Exam. Programing

IM-024. Introduction to CAD CAM Systems

Aug 13, 2011

Prof. Dr. Ing Oscar E. Ruiz S.

I am aware that this exam must be answered individually and only counting with the allowed information. I have been informed that I cannot introduce to the exam any files of any type, or electronic devices. I know that accepting / giving information of any type to other student, by any technological means during this exam collides with the disciplinary rules of EAFIT University and causes the grading of the test with 0.0 score and a report to the Academic Board of the University.

Student Name: ______________________________ **Student ID:** ______________________________

1 Indications

1. Sign and write the date of the exam.
2. Arrange all graphics in the following way:
   (a) Use the same sequence:

   ```
   axis([xmin, xmax, ymin, ymax, zmin, zmax])
   axis equal
   ```

   to dimension all Figures.
3. Read the complete exam before starting.
4. Functions or files given to you at the beginning of the exam:

   ```
   plt_axes_str(cords, L, A, B, C,label_x,label_y,label_z,label_o)
   vertebra_Geom.txt, vertebra_Topo.txt
   ```

2 Reading of the Geometry and Topology Information

Write the function:

```
function [points,triangs]=read_pseudo_vrml(geom_file_name,topo_file_name)
% INPUTS:
% geom_file_name: file with a (Nv x 3) array of real numbers.
%                 The (x,y,z) cartesian coordinates of line 'i'
%                  correspond to the vertex 'i' of a triangular shell.
% topo_file_name: file with a (Nf x 3) array of integer numbers.
%                 The (k,l,m) integer indexes in the line 'i' are the
%                 indexes of the triangle vertices in the Geometry File.
```

```
%                  The indices of the vertices start in ZERO, not in ONE.
%                  For this exercise you can assume that all the triangular
%                  face vertices are enumerated in CCW order w.r.t. the
%                  external normal vector.  You do not need to order the faces.
% OUTPUTS:
% points:         (Nv x 3 ) array or real numbers. Column 'j' corresponds to
%                  the coordinates of a planar face vertex.
% triangs:       (Nf x 3) array of integer numbers. Entry 'j' corresponds to
%                  the indices of the three vertices of face 'j'. The indices
% of the vertices start in ONE.
```

3 Filled Solid Display Function

Write the function

```
function draw_fill_solid(verts, loops, face_color)
% This function draws a polyhedron withouth holes
% whose faces are filled with a color
% INPUTS:
% verts:         (Nv x 3) or (Nv x 4) array or real numbers. Row 'i' contains
%                the coordinates x,y,x of the vertex 'i'.
% loops:         (Nf x 3) array of integer numbers. Row 'i' corresponds to
%                the indices of the vertices of triangle 'i'.
% face_color:     One of 'm', 'c', 'b', 'y', 'g', 'r', 'k'  corresponding to
%                  the color to draw the faces of the solid. If face_color is 'X'
%                  the drawing is in wireframe format with color 'k'.
```

4 Main

Program a 'main_path.m' function which performs the following actions:

4.1 Data Initialization and File Input

1. Clear the working space, figures and MATLAB prompt.

2. Define the following constants as follows:

   (a) $O_0 = [3450, 3917, 65]^T$

   (b) $Delta = [3300, 3700, 0]^T$

   (c) `WORLD` as the 4×4 identity matrix.

   (d) `AXES_SIZE=50`

(e) S_0 is the 4×4 identity matrix, but with $S_0(1:3,4) = O_0$.

3. Load the `vertebra_Geom.txt` and `vertebra_Topo.txt` files by using `read_pseudo_vrml( )`. The results of such a call must be called `points_cart` ($N\_points \times 3$) and `triangles` ($N_f \times 3$).

4. Register in `N_points` the number of vertices in the data set.

5. Open a figure and draw the object there with the `draw_fill_solid( )` function, using solid BLUE color.

6. Define the point set `points_h` ($4 \times N\_points$) in this manner: `points_h(1:3,:)  = points_cart'` and $points\_h(4,:)$ is filled with ones (1).

7. Open a second figure, to draw there the coordinate system $WORLD$ with the following parameters `AXES_SIZE`, colors 'k', 'b', 'r' and labels 'Xw','Yw','Zw','Ow'. Hold the figure ON. All subsequent drawing operations will appear in figure 2, unless said otherwise.

4.2 Translation

1. Define M_1 as the 4×4 identity matrix, but with $M_1(1:3,4) = -Delta$.

2. Calculate `points_2` by applying M_1 onto `points_h`.

3. Calculate S_2 by applying M_1 onto S_0. That is, pre-multiply S_0 by M_1.

4. Draw the object (`point_2`, `loops`) with the `draw_fill_solid( )` function, using wireframe format.

5. Draw the coordinate system S_2 using the following parameters: axes size equal to `AXES_SIZE`, axes colors black, blue, red, and labels 'X2', 'Y2' and 'Z2' and 'O2'.

4.3 Rotation

1. Define M_2 as the 4×4 matrix in this manner:

   (a) $u = [0.9501, 0.2311, 0.6068]^T$, and normalize it.

   (b) $temp = [0.4860, 0.8913, 0.7621]^T$.

   (c) $w = u \times temp$, and normalize it.

   (d) $v = w \times u$

   (e) $M_2(1:3,1:3) = [u,v,w]$, $M_2(1:3,4)$ being the null vector; $M_2(4,:) = [0,0,0,1]$.

   (f) show that R being the 3×3 rotation upper left sub-matrix of M_2 is $SO(3)$.

2. Calculate `points_3` by applying M_2 onto `points_2`.

3. Calculate S_3 by applying M_2 onto S_2.

4. Draw the object (`point_3`, `loops`) with the `draw_fill_solid( )` function, using red solid color.

5. Draw the coordinate system S_3 using the following parameters: axes size equal to `AXES_SIZE`, axes colors black, blue, red, and labels 'X3', 'Y3' and 'Z3' and 'O3'.

4.4 Projection

1. Define M_3 as the 4×4 matrix in this manner:
   (a) $M_3(1:3, 1:3) = [u, v, u+v]$, $M_3(1:3, 4)$ being the null vector; $M_3(4,:) = [0, 0, 0, 1]$.
   (b) show that Pr being the 3×3 projection upper left sub-matrix of M_3 is non invertible.
2. Calculate `points_4` by applying M_3 onto `points_2`.
3. Calculate S_4 by applying M_3 onto S_2.
4. Draw the object (`point_4`, `loops`) with the `draw_fill_solid( )` function, using wireframe format.
5. Draw the coordinate system S_4 using the following parameters: axes size equal to `AXES_SIZE`, axes colors black, blue, red, and labels 'X4', 'Y4' and 'Z4' and 'O4'.

4.5 Reflection

1. Define M_4 as the 4×4 matrix in this manner:
   (a) $M_4(1:3, 1:3) = [u, v, -w]$, $M_4(1:3, 4)$ being the null vector; $M_4(4,:) = [0, 0, 0, 1]$.
   (b) show that RF being the 3×3 reflection upper left sub-matrix of M_4 is $O(3)$ and not $SO(3)$.
2. Calculate `points_5` by applying M_4 onto `points_2`.
3. Calculate S_5 by applying M_4 onto S_2.
4. Draw the object (`point_5`, `loops`) with the `draw_fill_solid( )` function, using wireframe format.
5. Draw the coordinate system S_5 using the following parameters: axes size equal to `AXES_SIZE`, axes colors black, blue, red, and labels 'X5', 'Y5' and 'Z5' and 'O5'.
6. Carefully watch the coordinate system S_5. With the instruction `disp()` write text in which you explain why is it evident that M_4 is a reflection.