

# 1. Camera calibration theory

We start with a set of corresponding points between the planar pattern and an image. We say that the point  $(x, y)$  in the pattern (which may be measured in millimeters for example) corresponds to the pixel  $(u, v)$  in the image. From these correspondences, we can estimate an homography that transforms the coordinates of the planar pattern onto the corresponding coordinates in the image,

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (1)$$

The way this is done has been covered in a previous session.

Now given the projection matrix of the camera,  $P = K(R|t)$ , we can derive an expression for this homography that relates it to the camera matrix. Assume the world reference is attached to the pattern so that the pattern plane is  $z = 0$ . The projection equation writes

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim P \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} \sim K(R|t) \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} \sim K \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (2)$$

where  $r_1$  and  $r_2$  are the first two columns of  $R$ . From this and the homography definition above we deduce that  $H \sim K \begin{pmatrix} r_1 & r_2 & t \end{pmatrix}$ . It will be convenient to write the columns of  $H$  separately as

$$\begin{pmatrix} h_1 & h_2 & h_3 \end{pmatrix} \sim \begin{pmatrix} Kr_1 & Kr_2 & Kt \end{pmatrix} \quad (3)$$

and equivalently

$$\begin{pmatrix} K^{-1}h_1 & K^{-1}h_2 & K^{-1}h_3 \end{pmatrix} \sim \begin{pmatrix} r_1 & r_2 & t \end{pmatrix}. \quad (4)$$

We know  $H$  and our goal is to find  $K$  and eventually  $R$  and  $t$ . For this, we need to enforce some constraints on the unknowns. To start, we will use the fact that  $R$  is an orthonormal matrix. That is, its columns are unit vectors orthogonal to each other. In particular,  $r_1 \cdot r_2 = 0$  and  $r_1 \cdot r_1 = r_2 \cdot r_2 = 1$ . From this we get the following equations on  $K$ ,

$$h_1^T K^{-T} K^{-1} h_2 = 0 \quad (5)$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \quad (6)$$

Here,  $K^{-T} K^{-1}$  turns out to be the fancy *Image of the Absolute Conic*, which is usually written as  $\omega$ . For computational purposes, it will be simpler to first find  $\omega$  and then recover the camera matrix  $K$  from it. The above constraints are in fact linear on  $\omega$

$$h_1^T \omega h_2 = 0 \quad (7)$$

$$h_1^T \omega h_1 = h_2^T \omega h_2 \quad (8)$$

Lets write the terms in  $\omega$  as

$$\omega = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_4 & x_5 \\ x_3 & x_5 & x_6 \end{pmatrix} \quad (9)$$

and

$$\mathbf{x} = (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6)^T. \quad (10)$$

Now we can rewrite the expressions in (7) and (8) as

$$h_i^T \omega h_j = (h_{1i} \ h_{2i} \ h_{3i}) \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_4 & x_5 \\ x_3 & x_5 & x_6 \end{pmatrix} \begin{pmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \end{pmatrix} = \mathbf{v}_{ij}^T \mathbf{x} \quad (11)$$

with

$$\mathbf{v}_{ij}^T = (h_{1i}h_{1j}, \ h_{1i}h_{2j} + h_{2i}h_{1j}, \ h_{1i}h_{3j} + h_{3i}h_{1j}, \ h_{2i}h_{2j}, \ h_{2i}h_{3j} + h_{3i}h_{2j}, \ h_{3i}h_{3j}) \quad (12)$$

so that the equations write as

$$\begin{aligned} \mathbf{v}_{12}^T \mathbf{x} &= 0 \\ (\mathbf{v}_{11}^T - \mathbf{v}_{22}^T) \mathbf{x} &= 0. \end{aligned} \quad (13)$$

These are two linear equations on the coefficients of  $\omega$ . By using more than one image, we get more equations. Since there are 6 unknowns, we need at least 3 images of the planar pattern (there are actually only 5 unknowns since the scale of  $\omega$  is irrelevant). It is also possible to add additional linear equations based on prior knowledge on  $K$ , such as 0 skew, aspect ration 1 or known principal point.

The result is an homogeneous linear system that we can solve using the SVD decomposition, and get  $\omega$ . From that, we can recover the camera calibration matrix  $K$  by solving  $\omega = K^{-T} K^{-1}$  using the Cholesky factorization.

Then, knowing  $K$ , the camera rotation and translation can be recovered from equation (4).