Aalto-yliopisto
Aalto-universitetet
Aalto University

CS-E4850
Computer Vision

Homework 2
Pinhole Camera

Aitor Urruticoechea Puig
aitor.urruticoecheapuig@aalto.fi
Student N°101444219
September 2024

Contents

Exercise 1: Pinhole Camera

The perspective projection equations for a pinhole camera are

$$x_p = f\frac{x_c}{z_c}, \quad y_p = f\frac{y_c}{z_c}, \tag{1}$$

where $\mathrm{x_p} = [\mathrm{x_p}, \mathrm{y_p}]^\mathsf{T}$ *are the projected coordinates on the image plane,* $\mathrm{x_c} = [\mathrm{x_c}, \mathrm{y_c}, \mathrm{z_c}]^\mathsf{T}$ *is the imaged point in the camera coordinate frame, and* f *is the focal length. Give a geometric justification for the perspective projection equations.*
(Hint: Use similar triangles and remember that the image plane is located at a distance f *from the projection centre and is perpendicular to the optical axis, i.e., the* z*-axis of the camera coordinate frame.)*

Understanding that the image plane is at a distance f from the pinhole; and that the 3D points $\mathrm{x_c}$, $\mathrm{y_c}$, and $\mathrm{z_c}$ are projected into the 2D image plane; one can use similar triangles. Since the z-axis in 3D is perpendicular to the focal distance f, with similar triangles it can be seen that the ratios of the dimensions in the image plane with respect to the ones in the 3D space must remain constant:

$$\begin{cases} \frac{\mathrm{x_p}}{\mathrm{f}} & = \frac{\mathrm{z_c}}{\mathrm{y_c}} \\ \frac{\mathrm{y_p}}{\mathrm{f}} & = \frac{\mathrm{z_c}}{\mathrm{y_c}} \end{cases} \tag{2}$$

Which can easily be rewritten to get equation 1.

Exercise 2: Pixel coordinate frame

The image coordinates x_p and y_p given by the perspective projection equations 1 above are not in pixel units. The x_p and y_p coordinates have the same unit as distance f (typically millimeters), and the origin of the coordinate frame is the principal point (the point where the optical axis pierces the image plane). Now, give a formula which transforms the point x_p to its pixel coordinates $p = [u, v]^T$ when the number of pixels per unit distance in u and v directions are m_u and m_v, respectively, and the pixel coordinates of the principal point are (u_0, v_0), for the cases:

2.1 u and v axis are parallel to x and y axis, respectively.

To convert between millimiters and number of pixels, one will have to make use of the given m_u and m_v magnitudes which allow for the conversion. However, one must also take into account the fact that the the pixel coordinates of the principal point are not necessarily $(0, 0)$. Thus:

$$\begin{cases} u = m_u x_p + u_0 \\ v = m_v y_p + v_0 \end{cases} \tag{3}$$

where u_0 and v_0 are the pixel coordinates of the principal point.

2.2 u axis is parallel to x axis and the angle between u and v axis is θ.

Now the pixel axes are defined with an unknow angle θ between them. While the transformation between x and u can remain the same thanks to them stil being parallel; the relation between v and y is not trivial. Instead, one must introduce a rotation, just as it was seen in the previous batch of homework exercises. This results in:

$$\begin{cases} u = m_u x_p + u_0 \\ v = m_v \left[y_p \cos(\theta) - x_p \sin(\theta)\right] + v_0 \end{cases} \tag{4}$$

Exercise 3: Intrinsic camera calibration matrix

Use homogeneous coordinates to represent case (2.1) above with a matrix $K_{3\times3}$, also known as the camera's intrinsic calibration matrix, so that $\tilde{p} = Kx_c$. Here, $\tilde{p}$ is p in homogeneous coordinates.

As it was seen in the first batch of homework exercises, homogeneous coordiantes are obtained by adding a new dimension wich gets a unitary value. Thus, $\tilde{p}$ can be defined as:

$$\tilde{p} = [u, v, 1]^T \tag{5}$$

Departing from equation 3, that defines the translation from physical distance to pixel distance, it can be seen that for $\tilde{p} = Kx_c$ to be true, where $x_c = [x_p, y_p, 1]^T$, K needs to be defined as:

$$K = \begin{bmatrix} m_u & 0 & 0 \\ 0 & m_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Exercise 4: Camera projection matrix

Imaged points are often expressed in an arbitrary frame of reference called the world coordinate frame. The mapping from the world frame to the camera coordinate frame is a rigid transformation consisting of a 3D rotation R *and translation* t*:*

$$x_c = Rx_w + t \tag{7}$$

Use homogeneous coordinates and the result of the exercise 3 above to write down the 3×4 *camera projection matrix* P *that projects a point from world coordinates* x_w *to pixel coordinates. That is, represent* P *as a function of the internal camera parameters* K *and the external camera parameters* R, t.

Departing from the given equation 7, it is possible to once again use the tools defined in Homework 1 to represent this rotation and translation in homogeneous coordinates:

$$\begin{bmatrix} x_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ 1 \end{bmatrix} \tag{8}$$

where R is the rotation matrix and t is the translation vector. In homogeneous coordinates, "adding up" transformations is trivial, as one needs only multiply the matrices. This means the complete camera projection matrix P can be obtained by adding the K matrix previously defined in equation 6 is done as:

$$P = K\,[Rt] \tag{9}$$

which satisfies the defined goal:

$$\tilde{p} = Px_w \tag{10}$$

Exercise 5: Rotation matrix

A rigid coordinate transformation can be represented with a rotation matrix R and a translation vector t, which transform a point x to $x' = Rx + t$. Now, let the 3×3 matrix R be a 3-D rotation matrix, which rotates a vector x by the angle θ about the axis u (a unit vector). According to the Rodrigues formula, it holds that

$$Rx = \cos\theta x + \sin\theta u \times x + (1 - \cos\theta)(u \cdot x)u. \tag{11}$$

(Hint: Write down the cross product $u \times x$ and the scalar product $u \cdot x$ in terms of the elements of vectors u and x. Use the notations $u = (u_1, u_2, u_3)^T$ and $x = (x_1, x_2, x_3)^T$. You may also consult literature or public sources like Wikipedia.)

5.1 Give a geometric justification (i.e., derivation) for the Rodrigues formula.

The geometric justification for Rodrigues formula can be easily understood if one is to break down the rotation into easy-to-digest components.

- **Rotation in the plane perpendicular to** u: the component of x which is perpendicular to u gets rotated by θ. This results in the $\cos(\theta)x$ term.

- **Cross product term** $u \times x$: which represents the direction of the vector after rotating by $\sin(\theta)$ around the axis.
- **Parallel component** x: the component of x which is parallel to u should remain unchanged, yet needs to be scaled to account for the θ rotation. This forces the apparition of the term $(1 - \cos(\theta))$; which ultimately gets just multiplied to the projection of x onto u obtained by $(u \cdot x)$.

5.2 Derive the expressions for the elements of R as a function of θ and the elements of u.

Starting by writing the cross-product term in matrix form as $u \times x = Ux$, where U takes the form:

$$U = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \tag{12}$$

This, along with the decomposition of the dot product with $u \cdot x = u_1x_1 + u_2x_2 + u_3x_3$, equation 11 can be rewritten as

$$R = I + \sin(\theta)U + (1 - \cos(\theta))\, U^2 \tag{13}$$

with I being the identity matrix. By substituting U by its definition (equation 12) and calculating its squared from U^2, one can finally get:

$$R = I + \sin(\theta) \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} + (1 - \cos(\theta)) \begin{bmatrix} u_1^2 & u_1u_2 & u_1u_3 \\ u_1u_2 & u_2^2 & u_2u_3 \\ u_1u_3u_2u_3 & u_3^2 & \end{bmatrix} \tag{14}$$