LUNDS TEKNISKA HÖGSKOLA Institutionen för datavetenskap

Exam
EDAF80 Computer Graphics : Introduction to 3D

2021–10–28, 08.00–13.00

Please answer in English.
Please answer each question on a separate page.
Dictionaries for English are allowed.
Electronic calculators are **not** allowed. **Grading**: The maximum score is 6.0. A score of 3.0 or above is needed to pass.

---

1. Transformations
   a) Given the three matrices **A**: translation by the vector $\mathbf{v} = (4, 2, 1)$, **B**: rotation 90 degrees around the y-axis and **C**: a non-uniform scaling with 5 in x, 8 in y and 3 in z. Given the point $P = (1, 1, 1, 1)$, what is the location of $P' = \mathbf{BCA}P$? *(0.3p)*
   b) Why can normals not necessarily be transformed with the same matrix as points? How should they be transformed? *(0.2p)*
   c) What is backface culling and how is it determined? *(0.2p)*
   d) Describe the transform **M** needed to move the triangle A to A′ and the transform **N** that moves B to B′ in Figure 1 below. You can use the syntax used for the Node class in Assignment 1, e.g. Translate(x, y), RotationZ(θ), Scale(x, y). Include in your answer the order of operations. Assume that the origin is placed at the lower left corner of triangle A, and all operations operate in the XY plane. *(0.3p)*

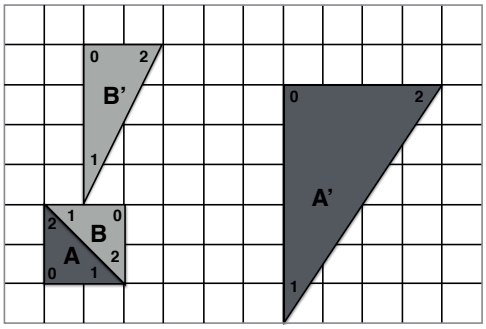

Figure 1: Transform from A to A′ and B to B′

2. Shading, GLSL
   a) Explain what Flat, Gouraud and Phong Shading are by comparing how they are calculated. *(0.3)*
   b) Vectors that are output from the Vertex Shader are used as inputs to the Pixel Shader. What must be done to these vectors before they are used for lighting in the Pixel Shader, and Why? *(0.2p)*
   c) Describe what the GLSL type qualifiers, `uniform` and `const` do, and how they are used in GLSL. What is the difference between them? *(0.2p)*
   d) Describe which vectors are used as input to the GLSL refract(...) function. *(0.2p)*
   e) What variable must a Vertex Shader set, and what does it represent? *(0.1p)*

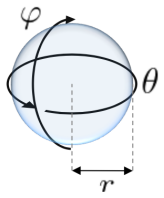

Figure 2: A parametric representation of a sphere.

3. Normal Mapping, Reflection Mapping and Homogeneous coordinates

   a) Given the parametric representation of a sphere shown in the figure, with a radius of (r = 1), the parametric form is given by

$$s(\theta, \varphi) = \begin{bmatrix} \sin\theta\cos\varphi \\ -\cos\varphi \\ \cos\theta\sin\varphi \end{bmatrix},$$

   where $\theta \in [0, 2\pi]$ and $\varphi \in [0, \pi]$. We can construct a tangent space at the point, s, where the tangent, $\mathbf{t}$, and binornal, $\mathbf{b}$, are

$$\mathbf{t} = \begin{bmatrix} \cos\theta\sin\varphi \\ 0 \\ -\sin\theta\sin\varphi \end{bmatrix} \mathbf{b} = \begin{bmatrix} \sin\theta\cos\varphi \\ \sin\varphi \\ \cos\theta\cos\varphi \end{bmatrix}.$$

   how do we compute the normal, $\mathbf{n}$? *(0.1p)*

   b) Given the normal, $\mathbf{n}$, is

$$\mathbf{n} = \begin{bmatrix} \sin\theta\sin\varphi \\ -\cos\varphi \\ \cos\theta\sin\varphi \end{bmatrix}.$$

   Compute the position and tangent space at the parametric value $(\theta, \varphi) = (\frac{3\pi}{4}, \frac{\pi}{2})$ *(0.2p)*

   c) For the point, $\mathbf{s}$, and normal, $\mathbf{n}$, derived in the previous question, we perform a texture lookup into a normal map texture. If the texture lookup returns the color (25, 200, 100), calculate the perturbed normal in object space? *(0.3p)*

   d) Reflection mapping is an approximation of true reflections. Why? Motivate your answer. *(0.2p)*

   e) What are homogeneous coordinates and why are they used in the graphics pipeline? *(0.2p)*

4. Miscellaneous

   a) The OpenGL Shading Language (GLSL) includes the `texture(S,I)` command. Give two different examples of the command's usage, and explain what the two parameters are for each usage of the command. *(0.2p)*

   b) How do you implement reflection mapping that reflects a background that surrounds a scene using a skybox? Explain the GLSL functions needed in the Pixel Shader, their inputs and outputs, and any other setup. *(0.3p)*

   c) How would you attach the camera to an object in a scene? For example, a camera that is attached to a planet in a solar system and moves around with the planet as it orbits the sun. Explain the steps that would need to be taken to achieve this within the bonobo framework used in the first assignment to make a solar system taking into account the effect of hierarchical modelling. Include the values that are needed, how they are computed, and how they are used to set up the camera for rendering. *(0.3p)*

   d) Mention at least two effects that are hard to reproduce in real-time graphics, but needed for high quality offline rendering. Motivate your answer. *(0.2p)*

5. Graphics Pipeline
   a) Give the three main stages of the Graphics Pipeline and describe what happens to a triangle as it passes through these stages. Include in your answer what inputs and outputs are generated. *(0.3p)*
   b) What are the most computationally costly parts of the graphics pipeline? Motivate your answer. *(0.3p)*
   c) Compare how rasterization and ray tracing determine visibility per pixel. Include in your answer the difference in how pixels and triangles are processed in the two algorithms. *(0.4p)*

6. General
   a) A vertex is transformed from object space to screen space by the graphics pipeline. A step in that transformation is the perspective divide. Where in the pipeline does it happen, and what does it do? *(0.2p)*
   b) Toon shading is a technique where the silhouette is enhanced, and a three-dimensional object appears to look flat, similar to a cartoon character. One example is shown in the Figure below. Notice the black border along the silhouette, the lack of 3D perception and the constant-colored specular peak.
   Design a pixel shader in GLSL that performs this effect. Exact syntax is not required, but it should be clear which inputs the shader needs and how the effect is obtained. Motivate your answer. *(0.3p)*

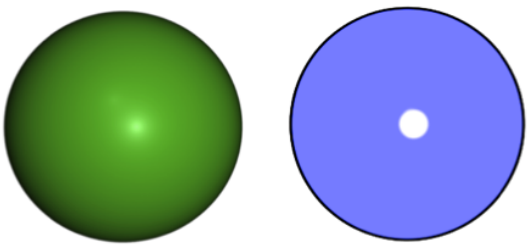

Figure 3: Two different shaders are applied to a sphere.

   c) Shadows add realism to rendered images. A shadow appears at a point P if there is an object O between P and the light source. Discuss how you could simulate shadows, both in a ray tracer and in a rasterization pipeline. Focus mostly on the latter.
   Hint: In a rasterizer, one could first place a camera at the light source and render the scene from that perspective. For each pixel, both a color and/or a depth value can be obtained. In a second pass (when rendering from the real camera), use information from the first rendering pass to obtain shadows. *(0.5)*

The end.