EECS 487 F08 Homework 4 (Revision)

Due 24 Nov 2008

November 21, 2008

1 Constructive Solid Geometry (CSG) (20 pts)

CSG uses well-defined primitives as building blocks for larger models. Each model is a tree or a directed-acyclic graph of basic blocks, transforms and set operations of union, intersection, difference etc.. For this reason, CSG models look like and integrate well with scenegraphs. Fig. 1 shows a few examples.

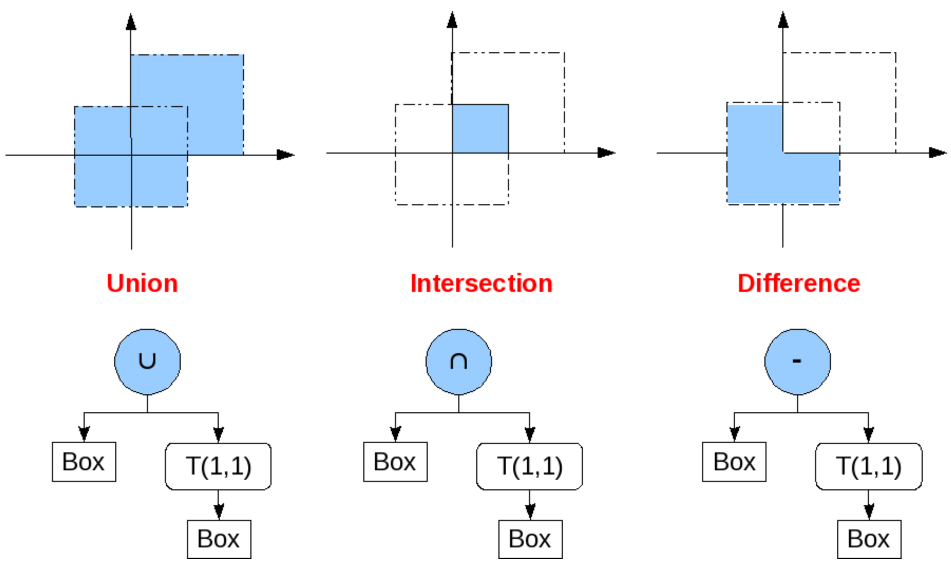

Figure 1: Examples of union, intersection and set-difference in constructive solid geometry (CSG) in 2D using a canonical box (length=2, centered at origin). The CSG subtrees are shown beneath the resulting images.

Using a canonical circle (center at origin, radius=1), canonical box (square of length=2, centered at origin) and canonical triangle (equilateral, side=1, centered at origin), construct CSG trees for the symbols (which represent basic logic gates, for those of you familiar with computer organization) shown in Fig. 2. Represent translations by $T(\Delta x, \Delta y)$, scaling by $S(s_x, s_y)$, and rotations by $R(\theta)$ (it is understood that these rotations are being performed about an imaginary $z-$axis coming out of the plane of the page/screen). Represent canonical objects with the tokens CIRCLE, BOX, and TRIANGLE without any parameters.

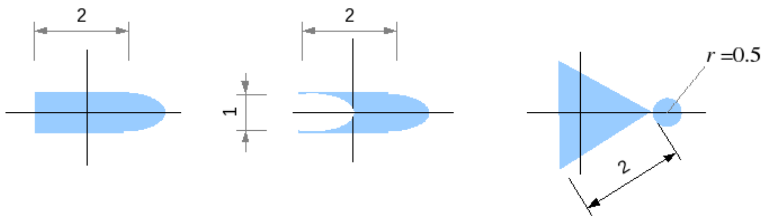

Figure 2: Symbols for which you must construct CSG trees

2 Splines (20 pts)

2.1 Quartic spline (10 pts)

Determine the expression for the basis and constraint matrices a quartic spline curve that of the form,

$$\mathbf{f}(u) = \sum_{i=0}^{4} \mathbf{a}_i u^i, \tag{1}$$

that interpolates/approximates 5 control points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ according to the following specifications:

1. The curve must interpolate $\mathbf{p}_0$, $\mathbf{p}_2$ and $\mathbf{p}_4$,
2. The slope at $\mathbf{p}_0$ must be twice the displacement from $\mathbf{p}_0$ to $\mathbf{p}_1$,
3. The slope at $\mathbf{p}_4$ must be thrice the displacement from $\mathbf{p}_3$ to $\mathbf{p}_4$.
4. The basis matrix $\mathbf{B}$ should be of the form,

$$\mathbf{B} \begin{bmatrix} \mathbf{p}_2 \\ \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{p}_1 \\ \mathbf{p}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_4 \\ \mathbf{a}_2 \\ \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_3 \end{bmatrix}. \tag{2}$$

Show the development of all expressions leading upto Eqn. (2). You may use calculators/computers to determine matrix inverses. Mention the tool.

2.2 Spline interconversion (8 pts)

OpenGL supports interpolation and approximation through *evaluators*. An evaluator is an engine which is initialized with 4 control points and it returns points on the Bezier curve formed by these four points. Because all cubic splines can be interconverted, evaluators support only Bezier curves and leave the task of expressing custom cubic splines in terms of a related Bezier curve to the programmer. Given the requirement of generating a Catmull-Rom spline with control points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, how would you go about instructing OpenGL to implement this using an evaluator?

2.3 Lagrange polynomial interpolation (2 pts)

The Lagrange-polynomial method of interpolation is way of interpolating an arbitrary set of points. What are its advantages and disadvantages?

3 Ray-tracing (30 pts)

A fundamental operation in ray-tracing is the computation of the intersection of a ray with a given object. The first test is to see if the ray intersects the object at all. If it does the next issue is to compute the reflected vector about the local surface normal. When the solid is formed using CSG or BRep, analytical formulae or methods exist to determine the surface normal.

In a real scene, canonical objects undergo a series of transforms to inherit their final properties. The transformation hierarchy is usually available to us in the form of a scenegraph. To compute the intersection of a ray with these possibly complicated objects, we need only take the ray and the model through the reverse sequence of modelview inverse transforms so that the model becomes the initial canonical object. The ray is then available in the object's model coordinates and we may reuse engines that we have previously written for determining intersection of rays with canonical objects.

Imagine that you are the artist whose task it is to create the "birth of the sandman" scene from Spiderman-3 whose clip we saw in class. Recall the frame where sandman notices his daughter's locket partially buried in the sand - the sand in the picture is created from particle simulation but the locket is a highly polished surface and must be ray-traced to reflect all the sand around it. Fig. 3 schematically depicts this. Assume the locket is well approximated by an ellipsoid with axes 3cm, 3cm and 1cm, i.e., the front face that the sandman sees is a circle of diameter 3cm. Recall that an ellipsoid is simply a scaled canonical sphere (radius=1, center at origin) with non-uniform scaling along the three axes (the same scale factor for each axis would simply give you a larger sphere). Fig. 3 shows the side-view. The locket is rotated 60^0 with respect to its model $z-$axis. With respect to the eye, the origin O of the locket model coordinate system is directly in front, 1m away and 0.5m below.

Assume the viewing ray to be of the form $\mathbf{p}(t) = \mathbf{e} + t\mathbf{g}$, where $\mathbf{e}$ (eye) is the origin of the ray, $\mathbf{g}$ is its (gaze) direction and t is the scalar distance along that direction. Assume that the ray direction $\mathbf{g}$ is available from pixel-space and perspective un-projection calculations. Determine the evolution of the viewing ray by solving the following sequence of subproblems:

1. A certain point on the viewing ray is at $t = t_1$. Now we perform a viewing transform on the entire scene where all vertices and directions (except normals) are multiplied by a modelview matrix $\mathbf{V}$. Suppose we denote the transformed point, origin and gaze direction by $\mathbf{p}'(t')$, $\mathbf{e}'$ and $\mathbf{g}'$ respectively. How are they related to $\mathbf{p}(t)$, $\mathbf{e}$, and $\mathbf{g}$? How is t' related to t?

2. What is the modeling transform that must be applied to the canonical sphere to get the ellipsoid?

3. The invisible part of the ellipsoid can be modeled by subtracting a box enclosing the invisible part in a CSG subtree. Present a CSG tree to describe the visible portion of the ellipsoid (portion above the $x - z-$plane). The only CSG primitives you have are the canonical sphere (radius=1, center at origin) and the canonical box (cube of length=2 centered at origin). Specify the box in the model coordinate system shown in Fig. 3 instead of in eye-space or in the canonical sphere's model coordinates.

4. Perform the following in the Fig. 3 model coordinate system:

   (a) What is the equation of the viewing ray in the ellipsoid's model coordinate system?

   (b) Give the procedure to determine if the viewing ray intersects the invisibility box.

   (c) Assuming the ray intersects the box, what is the intersection point in *model* coordinates?

   (d) What is the same intersection point in *world* coordinates?

5. Perform the following in the model coordinate system of the canonical sphere that forms the foundation for the ellipsoid:

   (a) What is the equation of the viewing ray?

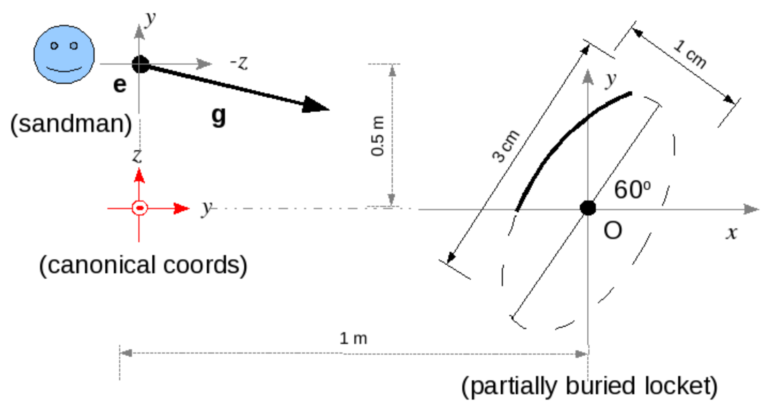

Figure 3: Partially visible ellipsoid to be ray-traced. The bold portion is the visible portion. The gray lines indicate the model $x-$ and $y-$axes. The z-axis comes out of the page. The $y-$axis for the eye and locket-model system are aligned. The $x-$axis of the locket's model coordinate system is aligned with the $-z-$axis in eye-space.

(b) Give the procedure to determine if the viewing ray in model coordinates intersects the canonical sphere.

(c) Assuming the ray intersects the sphere, what is the point of intersection in model coordinates?

(d) What is the same intersection point in *world* coordinates?

6. Having computed the intersection of the viewing ray with the tilted ellipsoid and the invisibility box and having calculated the respective points of intersection, what test would you apply to see if the viewing ray has indeed made contact with the visible portion of the ellipsoid and that ray tracing must continue?

7. Assume the ray intersects the visible portion of the ellipsoid at $t = t_0$ in model-space of its foundational canonical sphere.

   (a) What is the normal to the point of intersection in model coordinates?

   (b) What is the same normal in the world-space system?

   (c) Compute the reflection vector at the point of intersection in model-space.

   (d) Rewrite the same in world-space.

Notes:

- The canonical coordinate system for world-space are indicated in red in the figure.
- You may assume that the ray is given in eye-space
- The intersection test should fail for the buried part of the locket's front surface.
- Be careful with units. You can expect a mix of units in a real-life situation.
- You can express your answers as matrix-vector product-sequences.
- You can express square roots in symbolic form, e..g, $\sqrt{2}$ as opposed to 1.4142.... You can leave sines and cosines in symbolic form, eg., $\sin(\pi/4)$ or $1/\sqrt{2}$as opposed to 0.707...