

Note: Define  $x_w, y_w, z_w$ .

```

78 //define the x-y-z world coordinate
79 world.X[0] = 0.0; world.Y[0] = 0.0; world.Z[0] = 0.0; // origin
80 world.X[1] = 50.0; world.Y[1] = 0.0; world.Z[1] = 0.0; // x-axis
81 world.X[2] = 0.0; world.Y[2] = 50.0; world.Z[2] = 0.0; // y-axis
82 world.X[3] = 0.0; world.Y[3] = 0.0; world.Z[3] = 50.0; // z-axis
83
84 //define projection plane

```

Font Design

```

98 //-----letter-----*
99 letterL.X[0] = 10.0; letterL.Y[0] = 10.0;
100 letterL.X[1] = 20.0; letterL.Y[1] = 10.0;
101 letterL.X[2] = 20.0; letterL.Y[2] = 40.0;
102 letterL.X[3] = 40.0; letterL.Y[3] = 10.0;
103 letterL.X[4] = 50.0; letterL.Y[4] = 10.0;
104 letterL.X[5] = 30.0; letterL.Y[5] = 50.0;

```

For  $\sin\theta, \cos\theta, \sin\phi, \cos\phi$  in the matrix of the transformation pipeline.

```

159 //sin and cosine computation for world-to-viewer
160 float sPheta = Ye / sqrt(pow(Xe,2) + pow(Ye,2));
161 float cPheta = Xe / sqrt(pow(Xe,2) + pow(Ye,2));
162 float sPhi = sqrt(pow(Xe,2) + pow(Ye,2)) / Rho;
163 float cPhi = Ze / Rho;
164

```

Note: Define  $\vec{P}_s(x_s, y_s, z_s)$

```

167 world.X[45] = -200.0; world.Y[45] = 50.0; world.Z[45] = 200.0; // Ps (point source)
168 world.X[46] = 0; world.Y[46] = 0; world.Z[46] = 0; // arbitrary vector A on x-y plane
169 world.X[47] = 0; world.Y[47] = 0; world.Z[47] = 1; // normal vector for x-y plane

```

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

CMPE240

Spring 2023

31

```

171 //-----lambda for Intersection pt on xw-yw plane-----
172 float temp = (world.X[47]*(world.X[46]-world.X[45]))
173             +(world.Y[47]*(world.Y[46]-world.Y[45]))
174             +(world.Z[47]*(world.Z[46]-world.Z[45]));
175 float lambda = temp / ((world.X[47]*(world.X[45]-world.X[7]))
176                       +(world.Y[47]*(world.Y[45]-world.Y[7]))
177                       +(world.Z[47]*(world.Z[45]-world.Z[7])));
178 float lambda_2 = temp / ((world.X[47]*(world.X[45]-world.X[6]))
179                          +(world.Y[47]*(world.Y[45]-world.Y[6]))
180                          +(world.Z[47]*(world.Z[45]-world.Z[6])));
181

```

for  $\vec{R}$  Ray Equation's  $\lambda$

Find the intersection Points.

```

182 //-----ray equation to find intersection pts-----*
183 world.X[48] = world.X[45] + lambda*(world.X[45] - world.X[7]); // Ir
184 world.Y[48] = world.Y[45] + lambda*(world.Y[45] - world.Y[7]); // Ir
185 world.Z[48] = 0.0;
186

```

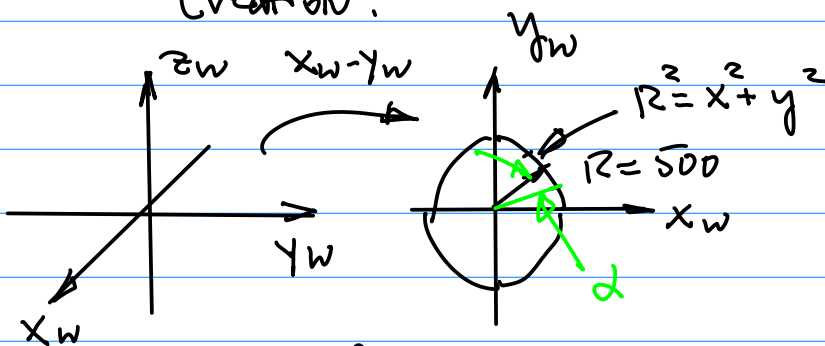
April 7 (Monday).

Note 1: Project in 3D is  
Due in 2 weeks.

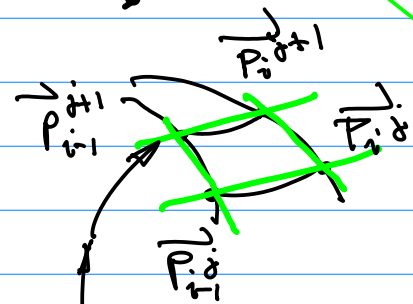
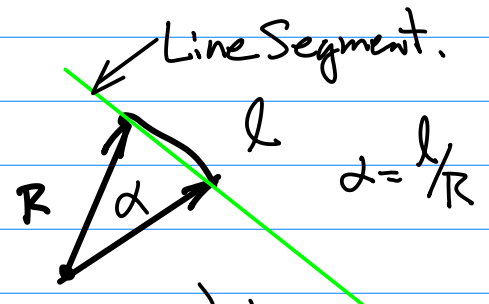
See the previous Announcement

(CANVAS Posting By the  
end of the Day Today),

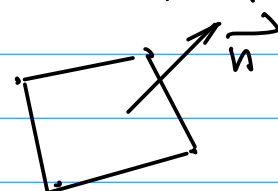
Q & A. Spherical Surface  
Creation.



Define incremental  $\alpha$   
make smaller  $\alpha \approx 5^\circ$



R Reduced By predefined  
proportion. make at least  
10 layers for Better  
Visualization.



In Summary, we'll create a  
Collection of Points

$$\left\{ \vec{P}_i(x_i, y_i, z_i) \right\}_{i=0}^{N-1}; i=0, 1, \dots, N-1$$

Example:

Ref:

Previous Project

2018F-115-lab-DiffuseReflection-Ru...

2018F-116-11diffuse20181114.cpp

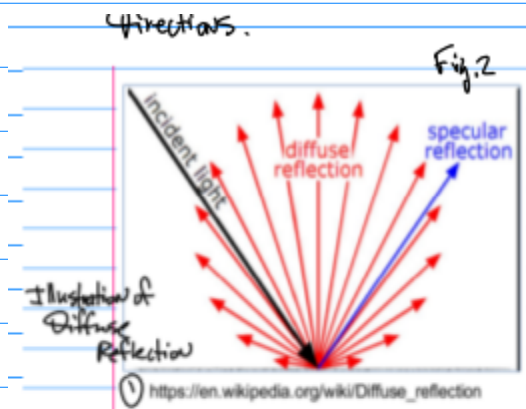
Sample  
code

Digital Differential Algorithm

2018F-117-12dda.cpp  $y = ax + b$

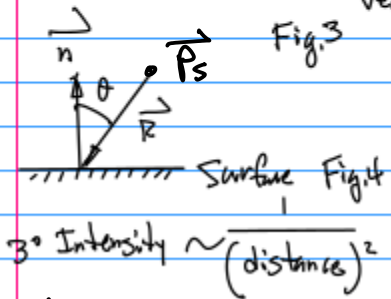
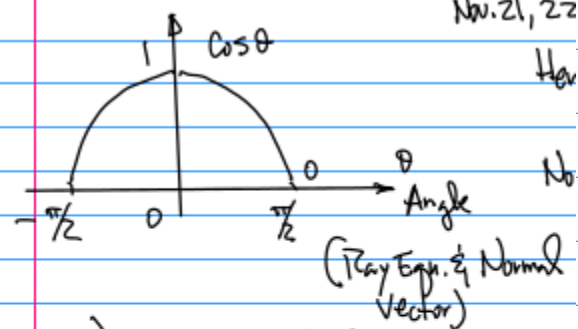
2018F-118-13diffuseInterpolation20...

1. Definition.



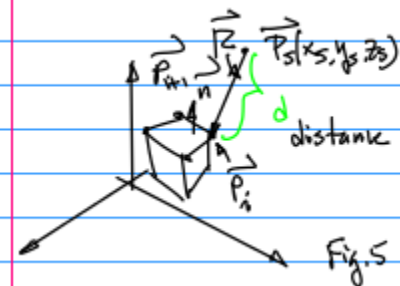
2. Intensity of the Diffuse Reflection. The Intensity of  $I(x, y) = (r(x, y), g(x, y), b(x, y))$   
red green blue  
depends on the incoming angle  
of the Ray Equation.

From Ref



Note:  $\vec{n}$  Normal Vector of the Surface  
 $\vec{R}$  Ray Equation from the  
light Source  $\vec{P}_s(x_s, y_s, z_s)$   
to the point of Interest

From pp. 48



Note: Ray Equations:  
 $\vec{r}_i$  from  $\vec{P}_s, \vec{P}_i$   
 $\vec{r}_{i+1} \dots \vec{P}_s, \vec{P}_{i+1}$   
 $\vdots$   
 $\vec{r}_{i+3} \dots \vec{P}_s, \vec{P}_{i+3}$

From the Ray Equation.

$$\vec{r} = \vec{P}_i + (\vec{P}_s - \vec{P}_i) \dots (1)$$

PP57.

$$\vec{n} \cdot \vec{r} = \|\vec{n}\| \|\vec{r}\| \cos \theta \dots (2)$$

$$\therefore \cos \theta = \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \dots (3)$$

$I_{diff}(x, y)$  OR  $I_d(x, y, z)$

↖ "World"

$$I_d(x, y, z) \approx \cos \theta = \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|}$$

... (4)

Next, Consider the distance (squared)

$$\|\vec{r}\|_z^2 = (x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2$$

then, update Eqn(4),

mode

$$I_d(x, y, z) \approx \frac{1}{\|\vec{r}\|_z^2} \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \dots (5)$$

↓

Now, Let's Consider Reflectivity.

$$\text{Reflectivity } \vec{K_d} = (K_{dr}, K_{dg}, K_{db})$$

Red Green Blue

... (6)

Update Eqn(5) with Reflectivity.

with Simplification, for Each Primitive Color.

$$\frac{dI}{dr} = K_d \frac{1}{\|\vec{r}\|_z^2} \frac{\vec{r} \cdot \vec{n}}{\|\vec{r}\| \|\vec{n}\|} \dots (7)$$

April 12 (Wed).

Note 1. Project Assignment is posted on CANVAS.

2. 5% Bonus for Using/Implementing Real 3D CAD Data.



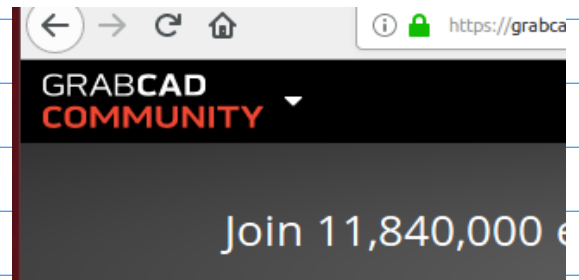
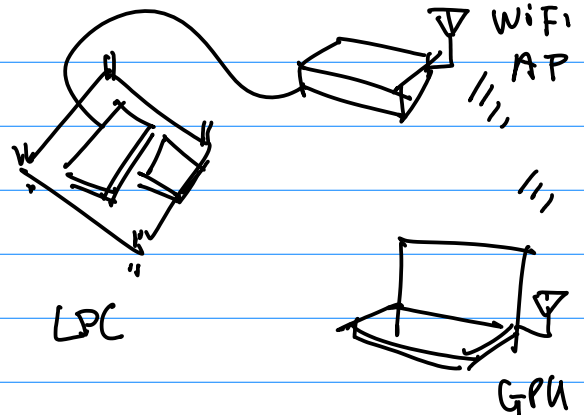
FreeCAD

<https://www.freecad.org>

FreeCAD: Your own 3D parametric modeler

FreeCAD is an open-source parametric 3D modeler made primarily to design real-life objects of any size. Parametric modeling allows you to easily modify your ...

Download · Installing on Linux · Your own 3D parametric modeler · User hub



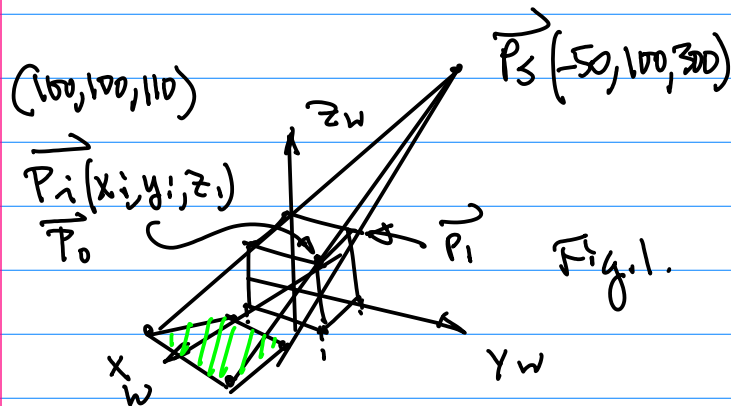


Fig. 1.

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

$$= 150^2 + 0^2 + 190^2$$

Hence,  $\frac{1}{\| \vec{r} \|} \ll \delta \dots (1)$

which makes  $I_d(x_i, y_i) \ll \delta$

Therefore, Suppose 8 bits per pixel

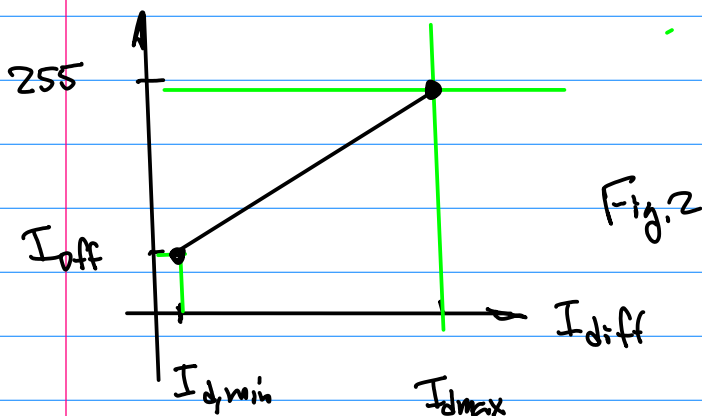


Fig. 2

Where  $I_{off} = z_0$ .

$(I_{dmin}, I_{off})$  is a point on Fig. 2.

$(I_{dmax}, 255)$  is the other point

define  
Let's a Linear mapping  
function

Let

$$(I_{dmin}, I_{off}) = (x_1, y_1) \dots (z_1)$$

$$(I_{dmax}, 255) = (x_2, y_2) \dots (z_2)$$

then,

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1} \dots (3)$$

$$y = bx + c \dots (4)$$

Now, Suppose we want to display  
diffuse Reflection for a pixel  
location  $(x_i, y_i)$

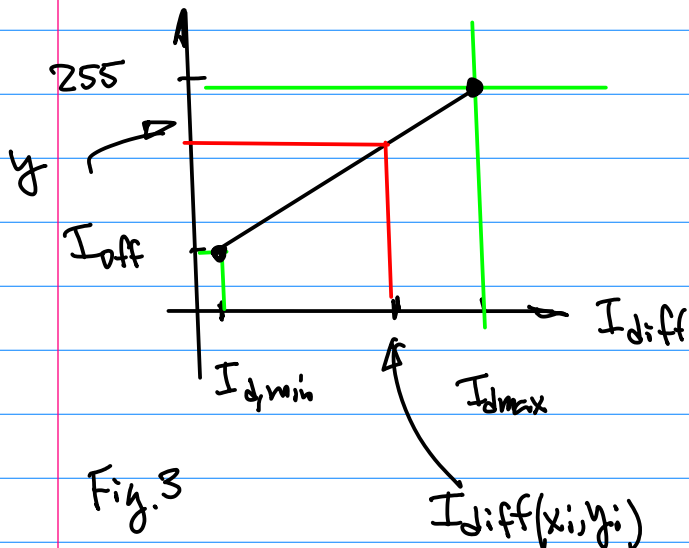
Step 1. Use Eqn (7), pp 39, to  
find  $I_{diff}(x_i, y_i)$

Step 2. Substitute

$I_{diff}(x_i, y_i)$  into this  
Eqn (4)

$$y = bx + c \quad \left| \quad x = I_{diff}(x_i, y_i) \right.$$

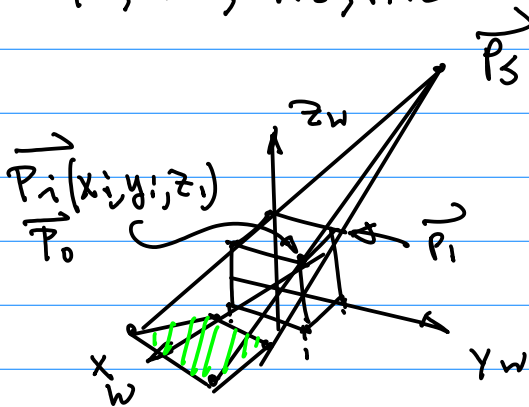
$$= b \cdot I_{diff}(x_i, y_i) + c$$



$y$  is the final intensity level for the color.

Now, we have 4 vertices with Diffuse Reflection Result.

$\vec{P}_i, \vec{P}_{i+1}, \vec{P}_{i+2}, \vec{P}_{i+3}$



Now, After Perspective Project

Fig. 4

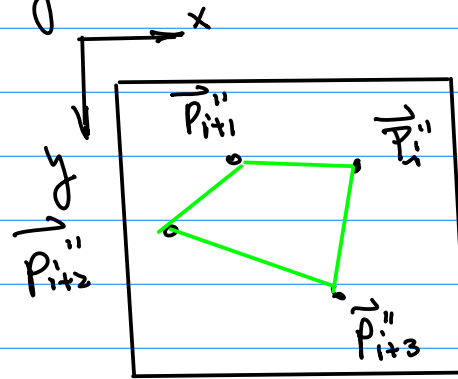


Fig. 5

Find Diffuse Reflection on the Boundary Lines (Green).

April 17 (Monday).

Note 1. Check Canvas for the Last Project Announcement.

Sample code Reading for Diffuse Reflection Computation.

2018F-116-11diffuse20181114.cpp

Example.

Preliminary 1. Intersection Pts. from the Ray Equations

```

182 //-----ray equation to find intersection pts-----*
183 world.X[48] = world.X[45] + lambda*(world.X[45] - world.X[7]);
184 world.Y[48] = world.Y[45] + lambda*(world.Y[45] - world.Y[7]);
185 world.Z[48] = 0.0;

```

Note 1. Define Reflectivity, Spring 2023

```

191 //-----diffuse reflection-----*
192 pt_diffuse diffuse; //diffuse.r[3]
193
194 //-----reflectivity coefficient-----*
195 #define Kdr 0.8 for Ized color.
196 #define Kdg 0.0
197 #define Kdb 0.0
198

```

Note 2. Distance. To Speed up the Computation, No. Sqrt Needed.

```

202 //-----compute distance-----*
203 float distance[UpperBD];
204 for (int i=48; i<=49; i++) {
205     distance[i] = sqrt(pow((world.X[i]-world.X[45]),2)+
206                       pow((world.Y[i]-world.Y[45]),2)+
207                       pow((world.Z[i]-world.Z[45]),2) );
208     //std::cout << "distance[i] " << distance[i] << std:::
209

```

Note 3. Compute Cosθ for Diffuse Reflection.

```

229 tmp_dotProd[i] = world.Z[i]-world.Z[45];
230 std::cout << " tmp_dotProd[i] " << tmp_dotProd[i] << std::endl;
231
232 tmp_mag_dotProd[i] = sqrt(pow((world.X[i]-world.X[45]),2)+
233                           pow((world.Y[i]-world.Y[45]),2)+
234                           pow((world.Z[i]-world.Z[45]),2) );
235 std::cout << " tmp_mag_dotProd[i] 1 " << tmp_mag_dotProd[i] << std::endl;
236
237 angle[i] = tmp_dotProd[i]/ tmp_mag_dotProd[i];
238 std::cout << "angle[i] " << angle[i] << std::endl;
239

```

Note 4. Theoretical Part of the Diffuse Reflection. The Result is Very Small

```

241 diffuse.r[i] = Kdr * angle[i] / pow(distance[i],2) ;
242 diffuse.g[i] = Kdg * angle[i] / pow(distance[i],2) ;
243 diffuse.b[i] = Kdb * angle[i] / pow(distance[i],2) ;
244 }

```

Very Big Distance

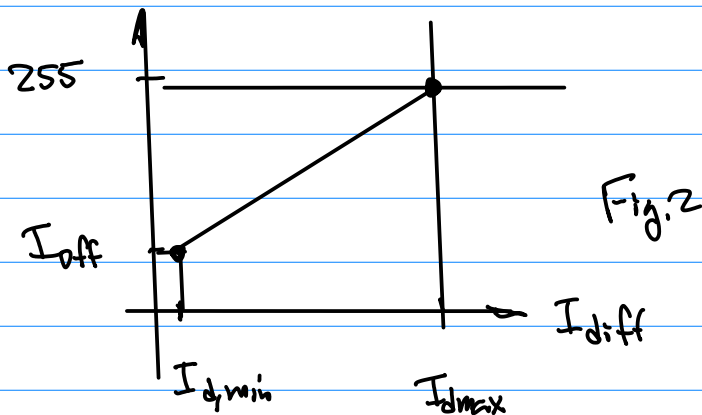


Fig. 2

Sample code for the post processing:

} Add offset = 20  
 } Map the diffuse reflection  
 [offset, 255]

CMPE240-Adv-Microprocessors / 2018F / 2022S-101-notes2-cmpe240-2022-04-18.pdf.pdf.20.pdf

Post processing function, PP13

$$\frac{x - x_2}{y - y_2} = \frac{x_1 - x_2}{y_1 - y_2} \dots (5)$$

495  
496  
497  
498  
499

```
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];
nlColor3f(r, g, b);
```

Example: Bi-Linear Interpolation of Diffuse Reflection.

From Eqn (5), PP14.

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

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

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

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

... (1)

$$a \quad y = bx + c, \quad y = \frac{b}{a} x + \frac{c}{a}$$

... (2)

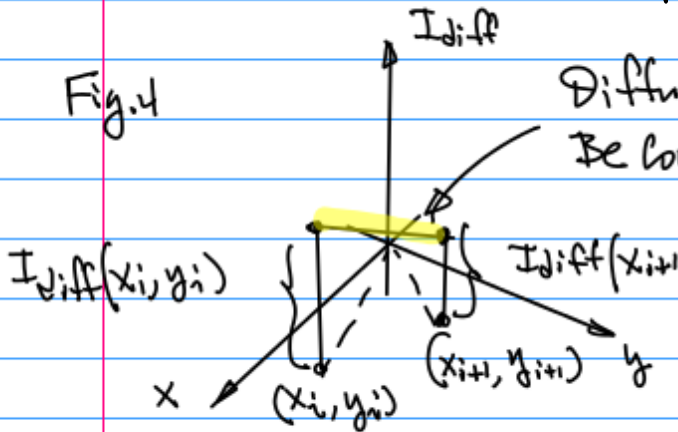
$$\frac{b}{a} = \frac{y_2 - y_1}{x_2 - x_1} \dots (2-b)$$



// 2018F / 2020S-APL29-BilinearDiff1.jpg

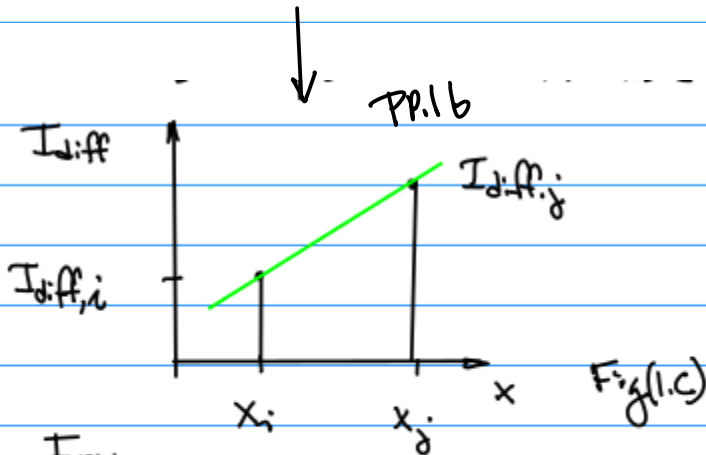
PP15.

Fig.4



x, y are from the Display Device

$\vec{P}_i(I_{diff,i})$  and  $\vec{P}_{i+1}(I_{diff,i+1})$  as Starting & Ending point.



From,

$$\frac{x - x_2}{y - y_2} = \frac{x_1 - x_2}{y_1 - y_2} \quad \dots (5)$$

PP.17.

then, derived the following Equations

$$y = \frac{y_2 - y_1}{x_2 - x_1} x - \frac{y_2 - y_1}{x_2 - x_1} x_2 + y_2 \quad \dots (1)$$

$$a) y = bx + c, \quad y = \frac{b}{a} x + \frac{c}{a} \quad \dots (2)$$

$$\frac{b}{a} = \frac{y_2 - y_1}{x_2 - x_1} \quad \dots (2-b)$$

$$\frac{c}{a} = -\frac{y_2 - y_1}{x_2 - x_1} x_2 + y_2 \quad \dots (2-c)$$

Therefore.

2020S-APL29-BilinearDiff2.jpg

$$I_{diff,x} = \frac{I_{diff,j} - I_{diff,i}}{x_j - x_i} x - \frac{I_{diff,j} - I_{diff,i}}{x_j - x_i} x_j + I_{diff,j} \quad \dots (3)$$

For  $I_{diff}$  w.r.t  $y$ . we have (Symmetric)

$$I_{diff,y} = \frac{I_{diff,j} - I_{diff,i}}{y_j - y_i} y - \frac{I_{diff,j} - I_{diff,i}}{y_j - y_i} y_j + I_{diff,j} \quad \dots (4)$$

Hence,

$$I_{diff} = \frac{1}{2} [I_{diff,x} + I_{diff,y}] \quad \dots (5)$$

April 19 (Wed)

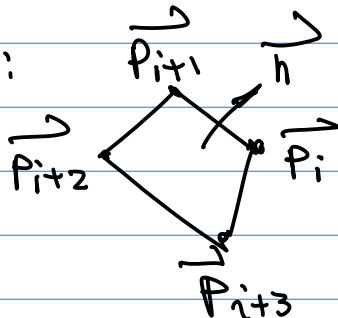
Final Exam:

### Group I Classes

Group I classes are those classes which meet M, W, F, MTW, MWR, MTWF, MWR, MTWRF, MW, WF, MWF, MF, TW, WR, MT, WS.

Regular Class Start Times	Final Examination Days	Final Examination Times
7:00 through 8:25 AM	Friday, May 19	7:15-9:30 AM
8:30 through 9:25 AM	Tuesday, May 23	7:15-9:30 AM
9:30 through 10:25 AM	Thursday, May 18	7:15-9:30 AM
10:30 through 11:25 AM	Monday, May 22	9:45 AM-12:00 PM
11:30 AM through 12:25 PM	Wednesday, May 17	9:45 AM-12:00 PM
12:30 through 1:25 PM	Friday, May 19	12:15-2:30 PM
1:30 through 2:25 PM	Tuesday, May 23	12:15-2:30 PM
2:30 through 3:25 PM	Thursday, May 18	12:15-2:30 PM
3:30 through 4:25 PM*	Monday, May 22	2:45-5:00 PM
4:30* through 5:25 PM*	Wednesday, May 17	2:45-5:00 PM

Example:



Detect the orientation

$$\{ \vec{P}_i, \vec{P}_{i+1}, \dots, \vec{P}_{i+3} \mid i=1, 2, \dots, N \}$$

$$(\vec{P}_i - \vec{P}_{i+3}) \times (\vec{P}_{i+2} - \vec{P}_{i+3}) \quad \dots (1)$$

Hence,

$$\vec{n} = \frac{(\vec{P}_i - \vec{P}_{i+3}) \times (\vec{P}_{i+2} - \vec{P}_{i+3})}{\|(\vec{P}_i - \vec{P}_{i+3}) \times (\vec{P}_{i+2} - \vec{P}_{i+3})\|_2} \quad \dots (2)$$

$$\vec{C} \times \vec{D} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ C_x & C_y & C_z \\ D_x & D_y & D_z \end{vmatrix} \quad \vec{S} \cdot \vec{B} = S$$

Google Ref.

$$\vec{S} \cdot \vec{A} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ S_x & S_y & S_z \\ A_x & A_y & A_z \end{vmatrix} = S(A_x \cos \theta - A_y \sin \theta)$$

Consider DDA Algorithm,  
Digital Differential Algorithm

$$y = bx + c \quad \dots (1)$$

To plot Equation/Line Segment  
on a finite Display Device.

HD, 4K etc.

Technical challenges:

- 1° "GAPS" problem
- 2° Removal of multiplication.

Consider Computation of  $y_k, y_{k+1}$  :  
We have

for  $x_k$ , from Eqn(1).

$$y_k = bx_k + c \quad \dots (1a)$$

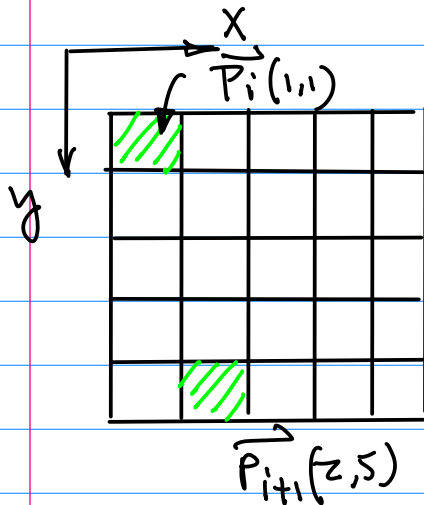
for  $x_{k+1} = x_k + 1$ , then

$$y_{k+1} = \underbrace{bx_{k+1} + c}_{\text{multiplication.}}$$

$$= b(x_k + 1) + c = bx_k + b + c \quad \dots (1b)$$

$$= y_k + b$$

Example: Given  $\vec{P}_i = (1, 1), \vec{P}_{i+1} = (1, 5)$   
Use Eqn(1a) or (1b) to plot a  
Line.



Print  $y_{k+1}$  on a pixel location  
With a gap  
To solve this problem, make the  
Slop of the given Line is less  
than 1.

(Absolute value of

$$b = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = 4$$

Solve for C :

$$y = bx + c \Big|_{b=4} = 4x + c$$

$$1 = 4 + c$$

$$\therefore c = -3$$

Hence

$$y = 4x - 3 \quad \dots (3)$$

Let  $x_k = 1, y_k = 1$  (From Eqn(3))

$$x_{k+1} = x_k + 1 = 1 + 1 = 2$$

From Eqn(1b)

$$y_{k+1} = y_k + b = 1 + 4 = 5$$

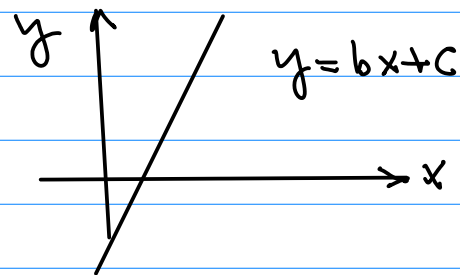
Great ! which is a problem.

April 24 (Monday).

Example: Continuation on  
DDA.

Note: When the slop  $|b| > 1$   
then Eqn (1-b), pp45,  
will land the Next

Consider  $y = bx + c$ , where  
 $|b| > 1$ . ... (1)



From Eqn(1),

$$y/b = x + c/b$$

$$x = \frac{1}{b}y - \frac{c}{b} \quad \dots (2)$$

where  $|\frac{1}{b}| < 1$

$$x_k = \frac{1}{b}y_k - \frac{c}{b}$$

$$y_{k+1} = y_k + 1 \quad \dots (3a)$$

$$x_{k+1} = \frac{1}{b}y_{k+1} - \frac{c}{b}$$

$$= \frac{1}{b}(y_k + 1) - \frac{c}{b}$$

$$= \frac{1}{b} + \frac{1}{b}y_k - \frac{c}{b}$$

$x_k$

$$x_{k+1} = x_k + \frac{1}{b} \dots (3b)$$

Going Back to the Same Example.

for  $k=1$ ,  $y_k=1$ ,  $x_k=1$ .

for  $k=2$ .

$$y_2 = y_1 + 1 \left( = y_{k+1} \Big|_{k=1} \right) = 2$$

$$x_2 = x_1 + \frac{1}{b} = 1 + \frac{1}{4} = 1.25 \approx 1$$

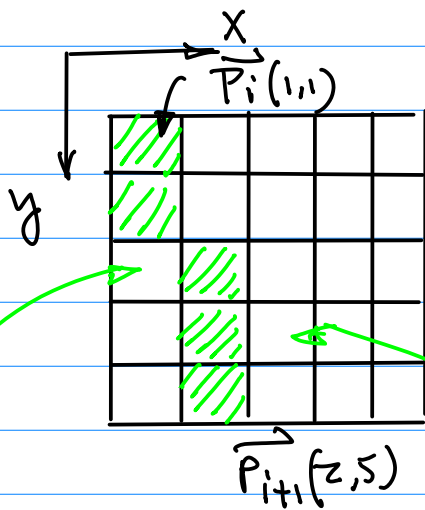


Fig. 1

for  $y_3=3$ ,

$$x_3 = x_2 + \frac{1}{b} = 1.25 + 0.25 = 1.5 \approx 2$$

for  $y_4=4$ ,

$$x_4 = x_3 + \frac{1}{b} = 1.5 + 0.25 = 1.75 \approx 2$$

Diffuse Reflection on the interior points.

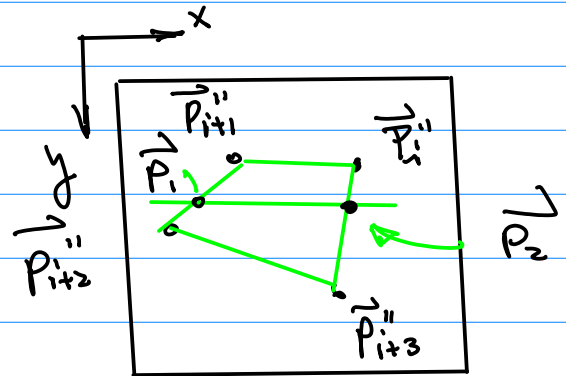


Fig. 2

$\vec{P}_1, \vec{P}_2$  Are Both ON the Boundary

Hence, their (1) Pixel Location

are Computed By DDA ;

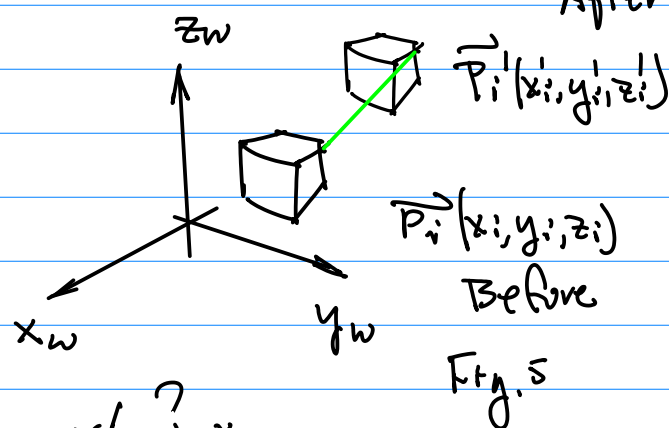
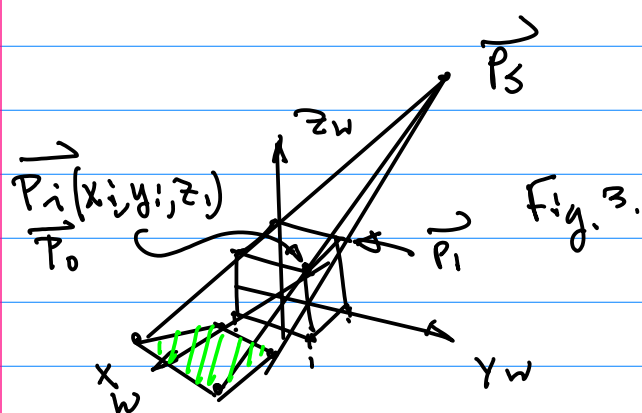
(2) Diffuse Reflection are Computed By Eqs (3), (4), and (5) on TP(4,45);

Therefore. 2020S-APL29-BilinearDiff2.jpg

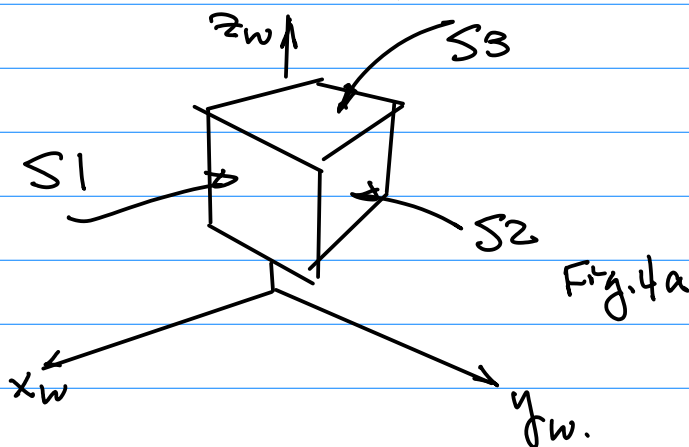
$$I_{diff,x} = \frac{I_{diff,j} - I_{diff,i}}{x_j - x_i} x - \frac{I_{diff,j} - I_{diff,i}}{x_j - x_i} x_j + I_{diff,j} \dots (3)$$

Example: Decoration Algorithm.

Background: 3D Transforms.



Decorate the Cube Surface.

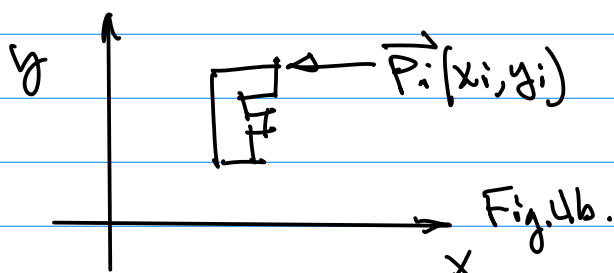


$$x_i' = x_i + \Delta x$$

After Before

$$y_i' = y_i + \Delta y, z_i' = z_i + \Delta z$$

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots (4)$$



$$\begin{pmatrix} x_i' \\ y_i' \\ z_i' \\ 1 \end{pmatrix} = T \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \dots (5)$$

Use 2D Pattern  $\{P_i(x_i, y_i) | i=1, 2, \dots, N\}$   
to Decorate 3D Surfaces.

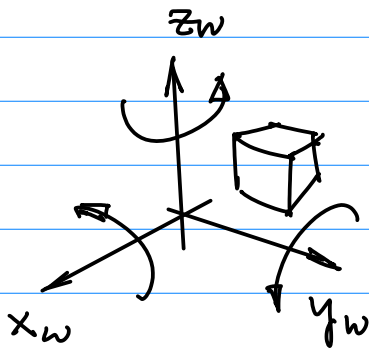


Fig. 6

From 2D Rotation Matrix

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \dots (6)$$

$$R_{x_w} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots (7)$$

April 26 (Wed).

Example: Linear Decoration Algorithm.

Surface  $\rightarrow$  plane

Continuation on pp 48, Fig. 4a ~ 4b.

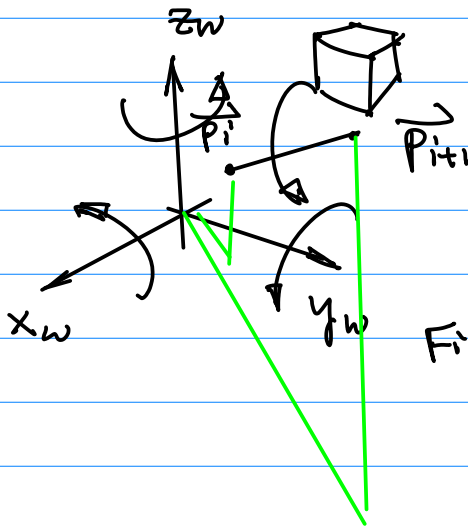


Fig. 7

Rotation w.r.t.  $x_w$ -axis.

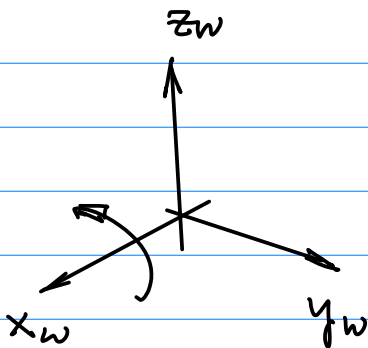


Fig. 8

Methodology: Observe/Discover the independent variable which stays the constant.  $x_n \rightarrow$

Rotation on  $y_w$ - $z_w$  plane.

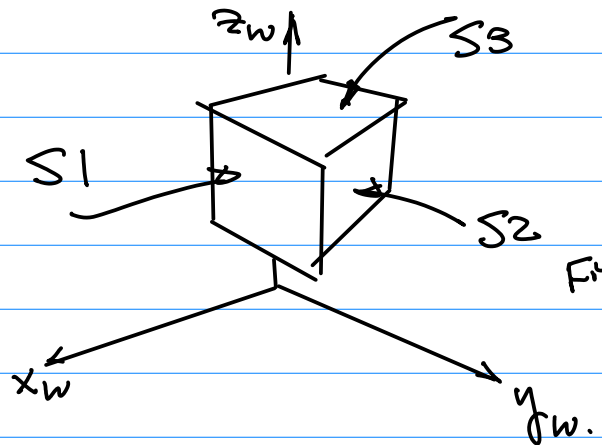


Fig. 4a

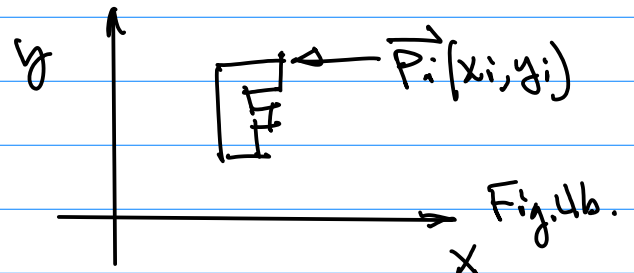
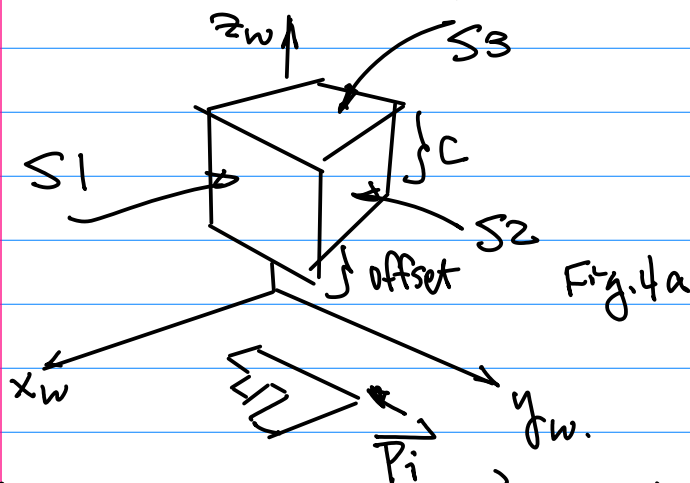


Fig. 4b.

Step 1. Given  $\{\vec{P}_i(x_i, y_i) | i=1, 2, \dots, K\}$

Redefine  $\{\vec{P}_i(x_i, y_i)\}$  in the World Coordinate System.

$$\{\vec{P}_i(x_i, y_i, z_i) \mid z_i = 0, i = 1, 2, \dots, K\} \quad \dots (1)$$



Note: Check the scale of  $\{\vec{P}_i(x_i, y_i, z_i)\}$  to make sure it match the Size of the Surface (plane) to be decorated:

Step 2. Consider Surface 3.

Remark: 1<sup>o</sup> Identify the parallel plane for the surface;

2<sup>o</sup> Identify the indep. Variable  $\hat{=}$  Function of the plane.

$x_w - y_w$

$x_w$  (Indep).  
v.s.  $y_w$  (Function)

After	$\gamma$	Before
$x_i'$	$=$	$x_i$ (Indep)
$y_i'$	$=$	$y_i$
$z_i'$	$=$	$C + \text{offset}$
		$\dots (1)$

Consider S1:

Find A parallel plane  $y_w - z_w$ .

Indep. v.s. Function  
Variable  $y_w$  v.s.  $z_w$ .

After	$\gamma$	Before
$y_i'$	$=$	$x_i$ (Indep.)
$z_i'$	$=$	$y_i$ (Function)
$x_i'$	$=$	$C$
		$\dots (2)$

Consider S2

Parallel plane  $z_w - x_w$

Indep:  $z_w$ , v.s. Function  $x_w$

After		Before
$z_i'$	$=$	$x_i$ (Indep.)
$x_i'$	$=$	$y_i$ (Function)
$y_i'$	$=$	$C$
		$\dots (3)$

Optional project Topic (Discussion):  
"Mirror" Image.

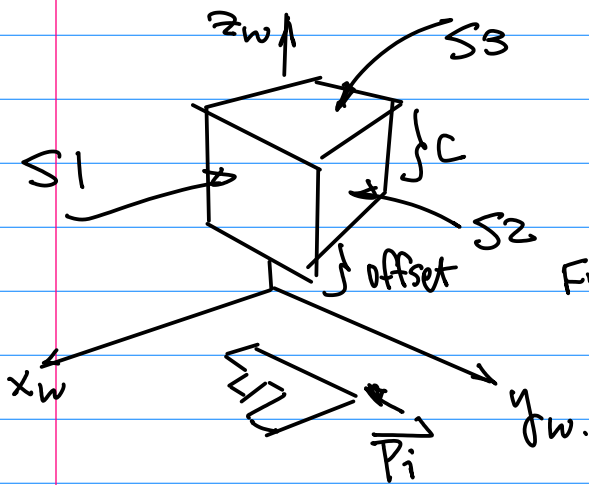
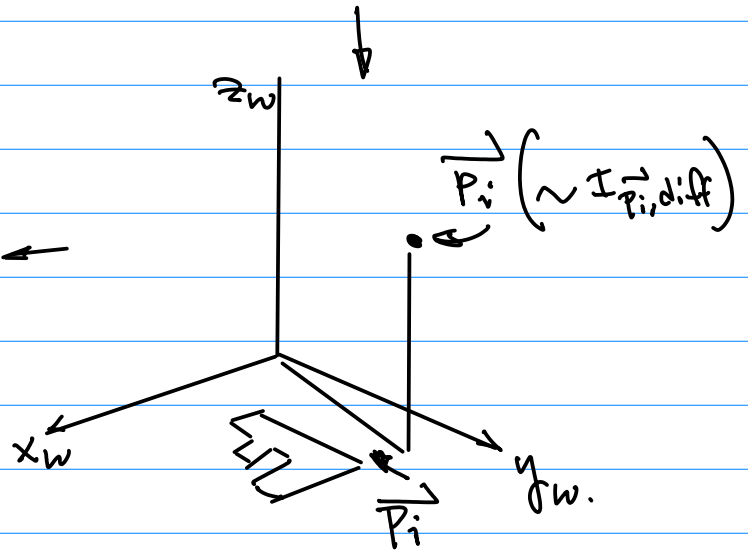
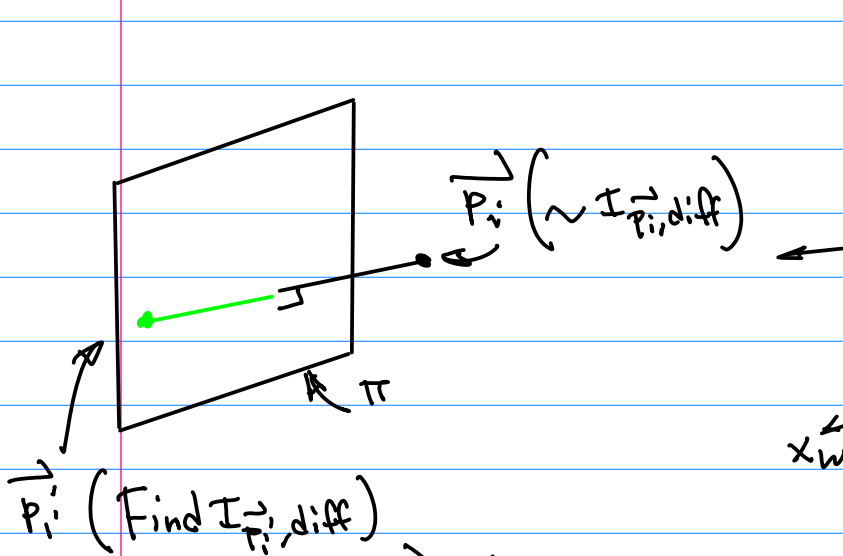
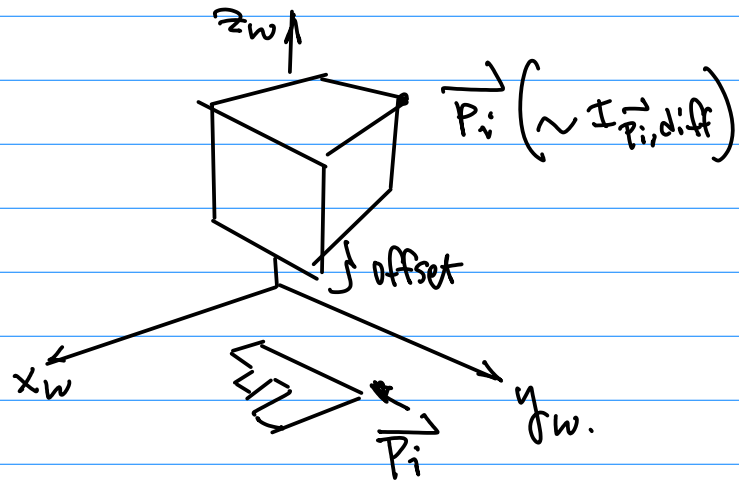


Fig. 4a



Line Eqn:  $\vec{P} = \vec{P}_i + \lambda \vec{d} \dots (4)$

$\vec{d}$  is defined by the plane.  
Cross product of 2 vectors  
on the " $\pi$ " plane

$\lambda$  (Defines the distance  
from  $\vec{P}_i$  to the  $\pi$  plane)

$2\lambda$  Reflection Point

Transformation Pipeline

Point on 2D  $\pi$ 's plane  
(Location)  
( $x_i'', y_i''$ )

Copy diff use Reflection from  
 $\vec{P}_i(x_i, y_i, z_i)$