8
Camera Models and Calibration

The previous chapter began an introduction to the problem of robotic perception, which consists of tasks related to sensing and understanding the robot's own movements as well as the environment in which it operates[1]. This chapter continues that discussion by diving more deeply into one of the most powerful and challenging tools in robotic perception: computer vision. In particular, this chapter will focus on some of the fundamental mathematical tools for calibrating cameras and processing their images to extract some useful information about the scene[2,3].

Camera Models and Calibration

As was discussed in the previous chapter, cameras provide a crucial sensing modality in the context of robotics. This is generally due to the fact that images inherently contain an enormous amount of information about the environment. However, while images do contain a lot of information, extracting the information that is relevant to the robot is quite challenging. One of the most basic tasks related to image processing is determining how a particular point in the scene maps to a point in the camera image, which is sometimes referred to as *perspective projection*. Last chapter, the *pinhole camera model* and the *thin lens model* were presented, and in this chapter the pinhole camera model is leveraged to further explore perspective projection[4].

8.1 Perspective Projection

The pinhole camera model, shown graphically in Figure 8.1, can be used to mathematically define relationships between points P in the scene and points p on the image plane. Notice that any point P in the scene can represented in two ways: in camera frame coordinates (denoted as P_C) or in world frame coordinates (denoted as P_W). The overall objective of this section is to find derive a mathematical model that can be used to map a point P_W expressed in world frame coordinates to a point p on the image plane. To accomplish this two transformations are combined together, namely a transformation of P from

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT Press, 2011

[2] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2011

[3] R. Hartley and A. Zisserman. "Camera Models". In: *Multiple View Geometry in Computer Vision*. Academic Press, 2002

[4] All results also hold under the thin lens model, assuming the camera is focused at ∞.

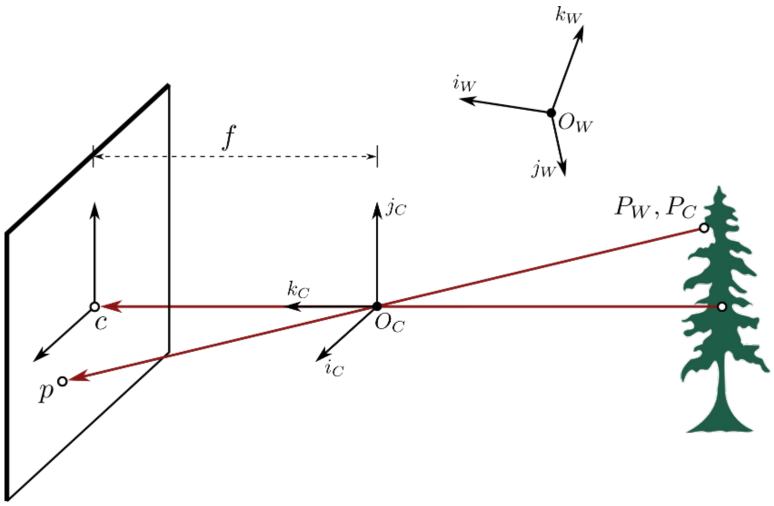

Figure 8.1: Graphical representation of the pinhole camera model. In this model the point O_C is the camera center, c is the image center, and f is the focal length of the camera. It is assumed that all light rays from point P in the scene pass through point O_C and are captured on the image plane at point p.

world frame coordinates to camera frame coordinates (P_W to P_C) and a transformation from camera coordinates to image coordinates (P_C to p).

8.1.1 *Mapping Camera Frame Coordinates to Image Coordinates ($P_C \rightarrow p$)*

The first step considered is the mapping from a point in the scene expressed in camera frame coordinates, P_C, to the corresponding point on the image plane, p, using the pinhole camera model. Recall from the previous chapter the pinhole camera equations:

$$x = f\frac{X_C}{Z_C}, \quad y = f\frac{Y_C}{Z_C}, \tag{8.1}$$

where $P_C = (X_C, Y_C, Z_C)$, $p = (x, y)$, and f is the focal length of the pinhole camera[5].

Note that the quantities x and y are coordinates in the *camera frame*, but it is often desirable to express the point p in terms of *pixel coordinates*. However, pixel coordinates are generally defined with respect to a reference frame in the lower corner of the image plane (to avoid negative coordinates). This new reference frame is shown in Figure 8.2, where the image center c is defined in this new reference frame with coordinates $(\tilde{x}_0, \tilde{y}_0)$, where $\tilde{(\cdot)}$ is the notation used to denote a coordinate with respect to this new reference frame. In this new reference frame, the point P_C gets mapped to the coordinates $(\tilde{x}, \tilde{y})$ by:

$$\tilde{x} = f\frac{X_C}{Z_C} + \tilde{x}_0, \quad \tilde{y} = f\frac{Y_C}{Z_C} + \tilde{y}_0. \tag{8.2}$$

Finally, these new coordinates can be mapped to pixel coordinates if the number of pixels per unit distance are known. In particular, the point P_C is mapped to pixel coordinates (u, v) by:

$$u = \alpha\frac{X_C}{Z_C} + u_0, \quad v = \beta\frac{Y_C}{Z_C} + v_0, \tag{8.3}$$

[5] The z term of p is generally not included simply because $z = f$ is a fixed value.

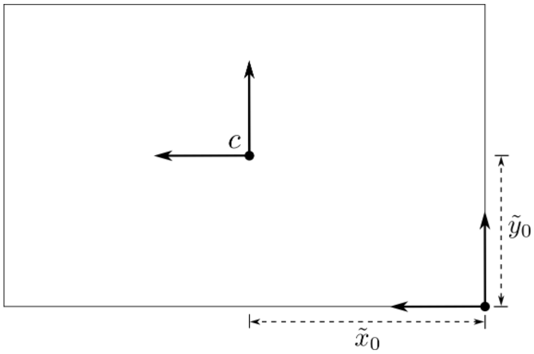

Figure 8.2: A new reference frame with coordinates denoted by $\tilde{(\cdot)}$ is defined with its origin in the lower corner of the image plane. The image center coordinates in this new frame are denoted $(\tilde{x}_0, \tilde{y}_0)$.

where $\alpha = k_x f$, $u_0 = k_x \tilde{x}_0$, $\beta = k_y f$, $v_0 = k_y \tilde{y}_0$, and k_x and k_y are the number of pixels per unit distance in image coordinates.

Homogeneous Coordinates: Note that the transformation from the point P_C in camera frame coordinates to p in pixel coordinates given by (8.3) is not linear. However, this transformation can be represented as a linear mapping[6] through an additional change of coordinates. In particular, the points P_C and p will be expressed in *homogeneous coordinates*.

For a 2D point (x_1, x_2) or a 3D point (x_1, x_2, x_3) in Euclidean space, the point can be represented in homogeneous coordinates by the transformation:

$$(x_1,\, x_2) \implies (\alpha x_1,\, \alpha x_2,\, \alpha), \quad \text{and} \quad (x_1,\, x_2,\, x_3) \implies (\alpha x_1,\, \alpha x_2,\, \alpha x_3,\, \alpha), \tag{8.4}$$

for any $\alpha \neq 0$. These new coordinates are called homogeneous coordinates because the scaling factor α can be chosen arbitrarily as long as $\alpha \neq 0$. A set of homogeneous coordinates can then be transformed back by:

$$(y_1,\, y_2,\, y_3) \implies \left(\frac{y_1}{y_3},\, \frac{y_2}{y_3}\right), \quad \text{and} \quad (y_1,\, y_2,\, y_3,\, y_4) \implies \left(\frac{y_1}{y_4},\, \frac{y_2}{y_4},\, \frac{y_3}{y_4}\right). \tag{8.5}$$

To denote when a point is described in homogeneous coordinates the superscript h will be used. For example, the point $P_C = (X_C, Y_C, Z_C)$ in camera frame coordinates can be expressed by:

$$P_C^h = (X_C, Y_C, Z_C, 1),$$

by choosing $\alpha = 1$, and the point $p = (u, v)$ in pixel coordinates can be expressed in homogeneous coordinates by:

$$p^h = (Z_C u,\, Z_C v,\, Z_C) = (\alpha X_C + u_0 Z_C,\, \beta Y_C + v_0 Z_C),$$

by choosing $\alpha = Z_C$ and substituting the expressions (8.3). With the expression of these points in homogeneous coordinates it can be seen that their relationship is transformed from the nonlinear relationship (8.3) to the *linear* relationship:

$$\begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha X_c + u_0 Z_c \\ \beta Y_c + v_0 Z_c \\ Z_c \end{pmatrix}. \tag{8.6}$$

[6] Expressing the perspective projection as a linear map will simplify the mathematics later on.

Often in practice a skewness parameter γ is also added (which generally ends up being close to 0), and this relationship can be written in the more compact form:

$$\begin{bmatrix} K & 0_{3\times 1} \end{bmatrix} P_C^h = p^h, \quad K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{8.7}$$

The matrix K defined in (8.7) is sometimes referred to as the *camera matrix* or *matrix of intrinsic parameters*. It is referred to in this way because it contains the five parameters that define the fundamental characteristics of the camera (from the perspective of the pinhole camera model). While these parameters may be specified by the camera manufacturer, they are often extracted by performing a camera calibration.

8.1.2 Mapping World Coordinates to Camera Coordinates ($P_W \rightarrow P_C$)

Recall from Figure 8.1 that a point P in the scene can either be expressed in terms of camera frame coordinates P_C or world frame coordinates P_W. While the previous section discussed the use of the pinhole model to map P_C coordinates to pixel coordinates p, this section will discuss the mapping between the camera and world frame coordinates of the point P (see Figure 8.3).

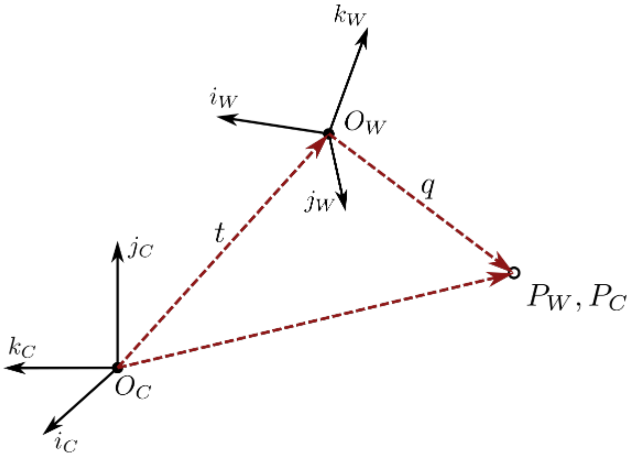

Figure 8.3: A depiction of the point P expressed either in camera coordinates, P_C, or in world frame coordinates, P_W. The world frame origin is denoted by O_W and the camera frame origin is denoted by O_C.

From Figure 8.3 it can be seen that P_C can be written as:

$$P_C = t + q, \tag{8.8}$$

where t is the vector from O_C to O_W expressed in camera frame coordinates and q is the vector from O_W to P expressed in camera frame coordinates. However, the vector q is in fact the same vector as P_W, just expressed in different coordinates (i.e. with respect to a different frame). The coordinates can be related by a rotation:

$$q = RP_W, \tag{8.9}$$

where R is the rotation matrix relating the camera frame to world frame and is defined as:

$$R = \begin{bmatrix} i_w \cdot i & j_w \cdot i & k_w \cdot i \\ i_w \cdot j & j_w \cdot j & k_w \cdot j \\ i_w \cdot k & j_w \cdot k & k_w \cdot k \end{bmatrix}, \tag{8.10}$$

where i, j, and k are the unit vectors that define the camera frame and i_w, j_w, and k_w are the unit vectors that define the world frame. To summarize, the point P_W can be mapped to camera frame coordinates P_C as:

$$P_C = t + RP_W, \tag{8.11}$$

where t is the vector in camera frame coordinates from O_C to O_W and R is the rotation matrix defined in (8.10). Similar to the previous section, these expressions can also be equivalently expressed for the case where the points P_W and P_C are expressed in homogeneous coordinates:

$$\begin{pmatrix} P_C \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{pmatrix} P_W \\ 1 \end{pmatrix}. \tag{8.12}$$

8.1.3 Mapping World Frame Coordinates to Image Coordinates ($P_W \rightarrow p$)

The original objective of perspective projection was to find a way to mathematically relate the position of a point P in world frame coordinates (denoted P_W) to the corresponding pixel coordinates p on the image plane. With the relationship (8.12) developed for mapping P_W to the camera frame coordinates P_C, and the relationship (8.7) for mapping P_C to pixel coordinates p, the direct mapping from P_W to p can now be defined. In particular, simply combining the two transformation together yields:

$$p^h = \begin{bmatrix} K & 0_{3\times 1} \end{bmatrix} \begin{bmatrix} R & t \\ 0_{1\times 3} & 1 \end{bmatrix} P_W^h,$$

which can then be simplified to:

$$p^h = K \begin{bmatrix} R & t \end{bmatrix} P_W^h. \tag{8.13}$$

In (8.13), P_W^h is the homogeneous coordinate representation of P_W and p^h is the homogeneous coordinate representation of p. Additionally, recall that the matrix $K \in \mathbb{R}^{3\times 3}$ is the matrix of intrinsic camera parameters, and the matrix $[R \quad t] \in \mathbb{R}^{3\times 4}$ contains *extrinsic* parameters (i.e. that describe the camera's position and orientation relative the points in the scene). Note that the total number of degrees of freedom is 11, where 5 are from the intrinsic parameters that define K, 3 are from the rotation matrix R, and 3 are from the position vector t.

8.2 Camera Calibration: Direct Linear Method

Before the expression (8.13) can be used in practice, the camera's intrinsic and extrinsic parameters need to be determined (i.e. K, R, and t). One approach is to use the direct linear transformation method for camera calibration, which requires a set of known correspondences $p_i \leftrightarrow P_{W,i}$ for $i = 1, \dots, n$.

8.2.1 Direct Linear Calibration: Step 1

First, each corresponding pair of points $p_i = (u_i, v_i)$ and $P_{W,i} = (X_{W,i}, Y_{W,i}, Z_{W,i})$ is written in homogeneous coordinates and the expression (8.13) is used to write:

$$p_i^h = MP_{W,i}^h, \quad i = 1, \dots n \tag{8.14}$$

where $M = K[R \;\; t]$ is referred to as the *homography*. The first step of the camera calibration process is to use the n correspondences to compute the homography M, and then later the intrinsic and extrinsic parameters can be extracted from M. To determine M, a useful first step is to rewrite M in terms of its rows:

$$M = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}, \tag{8.15}$$

where $m_i \in \mathbb{R}^{1\times 4}$ is the i-th row of M. By considering the rows of M individually, the relationship (8.14) can be written as:

$$\begin{bmatrix} \alpha u_i \\ \alpha v_i \\ \alpha \end{bmatrix} = \begin{bmatrix} m_1 \cdot P_{W,i}^h \\ m_2 \cdot P_{W,i}^h \\ m_3 \cdot P_{W,i}^h \end{bmatrix}, \quad i = 1, \dots n$$

which by mapping the homogeneous coordinates p_i^h back to the original coordinates p_i yields the $2n$ expressions:

$$\begin{aligned} u_i &= \frac{m_1 \cdot P_{W,i}^h}{m_3 \cdot P_{W,i}^h}, \quad i = 1, \dots, n \\ v_i &= \frac{m_2 \cdot P_{W,i}^h}{m_3 \cdot P_{W,i}^h}, \quad i = 1, \dots, n, \end{aligned}$$

or equivalently (via algebraic manipulation) the expressions:

$$\begin{aligned} u_i(m_3 \cdot P_{W,i}^h) - (m_1 \cdot P_{W,i}^h) = 0, \quad i = 1, \dots, n \\ v_i(m_3 \cdot P_{W,i}^h) - (m_2 \cdot P_{W,i}^h) = 0, \quad i = 1, \dots, n. \end{aligned} \tag{8.16}$$

Now, these $2n$ equations can be combined together in one large matrix equation:

$$\tilde{P}m = 0, \quad m = \begin{bmatrix} m_1^\top \\ m_2^\top \\ m_3^\top \end{bmatrix}, \tag{8.17}$$

where $m \in \mathbb{R}^{12\times 1}$ is a vector consisting of the stacked rows of M and $\tilde{P} \in \mathbb{R}^{2n\times 12}$ is a matrix of *known* coefficients determined by the quantities u_i, v_i, and $P^h_{W,i}$. For a more concrete representation of how $\tilde{P}$ is defined, the first couple rows are given by:

$$\tilde{P} = \begin{bmatrix} -(P^h_{W,1})^\top & 0_{1\times 4} & u_1(P^h_{W,1})^\top \\ 0_{1\times 4} & -(P^h_{W,1})^\top & v_1(P^h_{W,1})^\top \\ -(P^h_{W,2})^\top & 0_{1\times 4} & u_2(P^h_{W,2})^\top \\ \vdots & \vdots & \vdots \end{bmatrix}. \tag{8.18}$$

Note that $n \geq 6$ (i.e. at least 6 correspondences have been made) is a requirement to ensure that m can be uniquely defined. Ideally, with this sufficient number of correspondences the equation (8.18) could be directly solved. However, in practice a more robust procedure is to build $\tilde{P}$ with more than 6 points, which would lead to an overdetermined set of equations that may not have a solution[7]! Therefore, the determination of m is accomplished by formulation the optimization problem:

$$\begin{aligned} \min_{m}.\ & \|\tilde{P}m\|^2, \\ \text{s.t. } & \|m\|^2 = 1, \end{aligned} \tag{8.19}$$

where the constraint $\|m\|^2 = 1$ is required to ensure that the optimization problem does not simply choose $m_i = 0$ for each $i = 1, \dots, 12$. This optimization problem is called a *constrained least-squares* problem.

Example 8.2.1 (Constrained Least-Squares)**.** The constrained least squares problem

$$\begin{aligned} \min_{x}.\ & \|Ax\|^2, \\ \text{s.t. } & \|x\|^2 = 1, \end{aligned}$$

with $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m\times n}$ and $m > n$ is a finite-dimensional optimization problem. Consider the corresponding Lagrangian:

$$L = x^\top A^\top A x + \lambda(1 - x^\top x),$$

and the necessary optimality conditions:

$$\begin{aligned} \nabla_x L &= 2(A^\top A - \lambda I)x = 0, \\ \nabla_\lambda L &= 1 - x^\top x = 0. \end{aligned}$$

The first NOC can be rewritten as $A^\top A x = \lambda x$, and therefore any x that satisfies this condition must be an eigenvector of the matrix $A^\top A$. Additionally, while all the eigenvectors satisfy this condition the minimizer is the eigenvector associated with the smallest eigenvalue. This eigenvector can efficiently be computed by a singular value decomposition of $A = U\Sigma V^\top$ and then choosing m to be the column of V associated with the smallest singular value (since $A^\top A = V\Sigma^2 V^\top$).

[7] This is particularly true in real-world applications where noise corrupts the data.

8.2.2 Direct Linear Calibration: Step 2

Once the optimization problem (8.19) has been solved for m the homography M is completely defined. The next step in the camera calibration process is to extract the intrinsic and extrinsic camera parameters from the matrix M. For this section the matrix M is expressed in terms of its columns:

$$M = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \end{bmatrix},$$

where c_i is the i-th column of M. It is now possible to factorize M as:

$$M = K \begin{bmatrix} R & t \end{bmatrix}, \tag{8.20}$$

by taking the first three columns of M and performing a *RQ factorization*:

$$\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} = KR, \tag{8.21}$$

where R is an orthogonal matrix and K is an upper triangular matrix. Once K is known the vector t can be computed by $t = K^{-1}c_4$.

8.2.3 A Flexible Camera Calibration Method (Zhang, 2000):

The homography M is defined for a *specific* set of extrinsic parameters R and t. In practice it might be desirable to estimate the camera's intrinsic parameters from N *different* images from different perspectives (and therefore with N different homographies). In this case the procedure described in [8] can be used to extract the intrinsic parameters K.

This approach begins by assuming that the known points P_W for each individual image lie on a plane. For example the calibration "scene" might consist of a pattern (e.g. a checkerboard pattern) on a planar surface. In this case, it can simply be assumed that the world frame origin also lies on this plane such that $Z_W = 0$ for all points on the plane. Since $Z_W = 0$ the relationship between p^h and P_W^h given by (8.13) can be simplified to:

$$p^h = \tilde{M}\tilde{P}_W^h, \tag{8.22}$$

with

$$\tilde{M} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}, \quad \tilde{P}_W^h = \begin{bmatrix} X_W & Y_W & 1 \end{bmatrix}^\top, \tag{8.23}$$

where $\tilde{M}$ is the simplified homography matrix, $\tilde{P}_W^h$ is the simplified position of the point P in world frame written in homogeneous coordinates, and r_i is the i-th column of the rotation matrix R. Note that the homography matrix $\tilde{M}$ can still be estimated using the same procedure discussed before.

A set of constraints on the intrinsic parameter matrix K are next identified by writing the homography $\tilde{M}$ as:

$$\begin{bmatrix} \tilde{c}_1 & \tilde{c}_2 & \tilde{c}_3 \end{bmatrix} = \begin{bmatrix} Kr_1 & Kr_2 & Kt \end{bmatrix}.$$

[8] Z. Zhang. "A Flexible New Technique for Camera Calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000)

This relationship, and the knowledge that r_1 and r_2 are orthonormal, leads to the following constraints:

$$\tilde{c}_1^\top B\tilde{c}_2 = 0, \quad \tilde{c}_1^\top B\tilde{c}_1 = \tilde{c}_2^\top B\tilde{c}_2, \tag{8.24}$$

where $B = K^{-\top}K^{-1} \in \mathbb{R}^{3\times 3}$ is a *symmetric* matrix. Solving for the intrinsic camera parameters K can therefore be accomplished by using the constraints (8.24) to solve for the symmetric matrix B, and then to use the definition of B to back out the parameters that define K.

Several useful tricks can be employed to compute the matrix B from the constraints (8.24). The main trick is to notice that even though B consists of nine parameters, since it is symmetric only six parameters are required to fully specify it. Therefore $B \in \mathbb{R}^{3\times 3}$ is reparameterized as a vector $b \in \mathbb{R}^6$ as:

$$b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^\top . \tag{8.25}$$

This reparameterization is useful because it allows us to rewrite the expression $\tilde{c}_i^\top B\tilde{c}_j$ as:

$$\tilde{c}_i^\top B\tilde{c}_j = v_{ij}^\top b, \tag{8.26}$$

where:

$$v_{ij} = \begin{bmatrix} \tilde{c}_{i1}\tilde{c}_{j1}, & \tilde{c}_{i1}\tilde{c}_{j2} + \tilde{c}_{i2}\tilde{c}_{j1}, & \tilde{c}_{i2}\tilde{c}_{j2}, & \tilde{c}_{i3}\tilde{c}_{j1} + \tilde{c}_{i1}\tilde{c}_{j3}, & \tilde{c}_{i3}\tilde{c}_{j2} + \tilde{c}_{i2}\tilde{c}_{j3}, & \tilde{c}_{i3}\tilde{c}_{j3} \end{bmatrix}^\top ,$$

where $\tilde{c}_{ik}$ is the k-th element of the column vector $\tilde{c}_i$ and $\tilde{c}_{jk}$ is the k-th element of the column vector $\tilde{c}_j$. With this reparameterization, the constraints (8.24) can be rewritten as:

$$\begin{aligned} \tilde{c}_1^\top B\tilde{c}_2 = 0 &\implies v_{12}^\top b = 0 \\ \tilde{c}_1^\top B\tilde{c}_1 = \tilde{c}_2^\top B\tilde{c}_2 &\implies (v_{11} - v_{22})^\top b = 0, \end{aligned}$$

or by combining them:

$$\begin{bmatrix} v_{12}^\top \\ (v_{11} - v_{22})^\top \end{bmatrix} b = 0, \tag{8.27}$$

which is linear in the unknowns b. Importantly, while the homographies M are different for each image, the intrinsic camera parameters (i.e. the vector b) are the same! Therefore for N images from the same camera (but with potentially different perspectives) these constraints (8.27) can be stacked to give:

$$Vb = 0, \tag{8.28}$$

where $V \in \mathbb{R}^{2N\times 6}$. In the case where the skewness parameter γ is included in K there must be $N \geq 3$ images in order to specify B uniquely. Similar to how the homography for an image M was computed in the previous section, the vector b will be specified by the solution to the constrained least squares problem:

$$\begin{aligned} \min_{b}.\ & \|Vb\|^2, \\ \text{s.t. } & \|b\|^2 = 1. \end{aligned} \tag{8.29}$$

Once b has been determined, the intrinsic camera parameters K can be solved for recalling the definition of $B = K^{-T}K^{-1}$. In particular, the intrinsic parameters are given by:

$$\begin{aligned}
v_0 &= \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2}, \\
\lambda &= B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}}, \\
\alpha &= \sqrt{\frac{\lambda}{B_{11}}}, \\
\beta &= \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}}, \\
\gamma &= \frac{-B_{12}\alpha^2\beta}{\lambda}, \\
u_0 &= \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{\lambda},
\end{aligned} \tag{8.30}$$

where λ can be though of as a scaling parameter that accounts for the fact that there are five unknown camera intrinsic parameters but six degrees of freedom in B.

Once the camera intrinsic parameters K have been extracted from this procedure, given any new homography $\tilde{M}$ the extrinsic parameters can be computed by:

$$\begin{aligned}
r_1 &= \frac{K^{-1}\tilde{c}_1}{\|K^{-1}\tilde{c}_1\|}, \\
r_2 &= \frac{K^{-1}\tilde{c}_2}{\|K^{-1}\tilde{c}_2\|}, \\
r_3 &= r_1 \times r_2, \\
t &= \frac{K^{-1}\tilde{c}_3}{\|K^{-1}\tilde{c}_1\|}.
\end{aligned} \tag{8.31}$$

As one final step, it is noted that the matrix R defined with columns r_1, r_2, and r_3 will not in generally satisfy the properties of a rotation matrix (i.e. orthonormality). One final step to this overall procedure is to correct this issue by finding the rotation matrix that best corresponds to these column vectors. This is accomplished again by optimization, and in particular by formulating the problem:

$$\begin{aligned}
&\min_{R}.\ \|R - Q\|^2, \\
&\text{s.t. } R^\top R = I,
\end{aligned} \tag{8.32}$$

where

$$Q = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}.$$

This problem is solved by choosing $R = UV^\top$ where U and V are defined by the singular value decomposition of $Q = U\Sigma V^\top$.

8.3 Limitations

8.3.1 Radial Distortion

The pinhole camera model provides a nominal camera model for which it is relatively straightforward to develop a mathematical model of the perspective projection. However, in practice this model is not a perfect representation of the imaging process. One such effect that is not captured by the pinhole model is *radial distortion*, which is an effect seen in real lenses where either barrel distortion or pincushion distortion will affect the real pixel coordinates. Images showing both barrel and pincushion distortion are provided in Figure 8.4.

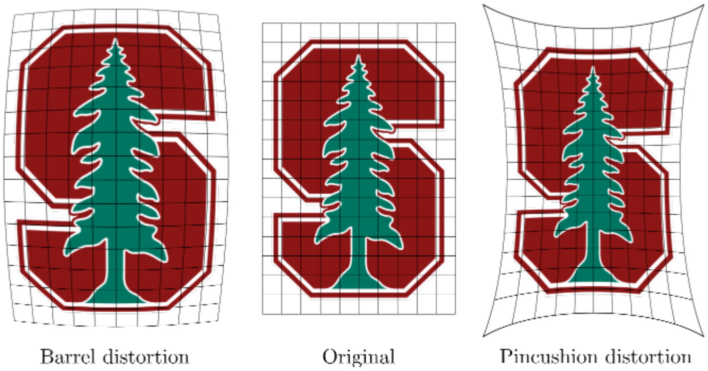

Figure 8.4: Different kinds of radial distortions that are seen in real lenses, which may affect the accuracy of the pinhole camera model.

There are methods that can be used to correct for image distortion. A simple and efficient way is to model the relationship between the ideal pixel coordinates (u, v) and the distorted pixel coordinates (u_d, v_d) as:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} u_d \\ v_d \end{bmatrix} (1 + kr^2) \begin{bmatrix} u - u_{cd} \\ v - v_{cd} \end{bmatrix} + \begin{bmatrix} u_{cd} \\ v_{cd} \end{bmatrix} \tag{8.33}$$

where $k \in \mathbb{R}$ is the radial distortion factor, (u_{cd}, v_{cd}) are the pixel coordinates of the image center, and $r^2 = (u - u_{cd})^2 + (v - v_{cd})^2$ is the square of the distance between the ideal pixel location and the center of distortion.. Note that k differs in different cameras and needs to be pre-determined.

8.3.2 Measuring Depth

Once the camera intrinsic and extrinsic parameters K, R, and t are known it is still not possible to map pixel coordinates to the corresponding point in space. Mathematically this is a result of the matrix M in (8.14) not being invertible, but intuitively this is because the distance along the line of sight from p to P in Figure 8.1 can not be determined!

However, there are some techniques that can enable depth estimates to be made with a single camera. One approach is known as *depth from focus*, where several images are taken until the projection of point P is in focus. Based on the thin lens model, when this occurs:

$$\frac{1}{z} + \frac{1}{Z} = \frac{1}{f},$$

where f is the focal length, Z is the depth of the point P in camera frame, and z is the depth of the image plane in the camera frame when the projection of point P is in focus. Since f and z are known, the depth Z can therefore be computed. If two cameras are used, depth estimation is possible via *binocular reconstruction* or *stereo vision*. This approach requires known corresponding pixel coordinates p and p' of each camera, and then uses *triangulation* to determine the 3D position of the source point P in the scene.

8.4 Exercises

8.4.1 Camera Calibration

Complete *Problem 1: Camera Calibration* located in the online repository:

`https://github.com/PrinciplesofRobotAutonomy/AA274A_HW3`,

where you will estimate the intrinsic parameters of a camera using the method described in Section 8.2.3.