Exam Paper: Computer Graphics (INBCG-08) 2016

Date: April 1, 2016 Time: 9:00am – 12:00noon

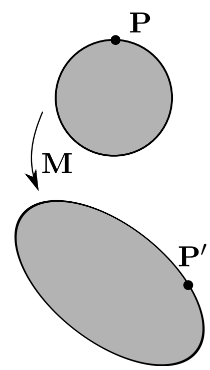
Instructions:

- Fill in your name, student number, and date (top right corner) on each of the answer sheets that you hand in.
- Write clearly and in English. Illegible parts will not be graded.
- In derivations, describe *all* the steps needed to arrive at your result.
- Use pseudo-code and sketches/diagrams where appropriate.
- It is recommended that you read all questions in full first.
- The use of books, lecture/lab slides, notes, other similar material, and electronic devices is *not allowed* during the exam.
- Zero tolerance cheating policy applies.

Structure:

- This exam paper consists of 9 questions on 3 pages.
- You have 3 hours to answer all the questions.
- Some questions are split into sub-questions, each with its own point maximum (shown in bold).
- The total number of points P_{max} is 100.
- The exam grade E is arrived at as follows: $E = P/10$, where P is your exam score.
- Your final course grade depends also on your grades for practical assignments.

Question 1: Mapping tangent and normal vectors



Suppose that $\mathbf{M}$ is a 4×4 matrix used in the OpenGL (or similar) graphics pipeline to map 3D geometry (e.g. from local coordinates to eye (or camera) coordinates). A point $\mathbf{P}$, represented in homogeneous coordinates as a column vector, on a shape is thus mapped to $\mathbf{P}' = \mathbf{MP}$.

(a) (**3 points**) How is a tangent vector $\mathbf{T}$ at $\mathbf{P}$ mapped, using matrix multiplication, so that $\mathbf{T}'$, the image of $\mathbf{T}$, is tangent to the mapped shape at $\mathbf{P}'$?

(b) (**7 points**) How is the (unit) normal vector $\mathbf{N}$ at $\mathbf{P}$ mapped, using matrix multiplication, so that $\mathbf{N}'$, the image of $\mathbf{N}$, is normal to the mapped shape at $\mathbf{P}'$?

(c) (**3 points**) Describe at least one transformation which is not a translation for which the answer to (a) and (b) is one and the same matrix, and at least one transformation for which the respective matrices are different.

In (a) and (b), derive the appropriate matrix from $\mathbf{M}$. In (c), explain why your answers lead to the same or different matrices in (a) and (b), respectively.

Question 2: Perspective projection

There are two major options when it comes to projecting geometry from 3D to 2D: orthographic and perspective projection.

(a) (**2 points**) Given a point $\mathbf{P}$ and a plane π, how do we project $\mathbf{P}$ onto π using orthographic projection? Include a sketch.

(b) (**2 points**) Assuming the canonical position, i.e., that π coincides with the (x, y) plane, derive the 4×4 projection matrix (in homogeneous coordinates) for the projection you described in (a).

(c) (**2 points**) Given a point $\mathbf{P}$, a plane π, and a centre of projection $\mathbf{E}$, how do we project $\mathbf{P}$ onto π using perspective projection? Include a sketch.

(d) (**4 points**) Assuming the canonical position, i.e., that $\mathbf{E}$ is at the origin and that π is given by $z = d$, derive the 4×4 projection matrix (in homogeneous coordinates) for the projection you described in (c).

(e) (**2 point**) Describe two contexts, one in which orthographic projection is preferable, and one in which perspective projection is preferable. Explain your answer.

Question 3: Marching squares

(**10 points**) Marching squares is an algorithm for polygonising implicitly given curves. Describe the algorithm using annotated pseudo-code and sketches. You may assume that we want to turn the curve given by $f(x, y) = l$, for some value of l, into a poly-line over a grid of $n \times m$ square cells. It is not necessary to explicitly enumerate all the possible cases for a single square arising in the algorithm.

Question 4: Colour

Colour plays a major role in computer graphics.

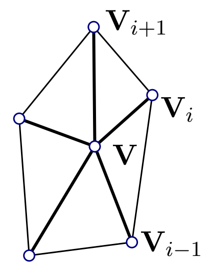
(a) (**3 points**) Describe metamerism in terms of light spectra and the human visual system. Use a sketch.

(b) (**3 points**) How do we exploit metamerism in computer graphics?

(c) (**2 points**) Describe the CMY colour space, why it is used and where.

(d) (**2 points**) Why and how does the CMYK colour space improve on (c) in practice?

Question 5: Vertex normals and shading



In some cases, meshes come with normals associated with vertices. In others, normals need to be computed to facilitate mesh shading.

(a) (**5 points**) Give a formula for computing the normal $\mathbf{N}$ associated with an inner vertex $\mathbf{V}$ of valency n of a uniform triangular mesh.

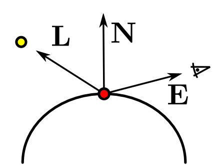
(b) (**4 points**) Assume that the triangles in the mesh vary widely in terms of their shapes and areas. How can this be taken into account to improve your formula from (a)?

(c) (**2 points**) Describe, in brief, what role is played by mesh normals in shading models.

Make sure that all the quantities used to compute $\mathbf{N}$ in (a) and (b) are described exclusively in terms of the vertex $\mathbf{V}$ and its cyclically ordered one-ring neighbourhood of vertices $\mathbf{V}_i$, $i = 0 \ldots, n-1$.

Question 6: Blinn-Phong illumination model

Blinn proposed a modification to the specular component of the Phong illumination model in order to improve its efficiency in certain scenarios.



(a) (**4 points each**) Given normal $\mathbf{N}$, direction $\mathbf{L}$ to a point light source, and direction $\mathbf{E}$ to the eye (all normalised) at a point on a surface, describe and give the formula for evaluating the specular component $\mathbf{I}_s^{\text{Phong}}$ and $\mathbf{I}_s^{\text{Blinn}}$ of the Phong and Blinn illumination model, respectively, at that point. Build on top of the sketch provided. If additional vectors are needed, explicitly derive them from the three given ones.

(b) (**2 points**) Describe two scenarios: one which is better suited, in terms of efficiency, for the Phong model and one for the Blinn model. Justify your answer.

(c) (**2 points**) Describe what the specular component models from the real world and how.

Question 7: Distributed ray-tracing

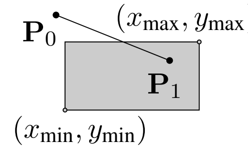
Standard ray-tracing is able to produce very good renderings, but it cannot model some effects that are necessary to achieve results closer to photo-realism.

(a) (**4 points**) Describe what distributed ray-tracing is and what implementation changes are necessary, using a standard ray-tracer as a starting point.

(b) (**2 points each**) Describe three effects that can be modelled using distributed ray-tracing. For each, describe the effect using a sketch, why it is not modelled by standard ray-tracing, and how distributed ray-tracing achieves the effect.

Question 8: Line clipping

Line clipping is one of the operations performed in the graphics (OpenGL) pipeline.



(a) (**3 points**) Describe a situation (using a sketch) where skipping line clipping would produce the wrong rendered result. For what other reason is clipping performed?

(b) (**6 points**) Describe the Cohen-Sutherland line-clipping algorithm in 2D screen space. Assume that the line-segment end-points are $\mathbf{P}_0$ and $\mathbf{P}_1$, and that the rectangular screen spans from $(x_{\min}, y_{\min})$ to $(x_{\max}, y_{\max})$.

(c) (**2 points**) What changes need to be made to (b) to extend the algorithm to handle clipping in 3D against the viewing frustum?

Question 9: Shaders

Shaders lie at the heart of the graphics pipeline.

(a) (**3 points**) In the context of modern GPUs and the modern graphics pipeline, what is a shader?

(b) (**2 points each**) Describe the main functionality of the vertex and fragment (pixel) shader, including their typical input and output.

(c) (**4 points**) Assume that no other shaders (on top of (b)) are used. What other major stages of the graphics pipeline take place between the two shaders described in (b)? How do these stages process the output of the vertex shader into the input of the fragment shader?

Include a diagram of the whole graphics pipeline (or relevant parts thereof).