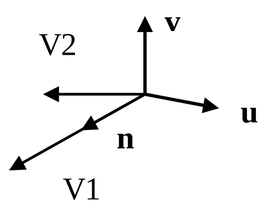
Assignment 1
CS 405/805-001: Computer Graphics

Instructor: Xue Dong Yang

Thursday, September 13, 2012

Due Date: **Thursday, September 27, 2012**

1. Written Question (10 marks)



Given two (non-collinear) vectors ***V1*** = *(V1x, V1y, V1z)* and ***V2*** = *(V2x, V2y, V2z)*. Three orthogonal unit length vectors **u**, **v**, and **n** can be defined such that **n** is along V1; **u** is perpendicular to ***V1*** and ***V2***; and **v** is perpendicular to **u** and **n**, as illustrated in the above figure.

Derive, based on ***V1*** and ***V2***, the formula for calculating the unit-length vectors:

$\mathbf{u} = (u_x, u_y, u_z)$

$\mathbf{v} = (v_x, v_y, v_z)$

$\mathbf{n} = (n_x, n_y, n_z)$

For example, $\mathbf{n} = \boldsymbol{V1}/|\boldsymbol{V1}|$, where $|\boldsymbol{V1}| = \sqrt{V1_x^2 + V1_y^2 + V1_z^2}$, then

$n_x = V1x/|V1|$,

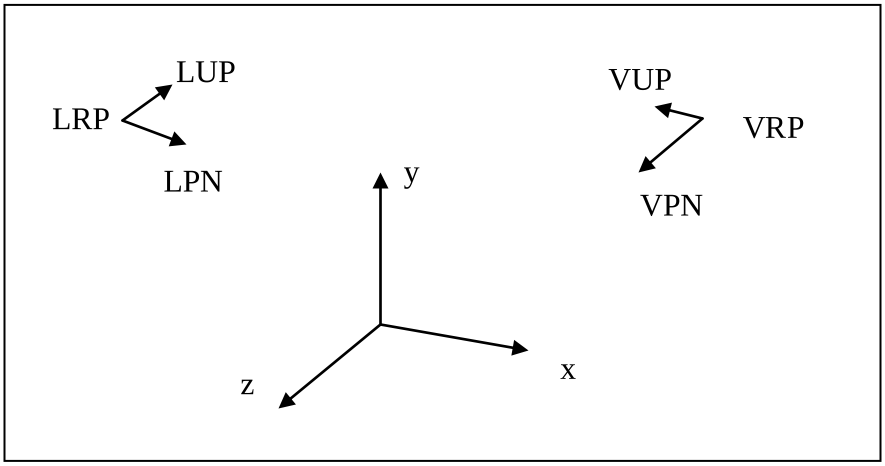
……

2. Programming Question (10 marks)

Write and implement a function (in C or C++) that takes two (non-collinear) vectors V1 and V2 that returns three orthogonal unit length vectors **u, v,** and **n**, as defined in Question 1.

3. Written and Programming Question (40 marks)

In computer graphics, objects are usually defined in the world coordinate system. We also often use a coordinate system that is associated with the camera, and sometimes a light coordinate system, as illustrated in the following diagram:



Where:

- **x-y-z**: world coordinate system
- ***VRP*** = (VRPx, VRPy, VRPz) – View Reference Point (3D point)
- ***VPN*** = (VPNx, VPNy, VPNz) – View Plane Normal (3D vector)
- ***VUP*** = (VUPx, VUPy, VUPz) – Up Direction for the Camera (3D vector)
- ***LRP*** = (LRPx, LRPy, LRPz) – Light Reference Point (3D point)
- ***LPN*** = (LPNx, LPNy, LPNz) – Light Plane Normal (3D vector)
- ***LUP*** = (LUPx, LUPy, LUPz) – Up Direction for the Light (3D vector)

To transform a 3D point from one coordinate system to another, we may need to construct six transformation (4X4) matrices:

M_{wc}: from world to camera
M_{cw}: from camera to world
M_{wl}: from world to light
M_{lw}: from light to world
M_{cl}: from camera to light
M_{lc}: from light to camera

Use the second method given in the lecture to construct these matrices. For example, the transformation from world to camera:

M_{wc} = R*T

where

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

M_{cw} is the inverse transformation of M_{wc}, thus can be derived as:

$M_{cw} = M_{wc}^{-1} = [R * T]^{-1} = T^{-1} * R^{-1}$

It is straightforward for T^{-1} (think about why?):

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & VRP_x \\ 0 & 1 & 0 & VRP_y \\ 0 & 0 & 1 & VRP_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a. Verify that $T * T^{-1} = I$.

b. Because R is an orthogonal matrix, $R^{-1} = R^T$, i.e.:

$$R^{-1} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} u_x & v_x & n_x & 0 \\ u_y & v_y & n_y & 0 \\ u_z & v_z & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What is the definition of orthogonal matrix?
Verify that $R * R^{-1} = I$.

c. Write and implement a function that takes ***VRP, VPN, VUP*** as the input, and constructs two transformation matrices as the results:

M_{wc}: from world to camera
M_{cw}: from camera to world

[Hint: You should utilize the function developed in Question 2.]

d. Write and implement a function that takes LRP, LPN, LUP as the input, and constructs two transformation matrices as the results:

M_{wl}: from world to light
M_{lw}: from light to world

e. We want to construct the following two transformation matrices:

M_{cl}: from camera to light
M_{lc}: from light to camera

Each of these can be constructed by concatenating two proper matrices obtained in (c) and (d). Write down them.

f. Write and implement a testing main program to test your functions.

A set of testing points will be provided separately to you. Required output will also be specified in the testing data file.

Your hand-in should the source program and printed testing results.

----------------------------------- **END** -------------------------------------