Course Notes 2: Camera Calibration*

1 Camera Calibration

To precisely know the transformation from the real, 3D world into digital images requires prior knowledge of many of the camera's intrinsic parameters. If given an arbitrary camera, we may or may not have access to these parameters. We do, however, have access to the images the camera takes. Therefore, can we find a way to deduce them from images? This problem of estimating the extrinsic and intrinsic camera parameters is known as **camera calibration**.

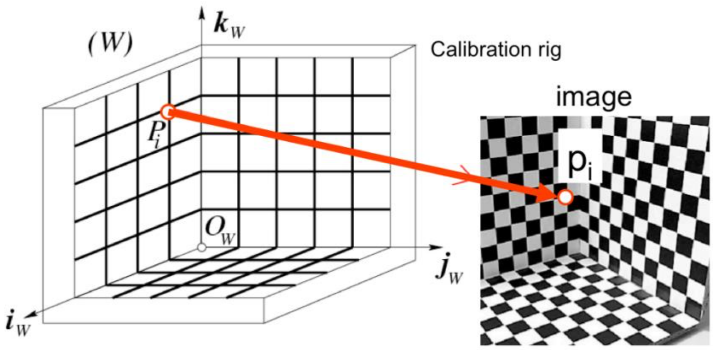

Figure 1: The setup of an example calibration rig.

Specifically, we do this by solving for the intrinsic camera matrix K and the extrinsic parameters $R, \mathbf{t}$ from Equation 1.

$$\begin{aligned} \mathbf{p}_i &= K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{P}_i \\ &= M\mathbf{P}_i \end{aligned} \tag{1}$$

We can describe this problem in the context of a calibration rig, such as the one shown in Figure 1. The rig usually consists of a simple pattern (i.e. checkerboard) with known dimensions. Furthermore, the rig defines our world reference frame with origin $\mathbf{O}_W$ and axes $\mathbf{i}_W$, $\mathbf{j}_W$, $\mathbf{k}_W$. From the rig's known pattern, we have known points in the world reference frame $\mathbf{P}_1, ..., \mathbf{P}_n$. Finding these points in the image we take from the camera gives corresponding points in the image $\mathbf{p}_1, ..., \mathbf{p}_n$.

*Most contents are from

- R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision (2nd Edition)
- K. Hata and S. Savarese. Course notes of Stanford CS231A

Denote the three rows of M as $\mathbf{m}_1^T, \mathbf{m}_2^T, \mathbf{m}_3^T$, i.e.,

$$M = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \tag{2}$$

We set up a linear system of equations from the n correspondences such that for each correspondence $\mathbf{P}_i, \mathbf{p}_i$:

$$\mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{P}_i^T \mathbf{m}_1}{\mathbf{P}_i^T \mathbf{m}_3} \\ \\ \frac{\mathbf{P}_i^T \mathbf{m}_2}{\mathbf{P}_i^T \mathbf{m}_3} \end{bmatrix} \tag{3}$$

As we see from the above equation, each correspondence gives us two equations and, consequently, two constraints for solving the unknown parameters contained in M. Before, we know that the camera matrix has 11 unknown parameters. This means that we need at least 6 correspondences to solve this. However, in the real world, we often use more, as our measurements are often noisy. To explicitly see this, we can derive a pair of equations that relate u_i and v_i with $\mathbf{P}_i$

$$\mathbf{P}_i^T \mathbf{m}_1 - u_i(\mathbf{P}_i^T \mathbf{m}_3) = 0$$
$$\mathbf{P}_i^T \mathbf{m}_2 - v_i(\mathbf{P}_i^T \mathbf{m}_3) = 0$$

Given n pairs of these corresponding points, the entire linear system of equations becomes

$$\begin{aligned} \mathbf{P}_1^T \mathbf{m}_1 - u_1(\mathbf{P}_1^T \mathbf{m}_3) &= 0 \\ \mathbf{P}_1^T \mathbf{m}_2 - v_1(\mathbf{P}_1^T \mathbf{m}_3) &= 0 \\ &\vdots \\ \mathbf{P}_n^T \mathbf{m}_1 - u_n(\mathbf{P}_n^T \mathbf{m}_3) &= 0 \\ \mathbf{P}_n^T \mathbf{m}_2 - v_n(\mathbf{P}_n^T \mathbf{m}_3) &= 0 \end{aligned}$$

This can be formatted as a matrix-vector product shown below:

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0 \tag{4}$$

(An expanded version of this equation is given in Equation 10). When $2n > 11$, this homogeneous linear system is overdetermined. For such a system $\mathbf{m} = 0$ is always a trivial solution. Furthermore, even if there were some other $\mathbf{m}$ that were a nonzero solution, then $\forall k \in \mathbb{R}, k\mathbf{m}$ is also a solution. Therefore, to constrain our solution, we complete the following minimization:

$$\begin{aligned} \underset{\mathbf{m}}{\text{minimize}} \quad & \|P\mathbf{m}\|^2 \\ \text{subject to} \quad & \|\mathbf{m}\|^2 = 1 \end{aligned} \tag{5}$$

To solve this minimization problem, we simply use singular value decomposition. If we let $P = U\Sigma V^T$, then the solution to the above minimization is to set $\mathbf{m}$ equal to the last column of V. The derivation for this solution is outside the scope of this class and you may refer to Section 5.3 of Hartley & Zisserman[1] on pages 592-593 for more details.

After reformatting the vector $\mathbf{m}$ into the matrix M, we now want to explicitly solve for the extrinsic and intrinsic parameters. Denote $R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}$, $\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$, and since $K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \frac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, we know our SVD-solved projection matrix (denoted by $\mathcal{M}$) is known up to scale, which means that the true values of the project matrix M are some scalar multiple of $\mathcal{M}$, i.e.,

$$\rho\mathcal{M} = M = \begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + u_0\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}. \tag{6}$$

Dividing by the scaling parameter gives

$$\mathcal{M} = \frac{1}{\rho}M = \frac{1}{\rho}\begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + u_0\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}. \tag{7}$$

Because we have obtained all the entries of $\mathcal{M}$, denote $\mathcal{M} = \begin{bmatrix} A & \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T & b_1 \\ \mathbf{a}_2^T & b_2 \\ \mathbf{a}_3^T & b_3 \end{bmatrix}$, solving for the intrinsics gives

$$\begin{aligned} \rho &= \pm\frac{1}{\|\mathbf{a_3}\|} \\ u_0 &= \rho^2(\mathbf{a_1}\cdot\mathbf{a_3}) \\ v_0 &= \rho^2(\mathbf{a_2}\cdot\mathbf{a_3}) \\ \cos\theta &= -\frac{(\mathbf{a_1}\times\mathbf{a_3})\cdot(\mathbf{a_2}\times\mathbf{a_3})}{\|\mathbf{a_1}\times\mathbf{a_3}\|\cdot\|\mathbf{a_2}\times\mathbf{a_3}\|} \\ \alpha &= \rho^2\|\mathbf{a_1}\times\mathbf{a_3}\|\sin\theta \\ \beta &= \rho^2\|\mathbf{a_2}\times\mathbf{a_3}\|\sin\theta \end{aligned} \tag{8}$$

The extrinsics are

$$\begin{aligned} \mathbf{r_1} &= \frac{\mathbf{a_2}\times\mathbf{a_3}}{\|\mathbf{a_2}\times\mathbf{a_3}\|} \\ \mathbf{r_2} &= \mathbf{r_3}\times\mathbf{r_1} \\ \mathbf{r_3} &= \rho\mathbf{a_3} \\ \mathbf{t} &= \rho K^{-1}\mathbf{b} \end{aligned} \tag{9}$$

We leave the derivations as a class exercise or you can refer to Section 1.3.1 of the Forsyth & Ponce textbook. Section 2 explains the derivation of Equation 4 using light linear algebra.

[1] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision (2nd Edition). Cambridge University Press, 2004

With the calibration procedure complete, we warn against degenerate cases. Not all sets of n correspondences will work. For example, if the points $\mathbf{P}_i$ lie on the same plane, then the system will not be able to be solved. These unsolvable configurations of points are known as **degenerate configurations**. More generally, degenerate configurations have points that lie on the intersection curve of two quadric surfaces. Although this is outside the scope of the class, you can find more information in Section 1.3 of the Forsyth & Ponce textbook [2].

2 [Detail] Derivation Using Light Linear Algebra

Given any pair of 3D-2D correspondence, i.e., $[X, Y, Z]^T \rightarrow [u, v]^T$, we have (in homogeneous coordinates)

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{aligned} su &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ sv &= m_{21}X + m_{22}Y + m_{23}Z + m_{24} \\ s &= m_{31}X + m_{32}Y + m_{33}Z + m_{34} \end{aligned}$$

$$\Rightarrow \begin{aligned} u &= \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ v &= \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \end{aligned}$$

$$\Rightarrow \begin{aligned} (m_{31}X + m_{32}Y + m_{33}Z + m_{34})\,u = m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ (m_{31}X + m_{32}Y + m_{33}Z + m_{34})\,v = m_{21}X + m_{22}Y + m_{23}Z + m_{24} \end{aligned}$$

$$\Rightarrow \begin{aligned} m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24} \end{aligned}$$

$$\Rightarrow \begin{aligned} m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u = 0 \\ m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v = 0 \end{aligned}$$

[2] D. Forsyth and J. Ponce. Computer Vision: A Modern Approach (2nd Edition). Pearson, 2012.

With n pairs of 3D-2D corresponding points, i.e., $\{[X_i, Y_i, Z_i]^T\} \rightarrow \{[u_i, v_i]^T\}$, $(1 \leq i \leq n)$, we have the system of linear equations as has been given in Equation 4 as:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

3 [Optional] Handling Distortion in Calibration

So far, we have been working with ideal lenses which are free from any distortion. However, as seen before, real lenses can deviate from rectilinear projection, which requires more advanced methods. This section provides just a brief introduction to handling distortions.

Often, distortions are radially symmetric because of the physical symmetry of the lens. We model the radial distortion with an isotropic transformation:

$$Q\mathbf{P}_i = \begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} M\mathbf{P}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \mathbf{P}_i \tag{11}$$

Let $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$ denote the three rows of matrix Q, we can rewrite this into a system of equations as before, we get

$$u_i\mathbf{q}_3\mathbf{P}_i = \mathbf{q}_1\mathbf{P}_i$$
$$v_i\mathbf{q}_3\mathbf{P}_i = \mathbf{q}_2\mathbf{P}_i$$

This system, however, is no longer linear, and we require the use of nonlinear optimization techniques, which are covered in Section 22.2 of Forsyth & Ponce. We can simplify the nonlinear optimization of the calibration problem if we make certain assumptions. In radial distortion, we note that the ratio between two coordinates u_i and v_i is not affected. We can compute this ratio as

$$\frac{u_i}{v_i} = \frac{\frac{\mathbf{m}_1\mathbf{P}_i}{\mathbf{m}_3\mathbf{P}_i}}{\frac{\mathbf{m}_2\mathbf{P}_i}{\mathbf{m}_3\mathbf{P}_i}} = \frac{\mathbf{m}_1\mathbf{P}_i}{\mathbf{m}_2\mathbf{P}_i} \tag{12}$$

Assuming that n correspondences are available, we can set up the system of linear equations:

$$v_1(\mathbf{m}_1\mathbf{P}_1) - u_1(\mathbf{m}_2\mathbf{P}_1) = 0$$
$$\vdots$$
$$v_n(\mathbf{m}_1\mathbf{P}_n) - u_n(\mathbf{m}_2\mathbf{P}_n) = 0$$

Similar to before, this gives a matrix-vector product that we can solve via SVD:

$$\begin{bmatrix} v_1\mathbf{P}_1^T & -u_1\mathbf{P}_1^T \\ \vdots & \vdots \\ v_n\mathbf{P}_n^T & -u_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \end{bmatrix} = 0 \tag{13}$$

Once $\mathbf{m}_1$ and $\mathbf{m}_2$ are estimated, $\mathbf{m}_3$ can be expressed as a nonlinear function of $\mathbf{m}_1, \mathbf{m}_2$, and λ. This requires to solve a nonlinear optimization problem whose complexity is much simpler than the original one.