G-IV Exercises

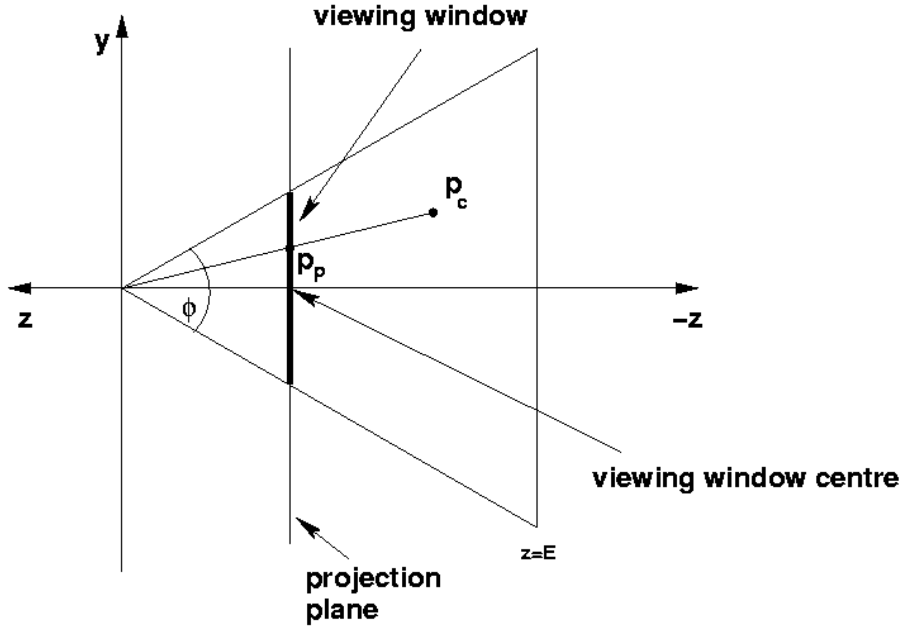
1. **Perspective Projection**

   The elements of the scene are given by vertices in camera coordinates where the camera is positioned at the origin looking down the z-axis (pointing in the direction of the negative z-axis). Furthermore, the centre of the viewing window in camera coordinates is $(0, 0, D)$.

   In order to compute the perspective projection of vertices in homogeneous camera coordinates in a normalised viewing volume the following parameters are given:

   - the angle $\phi \in (0, 2\pi)$ of the field of view with respect to the y axis,
   - the aspect ratio a: the ratio between the width and the height of the viewing window in the projection plane ($a =$ width/height),
   - the width w of the viewing window in the camera coordinate system (not in pixels, etc.),
   - a z-coordinate E indicating the maximum z-coordinate of elements inside the viewing volume.

   (a) Derive the matrix to compute the homogeneous coordinates $P_p = (x_p, y_p, z_p, w_p)$ of the projection of a vertex $P_c = (x_c, y_c, z_c, w_c)$ onto the viewing plane using the parameters ϕ, w, a.



   (b) To preserve the depth information for visible surface detection we map the perspective viewing volume onto a rectangular block rather than a plane, where the z axis represents the depth, and the x, y coordinates represent the projection onto the viewing plane.

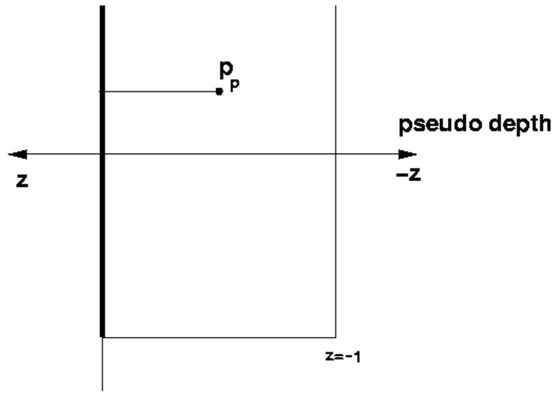
   Remark: One choice for the depth would be to use z_c/w_c. However, this approach fails to work well as it maps straight lines to curves in the rectangular block. So we cannot interpolate the depth of the points on the line linearly, which makes drawing lines more complicated. Drawing polygons has Similar problems.

   We choose to compute a pseudo-distance $d(z)$ for a Cartesian camera z-coordinate $z = z_c/w_c$ which preserves relative depths and causes lines to be mapped to lines. A good general form of $d(z)$ is

$$d(z) = A + \frac{B}{z}$$

   where we still have to determine A and B.

We wish to map the depth of a vertex on the viewing plane to the depth value $d(D) = 0$ and a vertex on the far end of the viewing volume to the depth value $d(E) = -1$. Using these conditions compute the parameters A and B. From the result adjust the homogeneous transformation matrix from (a) to preserve the depth information using the pseudo-distance as z-coordinate.



2. **Circle Clipping**

   Devise a method to clip a line segment from point p_1 to p_2 against a circular window with centre c and radius r in 2D. Outline all necessary equations.

3. **Shadows**

   (a) Extended light sources are non-point light sources which emit light from a surface rather than from a single point, you may e.g. consider a spherical light source. What major effect do extended light sources have on shadows?

   (b) Outline a method to compute shadows using an additional z-buffer for a scene with a single light source. Hint: a position is in the shadow of another polygon if there is a polygon between the pixel and the light source, i.e. the pixel is not visible from the light source position.

G-IV Exercise Solutions

1. **Perspective Projection**

(a)
- Let D be the z-coordinate of the viewing plane centre. Using similar triangles we get
$$\frac{y_c}{z_c} = \frac{y_p}{D}, \qquad \frac{x_c}{z_c} = \frac{x_p}{D}$$
- We get D from the angle ϕ and the viewing window height h:
$$\tan\left(\frac{\phi}{2}\right) = \frac{h}{2D}$$
- Using a and w we have
$$h = \frac{w}{a}$$
- So:
$$D = \frac{h}{2\tan\left(\frac{\phi}{2}\right)} = \frac{w}{2a\tan\left(\frac{\phi}{2}\right)}$$
$$\begin{aligned} y_p &= D\frac{y_c}{z_c}, & y'_p &= y_c \\ x_p &= D\frac{x_c}{z_c}, & x'_p &= x_c \\ z_p &= D, & z'_p &= z_c \\ w_p &= 1, & w'_p &= \frac{z_c}{D} \end{aligned}$$
- So the matrix for homogeneous coordinates is
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{2a\tan\left(\frac{\phi}{2}\right)}{w} & 0 \end{pmatrix}$$

(b) Conditions:
$$d(D) = A + \frac{B}{D} = 0$$
$$d(E) = A + \frac{B}{E} = -1$$
Solving the system:
$$-\frac{B}{D} + \frac{B}{E} = -1$$
$$\frac{B(D-E)}{DE} = -1$$
- Solution:
$$B = -\frac{DE}{D-E}$$
$$A = \frac{E}{D-E}$$
- So with (a) in homogeneous coordinates:
$$\begin{aligned} y_p &= D\frac{y_c}{z_c}, & y'_p &= D\frac{y_c}{w_c}, & y''_p &= Dy_c \\ x_p &= D\frac{x_c}{z_c}, & x'_p &= D\frac{x_c}{w_c}, & x''_p &= Dx_c \\ z_p &= A + \frac{Bw_c}{z_c}, & z'_p &= A\frac{z_c}{w_c} + B, & z''_p &= Az_c + Bw_c \\ w_p &= 1, & w'_p &= \frac{z_c}{w_c}, & w''_p &= z_c \end{aligned}$$

- So the matrix for homogeneous coordinates is

$$\begin{pmatrix} \frac{2a\tan\left(\frac{\phi}{2}\right)}{w} & 0 & 0 & 0 \\ 0 & \frac{2a\tan\left(\frac{\phi}{2}\right)}{w} & 0 & 0 \\ 0 & 0 & \frac{E}{D-E} & -\frac{DE}{D-E} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

2. **Circle Clipping**

- A point p is inside the circular window if $\|p-c\| \leq r$ or $(p-c)^t(p-c) \leq r^2$
- If both end-points of the line segments lie inside, draw the line.
- If one end-point lies inside and one outisde, line must be clipped:
  - Find intersection with circle by solving $\|p_1 + t(p_2 - p_1) - c\|^2 = r^2$ for $t \in [0, 1]$ (solution has to exist)
  - Draw line from $p_1 + t_1(p_2 - p_1)$ to the point inside.
- If two end-points lie outside, the line may have to be clipped:
  - Find intersections by solving $\|p_1 + t(p_2 - p_1) - c\|^2 = r^2$ for $t \in [0, 1]$ as above
  - If equation has two solutions t_1, t_2 in $[0, 1]$ draw line from $p_1 + t_1(p_2 - p_1)$ to $p_1 + t_2(p_2 - p_1)$. Otherwise reject the line.

3. **Shadows**

(a) Extended light sources cause soft shadows resulting into three regions to be consdiered: full shadow (umbra): the region where the extended light source is "invisible" and hence no direct light reaches this region; soft-shadow (penumbra): the region where the light source is partially visible and thus only a certain percentage of the full light source intensity reaches the region (depending on how much of the light source is visible from a position inside the soft-shadow area); full illumination: the region where the light source is fully visible and hence it is fully illuminated.

(b) To compute shadows using an additional z-buffer do a dual pass of the scene to compute shadow maps. In the first pass render the objects from the position of the light source and store only the depth information of each pixel in an additional z-buffer. Call the "viewing plane" with respect to the light source the light space (x, y coordinates for the position and a z coordinate for depth).

In the second pass, render the scene from the actual viewing position. But for each position to be drawn, first compute its position in the light space. If its depth in light space is larger than the depth recorded for its light space position in the first pass, there is a polygon between the position and the light source. Hence, the position is in the shadow of this polygon and the light computation should be done with the direct light from the light source.

Problems: approximation errors when computing the light space projection and comparing the depths may cause that some lit pixels are considered to be in the shadow and vice verse - using tolerances for the comparisons improves this. Also, while the z-buffer for the light sources does not have to be recomputed when we change the viewing position (without changing any object positions), we still have to do a lot of additional computations to compute the additional z-buffer values and compare the light space coordinates of the positions with these z-buffer values. Furthermore, with this method we cannot compute soft shadows. But there are various approaches towards approximating soft shadows in real-time, e.g. by computing and merging multiple shadow masks computed from multiple positions on an extended light source.