UU Graphics

academic year 2013/14 – 4th period

Theoretical Assignment #5: Rasterization & Shading

June 10 2014

Assignment #1: *Rasterization*

You are given a triangle with projected vertices

$$\mathbf{p}_1', \mathbf{p}_2', \mathbf{p}_3' \in \mathbb{R}^2.$$

These points are already given in screen coordinates [pixels]; however, the points can be placed anywhere – no guarantees. Further, your screen consists of $w \times h$ pixels.

Develop an algorithm, in pseudo-code, that rasterizes the triangle to the screen. The algorithm should be asymptotically optimal in the sense that the processing cost for a triangle are $\mathcal{O}(k)$ if the triangle has k fragments within the screen area (in other words, do not generate fragments outside the screen, or simply reject them after generation; this could have arbitrarily high run-times!).

Your solution does not need to make specific optimizations (integer arithmetic, incremental calculations etc.).

Hint: We did not talk about this in the lecture yet – so be creative! Any correct solution with $\mathcal{O}(k)$ runtime is acceptable.

Assignment #2: *No overpaint allowed! – a variant of the painter's algorithm.*

Imagine we need to create a variant of the painter's algorithm where no overwriting is allowed: Given the projection of n triangles into 2D, we have to cut the scene into smaller triangles such that invisible area is never drawn (in other words, if area overlaps with another triangle, it must be removed; the output are still triangles). Such an algorithm might for example be necessary to drive a plotter (a devise that draws wire-frame drawings of a 3D scene and cannot erase anything it has ever drawn).

Prove a quadratic lower-bound for the worst-case complexity. Show that there exists scenes with $\mathcal{O}(n)$ input triangles that create $\mathcal{O}(n^2)$ output triangles.

Hint: One can construct a scene (more specifically, a family of similar scenes with an arbitrary number of triangles) such that the number of required pieces grows quadratically.

Assignment #3: *The real painter's algorithm, and why it can be slow…*

Now we permit overdraw. Consider again a "correct" painters algorithm that cuts triangles into smaller pieces until the triangles can be brought into a fixed order such that painting them in that order makes sure that the parts always overwrite parts that are further away.

Prove that such an algorithm has also a quadratic worst-case complexity. More specifically, show that at least one scene exists, consisting of n triangles where $\mathcal{O}(n^2)$ sub-triangle (pieces) must be generated before such an ordering can be obtained.

Hint: One can construct a scene (more specifically, a family of similar scenes with an arbitrary number of triangles) such that the number of required pieces grows quadratically.

Remark: Assignment 3 is more difficult than assignment 2 – the example is not as easy to find as before.

Assignment #4: *Normals and back-face culling*

a) Calculate the unit normal vector of the following triangles! Use the CCW rule – points are given in counter-clockwise direction if viewed from the outside. The normal should point outside (towards the viewer) when looking at the visible side.

$$t_1 = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$$

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad \mathbf{p}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \qquad \mathbf{p}_3 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$t_2 = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$$

$$\mathbf{q}_1 = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}, \qquad \mathbf{p}_2 = \begin{pmatrix} 5 \\ 5 \\ 0 \end{pmatrix}, \qquad \mathbf{p}_3 = \begin{pmatrix} 7 \\ 2 \\ -1 \end{pmatrix}$$

b) Our rendering system implements back-face culling – triangles for which the normal vector (orthogonal to the triangle plane). The camera is located in the origin and looking down the positive z-axis. Which of the two following triangles is visible after CCW-back-face culling?

$$t_1 = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$$

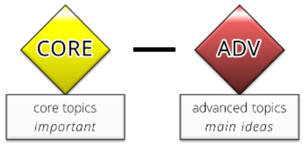
$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad \mathbf{p}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \qquad \mathbf{p}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$t_2 = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$$

$$\mathbf{q}_1 = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}, \qquad \mathbf{p}_2 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \qquad \mathbf{p}_3 = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

Explain briefly why.

Assignment #5: *Diffuse shading*



Consider the diffuse ("Lambertian") shading model. Explain why the brightness of the surfaces decreases with $\cos\theta$, where θ is the angle between the incident light direction $\mathbf{l}$ and the surface normal $\mathbf{n}$.

Hint: Consider a parallel ray bundle that hits a small piece of surface. Then vary the incident angle θ and describe what happens (does not need to be formally strict; formally strict would be very much rated "red").