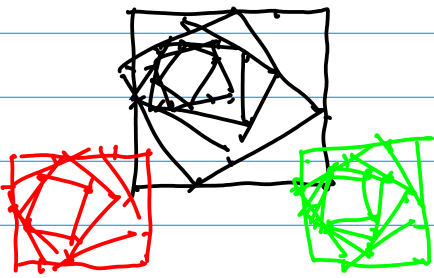
CMPE240

a A set of Squares Rotation with one same color

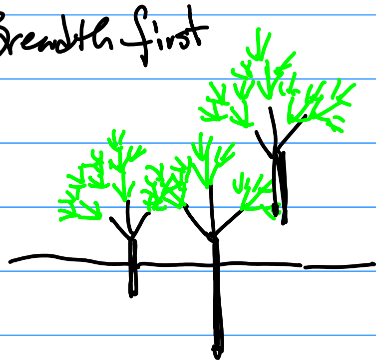
b Change location/color/size

c Keep the patterns, please don't erase them!



Part II Once Part I is Done Switch to Part I.

a Breadth first



b $L \geq 7$ c Create A patch of Forest, changing location/size

Note: Please Don't Erase the Drawing Till All the trees are constructed.

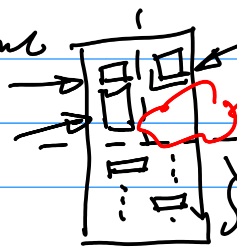
Note: Report Writting.

3D G.E. (Graphics Engine)

GPU Architecture

Jtag

CortexM3



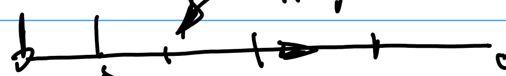
mem. Control

3D G.E.

Peripheral Controllers.

{GPP SPRs, SPI SPRs.

Solid obj HL, HS. Removal



A B

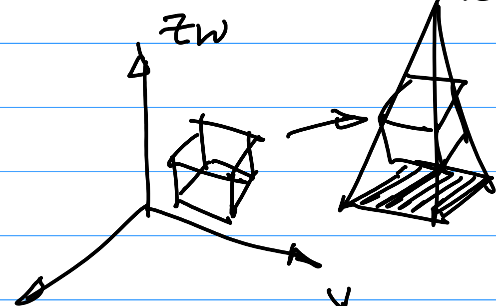
(2) Wireframe

3D Concepts + Algorithm

Shade

State-of-the-Art

(1) World-To-Viewer Transform



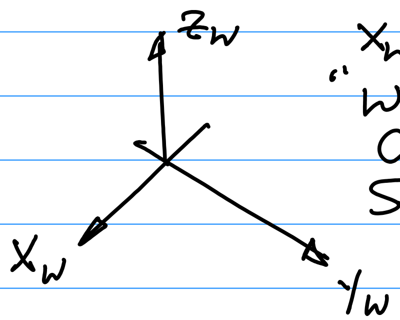
Lighting model. Diffuse Reflection

March 15 (monday)

Note: 1° midterm A week from this Wednesday (march 24th)

2° Smartphone (CAM Capability) A piece of Paper to the Class. Test the Environment

Ref: github: 2018F-114-3D Graphics

1° 

X_w-Y_w-Z_w "World" Coordinate System.

Fig. 1

Right-Hand System,

2° Virtual Camera

a. Virtual Enclosure

b. Optic Lens (Pin Hole)

$\Phi_d \simeq 0$ Very Small diameter

c. Projection plane, the plane to form projected Image when light passing through the lens and Reaches the projection plane.

3. Denote $\vec{E}(x_e, y_e, z_e)$ as

Virtual Camera Location

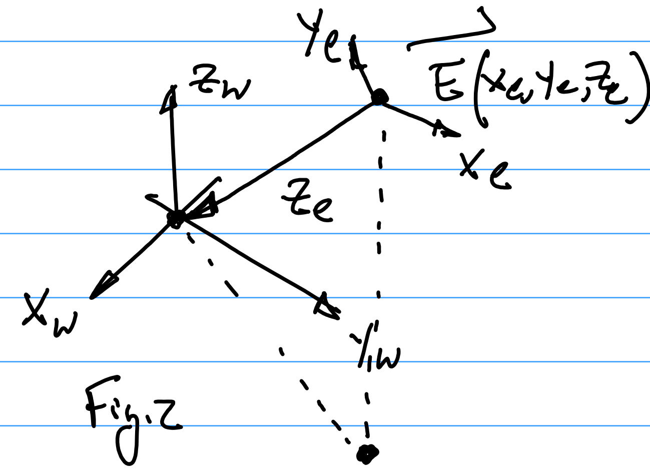


Fig. 2

Projection of $\vec{E}(x_e, y_e, z_e)$ on to X_w-Y_w plane in the X_w-Y_w-Z_w World Coordinate System

4. Viewer Coordinate System

Viewer ~ Virtual Camera

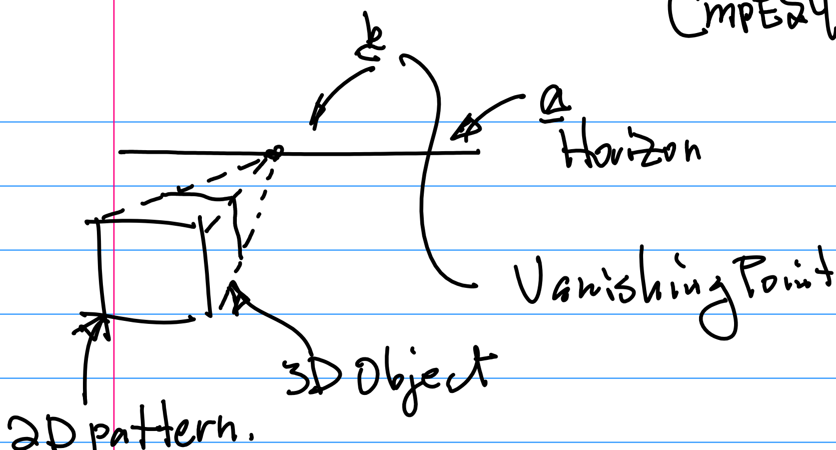
X_e-Y_e-Z_e Viewer Coordinate System. "Eye"

a. $\vec{E}(x_e, y_e, z_e)$ As the origin.

b. Left-Hand System.

c. Z_e-axis points to the origin.

5. Perspective Projection



3D Object formed on 2D plane

7. Transformation Pipeline

To Display 3D object(s) on to 2D LCD Display Screen.

W2V (Work-2-Viewer) Transform

Perspective Projection.

6. Perspective Projection → @ the Virtual Camera in X_w-Y_w-Z_w, in X_e-Y_e-Z_e.

Objects $\vec{E}$ in X_w-Y_w-Z_w → Viewer Coordinate System X_e-Y_e-Z_e

Projection Plane form Perspective Projection

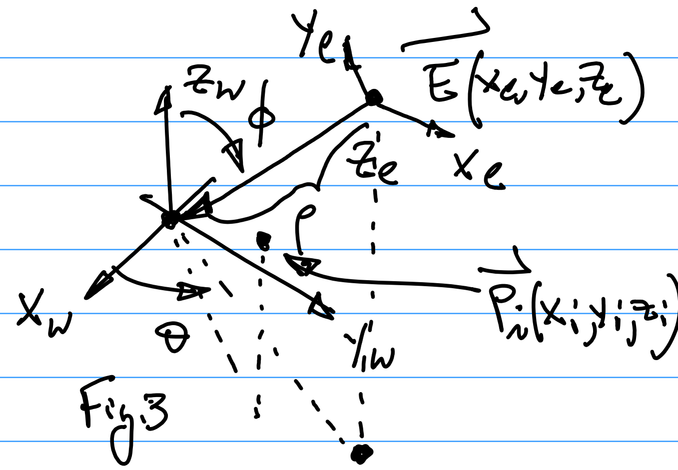


Fig.3

3 parameters:

a. theta θ Angle

b. Vector $\vec{E}$ to $\vec{O}$, "rho", Distance ρ

Mathematical Formulation

1° Formula: Transformation Pipeline
- a. World-to-Viewer Transform
- b. Perspective Projection

2° Foundation
- a. Translation
- b. Rotations
  - ~ wrt X_w
  - " Y_w
  - " Z_w
  - And wrt Arbitrary axis

$$\rho \triangleq \sqrt{x_e^2 + y_e^2 + z_e^2} \quad \cdots (1)$$

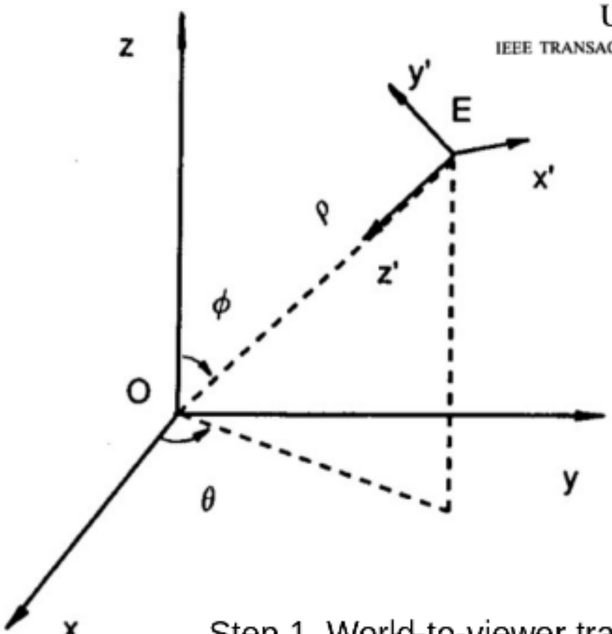
c. Angle Z_w and Z_e ϕ.

World_2_Viewer

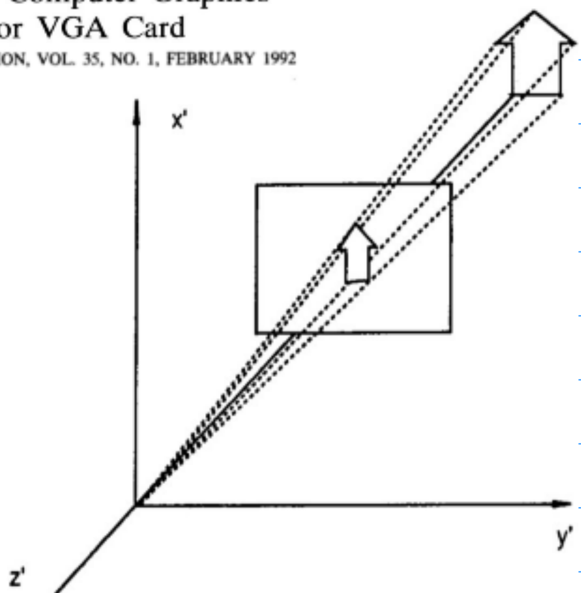
3D Transformation Pipeline Technique

Reference: H. Li Three-Dimensional Computer Graphics Using EGA or VGA Card

IEEE TRANSACTIONS ON EDUCATION, VOL. 35, NO. 1, FEBRUARY 1992



Step 1. World-to-viewer transform



Step 2. Perspective Projection

$$\mathbf{T} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_p = x_e\left(\frac{D}{z_e}\right)$$

$$y_p = y_e\left(\frac{D}{z_e}\right)$$

Harry Li, Ph.D.

Transform, map $\vec{P_i}$ from the World-coordinate to Viewer Coordinate.

$$\begin{pmatrix} x_i' \\ y_i' \\ z_i' \\ 1 \end{pmatrix} = T \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \quad \cdots (2)$$

After $x_e - y_e - z_e$

Before $x_w - y_w - z_w$

World to Viewer Transform.

$$\begin{pmatrix} \sin\theta & \cos\theta & 0 & 0 \\ \cos\theta & \sin\theta & " & " \\ \cos\theta & \cos\theta & " & " \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Part I. "θ"

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \cos\phi & \cos\phi & \sin\phi & " \\ \sin\phi & \sin\phi & \cos\phi & " \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Part II "ϕ"

$$\begin{pmatrix} - & & & \\ - & - & & \\ - & - & - & \rho \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Part III "_"

$\underline{\rho}$

March 17 (Wed)

Topics: 1° Hardware Architecture 2° Software SPRs Init & Config.

Example: 2D & 3D G.E. SPI I/F LCD Display to work with LPC1769.

GPIO (GPP) SPI

A set of 6 ~

- $\left\{ \begin{array}{l} \text{System Configuration} \left\{ \begin{array}{l} \text{Per. Cont.} \\ \text{PWR} \\ \text{CLK} \end{array} \right. \\ \text{- - - - - - - - - -} \\ \text{Configuration of the Peripheral Cont.} \end{array} \right.$ multiplexing

*

Compare to Rotation Matrix

$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

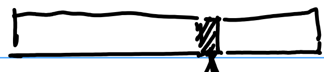
RTOS

Peripheral Controller $\left\{ \begin{array}{l} \text{GPP} \\ \text{SPI} \end{array} \right.$

SPI's SPR.

1. Naming Convention
   LPC_SC → PCONP
2. Power Up the Selected Peripheral Controller By Setting the Corresponding.

CmpE240



"1"

Bitwise "OR"

$|= (1<<10)$

$\underbrace{\&=}_{\text{"AND"}} \sim(3<<20)$

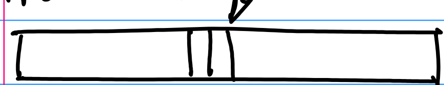
"11"

Negation, "00"

$\underbrace{|=}_{\text{"OR"}} (1<<20)$

"01"

Clear 2 Bits



Set 2 bits as "01"

$\underbrace{\&=}_{\text{"AND"}}\sim((3<<18)|(3<<16)|(3<<14));$

Nega.

"11"

Total 6 bits << 18

"AND" clear 6 bits

$\underbrace{|=}_{\text{"OR"}} ((2<<18)|(2<<16)|(2<<14)|)$

"10"

Set

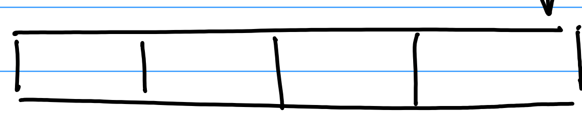
Note: CR0, CR1

LPC_SSP1 -> CR0

Control Register

Table 371

CR0



Tech Spec.

1° 8 bit Transfer

CR0[3:0] = 0111 = 0x7

2° SPI

CR0[5:4] = 00 (SPI)

3° Clock f_{SPI}

a CR0[15:8]

8 bits $255 = 2^8$

b

$$f_{SPI} = \frac{PCLK}{(SCR+1)\,CPSDVSR} \quad \cdots(1)$$

[0, 255]

Mauch 22nd.

Today's Topics:

1° Midterm Review

2° SPRs CR0, CR1 for SPI I/F

Ref:

1° CPU Datasheet. pp 431-433.

2° Sample code SPI init (Draw a Line)

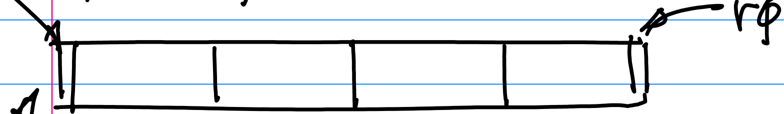
CMPE240

27

Example: SSP.C Source Code Walk-Through. 151-208

line 162-165 PINSEL0

Table 80, PP. 117 CPU Datasheet



PINSEL0 [31:30] = 10

Table 81. PINSEL1

= 0X2
10 from CPU Datasheet

PINSEL1 [1:0], PINSEL1 [3:2], PINSEL1 [5:4] → SSP1

SSEL0., MOSI0, MISO0

line 173

CR = 0X0707 → Tech spec.

.... 0000 0111 0000 0111

16 upper Bits all zeros.

CR0[15:8] SCR Clock

From Datasheet PP431

8 bit Transfer

CR[5:4] = 00 SPI

$$f_{SPI} = \frac{PCLK}{(SCR+1) * DVR}$$

CR0

CR0[15:8] = 0X7

CR1[2] = 0 for "Master"

$$f_{SPI} = \frac{1 \times 10^6}{(7+1) * DVR}$$

To find f_{SPI}.

$2^8 = 256$, SCR ∈ [0, 255]

PP. 433. CPSDVSR ∈ [2, 254]

Midterm Review

1° Video ON, Mandatory.

a Submission to CANVAS 15 min. File Uploading
No Late Submission After the Deadline
Paper will be disqualified

b If CANVAS Disrupted, then E-mail Submission

c file in "Zip"

FirstName + 4 Digits + CMPE242 mid.zip
SID

2° 3 Questions ±

Hardware
- CPU Block Diagram
- memory map
- SPRs.

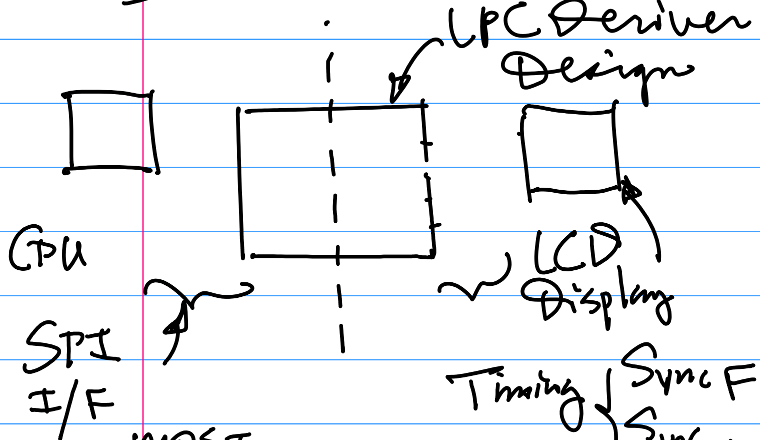
CKT, SCH Design

Software: Coding

SPR Coding → Binary Pattern for Init & Config, Debugging Purpose ← Tech Spec.

Algorithm:
2D Vector Graphics.
G.E.

a LPC CPU   LPC Driver.



MOSI
MISO
SCK
SSELn

Timing { Sync F, Sync H, Sync D }

$\overline{CS}$

b Virtual v.s. physical Display Transform.

c $\vec{P} = \vec{P_i} + \lambda(\vec{P_{i+1}} - \vec{P_i})$

Screen Saver. { i. Rotation! No, W/O Rotation Matrix; ii. 2D Transforms }

Composition of 2D Transform. { $R_{3\times3}$, $T_{3\times3}$ } → Tree.

Preprocess + $R_{3\times3}$ + Post ~

Formula: One Page Formula Sheet; is allowed. However, No Example, or Verbal Explanation is Not Allowed, Submission of the formula Page is required with your midterm paper.

No multiple choice question.

SCH: Requires All the pins needed in the design to have Label; wire: "Arrow" to indicate direction.

Block Diagram: wire(s), Label(s), direction (Arrow)

CPU Datasheet will be provided

C Code program will be provided for Answering questions, or for Redesign.

Calculator is allowed;

CMPE240

29

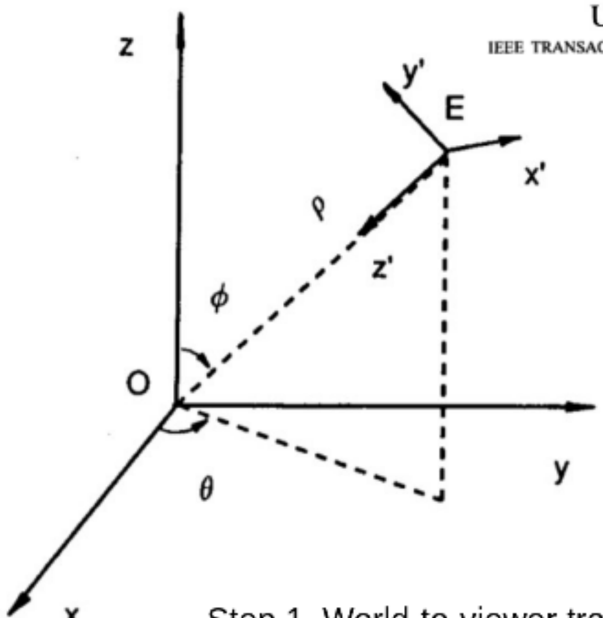
April 5 (Monday)

1° Midterm Key on github, "Key" To search.

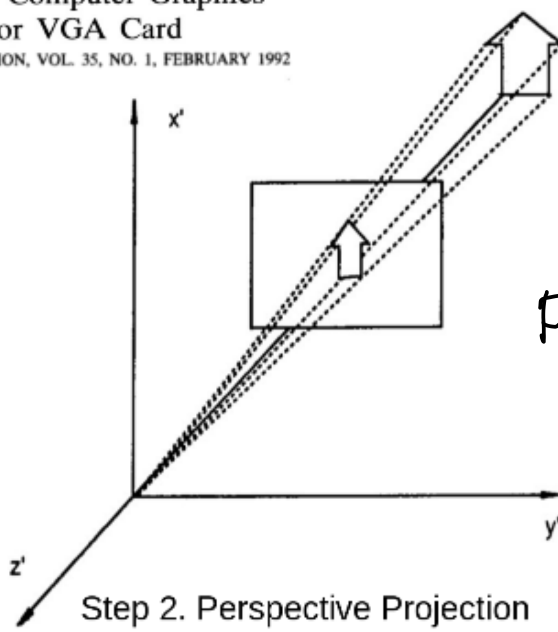
3D Transformation Pipeline Technique

Reference: H. Li Three-Dimensional Computer Graphics Using EGA or VGA Card

IEEE TRANSACTIONS ON EDUCATION, VOL. 35, NO. 1, FEBRUARY 1992



Step 1. World-to-viewer transform



Step 2. Perspective Projection

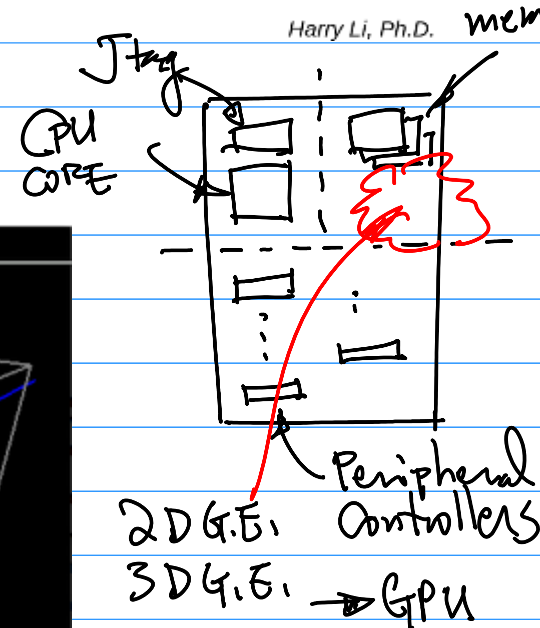
Fig 1.

$$\mathbf{T} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi & 0 \\ -\sin\phi\cos\theta & -\sin\phi\cos\theta & -\cos\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(1)$$

$$x_p = x_e\left(\frac{D}{z_e}\right) \qquad y_p = y_e\left(\frac{D}{z_e}\right) \quad \ldots(2)$$

Harry Li, Ph.D.

Today's Topics: 3D G.E.



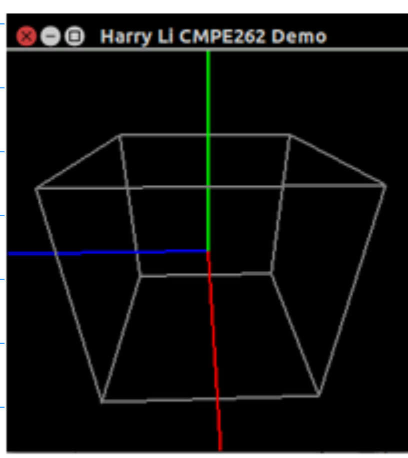


Fig 2a

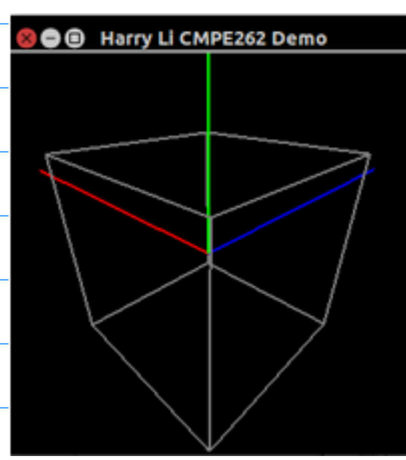


Fig 2b

from github. Key "111a, 111b, ..."

CMPE240

30

Note: 2D G.E. { Vector Graphics
Transformation
Primitive Graphics { a D.D.A. b line c Arc/circle etc.

Transistor/Gate Level.

Example: 3D Wireframe Model

First, X_w-Y_w-Z_w World Coordinate System.

a Right System, r-g-b for X_w, Y_w, Z_w axis

b Transformation Pipeline

{ 1st World-2-Viewer
$T_{4\times4}: (x_w, y_w, z_w) \rightarrow (x_v, y_v, z_v)$ ...(3)

2nd Perspective Projection
$P: (x_v, y_v, z_v) \rightarrow (x_p, y_p)$ ...(3b)

#define X_w_axis ~

⋮

#define z_w_axis ~

Now, Implementation (Drawing r-g-b axis) ← April 12 (M)

Homework: Draw r-g-b axis on your LPC Display. Bring your Program/Board to the Next Class.

Step 2. World_2_Viewer

$T_{4\times4}: (x_w, y_w, z_w) \in \mathbb{R}^3 \rightarrow (x_v, y_v, z_v) \in \mathbb{R}^3$

Second. Design of Dataset, e.g. 4 Vertices for X_w-Y_w-Z_w Axis

$\vec{P}(x, y, z)$ 3D pt. in X_w-Y_w-Z_w

Step 1. $\vec{P}(x, y, z) \in \mathbb{R}^3$

for the Origin $\vec{P}_0(x_0, y_0, z_0) = (0, 0, 0)$ ...(4)

$\vec{P}_x(x_x, y_x, z_x) = (100, 0, 0)$, ...(4-1)

$\vec{P}_y(x_y, y_y, z_y) = (0, 100, 0)$, and ...(4-2)

$\vec{P}_z(x_z, y_z, z_z) = (0, 0, 100)$ ...(4-3)

$$\begin{pmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{pmatrix} = T_{4\times4} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad \ldots(5)$$

"After" "Before"

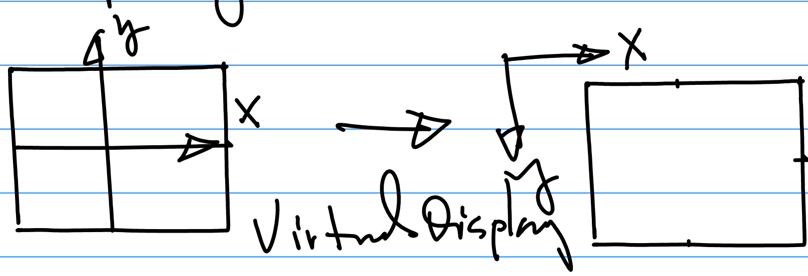
Now, find the X_w-axis in Viewer Coordinate

Step 3. Perspective Projection

$P : (x_e, y_e, z_e) \in \mathbb{R}^3 \to (x_i'', y_i'') \in \mathbb{R}^2$

Coordinate on your LPC1769 Display Device.

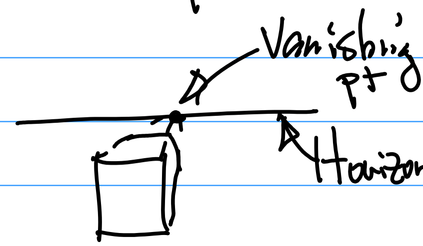
Note:
(x_i'', y_i'') is defined on 2D Virtual Display Coordinate
Need to perform Virtual to Physical Transformation in order to plot your Result.



Back projection plane: the only light that can reach the projection plane is the light passing through the "pin hole". (plane is at the Back) projection

Frontal projection plane: move the back projection plane to the outside of the virtual Camera.

D: Distance from the projection plane to the "pin hole".



$$\begin{cases} x_p = \dfrac{D}{z_v} x_v \quad \text{..(a1)} & (x_v, y_v, z_v) \in \mathbb{R}^3 \\ y_p = \dfrac{D}{z_v} \cdot y_v \quad \ldots \text{(a2)} & \end{cases}$$

Virtual Cam

focal Length

(x_p, y_p) On your 2D Display

In Homework, $D = 20$

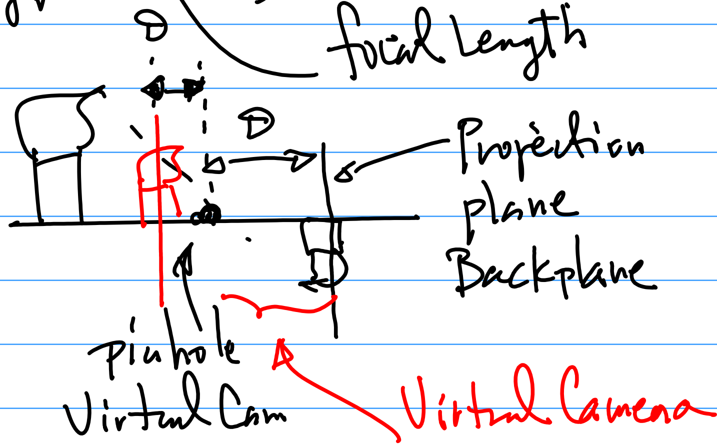
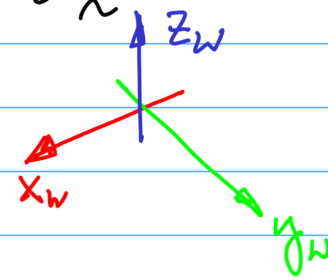


Fig 1.



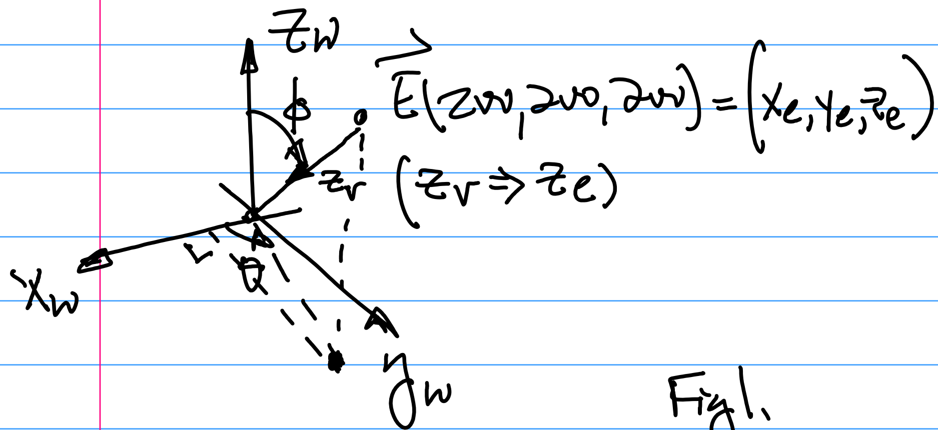
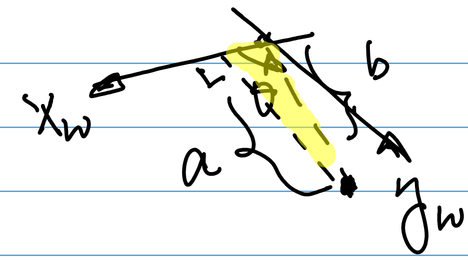


Fig 1.



$\sin\theta = \frac{a}{b}$

$a = y_e = 200$

$b = \sqrt{x_e^2 + y_e^2} = 200\sqrt{2}$

$\therefore \sin\theta = \frac{200}{200\sqrt{2}} = \sqrt{2}/2$

$\cos\theta = \frac{x_e}{b} = \frac{200}{200\sqrt{2}} = \sqrt{2}/2$

a. Vector Passing Through $\vec{E}$, Perpendicular to X_w-Y_w plane, So form an intersection pt.

b. Vector Passing through the intersection pt on X_w-Y_w, perpendicular to X_w axis.

Note: Angle θ (Theta) on X_w-Y_w plane

{ wrt Positive X_w-axis, And
{ Counter Clockwise Direction

Note: Angle ϕ (phi) on z_v-z_w plane.

Now, find each entry on $T_{4\times4}$ matrix, So World-2-Viewer Transform Can be Performed.

for Angle θ (Theta)

$\sin\theta$, and $\cos\theta$

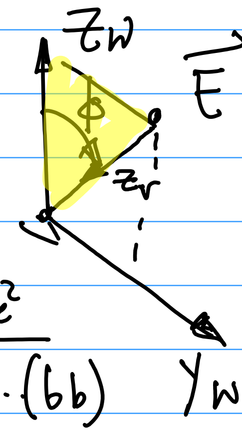
for ϕ (phi)

$\cos\phi = \frac{z_e}{\rho}$ ...(5b)

$\rho = \sqrt{x_e^2 + y_e^2 + z_e^2}$ ...(6)

"rho"

$\sin\phi = \frac{\sqrt{x_e^2 + y_e^2}}{\rho}$ ...(6b)



April 7 (Wed)

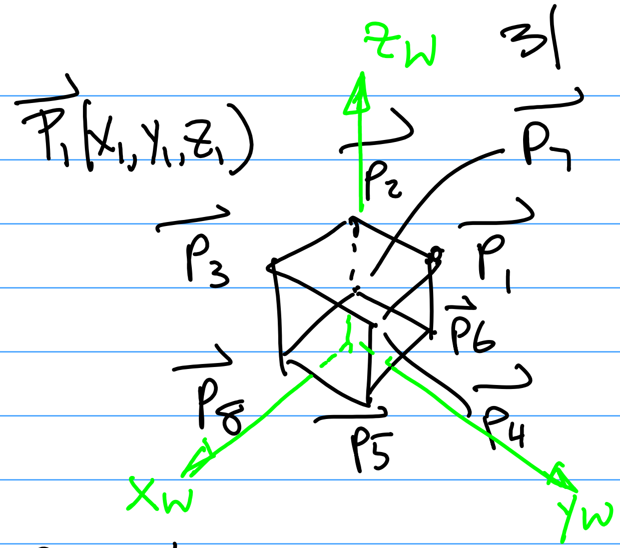
Note: X_v-Y_v-Z_v Left Hand System

$\cos\phi = \frac{z_e}{\rho} = \frac{200}{\sqrt{200^2 + 200^2 + 200^2}}$

$= \frac{200}{200\sqrt{3}} = \sqrt{3}/3$

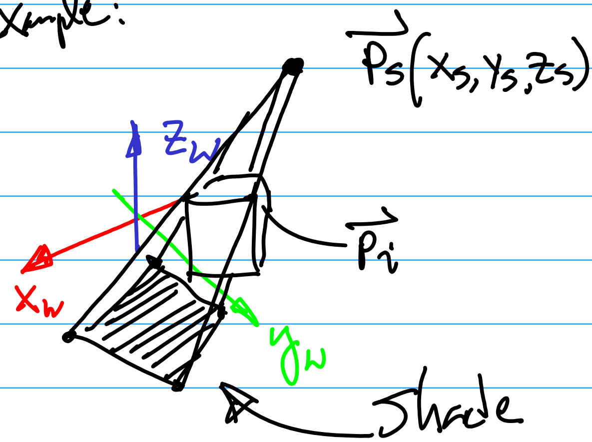
Similarly

$$\sin\phi = \sqrt{x_c^2 + y_c^2}/\rho = \frac{\cancel{20}\sqrt{2}}{\cancel{20}\sqrt{3}} = \sqrt{2/3}$$



Consider "Shade" Calculation.

Example:



One side of the cube overlapped with Z_W-axis size of 100.

$\vec{P}_1(0, 100, 110)$

$\vec{P}_2(0, 0, 110)$, $\vec{P}_3(100, 0, 110)$

$\vec{P}_4(100, 100, 110)$

Note: $\vec{P}_1, \vec{P}_2, \ldots,$ are arranged Counter clockwise

1. X_W-Y_W-Z_W.

April 12 (Monday)

Note: 1° Homework Submission (E-mail)
X_W-Y_W-Z_W Drawing.
Source code — Photo
(Exported Project) By Wednesday.
2° Bring your Prototype Board to Each Session for Inspection, Show & Tell.

$\vec{P}_5(100, 100, 10)$, $\vec{P}_6(0, 100, 10)$

$\vec{P}_7(0, 0, 10)$, $\vec{P}_8(100, 0, 10)$

Discussion on 3D Shade Computation

2. Point Light Source

$\vec{P}_s(100, 100, 200)$

Conditions for this discussion

1° Define $\{\vec{P}_i(x_i, y_i, z_i) \mid i = 1, 2, \ldots, 8\}$

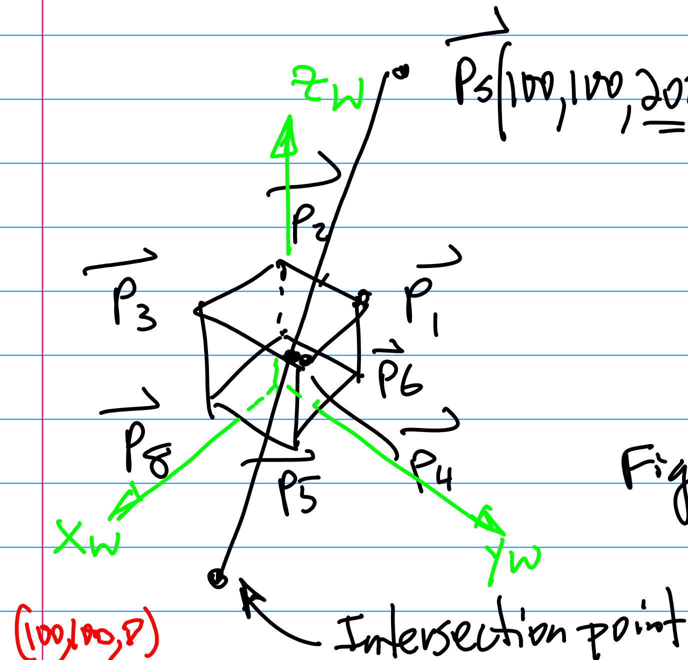


Fig1.

3. Ray Equation, Connecting $\vec{P_5}$ to $\vec{P_4}$,

$$\vec{R} = \vec{P_5} + \lambda(\vec{P_i} - \vec{P_5}) \quad \ldots (1)$$

$$\vec{P_i} = \vec{P_4}$$

OR,

$$\vec{P} = \vec{P_5} + \lambda(\vec{P_i} - \vec{P_5}) \quad \ldots (1b)$$

Find intersection point on X_w-Y_w plane.

4. plane Equation

<u>4a</u> Normal Vector $\vec{n}(n_x, n_y, n_z)$ perpendicular to the given plane, X_w-Y_w plane

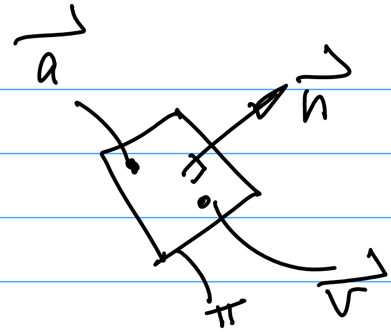


Fig.2

4b. plane π.

An Known, Arbitrary pt on π denoted $\vec{a}(a_x, a_y, a_z)$

An arbitrary point $\vec{v}(v_x, v_y, v_z)$ on π

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

OR

$$\underset{\text{Normal Vector}}{\vec{n}} \cdot \underbrace{(\vec{v} - \vec{a})}_{\text{line segment}} = 0 \quad \ldots (2\text{-}a) \checkmark$$

5. Intersection pt.

$$\begin{cases} \vec{R} = \vec{P_5} + \lambda(\vec{P_i} - \vec{P_5}) & \ldots (3a) \\ \vec{n} \cdot (\vec{v} - \vec{a}) = 0 & \ldots (3b) \end{cases}$$

From (3b),

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

$\vec{v} = \vec{R}$, $\vec{v} = \vec{P_s} + \lambda(\vec{P_i} - \vec{P_s})$

Sub. $\vec{v} = \vec{R}$ Above into (3b)

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

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

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

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

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

$$\lambda = \frac{\vec{n} \cdot (\vec{a} - \vec{P_s})}{\vec{n} \cdot (\vec{P_i} - \vec{P_s})} \quad \ldots (4)$$

---

from Eqn (3a), Ray Eqn.

with λ we can find the intersection Point.

In C/C++ Coding.

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

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

Example: Compute the shade by finding intersection pt formed by $\vec{P_s}$ and $\vec{P_4}$.

Sol: From Eqn (3a), we have

$$\vec{R} = \vec{P_s} + \lambda(\vec{P_4} - \vec{P_s})$$

$$= (x_s, y_s, z_s) + \lambda(x_4 - x_s, y_4 - y_s, z_4 - z_s)$$

from Eqn (4), where

$$\vec{n}(n_x, n_y, n_z) = (0, 0, 1)$$

$$\vec{a}(a_x, a_y, a_z) = (0, 0, 0)$$

Hence,

$$\lambda = \frac{0 \cdot (a_x - x_s) + 0 \cdot (a_y - y_s) + 1 \cdot (a_z - z_s)}{0 \cdot (x_4 - x_s) + 0 \cdot (y_4 - y_s) + 1 \cdot (z_4 - z_s)}$$

$$= \frac{a_z - z_s}{z_4 - z_s} = \frac{0 - 200}{110 - 200} = \frac{20\cancel{0}}{9\cancel{0}}$$

$$= 20/9$$

Therefore, Sub λ Back to the Ray Eqn.

$$\vec{R} = \vec{P_5} + \lambda(\vec{P_4} - \vec{P_5})\Big|_{\lambda = 20/9}$$

$$= (100, 100, 200) + \frac{20}{9}(x_4 - x_5, y_4 - y_5, z_4 - z_5)$$

$$= (100, 100, 200) + \frac{20}{9}(100 - 100, 100 - 100, 110 - 200)$$

$$= (100, 100, 200) + \frac{20}{9} \cdot (0, 0, -90)$$

$$= (100, 100, 200) + (0, 0, -\frac{20}{\cancel{9}} \times \cancel{90})$$

$$= (100, 100, 200) + (0, 0, -200) = (100, 100, 0)$$

Note: 1° Finish Last Homework, then Expand to a Cube. (Display it).

2° Compute/Implement this Algorithm, to Calculate (Hand) Each of Every 4 pts of top Surface of the Cube.

Note: Homework Submission

a e-mail ; b Subject:

First Name + SID (4 Digits) + CMPE240 + HW2