

Lecture 6: Projections

Adam Hawley

November 26, 2018

Recommended Reading:

- Prince, Chapter 14
- Nielsen 3.3
- Forsyth & Ponce Chapters 1 and 3

Contents

1	Viewing Systems	2
2	Perspective Projections	2
3	Parallel/Orthographic Projections	2
4	View Reference Coordinate System	2
4.1	Viewing Parameters	2
4.2	Viewplane Parameters	2
4.3	Window Parameters	2
4.4	Coordinate System in Different Projections	3
4.5	View Volumes for Graphics	3
4.6	Focal Length in Perspective Projection	3
4.7	Offset	3
4.8	Intrinsic Parameters	4
4.9	Extrinsic Parameters	4
4.10	3D Transformations	4
4.11	Camera Calibration	5

1 Viewing Systems

In real life we pick up objects, position them and then view them. In computer graphics objects are positioned in a fixed frame. The viewer moves to the appropriate position in order to achieve the desired view.

2 Perspective Projections

We assume that there exists a *Centre of Projection* (COP). Objects' distances from the **COP** cause the objects to appear larger or smaller on the intersecting viewplane.

Lines that are not parallel to the viewplane converge to a vanishing point. The *Principal Vanishing Point* exists for lines that are parallel to the principal axis. This can be clearly seen in the perspective projection of a cube, edges which are parallel to one another appear to converge.

3 Parallel/Orthographic Projections

Here the COP is **always** at infinity. This means that the viewplane is aligned with the axes and the *Direction of Projection* (DOP)

4 View Reference Coordinate System

4.1 Viewing Parameters

- Location of viewplane
- Window within viewplane
- Projection type
- Projection reference point (**prp**)

4.2 Viewplane Parameters

- View Reference Point (**vrp**)
- Viewplane Normal (**vpn**)
- View Up Vector (**vup**)

4.3 Window Parameters

- Width — u_{min} and u_{max}
- Height — v_{min} and v_{max}

4.4 Coordinate System in Different Projections

For **Perspective Projections** there is only one property:

- **prp** becomes **cop**

In **Parallel Projections** there are three properties:

- **cw** — The center of window
- **dop** — Direction of projection (**cw** – **prp**)
- The projection is *Orthographic* if **dop** and **vpn** are parallel

4.5 View Volumes for Graphics

The view volume for perspective projections is a truncated pyramid (frustum) while the view volume for parallel perspective projections is a cuboid. The view volumes define a specific region of the scene which is viewed in the current image. The view volume is defined by the clipping planes, the projection rays used by the projection system and by the image rectangle. They are sometimes used in graphics for visualisation such as creating smoke or mist defined only in a specific region of the scene.

4.6 Focal Length in Perspective Projection

Focal length (ϕ) is used for two things:

- Modelling the effect of the distance to the focal plane
- Modelling the density of the receptors (e.g. CCD pixels)

$$x = \frac{\phi u}{\omega} \quad y = \frac{\phi v}{\omega}$$

In practice, the receptors may not be square. In this case just use different focal length parameters for x and y dimensions:

$$x = \frac{\phi_x u}{\omega} \quad y = \frac{\phi_y v}{\omega}$$

4.7 Offset

Currently we assume that (0,0) is where the principal ray strikes the image plane however if this is not the case (and a different coordinate system is used e.g (0,0) in the top left corner) then one can use an offset (δ) or a skew parameter (γ):

$$x = \frac{\phi_x u + \gamma v}{\omega} + \delta_x \quad y = \frac{\phi_y v}{\omega} + \delta_y$$

4.8 Intrinsic Parameters

The aforementioned focal lengths, skew and offset form the set of **intrinsic parameters**:

$$\phi_x, \phi_y, \gamma, \delta_x, \delta_y$$

It is convenient to store these in an **intrinsic matrix**:

$$\begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix}$$

4.9 Extrinsic Parameters

If the camera itself is not centered at the origin of the world coordinate system which is often the case. If more than one camera is going to be used then it is very common for an arbitrary coordinates system to be used. To deal with this we express world coordinate points in the local coordinate system of the camera before they are passed to the projection model.

$$\begin{bmatrix} u' \\ v' \\ \omega' \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

or

$$P' = \Omega P + T$$

4.10 3D Transformations

Ω above is the product of 3D transformations where:

$$\Omega = R_x R_y R_z$$

The same principles apply but now to 4D-homogeneous space. Rotation is still defined with respect to a specific axis:

Axis	Positive Direction
x	$y \rightarrow z$
y	$z \rightarrow x$
z	$x \rightarrow y$

$$\begin{aligned}
R_z(\theta) &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
R_x(\theta) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
R_y(\theta) &= \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

4.11 Camera Calibration

How does one point in the scene relate to the observed point in the image. A 3D point of coordinates (X, Y, Z) from the scene is related to a 2D point (u, v) from the image by the equation:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x & 0 \\ 0 & \phi_y & \delta_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \tau_x \\ \omega_{21} & \omega_{22} & \omega_{23} & \tau_y \\ \omega_{31} & \omega_{32} & \omega_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$