CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year:	2013-2014
Examination Period:	Autumn
Examination Paper Number:	CMT107
Examination Paper Title:	Visual Computing
Duration:	2 hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are **3** pages.

There are **4** questions in total.

There are no appendices.

The maximum mark for the examination paper is **75** and the mark obtainable for a question or part of a question is shown in brackets alongside the question.

Students to be provided with:

The following items of stationery are to be provided:

ONE answer book.

Instructions to Students:

Answer **3** questions.

The use of a translation dictionary between English or Welsh and another language, provided that it bears an appropriate school stamp, is permitted in this examination.

1. Colour, Homogeneous Coordinates and Transformation

(a) Answer the following questions about **colour**.

(i) What is colour? [2]

(ii) What are *metamers*? [2]

(iii) Why do we generally use three values to represent colour? Give biological and physical reasons. [4]

(iv) Can we combine three *physical colours* such as RGB to represent all *perceivable colours*? Why? [2]

(b) Answer the following questions about **homogeneous coordinates**.

(i) What are *homogeneous coordinates*? [2]

(ii) Why *homogeneous coordinate* representation is necessary in visual computing? [2]

(iii)The centre of a camera is (0, 10, 0), and it looks at a point (0, 0, -5). Given the homogeneous coordinate representation of the direction of the camera optical axis. [3]

(c) Give the OpenGL instructions to put the following transformations on the modelview matrix stack: Rotate an object about the z axis direction through the point [1, 0, 1] by 45 degrees, and then rotate it about the y axis through the origin by 60 degrees. [8]

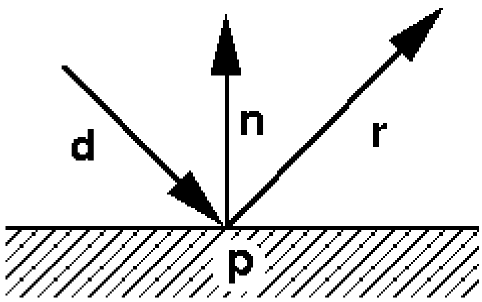
Total: [25]

2. Lighting and Shading

(a) Give the mathematical formula for the *Phong illumination model* and explain each term in detail. [8]

(b) Describe how the *Gouraud* and *Phong shading* techniques compute the colour of the pixels rendered for a polygon by a graphics pipeline. Explain the calculation required in terms of the vertices and vertex normals, however there is no need to describe how the illumination model is evaluated. [8]

(c) The figure below shows a perfect reflection ***r*** of a light ray ***d*** at a point *P* on a surface with normal direction ***n***. Give the equations to compute the *unit direction vector* ***r*** from vectors ***d*** and ***n***, and justify that your result ***r*** is a unit vector (Note: ***d*** and ***n*** are not unit vectors). [9]



Total: [25]

3. Filtering and Feature Detection

(a) What is a separable filter kernel? Why is separability useful? [4]

(b) Which of the following kernels are separable? Separate the kernels that are separable.

$$\underset{(A)}{\begin{bmatrix} 1 & 3 & 2 \\ 2 & 6 & 4 \\ 1 & 3 & 2 \end{bmatrix}} \qquad \underset{(B)}{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}} \qquad \underset{(C)}{\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}}$$

[4]

(c) The above kernel (B) can be used to detect edges. What properties are required for a kernel to be used to detect edges? Can we use kernel (C) to detect edges? Give your reasons. [6]

(d) Compute the convolution for the central pixel of the image patch below with the above kernel (C).

$$\begin{bmatrix} 10 & 15 & 5 \\ 12 & 10 & 3 \\ 8 & 5 & 2 \end{bmatrix}$$

[3]

(e) The Harris corner detection algorithm computes a 2 x 2 matrix at each pixel based on the first derivatives at that point and then computes the two eigenvalues of the matrix, λ_1 and λ_2, where $\lambda_1 \leq \lambda_2$. How can these two values be used to label each pixel as either a locally smooth region (S), an edge point (E), or a corner point (C)? [8]

Total: [25]

4. Camera Projection and Calibration

(a) Answer the following questions about vanishing points.

(i) What is a vanishing point? [3]

(ii) What is the maximum number of vanishing points that are defined for any scene? Why? [4]

(iii) How can you construct the vanishing point of a line? Draw a picture and explain. [6]

(b) An ideal pinhole camera has focal length 5mm. Each pixel is 0.02 mm x 0.02 mm and the image principal point is at pixel (600, 400). Pixel coordinates start at (0, 0) in the upper-left corner of the image.

(i) Give the 3x3 camera calibration matrix, ***K***, for this camera configuration [6]

(ii) Assuming the world coordinate frame and the camera coordinate frame have the same origins and the same axis directions, and the origins are at the camera's pinhole, what is the 3 x 4 matrix that represents the extrinsic, rigid-body transformation between the camera coordinate system and the world coordinate system? [2]

(iii) Combining your results from (a) and (b), compute the projection of scene point (10, 16, 500) into image coordinates. [4]

Total: [25]

1. (a)

(i) Colour is the result of interaction between physical light in the environment and our visual system. Colour is a psychological property of our visual experiences when we look at objects and lights, not a physical property of those objects or light. [2]

(ii) Metamers are colours with different spectral power distributions that are perceived as the identical colours. [2]

(iii) Biologically, the human visual system has only three types of cones that can perceive colours with different wavelengths. [2]
Physically, the spectral power distribution of colour is infinite dimensional, but combination of three colours can represent most of the perceivable colours.

(iv) No, we cannot. Because the real colour space is infinite dimensional, but combination of three colours can form a three-dimensional space. [2]

(b)

(i) Homogeneous coordinates are a system of coordinates, which are extended from Cartesian coordinate system by adding one more dimension into the original Cartesian coordinate representation. They can be used to represent points or directions. [2]

(ii) Homogeneous coordinate representation enables us to represent any affine transformation, including translation and perspective projection by matrix multiplication.

(iii) The optical axis direction is defined by a vector v = (0,0,-5) - (0,10,0) = (0, -10, -5), which can be represent by (0, -10, -5, 0) in the homogeneous coordinate system. [3]

(c)

- The OpenGL instruction are as follows:

```
1: glMatrixMode(GL MODELVIEW);
2: glLoadIdentity();
3: glRotatef(60.0, 0.0, 1.0, 0.0);
4: glTranslatef(1.0,0.0,1.0);
5: glRotatef(45.0, 0.0, 0.0, 1.0);
6: glTranslatef(-1.0,0.0,-1.0);
7: Draw an object
```

- Each line except line 7 scores one point. [6]
- Line 3 to 6 scores extra two points, if the order of the lines is correct. [2]

Total: [25]

2. (a)

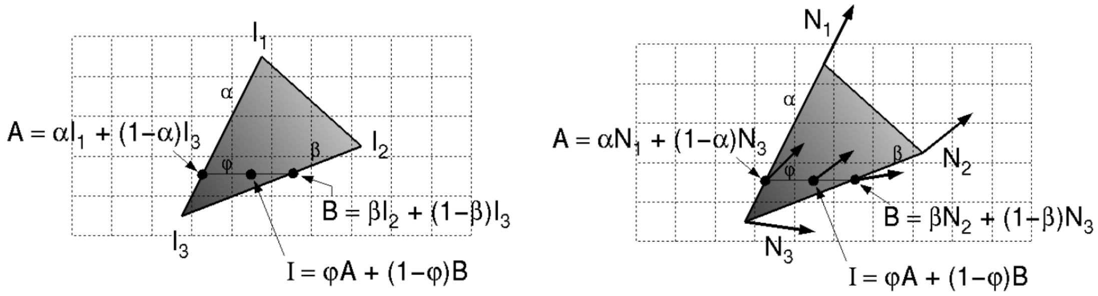
- For a given monochromatic light, Phong illumination model calculates the reflected light by

$$\boldsymbol{R}_a \boldsymbol{I}_a + \boldsymbol{R}_d (\boldsymbol{n}^T \boldsymbol{d}) \boldsymbol{I}_d + \boldsymbol{R}_s (\boldsymbol{r}^T \boldsymbol{v})^{\sigma} \boldsymbol{I}_s \qquad [3]$$

- $\boldsymbol{R}_a$, $\boldsymbol{R}_d$, and $\boldsymbol{R}_s$ are surface reflectance coefficients for ambient, diffuse, and specular light, respectively. [1]
- $\boldsymbol{I}_a$, $\boldsymbol{I}_d$, and $\boldsymbol{I}_s$ are the intensities of the incoming ambient, diffuse, and specular light, respectively. [1]
- $\boldsymbol{n}$ is the unit surface normal; $\boldsymbol{d}$ is the unit direction from surface point to light source; $\boldsymbol{r}$ is unit direction of perfect reflection, $\boldsymbol{v}$ is unit direction towards viewer position, and σ is the shininess exponent. [2]
- The summation over all light sources for red, green blue gives total intensity for all colours. [1]

(b)

- Gouraud shading: Evaluate illumination model for each vertex and bilinearly interpolate the results to get RGB intensities for each pixel. [2]
- Phong shading: Bilinearly interpolate the vertex normals for each pixel and evaluate the illumination model for each pixel with interpolated normals to get intensities (for red, green, blue). [2]
- Bilinear interpolation used in both techniques where interpolation parameters are available from the scan conversion process:



(diagrams not required, formulae are sufficient with indication of what the symbols are) [4]

(c)

- Normalise: $\boldsymbol{N} = \boldsymbol{n}/\|\boldsymbol{n}\|$ and $\boldsymbol{D} = \boldsymbol{d}/\|\boldsymbol{d}\|$ [2]
- The projection of $\boldsymbol{D}$ on $\boldsymbol{N}$ direction is $\boldsymbol{P} = \boldsymbol{N}(\boldsymbol{D} \cdot \boldsymbol{N})$. [2]
- Then $\boldsymbol{D} - \boldsymbol{r} = 2\boldsymbol{P}$, or $\boldsymbol{r} = \boldsymbol{D} - 2\boldsymbol{P}$. [2]
- To justify that $\boldsymbol{r}$ is a unit vector, we compute the length of $\boldsymbol{r}$.

$$\begin{aligned} \|\boldsymbol{r}\|^2 &= (\boldsymbol{D} - 2\boldsymbol{P}) \cdot (\boldsymbol{D} - 2\boldsymbol{P}) = \|\boldsymbol{D}\|^2 - 4\boldsymbol{P} \cdot \boldsymbol{D} + 4\|\boldsymbol{P}\|^2 \\ &= 1 - 4(\boldsymbol{D} \cdot \boldsymbol{N})^2 + 4(\boldsymbol{D} \cdot \boldsymbol{N})\boldsymbol{N} \cdot \boldsymbol{N}(\boldsymbol{D} \cdot \boldsymbol{N}) = 1 \end{aligned} \qquad [3]$$

Total: [25]

3. (a)

- A separable kernel is a 2D kernel that can be decomposed into the product of two 1D kernels. [2]
- For an nxn kernel, it can improve efficiency from $O(n^2)$ to $O(n)$. [2]

(b)

- (A) and (B) [2]
- $(A)=\begin{bmatrix}1\\2\\3\end{bmatrix}\begin{bmatrix}1 & 3 & 2\end{bmatrix}$, $(B)=\begin{bmatrix}1\\2\\1\end{bmatrix}\begin{bmatrix}1 & 0 & -1\end{bmatrix}$ [2]

(c)

- Kernels for edge detection are approximation of partial derivative operations. [2]
- The sum of the kernel elements is 0. [2]
- (C) can be used to detect edges, because it satisfy the above properties. [2]

(d) convolution = 10*0+15*1+5*2+12*(-1)+10*0+3*1+8*(-2)+5*(-1)+2*0 = -5 [3]

(e)

- The Harris corner detector first computes $R=\lambda_1\lambda_2-k(\lambda_1+\lambda_2)^2$, [2]
  and then
- labels a pixel S if $|R|\approx 0$ (or, alternatively, $\lambda_1\approx\lambda_2\approx 0$); [2]
- labels a pixel E if $R<T_1<0$ (or, alternatively, $\lambda_1\approx 0$
  (corresponding to the direction of the edge) and λ_2 is large
  (corresponding to the normal direction at the edge)), where T_1 is a threshold, or [2]
- labels a pixel C if $R>T_2>0$ (or, alternatively, λ_1 and λ_2 are both large), where T_2 is a threshold. [2]

Total: [25]

4. (a)

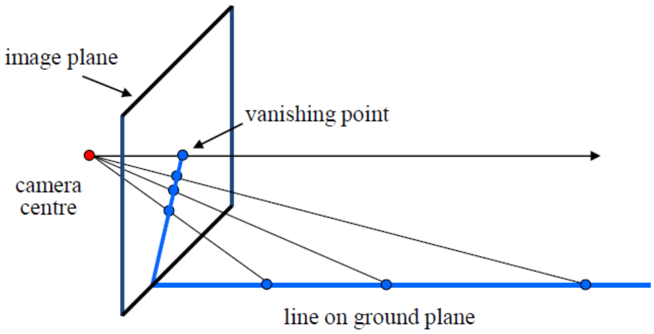
(i)

- A vanishing point is a point in the image plane determined by a line in space. [1]
- All parallel lines going in the same direction will converge to a vanishing point.[1]
- Directions parallel to image plane don't have vanishing point. [1]

(ii) Infinite, [2]

because the number of the vanishing points is determined by the number of directions of parallel lines in a scene, which has no bound. [2]

(iii) See the picture below.



Suppose we have a line on the ground, and we want to find its vanishing point on the image plane. Select at least two points on the line, and find the intersection points between the image plane and the lines linking the camera centre $\boldsymbol{c}$ and these points, separately. All this intersection points form a line $\boldsymbol{L}$ on the image plane. The vanishing point is a point $\boldsymbol{p}$ on the line $\boldsymbol{L}$, such that the line linking $\boldsymbol{c}$ and $\boldsymbol{p}$ is parallel to the line on the ground. [6]

(b)

(i)

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{fk_u} & 0 & \boldsymbol{u_0} \\ 0 & \boldsymbol{fk_v} & \boldsymbol{v_0} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 \cdot \dfrac{1}{0.02} & 0 & 600 \\ 0 & 5 \cdot \dfrac{1}{0.02} & 400 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 250 & 0 & 600 \\ 0 & 250 & 400 \\ 0 & 0 & 1 \end{bmatrix} \qquad [6]$$

(ii)

$$\boldsymbol{P_r} = \begin{bmatrix} \boldsymbol{r}_{11} & \boldsymbol{r}_{12} & \boldsymbol{r}_{13} & \boldsymbol{T_x} \\ \boldsymbol{r}_{21} & \boldsymbol{r}_{22} & \boldsymbol{r}_{23} & \boldsymbol{T_y} \\ \boldsymbol{r}_{31} & \boldsymbol{r}_{32} & \boldsymbol{r}_{33} & \boldsymbol{T_z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad [2]$$

(iii)

$$\begin{bmatrix} \boldsymbol{su} \\ \boldsymbol{sv} \\ \boldsymbol{x} \end{bmatrix} = \boldsymbol{KP_r} \begin{bmatrix} 10 \\ 16 \\ 500 \\ 1 \end{bmatrix} = \begin{bmatrix} 250 & 0 & 600 & 0 \\ 0 & 250 & 400 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 10 \\ 16 \\ 500 \\ 1 \end{bmatrix} = \begin{bmatrix} 302500 \\ 204000 \\ 500 \end{bmatrix}$$

so, the scene point projects to pixel (302500/500, 204000/500) = (605, 408) [4]

Total: [25]