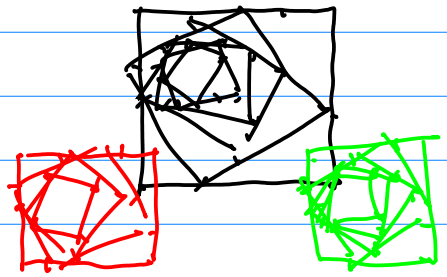
a A set of Squares Rotation with One Same Color

b Change location/Color/size

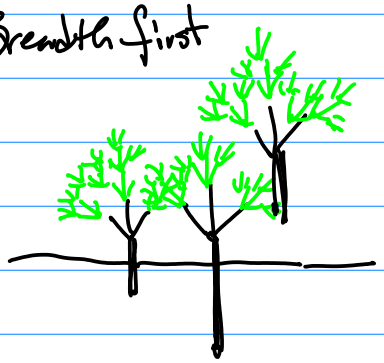
c



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)

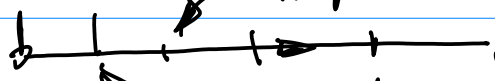
GPU Architecture

Jtag

Cortex M3

GPP SPRs, SPI SPRs.

Solid Obj HL, HS. Removal



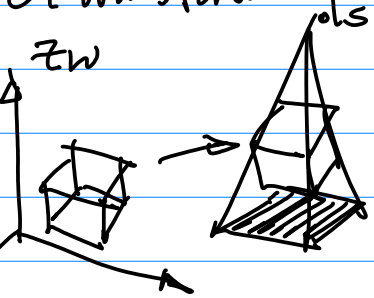
3D Concepts + Algorithm

(2) Wireframe

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

Virtual Camera Location

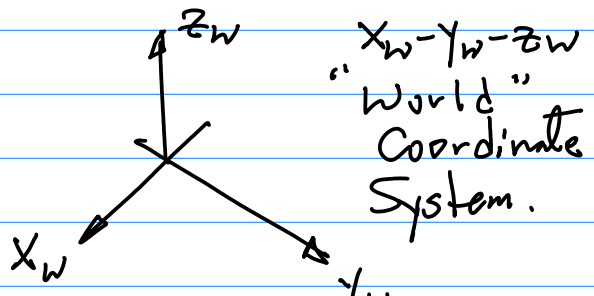
1° 

Fig. 1

Right-Hand System,

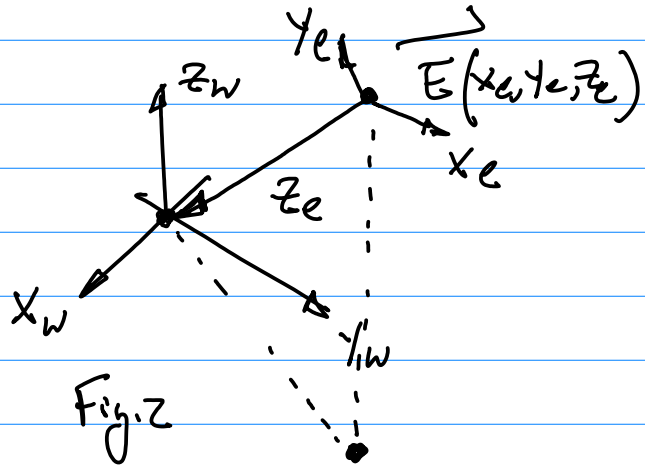


Fig. 2

Projection of $\overrightarrow{E}(x_e, y_e, z_e)$ on to x_w-y_w plane in the x_w-y_w-z_w world Coordinate System

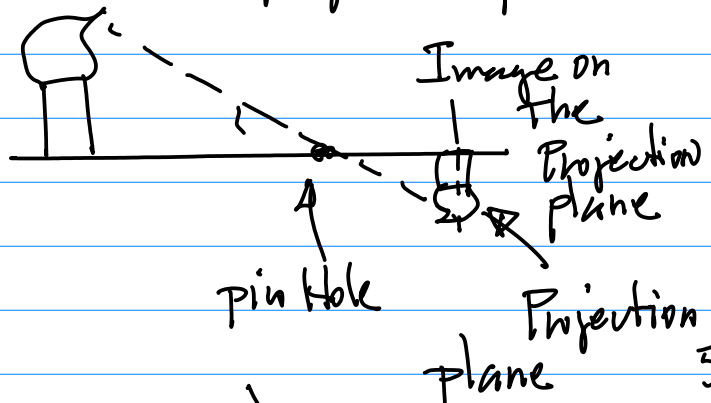
2° Virtual Camera

a. Virtual Enclosure

b. Optic Lens (Pin Hole)

$\phi_d \simeq \phi$ 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 $\overrightarrow{E}(x_e, y_e, z_e)$ as

4. Viewer Coordinate System

Viewer ~ Virtual Camera

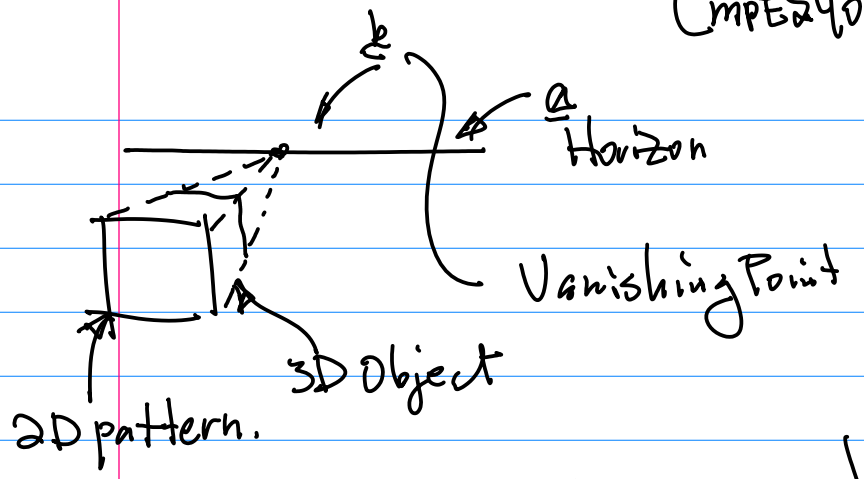
x_e-y_e-z_e Viewer Coordinate System. "Eye"

a. $\overrightarrow{E}(x_e, y_e, z_e)$ 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 in E } in X_w-Y_w-Z_w → Viewer Coordinate System X_e-Y_e-Z_e

Projection Plane form Perspective Projection

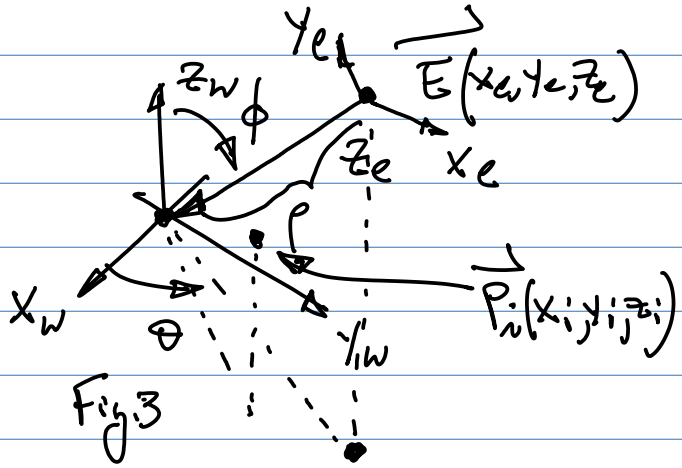


Fig 3

3 parameters:

a ≙ theta θ Angle

b ≙ Vector $\overrightarrow{E}$ to $\overrightarrow{O}$, "rohi", Distance ρ

Mathematical Formulation

1° Formula: Transformation Pipeline

2° Foundation

a ≙ Translation

b ≙ Rotations { ~ wrt X_w .. Y_w .. Z_w And wrt Arbitrary axis

a ≙ World_to_Viewer Transform

b ≙ Perspective Projection

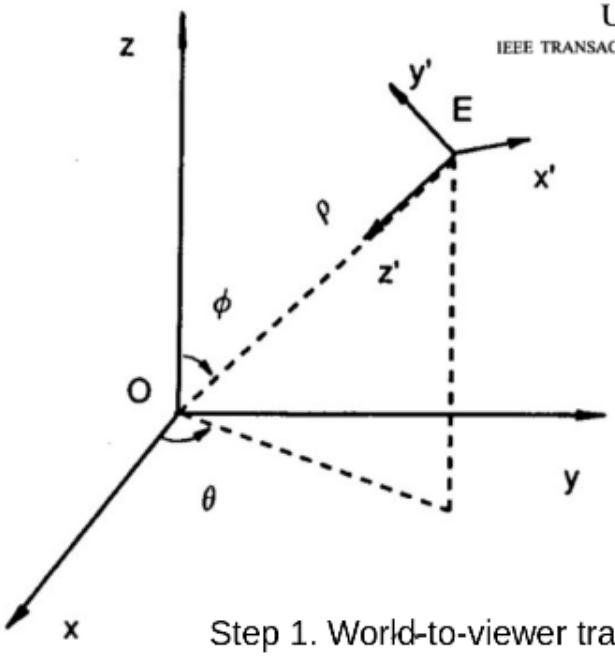
$$\rho \triangleq \sqrt{x_e^2 + y_e^2 + z_e^2} \quad \cdots (1)$$

≙ Angle Z_w and Z_e φ.

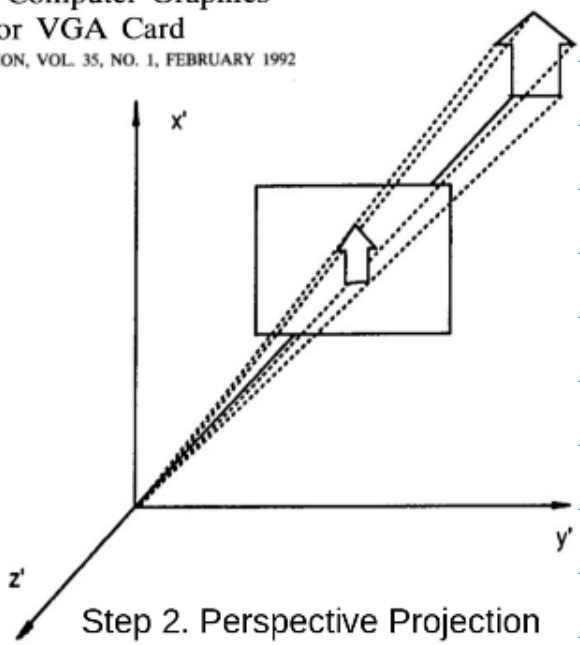
World_2_Viewer

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

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

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

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 b ~
- System Configuration (Per. Cont. PWR CLK multiplexing)
- Configuration of the Peripheral Cont.

Transform, Map 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.

Compose to Rotation Matrix

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

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

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

$$\begin{pmatrix} - & & & \\ - & - & & \\ - & - & - & P \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Part III "—"

roh

RTOS
Peripheral Controller { GPP, SPI }

SPI's SPR.
1. Naming Convention
   LPC_SC → PCONP
2. Power up the Selected Peripheral Controller By Setting the Corresponding.

Bitwise "OR"

1 = (1<<10)

& = ~(3<<20)

"AND" "11" Negation "00"

1 = (1<<20)

"OR" "01"

Clear 2 Bits

Set 2 bits as "01"

```
&=~((3<<18)|(3<<16)|(3<<14));
```

"AND" Nega. "11" Total 6 bits <<18

Clear 6 bits

```
|= ((2<<18)|(2<<16)|(2<<14)|)
```

"OR" Set "10"

Note: CR0 CR1

LPC_SSP1 -> CR0

Control Register

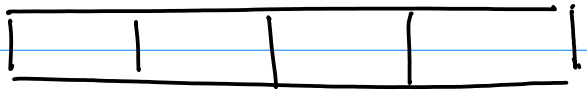


Table 571

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

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

SCR: [0, 255]

Much 22nd.

Today's Topics:

1° Midterm Review

2° SPRs CR0, CR1 for SPI I/F

Ref:

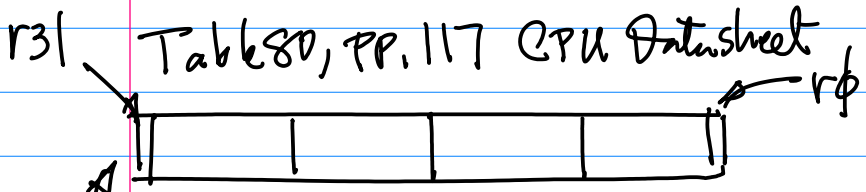
1° CPU Datasheet. PP431-433.

2° Sample code SPIinit (Draw a Line)

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

Line 162-165 PINSEL0

Tab680, PP. 117 CPU Datasheet



PINSEL0[31:30] = 10

Tab681. PINSEL1

= 0x2

10 from CPU Datasheet

PINSEL1[1:0], PINSEL1[3:2], PINSEL4[5:4] → SSP1

SSEL0, MOSI0, MISO0

Line 173

CR = 0x0707 → Tech Spec.

.... 0000 0111 0000 0111

16 upper Bits all zeros.

From Datasheet PP431

8 bit Transfer

CR0[15:8] SCR

CR[5:4] = 00 SPI

= Clock

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

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 G [0, 255]

PP. 433. CPSDVSR G [2, 254]

Midterm Review.

1° Video ON, Mandatory.

2 = Submission to CANVAS

15 min. File Uploading

No Late Submission

After the Deadline paper will be disqualified

2 If CANVAS Disrupted, then E-mail Submission

2 file in "zip"

FirstName + 4 Digits + CMPE240 mid.zip

SID

2° 30 Questions ±

Hardware { CPU Block Diagram

memory map

SFRs.

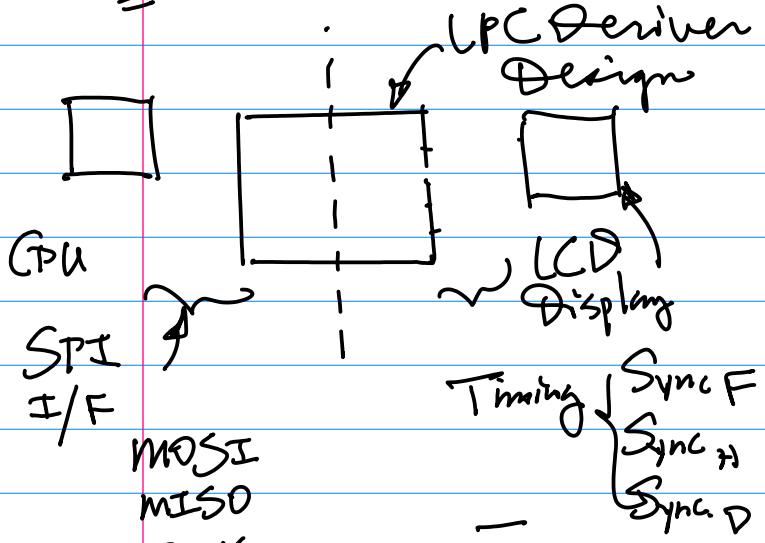
CKT, SCH Design

Software: Coding

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

Algorithm:
2D vector Graphics.
G.E.

a LPC CPU LPC Driver.



b Virtual V.S. physical Display Transform.

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

Screen Saver. { i. Rotation ! No W/O Rotation matrix
ii 2D Transforms

Composition of 2D Transform. { R_{3x3}, T_{3x3} → Tree.

Preprocess f R_{3x3} 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 mid-term 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 Re-design.

Calculator is allowed;

CMPE240

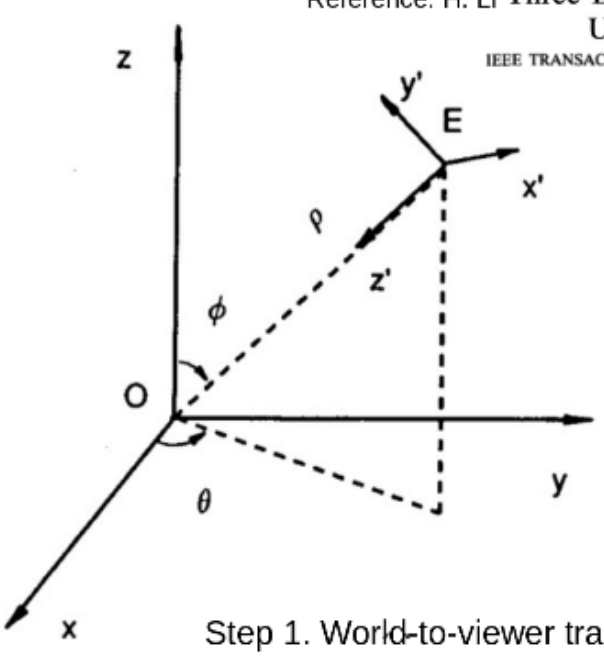
April 5 (Monday)

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

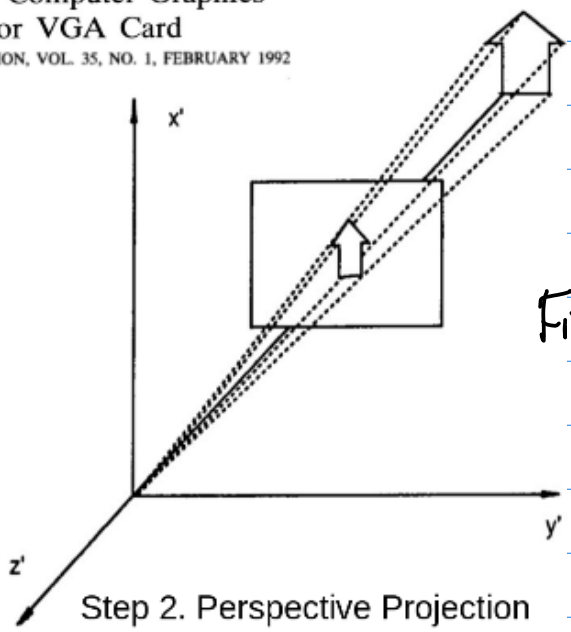
3D Transformation Pipeline Technique

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



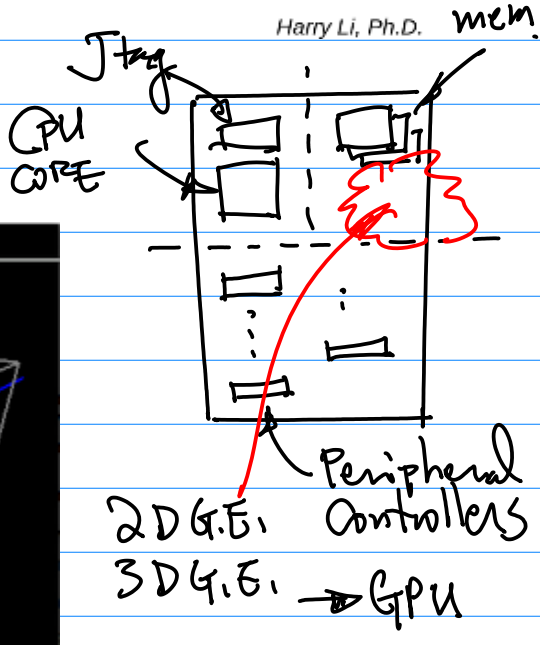
Step 2. Perspective Projection

Fig1.

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

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

Today's Topics: 3D G.E.



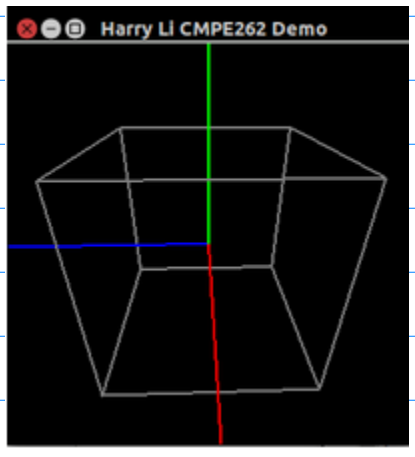


Fig2a

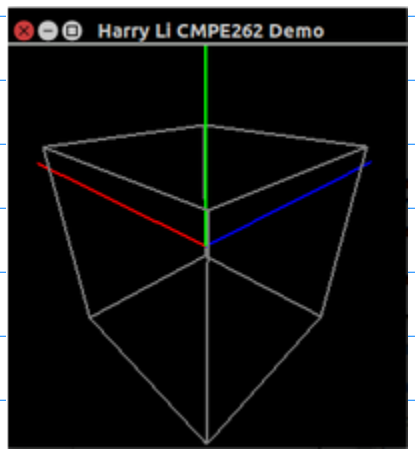


Fig2b

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

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
$T_{4\times4}: (X_w, Y_w, Z_w) \to (X_v, Y_v, Z_v)$ ...(3)
2nd Perspective Projection
$P: (X_v, Y_v, Z_v) \to (X_p, Y_p)$ ...(3b)

#define X_w_axis ~

:

#define Z_w_axis ~

Now, Implementation (Drawing r-g-b axis) → Atiil2(m)

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

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)

Step 2. World_2_Viewer

$T_{4\times4}: (X_w, Y_w, Z_w) \in \mathbb{R}^3 \to (X_v, Y_v, Z_v) \in \mathbb{R}^3$

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

"After" "Before"

Now, find the X_w-axis in Viewer Coordinate

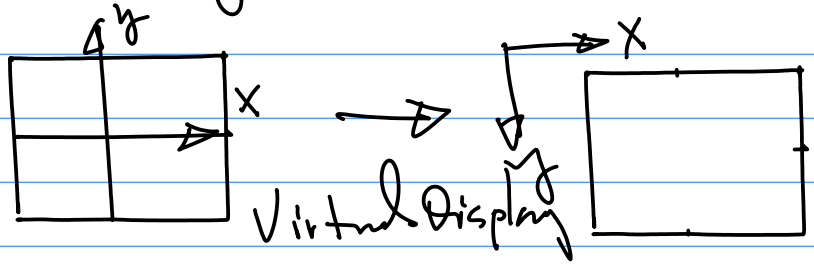
CmPE340

Step3. Perspective Projection

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

(x_i'', y_i''): Coordinate on your LPC1769 Display Device.

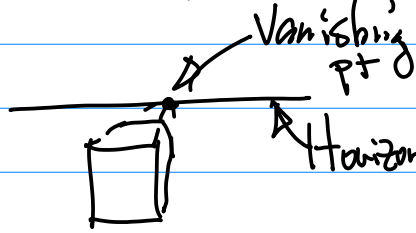
Note:
(x_i'', y_i'') is defined on 2D Virtual Display Coordinate
Need to perform Virtual to Physical Transformation in order to plot your result.



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".



$$\begin{cases} x_p = \frac{D}{z_v} x_v \quad \cdots (a1) \\ y_p = \frac{D}{z_v} \cdot y_v \quad \cdots (a2) \end{cases} \quad (x_v, y_v, z_v) \in \mathbb{R}^3$$

D: Virtual Cam focal length

(x_p, y_p) On your 2D Display

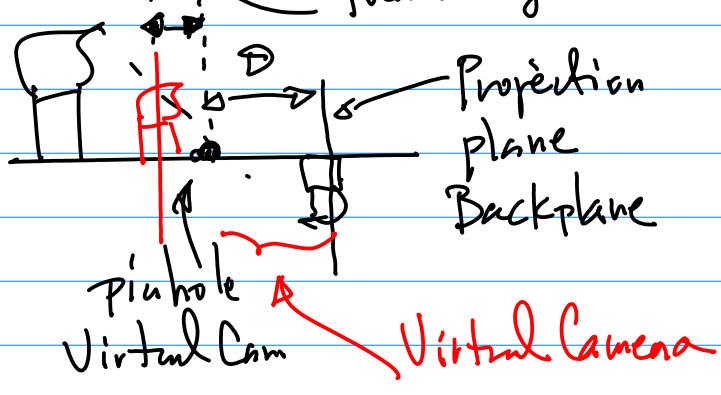
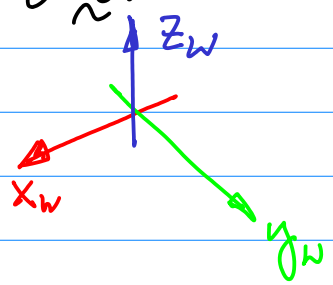


Fig1.

In Homework, $D = 2D$



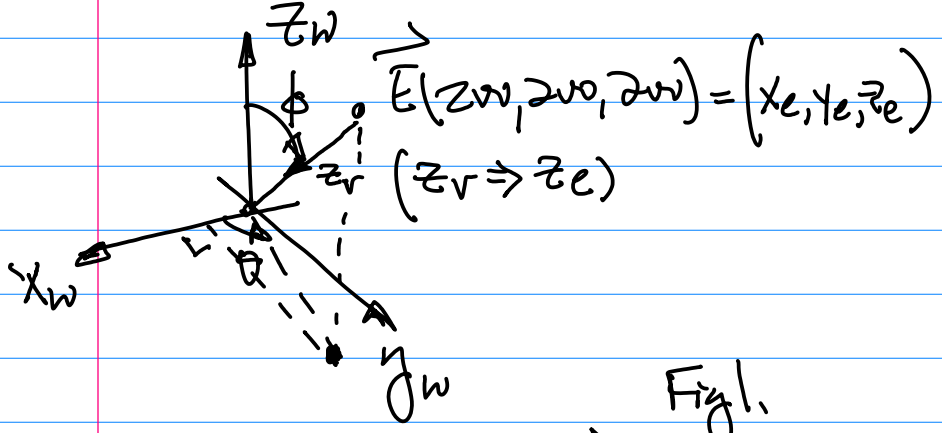
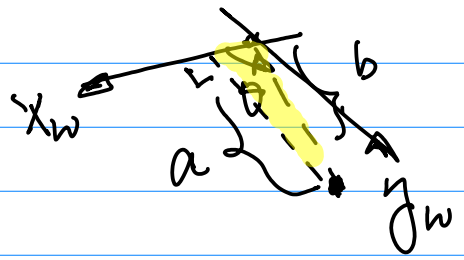


Fig 1.



$\vec{E}$ a Vector Passing Through 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 4x4 matrix, So World-2-Viewer Transform Can be Performed.

For Angle θ (Theta)

Sinθ and Cosθ

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

$$a = y_e = 200$$

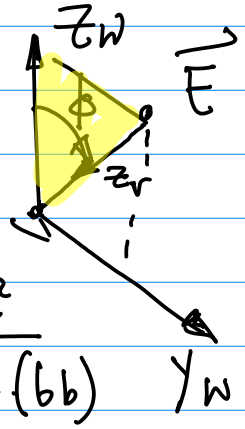
$$b = \sqrt{x_e^2 + y_e^2} = 200\sqrt{2}$$

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

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

for ϕ (phi)

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



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

"rho"

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

April 7 (Wed)

Note: X_v-Y_v-Z_v Left Hand System

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

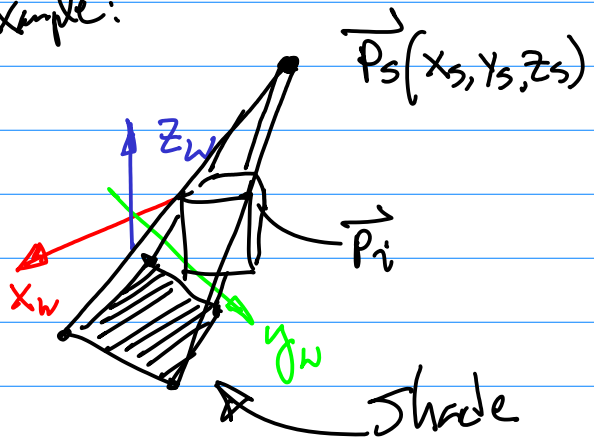
$$= \frac{200}{200\sqrt{3}} = \sqrt{3}/3$$

Similarly

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

Consider "Shade" Calculation.

Example:



1. x_w-y_w-z_w.