LUNDS TEKNISKA HÖGSKOLA Institutionen för datavetenskap

Example Exam Questions
EDAN35
High-Performance Computer Graphics

Please answer in English.
Please answer each question on a separate page.
Dictionaries for English are allowed.
Electronic calculators are **not** allowed.
This is a closed book exam, so no other resources are allowed to be used. **Grading**: The maximum score is 6.0. A score of 3.0 or above is needed to pass.

---

1. Rasterization
   a) The rasterization equation contains two factors called **depth complexity**, denoted d, and **overdraw**, denoted o. Explain what these two terms describe, and how they are related in theory. *(0.3p)*
   b) Assume we have a number of front facing, connected, non-overlapping triangles lying in a shared plane. For robust rasterization it is important that each pixel is written to at most one time. How can you solve that with edge functions (all cases must be handled)? *(0.4p)*
   c) How can edge functions be used to test whether a tile overlaps a triangle? Explain why this works. *(0.4p)*
   d) Derive a formula for an edge function from a = (ax, ay) to b = (bx, by). It is not enough to simply print down the formula. *(0.4p)*
   e) Define barycentric coordinates, and explain how and why they can be computed with edge functions. *(0.4p)*
   f) If you have a set of connected triangles (sharing edges, and vertices) in the plane, a sample point should only be covered by one triangle. What happens if a sample point and a triangle vertex (shared by many triangles) coincides? Describe how the problem is sovled. *(0.4p)*

2. Texture
   a) Texture caching obvisouly helps a lot in terms of reducing bandwidth. Two different texture filtering methods are mipmapping and nearest neighbor. Explain and argue which of these works best with texture caching. *(0.4p)*

3. Buffer Compression
   a) There are several differences between a texture compression/decompression algorithm, and buffer (e.g., color and depth) compression and decompression algorithms. Mention two important ones, and argue why these are important. *(0.4p)*
   b) Compressing the depth buffer is one way of reducing bandwidth. In contrast to texture compression, depth compression cannot be fixed rate. Explain how that is solved in a depth compression architecture. *(0.4p)*
   c) Argue why it is expected to be easier to compress a tile of depth values that a tile of color values. *(0.4p)*

---

d) Try to invent an efficient color buffer compression scheme. *(0.4p)*

e) In texture decompressors, it is common to expand a number, which is represented using few bits, to a number using more bits. Assume that a 4-bit number, x, is to be expanded into an 8-bit number. How would you do that efficiently, and why is that the correct way to do it? *(0.4p)*

4. Memory Bandwidth

a) Make some reasonable assumptions about the number of bits per component of the depth buffer, color buffer, and textures. Assume that we are constructing a GPU with a target depth complexity of d = 6, and that we use only bilinear filtering but with two textures per pixel. We have already implemented a texture cache, with cache miss ratio m = 0.25. Which one of the units that use bandwidth to memory is in most need of improvement (assuming that performance is directly proportional to memory bandwidth usage)? Use theoretical models to motivate your answer. *(0.4p)*

5. Culling

a) Explain how Zmin-culling works, and what kind of modifications are needed to an architecture not capable of either Zmin-culling and Zmax-culling. *(0.4p)*

b) Object culling using "occlusion queries" can reduce the amount of com- putations and bandwidth usage. How does that work? *(0.2p)*

c) Assume that you have an architecture that can switch (per tile) from using Zmin-culling to Zmax-culling and vice versa, i.e., you cannot use Zmin- culling and Z-max culling at the same time. When (and why?) would it be optimal to switch from Zmin to Zmax? How can that be determined? *(0.4p)*

6. Anti-Aliasing

a) When determining the color of a pixel, we often have to use point sampling in order to get practical algorithms. How would we ideally solve the problem? Argue why that is hard.*(0.4p)*

b) Mention three different types of aliasing in computer graphics. *(0.4p)*

c) Assume for a while that the human is only sensitive to aliasing on horizontal edges. Construct a sampling scheme that helps reduce aliasing as much as possible in this case. *(0.4p)*

7. Architecture

a) Explain the concept of unified shaders, and the advantages of them. *(0.4p)*