Module 3G4: Medical Imaging & 3D Computer Graphics

Solutions to 2015 Tripos Paper

1. **Ultrasound beamforming and focusing**

(a) The delay between firing the outer pair and the centre element can be found using Pythagoras's theorem. The distance between the mid-line of the centre element and the mid-line of the outer element is $31 \times 0.26 = 8.06$ mm. The required focus depth is given as 40 mm. By Pythagoras's theorem, the diagonal distance from the focal point to an outer element is therefore 40.804 mm. The extra 0.804 mm takes the ultrasound a time of 0.522 μs at 1.54 mm/μs.

The outer elements should therefore fire 0.522 μs before the centre element. [25%]

(b)(i) The echo time for the centre element is given by the round-trip travel time. For a depth z this will be $t_0 + 2z/v$ or

$$t_0 + \frac{2z}{1.54 \times 10^6} \text{ seconds} \qquad [25\%]$$

(b)(ii) The time for an echo to reach the outer-most element is found by working out the distance that the beam has to travel. First it goes out a distance z, then it comes back a distance $\sqrt{z^2 + 8.06^2}$. Hence the time is:

$$t_0 + \frac{z + \sqrt{z^2 + 8.06^2}}{1.54 \times 10^6} \text{ seconds} \qquad [25\%]$$

(b)(iii) The required delay as a function of z is given by;

$$\frac{\sqrt{z^2 + 8.06^2} - z}{1.54 \times 10^6} \text{ seconds}$$

Using $t = 2z/(1.54 \times 10^6)$ we obtain the delay $d(t)$ as follows:

$$d(t) = \sqrt{\frac{t^2}{4} + 2.74 \times 10^{-11}} - \frac{t}{2}$$

This tells us that a signal arriving at the centre element of the active aperture at time t should be delayed by $d(t)$ before being added to the signal from the outer-most element as part of the process of dynamic receive focusing. [25%]

Assessors' remarks: This question tested the candidates' understanding of transmit and receive focusing in ultrasonic medical imaging. The first part of the question required

a simple calculation of transmit delays for a fixed transmit focus. The second part of the question was carefully scaffolded and asked for a calculation of the time delay for dynamic receive focusing. Candidates performed well in both parts of the question. They were particularly successful in the early part of the dynamic receive focusing calculation. It was pleasing to note that this complex concept had been so well understood by the candidates.

2. **Magnetic resonance imaging**

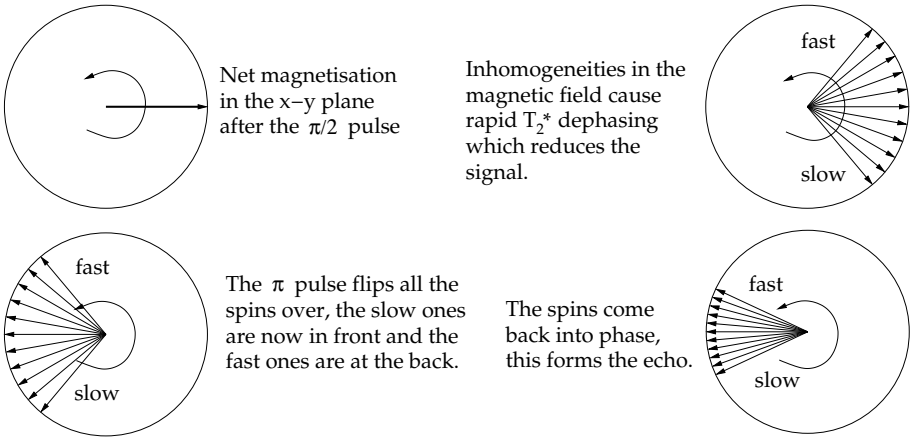
(a) When the net nuclear spin of a region of protons is disturbed from alignment with an external magnetic field, it will precess around the field direction at the Larmor frequency. The Larmor frequency is given by the product of the field strength and the gyromagnetic ratio of the material. For hydrogen nuclei, i.e. protons, the gyromagnetic ratio is 42.58 MHz/Tesla. [20%]

(b) For a field of 5 Tesla, the proton spin will precess at $5 \times 42.58 = 212.9$ MHz. [10%]

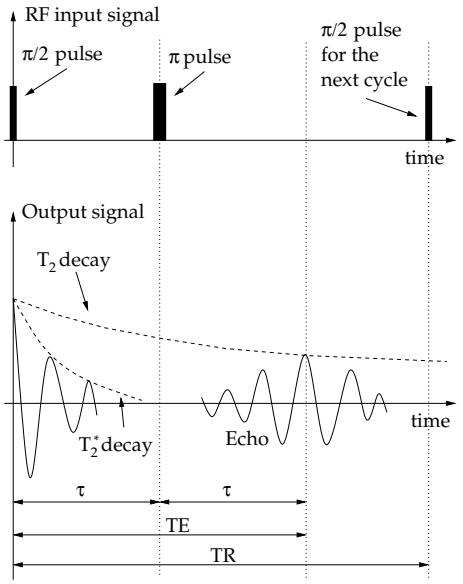
(c) T_1 is called "spin-lattice" relaxation. Protons revert from the excited anti-parallel state to the parallel state, until thermal equilibrium is restored. Energy is dissipated into the atomic and molecular environment — the lattice. The effect is recovery of the net magnetisation vector in the direction of the external field (the equilibrium z direction). The time constant of the T_1 process depends on the material. [20%]

(d) longest: T_1, T_2, T_2^* :shortest. [10%]

(e) The spin-echo imaging sequence is a way of overcoming the effect of T_2^* relaxation which causes rapid decay of the x–y component of the net spin because of local inhomogeneities in the magnetic field. The spin-echo sequence involves two RF pulses. The first pulse is designed to caused a 90° rotation to move the spins from alignment with the main z field into the x–y plane. The spins therefore begin to precess, but the signal decays rapidly because of T_2^* relaxation.



The second RF pulse is then applied which flips the spins through 180° and places the fastest moving spins at the back of the pack and the slowest ones at the front. The T_2^* inhomogeneities then cause the spins to come back into phase in the x–y plane and the output signal returns for a short time, before they dephase again.



MRI data is acquired by repeating the spin-echo sequence many times. We create a T_1-weighted image by reducing the repetition time, TR, between individual spin-echo sequences to the point where it is too short for the longitudinal (z) component of the spin to recover completely between sequences. Hence material with a short T_1 will have recovered most and will be most able to respond correctly to the next spin-echo sequence.

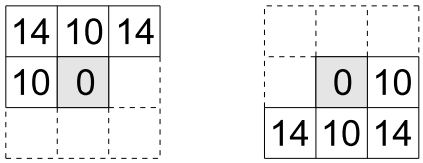
For a T_1-weighted image we have short TR, short time to echo (TE). As a result, tissue with a short T_1, like for instance fat, will respond strongly and come out bright in the image. [40%]

Assessors' remarks: This question tested the candidates' understanding of magnetic resonance, T_1 relaxation, and the spin echo imaging sequence. The answers were generally of a good quality but some lacked detail, and some candidates found it difficult to present their description of the concepts in a logical and coherent way. The most difficult concept was clearly the process of T_1 relaxation. Most candidates were aware of the relaxation process in general terms, but few could say which part of it related to T_1. Surprisingly, a sizeable minority of candidates could not explain what was meant by "Larmor frequency" or "gyromagnetic ratio".

3. **Distance transformations**

(a) A distance transform is an image where each pixel is replaced by a number representing the shortest distance to the border of some predefined object. The steps in calculating a distance transform from thresholded data are as follows.

- Go through the thresholded data and label each pixel which is next to the border of the object (in a 4-connected sense) as $+5$ (if inside the object) and -5 (if outside the object). All other pixels should be labelled with large positive values (if inside the object) or large negative values (if outside).



- Use a forward mask (as above left) to process the data left-to-right and top-to-bottom. With the central grey square over the current pixel, replace the pixel value with the *least* of each of the mask values *added* to the underlying pixel values. If the current pixel is negative (outside the object), replace with the *greatest* of each of the mask values *subtracted* from the underlying pixel values.
- Having processed all the data in this manner, use the backward (right hand) mask to process the data right-to-left and bottom-to-top, in the same manner as above.
- Finally, the border values (which could not be reached by the mask) can be set from their neighbouring pixels.

City-block distances are those only measured in horizontal and vertical directions, i.e. the distance from A to B if constrained only to walk in these directions. *Chamfer* distances, as those in the masks above, add diagonal directions to these. *Euclidean* distances are as we would normally measure distance, i.e. in the straightest direct path from A to B. [30%]

(b) Possible uses of the distance transform include:

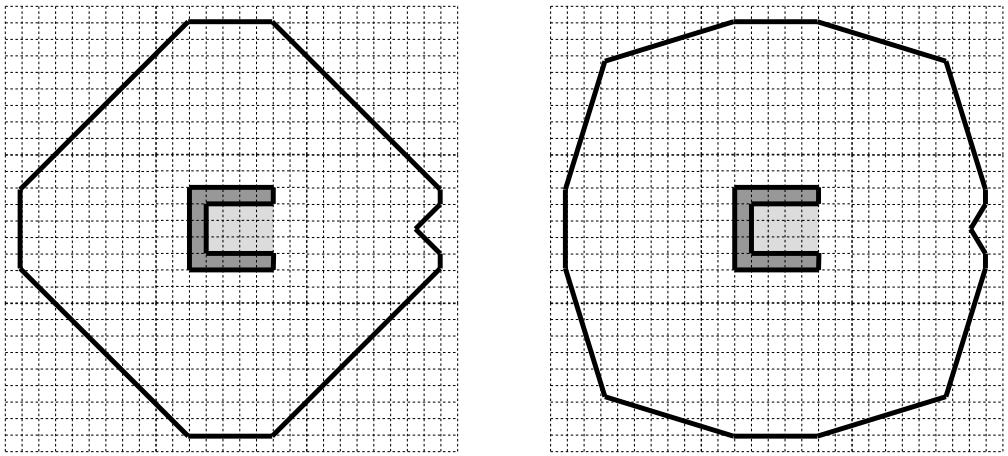
Shape-based interpolation This is where a new intermediate shape is created in-between two other shapes. Each shape is first distance-transformed, then these distance transforms are linearly interpolated, then finally the resulting new transform is thresholded at zero to give the new shape. This can be useful in creating 3D surfaces from sets of cross-sections of an object.

Shape expansion or contraction Thresholding a distance transform at a nonzero value will create a shape which is either larger or smaller than the original shape. This can be useful in radiotherapy planning, for adding a margin of error around a desired planning volume.

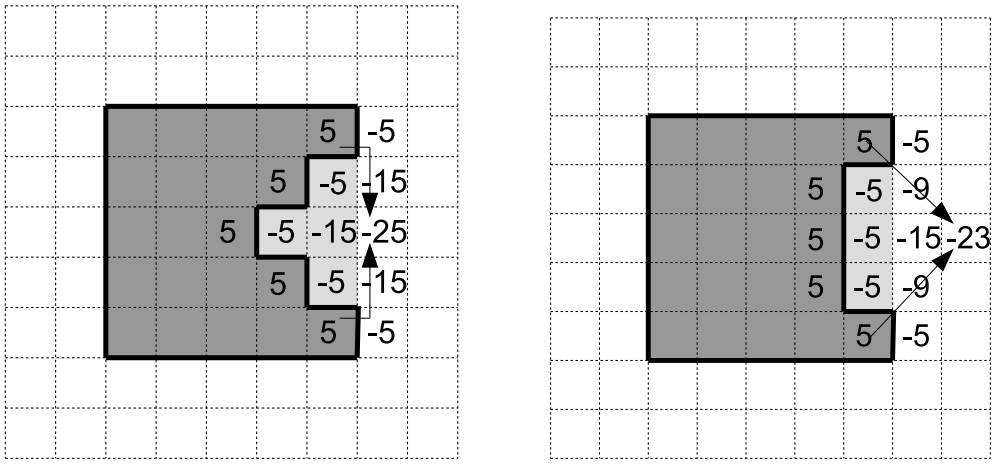
Shape description A shape can be described concisely by its skeletonisation, derived from the peak (inside, positive) values of the distance transform of the shape.

Registration Two different shapes can be aligned by comparing the differences between the distance-transformed values at different relative alignments. The best alignment has the least overall difference.

[15%]



(c) (i) The images above give the far-field result for the city-block transform (left) and chamfer transform (right). Notable features are the vertical and horizontal lines directly to the left, top and bottom of the shape, which retain the shape and extent of the original 'C'; the line directly to the right which only contains a slight indentation; and the remaining lines which form a diamond shape (city block) or octagon (chamfer). [20%]



(ii) The images above give the pixels which can be added to the 'C' without affecting the result of (i) for the city-block (left) and chamfer (right) transforms. The numbers show the important distance transformed values: in each case, the pixel at the far right is still derived from the corner pixels of the 'C', and hence any pixels beyond these are not affected by the additional pixels added to the centre. [20%]

(iii) The main consequence of (i) is that the far-field shape is increasingly not affected by the actual original shape, but instead is affected by the choice of distance transform. If far-field distances are important, then it is better to use a more accurate distance transformation. (ii) shows that small features can sometimes have only a limited local effect on the distance transform. In general this is a good thing: whilst we want to preserve these

features locally, they should not affect how the whole shape is treated. So, for instance, in shape-based interpolation, the concavity in the middle of the 'C' would be preserved, but it would have very little impact on the location of any external shape edges. [15%]

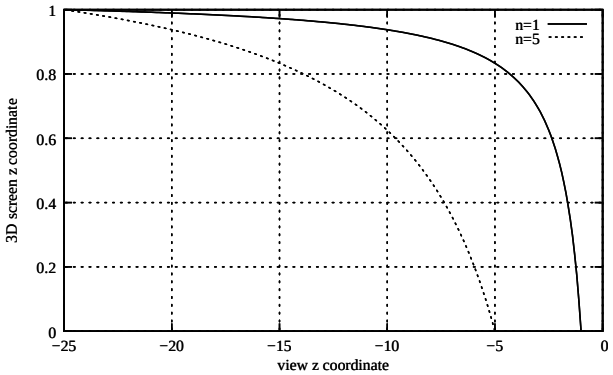
Assessors' remarks: This question concerned distance transformations. The book work in (a) was very well answered, demonstrating that candidates did understand the overall concept, though few could suggest three applications in (b). Answers to (c) were more variable, with some candidates providing perfect solutions for the far-field transform shapes and ignorable pixels, while others clearly had a very limited understanding of what was required, occasionally not even noting that the threshold of -100 was well away from the actual shape. Answers to (c)(iii) were generally reasonable, but many candidates lost marks here by not even attempting a discussion of these points.

4. **3D screen space, z-buffers and hidden surface removal**

(a) The z-buffer algorithm is used for hidden surface removal in most practical implementations of the surface rendering pipeline. Earlier stages of the pipeline express all polygon vertices in 3D screen coordinates (x_s, y_s, z_s). z_s is a normalised representation of a point's depth, in the range 0 to 1. Nearby vertices have small z_s, distant ones have larger z_s. Depth values for individual pixels are derived by bilinear interpolation of the vertex values.

The z-buffer is an area of memory with the same dimensions as the frame buffer. When we write a pixel into the frame buffer, we also write its z_s value into the z-buffer. If a subsequent polygon attempts to shade the same pixel as an earlier one, we compare the new z_s with the value currently in the z-buffer, and write over the existing pixel in the frame buffer and the z-buffer only if the new point is nearer to the viewer. Initially, all entries in the z-buffer are set to $z_s = 1$. [20%]

(b) (i)

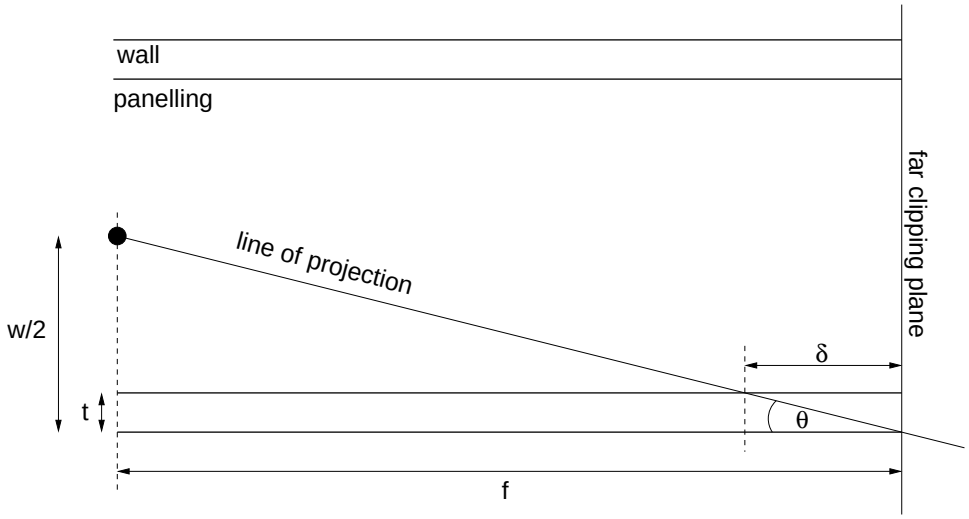


The curves above are for $f = 25$, $n \in \{1, 5\}$. [10%]

(ii) The nonlinear definition of z_s normalises depth values to the range 0 to 1. More importantly, it is designed to ensure that lines map to lines and planes map to planes. If this were not the case, we would not be able to use linear interpolation in 3D screen space for clipping and depth interpolation. [10%]

(iii) In hardware implementations, z_s is quantised to a finite set of values. At the back of the view volume, a significant range of z_v values will produce the same quantised z_s value, making depth discrimination impossible. This is especially the case when the ratio of the far to near clipping plane distances is large: compare the $n = 1$ and $n = 5$ curves. [10%]

(c) (i) The viewing geometry is shown below.



It is clear from the diagram that

$$\tan\theta = \frac{w/2}{f} = \frac{t}{\delta} \Leftrightarrow \delta = \frac{2tf}{w}$$

δ is the difference between the z_v values of the wall and the panelling at the back of the view volume. The problem is that δ is so small in Fig. 1(a) that the wall and panelling map to the same quantised z_s value. The artefact does not affect Fig. 1(b) since w is smaller and δ is consequently larger. [15%]

(ii) Assume a k-bit z-buffer and that the artefact just kicks in at the back of the view volume in Fig. 1(a). It follows that for the panelling at $z_v = -f + \delta$, $z_s = 1 - 2^{-k} + \epsilon$ (where ϵ is vanishingly small), since this will be stored as 2^k in the hardware z-buffer. Hence

$$\begin{aligned} 1 - 2^{-k} &= \frac{f(1 + n/(-f + \delta))}{f - n} = \frac{1000(1 + 1/(-1000 + 0.06))}{999} = 1 - 6.006 \times 10^{-8} \\ \Leftrightarrow k &= -\log_2 6.006 \times 10^{-8} = 24.0 \end{aligned}$$

So the z-buffer precision appears to be around 24 bits. [25%]

(iii) There is no need for the near clipping plane to be so near. Given the 30° field of view, we cannot even see the walls until they are around 370 m from the viewpoint. Setting n to, say, 10 m would remove the artefact. [10%]

Assessors' remarks: This question tested candidates' understanding of 3D screen coordinates and the z-buffer algorithm. Most candidates managed to describe the algorithm correctly and explain that its precision is compromised when f/n is large. Answers to (b)(ii) were disappointing, with far too many candidates claiming incorrectly that the particular z_s–z_v relationship is responsible for perspective foreshortening in the image. The correct answer (that this formulation preserves collinearity and coplanarity for linear interpolation) should have been familiar from question 9(d) on the examples paper. Also disappointing was the candidates' inability to perform simple geometric calculations in (c)(ii): there were only two completely correct answers.

Andrew Gee, Richard Prager & Graham Treece
May 2015

Examiner's Comments

Question 1.
This question tested the candidates' un- derstanding of transmit and receive focusing in ultrasonic medical imaging. The first part of the question required a simple calculation of transmit delays for a fixed transmit focus. The second part of the question was carefully scaffolded and asked for a calculation of the time delay for dynamic receive focusing. Candidates performed well in both parts of the question. They were particularly successful in the early part of the dynamic receive focusing calculation. It was pleasing to note that this complex concept had been so well understood by the candidates.

Question 2.
This question tested the candidates' understanding of magnetic resonance, T1 relaxation, and the spin echo imaging sequence. The answers were generally of a good quality but some lacked detail, and some candidates found it difficult to present their description of the concepts in a logical and coherent way. The most difficult concept was clearly the process of T1 relaxation. Most candidates were aware of the relaxation process in general terms, but few could say which part of it related to T1. Surprisingly, a sizeable minority of candidates could not explain what was meant by "Larmor frequency" or "gyromagnetic ratio".

Question 3.
This question concerned distance transformations. The book work in (a) was very well answered, demonstrating that candidates did understand the overall concept, though few could suggest three applications in (b). Answers to (c) were more variable, with some candidates providing perfect solutions for the far-field transform shapes and ignorable pixels, while others clearly had a very limited understanding of what was required, occasionally not even noting that the threshold of −100 was well away from the actual shape. Answers to (c)(iii) were generally reasonable, but many candidates lost marks here by not even attempting a discussion of these points.

Question 4.
This question tested candidates' understanding of 3D screen coordinates and the z-buffer algorithm. Most candidates managed to describe the algorithm correctly and explain that its precision is compromised when f/n is large. Answers to (b)(ii) were disappointing, with far too many candidates claiming incorrectly that the particular zs– zv relationship is responsible for perspective foreshortening in the image. The correct answer (that this formulation preserves collinearity and coplanarity for linear interpolation) should have been fa- miliar from question 9(d) on the examples paper. Also disappointing was the candidates' inability to perform simple geometric calculations in (c)(ii): there were only two completely correct answers.