CS 4620 Prelim 1

Tuesday 22 October 2013—90 minutes

Problem 1: Transformations (20 pts)

Consider the affine transformation on R^3 defined in homogeneous coordinates by the matrix:

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Describe this transformation as

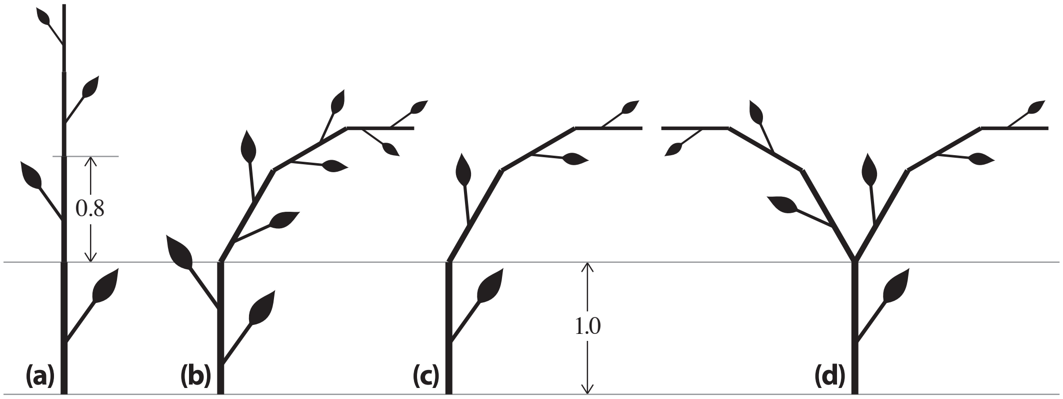
(a) a rotation about an axis through the origin followed by a translation;

(b) a translation followed by rotation about an axis through the origin; and

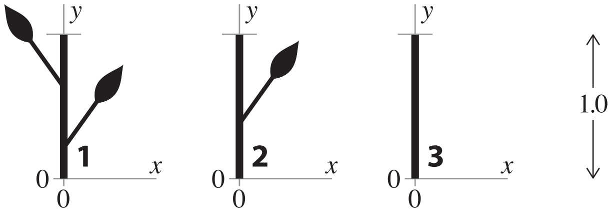
(c) a rotation about an axis not through the origin.

Specify axes using any point on the axis and a vector (of any length) parallel to the axis, and give rotation angles in the range $[0, 360°)$.

Problem 2: Hierarchies (30 pts)



Many plants have repetitive, hierarchical structures that are amenable to description using transformation hierarchies. The four plant-like 2D structures shown above are generated from the shapes #1 and #2 below:



Each of the plant structures can be drawn using a hierarchy that has the same geometry at every node and contains just one or two distinct transformations aside from the identity. That is, either all the nodes with non-identity transformations have the same transformation T, or some nodes have T_1 and the others all have T_2.

For instance, the letter E could be made using the hierarchy:

$$\begin{array}{c} I \\ | \\ T_1 \\ | \\ T_2 \\ | \\ T_2 \end{array} \qquad T_1 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

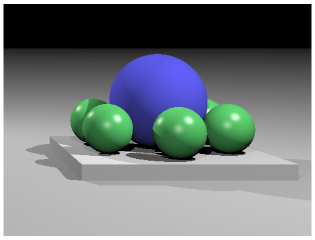
where all four nodes have the geometry #3 above.

For each of the plant structures (a) through (d), draw a hierarchy that has each node labeled with a transformation, and specify each distinct transformation as a matrix or product of matrices. Since we have not said where these plants are positioned in space, just let the transformation of the root node be the identity—this does not count as one of the transformations you're using. All rotations and scales are equal in magnitude.

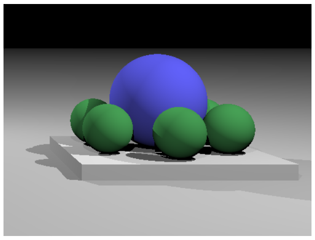
Problem 3: Ray tracing bugs (18 pts)

Below are eight renderings produced by a ray tracer in various states of disrepair. Your job is to match the bugs listed below with the corresponding incorrect outputs.

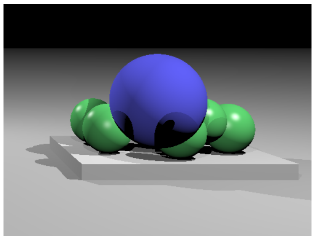
For example, the image labeled “Example” would match the bug “You forget to normalize the surface normals when you compute them for spheres.”



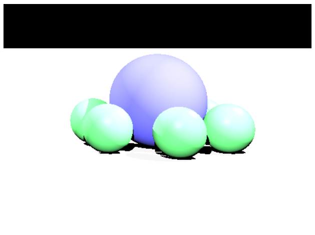
Correct



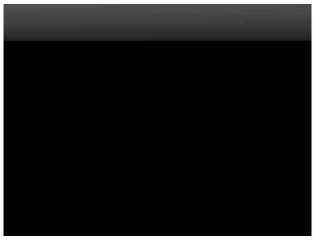
Example



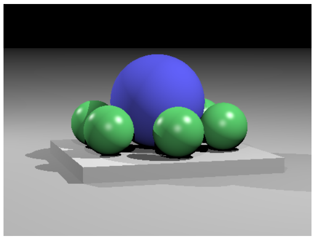
A



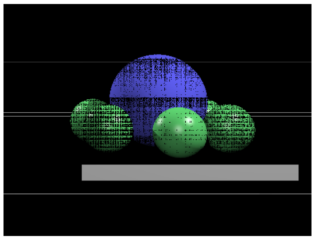
B



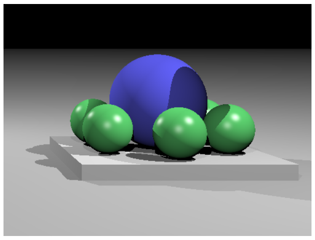
C



D



E



F

(a) You accidentally swap the viewpoint with the image plane point in the camera code.

(b) You start your shadow rays exactly at the reflection point.

(c) You forget to normalize the eye and light directions in Lambertian shading.

(d) Rather than checking all the objects to find the closest intersection along a viewing ray, your intersection method always returns the first one that is found.

(e) You accidentally use the view direction for the image plane normal, leaving the supplied image plane normal unused.

(f) When you detect a shadow ray intersection, you neglect to check that it happened between the shading point and the light.

Problem 4: Meshes (20 pts)

(a) Give an indexed triangle mesh for a triangulated unit cube, with one corner at the origin and the opposite corner at (1,1,1). Be sure to use consistent orientation so that the front sides of triangles are on the outside.

Put the vertices in dictionary order—that is, sort the triples of numbers as if they were words, ordering first by the x coordinate, then by the y coordinate, then by the z coordinate. So vertex 0 is $(0, 0, 0)$, followed by vertex 1 $(0, 0, 1)$, and so forth.

For example, a tetrahedron (not a regular one) could be specified this way:

vertices:

v	x	y	z
0	0	0	0
1	0	0	1
2	0	1	0
3	1	0	0

triangles:

t	v_0	v_1	v_2
0	0	3	2
1	0	1	3
2	0	2	1
3	1	2	3

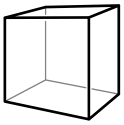
Organize the triangles so that the minimum-x face of the cube comes first, followed by the maximum-x face, then the y faces and z faces in similar order.

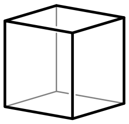
(b) In a triangle-neighbor structure, each triangle has connections to three other triangles. Give the three triangle indices that would be stored for the first triangle in your mesh.

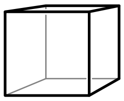
(c) In a full winged-edge structure, each edge has connections to four other edges, called head-next, head-prev, tail-next, and tail-prev, and connections to two faces, called left and right. Give the edges and vertices that would be stored for the edge from vertex 0 to vertex 1. Name faces using the indices from (a), for example "triangle 7," but name edges by the indices of their two vertices, for example "edge 2–7" (since we have not established a numbering for the edges).

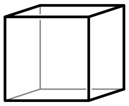
Problem 5: Viewing (12 pts)

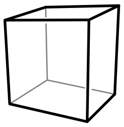
Classify each of the following views of a cube as orthographic or perspective and oblique or non-oblique. Any oblique views will have clear indications that they are oblique—you don't need to second-guess tiny details.


(a)


(b)


(c)


(d)


(e)