Lab Assignment # 3 – Transformations and MATLAB Graphics

Due: Tue. Sep. 30, 2008

Directions: In this assignment you'll practice with some MATLAB functions for 3D graphics and experiment with rotation/scaling and translation transformation of points/vectors in 3D space.

You can work with others and discuss the problems, but each student must write his/her own, independent solution. If you are unsure about what i mean by this, please ask!

What to turn in? Each assigned problem specifies sample outputs you should produce and submit. The solution to each problem should include a print out of the function / script file followed by the specified output (*e.g.*, a plot, a vector, a matrix, etc.)

Problem 1. Read the documentation for MATLAB's built-in functions `meshgrid`, `surf`, `surfnorm`, and `quiver3`. Try to reproduce the example suggested in the help section of the `quiver3` function.

Now, write a MATLAB script that:

1. creates the 3×3 mesh in the xy-plane (*i.e.*, `[x,y] = meshgrid(0:0.5:1,0:0.5,1);`)

   and defines `z = zeros(3,3);`

2. lets `u`, `v`, and `w` hold the coordinates of the unit normal vectors to the surface $z = 0$.

3. creates three 3×3 random matrices `u1`, `v1`, and `w1` holding the x-, y-, and z-coordinates of a set of nine vectors and normalizes them,

4. uses `quiver3` and `surface` to display in the same figure (1) the plane $z = 0$, (2) its unit normal vectors, and (3) the nine random unit vectors with coordinates `u1`m `v1`, and `w1` placed on the `xy`- meshgrid.

5. creates a 3×3 matrix of random angles ($\in [0, \pi]$), and rotates each normal vector to $z = 0$ about the random vector located at its same position by the corresponding `theta` angle. TO complete this part, you will need to write a function that creates a rotation matrix about a given axis (random unit vector) by an angle `theta`.

6. uses `quiver3` and `surface` to display in the same figure (1) the plane $z = 0$, (2) the nine random unit vectors with coordinates `u1`m `v1`, and `w1` placed on the `xy`- meshgrid!, and (2) the rotated unit normal vectors to $z = 0$.

Problem 2. Write a MATLAB function called `transform3D` that takes as input a vector `P`, an angle `theta`, a unit vector `A`, a vector `scale` holding 3 scaling factors (one per coordinate) and a translation vector `T`, and applies the rotation (by `theta` about `A`), scaling and translation to `P` as a single matrix multiplication. The vector `P` needs to have a 4th coordinate added to it, and the transformation matrix should be 4×4. Write a MATLAB script that rotates, scales, and translates the normal vectors to the plane $z = 0$ computed in problem 1 and displays them after they've been transformed.