IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

BEng Honours Degree in Computing Part III
BEng Honours Degree in Electronic and Information Engineering Part III
MEng Honours Degree in Electronic and Information Engineering Part III
MEng Honours Degree in Electronic and Information Engineering Part IV
MEng Honours Degree in Mathematics and Computer Science Part IV
BEng Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degrees in Computing Part III
MSc Advanced Computing
MSc Computing
MSc in Computing (Specialism)
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant assessments for the Associateship of the City and Guilds of London Institute

PAPER COMP60005=COMP96009=COMP96010

GRAPHICS

Wednesday 24 March 2021, 10:00
Duration: 140 minutes
Includes 20 minutes for access and submission

Answer ALL TWO questions
Open book assessment

Paper contains 2 questions

1 Ray tracing

a Setting the stage for ray tracing.

i) What is a **vertex** in the context of a Computer Graphics shading pipeline? (choose one)

- ☐ **pixel candidate, i.e., a list of colours**
- ☐ **whirling mass of fluid or air, especially a whirlpool or whirlwind.**
- ☐ **node in a polyhedral mesh**
- ☐ **small part of the overall shader code**
- ☐ **picture element, i.e., the final output colour for a pixel in a render window on a physical screen**
- ☐ **a certain number of operations**

ii) What is a **fragment** in the context of a Computer Graphics shading pipeline? (choose one)

- ☐ **pixel candidate, i.e., a list of colours**
- ☐ **a triangle that is generated when splitting up more complex geometry**
- ☐ **node in a polyhedral mesh**
- ☐ **small part of the overall shader code**
- ☐ **picture element, i.e., the final output colour for a pixel in a render window on a physical screen**
- ☐ **a certain number of operations**

iii) What is a **pixel** in the context of a Computer Graphics shading pipeline? (choose one)

- ☐ **pixel candidate, i.e., a list of colours**
- ☐ **whirling mass of fluid or air, especially a whirlpool or whirlwind.**
- ☐ **node in a polyhedral mesh**
- ☐ **small part of the overall shader code**
- ☐ **picture element, i.e., the final output colour for a pixel in a render window on a physical screen**
- ☐ **a certain number of operations**

iv) Given the projection of a polyhedral mesh on an image plane as shown in Figure 1,

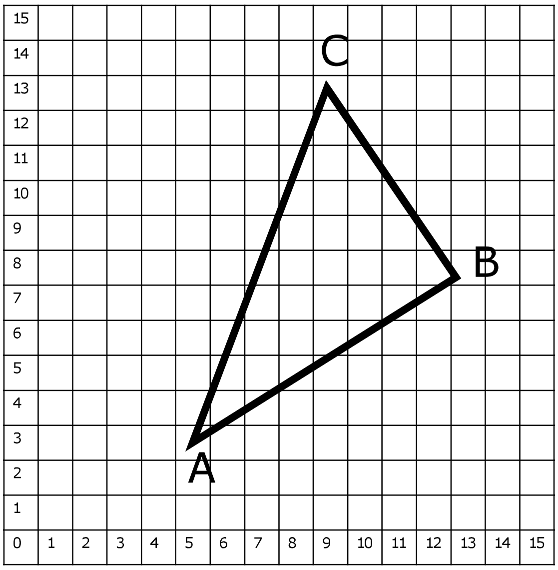

Fig. 1: A polyhedral mesh projected onto a low resolution image plane. The numbered pixels are part of the plane.

A) how many elements are processed by the **object vertex** shader?

B) how many elements are processed by the **object fragment** shader?

(hint: standard shading pipeline, not render-to-texture)

v) What changes if the scene is rendered through a render buffer and then processed through a screen-aligned quad? (render-to-texture as used for coursework 5 and 6)

A) how many elements are processed by the **final vertex** shader?

B) how many elements are processed by the **final fragment** shader?

(hint: render-to-texture pipeline, not standard shading pipeline)

b A vertex shader calculates the viewing directions for the vertices of the screen-aligned render quad in Figure 1 to be

- $(0,0) : (-1,-1,-1)$
- $(0,15) : (-1,1,-1)$
- $(15,0) : (1,-1,-1)$
- $(15,15) : (1,1,-1)$

i) What's the viewing direction at pixel $(7,7)$?

ii) What's the viewing direction at pixel $(3,3)$?

iii) The triangle coordinates in world space are approximately

- $A : (-2,-4,-6)$
- $B : (4,0.1,-2)$
- $C : (2,4,-1)$.

What are the coordinates of the intersection point with the triangle ABC of the ray starting at pixel $(7,7)$ if this pixel happens to be located at the world origin $(0,0,0)$

iv) Name the most efficient algorithm that we have discussed that can be used to intersect any given viewing ray with any given triangle.

v) Outline an efficient algorithm that can be used to intersect any given viewing ray with any given triangle. You may use pseudo code.

c i) Tick all illumination effects that are feasible with ray tracing but challenging with conventional rendering of projective geometry.

☐ **rendering of complex scenes with high simplex count**

☐ **shadows including soft shadows**

☐ **Phong shading**

☐ **global illumination**

☐ **caustics**

☐ **Gouraud shading**

☐ **refractions**

☐ **flat shading**

☐ **diffraction**

☐ **reflections**

☐ **animations**

ii) Illustrate or describe how umbras and penumbras (soft shadows) can be approximated with the ray tracing algorithm.

iii) Given floor plane intersection points $a - j$ as shown in Figure 4 and three shadow rays per intersection a, e, and j, what is the resulting shadow attenuation that has to be deducted from these intersection's colour values?

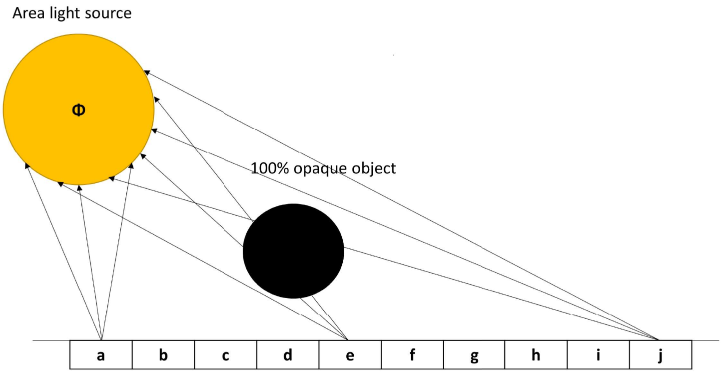

Fig. 2: A very simple soft-shadow implementation.

iv) Name one disadvantage of calculating *soft shadows*.

v) Naïve shadow generation will lead to so called "self-shadowing artefacts". Describe the reason for these artefacts and possible corrective measures.

vi) Tick all values computed during ray tracing and intersection calculation that are suitable ray termination criteria.

- ☐ **encoding a special material value**
- ☐ **ray depth (how often did the ray already hit a surface)**
- ☐ **threshold on contribution towards final pixel colour**
- ☐ **number of multiply-add operations**
- ☐ **'ray exits the scene', no further intersections found**
- ☐ **screen time of the user**

vii) What data structures can be used to reduce the number of necessary primary rays, thus to speed up the rendering process?

- ☐ **axis-aligned bounding regions**
- ☐ **object aligned bounding regions**
- ☐ **arbitrary convex regions**
- ☐ **bounding spheres**
- ☐ **regular grid**
- ☐ **adaptive grids/octtrees**
- ☐ **Binary Space Partition Trees**

viii) How many primary rays could have been skipped in Figure 1 if a half-space test with the object's projected and axis-aligned bounding box would have been performed before processing the ray?

ix) What's the expected speed-up factor assuming that a half-space test per pixel requires one multiply-add operation and processing a ray intersection including illumination requires four multiply-add operations. Please round to one digit after the comma.

The three parts carry, respectively, 25%, 30%, and 45% of the marks.

2 Illuminating and displaying textures

Raster images can be used to enhance the details of surface meshes. The process of virtual wrapping of images around polyhedral geometry is called *texture mapping* (part a). Textures are commonly used together with *illumination* models (part b). Resulting pixel *colours* have to be adjusted to yield realistic results on a physical screen (part c).

a i) A) Sketch a common graphics pipeline.

B) Identify the part in the graphics pipeline during which texture mapping is performed.

C) Which coordinate transformations are required to convert from the identified part in the graphics pipeline to two-dimensional texture space? Name the used coordinate axes (x, y, z), (u, v, w), and (s, t, p) according to the commonly used naming convention for texture mapping.

ii) Textures are usually sampled from a normalised coordinate space mapped to $(0 <= s <= 1, 0 <= t <= 1)$. Select four common schemes to handle out-of-bounds samples when texture coordinates are sampled $(s < 0 || s > 1, t < 0 || t > 1)$.

- ☐ **Clamped, last row in s an t are repeated**
- ☐ **Static/constant color**
- ☐ **Repeated, texture repeated**
- ☐ **Texture indempodenting**
- ☐ **Mirrored, texture mirror repeated.**
- ☐ **Random colours**
- ☐ **Cross correlation**
- ☐ **Texture Splatting**

iii) When texture coordinates are added as additional parameters to vertices, these parameters are interpolated during the rasterization stage like any other payload parameter (colour, normal, etc.).

A) Will such an interpolation result in correct texture coordinates for all fragments? Justify your answer.

B) Name two different and valid schemes to convert screen space fragment coordinates to texture coordinates.

iv) A point p is defined as $p = (1,3,3)$ and a point r as $r = (4,0,3)$ **in world coordinates**. p and r are projected to $p' = (1,3,3)$ and $r' = (1,1.5,3)$ on a viewing plane, which is defined **in world coordinates** by a point $a = (1,1,0)$ and its normal vector $n = (1,0,0)$. Focal length is $f = 1$ and the position of the eye point is at $c = (0,2,3)$ **in world coordinates**.

p carries the texture coordinates $p_{st} = (1,0)$ and r carries the texture coordinates $r_{st} = (0,0)$. Calculate the correct texture coordinates for a point q that is on a line between p and r. The projected point q' yields the texture coordinates $q'_{st} = (0.25,0)$ after rasterization. What's the perspectively corrected, normalised texture coordinate for q_{st}?

You might find it helpful to draw a sketch in 2D of a projection of the scene.

b You are given a rectangle $ABCD$ in screen space that has been textured with a texture from part a. Each vertex of the rectangle has now an RGB colour and texture coordinates, to be used for Gouraud shading:

Vertex	Screen Coordinates	(R,G,B) Colour	texture coordinates
A	(0,1)	(128,128,0)	(0,0)
B	(8,1)	(250,0,0)	(1,0)
C	(8,6)	(250,0,250)	(1,1)
D	(0,6)	(0,0,250)	(0,1)

i) Express point E = (6, 1) as an affine combination of points A and B.

ii) Express point F = (1, 6) as an affine combination of points C and D.

iii) Express point G = (4, 4) as an affine combination of points E and F

iv) If Gouraud shading is used at point G, what should its RGB colour and texture coordinates be at points E, F and G before perspective division?

v) Now you would like to replace Gouraud shading with Phong shading. Outline the steps that are necessary to achieve this. Do not recalculate iv)! Please use bullet points.

vi) What are the consequences for Phong illumination if the light source and the viewer are placed infinitely far away from the scene? Assume a flat surface.

c A customer has very strict requirements regarding colour fidelity of textures on their special monitors. Thus you need to work with various colour spaces and monitor specifications.

i) Assuming an HSV colour representation, which of hue, saturation and value are affected by each of the following operations?

A) Mixing white with a coloured paint

B) Replacing an RGB colour (r, g, b) with (sr, sg, sb)

Two monitors $\mathcal{M}_1$ and $\mathcal{M}_2$ have been calibrated and their three LEDs have the following CIE diagram coordinates:

$$\mathcal{M}_1 = \begin{array}{lcc} & \text{x} & \text{y} \\ \text{Red} & 0.6 & 0.3 \\ \text{Green} & 0.2 & 0.6 \\ \text{Blue} & 0.1 & 0.1 \end{array} \qquad \mathcal{M}_2 = \begin{array}{lcc} & \text{x} & \text{y} \\ \text{Red} & 0.6 & 0.25 \\ \text{Green} & 0.1 & 0.7 \\ \text{Blue} & 0.15 & 0.1 \end{array}$$

ii) Assume that RGB colours are represented by values in the range [0, 1]. Find the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ that will convert (r, g, b) values from the first monitor $\mathcal{M}_1$ into (x, y, z) coordinates.

iii) Derive an expression that converts (r, g, b) values on monitor $\mathcal{M}_1$ into the corresponding (r, g, b) values of monitor $\mathcal{M}_2$. You can assume that the inverses of $\mathbf{M}_1$ and $\mathbf{M}_2$ are as given below:

$$\mathbf{M}_1^{-1} = \begin{pmatrix} 2 & -0.61 & -0.17 \\ -1 & 2.04 & -0.13 \\ 0 & -0.43 & 1.30 \end{pmatrix} \quad \mathbf{M}_2^{-1} = \begin{pmatrix} 1.82 & -0.16 & -0.34 \\ -0.62 & 1.54 & -0.08 \\ -0.20 & -0.38 & 1.42 \end{pmatrix}$$

iv) Assume that a colour is shown in RGB on monitor $\mathcal{M}_1$ as $(0.7, 0.1, 0.1)$. What RGB colour does this correspond to on monitor $\mathcal{M}_2$?

The three parts carry, respectively, 20%, 45%, and 35% of the marks.