CS184 : Foundations of Computer Graphics

Professor David Forsyth
Final Examination

(Total: 100 marks)

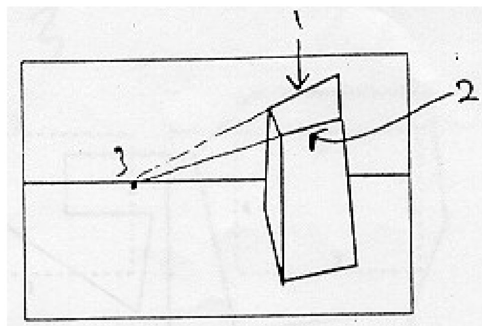

Figure 1: *A perspective view of a polyhedron on an infinite plane.*

Cameras and Perspective

Rendering in a perspective camera requires rotating and translating from world coordinates to camera coordinates (sometimes called the *viewing transformation*) and then transforming the frustrum to a canonical form (call this transformation the *view mapping* transformation).

1. Which camera parameters are required to determine the viewing transformation (3)?

2. Which camera parameters are required to determine the view mapping transformation (i.e. from camera coordinates to the canonical frustrum) (3)?

3. Give one advantage and one disadvantage of clipping in homogenous coordinates (2).

4. Figure 1 shows a perspective view of a world containing a polyhedron on an infinite plane. The polyhedron contains two lines that are visible, parallel and parallel to the plane. Mark them, explaining your reasons. (4)

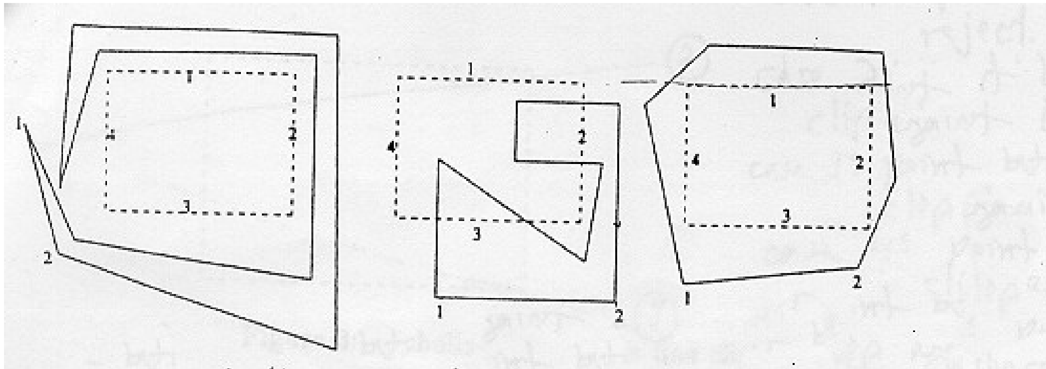

Figure 2: *Polygon clipping question: the dashed rectangles are the clip regions, which are traversed in the order shown by numbers next to each edge. The polygons are traversed in the order given by the two marked vertices.*

Clipping

- Each of the polygons given in figure 2 is clipped using Sutherland-Hodgeman polygon clipping against the triangle shown: the vertices of the polygon are traversed in the order show, and the polygon is clipped against the edges of the square in the order show. Supply a drawing of the results - there is space in the figure -, with vertices marked to indicate the order in which they are produced (9).
- Figure 3 shows a square clip boundary with a single vertex. Nicholls-Lee-Nicholls line clipping is efficient, because it does a case by case analysis of the edges against which a line must be clipped. Given that one vertex of the line is the one shown, draw on top of the figure the clipping cases that would occur for the other vertex (5).
- Give:

1. the maximum number of vertices obtained by clipping an n-vertex convex polygon using Sutherland-Hodgeman against an m-vertex convex polygon in the plane (3) :
2. one reason Sutherland-Hodgeman polygon clipping fails for clipping a general polygon against a general polygon (1):
3. one reason parametric line clipping is more efficient than Cohen Sutherland line clipping (1):
4. one disadvantage of Nicholls-Lee-Nicholls line clipping (1):

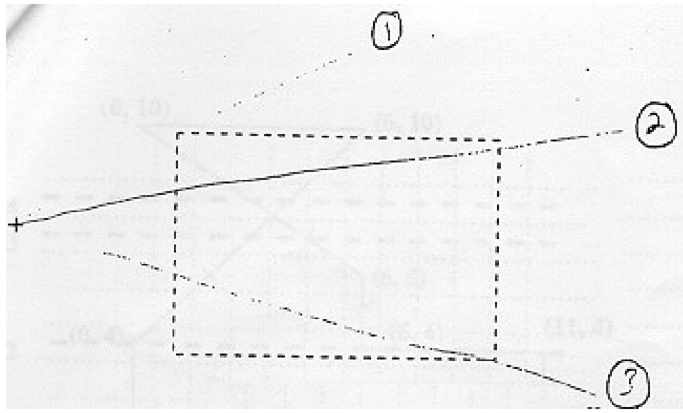

Figure 3: *Nicholls-Lee-Nicholls line clipping; vertex 1 is the cross.*

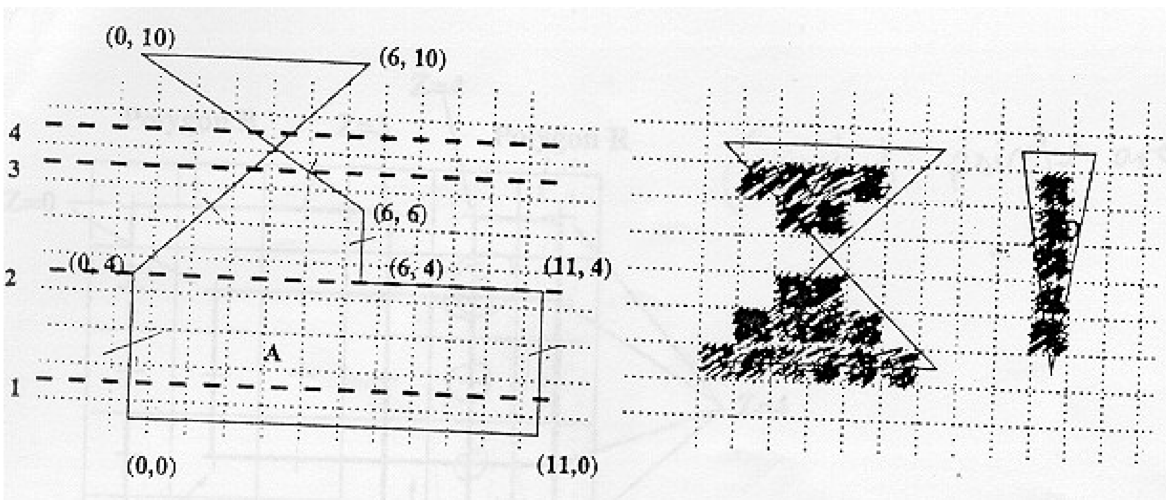

Figure 4: *Polygons for the scan conversion question.*

Polygon scan conversion

1. For polygons C and D in figure 4, fill in the pixels that lie inside the polygon, using the convention used in class (4).

2. For polygon A in figure 4, show the edge table (4):

3. For each of the four scanlines drawn in the figure on polygon A, show the state of the active edge list immediately before the pixels on that scanline are filled (8) :

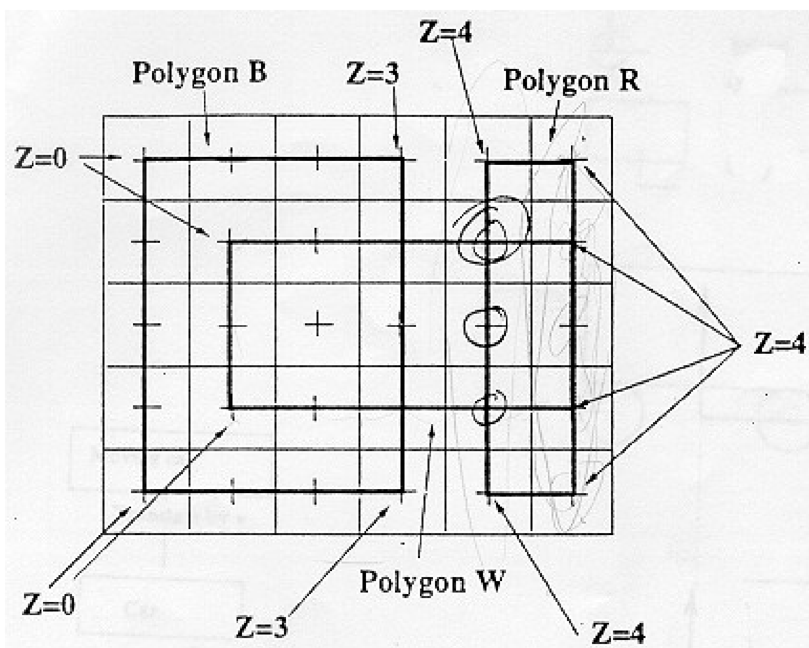

Figure 5: *Polygons for the* visibility question

Visibility and Z-buffers

- Figure 5 shows a 30 pixel frame buffer, which has a 2 bit z-buffer. The z-value used for a pixel is the z-value at the center point which is truncated to be either 0, 1, 2, or 3. (i.e. z73 becomes z = 3.) The center point is also used (as in class) to decide whether a pixel lies inside a polygon. All *polygon vertices lie on the center of their pixels*. The z-buffer algorithm here is eccentric: if a new pixel has the same z-value as an existing pixel, the two are averaged. In the figure polygon B is black, polygon W is white and polygon R is red. For each figure, mark the pixels that are pink (i.e. where W and R is averaged. (14)

- the "Painters algorithm" sorts polygons by the depth of their center of gravity, and draws the furthest polygon first. Sketch one example where this algorithm fails.

- Name one advantage and one disadvantage (other than difficulties with quantization!) of Z-buffers (2)

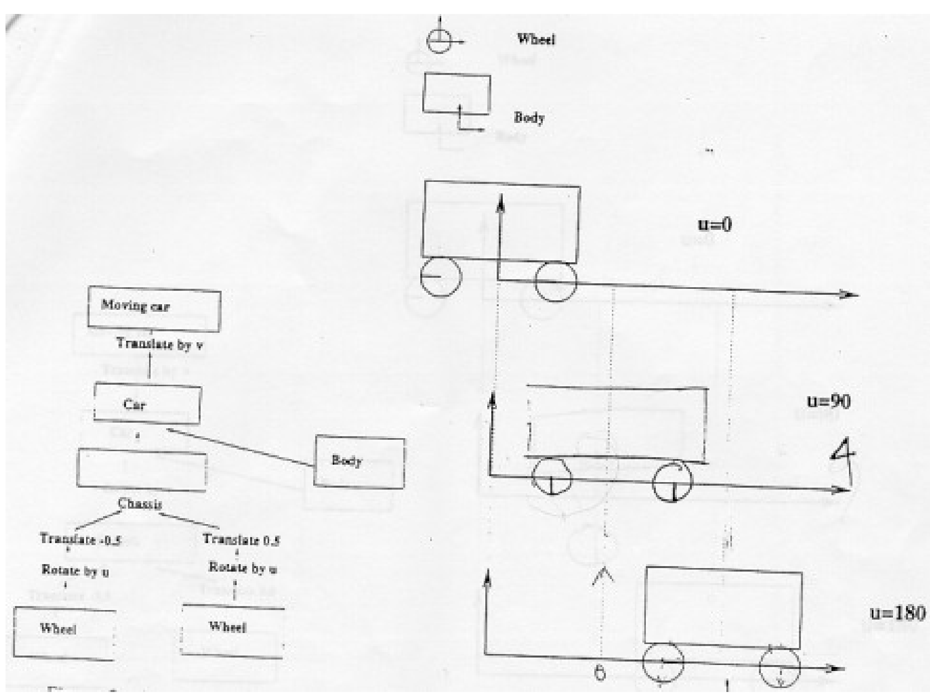

Figure 6: *A model hierarchy; the top figure shows the definitions for a wheel and a body. On the right are renderings for u=0, v=0 and u=180, v=2. Draw the result of rendering for u=90, v=1 in the space provided: u is in degrees*

Hierarchies

- Figure 6 shows a hierarchical model; fill in the details requested in the caption (4)

- Figure 7 shows a hierarchical model; fill in the details requested in the caption (9)

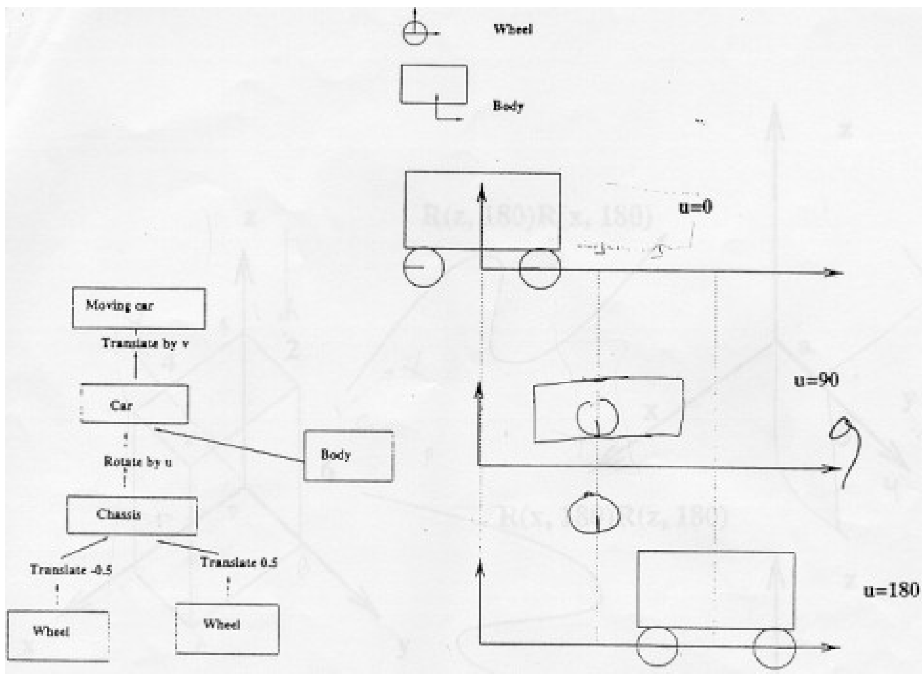

Figure 7: *A model hierarchy; the top figure shows the definitions for a wheel and a body. On the right are renderings for u=0, v=0 and u=180, v=2. Draw the result of rendering for u=90, v=1 in the space provided: u is in degrees*

Rotations

- In figure 8, fill in the appearance of the box as rotated by the amount given in the figures (6)
- In figure 9, fill in the appearance of the box as rotated by the amount given in the figures (6)
- What is curious about the case shown in figure 8? (2)

- Show that any rotation matrix in 3D is orthogonal - i.e. that $R^T R$ = *Identity*. (7)

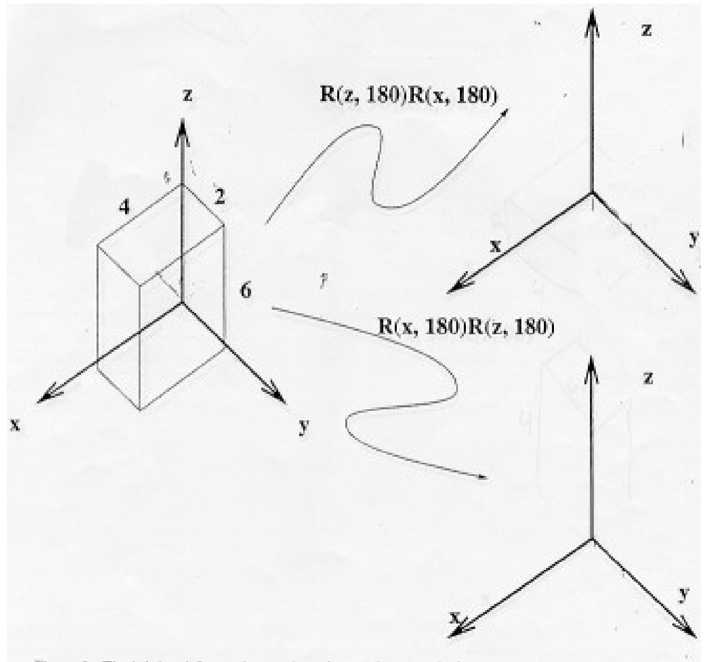

Figure 8: *The left hand figure shows a box; the numbers are the lengths of the side of the box. On the right hand are two sets of axes, for you to draw different rotations of the box. In this figure, R(x,180) means rotation about the x axis by 180 degrees. Fill in the two cases. You do not need to draw the box to scale, but you do need to fill in the lengths of the edges.*

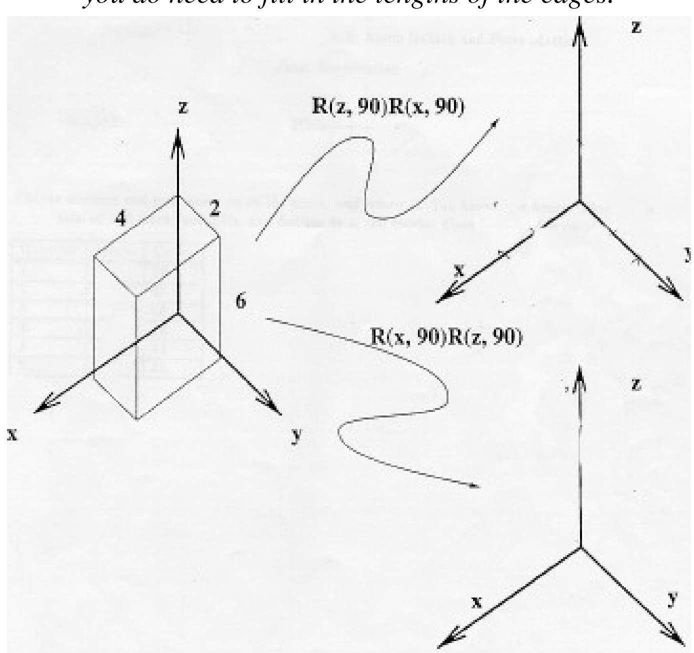

Figure 9: *The left hand figure shows a box; the numbers are the lengths of the side of the box. On the right hand are two sets of axes, for you to draw different rotations of the box. In this figure, R(x,90) means rotation about the x axis by 90 degrees. Fill in the two cases. You do not need to draw the box to scale, but you do need to fill in the lengths of the edges.*