Engineering Tripos Part IIA THIRD YEAR

Module 3G4: Medical Imaging & 3D Computer Graphics

Solutions to 2018 Tripos Paper

1. **Medical ultrasound imaging**

(a) Measurements in ultrasound images in the direction of ultrasound propagation (the axial direction) are based on an assumed sound speed of 1540 m/s. If the real speed of sound in the material is different, then the distances will appear either too large or too small in the resulting B-scan image. Distances parallel to the transducer face are measured in a different way by the imaging system, primarily based on the known distance between the transducer elements in the probe array. Distances in this direction are therefore not affected by speed of sound errors in the same way.

Liver has a sound speed that is faster than 1540 m/s, it therefore appears too thin in the B-scans because the sound delay between echos is shorter than it should be (as the sound speed is fast). Fat has a sound speed that is slower than 1540 m/s, therefore it appears too thick in B-scan images. [15%]

(b) (i) The distance between P and Q will appear longer in the image from scan B because it is in the axial direction of the scan. The perceived distance will be:

$$3.3 \times \frac{1540}{1447} = 3.51\,\mathrm{cm}$$ [20%]

(ii) The true distance between P and Q is 3.3 cm. [10%]

(iii) Let f be the factor by which axial distances are increased because of the speed of sound distortion. Using Pythagoras's theorem on distances from scan A and scan B, we get two simultaneous equations.

$$x^2 + (fy)^2 = 3.57^2$$
$$(fx)^2 + y^2 = 3.45^2$$

These are *linear* simultaneous equations in x^2 and y^2. We can thus solve for x^2 and y^2 easily.

$$x^2 = 2.60$$
$$y^2 = 8.95$$

Hence, the real distance between wire P and wire R is given by

$$\sqrt{2.60 + 8.95} = 3.40\,\mathrm{cm}$$ [40%]

(iv) The perceived areas are given by the following expressions.

$$\text{Scan A:} \qquad \frac{1}{2} \times \text{base} \times (f \times \text{height})$$

$$\text{Scan B:} \qquad \frac{1}{2} \times (f \times \text{base}) \times \text{height}$$

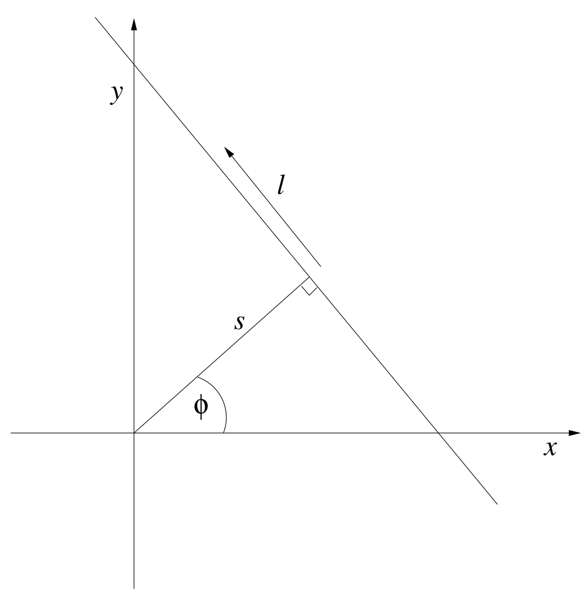
As these evaluate to the same thing, there will be no difference in the areas measured in scan A and scan B. [15%]

Assessors' remarks: This question tested candidates' understanding of the effect that variations in the speed of sound in tissue can have in ultrasonic imaging. The topic was generally well understood and the standard of answers was good. This resulted in a high average mark with 13 candidates scoring 18 or above out of 20. The weakest part of most answers was the calculation in (b)(iii). Most candidates were able to set up appropriate equations, but many made mistakes when attempting to solve them.

2. **Radon transforms**

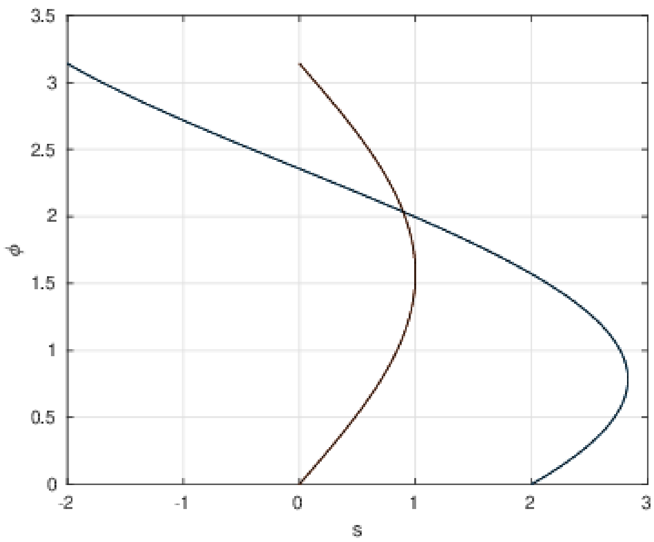
(a) Photoelectric absorption: the energy of the X-ray photon is completely absorbed as it ejects an electron from an inner shell. The excess energy of the photon over the binding energy of the electron is carried off as kinetic energy by the ejected electron. Lower energy characteristic radiation is emitted, in the same direction as the original photon, as an electron from an outer shell falls into the hole. [20%]

(b) The two-dimensional Radon transform maps a function $f(x, y)$ to the set of its integrals over lines at perpendicular angles ϕ and distances s from the origin.



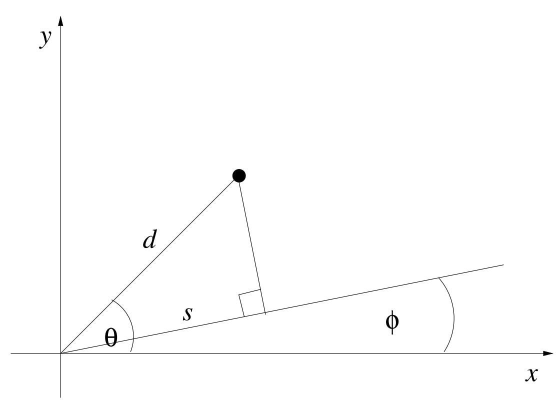
$$\mathcal{R}[f(x,y)] = \int_{-\infty}^{+\infty} f(s\cos\phi - l\sin\phi, s\sin\phi + l\cos\phi)\, dl$$ [20%]

(c)



[30%]

(d)



From the diagram above,

$$\frac{s}{d} = \cos(\theta - \phi)$$
$$\Rightarrow s = d\cos(\theta - \phi)$$

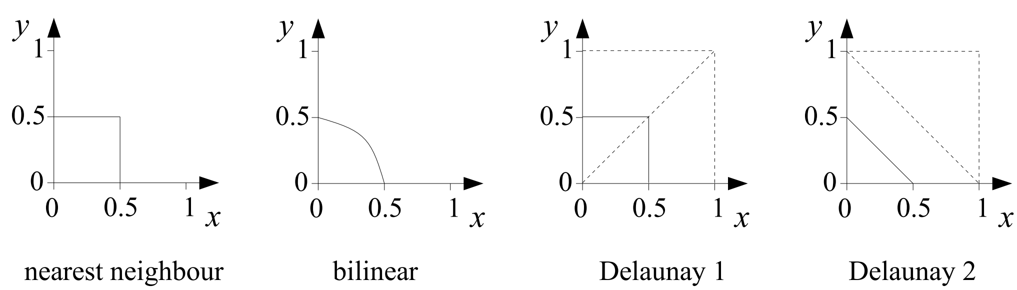
Hence, the Radon transform is made up of two components, the first with $\theta = \pi/4, d = 2\sqrt{2}$ and the second with $\theta = \pi/2, d = 1$. The Radon transform is therefore given by:

$$\mathcal{R}[f(x,y)] = \delta\left(s - 2\sqrt{2}\cos\left(\frac{\pi}{4} - \phi\right)\right) + \delta\left(s - \cos\left(\frac{\pi}{2} - \phi\right)\right)$$ [30%]

Assessors' remarks: This popular question tested candidates' understanding of the Radon transform. Some candidates found it quite challenging because it required them to really understand the principles behind the transform rather than simply to plug-in standard formulae. Nevertheless, a significant number of candidates produced high quality answers. As expected, most candidates found the more quantitative exercises in (c) and (d) the hardest. However, a small number of candidates got the algebra correct, but were unable to remember the details of photoelectric absorption in (a).

3. **2D data interpolation**

(a) Cubic parametric splines can easily provide C1 and G1 continuity (first order parametric and geometric) across the surface while interpolating the scalar data. They can also provide C2 and G2 continuity if they are allowed to approximate the data. It is also fairly easy to guarantee the convex hull property — i.e. the interpolated data is guaranteed to be within the limit of the sampled data on which it is based. [20%]



(b) (i) The figure above shows the result for nearest neighbour interpolation, the equations are:

$$\begin{aligned} x = 0.5 &\quad \text{for } 0 \leq y \leq 0.5 \\ y = 0.5 &\quad \text{for } 0 \leq x \leq 0.5 \end{aligned}$$

[10%]

(ii) The figure above also shows the results for bilinear interpolation. The interpolant weights the data from each corner of the unit square by one minus the distance from that corner. Since only one corner (at $(0, 0)$) has a nonzero value, the function is:

$$2(1 - x)(1 - y)$$

This is equal to 1 at the iso-contour, hence:

$$\begin{aligned} 2(1 - x)(1 - y) &= 1 \\ y &= 1 - \frac{0.5}{1 - x} \end{aligned}$$

[20%]

(c) (i) The Delaunay triangulation proceeds by ensuring that the circumcircle of each triangle does not contain any points from any other triangles. Since the four corners of the unit square are all on the same circumcircle, there are two different sets of triangles which are both correct Delaunay triangulations. These are shown dotted in the figure above. [10%]

(ii) The iso-contours are shown as continuous lines in the figure above. The scalar data at each vertex is first interpolated along the edge of each triangle to see where the iso-contour at 1 will intersect it: this is exactly at the mid-point of each edge, since the data at each vertex is 2 or 0. The iso-contour then comprises straight lines joining these points, since linear interpolation within a triangle produces a planar data surface. [20%]

(d) α affects the radial extent of the basis function, which is a 2D Gaussian. Smaller values of α will result in larger basis functions, since r must consequently be larger to achieve the same exponent. But the basis function will always interpolate the data, so this has the effect of increasing the *tension* of the interpolant, i.e. if it is a rubber sheet, how tightly it is stretched to the data.

The size should ideally be similar to the sample spacing: much smaller and there will just be individual Gaussian functions at each sample, much larger and the interpolant becomes less smooth. Given that the horizontal and vertical spacing is 1, any value of α around 1 is sensible. [20%]

Assessors' remarks: This question tested the candidates' understanding of various interpolation techniques applied to rectilinear sampled 2D scalar data. Answers to (a) demonstrated good knowledge of cubic splines, but too often this was in the context of curve-fitting rather than, as the question asked, interpolating 2D scalar data. (b)(i), on nearest-neighbour interpolation, was handled well, whereas correct answers were rarer for the bilinear interpolation in (b)(ii), with even fewer noting that the iso-contour was not linear. (c), on Delaunay triangulation, was answered very well. Most candidates also produced good answers to the Radial Basis Function question in (d), including sensible values for α.

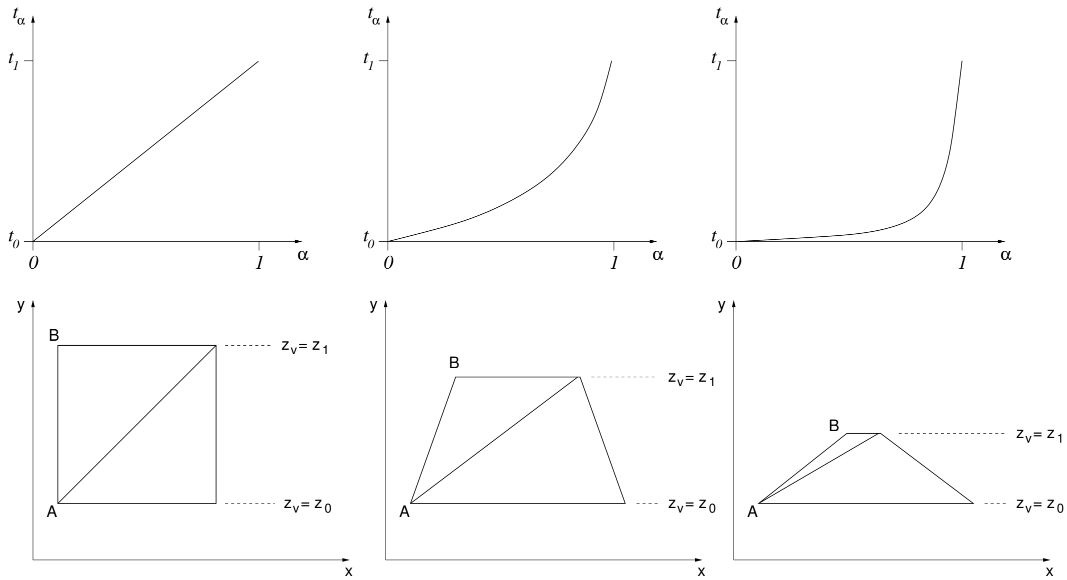
4. **Interpolation and texture mapping**

(a) Both Gouraud and Phong shading work with *vertex normals*, which are found by averaging the normals of all polygons incident at a vertex. 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.

Phong shading interpolates the normals instead of the intensities. Even though the normals can be interpolated using incremental calculations, the interpolation handles the three components independently, so the vector must be renormalized at each pixel. Then, a *separate* intensity for each pixel is calculated using the Phong model.

For both shading methods, bilinear interpolation is used to derive depth (z_s) values within the interior from those at the vertices. [25%]

(b) (i)



From left to right: $z_1 = z_0$, $|z_1| > |z_0|$ and $|z_1| \gg |z_0|$. [30%]

(ii) The extreme case illustrates the potential for significant magnification of the texture map at the front of the painting, since a small number of texture rows are spread over a large number of screen rows. If t_α is used to sample just a single texture value (i.e. nearest neighbour interpolation), then the rendered image will be very blocky at the front of the painting. These magnification artefacts could be suppressed by bilinear interpolation of the texture map, with the texture at each screen pixel set to a distance-weighted average of the four texture samples surrounding each (s_α, t_α) value. Higher order interpolation is also possible, e.g. cubic.

At the back of the painting, in contrast, there is significant minification of the texture map, since a large number of texture rows are spread over a small number of screen rows. If t_α is used to sample just a single texture value, we can expect a fairly random outcome, depending on which texture rows happen to be picked up. This is essentially an aliasing artefact caused by severe undersampling of the texture map. To fix this, we could generate a smaller version of the texture map, by proper downsampling with anti-aliasing smoothing. Then we could safely index into this smaller texture map at the back of the painting. The downsampled versions of the texture map can be precomputed and then selected when required: this is called *mipmapping*. [40%]

Assessors' remarks: This question tested the candidates' knowledge of Gouraud and Phong shading, texture mapping and texture interpolation/aliasing. The book work in (a) was well answered, the only common misunderstanding being the type of interpolation required for z_s. In (b)(i), almost all candidates identified how perspective interpolation becomes linear in the special case, though there were many inaccurate attempts at sketching the interpolating function for the general case. Candidates were evidently attempting to sketch the function directly from its formula (tricky), instead of leveraging their understanding of the nature of perspective projection. In (b)(ii), most candidates mentioned aliasing, and around half identified texture expansion at the front and compression at the back, but a disappointing few came up with good suggestions for suppressing the consequent artefacts.

Andrew Gee, Richard Prager & Graham Treece
May 2018