AEROSP 584: Navigation and Guidance: From Perception to Control **F22**

Lecture 16: Oct 25

Instructor: Vasileios Tzoumas of U-M *Author: Luca Carlone of MIT*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Author(s) and Instructor(s).*

This lecture derives the pinhole camera model, following (with a slightly heavier but more evocative notation) Chapter 3 in [1].

1 Perspective Projection of a 3D Point: the Pinhole Camera

Consider a 3D point $\boldsymbol{p}^{\mathrm{c}}$ as shown in Fig. 1.

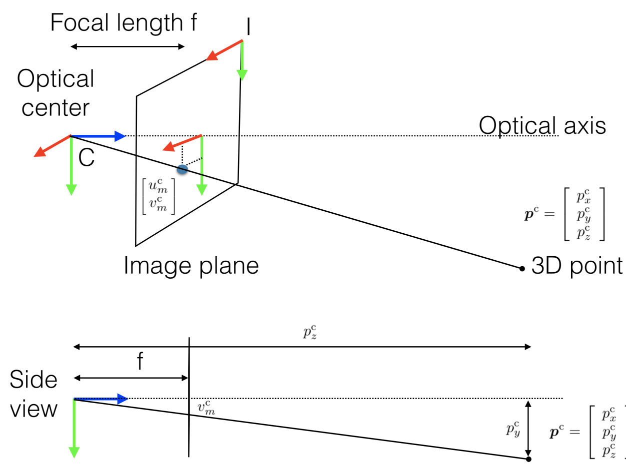

Figure 1: Pinhole Model.

The position $(u^{\mathrm{c}}_m, v^{\mathrm{c}}_m)$ of the projection of $\boldsymbol{p}^{\mathrm{c}}$ on the camera plane is:

$$u^{\mathrm{c}}_m = f\frac{p^{\mathrm{c}}_x}{p^{\mathrm{c}}_z} \qquad\qquad v^{\mathrm{c}}_m = f\frac{p^{\mathrm{c}}_y}{p^{\mathrm{c}}_z} \tag{1}$$

where f is the *focal length* (in meters), and the subscript m denotes that $(u^{\mathrm{c}}_m, v^{\mathrm{c}}_m)$ are still expressed in meters (later on we convert them into pixels).

The relations in (1) can be written, by moving p^{c}_z to the left-hand-side and using a matrix notation as:

$$p^{\mathrm{c}}_z \begin{bmatrix} u^{\mathrm{c}}_m \\ v^{\mathrm{c}}_m \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p^{\mathrm{c}}_x \\ p^{\mathrm{c}}_y \\ p^{\mathrm{c}}_z \end{bmatrix} \tag{2}$$

where the last equation simply states $p_z^{\rm c} = p_z^{\rm c}$. Rewriting $\boldsymbol{p}^{\rm c}$ in homogeneous coordinates:

$$\boldsymbol{p}^{\rm c} = \begin{bmatrix} p_x^{\rm c} \\ p_y^{\rm c} \\ p_z^{\rm c} \end{bmatrix} \qquad \xrightarrow{\text{homogeneous coordinates}} \qquad \tilde{\boldsymbol{p}}^{\rm c} = \begin{bmatrix} p_x^{\rm c} \\ p_y^{\rm c} \\ p_z^{\rm c} \\ 1 \end{bmatrix} \tag{3}$$

which allows writing (2) as:

$$p_z^{\rm c} \begin{bmatrix} u_m^{\rm c} \\ v_m^{\rm c} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \tilde{\boldsymbol{p}}^{\rm c} \tag{4}$$

which represents the canonical projection of a point $\tilde{\boldsymbol{p}}^{\rm c}$ whose coordinates are given in the camera frame.

1.1 Conversion to Pixels

We already noticed that the quantities $(u_m^{\rm c}, v_m^{\rm c})$ are expressed in meters. Moreover, during the first lecture, we noticed that the image coordinate frame typically has origin in the top-left corner of the image (see frame "I" in Fig. 1).

Therefore, we can convert $(u_m^{\rm c}, v_m^{\rm c})$ to pixels and change the reference frame as follows:

$$u^{\rm I} = s_x u_m^{\rm c} + o_x \qquad\qquad v^{\rm I} = s_y v_m^{\rm c} + o_y \tag{5}$$

Which using a compact matrix notation becomes:

$$\begin{bmatrix} u^{\rm I} \\ v^{\rm I} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_m^{\rm c} \\ v_m^{\rm c} \\ 1 \end{bmatrix} \tag{6}$$

If the pixels are skewed (non-rectangular), the expression can be generalized to:

$$\begin{bmatrix} u^{\rm I} \\ v^{\rm I} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_m^{\rm c} \\ v_m^{\rm c} \\ 1 \end{bmatrix} \tag{7}$$

Substituting (7) into (4):

$$p_z^{\rm c} \begin{bmatrix} u^{\rm I} \\ v^{\rm I} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \tilde{\boldsymbol{p}}^{\rm c} \tag{8}$$

Multiplying the first two matrices in the right-hand side:

$$p_z^{\rm c} \begin{bmatrix} u^{\rm I} \\ v^{\rm I} \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} s_x\, f & s_\theta\, f & o_x \\ 0 & s_y\, f & o_y \\ 0 & 0 & 1 \end{bmatrix}}^{\boldsymbol{K}} \overbrace{\begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix}}^{\boldsymbol{\Pi}_0} \tilde{\boldsymbol{p}}^{\rm c} \tag{9}$$

Terminology:

- $[o_x\ o_y]$ is called the *principal point*

- $s_x\, f$ is the focal length in horizontal pixels (s_x number of horizontal pixels per meter)
- $s_y\, f$ is the focal length in vertical pixels (s_y number of vertical pixels per meter)
- s_x/s_y is called the *aspect ratio*
- $s_\theta\, f$ is called the *skew* of the pixel
- $\boldsymbol{K}$ is called the *intrinsic (calibration) matrix*
- $\boldsymbol{\Pi}_0$ is called the *canonical projection*

1.2 Points Coordinates in the World Frame

When the point coordinates are given with respect to an external world frame w, i.e., $\boldsymbol{p}^{\mathrm{w}}$, it is straightforward to adapt (9) as:

$$p_z^{\mathrm{c}} \begin{bmatrix} u^{\mathrm{I}} \\ v^{\mathrm{I}} \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} s_x\, f & s_\theta\, f & o_x \\ 0 & s_y\, f & o_y \\ 0 & 0 & 1 \end{bmatrix}}^{\boldsymbol{K}} \overbrace{\begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix}}^{\boldsymbol{\Pi}_0} \boldsymbol{T}_{\mathrm{w}}^{\mathrm{c}}\ \tilde{\boldsymbol{p}}^{\mathrm{w}} \tag{10}$$

Recalling the structure of the pose matrix $\boldsymbol{T}_{\mathrm{w}}^{\mathrm{c}}$ and multiplying by the canonical projection $\boldsymbol{\Pi}_0$:

$$p_z^{\mathrm{c}} \begin{bmatrix} u^{\mathrm{I}} \\ v^{\mathrm{I}} \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} s_x\, f & s_\theta\, f & o_x \\ 0 & s_y\, f & o_y \\ 0 & 0 & 1 \end{bmatrix}}^{\text{intrinsic calibration}} \overbrace{\begin{bmatrix} \boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}} & \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}} \end{bmatrix}}^{\text{extrinsic calibration}} \quad \tilde{\boldsymbol{p}}^{\mathrm{w}} = \boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}} \tag{11}$$

Terminology:

- $\begin{bmatrix} \boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}} & \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}} \end{bmatrix}$ is called the *extrinsic (calibration) matrix*
- $\boldsymbol{\Pi}$ is called the *projection matrix*

In this course, we typically assume that the intrinsic matrix $\boldsymbol{K}$ is known from prior calibration.

1.3 Projection in Practice

In the following lectures, we will consider different uses of projective geometry. Here are two examples:

- computing pixel projection of a 3D point: if the pose of the camera $\boldsymbol{T}_{\mathrm{w}}^{\mathrm{c}}$ is known then we can project a 3D point $\tilde{\boldsymbol{p}}^{\mathrm{w}}$ as $\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}}$ and then retrieve the pixel projection of $\tilde{\boldsymbol{p}}^{\mathrm{w}}$ as:

$$u^{\mathrm{I}} = \frac{[\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}}]_1}{[\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}}]_3} \qquad\qquad v^{\mathrm{I}} = \frac{[\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}}]_2}{[\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}}]_3} \tag{12}$$

  where $[\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}}]_i$ denotes the i-th entry in $\boldsymbol{\Pi}\tilde{\boldsymbol{p}}^{\mathrm{w}} \in \mathbb{R}^3$.

- if the pixel projection $(u^{\mathrm{I}}, v^{\mathrm{I}})$ and the corresponding 3D point are given, we can use the expression (11) to infer the pose of the camera (this requires multiple 3D points)

1.4 Distortion

Cameras with wide field of view are often subject to radial distortion, which can be modeled as:

$$u^{\mathrm{c}} = (1 + a_1 r^2 + a_2 r^4) u^{\mathrm{c}}_{distort} \qquad v^{\mathrm{c}} = (1 + a_1 r^2 + a_2 r^4) v^{\mathrm{c}}_{distort} \tag{13}$$

where $r^2 = (u^{\mathrm{c}}_{distort})^2 + (v^{\mathrm{c}}_{distort})^2$ and a_i are the so-called *distortion coefficients*. Note that the equation above is expressed in the camera frame (origin at the center of the image, rather than on the top left) since the distortion increases with the distance from the center of the image. It is possible to rewrite the model using the image frame as:

$$u^{\mathrm{I}} = (1 + a_1 r^2 + a_2 r^4)(u^{\mathrm{I}}_{distort} - o_x) + o_x \qquad v^{\mathrm{I}} = (1 + a_1 r^2 + a_2 r^4)(v^{\mathrm{I}}_{distort} - o_y) + o_y \tag{14}$$

and $r^2 = (u^{\mathrm{I}}_{distort} - o_x)^2 + (v^{\mathrm{I}}_{distort} - o_y)^2$. We typically assume the distortion coefficients to be known from previous calibration, hence we can always compensate for the calibration using the model (14).

References

[1] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision.* Springer, 2004.