Camera calibration (DLT, Zhang's)

Prepared by: Udit Singh Parihar, Sai Shubodh P

In this note, we discuss how to obtain the intrinsic and extrinsic calibration parameters of a camera given the coordinates of object points $\vec{X}$ and the observed coordinates of those object points in the image $\vec{x}$. There are two methods to do this:

1. Direct Linear Transform: This method maps any object point $\vec{X}$ to the image point $\vec{x}$ and assumes that we do not know anything about the intrinsic nor extrinsic parameters. Here, the object points should be non-planar in the 3 dimensional world. In this method, we will be needing at least 6 pairs of points to extract the parameters.

2. Zhang's method: This also assumes that we do not know anything about the intrinsic nor extrinsic parameters. But here, the object points are considered to lie on a 2D known structure of known size, typically a checkerboard. We leverage the fact that they lie on the same plane to make it easier to extract the world coordinates using known structures. Hence it reduces the number of minimum pairs require to obtain camera's parameters, specifically equal to 4 pairs.

1 Direct Linear Transform

The pin hole camera equation:

$$\vec{x} = KR[I_3| - X_O]\vec{X} \tag{1}$$

1.1 Finding combined calibration matrix, P

We can combine the camera intrinsic and extrinsic matrices as a single matrix P:

$$\vec{x} = P\vec{X} \tag{2}$$

Expanding x and X:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix}$$

$$\begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} = P \begin{bmatrix} U/T \\ V/T \\ W/T \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$y = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

The number of unknowns in the above equations are 11 (And not 12 since we are using homogeneous coordinates and 1 will be for normalization). Each point will give us the above two equations and we have 11 unknowns, therefore we will be needing at least 6 points i.e. I≥6.

Therefore, for a point i:

$$\vec{x_i} = P\vec{X_i} \qquad i = 1, 2..., I \tag{3}$$

We now rearrange the equations to get a solvable system of equations:

$$x_i = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} X_i$$

$$= \begin{bmatrix} A^T \\ B^T \\ C^T \end{bmatrix} X_i$$

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \begin{bmatrix} A^T X_i \\ B^T X_i \\ C^T X_i \end{bmatrix}$$

$$x_i = \frac{u_i}{w_i} = \frac{A^T X_i}{C^T X_i}$$

$$y_i = \frac{v_i}{w_i} = \frac{B^T X_i}{C^T X_i}$$

By multiplying both sides of the above two equations with $C^T X_i$ and taking transpose on both sides, we get

$$-X_i^T A \qquad + x_i X_i^T C = 0$$
$$-X_i^T B + y_i X_i^T C = 0$$

The above system of equations is linear in the parameters A, B and C. Collecting the elements of P within a vector p:

$$p = (p_k) = \begin{bmatrix} A \\ B \\ C \end{bmatrix} = vec(P^T)$$

Note that p is written with the rows of P as column vectors, one below the other with dimensions 12×1. Stacking A, B, C in p and rewriting the above two equations:

$$a_{x_i}^T p = 0$$
$$a_{y_i}^T p = 0$$
$$\text{where} \quad p = (p_k) = vec(P^T)$$
$$a_{x_i}^T = (-X_i^T, 0^T, x_i X_i^T)$$
$$= (-X_i, -Y_i, -Z_i, -1, 0, 0, 0, 0, x_i X_i, x_i Y_i, x_i Z_i, x_i)$$
$$a_{y_i}^T = (0^T, -X_i^T, y_i X_i^T)$$
$$= (0, 0, 0, 0, -X_i, -Y_i, -Z_i, -1, y_i X_i, y_i Y_i, y_i Z_i, y_i)$$

Expressing the above for I points in a single equation:

$$\begin{bmatrix} a_{x_1}^T \\ a_{y_1}^T \\ ... \\ a_{x_i}^T \\ a_{y_i}^T \\ ... \\ a_{x_I}^T \\ a_{y_I}^T \end{bmatrix} = M_{2I \times 12} \quad p_{12 \times 1} \neq 0 \quad \text{(Not equal to?)}$$

The left hand side of the above equation is not exactly equal to 0 because the observations can be redundant/inaccurate. So we model it as a minimization problem as follows:

$$\text{Find p in } Mp = w$$
$$\text{such that} \quad \|w\| \quad \text{is minimized with} \quad \|p\| = 1.$$

Decomposing M using Singular Value Decomposition (SVD), we get

$$M_{2I \times 12} = U_{2I \times 12} S_{12 \times 12} V_{12 \times 12}^T$$

Applying SVD and knowing its properties, we can conclude the solution to above minimization problem is the last column vector in V that is $v_{12}, i.e.$ p $= v_{12}$.

1.2 Finding K, R and T

However, we have only obtained P as of now, we have to now find K, R and X_O through decomposition of P.

Rewriting the camera equation,

$$x = KR[I_3| - X_O]X = [H_{\infty_{3 \times 3}}|h_{3 \times 1}]$$

$$where \quad H_\infty = KR \quad and \quad h = -KRX_O$$

We can easily find the projection center or translation vector X_O using $X_O = -H_\infty^{-1}h$.

For finding K and R, we take QR decomposition of H_∞^{-1} which yields us the matrices R^T and K^{-1}.

$$H_\infty^{-1} = (KR)^{-1} = R^{-1}K^{-1} = R^TK^{-1}$$

We do not take QR decomposition of H_∞ directly because QR decomposes a matrix as a product of orthogonal matrix (first term) and upper triangular matrix (second) in the order it is written here (and not the other way round) but H_∞ is of the form triangular matrix first and then rotation (orthogonal) matrix. Now we can just take transpose and inverse of those obtained matrices to get R and K. Hence, we have thus obtained K, R and T matrices!

2 Zhang's method for calibration using checkerboard

In Zhang's method, in order to calculate the camera's intrinsic (K) and extrinsic parameters $(R$ and $t)$, we would perform our experiment on planar world coordinates. In particular, checkerboard is used as planar world and world's origin is fixed at one of its corner.

Due to the known dimensions of checkerboard's boxes and known location of world's origin at one of the corners, we could find out the location of all other points on the checkerboard by exploiting Cartesian coordinate arrangement of the boxes. So a point, p_i on checkerboard is given by:

$$p_i = \begin{bmatrix} X_i \\ Y_i \\ 0 \end{bmatrix}$$

Similarly, in order to find out the corresponding pixels in image for our $3D$ world points, we could use feature detectors algorithms like Harris Corner Detector to detect corners of the boxes of the board.

Now we have known $3D$ points and their corresponding $2D$ pixels coordinates. So we can use linear algebra and optimization techniques to obtain K, R and t.

2.1 Finding Homography, H

Rewriting camera equation by writing rotation matrix, R in column's notation as:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} c & s & x_0 \\ 0 & c(1+m) & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Since $Z_i = 0$, the above equation reduces to:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_{3\times1} = K_{3\times3} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}_{3\times3} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}_{3\times1}$$

Let, $H = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$, be a homography matrix. Then,

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_{3\times1} = H \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}_{3\times1}$$

Writing, $P_i = \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$ and $H = \begin{bmatrix} H_1^T \\ H_2^T \\ H_3^T \end{bmatrix}$, then writing above equation as:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \begin{bmatrix} H_1^T P_i \\ H_2^T P_i \\ H_3^T P_i \end{bmatrix}$$

Comparing LHS and RHS, element-wise:

$$\mathrm{x}_i = \frac{H_1^T P_i}{H_3^T P_i}$$
$$\mathrm{y}_i = \frac{H_2^T P_i}{H_3^T P_i}$$

Converting above system of linear equations into matrix form:

$$\begin{bmatrix} -P_i^T{}_{1x3} & 0_{1\times3}^T & x_i P_i^T \\ 0_{1x3}^T & -P_i^T{}_{1\times3} & y_i P_i^T \end{bmatrix}_{2I\times9} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}_{9\times1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{2I}$$

Representing above equation as $M\,\vec{h} = \vec{0}$, and solving using SVD decomposition of M matrix to obtain vector $\vec{h}$, similar to technique mention in DLT section. So our H matrix would be:

$$H = h.reshape(3,3)$$
$$H = \frac{H}{H[2,2]}$$

Projecting $3D$ points to image using obtained homography matrix, H.

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = H\vec{P_i}$$

The re-projection error between projected pixel points, $\vec{x_i}'$ and real pixel points, $\vec{x_i}$, is given by:

$$error = \sum_{i=0}^{I} \|\vec{x_i} - \vec{x_i}'\|_2$$
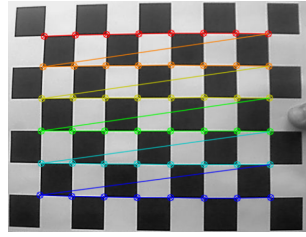

Figure 1: 3D points projected onto checkerboard's image

2.2   Finding camera matrix, K

Now we could decompose homography matrix, H to obtain K.

Writing H in column form, we have $H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$. So,

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$
$$r_1 = K^{-1}h_1$$
$$r_2 = K^{-1}h_2$$

As r_1 and r_2 are orthonormal vectors, so they follow:

$$\vec{r_1}^T \vec{r_2} = 0$$
$$\|\vec{r_1}\| = \|\vec{r_2}\| = 1$$

Substituting value of $\vec{r_1}$ and $\vec{r_2}$ from above equations, we have:

$$h_1^T K^{-T} K^{-1} h_2 = 0$$
$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2$$

Then taking RHS to LHS, we have:

$$h_1^T K^{-T} K^{-1} h_1 - h_2^T K^{-T} K^{-1} h_2 = 0$$

Let, $B = K^{-T} K^{-1}$, which would be a symmetric matrix, so our pair of equations reduced to:

$$h_1^T B h_2 = 0$$
$$h_1^T B h_1 - h_2^T B h_2 = 0$$

Writing above equations in matrix form:

$$V\vec{b} = \vec{0}$$

where, V is given by:

$$V = \begin{bmatrix} \vec{v}_{12}^T \\ \vec{v}_{11}^T - \vec{v}_{22}^T \end{bmatrix}$$

again, $\vec{v}_{ij}$ is a column vector, composed of elements of H matrix, given by:

$$\vec{v}_{ij} = \begin{bmatrix} h_{1i}h_{1j} \\ h_{1i}h_{2j} + h_{2i}h_{1j} \\ h_{3i}h_{1j} + h_{1i}h_{3j} \\ h_{2i}h_{2j} \\ h_{3i}h_{2j} + h_{2i}h_{3j} \\ h_{3i}h_{3j} \end{bmatrix}$$

and, $\vec{b} = [b_{11}, b_{12}, b_{13}, b_{22}, b_{23}, b_{33}]^T$

As $\vec{b}$ is a symmetric matrix, we only consider the 6 upper triangular elements. Now, we can solve the equation $V\vec{b} = \vec{0}$, using SVD decomposition of V to obtain $\vec{b}$.

After reshaping $\vec{b}$ into a 3×3 matrix, we could get original B. As B is symmetric matrix, it cholesky decomposition would give K:

$$AA^T = choleskyDecomp(B)$$

As we assumed before, that $B = K^{-T}K^{-1}$, then A would be:

$$A = K^{-T}$$
$$\text{so, } K = A^{-T}$$

2.3   Finding camera's Pose, R and T

Now we know homography, H and camera matrix, K. Next we want to find out the pose, R and t of camera. So, We have:

$$H = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$
$$K^{-1}H = \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

Let, $K^{-1}H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$, then ideally we would have rotation matrix M and translation vector t, given by:

$$M = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix}, \text{where,}$$
$$m_1 = \frac{h_1}{\|h_1\|_2}$$
$$m_2 = \frac{h_2}{\|h_1\|_2}$$
$$m_3 = m_1 \times m_2$$
$$\text{and,}$$
$$t = \frac{h_3}{\|h_1\|}$$

But for a matrix to be a rotation matrix, its column should be orthonormal and its determinant should be 1. However there is no condition on a vector to be a translational vector. So we could accept t as translation of checkerboard from camera. But we need to find a matrix, R close to current matrix, M

and should also follows rotation matrix property, i.e. $R \in SO(3)$.
We could find such R, by solving:

$$\underset{R \in SO(3)}{argmin} \|R - M\|_F^2$$

The solution is given by SVD of matrix, M as:

$$USV^T = SVD(M)$$

Then matrix R would be:

$$R = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(UV^T) \end{bmatrix} V^T$$

By approximating S matrix to the above form, we are ensuring $R \in SO(3)$, i.e., diagonal matrix is ensuring that $det(R) = 1$, while product of U and V^T, is ensuring that resulting matrix has orthogonal columns.

References

[1] An Invitation to 3-D Vision: From Images to Geometric Models, Jana Koeck et al.

[2] Lecture: Cyrill Stachniss, Photogrammetry I, YouTube.

[3] Lecture: Kostas Daniilidis, CIS 390, Robotics: Planning and Perception.