

Nov. 6th (Monday).

Note 1. Midterm Exam is Scheduled ON Next Monday.

Review Session in-class is scheduled ON Wednesday.

Today's Topics:

1° SSPI Init Code.

2° Hardware Design for LCD/CPU Interface Design.

Example: Line 210-246.

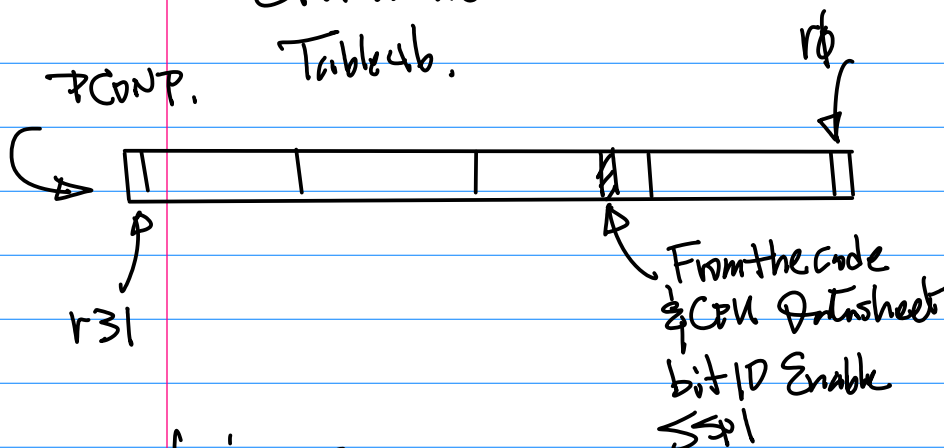
Line 224 Naming Convention

LPC_SC → PCOMP

Section 4.8.5. PP61

CPU Datasheet.

Table 4b.



Line 227.

LPC_SC → PCUKSEL

Line 229

Line 230

LPC_SC → PINSEL0.

TP.58, Table 42.

Section 4.7.3. TP57.

Note: Draw a 32 Bit SPR illustration.

Connect to CPU Datasheet (Tables for the SPR).

Memory Bank holding this SPR. Code Implementation.

Note: The Sequence to Init & Config SPRs for SSPI Interface. (SPI)

PCOMP → PCUKSEL → PINSEL0

Line 233/234 CS (Chip select: e.g. Select/Enable my LCD Display module)

Line 238

SSPI Init.

TP.431. Table 371

Use Case Leads to Design Requirements, for Example, 2D G.E. Design. Frame Rate, Resolution of the Display →

Carry out the Design By Using CPU Datasheet, and formula

$$f_{PWM} = \frac{PCLK}{CPDSUPR * (SCR + 1)}$$

Then, Coding.

Homework: Due Nov. 14th (Sunday).

1st Requirements:

a) Based on the Homework of Drawing A wire frame Cube in $x_w-y_w-z_w$, Add a point Light Source, such as

$$P_s(x_s, y_s, z_s) = (-5, 50, 250),$$

b) Use the vertices from the Top Surface of the cube to generate 4 Ray Equations, then Compute the intersection Points on x_w-y_w plane

Note: Computation is Carried out Before the Transformation Pipeline, e.g., in $x_w-y_w-z_w$ Coordinate.

c) Draw the shadow first Before Drawing the Cube.

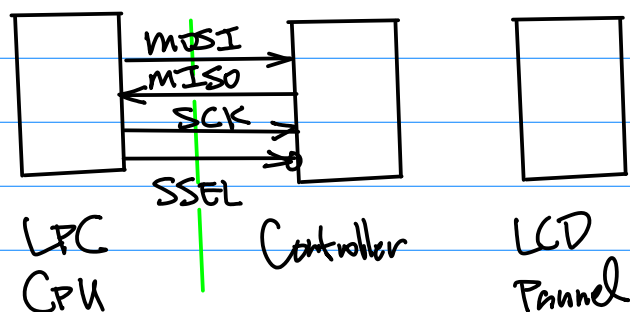
Note: Design/Select A proper dark color for the shadow.

Note: please work/Discuss this homework with your team. But coding has to Be individually, No code can be shared.

Note: Please Bring Your Board for

Show & Tell, Demo On Monday, Nov. 20 (Monday).

Consider Hardware Design for LCD Controller Interface.



LCD with Controller Build In.

Note: please provide Clear indication of the Signal Flow By Drawing Arrow on Each Signal Line.

Also, place a circle "0" for Active Low Signal.

Nov. 8 (Wed).

Note 1. Midterm Exam is Next Monday.

- please Bring your Prototype System;
- Bring Blank Papers for the exam hand Calculation;

Cmpe240
F2023

- use
- c) "printf" to print your First Name, Last Name and SID, while executing Program During the example.
- d) There are 3 Questions.

Question in the CPU Architecture, memory, SPRs. Memory Bank,

Question on the subject of Building A prototype System.

SCH, pin Connectors, functionality of the pins. and interface to LCD.

Question in the Area Design. Debugging, hand Calculation.

- e) Naming Convention of the Zip.

FirstName-LastName-SID-Cmpe240-mid.zip

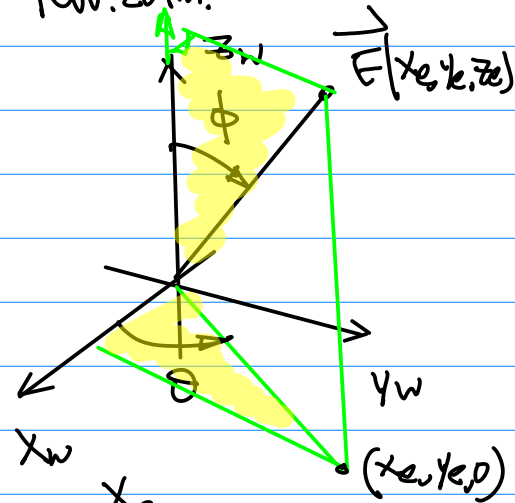
One pdf (integration of All pdf files)

- f) Resolution of the photos.
Not too high, Not too low
Resolution, 1 ~ 4 MB.

- g) Submission on CANVAS.
No Late Submission, No E-mail Submission

Homework 53/

Note: Shadow Computation
Due Nov 19 (Sun).
Please Bring your Board for Demo on Monday.
Nov. 20th.



$$\cos \theta = \frac{x_e}{\sqrt{x_e^2 + y_e^2}}$$

sin

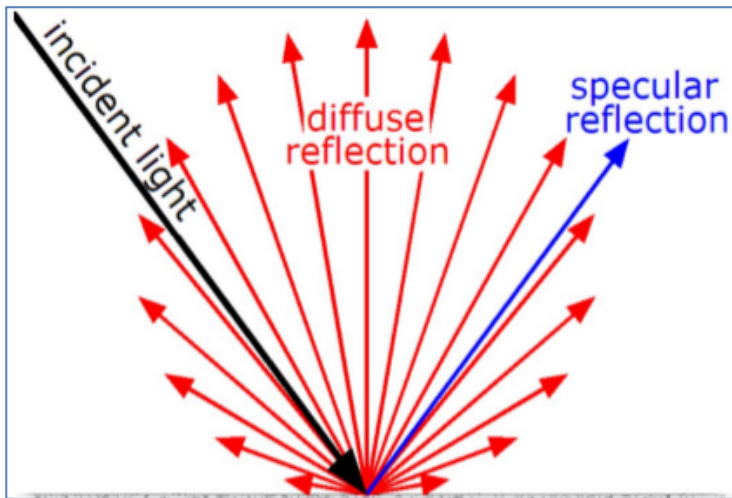
$$\cos \phi = \frac{z_e}{\rho} = \frac{z_e}{\sqrt{x_e^2 + y_e^2 + z_e^2}}$$

Nov.15 (Wed).

Ref: Pp.1. Definition/Concept of Diffuse Reflection.

2018S-23-lec7-DiffuseReflection-v6-2018-4-25.pdf

Diffuse Reflection



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., $1/(r^2)$, where r is being the distance from the light source to the surface.

Specular vs. 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.

Harry Li, Ph.D.

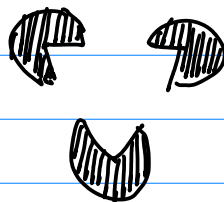
Ref2: on the class github.

2022F-101-notes-cmpe240-2022-11-30 (1).pdf

Example: Background On Diffuse Reflection.

Ref3:

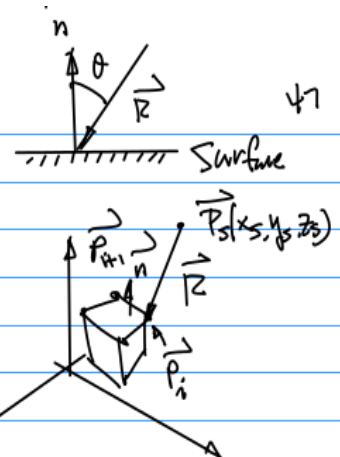
Pp54.



Brief Introduction, And 3 Lighting models.

CMPE240

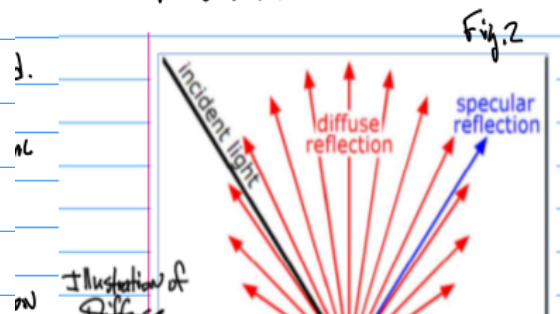
Nov.14, 22



Note:

1st Definition: PP.St. Reflection Uniformly in All different Directions.

Fig.1.



Three models. see Ref 3. pp. 54

$$I(x, y) = I_1(x, y) + I_2(x, y) + I_3(x, y)$$

Diffuse
Reflection

Specular
Reflection

Ambient
Light

Note: Specular

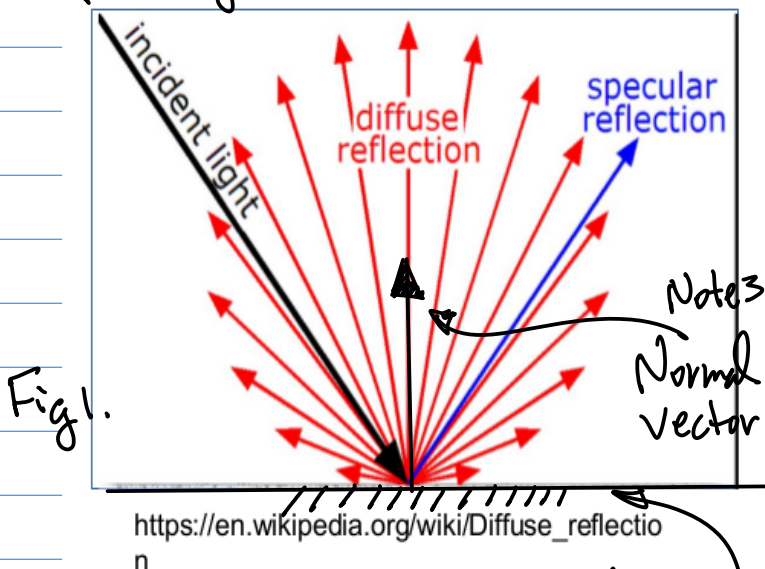
... (2)

Definition of Diffuse Reflection.

Ref. 1. Pp 1.

Reflection Uniformly reflects
the Incoming light in All
Different Direction.

Note 1. $P_s(x_s, y_s, z_s)$ Incident Light, White Color.
for example $r=g=b=255$ Diffuse Reflection



Note 3.
Normal
Vector.

Note 2:

Surface of Reflection

Color of A surface : physical
Characteristic.

Definition of Reflectivity to Describe

the Characteristics of A surface
Color.

Reflection of the color
leads to the perception
of the color.

Two Key Characteristics:

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

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

Specular vs. diffuse reflection

where k_r : Reflectivity for red

k_g : green

k_b : blue

Normalized, so

$k_r \in [0, 1]$

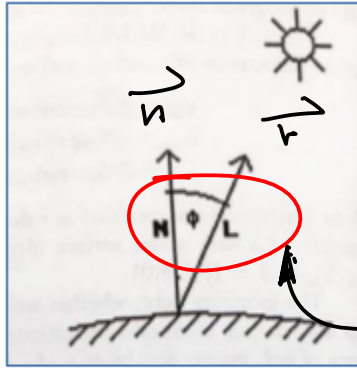
$k_g, k_b \in [0, 1]$... (2)

for Example for the Blue Chair,

$k_r = 0, k_g = 0, 0 < k_b < 1$

Consider the Normal \vec{n} and its
Angle formed by the incident light

PP.3. Ref. 1.

e, then
(x,y,z) canReference: Computer Graphics, C. K.
Pokorny, C. F. Gerald, pp. 514

$$\vec{n} \cdot \vec{r} = \|\vec{n}\| \|\vec{r}\| \cos \phi \quad \dots (3)$$

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

$$\phi = \cos^{-1} \left(\frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \right)$$

Instead, use $\cos \phi$.

$$I_d(x, y) \sim K_d \cos \phi$$

$$= K_d \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|}$$

PP.3. Ref. 1.

Note 1. Regarding Angle ϕ

$$I_r = K_{dr} \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \frac{1}{\|\vec{r}\|_z} \quad \dots (1.1)$$

where
e

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

$$I_g = K_{dg} \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \frac{1}{\|\vec{r}\|_z^2} \quad \dots (1.2)$$

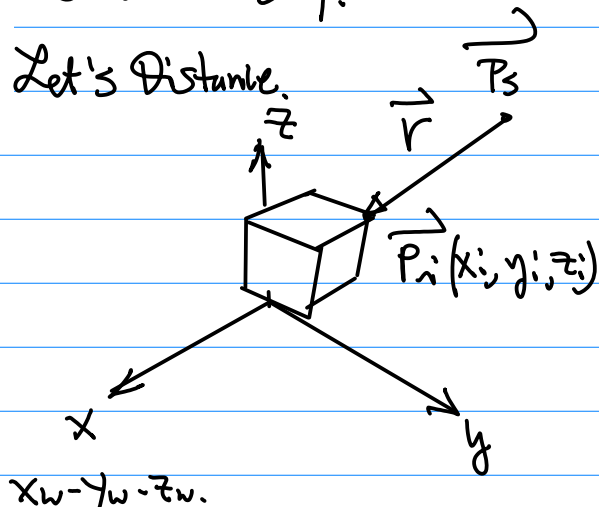
$$I_b = K_{db} \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \frac{1}{\|\vec{r}\|_b^2} \quad \dots (1.3)$$

intermediate Result $\dots (4)$
in Eqn (4).

Nov. 20 (Monday)

Ref:

2021F-101b-notes-cmpe240-2021-12-1 (1).pdf

Example: Add Distance factor
into Eqn (4), e.g., the shorter
distance gives stronger
Color intensity.

Use Vector Dot Product.

 \vec{n} : Normal Vector. \vec{r} : Ray Equation (Reformulated).
To Point to \vec{P}_s , to get rid of
Negative Sign.) $x_w - y_w - z_w$.

distance,

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

... (1)

$$I_{\text{diff}}(x, y, z) = k_d \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \cdot \frac{1}{\|\vec{r}\|^2}$$

... (5)

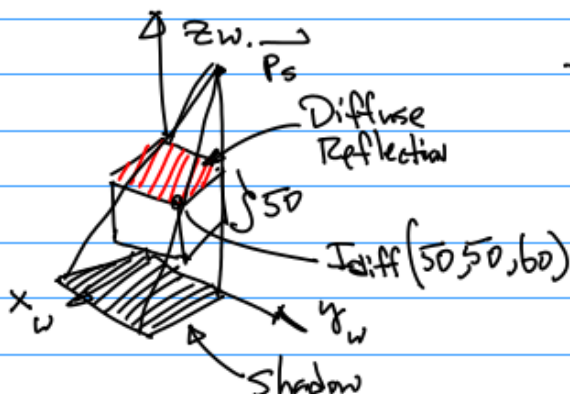
Note: from the ref. Hand Calculation.

Hence:

$$\begin{aligned} I_{\text{diff}} &= k_d \cdot \frac{1}{\|\vec{r}\|^2} \cdot \cos \alpha \\ &= k_d \cdot \frac{1}{\|\vec{r}\|^2} \cdot \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \end{aligned}$$

... (4)

Example: Given Conditions



$$\text{Cmp240(I)} \quad (50, 50, 60) - (40, 60, 120) = (10, 10, -60) \quad 50$$

$$\vec{r} = \vec{P_i} + \lambda (\vec{P_s} - \vec{P_i})$$

! for this calculation

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

$$= (50, 50, 60) + \lambda (-10, 10, 60)$$

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

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

$$\vec{n} \cdot \vec{r} = (0, 0, 1) \cdot (40, 60, 120) = 0 \times 40 + 0 \times 60$$

$$+ 1 \times 120 = 120$$

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

therefore...

Note 1. Diffuse Reflection

Result is very Small Due to

$\frac{1}{\|\vec{r}\|^2}$, in the coding, add offset, constant to make it

Visible, such as $z_{\text{into}} \in [0, 255]$.
Sample code is on the github.

Keyword "diffusecpp"
20/8F-116 ~

Now, Consider Engineering Implementation.

Condition:

4 Vertices' color has been Computed in x_w, y_w, z_w .

Now, Consider Computation After Perspective Projection

$I_{diff}(x_{i+1}, y_{i+1}, z_{i+1})$

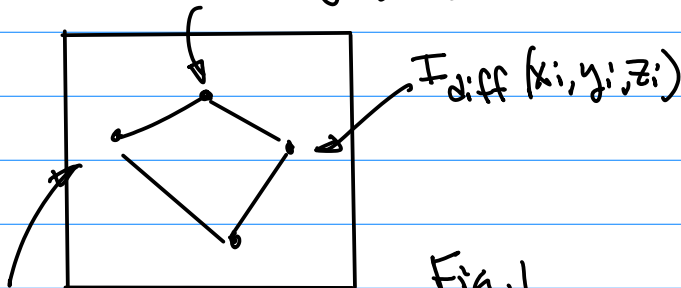


Fig.1

$I_{diff}(x_{i+2}, y_{i+2}, z_{i+2})$ ↓ Step 1. Colors @ all Find Boundaries

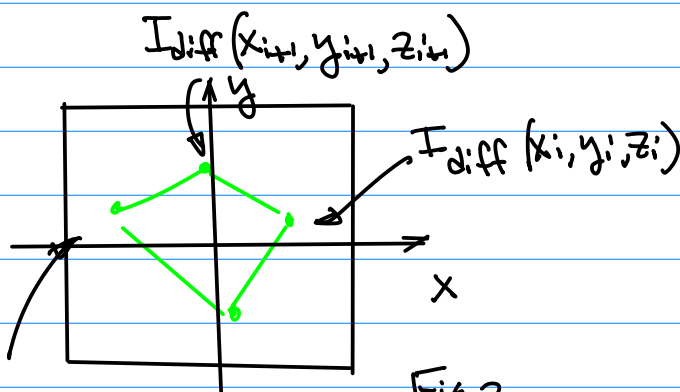


Fig.2

$I_{diff}(x_{i+2}, y_{i+2}, z_{i+2})$

↓ Step 2.

Find colors for all the interior Points

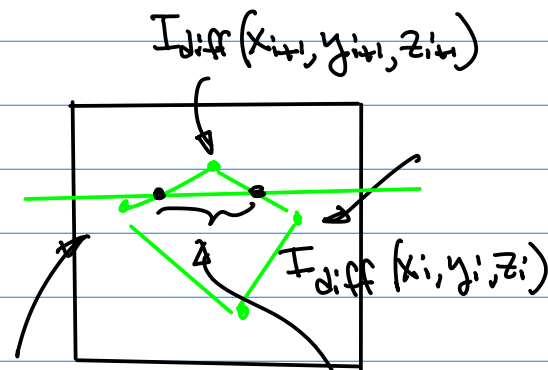


Fig.3

All Colors on this Line (Arbitrary Line)

Example: For Step 1.

Given 2 points of Diffuse Reflection colour, Find all colors on this Line.

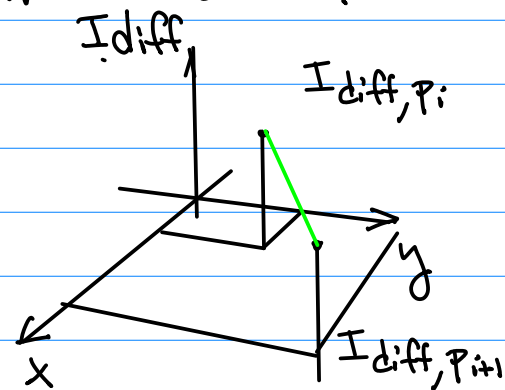
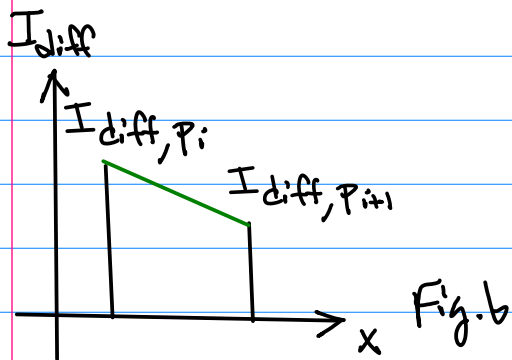
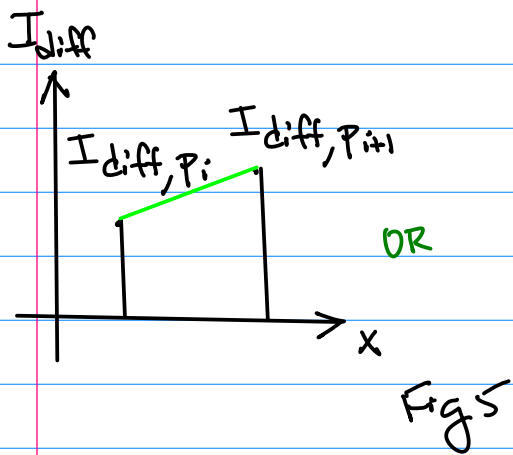


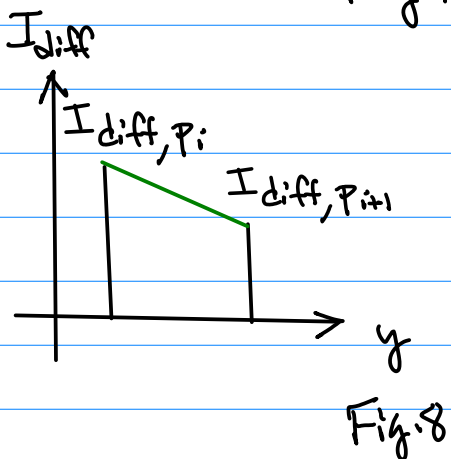
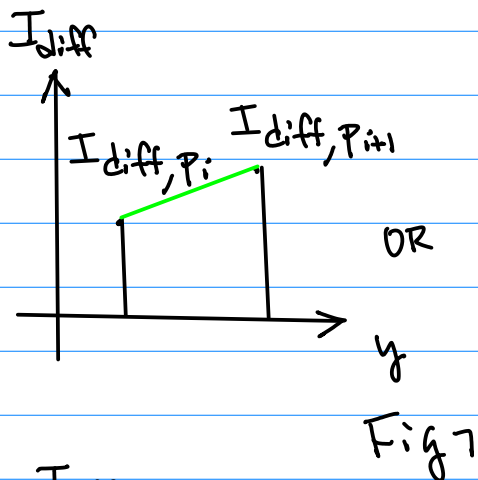
Fig.4

Fig.4

Projection to $x-I_{diff}$

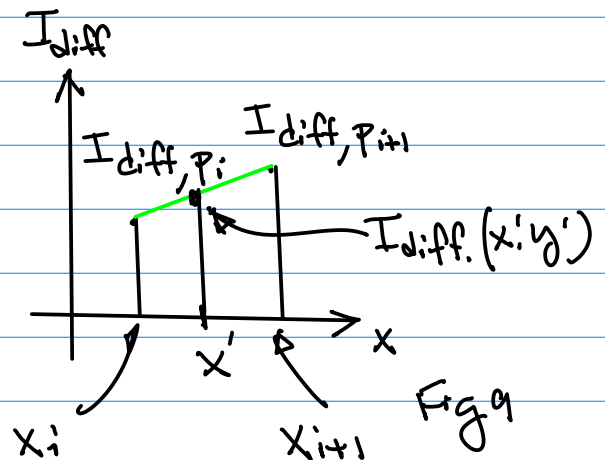


Similarly, Projection for y.

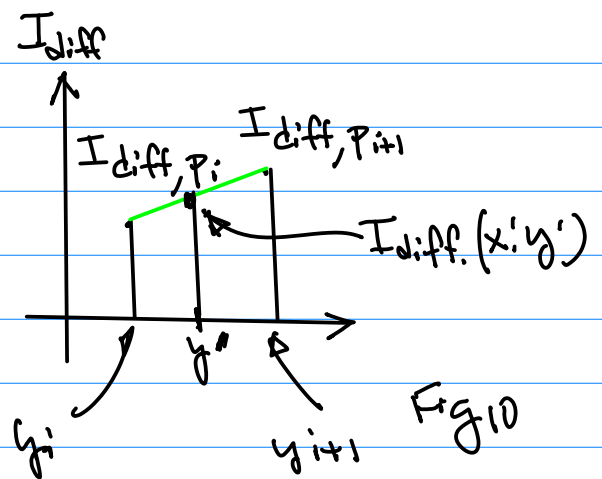


Use Linear Interpolation to find the Color of each and every point on this Line.

for example: for (x', y') find its color.



And for y.



To Combine them Final Dec. 13 (Wed)
12:15 - 2:30 PM.

$$\left(\begin{array}{c} \text{Diffuse} \\ \text{Reflection} \\ \text{at Pt } (x', y') \end{array} \right) = \left(\begin{array}{c} \text{Diffuse} \\ \text{Reflection} \\ \text{from } x \end{array} \right) + \left(\begin{array}{c} \text{Diffuse} \\ \text{Reflection} \\ \text{from } y \end{array} \right)$$

Z ... (b)

Projection Diffuse Reflection
Due Dec. 10 (Sun).

Nov. 27 (Monday).

Continuation on interpolation
for the Diffuse Reflection.

Ref: for the Diffuse Reflection.

2021F-101b-notes-cmpe240-2021-12-1 (1).pdf

From the Reference.

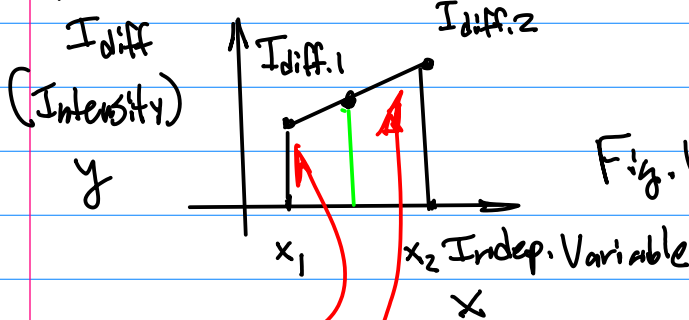


Fig. 1-a

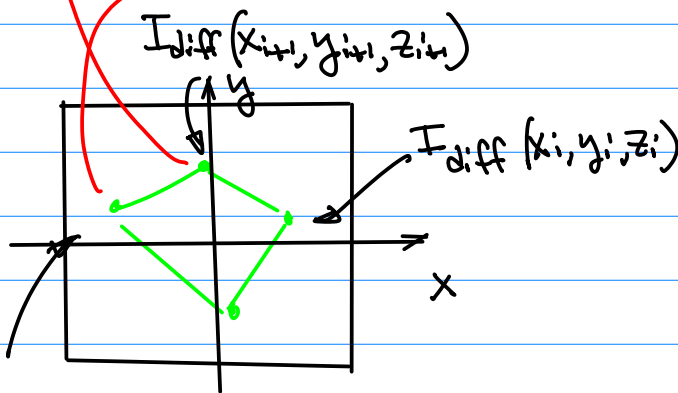


Fig. 1-b.

$I_{diff}(x_{i+2}, y_{i+2}, z_{i+2})$

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

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

Hand Calculation on pp. 61 (Ref)

From Eqn (1-b), Given (10, 0.78)
and (20, 0.5)

$$I_{diff} = \frac{0.78 - 0.5}{20 - 10} x - \frac{0.78 - 0.5}{20 - 10} \cdot 10 + 0.5$$

Notation
 $I_{diff, x}$

For $I_{diff, y}$, we have:

$$I_{diff, y} = \frac{0.78 - 0.5}{30 - 40} y - \frac{0.78 - 0.5}{30 - 40} \cdot 40 + 0.5$$

For $x=15, y=35$, substitute
them into the Above equations
to find the diffuse Reflection

$I_{diff, x}, I_{diff, y}$

So, the diffuse Reflection @ pt.

(15, 35) is .

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

$y = ax + b$
Slope offset

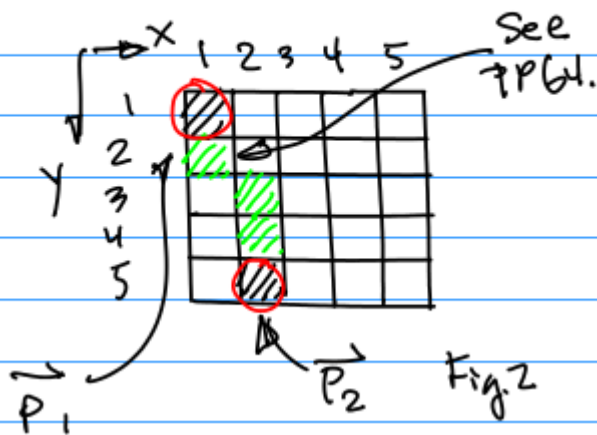
Therefore, Linear Interpolation
with respect to X variable

$$I_d(15, 10) = \frac{80 + 90}{2} = 85$$

Now, to find the actual points on Each Boundary Line, we will need DDA algorithm.
(Digital Differential Algorithm).

Example: Given 1. A finite Display device, 5x5 Below. 2. Starting pt $P_1(1,1)$, and Ending pt $P_2(2,5)$.

Digital Differential Algorithm. DDA



Find the Slope a .

$$a = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{5 - 1}{2 - 1} = 4$$

Note $|a| \geq 1$

To plot the line

$$y_{k+1} = ax_{k+1} + b \quad \dots (3)$$

where $\rightarrow b = -\frac{y_k - y_{k+1}}{x_k - x_{k+1}} x_{k+1} + y_{k+1} = -4 \times 2 + 5 = -3$

Verify it. Let $x=1$, find
 $y = ax + b = 4 \cdot 1 - 3 = 1$

To plot the Line.

$$x_k = 1,$$

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

hence

$$y_{k+1} = a \cdot x_{k+1} + b \\ = 4 \cdot 2 - 3 = 5$$

Swap the x and y

$$\frac{1}{a} y_{k+1} = x_{k+1} + b/a$$

Hence $|\frac{1}{a}| \leq 1$

$$x_{k+1} = \frac{1}{a} y_{k+1} - \frac{b}{a} \quad \dots (4)$$

This will fix the gap problem.

To verify $y_k = 1$

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

$$x_{k+1} = \frac{1}{4} y_{k+1} + b = \frac{1}{4} \times 2 + 3/4 = 5/4 \\ \approx 1$$

Summary: Compute 4 Anchor Points Diffuse Reflection, then perspective projection for these 4 points. After that use Bi-Linear Interpolation + DDA Algorithm to find all Boundary Color.

One more time as illustrated Below to find the color of the interior points.

Nov. 29 (Wed).

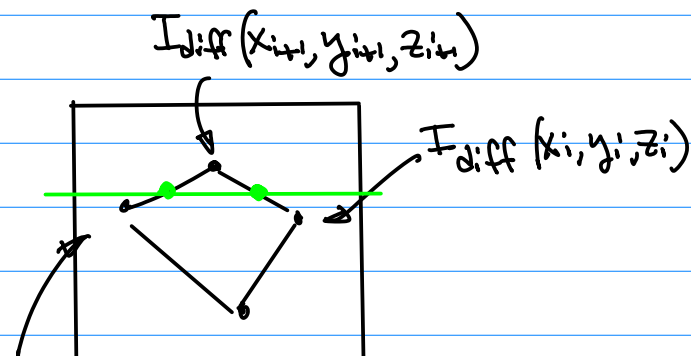
Remark On the Diffuse Reflection Computation.

Step 1. Carry out the Diffuse Reflection Computation for the vertices of the Cube in x_w, y_w, z_w (World Coordinate)

Step 2. Perform Transformation pipeline Computation, so the locations for the 4 vertices now are defined on the Display Device. So are their Color Intensities from Step 1. Plot these 4 pixels

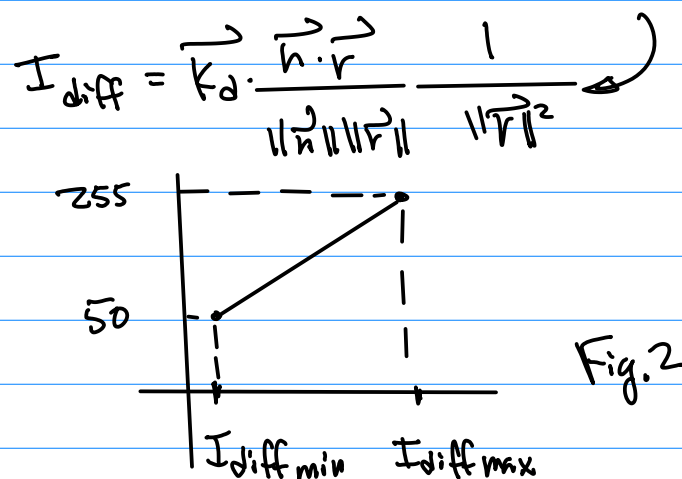
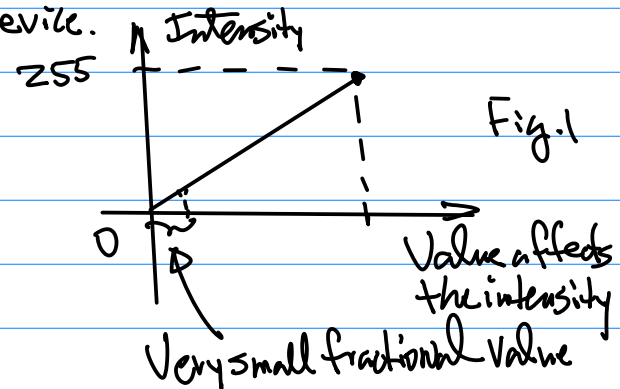
Step 3. Take Care of the color then on the 4 Boundary lines by using Bi-Linear Interpolation where the exact location of the color is defined by DDA algorithm.

Step 4. Apply Linear Interpolation



$I_{diff}(x_{i+2}, y_{i+2}, z_{i+2})$

Implementation Consideration 1.
Dynamic Range of A LCD Display Device.

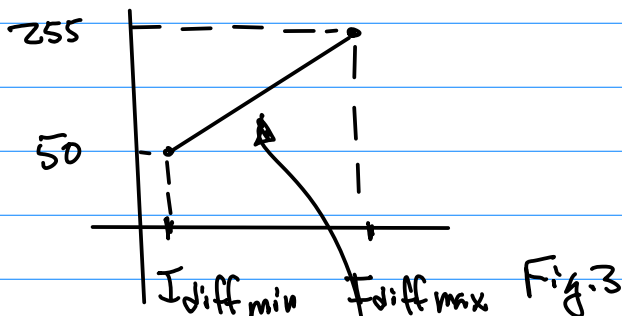


$$I_{diff} = K_d \cdot \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|} \frac{1}{\|\vec{r}\|^2}$$

Note $I_{diff, min}$ is the point that is the farthest away from \vec{P}_s (point light).

$I_{diff, max}$ corresponds to the pt. that gives the shortest distance to \vec{P}_s .

Let's make the $I_{diff, min} = 50$ for display, And $I_{diff, max} = 255$ for the display.



Using $y = ax + b$ Linear interpolation.

Where $y = \text{Intensity Value}$

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

... (1b)

Where $a = \frac{y_1 - y_2}{x_1 - x_2}$ and $b = -\frac{y_1 - y_2}{x_1 - x_2} x_2 + y_2$

... (1-b) ... (1-c)

In Eqn(1), (ta) , $(1-b)$

x for the Diffuse Reflection Computed @ Step 4.

y for the Display Device Intensity level.

githwb
2018F-11b-11 ~

CMPE240

F2023

64

```
191 //-----diffuse reflection-----*
192 pt_diffuse diffuse; //diffuse.r[3]
193
194 //-----reflectivity coefficient-----*
195 #define Kdr 0.8
196 #define Kdg 0.0
197 #define Kdb 0.0
198
199 // define additional pts to find diffuse reflection
200 //world.X[49] = world.X[45] + lambda_2*(world.X[45] - world.X[6]);
201
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::endl;
209 }
```

Note1. $\vec{K_d}$ Reflectivity

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

Note2. distance, Ray Equation \vec{r} , then $\|\vec{r}\|^2$, No Need for "Sqrt".

Note3. $\cos\theta = \frac{\vec{n} \cdot \vec{r}}{\|\vec{n}\| \|\vec{r}\|}$

```
219 //-----compute angle-----*
220 float angle[UpperBD], tmp_dotProd[UpperBD], tmp_mag_dotProd[UpperBD];
221
222 for (int i=48; i<=49; i++){
223     /*
224     tmp_dotProd[i] = (world.X[i]-world.X[45])*world.X[47]+ //...[47] for normal vector
225                     (world.Y[i]-world.Y[45])*world.Y[47]+ //...[45] for pt light source
226                     (world.Z[i]-world.Z[45])*world.Z[47];
227
228     */
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)+ // [45] pt light source
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 }
```

$\cos(\theta)$.

Note 1. Diffuse Reflection for r, g, b.

```

240 //compute color intensity
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 }
245

```

Note: 1st DBA Algorithm.

Sample code on the github

2018F-117 ~

Note: Check the stop $|a| \geq 1$. if Not Increment

```

--
16 double X1, Y1, X2, Y2;
17
18 float round_value(float v)
19 {
20     return floor(v + 0.5);
21 }
22 void LineDDA(void)
23 {
24     double dx=(X2-X1);
25     double dy=(Y2-Y1);
26     double steps;
27     float xInc,yInc,x=X1,y=Y1;
28     /* Find out whether to increment x or y */
29     steps=(abs(dx)>abs(dy))?(abs(dx)):(abs(dy));
30     xInc=dx/(float)steps;
31     yInc=dy/(float)steps;
32

```

$X_{k+1} = X_k + 1$
 o/w (otherwise), increment

$Y_{k+1} = Y_k + 1$

$y = ax + b$ without
 multiplication.
 multiplier

Dec. 4th (Monday).

Linear Decoration Algorithm

Ref. github.

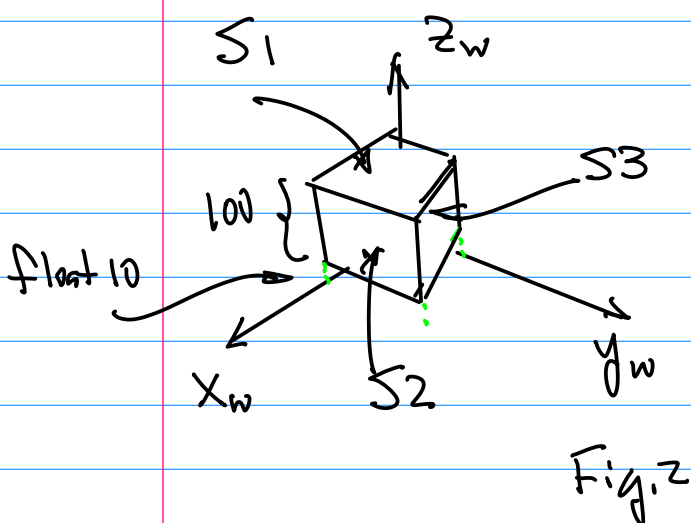
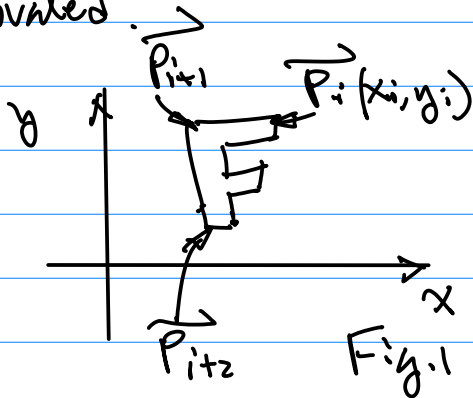


Objectives: To use 2D Graphics
Pattern

$\{\vec{P}_i(x_i, y_i), i=1, 2, \dots, N\}$
to Decorate 3D plane Surface(s).

Therefore, it Requires Linear
Decoration.

Example: Suppose 2D vector
Graphics is given in Fig.1
And 3D Cube is defined in
 $x_w-y_w-z_w$ in Fig.2 to Be
decorated.



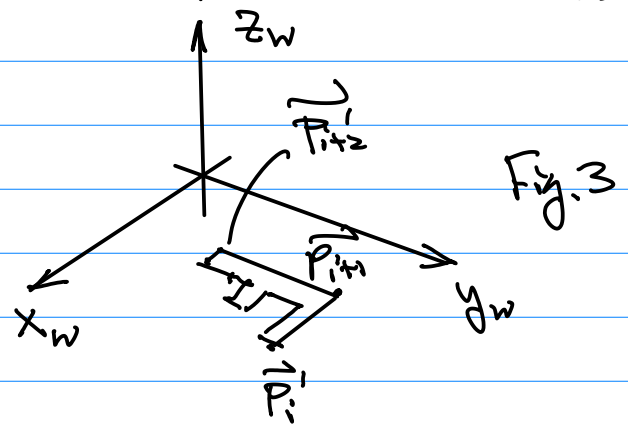
Decorate plane Surface S_1 .

The process of Doing this Decoration:

Step 1. Define 2D Pattern $\{\vec{P}_i(x_i, y_i) | i=1, 2, \dots, N\}$ into 3D $x_w-y_w-z_w$ Space.

$$\{\vec{P}_i(x_i, y_i)\} \rightarrow \{\vec{P}_i(x'_i, y'_i, z'_i)\}$$

$$\text{Where } x'_i = x_i, y'_i = y_i, z'_i = 0 \dots (1)$$



Step 2. Use Combination of Translations
and Rotations, to move the
Pattern in Fig.3 to match the
Right location of S_1 in Fig.2

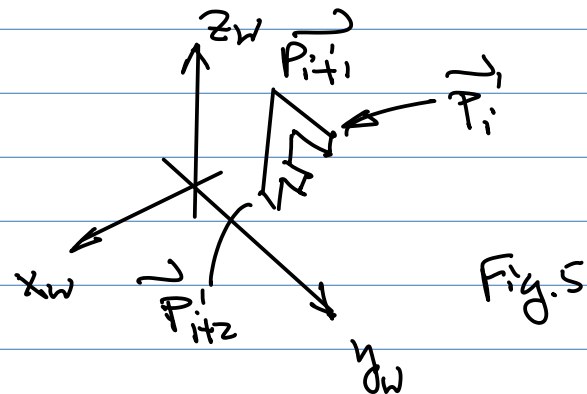
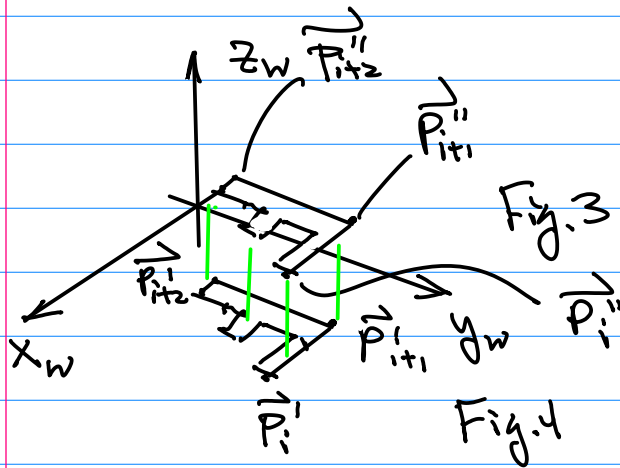
Therefore Eqn(1):

$$z''_i = (\text{Size of the Cube}) + \text{Elevation} = C + 10$$

So

$$y_i' = x_i$$

$$z_i' = y_i$$



Step 3: To plot the points.

After Before

$$\begin{cases} x_i'' = x_i & \dots (2-1) \\ y_i'' = y_i & \dots (2-2) \\ z_i'' = C + D & \dots (2-3) \end{cases}$$

Now, for S_2 : (y_w - z_w plane)
Match the indep. Variable x in
Fig. 3 to the indep. Variable
in Fig. 2. y . e.g.

After. $y_i' = x_i$ Before

Then, similarly, match the
functions.

Hence, before
 $z_i' = y_i$

move to the right location.

$$x_i' = C$$

Hence, All together.

$$\begin{cases} x_i'' = C & \dots (3-1) \\ y_i'' = x_i & \dots (3-2) \\ z_i'' = y_i & \dots (3-3) \end{cases}$$

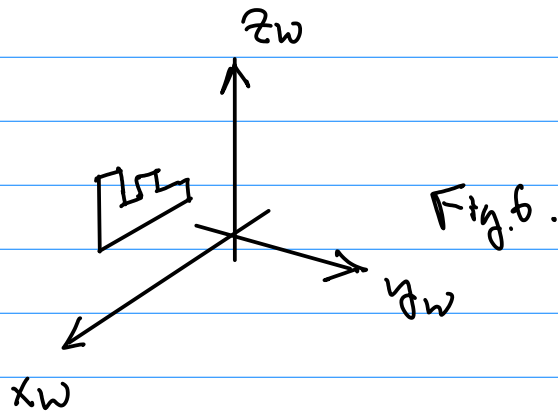
Similarly for S_3 .

Indep: z_w . Function: x_w

(Because the Right Hand
System, How the Vector
Cross Product Defines y_w -axis).

After Before

$$\begin{aligned} z_i' &= x_i \quad (\text{Indep.}) \\ x_i' &= y_i \quad (\text{Func}) \end{aligned}$$



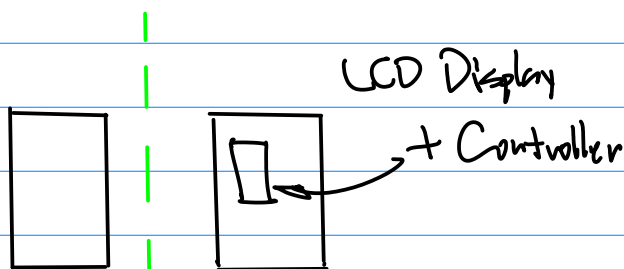
Match S_3 Surface
By Translation, so

$$y_i'' = C$$

Therefore, After Before

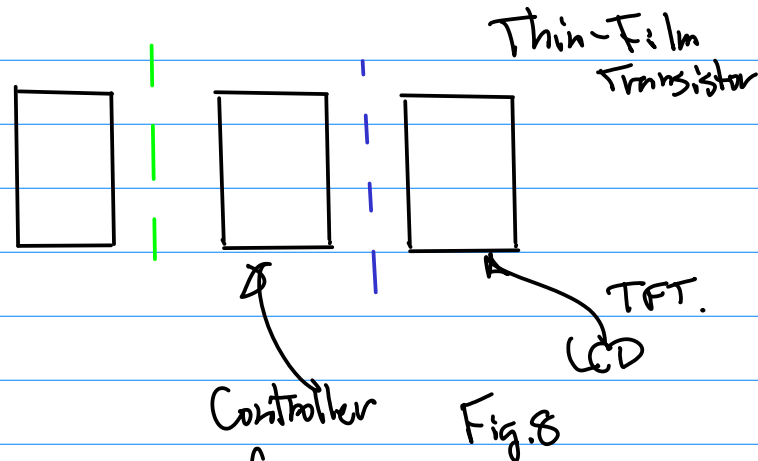
$$\begin{cases} z_i'' = x_i \text{ (Indep)} \dots (4-1) \\ x_i'' = y_i \text{ (Function)} \dots (4-2) \\ y_i'' = C \dots (4-3) \end{cases}$$

The Hardware Controller Design.



LPC1114
or
LPC1114
or
CPU.

Fig. 7



"Green" Interface is done.

CPU v.s. LCD module (with a Controller), for SPI "3+1"

$\begin{cases} \text{MOSI} \\ \text{MISO} \\ \text{SCK} \\ \text{SS/SEL} \end{cases}$

TFT LCD Device:

$\begin{cases} \text{Addr. Bus} \rightarrow \text{Sync.} \\ \text{Data Bus} \\ \text{Control Signal} \end{cases}$

$\begin{cases} \text{Frame} \\ \text{Line} \\ \text{Data} \end{cases}$

Compezo
F2023.

691