CmpE240
F2023

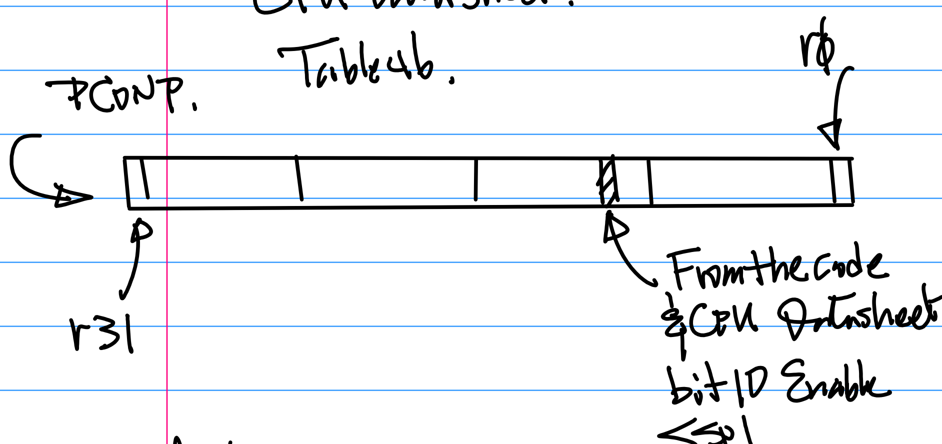
51

Nov. 6th (Monday).
Note1. Midterm Exam is scheduled on Next Monday.
Review Session in-class is scheduled on Wednesday.

Today's Topics:

1° SSPInit Code.
2° Hardware Design for LCD/CPU Interface Design.

Example: Line 210-246.
Line 224 Naming Convention
LPC_SC -> PCONP
Section 4.8.5. pp61
CPU Datasheet.
Table46.



Line 227.
LPC_SC -> PCLKSEL

Line 229
Line 230
LPC_SC -> PINSEL0.

pp.58, Table 42.
Section 4.7.3. pp57.
Note: Draw a 32 Bit SPR illustration.
Connect to CPU Datasheet
(Tables for the SPR).
The memory Bank holding this SPR.
Code Implementation.

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

PCONP -> PCKSEL -> PINSEL0

Line 233/234 CS (Chip select: e.g. Select/Enable my LCD Display module)
Line 238
SSP1 CR0.
pp.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

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

then, Coding.

CMPE240

F2023

Homework: Due Nov. 19th (Sunday).

1° 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 $\vec{P}_s(x_s, y_s, z_s) = (-5, 50, 250)$,

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.

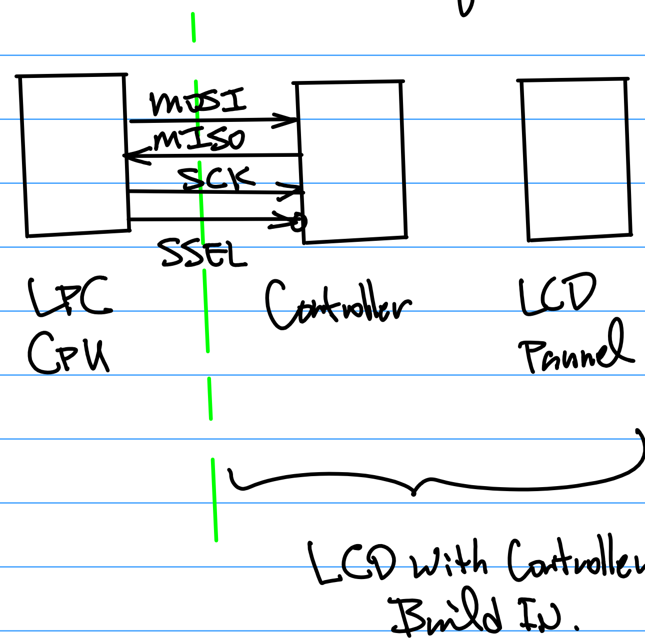
c) Draw the shadow first Before Drawing the Cube.

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.



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

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

Nov. 8 (Wed).

Note 1. Midterm Exam is Next Monday.

a) Please Bring your Prototype System;

b) Bring Blank Papers for the exam hand Calculation;

CmpE240

F2023

Homework 53

c) use "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.

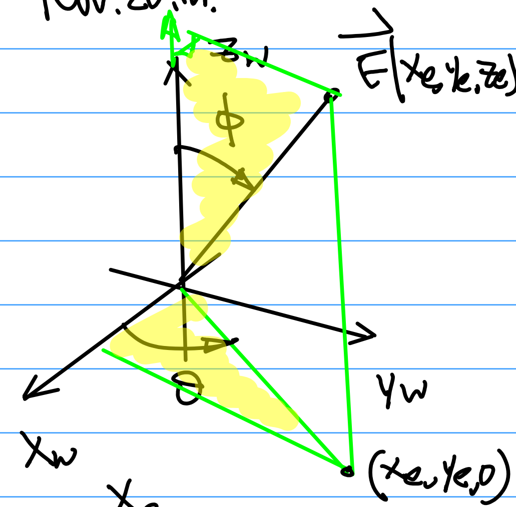
FirstName_LastName_SID_Cmpe240_mid.zip

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

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



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

$\sin\theta$

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

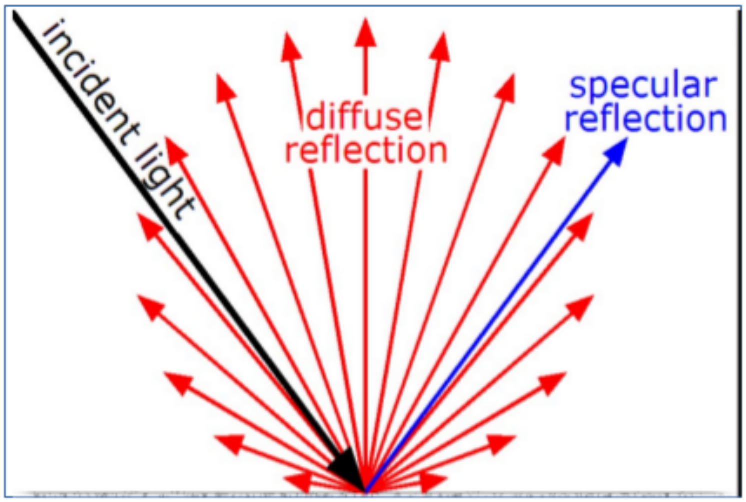
CmpE240 F2023 Part II

Nov. 15 (Wed).

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

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

Diffuse Reflection



https://en.wikipedia.org/wiki/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*r), where r is bing the distance from the light source to the surface.

Specular vs. 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.

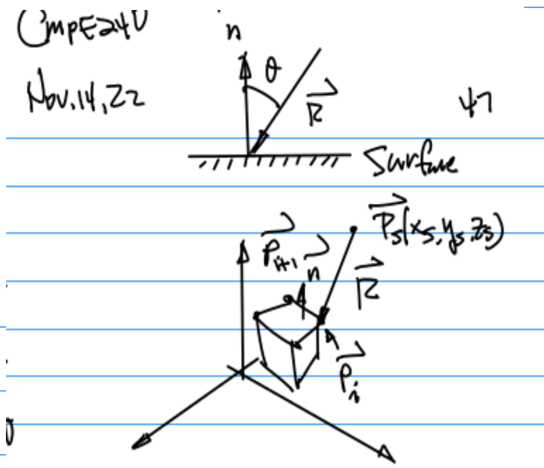


Fig. 1.

Ref2: on the Class github.

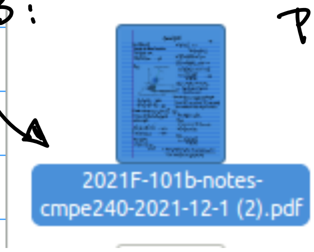
2022F-101-notes-cmpe240-2022-11-30 (1).pdf

Note:

1° Definition: PP.54. Reflection Uniformally in All different Directions.

Example: Background on Diffuse Reflection.

Ref3: Pp54.



2021F-101b-notes-cmpe240-2021-12-1 (2).pdf

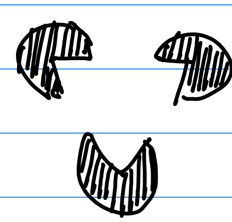


Fig. 2

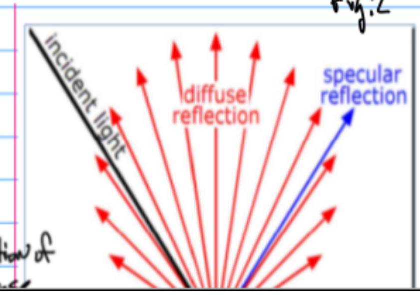


Illustration of Diffuse

Brief Introduction, And 3 Lighting models.

Three models. see Ref3. 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. P81.

Reflection Uniformly Reflects the Incoming light in All Different Direction.

Note1. $\vec{P_s}(x_s, y_s, z_s)$ Incident Light, White Color. for example $r=g=b=255$

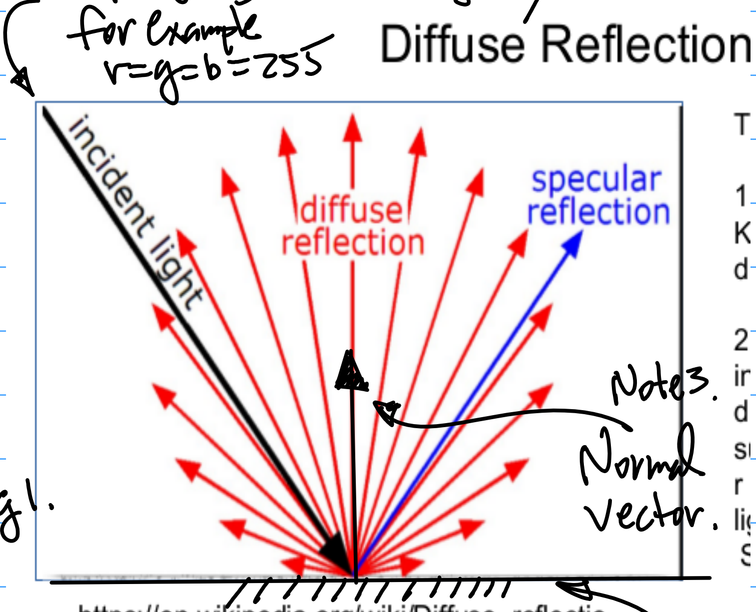


Fig1.

https://en.wikipedia.org/wiki/Diffuse_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., diffuse coefficients; ... (1)

diffuse Reflection

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

Where K_r : Reflectivity for red

K_g : " " green

K_b : " " blue

Normalized, so

$$K_r \in [0,1], \quad K_g, K_b \in [0,1] \quad \ldots (2)$$

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

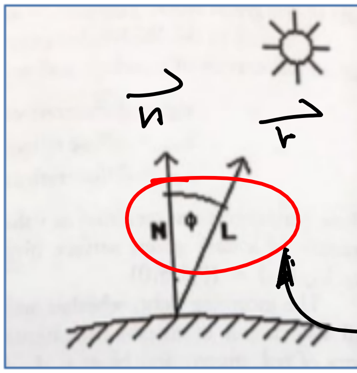
CMPE240

F2023.

56

PP.3. Ref. 1.

e, then
(x,y,z) can



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

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

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

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

Instead, use $\cos\phi$.

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

$$= K_d \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \quad \ldots (4)$$

intermediate Result in Eqn (4).

TP.3.Ref.1. Note 1. Regarding Angle ϕ

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

where

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

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

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

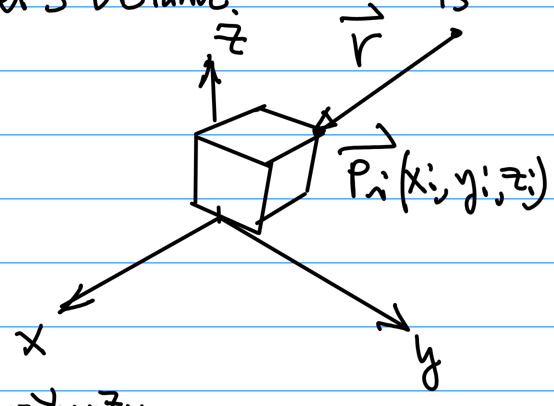
Nov. 20 (Monday)

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.

Let's Distance.



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.)

distance,

$$\|\vec{r}\| = \sqrt{(x_s-x_i)^2+(y_s-y_i)^2+(z_s-z_i)^2} \quad \dots (1)$$

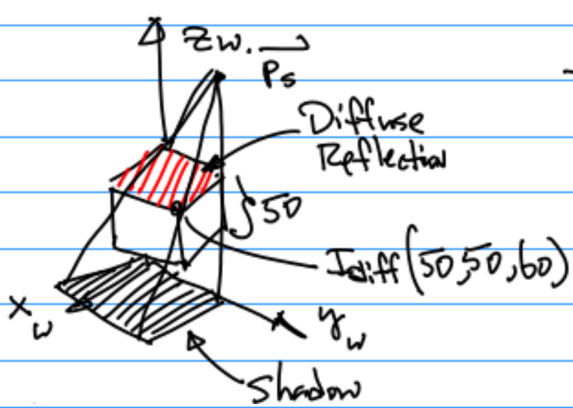
$$I_{diff}(x,y,z) = K_d \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|} \cdot \frac{1}{\|\vec{r}\|^2} \quad \dots (5)$$

Note: from the Ref. Hand Calculation.

Hence:

$$\boxed{\begin{aligned}\vec{I}_{diff} &= K_d \cdot \frac{1}{\|\vec{r}\|^2} \cdot \cos\alpha \\ &= K_d \cdot \frac{1}{\|\vec{r}\|^2} \cdot \frac{\vec{n}\cdot\vec{r}}{\|\vec{n}\|\|\vec{r}\|}\end{aligned}} \quad \dots (4)$$

Example: Given Conditions



CmpE240(II) $(50,50,60)-(40,60,120) = (10,-10,-60)$ 57

$\vec{r} = \vec{P_i} + \lambda(\vec{P_s} - \vec{P_i})$! for this Calculation

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

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

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

$= (40,60,120)$

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

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

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 20 to 40. $\in [0, 255]$.
Sample code is on the github.

Keyword " diffuse ... .cpp"
2018F-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

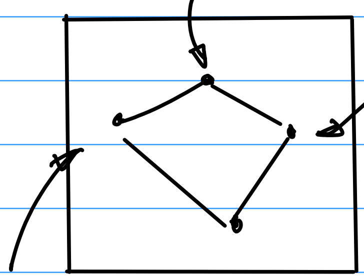


Fig.1

↓ Step 1. Colors @ all, Find Boundaries

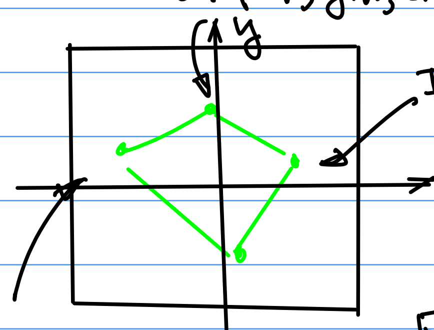


Fig.2

↓ Step 2.

Find colors for all the interior Points

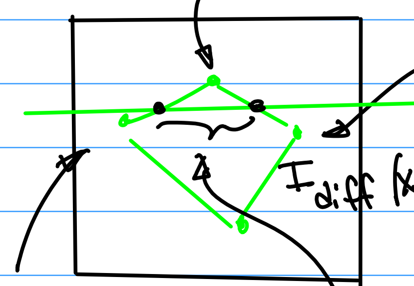


Fig.3

All Colors on this line (Arbitrary line)

Example: For Step 1.

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

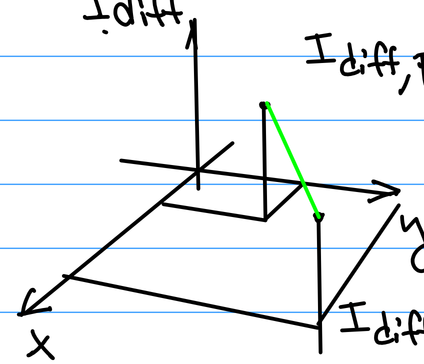


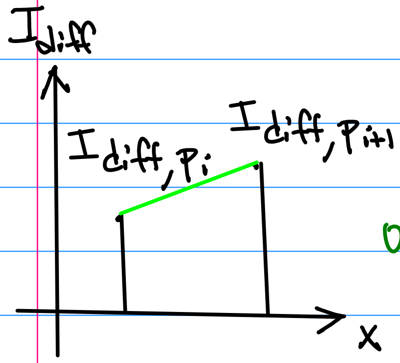
Fig.4

Fig.4

Projection to x–Idiff

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

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



OR

Fig 5

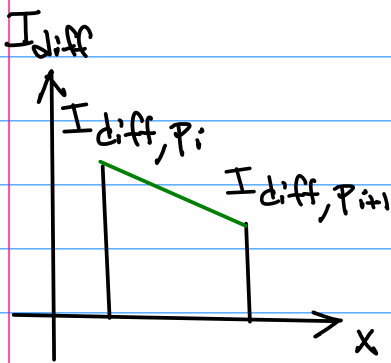


Fig. 6

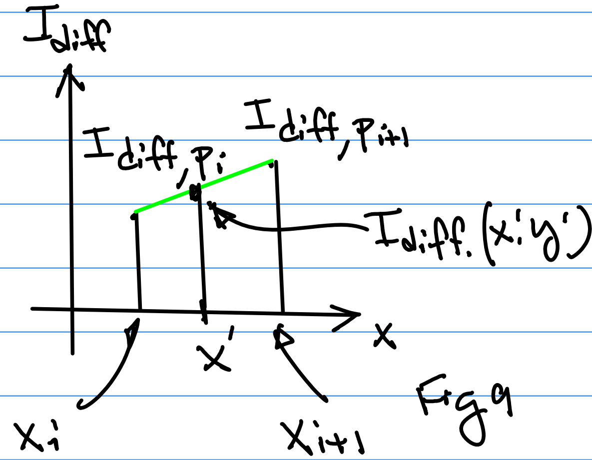
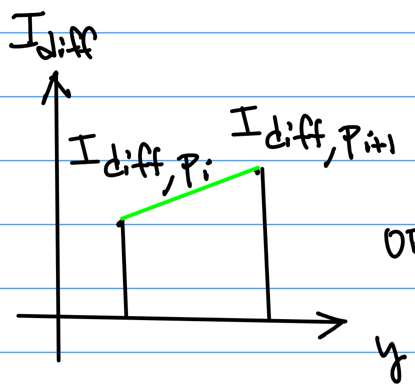


Fig 9

Similarly, Projection for y.

And for y.



OR

Fig 7

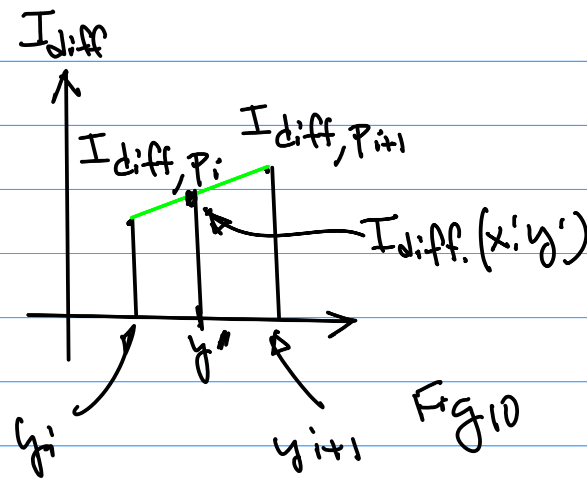


Fig 10

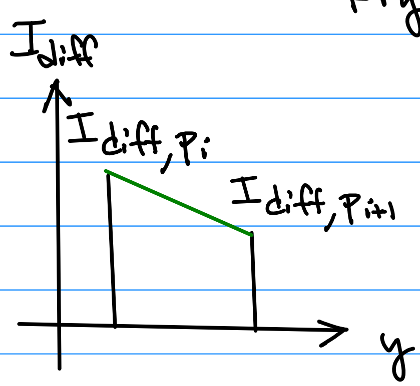


Fig. 8

To Combine them

Final Dec. 13 (Wed) 12:15 – 2:30 pm.

$$\left(\begin{array}{c}\text{Diffuse}\\ \text{Reflection}\\ \text{@ pt } (x', y')\end{array}\right) = \frac{\left(\begin{array}{c}\text{Diffuse}\\ \text{Reflection}\\ \text{from } x\end{array}\right) + \left(\begin{array}{c}\text{Diffuse}\\ \text{Reflection}\\ \text{from } y\end{array}\right)}{2} \quad \ldots (6)$$

CMPE240
F2023

Project on 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.

I_{diff} (Intensity) y

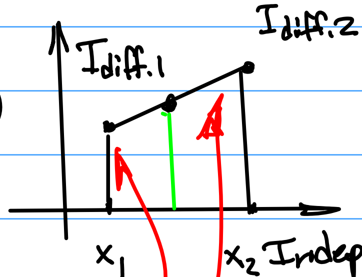


Fig. 1-a

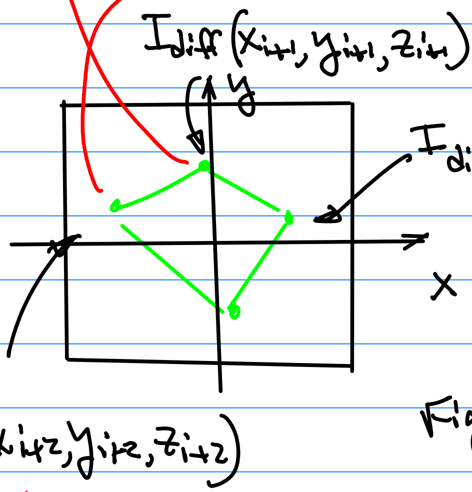
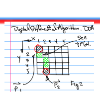


Fig. 1b.

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

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



$y = ax + b$

slop offset

60/

Hard Calculation on pp. 61 (Ref)

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

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

$I_{diff,x}$ Notation

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

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

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.

$$I_{diff} = \frac{1}{2}\left(I_{diff,x} + I_{diff,y}\right) \quad \ldots (2)$$

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 finit Display device, 5×5 Below. 2. Starting pt $\vec{P_1}(1,1)$, and Ending pt $\vec{P_2}(2,5)$.

Digital Differential Algorithm. DDA

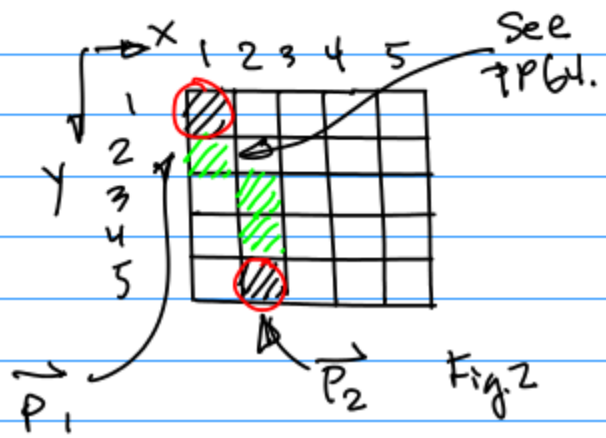


Fig.2

Find the Slop 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

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

where

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

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 $\left|\frac{1}{a}\right| \leq 1$

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

This will fix the gap problem.

To verify $y_k = 1$

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

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

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

Nov. 29 (Wed).

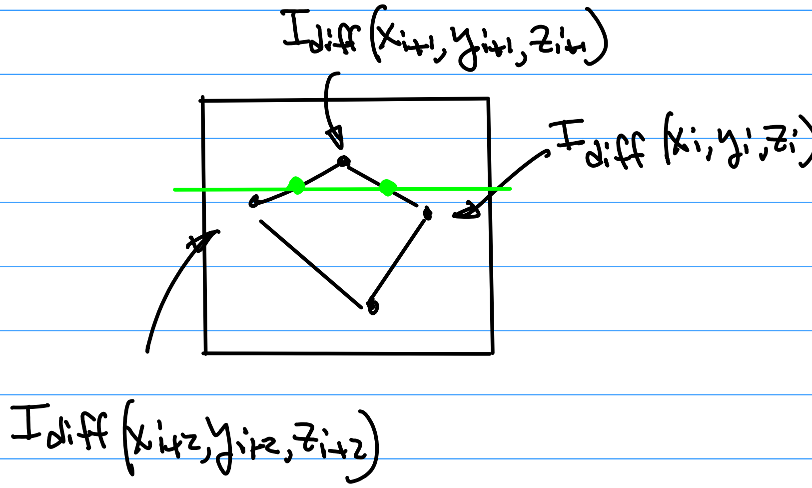
Remark On the Diffuse Reflection Computation.

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



Implementation Consideration 1.

Dynamic Range of A LCD Display Device.

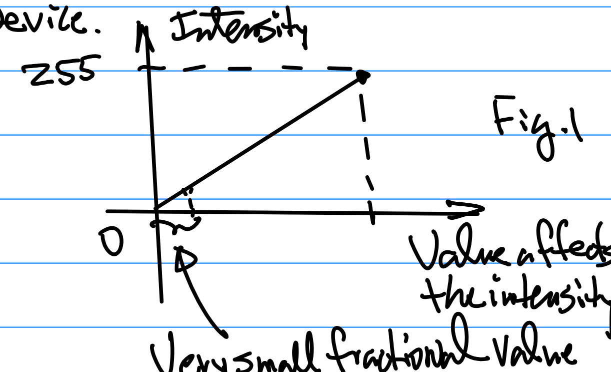


Fig. 1

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

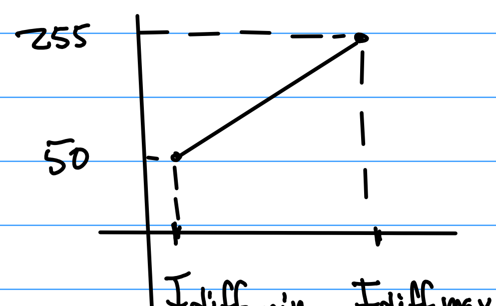


Fig. 2

Note $I_{diff,min}$ ~~is~~ the point that is the ~~corresponds to~~ fartest Away from P_s (point Light).

$I_{diff,max}$ Corresponds to the pt. that gives the shortest distance to P_s.

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

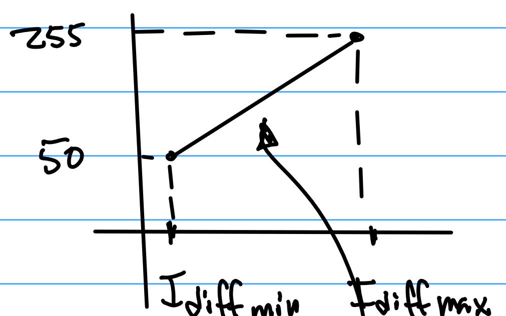


Fig.3

Using $y = ax + b$ Linear interpolation.

Where y = Intensity Value

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

$\ldots (1b)$

Where $a = \dfrac{y_1 - y_2}{x_1 - x_2}$ $\ldots$ (1-b) and $b = -\dfrac{y_1 - y_2}{x_1 - x_2} x_2 + y_2$ $\ldots$ (1-c)

In Eqn(1), (1-a), (1-b)

x for the Diffuse Reflection Computed @ Step 4.

y for the Display Device Intensity level.

github
2018F-116-11 ~

```
        //-------------diffuse reflection-----------*
        pt_diffuse  diffuse;   //diffuse.r[3]

        //-------reflectivity coefficient-----------*
        #define      Kdr      0.8
        #define      Kdg      0.0
        #define      Kdb      0.0

        // define additional pts to find diffuse reflection
        //world.X[49] = world.X[45] + lambda_2*(world.X[45] - world.X[6]);

        //--------compute distance-------------------*
        float distance[UpperBD];
        for (int i=48; i<=49; i++) {
        distance[i] = sqrt(pow((world.X[i]-world.X[45]),2)+                   //intersect pt p7
                           pow((world.Y[i]-world.Y[45]),2)+
                           pow((world.X[i]-world.X[45]),2) );
        //std::cout << "distance[i]  " << distance[i] << std::endl;
        }
```

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

```
        //--------compute angle---------------------*
        float angle[UpperBD], tmp_dotProd[UpperBD], tmp_mag_dotProd[UpperBD];

        for (int i=48; i<=49; i++){
    /*
        tmp_dotProd[i] =  (world.X[i]-world.X[45])*world.X[47]+    //...[47] for normal vector
                          (world.Y[i]-world.Y[45])*world.Y[47]+    //...[45] for pt light source
                          (world.Z[i]-world.Z[45])*world.Z[47];

    */
        tmp_dotProd[i] = world.Z[i]-world.Z[45];
        std::cout << " tmp_dotProd[i]  " <<  tmp_dotProd[i] << std::endl;

        tmp_mag_dotProd[i] = sqrt(pow((world.X[i]-world.X[45]),2)+         //[45] pt light source
                                  pow((world.Y[i]-world.Y[45]),2)+
                                  pow((world.Z[i]-world.Z[45]),2) );
        std::cout << " tmp_mag_dotProd[i]  1 " <<  tmp_mag_dotProd[i] << std::endl;

        angle[i] = tmp_dotProd[i]/ tmp_mag_dotProd[i];
        std::cout << "angle[i]  " << angle[i] << std::endl;
```

$\cos(\theta)$.

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

```
//compute color intensity
diffuse.r[i] = Kdr *  angle[i] /  pow(distance[i],2) ;
diffuse.g[i] = Kdg *  angle[i] /  pow(distance[i],2) ;
diffuse.b[i] = Kdb *  angle[i] /  pow(distance[i],2) ;
}
```

Note: 1° DDA Algorithm.

Sample code on the github

2018F-117 ~

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

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

o/w (otherwise), increment

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

$y = ax + b$ without multiplication.

multiplier

```
double X1, Y1, X2, Y2;

float round_value(float v)
{
  return floor(v + 0.5);
}
void LineDDA(void)
{
  double dx=(X2-X1);
  double dy=(Y2-Y1);
  double steps;
  float xInc,yInc,x=X1,y=Y1;
  /* Find out whether to increment x or y */
  steps=(abs(dx)>abs(dy))?(abs(dx)):(abs(dy));
  xInc=dx/(float)steps;
  yInc=dy/(float)steps;

```

Dec. 4th (Monday).

Linear Decoration Algorithm → Ref. github.

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

Objectives: To use 2D Graphics Vector Pattern
$\{\vec{P_i}(x_i, y_i),\ i = 1, 2, \ldots, N\}$
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.

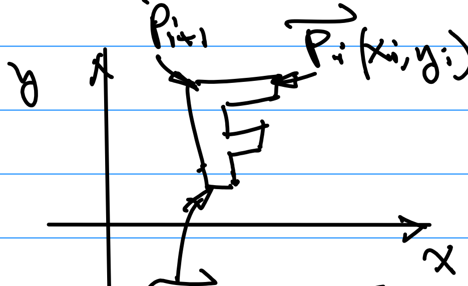


Fig.1

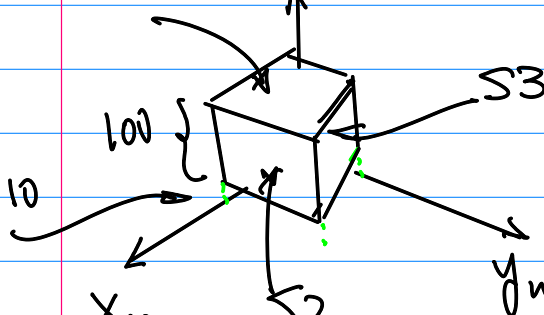


Fig.2

Decorate plane Surface S_1.
The process of Doing this Decoration:
Step1. Define 2D Pattern $\{\vec{P_i}(x_i, y_i) \mid i = 1, 2, \ldots, N\}$ into 3D X_w-Y_w-Z_w Space.

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

where $x_i' = x_i$, $y_i' = y_i$
$z_i' = 0$ ... (1)

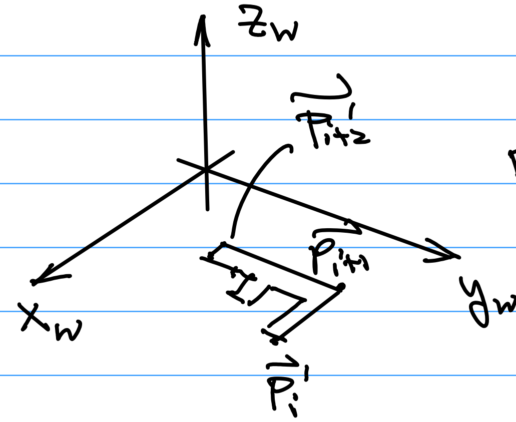


Fig.3

Step2. Use Combination of Translations and Rotations to move the Pattern in Fig.3 to match the Right location of S1 in Fig.2

Therefor Eqn(1):

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

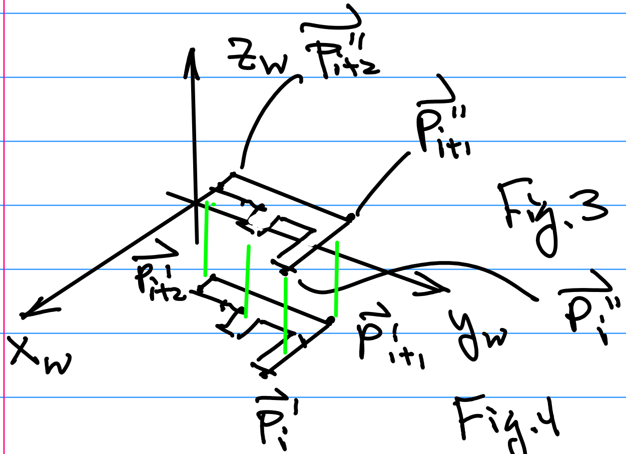


Fig.3

Fig.4

So

$$y_i' = x_i$$
$$z_i' = y_i$$

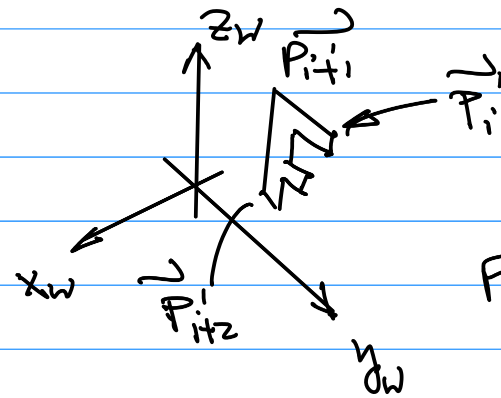


Fig.5

Step3: To plot the points.

After Before

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

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. Before

$$y_i' = x_i$$

Then, similarly, match the functions.

Hence, before

$$z_i' = y_i$$

Move to the right location.

$$x_i' = C$$

Hence, All together.

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

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

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

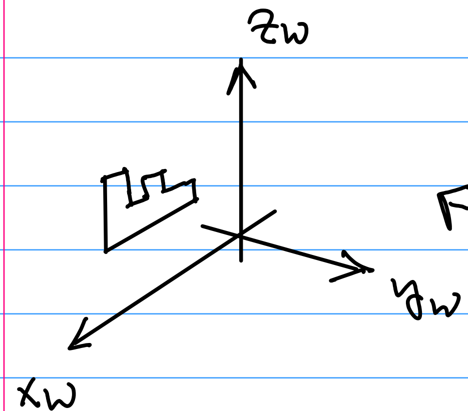


Fig.6.

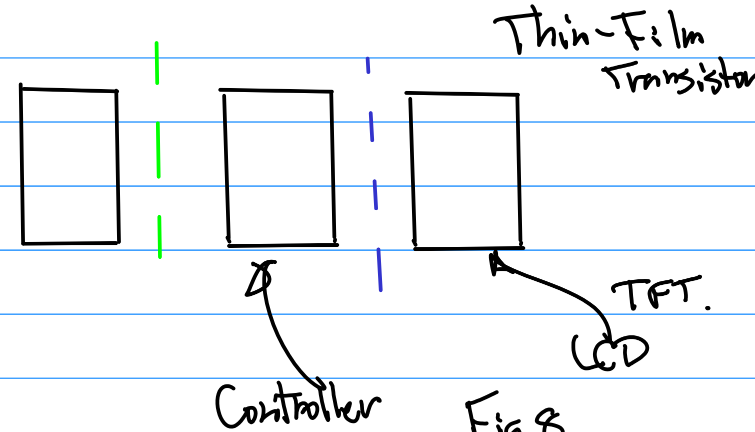


Fig.8

Match S_3 Surface By Translation, so

$y_i'' = C$

Therefore, After Before

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

"Green" Interface is done.

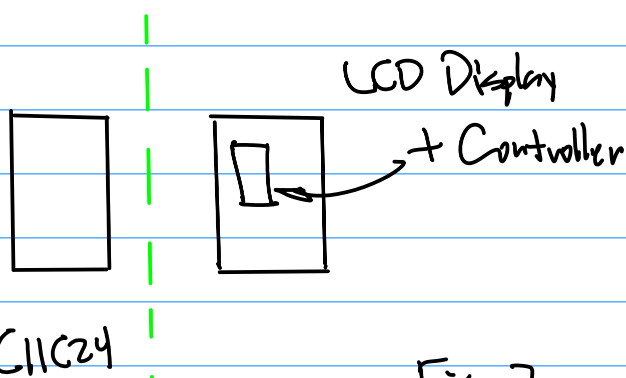
CPU v.s. LCD module (with a Controller), for SPI "3+1"

- MOSI
- MISO
- SCK
- SSEL

TFT LCD Device:

- Addr. Bus → Sync. { Frame, Line, Data }
- Data Bus
- Control Signal

The hardware Controller Design.



LPC11C24
OR
LPC1769
OR
CPU.

Fig. 7

Dec. 6 (Wed).

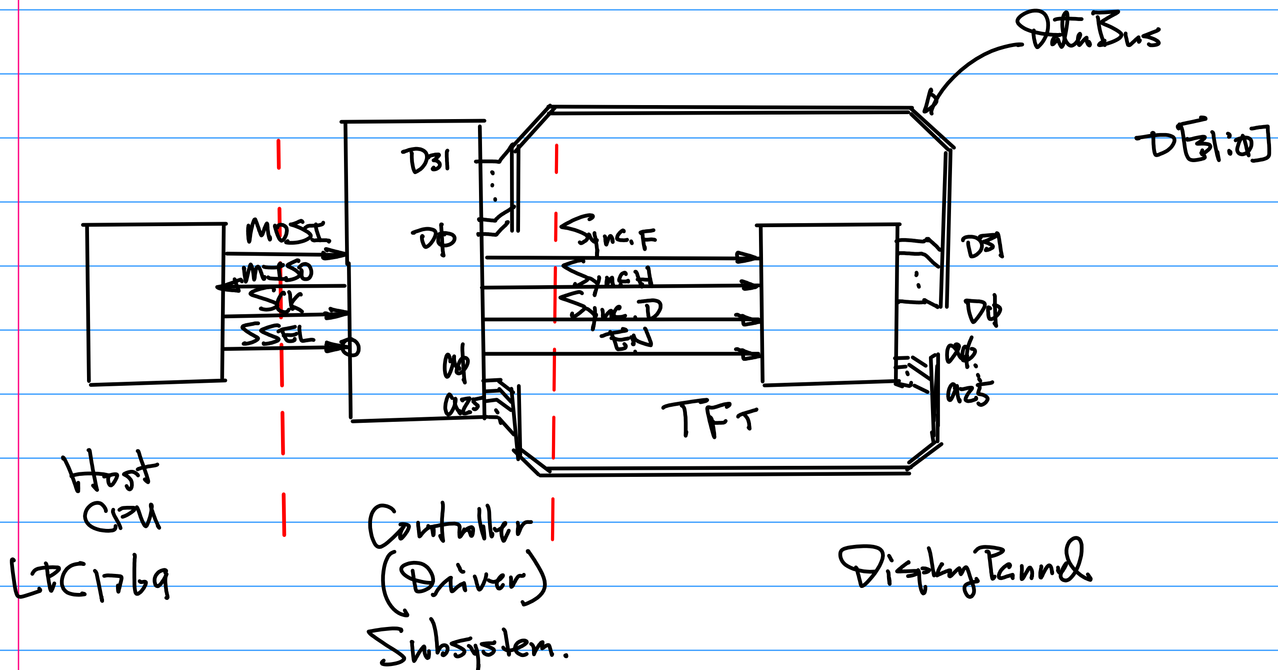
Continuation on the Control Design.

The Design involves conventional Approach with Bus Systems (Addr. + Data), Control Signals. plus the Timing Signals

CMPE240

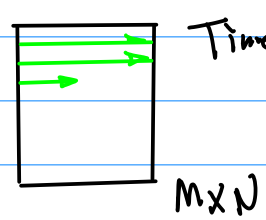
F2023.

Note 1° Data Bus. 32 Bits; Note 2° Addr. Bus. 26 Bit for Example.



The motivation for Sync. Timing: To control the Display, e.g., when to light up the pixel(s). — Previous From the Lecture Note

1. Sync. Signals
   - ① Frame Sync. Example: 30 F.P.S. $Sync._F \rightarrow f_F = 30\,Hz$ ...(1)
   - ② Horizontal Sync. for $M \times N$ Resolution, it Repeats N Times Per frame.
   - ③ Data Sync. (pixel) for $M \times N$ Resolution, Per Each Line, it Repeats M



Time to Scan Each Row is determined by Horizontal Sync.

CmpE240

F2023

Note: Design Hardware Counter, Counts by N. PCLK → $Sync_F$

Consider the Relationships Between $Sync_F$ & $Sync_H$, Denoted as f_H

$$f_H = N \cdot f_F \quad \ldots (2a)$$

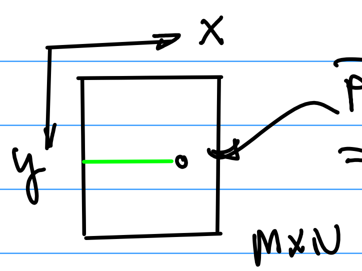
$$Sync_H = N \cdot Sync_F \quad \ldots (2b)$$

Consider $Sync_D$ V.S. $Sync_H$.

$$f_D = M \cdot f_H \quad \ldots (3a)$$

$$Sync_D = M \cdot Sync_H \quad \ldots (3b)$$ M: No. of Cols.

Example:



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

Calculate the Location Information Based on Timing ($Sync_F$, $Sync_H$, $Sync_D$)

Given the frame rate, find the timing information to plot pixel $\vec{P_i}(x_i, y_i)$.

Assume Frame Rate 30FPS.

Sol: Since the frame Rate is

$$Sync_F = f_F \quad \ldots (4)$$

therefore, the time Needed for the Display Device to ~~Reach~~ Complete the scanning of line $y_i - 1$, hence,

$$T_1 = \frac{1}{f_H} \cdot (y_i - 1) = \frac{1}{Sync_H}(y_i - 1) \quad \ldots (5)$$

Then, the Time to Complete the Drawing/Scanning of $\vec{P_i}(x_i, y_i)$ on line y_i is:

$$T_2 = \frac{1}{f_D} x_i = \frac{1}{Sync_D} x_i \quad \ldots (6)$$

So, the total Time is $T_\Sigma = T_1 + T_2 = \frac{1}{Sync_H}(y_i - 1) + \frac{1}{Sync_D} x_i$

CmpE240
F2023

7/1

Group I Classes

Group I classes are those classes which meet M, W, F, MTW, MWR, MTWF, MWRF, 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	Monday, December 11	7:15-9:30 AM
8:30 through 9:25 AM	Wednesday, December 13	7:15-9:30 AM
9:30 through 10:25 AM	Friday, December 8	7:15-9:30 AM
10:30 through 11:25 AM	Tuesday, December 12	9:45 AM-12:00 PM
11:30 AM through 12:25 PM	Thursday, December 14	9:45 AM-12:00 PM
~~12:30 through 1:25 PM~~	~~Monday, December 11~~	~~12:15-2:30 PM~~
1:30 through 2:25 PM	Wednesday, December 13	12:15-2:30 PM
~~2:30 through 3:25 PM~~	~~Friday, December 8~~	~~12:15-2:30 PM~~
3:30 through 4:25 PM*	Tuesday, December 12	2:45-5:00 PM
4:30* through 5:25 PM*	Thursday, December 14	2:45-5:00 PM