Engineering Tripos Part IIB FOURTH YEAR

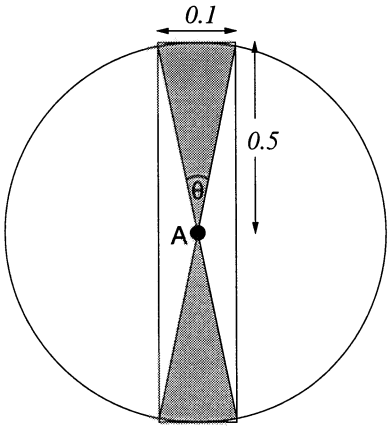
Module 4F9: Medical Imaging & 3D Computer Graphics

Solutions to 2004 Tripos Paper

1. **Nuclear medicine imaging**

(a) SPECT stands for single photon emission computed tomography. PET stands for positron emission tomography. SPECT has the advantage of relatively cheap scanners and ready availability of suitable radiopharmaceuticals without the need for an on-site cyclotron. PET scanners are orders of magnitude more sensitive than SPECT scanners, allowing reduced patient doses, shorter acquisition times and better signal to noise ratios. However, the scanners are expensive and the radionuclides short-lived, requiring an on-site cyclotron production facility. [20%]

(b) (i) The photons are emitted uniformly over the sphere, but only those emitted within the angle θ hit the detector ring.



side view

If the surface area of the section of the sphere subtended by the angle θ is S, then the proportion of photons hitting the detector ring is $S/4\pi r^2$, where $r = 0.5$ m. Approximating S by the surface area of the cylindrical detector ring, we get $S = 0.1 \times 2\pi r$, so the expected number of pairs hitting the ring is $10^7 \times 0.1 \times 2\pi r/4\pi r^2 = 10^6$ s^{-1}. S can be found exactly by integration, giving a precise answer of 9.95×10^5 s^{-1} (not required). [10%]

(ii) Assuming each photon travels 20 cm through the attenuating material (because of the ring's finite thickness, most will travel slightly further than this), the probability of any given photon reaching the detector ring is $e^{-\mu x} = e^{-0.1\times 20} = 0.135$. Because the fate of each photon is independent, the probability of both photons making it to the ring is $0.135^2 = 0.0183$. The expected number of pairs reaching the detector ring is therefore $0.0183 \times 10^6 = 1.83 \times 10^4$ s^{-1}. [20%]

(c) (i) We use Poisson statistics to calculate the probability of photons being emitted (and therefore detected, since there's no attenuation) within a particular time period. For an expected ring impact rate of $r = 10^6$ s^{-1}, the probability of no photons arriving within $\Delta t = 10$ ns is

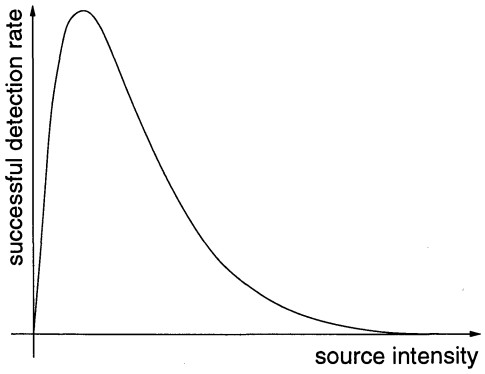
$$p_r(0) = \frac{e^{-r\Delta t}(r\Delta t)^0}{0!} = e^{-10^6\times 10\times 10^{-9}} = 0.9900$$

The expected count rate at detectors B and C is $10/180 \times 10^6 = 5.56 \times 10^4$ s^{-1}. The probability of no further pairs arriving within 300 ns is

$$p_r(0) = \frac{e^{-r\Delta t}(r\Delta t)^0}{0!} = e^{-5.56\times10^4\times300\times10^{-9}} = 0.9835$$

Assuming both types of rejection occur independently (ie. neglecting the small probability that further photons arrive at B and C within 10 ns), the probability that the measurement is accepted is $0.9900 \times 0.9835 = 0.9737$. The expected successful detection rate is therefore $0.9737 \times 10^6 = 9.74 \times 10^5$ s^{-1}. [20%]

(ii) The number of pairs successfully detected is of the form xe^{-kx}.



Thus, with increased radioactivity the detection rate will eventually decline exponentially, since there will be too many collisions at the detectors. [10%]

(d) In 2D mode, the detector rings are separated by retractable septa, so that only activity from inside the ring is detected. This allows the use of relatively straightforward 2D reconstruction algorithms, and there are seldom any problems with detection rate and collisions. However, the majority of the emitted photons — the difference between the 10^7 emission rate and the 10^6 impact rate in part (b) — are absorbed by the septa, meaning reduced sensitivity and increased Poisson noise.

In 3D mode, the septa are retracted and most of the photons impinge on the detector modules. Sensitivity and SNR are increased, but full 3D reconstruction algorithms must be employed. Also, fast detectors are essential to prevent the detection drop-off at the right of the graph in (c) above. [20%]

Aside. To check the appropriateness of the independence assumption in (c)(i), we can work out the joint probability precisely (this is absolutely *not* required by the question, but is included in this crib for interest). Consider the three events:

A. That one or more photon pairs hit detectors C and D in the time period t to $t + 10$ns.

B. That one or more photon pairs hit detectors C and D in the time period $t+10$ ns to $t + 300$ ns.

C. That one or more photon pairs hit the rest of the ring (not detectors C and D) in the time period t to $t + 10$ns.

These three events involve distinct photons and are therefore independent, so we can simply multiply to find joint probabilities. The probability of successful detection is the probability that none of these events occur.

$$\begin{aligned} p(\bar{A}\bar{B}\bar{C}) &= e^{-10/180\times10^6\times10\times10^{-9}} \times e^{-10/180\times10^6\times290\times10^{-9}} \\ &\quad \times e^{-170/180\times10^6\times10\times10^{-9}} \\ &= e^{-1/180\times10^6\times10^{-9}(10\times10+10\times290+170\times10)} = 0.974226836 \end{aligned}$$

This is almost the same as the estimate calculated in (c)(i).

Assessors' remarks: This question covered qualitative and quantitative aspects of nuclear medicine imaging. The qualitative parts were well answered, with most candidates exhibiting a good understanding of the strengths and weaknesses of PET and SPECT, and of 2D vs. 3D imaging. Less impressive were the candidates' calculations of attenuation in (b)(ii) and collision probabilities in (c)(i). Many candidates didn't know to use an exponential term for the former and the Poisson distribution for the latter, though three candidates did produce excellent answers.

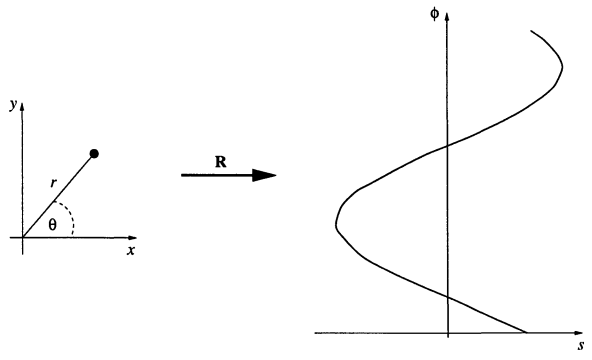
2. **X-ray computed tomography and the Radon transform**

(a) The two-dimensional Radon transform is defined as follows:

$$\mathcal{R}\left[f(x,y)\right] = \int_{-\infty}^{+\infty} f(s\cos\phi - l\sin\phi, s\sin\phi + l\cos\phi)\, dl$$

It maps $f(x,y)$ onto the set of its integrals over lines at perpendicular angles ϕ and distances s from the origin. In other words, it maps $f(x,y)$ onto the set of its one-dimensional projections at all projection angles ϕ. This essentially describes the forward X-ray CT process, in which a set of projections $p(s,\phi)$ is recorded of the attenuation function $\mu(x,y)$. The goal of the CT reconstruction algorithm is to recover $\mu(x,y)$ from $p(s,\phi)$, ie. invert the Radon transform. [20%]

(b) (i) The Radon transform of a single point in the (x,y) plane is zero everywhere except along a sinusoidal wave in the (s,ϕ) plane (this is why a 2D Radon transform is often referred to as a sinogram).



[10%]

(ii) The transform will be zero everywhere except where the line integral passes through the point $(r\cos\theta, r\sin\theta)$, ie. except where

$$s\cos\phi - l\sin\phi = r\cos\theta \quad \text{and} \quad s\sin\phi + l\cos\phi = r\sin\theta$$

To derive the equation of the non-zero locus, we need simply eliminate l from the above two equations.

$$\begin{aligned} \frac{l\sin\phi}{l\cos\phi} &= \frac{s\cos\phi - r\cos\theta}{r\sin\theta - s\sin\phi} \\ \Leftrightarrow r\sin\phi\sin\theta - s\sin^2\phi &= s\cos^2\phi - r\cos\phi\cos\theta \\ \Leftrightarrow r(\cos\phi\cos\theta + \sin\phi\sin\theta) &= s(\cos^2 + \sin^2) \\ \Leftrightarrow s &= r\cos(\phi - \theta) \end{aligned}$$

[20%]

(c) (i) $q(s)$ is the inverse Fourier transform of $|\omega|$:

$$q(s) = \mathcal{F}_1^{-1}\left[|\omega|\right] = \int_{-\infty}^{+\infty} |\omega| e^{i\omega s} d\omega$$

[10%]

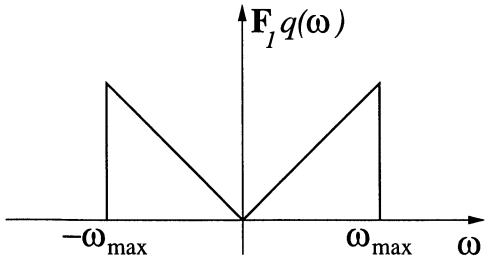
(ii) The inverse Radon transform can be described as follows:

(a) Start with the projection at $\phi = 0$. The integrand of the inverse Radon transform says to convolve the projection with the filter kernel $q(s)$. This is the "filtering" bit of the filtered backprojection algorithm. The result is a function of s only, where s is the direction perpendicular to the X-rays.

(b) The left hand side of the inverse Radon transform is defined over a two-dimensional domain (x, y), but is nevertheless a function of s only. Thus, the next step is to backproject the filtered projection so it covers the entire field of view in the (x, y) plane. This is the "backprojection" bit of the filtered backprojection algorithm.

(c) Finally, the integration part of the inverse Radon transform indicates accumulation of the backprojections over the full range of projection angles $0 \leq \phi \leq \pi$. [20%]

(iii) The filter kernel $q(s)$ is divergent in frequency and therefore unattainable in practice. An alternative, practical filter is the Ram-Lak filter, which is identical to the ideal filter up to a cut-off frequency $\omega_{\max}$:



[10%]

(iv) The width t of the X-ray beam determines the spatial sampling of the scanned subject. Any features with spatial frequency above $1/t$ (the first zero in the Fourier transform of the beam's pulse shape) will not be accurately captured by the scanner. The reconstruction filter can therefore be cut off at angular frequency $\omega_{\max} = 2\pi/t$. [10%]

Assessors' remarks: This question tested the candidates' understanding of the Radon transform and CT reconstruction by filtered backprojection. The question was mostly book-work and well answered by almost all the candidates who attempted it. The only part of the question to cause problems was (b)(ii), where candidates were asked to derive the Radon transform of a single point in the plane. Most of the correct answers built on a prior knowledge that the answer should be some sort of cosine wave: only a couple of candidates were able to derive the equation from first principles.

3. **B-splines and duplicated control points**

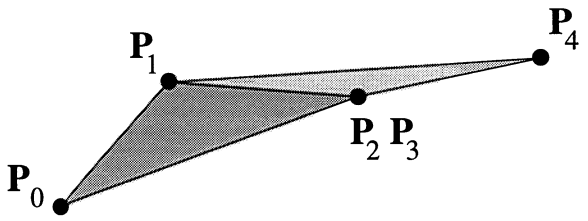
(a) The main factors are parametric and geometric continuity, interpolation vs. approximation, subdivision and the convex hull property.

Some sort of continuity is almost always essential when using cubic curve segments in computer graphics applications. Geometric continuity would suffice for static applications when the curve need only appear smooth to the eye. Parametric continuity would be an issue in any sort of animation application (eg. camera motion paths, robotic path planning, warping and morphing), where we require that the motion of a point along the curve is smooth. Of the forms mentioned in the question, the B-spline provides the best continuity (C^2 in the general case).

If the curve is required to interpolate all the control points, then the Catmull-Rom spline would be best. The Hermite form also interpolates the control points, though it also requires the definition of a number of tangents. Approximating curves, like B-splines, may only pass close to the control points, but generally exhibit better continuity.

When selecting a form to use as the internal representation inside some rendering software, the ease of subdivision is an issue. This is because the curve can be rapidly rendered by repeatedly subdividing the control points, then rendering the polyline formed by the control points: this is generally quicker than evaluating cubic expressions. Bézier curves are the best in this respect. Another issue for this sort of application is the convex hull property: by knowing that the entire curves lies inside the convex hull of its control points, various stages of the rendering process, especially intersection tests, can be speeded up. Again, Bézier curves are good in this respect (as are B-splines). [40%]

(b) Since B-splines exhibit the convex hull property, the join point between two segments must lie inside the convex hulls of both segments' sets of control points. When $\mathbf{P}_2 = \mathbf{P}_3$, the two convex hulls intersect along a line, as shown below.

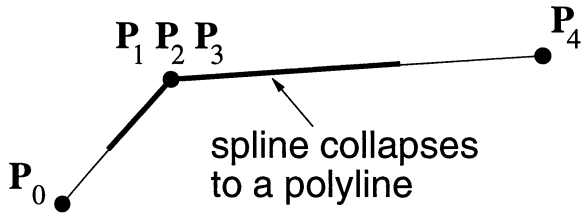


So the join point must lie somewhere on the straight line between $\mathbf{P}_1$ and $\mathbf{P}_2$. To find out exactly where, we evaluate the curve at the end of the first segment:

$$\mathbf{Q}_1(1) = [\,1\;1\;1\;1\,]\mathrm{M}_s\mathrm{G}_{s1} = \frac{1}{6}[\,0\;1\;4\;1\,]\begin{bmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_2\end{bmatrix} = \frac{1}{6}\mathbf{P}_1 + \frac{5}{6}\mathbf{P}_2 = \mathbf{P}_1 + \frac{5}{6}(\mathbf{P}_2 - \mathbf{P}_1)$$

So the join is 5/6 of the way along the line from $\mathbf{P}_1$ to $\mathbf{P}_2$. [25%]

(c) When three control points are coincident, the two convex hulls collapse to lines which meet at one point. The join point is at this point, and the curve, which must be contained within the two convex hulls, collapses to a polyline with a kink at the join point.



We have clearly lost G^1 continuity. To investigate parametric continuity, we need expressions for the derivatives of $\mathbf{Q}(t)$:

$$\mathbf{Q}'(t) = [\,3t^2\;2t\;1\;0\,]\mathrm{M}_s\mathrm{G}_s \;\text{ and }\; \mathbf{Q}''(t) = [\,6t\;2\;0\;0\,]\mathrm{M}_s\mathrm{G}_s$$

Evaluating $\mathbf{Q}''(t)$ at the $t = 1$ end of segment 1 and the $t = 0$ end of the segment 2 reveals that we still have C^2 continuity:

$$\mathbf{Q}_1''(1) = [\,6\;2\;0\;0\,]\mathrm{M}_s\mathrm{G}_{si} = \frac{1}{6}[\,0\;\;6\;\;-12\;\;6\,]\begin{bmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_1\\ \mathbf{P}_1\end{bmatrix} = 0$$

$$\mathbf{Q}_2''(0) = [\,0\;2\;0\;0\,]\mathrm{M}_s\mathrm{G}_{s(i+1)} = \frac{1}{6}[\,6\;\;-12\;\;6\;\;0\,]\begin{bmatrix}\mathbf{P}_1\\ \mathbf{P}_1\\ \mathbf{P}_1\\ \mathbf{P}_4\end{bmatrix} = 0$$

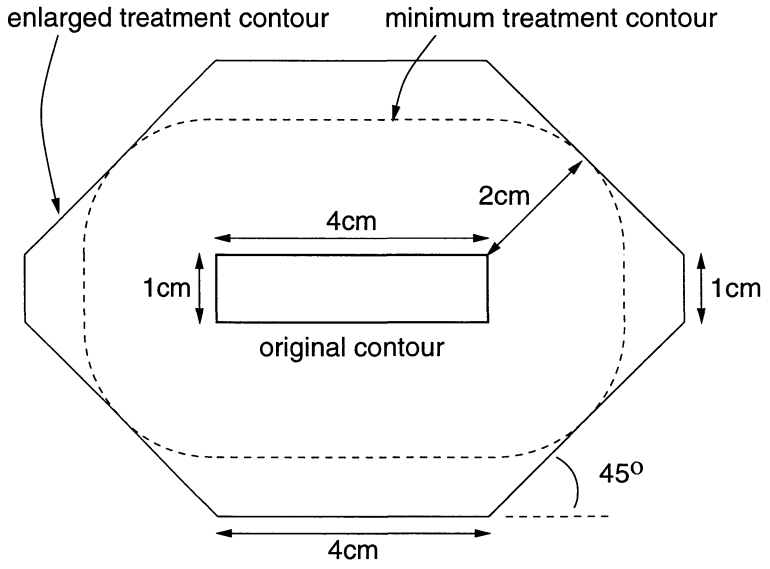
[25%]

(d) C^n continuity does imply G^n continuity, except when $\mathbf{Q}^{(n)}(t) = 0$. Under these circumstances, it is possible to have parametric continuity without geometric continuity. [10%]

Assessors' remarks: This question tested the candidates general knowledge of the properties of spline curves (Hermite, Bezier, B-spline and Catmull-Rom), as well as the concepts of convex hulls and continuity. This was fairly well answered by those who attempted it, though there was some confusion regarding what continuity the curves offered, and the distinction between geometric and parametric continuity. Part (b) was answered well — many candidates however dropped marks on part (c) by claiming G^2 continuity even though they had only proved C^2 continuity.

4. **Interpolation and distance transforms**

(a) (i)



[20%]

(ii) The actual treatment area A_i can be considered as the initial contour, plus four rectangles at each side, plus the four corner triangles, thus:

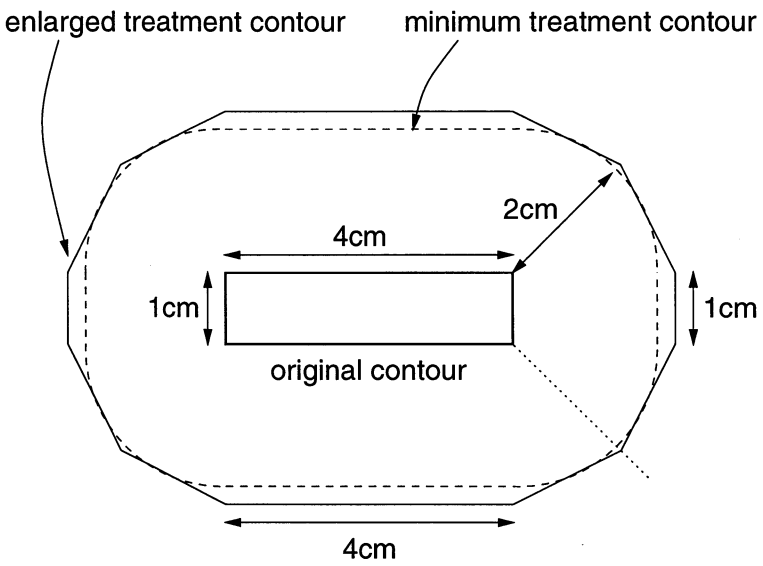
$$A_i = 4 * 1 + 2\sqrt{2} * (1 + 1 + 4 + 4) + 4 * \frac{1}{2} * 2\sqrt{2} * 2\sqrt{2} = 48.284 \text{ cm}^2$$

The minimum treatment area A_m is similarly given by:

$$A_m = 4 * 1 + 2 * (1 + 1 + 4 + 4) + 4 * \frac{1}{4} * \pi * 2^2 = 36.566 \text{ cm}^2$$

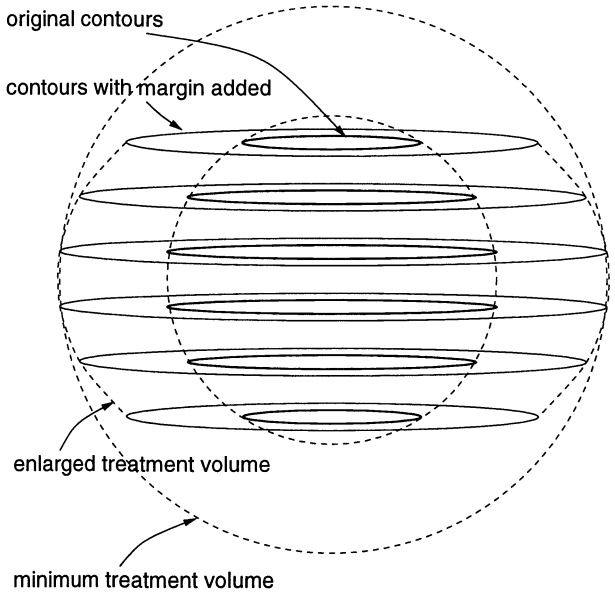
The percentage increase in area is thus 32%. [20%]

(iii)



This estimate produces a contour which is considerably closer to the required minimum treatment contour, and is probably acceptable given the large 2 cm safety margin. [20%]

(b) (i) In the first procedure, the distance transform was performed for each slice in 2D, which means it calculated the closest *planar* distance to each contour. This is not the same as the closest *3D* distance to the interpolated surface. For example, imagine a set of parallel contours through a spherical volume, drawn end on:



So the interpolation in the first procedure underestimates the required volume[1]. The

[1]This is not to be confused with the effect of the crude city-block distance estimate, which is to overestimate the required area in each of the original slices.

interpolation in the second procedure uses a 3D distance transform which produces a correct estimate. [20%]

(ii) The first step in shape-based interpolation is to calculate a 2D distance transform of each contour to be interpolated. In the first procedure, the 2D distance transform is interpolated between the slices and an isosurface extracted from the resulting volume. In the second procedure, the distance transformation is *after* interpolation. In practice, we would have to calculate 2D distance transforms for each contour in order to do the shape-based interpolation, but then throw these away and calculate a new 3D distance transform before extracting the isosurface. The second procedure is therefore much more expensive than the first. [20%]

Assessors' remarks: This question tested the candidates knowledge of distance transforms and shape-based interpolation. Only one candidate made a serious attempt at the question and did reasonably well, losing marks mainly on (a)(ii) by forgetting to add the safety margin to the minimum target area.

5. **Reflection models and shading algorithms**

(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 model can be extended to incorporate multiple light sources i, depth cueing and shadows as follows:

$$I_\lambda = c_\lambda I_a k_a + \sum_i S_i f_{\text{att}} I_{pi} (c_\lambda k_d \mathbf{L}_i.\mathbf{N} + k_s\,(\mathbf{R}_i.\mathbf{V})^n)\ .$$

The diffuse and specular terms are attenuated by a shadow factor S_i, where $0 \leq S_i \leq 1$. S_i is the fraction of the pixel shaded from the light source, often calculated using a shadow Z-buffer algorithm. There is also a depth attenuation factor f_{att}, usually of the form

$$f_{\text{att}} = \min\left(\frac{1}{a_1 + a_2 d + a_3 d^2}\ ,\ 1\right)$$

where a_1, a_2 and a_3 are constants, and d is the distance from the light source to the surface point. The depth cueing ensures that surfaces with the same orientation, but at different distances from the viewer, are not assigned the same intensity. [40%]

(b) The circumcentre is equidistant from each vertex, so interpolating the vertex normals to this point amounts to simple averaging of the three values. The vertex normals should be normalised before averaging, though in this case it makes little difference since they are all of similar magnitude. The surface normal $\mathbf{N}$ at the circumcentre is therefore $[0\ 0\ 1]^T$. The mirror direction $\mathbf{R}$ is coplanar with $\mathbf{N}$ and $\mathbf{L}$, with $\mathbf{N}$ bisecting $\mathbf{R}$ and $\mathbf{L}$. By inspection, the mirror direction $\mathbf{R}$ is therefore $[0\ 1\ 1]^T$. Substituting the values into the basic Phong equations, and remembering to normalise the vectors before taking the scalar products, we obtain $I_r = I_g = 1$, $I_b = 1.85$.

Note how the viewing direction is aligned precisely with the mirror direction, resulting in a prominent white specular highlight at the circumcentre, superimposed on the blue ambient and diffuse components. Given the high specular exponent n, the highlight will be far less prominent at each of the three vertices, where the mirror and viewing directions are not aligned. Since Gouraud shading interpolates intensities from the vertices into the interior, it is likely that the specular highlight at the circumcentre will be entirely missed, leaving just the blue ambient and diffuse terms. [40%]

(c) Gouraud shading proceeds by calculating a colour at each vertex using the vertex normal and the Phong model. Colours for interior pixels are found by bilinear interpolation. For efficiency, the interpolation can be formulated using fast, incremental calculations, amounting to one addition per rendered pixel.

Phong shading interpolates the normals instead of the intensities, a much more expensive operation. Even though the normals can be interpolated using incremental calculations, there are three values to be interpolated (the three components of the normals), compared to one for Gouraud shading (the intensity value). Also, the normal vectors have to be normalised, and then a *separate* intensity for each pixel calculated using Phong's model.

So Phong shading is much more expensive than Gouraud shading. Both are more expensive than flat shading, though the latter is only viable when viewing facetted surfaces under simple illumination.

Incremental bilinear interpolation, which lies at the heart of both Gouraud and Phong shading, requires one division per scanline to obtain the increments, then one addition per rendered pixel to calculate the intensities. For a fixed size triangle, the process

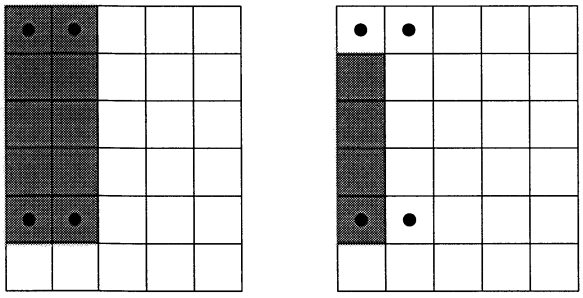
will be slightly quicker (by virtue of fewer divisions) when the triangle is oriented with its longest aspect horizontal, so that it spans the smallest number of scanlines. When the triangle is viewed end-on with respect to the pixel array, it doesn't need shading at all! [20%]

Assessors' remarks: This question was about the Phong reflection model and how it is employed in common shading algorithms. Parts (a) and (c) were largely book-work and well answered by the vast majority of candidates, who were clearly well-prepared for the examination. Part (b) was less well answered, largely because many candidates failed to realise that a triangle's circumcentre is equidistant from its three vertices. Despite this, most candidates were able to speculate intelligently on the likely benefits of Phong shading compared with Gouraud shading.

6. **Rasterisation and aliasing**

(a) *Rasterisation* refers to the process of displaying graphics primitives (lines, polygons etc.) on discrete, pixelised displays. Rasterisation algorithms generate a list of pixels which are shaded in order to display the primitive. More advanced rasterisation algorithms incorporate anti-aliasing features. Rasterisation is also sometimes referred to as *scan conversion.* [15%]

(b)



(i) The edge rasterisation algorithms outputs the pixels shown above left (the dots are the rectangle's vertices). [10%]

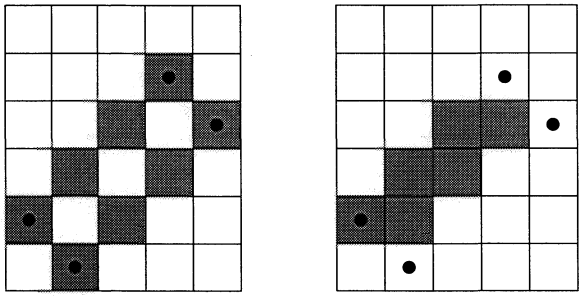
(ii) To preserve areas as closely as possible, the convention is that the top scanline of the polygon is discarded, and each span shaded up to but not including the rightmost pixel. This results in a rendered polygon of area four pixels, as shown above right. [15%]

(iii) A simple matrix multiplication reveals the vertex coordinates of the rotated rectangle:

$$\begin{bmatrix} \cos 45° & \sin 45° \\ -\sin 45° & \cos 45° \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 4 & 4 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2.82 & 3.54 & 0.71 \\ 0 & 2.82 & 2.12 & -0.71 \end{bmatrix}$$

Rounding to the nearest pixel, we find the rotated rectangle vertices at

$$\begin{bmatrix} 0 & 3 & 4 & 1 \\ 0 & 3 & 2 & -1 \end{bmatrix}$$



The rectangle would be rasterised as above, giving a rendered area of six pixels. [30%]

(c) Aliasing artefacts arise because images are rendered into a discrete sampling array in space (and, in the case of animations, in time). The most commonplace aliasing artefact is the ubiquitous jagged edge of a polygon or shadow. "Jaggies" are particularly noticeable in animated sequences, where edges appear to "crawl" and small or thin objects may appear and disappear depending on their orientation ("scintillate"). In (b) we saw a less severe example of scintillation, where the area of the rectangle changed by up to 50% depending on its orientation.

More advanced rasterisation techniques go some way towards suppressing aliasing artefacts. (i) "Supersampling", or "postfiltering", involves rendering the image into an intermediate framebuffer at n times the display resolution. The supersampled image is then low-pass filtered at the Nyquist limit of the final display framebuffer, which is filled by subsampling the intermediate framebuffer. (ii) "Prefiltering", or "area sampling", involves rasterising each polygon's edges to sub-pixel accuracy, so that each pixel can be classified as, for example, 30% polygon A, 20% polygon B and 50% polygon C. Then the intensity of the pixel is set to a weighted average of the individual polygon's contributions. Both techniques are computationally expensive, though hardware postfiltering is now a feature of desktop PC graphics cards. [30%]

Assessors' remarks: This question covered familiar topics in rasterisation, before asking the candidates to comment on some unfamiliar anti-aliasing schemes. Surprisingly, the candidates fared better with the unfamiliar material, coming up with some good discussions about how the schemes could work in practice. Less impressive were the answers to part (b), which asked the candidates to calculate the output of a rasterisation algorithm. There were many careless slips, perhaps due to lack of time.

Andrew Gee & Graham Treece
April 2004