**Project 8 Part 1 Rotating a platonic solid with OpenCV**

Name: Aditya Vasantharao Period: 3 Date: 4/27/22

Did you name your file l081.cpp (Lower case L, then 081)? Yes

Does your file compile & run on terminals/jupyterhub? Yes

Did you use a rotation matrix? Yes

Did you do orthographic rendering? Yes

Did you start from the coordinates I provided for the cube?? Yes

Describe here in words all the transformations you applied to vertices, for each describe how you implemented it in your code (by multiplying with a matrix, what was the matrix, or by adding a matrix, what was that matrix… be specific):

I applied a scale transformation, 3 rotation transformations (one on each axis), and a translation transformation.

First, I multiplied each input coordinate by the following matrix:

x 0 0 0

0 y 0 0

0 0 z 0

0 0 0 1

Where x, y, z were all equal to 50, and these were the factors to scale the coordinates by.

After this, for each coordinate, I multiplied each resulting coordinate by the following matrix:

1 0 0 0

0 cos(theta) -sin(theta) 0

0 sin(theta) cos(theta) 0

0 0 0 1

Which rotates the coordinates about the x axis by angle theta.

After this, for each coordinate, I multiplied each resulting coordinate by the following matrix:

cos(theta) 0 sin(theta) 0

0 1 0 0

-sin(theta) 0 cos(theta) 0

0 0 0 1

Which rotates the coordinates about the y axis by angle theta.

After this, for each coordinate, I multiplied each resulting coordinate by the following matrix:

cos(theta) -sin(theta) 0 0

sin(theta) cos(theta) 0 0

0 0 1 0

0 0 0 1

Which rotates the coordinates about the z axis by angle theta.

The thetas for each of the 3 rotation transformations are iterated from 0 to 2pi in increments of pi/150, creating 300 frames.

After this, for each coordinate, I multiplied each resulting coordinate by the following matrix:

1 0 0 x

0 1 0 y

0 0 1 z

0 0 0 1

Where x=400, y=300, and z = 0. This translates the figure to the middle of the 800x600 canvas.

Did you use homogenous coordinates? Yes

(that allows you to combine all transformations into one matrix)

Did you combine all those transformations into one single matrix? No

If you used only one transformation matrix, what was it?

Did you name your video rotation.avi? Yes

What functions/methods from OpenCV did you use?

Mat, VideoWriter, Mat::begin<double>(), VideoWriter::open(), Size, Point

What functions/methods from OpenCV did you experiment with but ended not using?

Obs.: feel free to rotate any platonic solid, around any line.