4F9

Dr A Gee ahg@eng.cam.ac.uk

Engineering Tripos Part IIB FOURTH YEAR

Module 4F9: Medical Imaging & 3D Computer Graphics

Solutions to 2006 Tripos Paper

1. **Medical imaging modalities, B-scan artefacts**

(a) (i) The brightness of each point on a planar X-ray represents the cumulative attenuation (X-ray transform) along a line through the subject. The X-ray linear attenuation coefficient depends on the atomic number and density of the material as well as the energy of the photons in the X-ray beam.

(ii) Each point in a CT slice depends on the attenuation of the X-rays at a single point in 3D space. This is called the CT number and is measured in *Hounsfield units.*

(iii) Each point in a PET slice depends on the rate at which a particular radio-pharmaceutical is decaying close to a particular point in 3D space. If the radio-pharmaceutical has been used to tag a particular bio-chemical activity then the PET scan will give an indication of the locations where that activity is taking place. Attenuation and scattering of the radiation in the rest of the body will affect the quality of the scan.

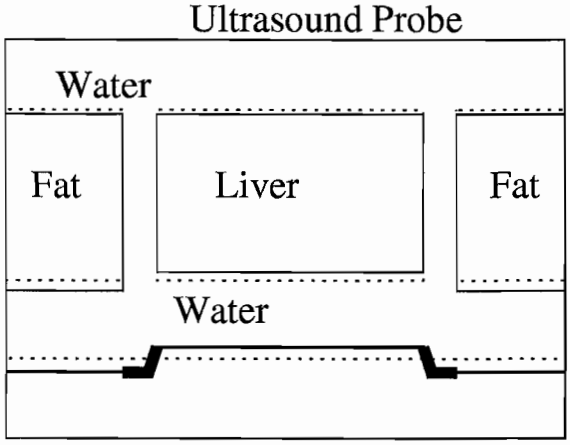
(iv) A B-scan displays the extent to which a high frequency sound beam is back-scattered in the vicinity of each point in space. The amount of back-scattering is caused by variations in the density, elasticity and sound speed of the material. The image is corrupted by uneven attenuation of the beam between the probe and each point where the back-scattering takes place. [40%]

(b) There are two main reasons why one might combine PET and CT scans. PET scans are distorted by variations in the total attenuation along different lines through the body. It is possible to correct for this by passing a gamma source around the subject and recording a transmission scan just before doing the PET (emission) scan. This transmission scan is effectively a CT scan. In practice, this is not often done as clinicians can usually correct for the distortion as part of the process of interpreting the PET scan.

An X-ray CT scan provides a relatively high resolution representation of anatomical structure. A PET scan can be used to represent an aspect of the *function* of the body — eg. to map a particular biochemical activity. However, PET is lower resolution than CT. By registering CT and PET scans together, the location of the function from the PET scan can be associated with the higher resolution anatomical information provided by CT. [10%]

(c) The ultrasound machine assumes that the speed of sound in the body is 1540 m/s. Hence it travels slightly too slowly through water, even slower through fat, and rather too fast through liver. The dimensions in the horizontal direction will be

approximately correct as they depend on the spacing of the transducer elements. In the vertical direction, the fat and water sections will be too long with this effect worse in the fat than in the water. The liver section will be too short, such that the liver will appear to be about 10% smaller than the fat, even though they are in reality the same size. Note that the same distortion will apply to the shape of the bottom of the water bath, which will appear to be closer to the probe underneath the liver. The bottom of the water bath will appear to be brighter below the gaps in the blocks of tissue because the beam will not have been attenuated by tissue above those sections.

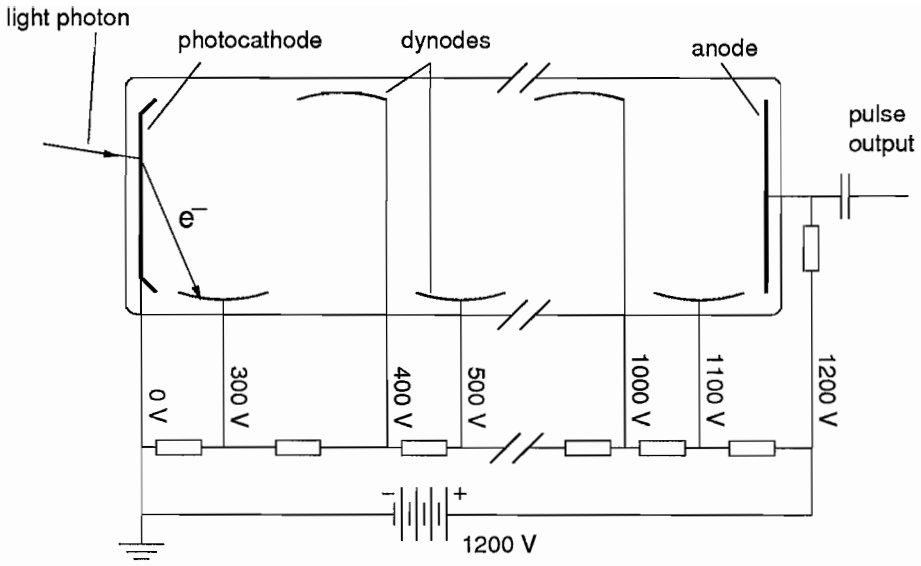


[50%]

Assessors' remarks: This question tested the candidates understanding of the physical principles involved in several medical imaging modalities. It finished with an approximate calculation of the changes in the sizes of objects in ultrasound scans as a results of the different speeds of sound in different materials. There were a large number of good answers and quite a few that showed a high level of understanding. Some candidates lost marks through not being able to differentiate between planar X-rays and CT slices. Several people produced excellent discussions of the distortions inherent in the ultrasonic imaging process.

2. **Nuclear medicine detectors**

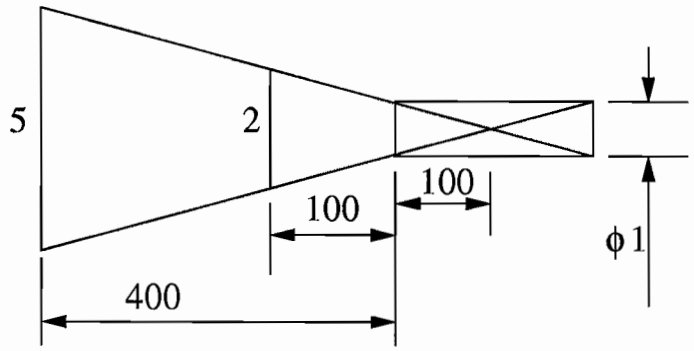
(a) Light is incident on a photocathode at the left hand end of the photomultiplier tube. This results in the emission of a photo-electron into the vacuum inside the tube. This accelerates towards the closest dynode which is positively charged relative to the cathode. When the electron hits the dynode, several electrons are released and they accelerate toward the next dynode to the right, which is more positive than the first dynode. As the electrons hit each dynode, they release more electrons and the current passing down the tube is amplified. This is eventually picked up by the anode at the right hand end.



[35%]

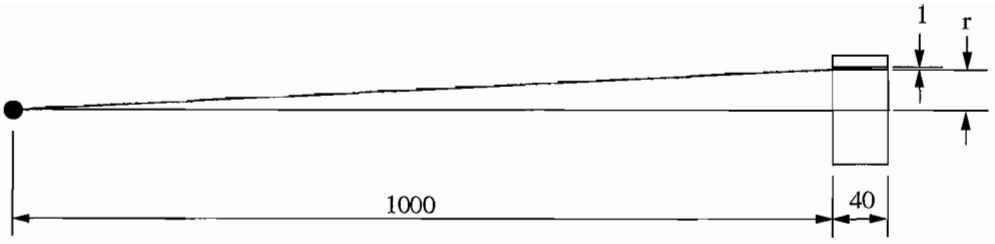
(b) In a scintillation crystal photomultiplier coupled detector, the magnitude of the output is proportional to the energy of the incoming gamma photon. Such detectors are fast and have high efficiency, but can only be packed at a relatively low density compared with other detectors (eg. the photodiode coupled detector). [10%]

(c) The geometry is as shown in the diagram below. The diagram is not to scale; the vertical axis has been expanded for clarity. By similar triangles, the collimator will have a diameter of 1 mm, and a length of 20 cm.



[25%]

(d) Using the figure below we can see that the radius r at which the radiation just gets to the back of the hole is 25 mm. This means that the radiation gets through the centre hole, the 8 holes in the first ring and the 16 holes in the second ring. This is a total of 25 holes with a combined surface area of $0.5^2 \times \pi \times 25 = 19.635\,\text{mm}^2$.



The total area of the disk is $50^2 \times \pi = 7853.98\,\text{mm}^2$.

The distance of the source from the collimator is ten times the diameter of the collimator so we can assume that the radiation is relatively uniform across the face of the collimator. Hence the proportion of the incident photons reaching the back will be $\frac{19.635}{7853.98} = 0.0025 = 0.25\%$. [30%]

Assessors' remarks: This question was about the gamma camera, particularly focusing on the operation of the collimator. It seemed to have been chosen by some of the weaker students as several appeared to be unable to do the geometry required in the collimator resolution calculation. The reproduction of the description of the photomultiplier tube from the notes was also of a surprisingly low standard.

3. **Marching cubes and volume estimation**

(a) Marching Cubes is a technique for extracting a triangulated isosurface from a regular voxel array of data. It breaks the problem down into that of triangulating a cube with one data point at each corner. The *form* of the triangles required to construct the surface within this cube is a function only of whether each data point at the corners of the cube is inside or outside the surface to be defined. This determines the *triangulation case*. Having found the case in which this cube falls, a lookup table is used to give the number and location of the triangles to form the isosurface within this cube.

This lookup table only defines which cube edge the vertices of each triangle lie on. The exact location of each triangle vertex is determined by linear interpolation of the original data values along the required cube edge. For example, if the isosurface is at the value 10, and the data values at the corners are 7 and 13, then this vertex will lie exactly half way along the edge.

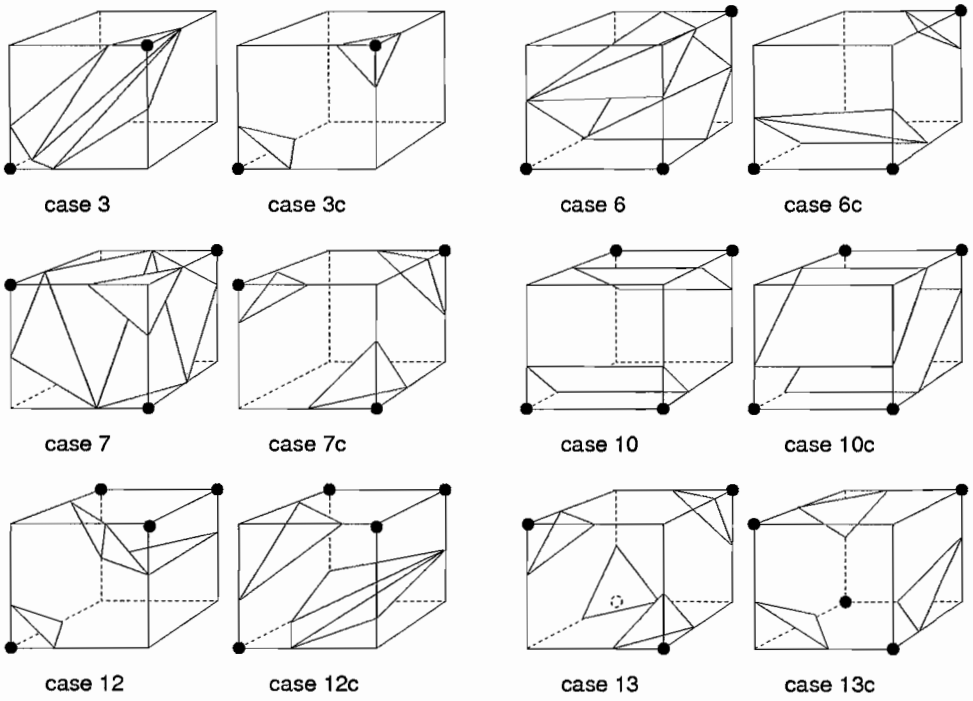
Having triangulated one such cube, we then gradually move (or *march*) the cube through the entire voxel array. The complete isosurface is then given by the set of all the triangles from each of the cubes.

Marching Cubes will generate an isosurface with several triangles per cube through which the surface passes, however, the surface is a good approximation to the real location of the data. [30%]

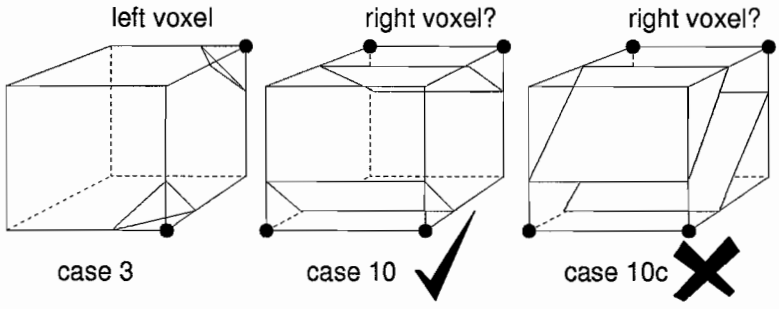
(b) Since there are 8 data values (one per cube corner) and each of these can be either inside or outside the surface, then there are 2^8, or 256 possible cases. These can be

reduced to only 15 by symmetry, and only 14 of these actually require triangulation, since one of these cases represents all the data values being either inside or outside the surface. [10%]

(c) Six of the 15 cases have alternative triangulations. These are given below: any one will do for this question.

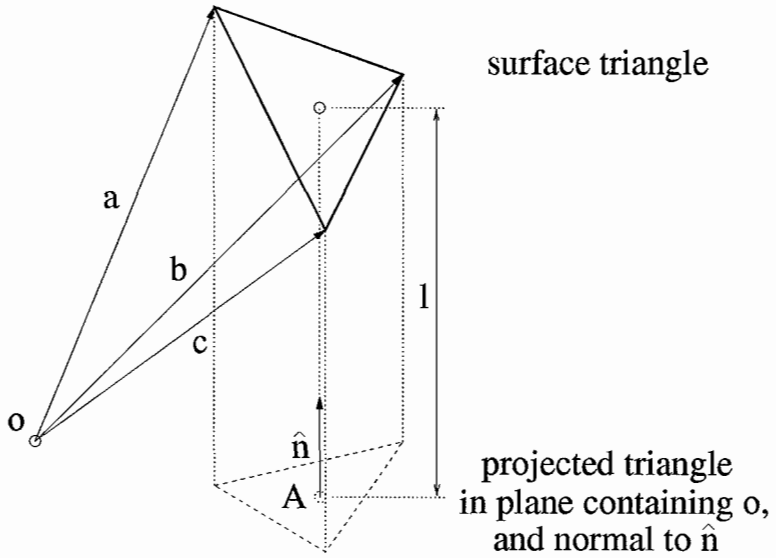


The diagram below shows what happens if the wrong case is used. When all the triangulated cubes are connected, a gap remains in the surface.



[20%]

(d) The volume of a closed, triangulated surface can be calculated by summing the projected volumes of each triangle, shown below, across the whole surface.



The area of a triangle, with vertices defined by the position vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, is given by:

$$\text{area} = \frac{1}{2}(\mathbf{b}-\mathbf{a}) \times (\mathbf{c}-\mathbf{a}) \tag{1}$$

Hence the projected area A in the direction of projection $\hat{\mathbf{n}}$ is:

$$A = \frac{1}{2}((\mathbf{b}-\mathbf{a}) \times (\mathbf{c}-\mathbf{a})) \cdot \hat{\mathbf{n}} \tag{2}$$

The distance from the centre of the triangle to its projection, l, is:

$$l = \frac{1}{3}(\mathbf{a}+\mathbf{b}+\mathbf{c}) \cdot \hat{\mathbf{n}} \tag{3}$$

Hence the enclosed projected volume is:

$$V = Al = \frac{1}{6}[((\mathbf{b}-\mathbf{a}) \times (\mathbf{c}-\mathbf{a})) \cdot \hat{\mathbf{n}}]\,[(\mathbf{a}+\mathbf{b}+\mathbf{c}) \cdot \hat{\mathbf{n}}] \tag{4}$$

The summation of this volume across all the N triangles in the surface gives the total enclosed volume of the surface. The normal direction $\hat{\mathbf{n}}$ can be chosen arbitrarily, so long as it is the same for all triangles.

In order for this volume to be meaningful, the triangulated surface must be closed (no gaps in the surface), must not self-intersect, must be manifold, and the ordering of the vertices in each triangle must be consistent.. [40%]

Assessors' remarks: In general a well answered question. In part (a), few candidates mentioned that the location of the vertices in the triangulation is derived from the original data values. There were some very good answers to part (d), deriving the correct equation, with a good textual explanation as well.

4. **Terminating Catmull-Rom splines and B-splines**

(a) Both Catmull-Rom splines and B-splines use exactly the same geometry matrix, which is simply a vector of four points. The curve segment is generally between the second and third point in this vector. Both types of spline are inherently multi-segment — segments can be joined by sharing three of the control points in each geometry matrix. Hence $N+3$ points are needed to define N curve segments.

However, the Catmull-Rom spline *interpolates* the points in its geometry matrix, whereas the B-spline does not. Each segment in a Catmull-Rom spline will start at the second control point and end at the third. Segments in a B-spline will also tend to occupy the space near the second and third control points, but do not (in general) interpolate them.

The B-spline, on the other hand, possesses better continuity, being continuous in the second parametric derivative (i.e. C^2), whereas the Catmull-Rom spline only possess C^1 continuity. The B-spline is also guaranteed to be entirely contained within the convex hull of its control points, which is not the case for the Catmull-Rom spline. [20%]

(b) We want curve values at the $t=1$ end of the curve, in which case the parameter matrix is $[\,1\ \ 1\ \ 1\ \ 1\,]$, $[\,3\ \ 2\ \ 1\ \ 0\,]$ and $[\,6\ \ 2\ \ 0\ \ 0\,]$ for the value, first and second derivatives respectively. If we multiply this into the basis matrix for the Catmull-Rom spline, we get $[\,0\ \ 0\ \ 1\ \ 0\,]$, $[\,0\ \ \frac{-1}{2}\ \ 0\ \ \frac{1}{2}\,]$ and $[\,-1\ \ 4\ \ -5\ \ 2\,]$ respectively. Hence the location $\mathbf{p}$, first derivative $\mathbf{p}'$ and second derivative $\mathbf{p}''$ for the Catmull-Rom spline are:

$$\mathbf{p} = \mathbf{c}, \quad \mathbf{p}' = \frac{1}{2}(\mathbf{d}-\mathbf{b}), \quad \mathbf{p}'' = (-\mathbf{a}+4\mathbf{b}-5\mathbf{c}+2\mathbf{d})$$

Similarly for the B-spline:

$$\mathbf{p} = \frac{1}{6}(\mathbf{b}+4\mathbf{c}+\mathbf{d}), \quad \mathbf{p}' = \frac{1}{2}(\mathbf{d}-\mathbf{b}), \quad \mathbf{p}'' = (\mathbf{b}-2\mathbf{c}+\mathbf{d}) \qquad [20\%]$$

(c)(i) For $\mathbf{d}$ located as in Fig. 1 (a), for the Catmull-Rom spline, we have:

$$\mathbf{p} = \mathbf{c}, \quad \mathbf{p}' = 0, \quad \mathbf{p}'' = (-\mathbf{a}+6\mathbf{b}-5\mathbf{c})$$

which means that the direction of the gradient at the end-point is (unusually) not constrained. For the B-spline:

$$\mathbf{p} = \frac{1}{6}(2\mathbf{b}+4\mathbf{c}), \quad \mathbf{p}' = 0, \quad \mathbf{p}'' = (2\mathbf{b}-2\mathbf{c})$$

which means that the end point is two-thirds of the way between $\mathbf{b}$ and $\mathbf{c}$, has zero velocity, and the final acceleration is in the direction from $\mathbf{c}$ to $\mathbf{b}$.

For $\mathbf{d}$ located as in Fig. 1 (b), $\mathbf{d} = 2\mathbf{c}-\mathbf{b}$, hence for the Catmull-Rom spline, we have:

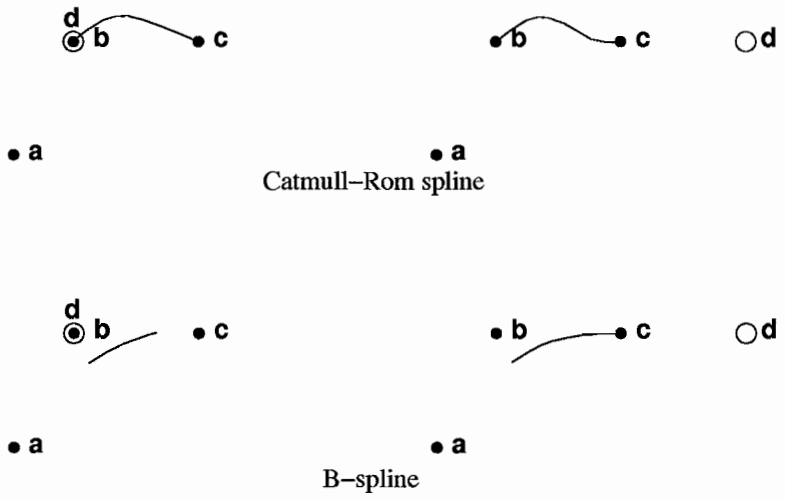
$$\mathbf{p} = \mathbf{c}, \quad \mathbf{p}' = (\mathbf{c}-\mathbf{b}), \quad \mathbf{p}'' = (-\mathbf{a}+2\mathbf{b}-\mathbf{c})$$

which means that the direction of the gradient at the end-point is along the line from **b** to **c**. For the B-spline:

$$\mathbf{p} = \mathbf{c}, \quad \mathbf{p}' = (\mathbf{c} - \mathbf{b}), \quad \mathbf{p}'' = 0$$

which means that the curve ends at **c**, with zero acceleration and hence a constant velocity along the line from **b** to **c**.

These splines are sketched below.



[40%]

(ii) The first option is probably the best for the Catmull-Rom spline. The user has not defined any points beyond **c**, so we have no idea what the gradient should be at **c**. The first option effectively allows the curve to take the simplest path between **b** and **c** without specifically constraining the gradient.

For the B-spline, either option could be useful. If the curve is being used to define a motion path, the first has the desirable property that there is zero velocity at the end point. On the other hand, the second guarantees that the path will stop at exactly the last user-defined point. Which of these is best will be implementation-dependent. [20%]

Assessors' remarks: There was a wide spread of marks for this question. Several candidates fell down on the simple matrix multiplications required in part (b), despite knowing what they were supposed to be doing. Equally, several candidates lost marks in (c)(ii) for poor sketching, or not attempting the sketches at all. It was a very common mistake to draw the splines as starting from point **a** rather than point **b**. On the other hand, there were also some nearly perfect answers.

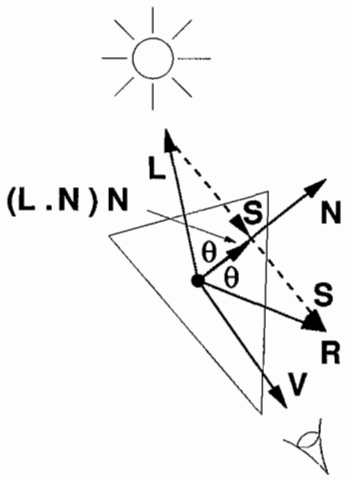
5. **Illumination and reflection**

(a) I_λ is the intensity of the reflected light of colour λ, where $\lambda \in \{r, g, b\}$ for red, green and blue.

I_λ depends on several terms. First, there is the ambient reflection term, $c_\lambda I_a k_a$, which models indirect illumination of the surface. c_λ, where $0 \leq c_\lambda \leq 1$, specifies the colour of the surface. I_a is the intensity of the general background illumination, and k_a is the surface's ambient reflection coefficient.

The next two terms in the model are calculated for a point light with intensity I_p. First there is the diffuse reflection term, $c_\lambda k_d \mathbf{L}.\mathbf{N}$, which models even reflection of the light source in all directions. Diffuse reflection is greatest when the surface in pointing directly towards the light source, and tails away to zero when the surface is side-on to the light source. $\mathbf{L}$ is the unit vector from the surface point towards the light source, $\mathbf{N}$ is the unit surface normal and k_d is the surface's diffuse reflection coefficient (small for dark surfaces, high for bright surfaces).

Finally, there is the specular reflection term, $k_s\,(\mathbf{R}.\mathbf{V})^n$, which models directional reflection of the light source along the unit mirror vector $\mathbf{R}$. $\mathbf{V}$ is the unit vector from the surface point towards the viewer. The viewer only perceives the specular highlight (or glint) when looking along the mirror direction, or at least close to it. k_s is the surface's specular reflection coefficient (small for matte surfaces, high for shiny surfaces), and n is the specular exponent that determines the tightness of the glint. n is high for a tight highlight (eg. a perfect mirror) and small for a more blurred highlight (eg. aluminium).



The component of $\mathbf{L}$ along $\mathbf{N}$ is $(\mathbf{L}.\mathbf{N})\mathbf{N}$ (we assume that all vectors are normalised). By inspection of the diagram, $\mathbf{R}$ is given by $(\mathbf{L}.\mathbf{N})\mathbf{N} + \mathbf{S}$, where $\mathbf{S} = (\mathbf{L}.\mathbf{N})\mathbf{N} - \mathbf{L}$. Hence

$$\mathbf{R} = (\mathbf{L}.\mathbf{N})\mathbf{N} + (\mathbf{L}.\mathbf{N})\mathbf{N} - \mathbf{L} = 2(\mathbf{L}.\mathbf{N})\mathbf{N} - \mathbf{L} \qquad [40\%]$$

(b) Remembering that we need *unit* vectors, we have

$$\mathbf{L} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 0 \\ 1/2 \\ \sqrt{3}/2 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Working out the various terms required by the Phong model, we get

$$\mathbf{L}.\mathbf{N} = \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right) \ , \ \ \mathbf{R}.\mathbf{V} = y \text{ component of } 2(\mathbf{L}.\mathbf{N})\mathbf{N} - \mathbf{L} = \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right)$$

and hence

$$I_p k_s (\mathbf{R}.\mathbf{V})^n = \left(\frac{1}{2} + \frac{1}{2\sqrt{3}}\right)^{10} = 0.0931 \ , \ \ I_a k_a + I_p k_d \mathbf{L}.\mathbf{N} = 0.5 + \frac{1}{4}\left(1 - \frac{1}{\sqrt{3}}\right) = 0.606$$

So $I_r = I_g = 0.0931$, $I_b = 0.0931 + 0.606 = 0.699$. [20%]

(c) Calculating **R** involves considerable computational expense. The advantage of the **N**.**H** alternative (which produces very similar specular highlights) becomes apparent when the light source and viewer are both at infinity. Under these conditions, **L**, **V** and **H** are constant across the scene. So calculating the specular component requires only a single dot product **N**.**H** for each vertex. [15%]

(d) The Phong model considers only three particular colours (the primaries red, green and blue) and the way a surface reflects these three colours. This amounts to severe undersampling of the underlying, continuous spectra. The likely consequence is *colour aliasing*, whereby the rendered colour is not correct given the true illumination and reflectivity functions. Compared with the other gross approximations in the surface rendering pipeline, colour aliasing is subtle and rarely of any consequence, except in those applications where accurate colour simulation is important (architectural modelling, perhaps). The model's treatment of colour could be improved by sampling the illumination and reflectivity spectra at n wavelengths and evaluating the model at each wavelength separately. The n outputs could be reduced to RGB components in a post-processing stage (RGB components are needed to drive a monitor). Research has shown that $n = 9$ is sufficient for most applications. [25%]

Assessors' remarks: This popular question tested the candidates' understanding of the Phong model. Part (a) was book work and answered well by almost all candidates. Part (b) required numerical manipulation of the model: very few candidates were able to get the correct answer, the most common mistake being forgetting to normalise the vectors. Most candidates knew about the advantages of the half-way vector in (c), though many failed to mention that it is only useful when the light source and viewer are at infinity. There were very few good answers to (d), where the candidates were asked to comment on the model's three-component treatment of colour. While several had the right idea, only one candidate was able to discuss the issues in the context of sampling theory.

6. **Clipping, triangle decomposition**

(a) Without a clipping stage, the pipeline would render every polygon in the scene, including those off the edge of the screen and even those behind the viewpoint. So it is common practice to specify a *view volume*, usually a truncated rectangular pyramid

with its apex at the centre of projection, and clip all polygons to this view volume. Rendering only those polygons inside the view volume has several advantages:

- We can selectively cut out portions of the scene we are not interested in. We can even look behind objects by pushing the front clipping plane further back.
- The far clipping plane can be used to cut out distant objects which would be very small in the image: such objects often produce aliasing artifacts.
- Cutting out objects we are not interested in can drastically reduce the number of polygons to be rendered, increasing the speed of rendering.

If we write the homogeneous 3D screen coordinates as $[X\ Y\ Z\ w]^T$, then the view volume is defined by

$$-w \leq X \leq w\ ,\ \ -w \leq Y \leq w\ ,\ \ 0 \leq Z \leq w$$

[20%]

(b) (i) All of B's vertices lie inside the view volume. The first two of A's vertices are inside, the third is outside ($Z > w$). The inside/outside test is usually performed in homogeneous 3D screen coordinates, since the inequalities are independent of the particular view volume, making hardware implementation relatively straightforward. Also, by working in homogeneous coordinates as much as possible, the computationally expensive divide by w can be avoided for polygons found to be outside the view volume. [20%]

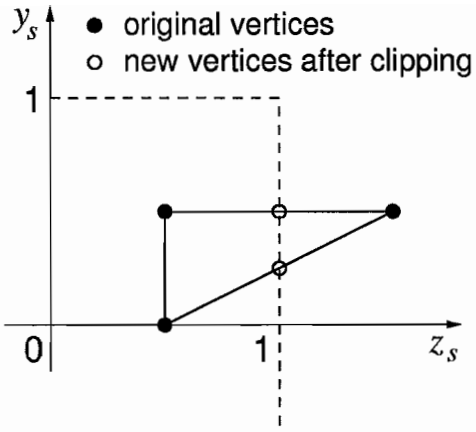
(ii) In non-homogeneous 3D screen coordinates, A and B are:

$$\text{A}: \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 1.5 \end{bmatrix} \qquad \text{B}: \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix}$$

So A and B have the same 3D screen coordinates, except B's z_s coordinates are 0.5 less than A's. Given that A projects to the same 3D screen area as B, but is more distant, it follows that A must be larger than B. How much larger depends on the degree of perspective compression (the ratio of the near and far clip plane distances).

The last two vertices of each polygon have the same (x_s, y_s) values and are therefore coincident on the screen. So the polygons are being viewed end on and will not be rasterised: they will not occupy any screen area at all. [20%]

(iii) Two of A's edges cross the far clipping plane.



A's new vertex list is therefore

$$\begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0.25 \\ 1 \end{bmatrix}$$ [20%]

(c) The benefits of triangle decomposition are:

Rasterisation: triangles have just one span per scanline. Non-convex polygons may have multiple spans which will require sorting in the rasterisation stage.

Bilinear interpolation: the results of interpolating information from the vertices of a triangle are invariant to image rotation. This is not the case for polygons with more than three vertices.

Triangle decomposition usually takes place immediately after the clipping stage. Although clipping non-convex polygons can be problematic compared with triangles (non-convex polygons may clip to multiple, disjoint polygons), it's fairly pointless making triangles earlier: as we saw above, a clipped triangle may have more than three vertices, so we'd need to decompose into triangles again. [20%]

Assessors' remarks: The few candidates who attempted this question demonstrated, on the whole, an excellent grasp of viewing geometry and clipping. The average mark was brought down by one token attempt and some algebraic slips.

Andrew Gee, Richard Prager & Graham Treece
May 2006