \frac{y_2-y_1}{x_2-x_1}x - x_2\frac{y_2-y_1}{x_2-x_1}$$

$$y = \frac{y_2-y_1}{x_2-x_1}x + \left(y_2 - x_2\frac{y_2-y_1}{x_2-x_1}\right) \quad \cdots (6*)$$

Suppose:

$$\frac{1}{\sqrt{x_i^2+y_i^2+z_i^2}\cdot\sqrt{(x_s-x_i)^2+(y_s-y_i)^2+(z_s-z_i)^2}} = \Delta_i \quad \cdots (7)$$

find $\min_{i\in[0,1,\cdots,6]}\{\Delta\}$, $\max_{i\in[0,1,\cdots,6]}\{\Delta\}$ $\cdots (7*)$

Hence, 2 pts can be found, in Eqn (6*)

Indep. Variable is Δ_i, function (y, Intensity) is intensity.

$$(x_2, y_2) = (\min \Delta_i, 50)$$
$$(x_1, y_1) = (\max \Delta_i, 220) \quad \cdots (8)$$

Harry Li, Ph.D

11-8-2018 Diffuse Reflection Calculation (2)

CMPE163 C.G. & AR. Nov 8, 2018 5/.

From Eqn (8), we have $\vec{P_4}$ (Shortest Distance pt to $\vec{P_s}$) → $Max\,\Delta_i = Max\,\Delta_4$.

$\vec{P_0}$ (Farest Point from $\vec{P_s}$) → $Min\,\Delta_0$

Hence Eqn (6*) is well defined.

Now, For the rest of the pts, $\vec{P_j}(x_j, y_j, z_j)$

From Diffuse Reflection Eqn (1.1), we have

$$\Delta_j = \frac{1}{\sqrt{x_j^2+y_j^2+z_j^2}\,\sqrt{(x_s-x_j)^2+(y_s-y_j)^2+(z_s-z_j)^2}}$$

then. Let $\Delta_j = x$ in Eqn (6),

find the Intensity with this x value.

Example:
$$y = I_{d\_red} = \frac{50-220}{Min\Delta_0 - Max\Delta_4}\Delta_j + \left(50 - Min\Delta_0 \cdot \frac{50-220}{Min\Delta_0 - Max\Delta_4}\right) \quad ..(9)$$

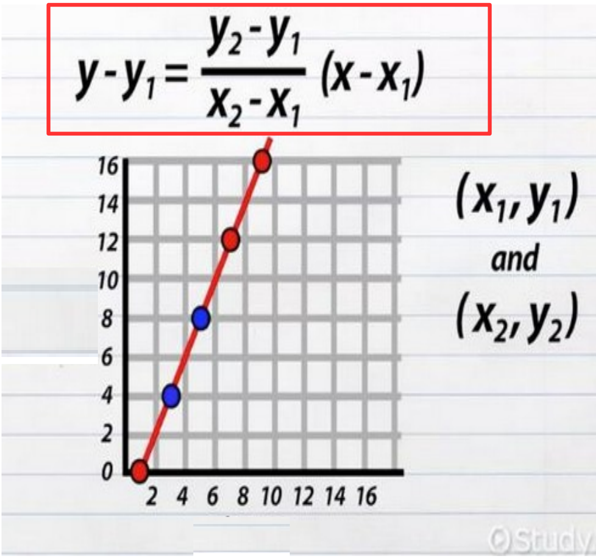
$\vec{P_s}(10,200,50)$

$\vec{P_4}(20,200,200) \Rightarrow I_{d\_red} = 220$

$\vec{P_0}(100,100,200) \Rightarrow I_{d\_red} = 50$ $\quad Min\Delta_0 = ?$

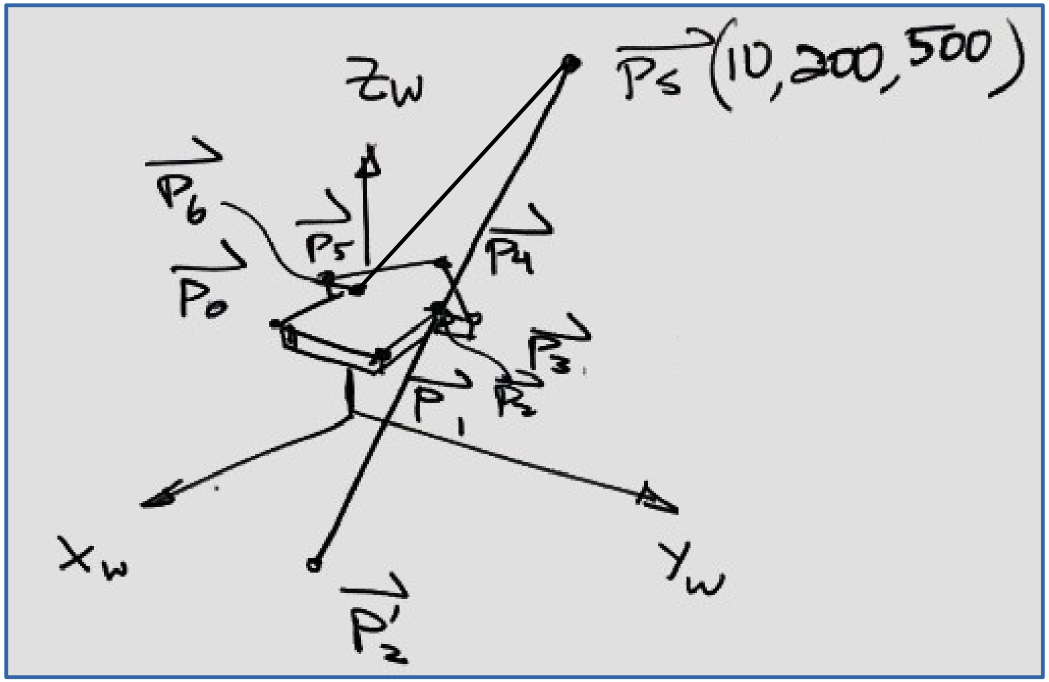
$\vec{P_2}(60,150,200) \Rightarrow I_{d\_red} = ?$ $\quad Max\Delta_4 = ?$

$$\Delta_j = \Delta_2 = \frac{1}{\sqrt{60^2+150^2+200^2}} \cdot \frac{1}{\sqrt{(10-60)^2+(200-150)^2+(50-200)^2}}$$



Interpolation Formula

Diffuse Reflection Example



From equation (1.1),

$$I_r = K_{d_r} \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \frac{1}{\|\vec{r}\|_2^2} \quad \ldots (1.1)$$

First, find ray equation to, say, one of the vertex, P2(25, 70, 50).

Then find the distance from light source to P2.

Then use the given condition, find the color internsity at P2 location.

Repeat this process to find color intensity for all the vertex from P0 to P6.

Example: Suppose we have a single light source P_s(10,200, 500), now define its (r, g, b) color, so we have single color light source as I_s(r_s, g_s, b_s) = (1.0, 0.0, 0.0), Find the diffuse reflection on the 3D floating arrow by first find color intensity on each of the marked vertex, and then find the color of each pixel of the cursor.
Assume reflection coefficient Kd=(1.0, 0.0, 0.0)

Harry Li, Ph.D

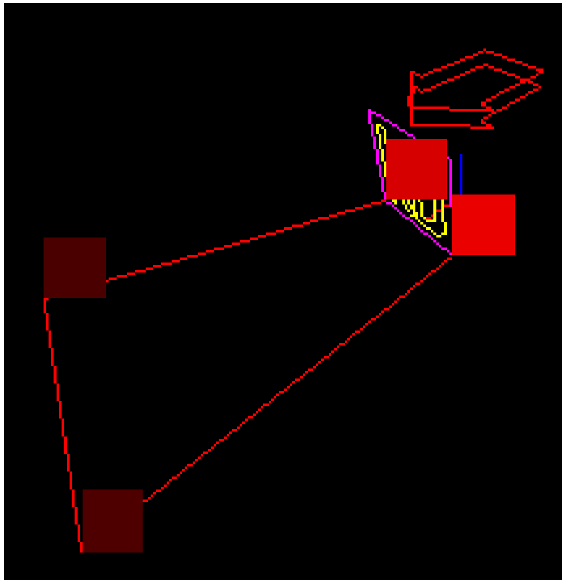
11-15-2018 Compute “Anchor” Point

```
//---------plot diffuse reflection-----------*

glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
#define     display_scaling          200000.0
#define     display_shifting    0.2
for (int i=48; i<=49; i++) {
float r, g, b;
r = display_scaling*diffuse.r[i]+display_shifting;
//r = display_scaling*diffuse.r[i];
g = diffuse.g[i]; b = diffuse.b[i] ;
glColor3f(r, g, b);
std::cout << "display_scaling*diffuse.r[i]  " << r << std::endl;
glBegin(GL_POLYGON);
glVertex2f(perspective.X[i],perspective.Y[i]);
glVertex2f(perspective.X[i]+0.1,perspective.Y[i]);
glVertex2f(perspective.X[i]+0.1,perspective.Y[i]+0.1);
glVertex2f(perspective.X[i],perspective.Y[i]+0.1);
glEnd();
}
```

Pt 5

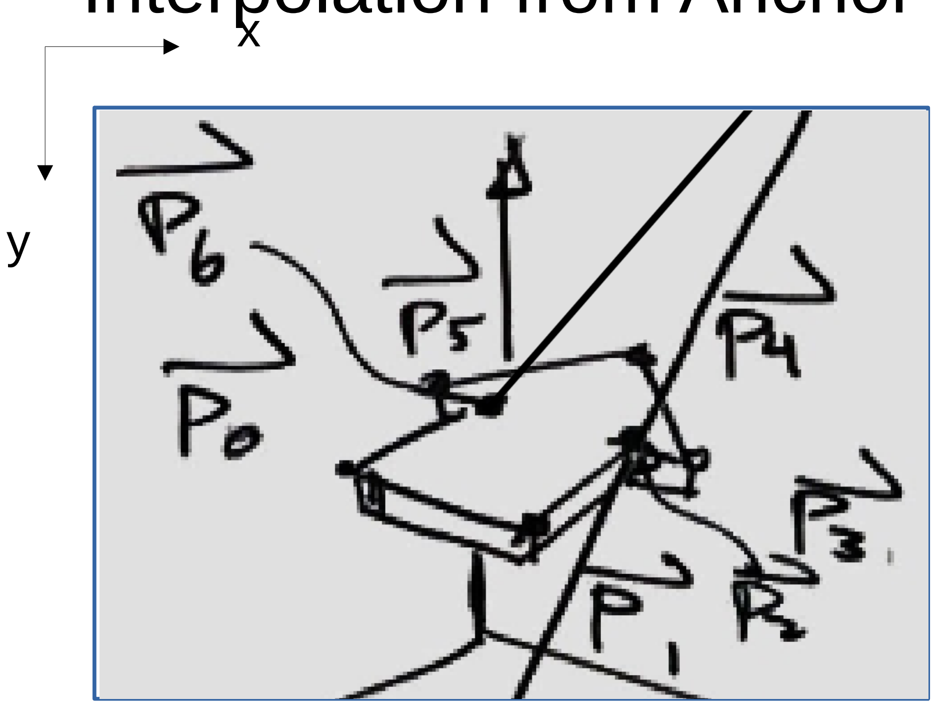
Pt 47



Pt 48

Pt 4

Interpolation from Anchor Points to Boundary Lines



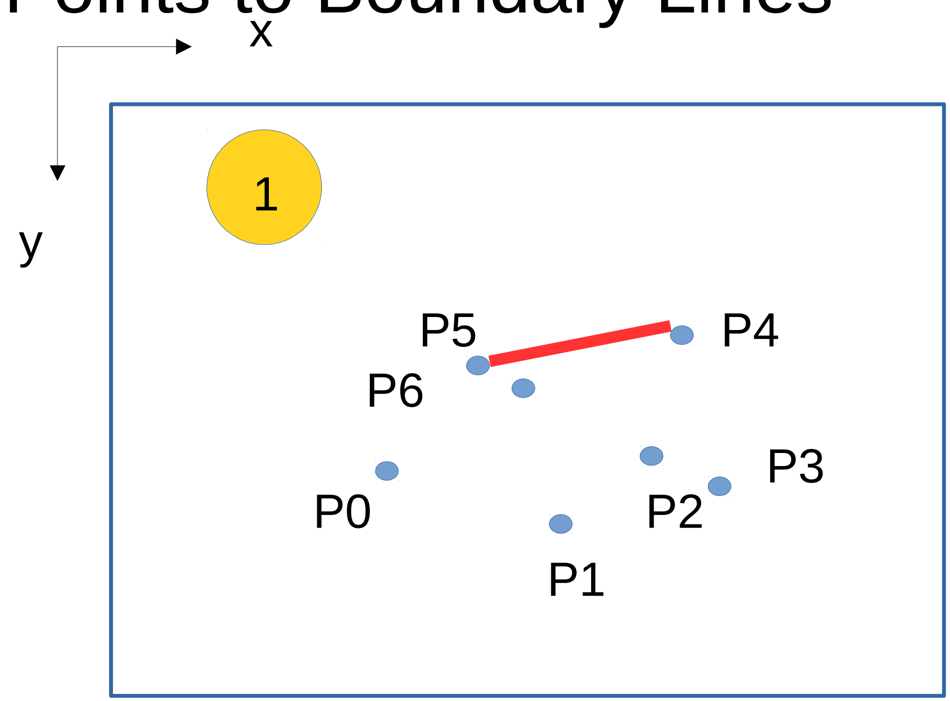
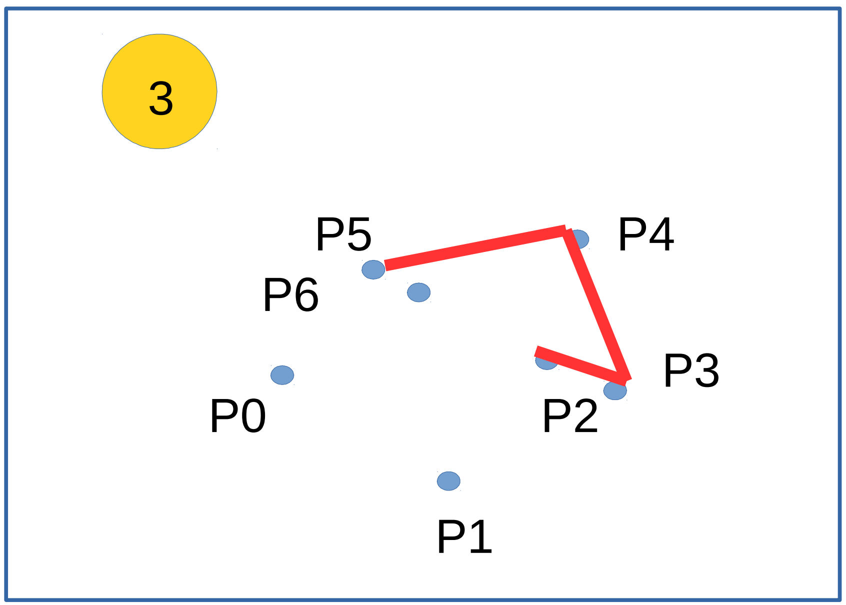
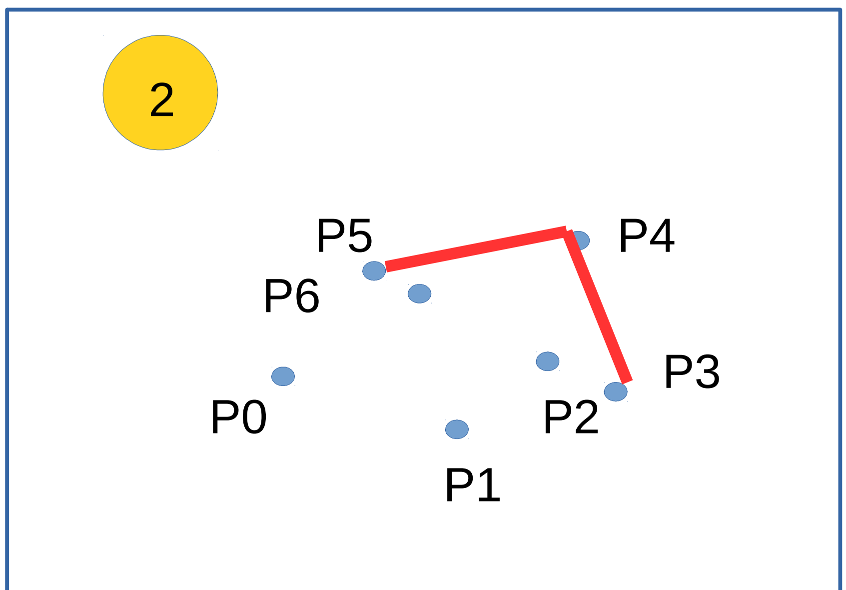


Image (graphics) plane after perspective projection.

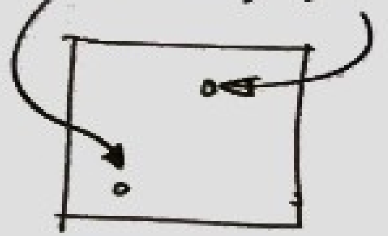




Harry Li, Ph.D

Form Boundary Lines via DDA Algorithm (1)

$\vec{P_i}(x_i, y_i)$, $\vec{P_{i+1}}(x_{i+1}, y_{i+1})$



D.D.A.
(Digital Differential Algorithm)

$y = ax + b$

$$\begin{cases} y_{k+1} = a x_{k+1} + b \\ x_{k+1} = x_k + 1 \end{cases} \quad \text{when } |a| \le 1$$

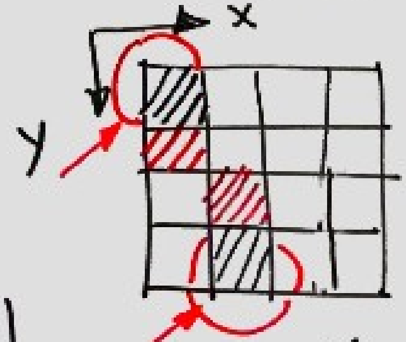
O/W, $|a| > 1$ $\cdots(2)$

$y = ax + b$, $\frac{1}{a} y = x + \frac{b}{a}$

$x = \frac{1}{a} y - \frac{b}{a}$, Hence,

$$\begin{cases} x_{k+1} = \frac{1}{a} y_{k+1} - \frac{b}{a} \\ y_{k+1} = y_k + 1 \end{cases} \quad \cdots(3)$$

Example: $\vec{P_i}(1,1)$, $\vec{P_{i+1}}(2,4)$



Sol

$$\frac{x - x_i}{y - y_i} = \frac{x_{i+1} - x_i}{y_{i+1} - y_i}$$

$$\frac{x-1}{y-1} = \frac{2-1}{4-1}$$

$$\frac{x-1}{y-1} = \frac{1}{3}, \quad y - 1 = 3(x-1)$$

$$y = 3x - 3 + 1, \quad y = 3x - 2$$

$\because |a=3| > 1$, Hence, Eqn(3):

$$x_{k+1} = \frac{1}{3} y_{k+1} - \frac{(-2)}{3}$$

Harry Li, Ph.D

Form Boundary Lines via DDA Algorithm (2)

$$\begin{cases} x_{k+1} = \frac{1}{3} y_{k+1} + \frac{2}{3} \\ y_{k+1} = y_k + 1 \end{cases}$$

for $y_k = 1$, $x_k = 1$

$$y_{k+1} = y_k + 1 = 1 + 1 = 2$$

$$x_{k+1} = \frac{1}{3} \cdot 2 + \frac{2}{3} = \frac{4}{3} = 1.33 \simeq 1$$

then, for

$$y_{k+1} = y_k + 1 = 2 + 1 = 3$$

$$x_{k+1} = \frac{1}{3} \cdot 3 + \frac{2}{3} = \frac{5}{3} = 1.67 \simeq 2$$

Interpolation To Find Boundary Color (1)

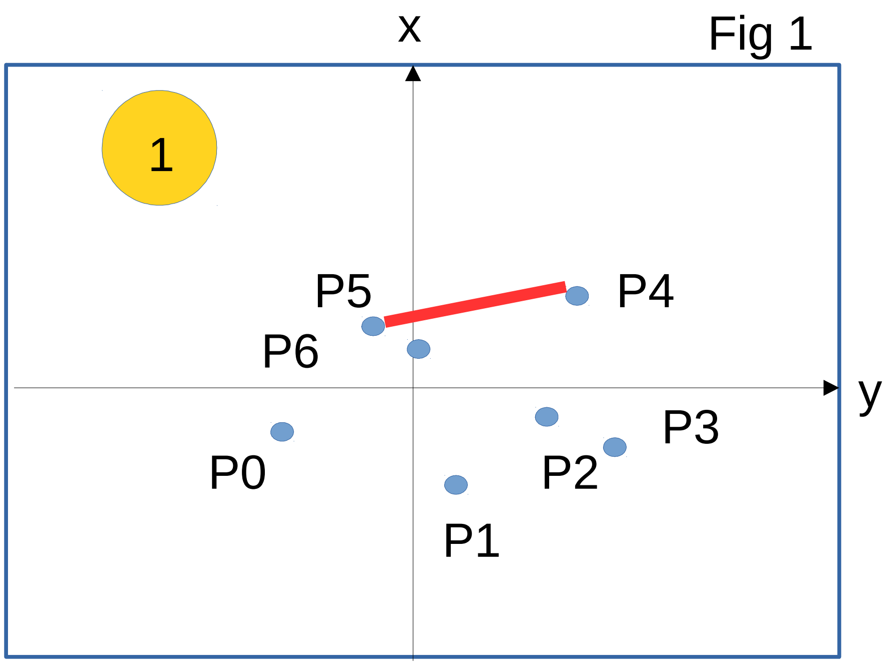
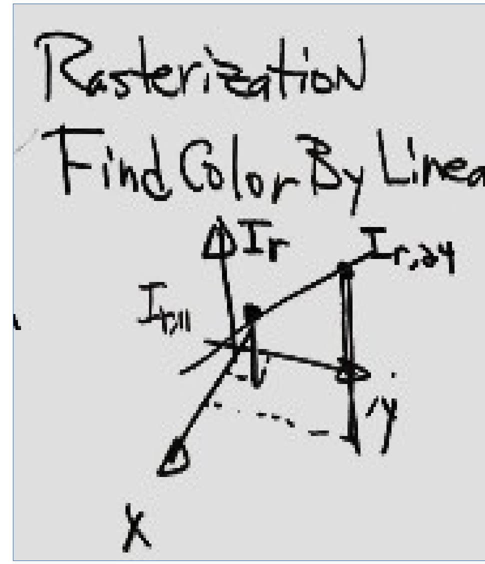


Fig 1

Fig 2


Two phase process for interpolation:
Phase I for x: left to right
Phase 2 for y: bottom up

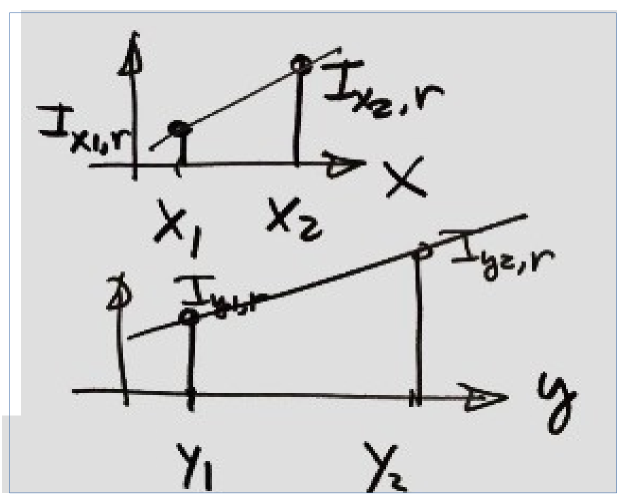


Fig 3. “r“ for red color

Phase I (L2R)

$$\boxed{\frac{x-x_i}{I_r-I_{ri}}=\frac{x_{i+1}-x_i}{I_{ri+1}-I_{ri}}}$$

$$(I_r-I_{ri})(x_{i+1}-x_i)$$
$$=(x-x_i)(I_{ri+1}-I_{ri})$$

$$I_r(x_{i+1}-x_i)-I_{ri}(x_{i+1}-x_i)=(I_{ri+1}-I_{ri})(x-x_i)$$

$$\boxed{I_r = I_{ri} + \left(\frac{I_{ri+1}-I_{ri}}{x_{i+1}-x_i}\right)(x-x_i)} \quad \ldots(4)$$

From the Example, $x_i=1$, Sub it into Eqn(4)

$$I_r = I_{ri}+\frac{I_{ri+1}-I_{ri}}{x_{i+1}-x_i}\overset{0}{(\cancel{x}_1-x_i)} = I_{ri}$$

$x_i=2$,

$$I_r = I_{ri}+\frac{I_{ri+1}-I_{ri}}{x_{i+1}-x_i}(2-1) = I_{ri}+\frac{I_{ri+1}-I_{ri}}{2-1}$$
$$= I_{ri}+I_{ri+1}-I_{ri} = I_{ri+1}$$

Harry Li, Ph.D

Interpolation To Find Boundary Color (2)
Average Up the L2R and T2B (or Bottom Up)

For Phase 2.

$$I_r = I_{ri} + \frac{I_{ri+1} - I_{ri}}{y_{i+1} - y_i}(y - y_i) \quad \ldots(5)$$

Ending Pt. (y_{i+1}) Starting Pt (y_i)

From, y_i Starting Point, y_{i+1} Ending Pt.

For $y=2$, From Eqn (5)

$$I_r = I_{ri} + \frac{I_{ri+1} - I_{ri}}{4-1}(2-1) = I_{ri} + \frac{I_{ri+1} - I_{ri}}{3} = \frac{2}{3}I_{ri} + \frac{I_{ri+1}}{3}$$

Continue till all "top-Down" traveling Done.

By Averaging the 2 Intensity

From Phase 1, @ (1,2)

$$I_{r_{L/R}} = I_{ri},$$

From Phase 2, @ (1,2)

$$I_{r_{T/B}} = \frac{2}{3}I_{ri} + \frac{1}{3}I_{ri+1}$$

Hence,

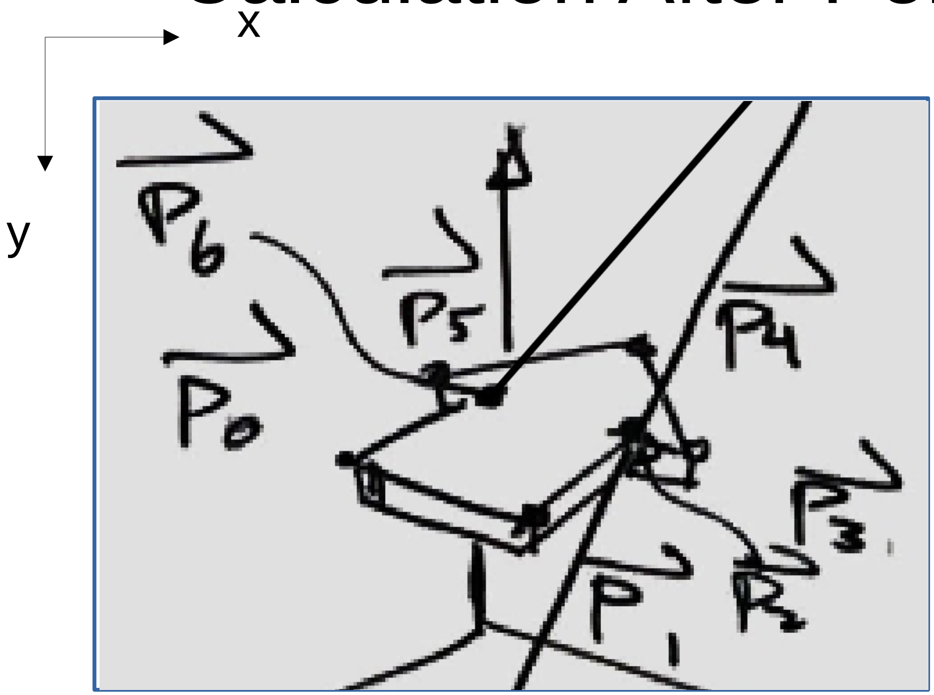
$$I_r = \frac{1}{2}\left(I_{r_{L/R}} + I_{r_{T/B}}\right)$$

$$= \frac{1}{2}\left(I_{ri} + \frac{2}{3}I_{ri} + \frac{1}{3}I_{ri+1}\right)$$

$$= \frac{1}{2}\left(\frac{5}{3}I_{ri} + \frac{1}{3}I_{ri+1}\right)$$

$$= \frac{1}{6}\left(5I_{ri} + I_{ri+1}\right)$$

Calculation After Perspective Projection



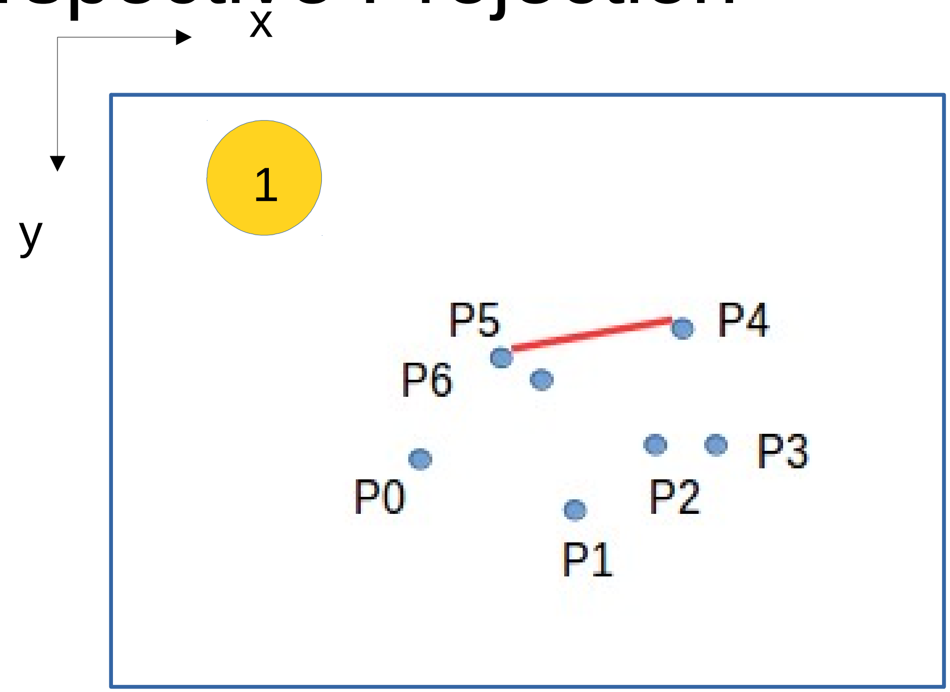


Image (graphics) plane after perspective projection.