Pavan Vishwanath

Developement of a Robotic Data Acquisition Setup for Camera Based Electrode Digitalization

Institute of Medical Technology
Building E | 21073 Hamburg
www.tuhh.de/mtec

A project paper written at the Institute of Medical Technology and submitted in partial fulfillment of the requirements for the degree Master of Science.

Author: Pavan Vishwanath

Title: Developement of a Robotic Data Acquisition Setup for Camera Based Electrode Digitalization

Date: September 30, 2020

Supervisors: Martin Gromniak (MSc.)
Alexander Schlaefer (Dr.-Ing.)

Referees: Prof. Dr.-Ing. Alexander Schlaefer
Martin Gromniak (MSc.)

1 Background

1.1 Camera fundamenals

A gentle introduction to the pinhole camera model, projection matrix, calibration (internal and external), the underlying mathematics.

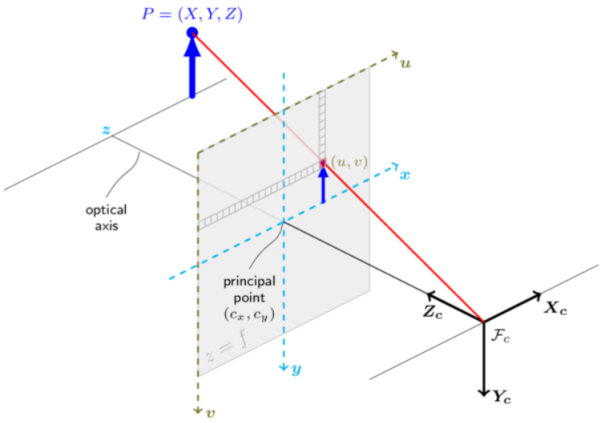

Fig. 1.1: Pinhole camera geometry. $\boldsymbol{F_c}$ *is the camera center,* (C_x, C_y) *is the principal point,* $\boldsymbol{P}$ $(X, Y, Z)^T$ *is world point,* $\boldsymbol{P_c}$ $(X_c, Y_c, Z_c)^T$ *is world point measured in camera coordinate system.[1]*

A simplified perspective projection is shown in fig. 1.1. A world point $(X_c, Y_c, Z_c)^T$ is mapped to $(f\frac{X_c}{Z_c}, f\frac{Y_c}{Z_c}, f)^T$ on the image plane placed at a focal length distance f from the camera center (F_c). Two key facts can be derived from the above projection equation that farther the object is (larger the Z_c) from the camera, smaller the size in image plane (shrinking operation) and these shrunk points are magnified by the focal length f to be placed on the image plane. World points are first shrunk $(x, y)^T = (\frac{X_c}{Z_c}, \frac{Y_c}{Z_c})^T$ and then magnified $(f\frac{X_c}{Z_c}, f\frac{Y_c}{Z_c}, f)^T$.

Often, the world points are represented in homogeneous coordinates due to several advantages, being able to express infinite quantities is one of them. The linear mapping between world and image points is evident in the homogeneous representation as shown below.

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{1.1}$$

The linear mapping can be expressed in a compact form $\boldsymbol{x} = \mathtt{P}\boldsymbol{X_c}$ where $\boldsymbol{x}$ is a vector of image points, $\boldsymbol{X}$ is a vector of world points and $\mathtt{P}$ is a 3x4 homogeneous matrix called

camera projection matrix. In general, the origin of the image plane may not coincide with the principal point, therefore it is necessary to map the projected points to pixels before using the image for further use as show in fig. 1.2.

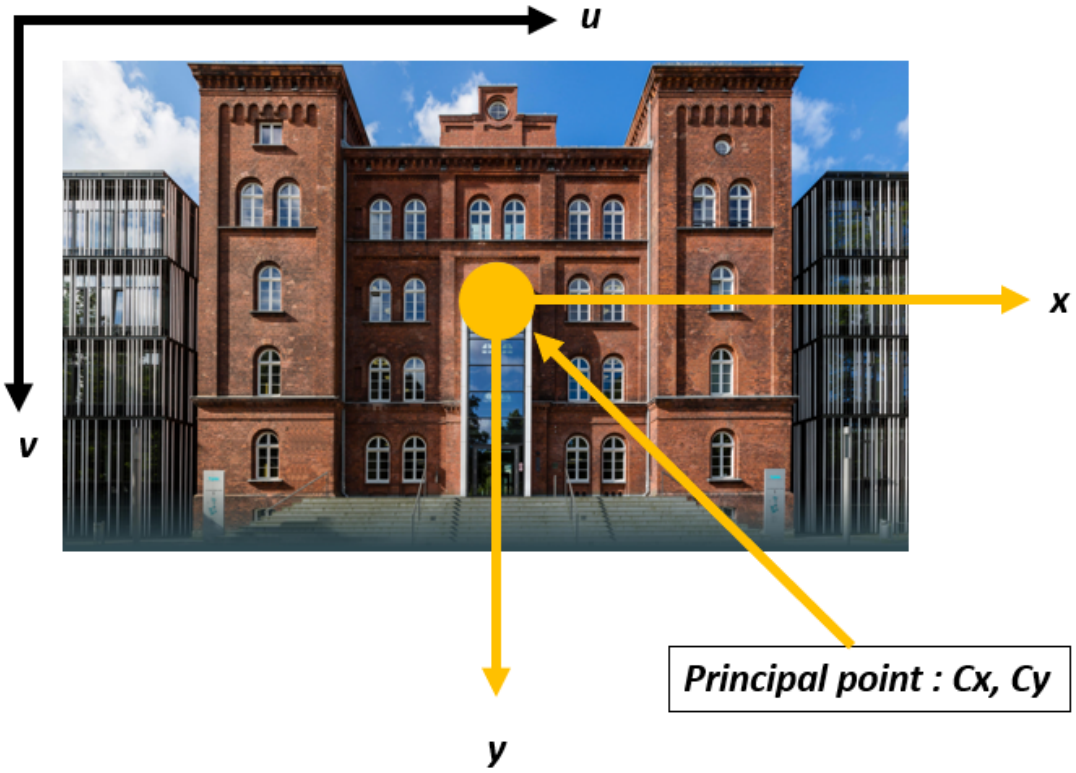

Fig. 1.2: Pixel coordinates (u, v) and the camera coordinates (x, y).

The equation 1.2 depicts mapping image to pixel coordinates. wehere (S_x) is size of pixel width and (S_y) is size of pixel height.

$$\begin{aligned}(u - C_x) &= \frac{x}{S_x} \\ (v - C_y) &= \frac{y}{S_y}\end{aligned} \tag{1.2}$$

Therefore a 3D world point $(X_c, Y_c, Z_C)^T$ is mapped to $(f\frac{X}{Z} + C_x, f\frac{Y}{Z} + C_y, f)^T$ to the pixel coordinates (u, v, z) and can be expressed as a matrix multiplication. Where α_x is $(\frac{f}{S_x})$, α_y is $(\frac{f}{S_y})$ and S' is slant factor, when the image plane is not normal to the optical axis.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{S_x} & S' & C_x \\ 0 & \frac{1}{S_y} & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1.3}$$

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & C_x \\ 0 & \alpha_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{1.4}$$

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} K_{3\times3} \end{bmatrix} \begin{bmatrix} I_{3\times3}|0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{1.5}$$

In summary, the equation 1.1 maps the world point measured in camera coordinates to image coordinates while equation 1.3 converts these image coordinates to pixel coordinates. Overall mapping from camera coordinates to the pixel coordinates is depicted in the equation 1.4. K in equation 1.5 is known as the camera intrinsic matrix and depends only on the internals of the camera. In general, a 3D world point $(X, Y, Z)^T$ may not be known in the camera coordinate system however it can be mapped using 4×4 homogeneous transformation matrix. equation 1.6 is a mapping from a 3D world point to pixel coordinates. The 4×4 homogeneous transformation matrix is known as an extrinsic matrix and describes the position and orientation of the 3D world point in the camera coordinate system.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & C_x \\ 0 & \alpha_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.6}$$

The equation 1.6 can be compactly written by combining both intrinsic and extrinsic matrix known as the projection matrix P_{3X4}. Where P = K[R|t], and λ is a arbitrary scaling factor for which equation 1.7 is satisfied.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & r_{22} & r_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.7}$$

We extract the camera center from the projection matrix as the camera center C = $-R^{-1}t$ or in other words, $t = -RC$. Therefore the projection matrix can be written as,

$$\begin{aligned} \mathtt{P} &= \mathtt{K}\,[\mathtt{R} \mid \boldsymbol{t}] \\ &= \mathtt{KR}\,[\mathtt{I} \mid -\boldsymbol{C}] \\ &= \mathtt{M}\,[\mathtt{I} \mid \mathtt{M}^{-1}\ p_4] \end{aligned} \tag{1.8}$$

where M = KR, K is a 3x3 upper triangular camera matrix, R is a 3x3 rotation matrix and p_4 is the last column of the projection matrix.

1.2 Camera calibration

Often in practice, estimating intrinsic and extrinsic parameters are important and there are many ways to do so. Two approaches will be presented here, one using projection matrix from equation 1.7 (2D/3D correspondence) and using homography (2D/2D correspondence).

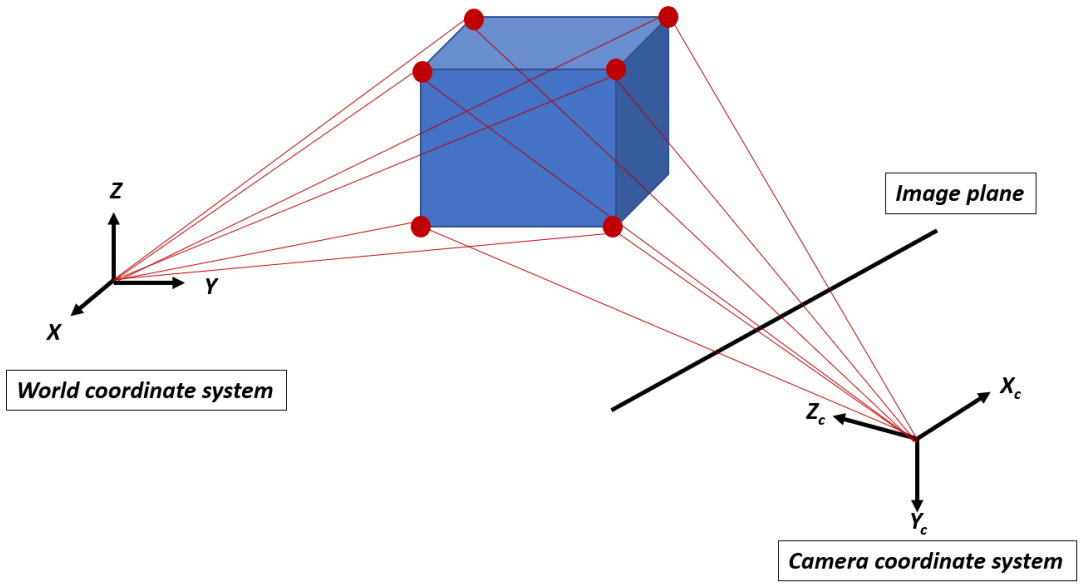

Fig. 1.3: Mapping 3D world points on to a 2D image plane.

1.2.1 2D/3D correspondence

The fig. 1.4 depicts a 2D image of a 3D object with known 6 unique points. Recall the equation 1.7 projection matrix P maps the world point to pixel coordinates and the equation is valid for arbitrary scaling factor λ. First, the projection matrix P will be estimated using a given set of 3D world points and 2D image points and then it will be decomposed to intrinsic and extrinsic matrices as P = K[R|t]. The first step in estimating projection matrix is to convert the equation 1.7 into least square problem of type II (see appendix) Ap = 0 subjected to $\|p\| = 1$ and solve for vector $\boldsymbol{p}$ which is reshaped version of non-trivial elements of the projection matrix P.

Given a set of N corresponding 2D/3D points (u_i,X_i), a projection matrix P is needed such that

$$\lambda u_i = PX_i \ , where \ i = 1.....N$$

since the scaling factor λ is unknown and has to be estimated while estimating P. we will make use of DLT (direct linear transformation) to convert the above equation to the form Ap = 0.

$$\begin{bmatrix} \lambda u_i \\ \lambda v_i \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & r_{22} & r_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

extracting the bottom row, equation for λ and substituting λ in other 2 rows yileds

$$\lambda = p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}$$

$$u_i\lambda = u_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}$$
$$v_i\lambda = v_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}$$

Rearrangement of the above equations leads to a linear system of equations with non-trivial elements of the projection matrix being reshaped into a vector $\boldsymbol{p}$.

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \mathbf{p} = \mathbf{0} \tag{1.9}$$

$$with\ \mathbf{p} = \Big(p_{11}, p_{12}.....p_{33}, p_{34}\Big)^T \in \mathcal{R}^{12}$$

A projection matrix has 12 non-trivial elements thus 11 degrees of freedom (ignoring scaling) therefore it is necessary to have 11 equations to solve for P. Each pair of 2D/3D point correspondences leads to 2 equations thus a minimum of 6 2D/3D point correspondences are required. Stacking these 6 equations along rows leads to a linear system of equations Ap=0 as shown in eq. (1.10).

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_6 & Y_6 & Z_6 & 1 & 0 & 0 & 0 & 0 & -u_6X_6 & -u_6Y_6 & -u_6Z_6 & -u_6 \\ 0 & 0 & 0 & 0 & X_6 & Y_6 & Z_6 & 1 & -v_6X_i & -v_6Y_6 & -v_6Z_6 & -v_6 \end{bmatrix} \mathbf{p} = \mathbf{0} \tag{1.10}$$

Exact solution. With a minimum of 6 correspondences, the solution to 1.10 will be exact which means 3D world points will be exactly gets projected to their measured image points correspondingly. The exact solution p, to $\mathrm{Ap} = 0$ is the right nullspace of matrix A.

Over-determined solution. Practical measurements will be noisy due to various reasons and therefore we may require more than 6 2D/3D correspondences. In this case, the solution to $\mathrm{Ap} = 0$ will be obtained by minimizing the algebraic or geometric error of projection, subjected to some valid constraints.

Minimizing algebraic error. In this case the approach is,

$$\min \quad \|Ap\| \quad \text{s.t.} \quad \|p\| = 1 \tag{1.11}$$

The solution to eq. (1.11) is obtained from unit singular value of A corresponding to the smallest singular value (the least square problem of type II, see appendix).

Minimizing geometric error. First let us define what is geometric error. Let us recall eq. (1.7), $u_i = \mathrm{P}X_i$, suppose we know 3D world points X_i far more accurately than the measured image points then the geometric error in the image is

$$\sum_{i=1}^{n} d(u_i, \hat{u}_i)^2$$

where u_i is measured point in the image and $\hat{u}_i$ is P X_i which is the exact projection of X_i on to the image under P. If the measurement errors are Gaussian then the solution to

$$\min_P \sum_{i=1}^{n} d(u_i, PX_i)^2 \tag{1.12}$$

is the maximum likelihood of P as per [2]. Minimizing geometric error requires non-linear iterative methods such as Levenberg-Marquardt (LM) algorithms. Local minima can be found via LM, in order to find the global minima, the initial starting point can be linear solution obtained from eq. (1.11).

Having estimated projection matrix task at the hand is to decompose $P = K[R \mid t] = M[I \mid M^{-1}\ p_4]$ where $M = KR$. Decomposing M to K and R can be achieved using RQ decomposition with the constraint that diagonal entries of the K matrix has to be positive.

1.2.2 2D/2D correspondence

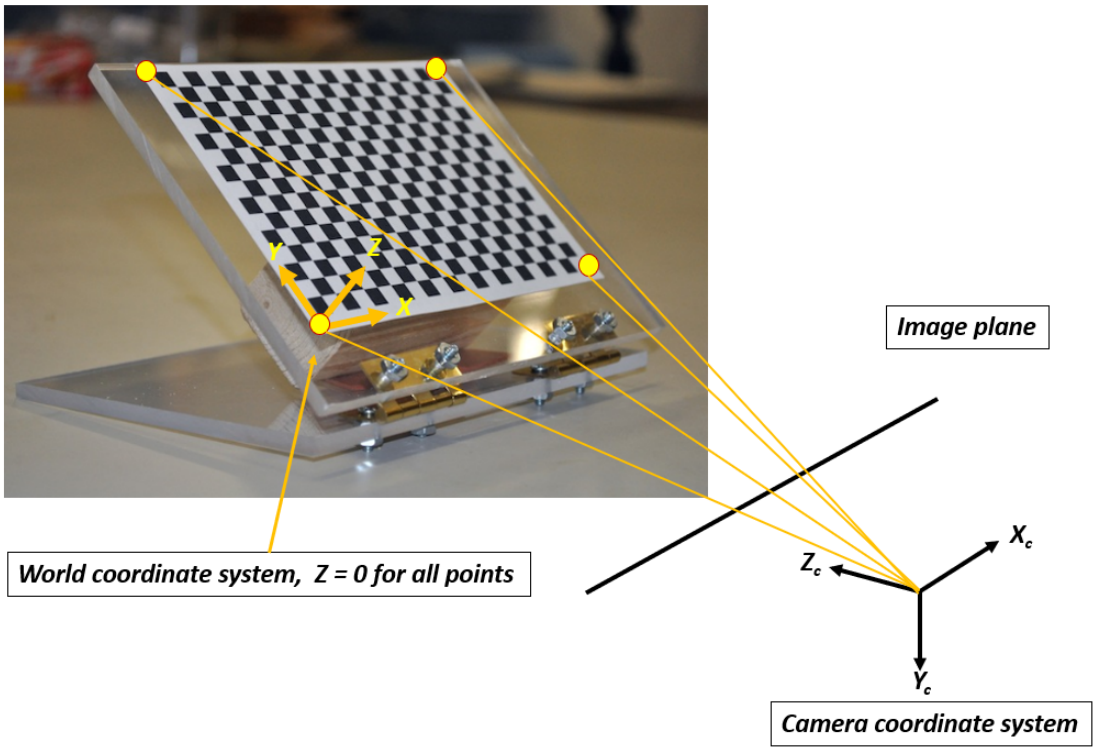

Fig. 1.4: Mapping 2D planar world points (Z = 0) on to a 2D image plane.

The disadvantage of 2D/3D correspondence way of estimating the projection matrix is that complete knowledge of the 3D location has to be known and it has to be precise as well. There is a flexible and computationally easy method to estimate the projection matrix thereby intrinsic and extrinsic camera parameters, developed by Zhang [3] using a 2D planar object in 3D space as shown in fig. 1.4 along with its 2D image. Let us recall the equation 1.6

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & C_x \\ 0 & \alpha_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Using 2D planar object we eliminate Z coordinate thereby eliminating 3^{rd} column of the extrinsic matrix.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & C_x \\ 0 & \alpha_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & C_x \\ 0 & \alpha_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{1.13}$$

As in the case of 2D/3D problem setup, The equation 1.13 can be compactly written by combining both intrinsic and extrinsic matrices known an Homography matrix $H_{3\times3}$. And λ is a arbitrary scaling factor for which equation 1.14 is satisfied. In other words, A projection matrix $P_{3\times4}$ in planar case reduces to homography matrix $H_{3\times3}$.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{1.14}$$

As in the case of 2D/3D problem, the homography matrix H will be estimated using a given set of 2D planar world points and 2D image points and then it will be decomposed to intrinsic and extrinsic matrices.

Given a set of N corresponding 2D/2D points (u_i,X_i), a homography matrix H is needed such that

$$\lambda u_i = HX_i \ , where \ i = 1.....N$$

Problem set up in 2D/2D case is very similar to 2D/3D problem except that the homography matrix has 9 non-trivial elements thus 8 degrees of freedom (ignoring scaling) therefore 8 independent equations are necessary to estimate H, hence 4 2D/2D point correspondences are required. At this point, the same estimating procedure used previously can be employed.

Exact solution. With a minimum of 4 correspondences, the solution to $Ah = 0$ will be exact which means 2D world points will be exactly gets projected to their measured image points correspondingly. The exact solution h, to Ah = 0 is the right nullspace of matrix A.

Over-determined solution. In this case solution to Ah = 0 will be obtained by minimizing the algebraic or geometric error of projection, subjected to some valid constraints.

Minimizing algebraic error. In this case the approach is,

$$\min \quad ||Ah|| \quad \text{s.t.} \quad ||h|| = 1 \tag{1.15}$$

The solution to eq. (1.15) is obtained from unit singular value of A corresponding to the smallest singular value (the least square problem of type II, see appendix).

Minimizing geometric error The geometric error in the image in 2D/2D case is

$$\sum_{i=1}^{n} d(u_i, \hat{u}_i)^2$$

where u_i is measured point in image and $\hat{u}_i$ is H X_i which is the exact projrction of X_i on to the image under H. If the measurement errors are Gaussian then the solution to

$$\min_{P} \sum_{i=1}^{n} d(u_i, HX_i)^2 \tag{1.16}$$

is the maximum likelihood of H. Minimizing geometric error requires non-linear iterative methods such as Levenberg-Marquardt (LM) algorithms. Local minima can be found via LM, in order to find the global minima, initial starting point can be linear soution obtained from 1.11.

The disadvantage of 2D/2D homography is that with a single homography, not all the parameters of a projection matrix can be determined as Z coordinate of the world point is eliminated.

$$\mathbf{H} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_4 \end{bmatrix}$$

The decomposition of H to obtain extrinsic and intrinsic matrices is not possible, therefore, follows a different approach. We will first see how to get the intrinsic matrix K and as soon we have K, obtaining extrinsic is trivial.

Intrinsics Recall that $H = K\ [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3]$ and the fact that R is a rotational matrix and can be expressed as,

$$\begin{aligned} \mathbf{r}_1^\intercal \mathbf{r}_2 &= 0 \\ \mathbf{r}_1^\intercal \mathbf{r}_1 = \mathbf{r}_2^\intercal \mathbf{r}_2 &= 1 \end{aligned}$$

rewriting with K gives,

$$\begin{aligned} \mathbf{p}_1^T K^{-T} K^{-1} \mathbf{p}_2 &= 0 \\ \mathbf{p}_1^T K^{-T} K^{-1} \mathbf{p}_1 = \mathbf{p}_2^T K^{-T} K^{-1} \mathbf{p}_2 &= 1 \end{aligned}$$

defining $\omega = K^{-T} K^{-1}$ as a symmetric matrix, above equations can be written as,

$$\omega = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix}$$

$$\begin{aligned} \mathbf{p}_1^T \omega \mathbf{p}_2 &= 0 \\ \mathbf{p}_1^T \omega \mathbf{p}_1 - \mathbf{p}_2^T \omega \mathbf{p}_2 &= 0 \end{aligned} \tag{1.17}$$

$\mathbf{p}_1$ and $\mathbf{p}_2$ are known from homography matrix and $\boldsymbol{\omega}$ has to be calculated. $\boldsymbol{\omega}$ can be estimated with techniques that we employed earlier with DLT and SVD by defining $\mathbf{b} = (\omega_{11}\ \omega_{12}\ \omega_{13}\ \omega_{22}\ \omega_{23}\ \omega_{33})^T$ and solving $\mathbf{Ab} = 0$. Each homography provides 2 independent rows therfore 3 such homographies are required to estimate $\boldsymbol{\omega}$. Once the $\boldsymbol{\omega}$

is computed it can be decomposed in to $K^{-T}K^{-1}$ using Cholesky decomposition.

Extrinsics. Computing extrinsics post intrinsics is very straight forward as given by [3] we will first calculate rotation matrix followed by translation vector. $H = K\ [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]$ having known K we can rearrange as $K^{-1}\ H = H^{'} = [\mathbf{h}_1^{'}\ \mathbf{h}_2^{'}\ \mathbf{h}_3^{'}]$ which is theoritically equal to $[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]$. Since the rotational matrix is a orthogonal matrix, third column is othogonal to first 2 and therefore can be calculated by vector cross product $[\mathbf{h}_1^{'} \times \mathbf{h}_2^{'}]$ hence, the complete rotational matrix is $Q = [\mathbf{h}_1^{'}\ \mathbf{h}_2^{'}\ \mathbf{h}_1^{'} \times\ \mathbf{h}_2^{'}]$ again which is theoritically equal to $R = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_1\ \times\ \mathbf{r}_2\ ]$ which may not be true due to the noisy data. Therefore, it necessary to find a best 3x3 rotation matrix R from a given 3x3 matrix Q, here "best" in the sense of smallest Frobenius norm of the difference $[R - Q]$ (see appendix). If the SVD of given matrix is $Q = UDV^T$, then the best rotation matrix is given by UV^T. The translation vector can be calculated as $\mathbf{t} = \mathbf{h}_3^{'}$ but the vector $\mathbf{t}$ has to be normalised therefore $\mathbf{t} = \mathbf{h}_3^{'}/\|\mathbf{h}_1^{'}\|$.

1.2.3 Lens distortions

A world point is mapped to image plane as $(x, y, z)^T = (f\frac{X_c}{Z_c}, f\frac{Y_c}{Z_c}, f)^T$ then image points are mapped to pixel coordinates as $(u, v, z)^T = (x+C_x, y+C_y, f)^T$ however, usually, there are radial and tangential distortions in the lens as depicted in fig. 1.5 therefore, image coordinates have to be corrected before mapping to pixel coordinates. The corrected image coordinates are

$$\begin{aligned} x^{'} &= x \times \frac{1 + k_1r^2 + k_2r^4 + k_3r^6}{1 + k_4r^2 + k_5r^4 + k_6r^6} + 2p_1xy + p_2(r^2 + 2x^2) \\ y^{'} &= y \times \frac{1 + k_1r^2 + k_2r^4 + k_3r^6}{1 + k_4r^2 + k_5r^4 + k_6r^6} + p_1(r^2 + 2y^2) + 2p_2xy \end{aligned} \tag{1.18}$$

where $r^2 = (x^2 + y^2)$, $k_1.....k_6$ are the radial distortion coefficients and p_1, p_2 are the tangential distortion coefficients. It is these corrected image coordinates are mapped to pixel coordinates as $(u, v, z)^T = (x^{'} + C_x, y^{'} + C_y, f)^T$.

Fig. 1.5: Barrel distortion ($typically\ k_1 > 0$) and pincushion distortion ($typicallyk_1 < 0$) [1]

Camera calibration is the process of determining, intrinsic, extrinsic parameters along with distortion coefficients. The tab. 1.1 summarizes the calibration process discussed above.

Tab. 1.1: Calibration summary table

correspondences	Min. points	No.of images	[R | t]	K	world points
2D im./3D world	≥ 6	1	✓	✓	given
2D im./2D world	≥ 4	3	✓	✓	given
2D im./2D world	≥ 4	1	✓	given	given

1.3 Camera calibration and pose estimation

Camera calibration is carried out using open-source computer vision library OpenCV [1]. Camera calibration is based on 2D/2D point correspondences with the chessboard as a 2D planar object. The homography is calculated using square corners of a chessboard (world points) and its image points. OpenCV needs an arrays of world points and image points and the grid size of the chessboard (in our case its 5 rows, 8 columns). Therefore, 10 RGB images of the chessboard at different position and orientation was recorded and an array of world points (x,y) location of chessboard corners [(0,0), (40,0), (80,0)...] was fed to the algorithm. OpenCV automatically detects these chessboard corners from the images as shown in fig. 1.6 and refines them accordingly.

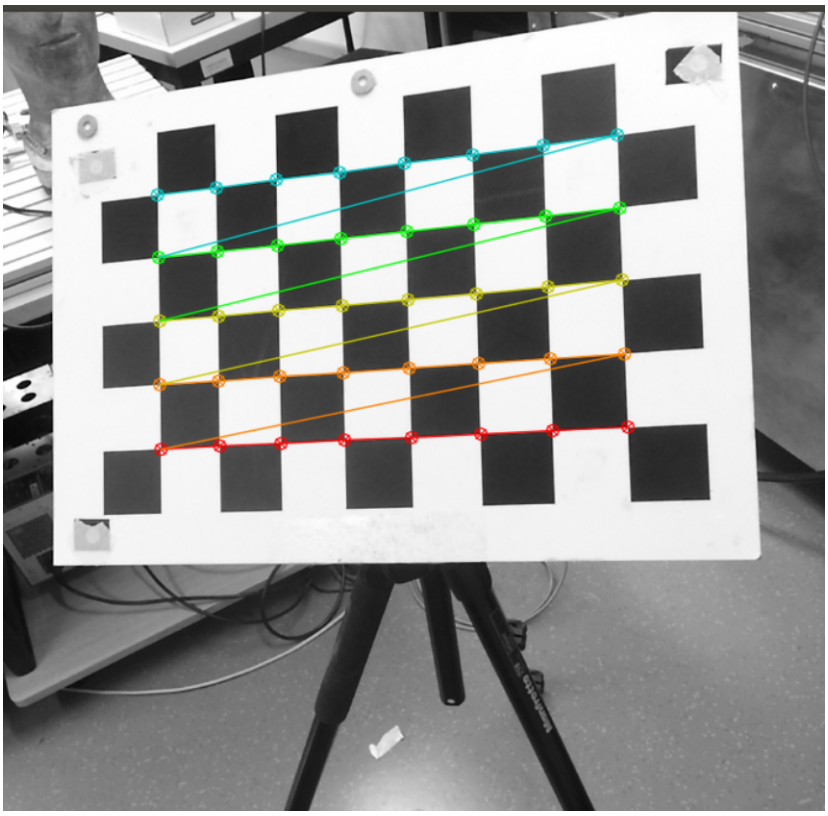

Fig. 1.6: Chessboard corners(image points) detection in OpenCV

```
cv2.calibrateCamera(object_points, image_points, ...)
output: rms, camera_matrix, dist_coeffs, rot_vecs, trans_vecs
```

The OpenCV function cv2.calibrateCamera takes in these world points (object points), image points and some more arguements then outputs geometric error of reprojection (rms), intrinsic parameters (camera_matrix) distortion coefficients (dist_coeffs) and extrinsic parameters ($rot\_vecs, trans\_vecs$).

Having completed the camera calibration we can now make use of the intrinsic parameters and the distortion coefficients as an input to the pose estimation algorithm provided by OpenCV.

```
cv2.solvePnP(object_points, image_points, intr_mat, dist_coeffs)
output: rot_vecs, trans_vecs
```

The OpenCV function cv2.solvePnP takes object points, image points, intrinsic matrix, and distortion coeffcients as the arguments and computes rotation and translation vectors. Rotation vector can be converted to 3×3 rotational matrix using cv2.Rodrigues function provided by openCV. By combining rotation matrix and translation vector we can form 4×4 homogenious matrix for further manipulation.

Bibliography

[1] opencv.org. (1999). "MS Windows NT kernel description," [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html (visited on 05/20/2020).

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003, ISBN: 0521540518.

[3] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000, ISSN: 0162-8828. DOI: 10.1109/34.888718.

Declaration

Hiermit versichere ich, dass ich meine Abschlussarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum: .....

(Unterschrift)