Computer Graphics

(Comp 4190.410)

Midterm Exam: October 27, 2014

1. (20 points)

   (a) (10 points) In the Bresenham Algorithm for line drawing, show that the initial value

   $$p_0 = 2\Delta y - \Delta x.$$

   $$\begin{aligned} p_k &= \Delta x(d_{lower} - d_{upper}) = \Delta x(2mx_k - 2y_k + 2m + 2b - 1) \\ p_0 &= \Delta x[2mx_0 - 2y_0 + 2m + 2b - 1] = \Delta x[2mx_0 - 2(mx_0 + b) + 2m + 2b - 1] \\ &= \Delta x[2m - 1] = 2\Delta y - \Delta x \end{aligned}$$

   (b) (5 points) The Liang-Barsky line clipping algorithm can be extended to clipping cubic Bézier curves which are monotone along the x- and y-axis directions. Explain why. The Liang-Barsky algorithm can also be extended to the curve clipping against an arbitrary convex polygonal window under some conditions. What are these conditions?

   - The Liang-Barsky algorithm is mainly based on the fact that the line segment (to be clipped) may intersect each boundary (infinite) line at most once. This property also holds for the x and y-monotone Bézier curve segments.
   - As long as the Bézier curve segment satisfies the property of at most one intersection against each boundary line, the extension works.

   (c) (5 points) The Sutherland-Hodgman Algorithm assumes that the result of clipping a connected polygon region (against a half-plane) is again a connected region. But, this assumption fails in general. Nevertheless, there are some special cases where it works. What are these special cases.

   - Convex polygonal region
   - Star-shaped polygon when the center is contained in the window.

2. (15 points) Consider a trackball of radius 1 with its center located at the origin $(0, 0, 0)$.

   (a) (7 points) When we try to move a point $\mathbf{p} \in S^2$ to a tangential direction $\mathbf{d}$ (i.e., $\langle \mathbf{p}, \mathbf{d} \rangle = 0$), find the 3D rotation (i.e., axis and angle) that is the most reasonable to the user input $\mathbf{d}$.

   - Axis: $(\mathbf{p} \times \mathbf{d})/\|\mathbf{p} \times \mathbf{d}\|$
   - Angle: $\|\mathbf{d}\|$

   (b) (8 points) When the input direction $\mathbf{d}$ is not tangential to the trackball (i.e., the vector $\mathbf{d}$ is not orthogonal to the position vector $\mathbf{p}$), we need to project the vector $\mathbf{d}$ to a tangential vector to S^2 at $\mathbf{p}$. What is the most reasonble 3D rotation (i.e., axis and angle) corresponding to the user input direction $\mathbf{d}$?

   - Axis: $(\mathbf{p} \times \mathbf{d})/\|\mathbf{p} \times \mathbf{d}\|$
   - Angle: $\|\mathbf{p} \times \mathbf{d}\|$

3. (25 points) Consider a perspective projection of 3D points $\mathbf{x}_i$ (from $\hat{\mathbf{v}}$) to 2D points $\mathbf{x}'_i$ (on $\hat{\mathbf{n}}$):

$$\hat{\mathbf{x}}'_i = \langle \hat{\mathbf{n}}, \hat{\mathbf{v}} \rangle \hat{\mathbf{x}}_i - \hat{\mathbf{v}} \langle \hat{\mathbf{n}}, \hat{\mathbf{x}}_i \rangle = P\hat{\mathbf{x}}_i, \quad \text{for i} = 1, 2, 3.$$

(a) (10 points) The projection $\mathbf{x}'_c$ of the center point $\mathbf{x}_c = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3)/3$ is not the same as the center point $(\mathbf{x}'_1 + \mathbf{x}'_2 + \mathbf{x}'_3)/3$ of the three projections $\mathbf{x}'_1$, $\mathbf{x}'_2$, and $\mathbf{x}'_3$ onto the plane $\hat{\mathbf{n}}$. Where is the projection $\mathbf{x}'_c$ located on the projection plane $\hat{\mathbf{n}}$?

- Let $\hat{\mathbf{x}}_1 = [\mathbf{x}_1, 1]^t$, $\hat{\mathbf{x}}_2 = [\mathbf{x}_2, 1]^t$, and $\hat{\mathbf{x}}_3 = [\mathbf{x}_3, 1]^t$, then
  $\hat{\mathbf{x}}'_1 = P\hat{\mathbf{x}}_1 = [w'_1\mathbf{x}'_1, w'_1]^t$, $\hat{\mathbf{x}}'_2 = P\hat{\mathbf{x}}_2 = [w'_2\mathbf{x}'_2, w'_2]^t$, and $\hat{\mathbf{x}}'_3 = P\hat{\mathbf{x}}_3 = [w'_3\mathbf{x}'_3, w'_3]^t$,
  where $w'_i \neq w'_j$ $(i \neq j)$ in general.
  Let $\hat{\mathbf{x}}_m = [\mathbf{x}_m, 1]^t = [(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3)/3, 1]^t = [\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3, 3]^t$, then
  $\hat{\mathbf{x}}'_m = P\hat{\mathbf{x}}_m = P[\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3, 3]^t = P\hat{\mathbf{x}}_1 + P\hat{\mathbf{x}}_2 + P\hat{\mathbf{x}}_3 = \hat{\mathbf{x}}'_1 + \hat{\mathbf{x}}'_2 + \hat{\mathbf{x}}'_3$
  $= [w'_1\mathbf{x}'_1 + w'_2\mathbf{x}'_2 + w'_3\mathbf{x}'_3, w'_1 + w'_2 + w'_3]^t$.
  Consequently, $\mathbf{x}'_m = (w'_1\mathbf{x}'_1 + w'_2\mathbf{x}'_2 + w'_3\mathbf{x}'_3)/(w'_1 + w'_2 + w'_3)$.

(b) (15 points) Where is the preimage of the center point $(\mathbf{x}'_1 + \mathbf{x}'_2 + \mathbf{x}'_3)/3$ that is located on the triangle $\Delta\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ determined by the given three points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$?

- The mapping from the given triangle $\Delta\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ to the projected triangle $\Delta\mathbf{x}'_1\mathbf{x}'_2\mathbf{x}'_3$ can be represented as a perspective transformation

$$\begin{bmatrix} w'_2 & 0 & 0 \\ 0 & w'_3 & 0 \\ w'_2 - w'_1 & w'_3 - w'_1 & w'_1 \end{bmatrix},$$

  which has an inverse mapping

$$\begin{bmatrix} w'_1 w'_3 & 0 & 0 \\ 0 & w'_1 w'_2 & 0 \\ w'_1 w'_3 - w'_2 w'_3 & w'_1 w'_2 - w'_2 w'_3 & w'_2 w'_3 \end{bmatrix}$$

  Thus, the preimage of $L(s,t)$ is given as a point $(1 - \alpha(s,t) - \beta(s,t))\mathbf{x}_1 + \alpha(s,t)\mathbf{x}_2 + \beta(s,t)\mathbf{x}_3$ on the triangle $\Delta\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$, where

$$\alpha(s,t) = \frac{s w'_1 w'_3}{s w'_1 w'_3 + t w'_1 w'_2 + (1 - s - t) w'_2 w'_3}, \beta(s,t) = \frac{t w'_1 w'_2}{s w'_1 w'_3 + t w'_1 w'_2 + (1 - s - t) w'_2 w'_3}.$$

4. (20 points)

(a) (10 points) Design a recursive bottom-up algorithm for constructing an OBB tree for an open polygonal chain $\mathcal{C}$ that connects a sequence of points $\mathbf{p}_i = (x_i, y_i)$, for $i = 0, \cdots, 2^{10}$.

- At the leaf level, the OBB is a line segment connecting two adjacent points $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$, for $i = 1, \cdots, 2^{10}$. At this leaf level, there is no approximation error and the thickness of the corresponding OBB volume is zero.
- At an intermediate level, assume that the left-child OBB is along the direction of the line segment $\overline{\mathbf{p}_{(i-1)*2^h}\mathbf{p}_{i*2^h}}$ and the right-child OBB is along the direction of the line segment $\overline{\mathbf{p}_{i*2^h}\mathbf{p}_{(i+1)*2^h}}$, for $i = 1, \cdots, 2^{(k-h)}$.
- The parent OBB is along the direction of the line segment $\overline{\mathbf{p}_{(i-1)*2^h}\mathbf{p}_{(i+1)*2^h}}$, and the minimum bounding box containing the eight cornet points of the left and right children OBBs.

(b) (5 points) Discuss the relative advantages and disadvantages of the bottom-up OBB-tree construction against the top-down approach you have taken for Programming #3-1.

- More efficient than the top-down construction.
- But not as tight as the OBBs generated by the top-down approach.

(c) (5 points) For an AABB-tree construction for the polygonal chain $\mathcal{C}$, do the two approaches (i.e, top-down and bottom-up approaches) generate different AABB trees? Why?

- The two approaches generate the same AABBs.
- The minimum AABB containing two children AABBs is in fact the minimum AABB for the corresponding subchain.

5. (20 points) Fill in the blanks in the following OpenGL program segments