Mechanical Engineering

Computer Graphics and Image Processing

1. Describe an algorithm to draw a straight line using only integer arithmetic.You may assume that the line is in the first octant, that the line starts and ends at integer co-ordinates, and that the function setpixel(x, y) turns on the pixel at location (x,y). [8 marks]
2. Describe Douglas and Pucker's algorithm for removing superfluous points from a line chain. [10 marks]
3. Under what circumstances would it be sensible to employ Douglas and Pucker's algorithm? [2 marks]
4. Describe the limitations of human vision in terms of:
   i. spatial resolution
   ii. luminance
   iii. color

   and explain the implications that each of these limitations has on the design of display devices. [10 marks]
5. In image compression we utilize three different mechanisms to compress pixel data:
   i. mapping the pixel values to some other set of values
   ii. quantizing those values
   iii. symbol encoding the resulting values

   Explain each mechanism, describe the way in which it helps us to compress the image, and describe in what way it affects the visual quality of the resulting (decompressed) image when compared with the original. [10 marks]
6. Describe the z-buffer polygon scan conversion algorithm. [10 marks]
7. In ray tracing, once we have determined where a ray strikes an object, the illumination at the intersection point can be calculated using the formula:

$$I = I_a k_a + \sum_i I_i k_d (\mathbf{L}_i \cdot \mathbf{N}) + \sum_i I_i k_s (\mathbf{R}_i \cdot \mathbf{V})^n$$

   Explain what real effect each of the three terms is trying to model and explain what each of the following symbols means, within the context of this formula:

$$I,\ I_a,\ i,\ I_i,\ k_a,\ k_d,\ k_s, \mathbf{L}_i,\ \mathbf{N},\ \mathbf{R}_i,\ \mathbf{V},\ n$$

[10 marks]

8. Describe an algorithm which draws a Bezier cubic curve to a specified tolerance using straight lines. [7 marks]
9. Describe an algorithm for clipping a line against a rectangle. [8 marks]
10. A Bezier cubic curve could be clipped and drawn using the algorithm in (a) to produce straight lines and the algorithm in (b) to do the clipping. Describe a more efficient algorithm which draws a Bezier cubic curve clipped against a rectangle. [5 marks]
11. With reference to the characteristics and performance of the human visual system, provide an estimate for each of the following. In each case you are expected to justify your estimate:
    i. the maximum resolution required by a display device;
    ii. the maximum number of distinct intensity levels required by a display device;
    iii. the optimal number of dimensions required to represent colour;
    iv. the maximum refresh rate required of a CRT monitor.

    [5 marks]
12. A programmer suggests three different implementations of a polygon drawing algorithm:
    i. standard z-buffer;
    ii. standard A-buffer with an 8 x 8 mask size;
    iii. standard z-buffer at 8 x 8 normal resolution followed by averaging operation which produces a normal resolution image by finding the average value of each 8 x 8 block.

    Compare the three implementations in terms of both execution speed and resulting image quality. Which would be the best implementation to use if the average polygon covers 50 pixels? Which would be the best implementation to use if the average polygon covers 2 pixels? Which would be the best implementation to use if the display resolution was equal to the answer that you gave to (a)(i) above? [7 marks]
13. Show how to perform 2D rotation about an arbitrary point. Provide a matrix in homogeneous coordinates for each step in the operation. [2 marks]
14. Show how to perform 3D rotation about an arbitrary axis. Again, give matrices in homogeneous coordinates for each step in the operation. [6 marks]

15. Pick two of these three colour spaces: Lab, CMYK, HLS. For each of your chosen two colour spaces, explain what each of the dimensions represents and for what uses the colour space is best suited. [6 marks]
16. Describe a run-length encoding method for greyscale images. [6 marks]
17. The following is a Bezier curve drawing algorithm which includes bounding box clipping. Provide pseudocode for the functions **InBoundingRect** and **DrawUnclippedBezier**.

```
function DrawClippedBezier(float x1, y1, x2, y2, x3, y3, x4, y4)
begin
  if NearlyStraight(x1, y1, x2, y2, x3, y3, x4, y4)
  then DrawClippedLine(x1, y1, x4, y4)
  else begin
      r = InBoundingRect(x1, y1, x2, y2, x3, y3, x4, y4) ;
      if(r==0) then DrawUnclippedBezier(x1, y1, x2, y2, x3, y3, x4, y4);
      if(r==1) then begin
          DrawClippedBezier(x1,y1, (x1+x2)/2,(y1+y2)/2,
            (x1+2*x2+x3)/4,(y1+2*y2+y3)/4,
            (x1+3*x2+3*x3+x4)/8,(y1+3*y2+3*y3+y4)/8);
          DrawClippedBezier((x1+3*x2+3*x3+x4)/8,(y1+3*y2+3*y3+y4)/8,
            (x2+2*x3+x4)/4,(y2+2*y3+y4)/4,(x3+x4)/2,(y3+y4)/2,x4,y4);
      end ;
      if(r==2) then return ;
  end ;
end;
```

Notes: The bounding rectangle is defined by the four (global) floating point variables **left, right, top, and bottom**. You may assume that we have two-line drawing functions available **DrawClippedLine** and **DrawUnclippedLine**. The former draws a line having first clipped it to the bounding rectangle, the latter just draws a line without regard for the bounding rectangle (which should therefore only be used if the programmer has assured him- or herself that the line will not extend beyond the bounding rectangle). The two functions **DrawClippedBezier** and **DrawUnclippedBezie**r do the same for Bezier curves. The function NearlyStraight returns true if the Bezier curve lies within half a pixel of a straight line from its first to its last point along its entire length, otherwise it returns **false**. [8 marks]