# Camera Calibration using Zhang's Technique[1]

Belal Hmedan[1] and Thien Bao Bui[2]

***Abstract*— Camera calibration is fundamental in computer vision science, since it highlights the core concepts of imaging and moving from 3D to 2D space. This paper discusses Zhang's calibration procedure which is composed of a closed-form solution and a nonlinear refinement based on the maximum likelihood criterion. Compared with classical techniques, Zhang's technique is considerably more flexible. Also it has considerable degree of robustness compared to self-calibration[1].**

## I. INTRODUCTION

**Motivation:** The development of Zhang's calibration method is aimed at providing a flexible, low cost and robust solution for desktop vision system. These criteria, on the contrary, are not satisfied in the other approaches. The protogrammetric calibration requires an expensive equipment setup, while self-calibration seems to be flexible enough but still immature, or difficult to initialize (Bill Triggs self-calibration). This method eventually becomes the simplest and efficient technique used in the field of computer vision for camera calibration.[2]. Camera calibration means finding the quantities internal and external to the camera that affects the following imaging properties:

1) Internal Parameters (intrinsic)
   a) Image center (principal point) $(x_0, y_0)$
   b) Focal length $(f)$
   c) Lens distortion parameters $(\check{u_0}, \check{v_0})$
2) External Parameters (extrinsic)
   a) Rotation matrix
   b) Translation vector

Precise calibration is required for: 3D interpretation of images, reconstruction of world models, robot interaction with the world, and wide range of applications. The paper is organized as follows. Section 2 describes the mathematical model, and the development of the model. starting with a analytical solution, followed by nonlinear optimization. Radial and tangential lens distortion, are also modeled. Section 3 describes the experimental work. Section 4 summarizes the conclusion of the research. In the Appendix provides a brief matrix transformation explanation.

## II. MATHEMATICAL MODEL

**Projection:** using the pinhole camera model, projection of 3D point $M_i = [X_i, Y_i, Z_i]$ to the image plane $[u_i \quad v_i]$ is:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{f}{Z_i} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \tag{1}$$

Notation: Augmented (homogeneous) 2D point will be denoted by: $\tilde{m} = [u \quad v \quad 1]^\intercal$, while Augmented 3D point is denoted by: $\tilde{M} = [X \quad Y \quad Z \quad 1]^\intercal$][2], the projection between 3D space, and 2D image on the pinhole camera is given by the equation:

$$s\tilde{m} = K[R \mid t]\tilde{M}, \quad with \quad K = \begin{bmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

[1]Belal is a Master degree student in Computer Vision and Robotics(VIBOT), University of Burgundy, 720 Avenue de l'Europe
Email: Belal_Hmedan@etu.u-bourgogne.fr

[2]Bao is a Master degree student in Computer Vision and Robotics(VIBOT), University of Burgundy, 720 Avenue de l'Europe
Email: Thien-Bao_Bui@etu.u-bourgogne.fr

with $\alpha = k_u f$, $\beta = k_v f$.

$K$ is the camera intrinsic matrix, $f$ is the focal length, $(k_u, k_v)$ magnification factors: the number of pixels per unit distance in $u$ and $v$ directions. $s$ is an arbitrary scaling factor $c_0 = [u_0 \quad v_0]^\intercal$ is called the principal point, usually the coordinates of the image center. $c$ is the skew, only non-zero if $u$ and $v$ are non-perpendicular. A camera is calibrated when intrinsic and extrinsic matrix is known. Computer-vision applications focuses on estimating the camera extrinsic matrix[3], where extrinsic matrix $[R \mid t]$ is a 3×4 matrix (*also called camera pose*), that corresponds to the euclidean transformation from a world coordinate system to the camera coordinate system. R represents a 3×3 rotation matrix and t a 3×1 translation vector.

$$[R \mid t] = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \end{bmatrix} = [R_1 \quad R_2 \quad R_3 \quad t]$$

***Homography:***

Without loss of generality, assuming the model plane is on $Z = 0$ of the world coordinate system, hence dropping the $Z$ axis and third column in rotation matrix through the calculations[1].

$$H = [h_1 \quad h_2 \quad h_3] = K[R_1 \quad R_2 \quad t] \tag{3}$$

hence: $R_1 = K^{-1}.h_1, \quad R_2 = K^{-1}.h_2$.
Knowing that $R_1 \perp R_2$, orthogonality of $R_1$ and $R_2$ implies $R_1^\intercal R_2 = 0$, hence:

$$h_1^\intercal K^{-\intercal} K^{-1} h_2 = 0 \tag{4}$$

orthogonality also implies $\|R_1\| = \|R_2\| = 1$ hence: $R_1^\intercal R_1 = R_2^\intercal R_2 = 1$

$$h_1^\intercal K^{-\intercal} K^{-1} h_1 = h_2^\intercal K^{-\intercal} K^{-1} h_2 \tag{5}$$

Homography has 8 degrees of freedom and there are 6 extrinsic parameters (Pitch, Yaw, Roll for rotation and $X$,$Y$,$Z$ for translation), resulting 2 constraints on the intrinsic parameters[1].

***Analytical Solution:***

The idea to compute these parameters is to solve the equations given by each pair of correspondent 2D and 3D points. However, to calculate these parameters directly from this mapping equation (2D to 3D):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & -Rt \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{bmatrix} \tag{6}$$

is difficult since the unknowns are multiplied with one another. Thus, the calculation is divided in two stage. First, the intrinsic parameters are computed. Second, once this done, the extrinsic parameters will be extracted. We are going to calculate the intrinsic parameters. Let

$$B = K^{-T} K^{-1} \Rightarrow B^T = B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

B is symmetric, and is defined by a 6D vector:

$$b = [B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33}]^T$$

Let the $i^{th}$ column of $H$ be $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$. We have:

$$h_i^T B h_j = v_{ij}^T b$$

See appendix A for details of the transformation.
Then, from equation (4) and (5), which are known as the constraint on the intrinsic parameters, we can deduce:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \tag{7}$$

If $n$ images of model plane are observed, by stacking several equations (7), we have:

$$Vb = 0 \tag{8}$$

where $V$ is $2n{\times}6$ matrix.
Clearly, solving the equation (8) gives us the values of matrix B, which leads to the computation of matrix K - the intrinsic parameters. Equation (8) is solved by using Singular Value Decomposition. If $n \geqslant 3$ , the equation has an unique solution $b$ in general defined up to a scale factor. If $n = 2$, the number of equation is smaller than the number of unknowns. Therefore, we set the skewless constraint $c = 0$. This will add an extra equation to equation (8). If $n = 1$, similarly, we set the skewless constraint $c = 0$. Assuming the coordinate of principal point $u_0, v_0$ is already known, we can find the value of $\alpha = k_u f$ , $\beta = k_v f$. The value of matrix B is determined, hence we can calculate the intrinsic parameters. See the appendix B for details.
With the value of the intrinsic parameters, the value of extrinsic parameters is determined from equation (2):

$$\begin{cases} r_1 = \lambda A^{-1} h_1 \\ r_2 = \lambda A^{-1} h_2 \\ r_3 = r_1 r_2 \\ \;t = \lambda A^{-1} h_3 \end{cases}$$

with

$$\lambda = \frac{1}{\parallel A^{-1} h_1 \parallel} = \frac{1}{\parallel A^{-1} h_2 \parallel}$$

Due to noise of data, the matrix $R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$ generally is not a rotation matrix since it does not satisfy $R^T R = I$. The method to estimate the best rotation matrix from a general $3{\times}3$ matrix can be found in this technical report [2].

***Non-Linear Optimization:***

Ideally homography between model plane and it's image should satisfy (3), in practice that doesn't happen because of noise in the extracted images, so we try to estimate homography using maximum likelihood criterion. Given $n$ images with $m$ points on the model plane, assuming that the image points are corrupted by independent identically distributed noise[2], hence cost function can be written as following:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \|m_{ij} - \hat{m}(K, k, R_i, t_i, M_j)\|^2 \tag{9}$$

$\hat{m}$ is the projection of point $M_j$ on image $i$, while $k$ refers to lens distortion coefficients, minimizing (9) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm *(combines gradient decent and Newton's method)* It requires an initial guess of camera pose which can be obtained using the technique described in the analytical solution section[2].

***Lens Distortion:***

This section discusses the most common models of lens distortion:
***Radial Distortion:*** Conventional cameras usually have a significant lens distortion. Let $(u, v)$ be the ideal distortion-free pixel image coordinates, and $(\tilde{u}, \tilde{v})$ the corresponding real observed image coordinates. Similarly, $(x, y)$ and $(\tilde{x}, \tilde{y})$ are the distortion-free and distorted normalized image coordinates. We have:

$$\tilde{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

$$\tilde{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

, where $k_1, k_2$ are the coefficients of the radial distortion. The principal point and the center of radial distortion are the same. From $\tilde{u} = u_0 + \alpha\tilde{x} + c\tilde{y}$ and $\tilde{v} = v_0 + \beta\tilde{y}$, we have:

$$\tilde{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{10}$$

$$\tilde{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{11}$$

As the radial distortion is expected to be small, one can estimate the other five intrinsic parameters ignoring distortion. It gives us the ideal pixel coordinates $(u, v)$.
Then we have two equations for each point in each image:

$$\begin{bmatrix} (u - u_0)(x^2 + y^2) & (u - u_0)(x^2 + y^2)^2 \\ (v - v_0)(x^2 + y^2) & (v - v_0)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \tilde{u} - u \\ \tilde{v} - v \end{bmatrix}$$

Given $m$ points in $n$ images, we can stack all equations together to obtain in total $2mn$ equations, or in matrix form as $Dk = d$, where $k = [k_1, k_2]^T$. The linear least-squares solution is given by

$$k = (D^T D)^{-1} D^T d \tag{12}$$

Once $k_1, k_2$ are estimated, one can refine the estimate of the other parameters by solving (9) with $\hat{m}(A, R_i, t_i, M_j)$ replaced by (10) and (11). We can alternate these two procedure until convergence.
Experimentally, we found the convergence of the above alternation technique is slow. A natural extension to (9) is then to estimate the complete set of parameters by minimizing the following functional:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \parallel m_{ij} - \check{m}(A, k_1, k_2, R_i, t_i, M_j) \parallel^2, \tag{13}$$

where $\check{m}(A, k_1, k_2, R_i, t_i, M_j)$ is the projection of point $M_j$ in image $i$ according to equation (2), followed by distortion according to (10) and (11).
***Tangential Distortion:*** centers of curvature of lens surfaces are not always strictly collinear. This introduces another common distortion type, decentering distortion which has a tangential component. (*One of the first introduction of the tangential distortion models was "Brown-Conrady model"*)[4]. Tangential distortion often expressed by:[5]

$$\begin{bmatrix} \delta u_i^{(t)} \\ \delta v_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1\tilde{u}_i\tilde{v}_i + p_2(r_i^2 + 2\tilde{u}_i{}^2) \\ p_1(r_i^2 + 2\tilde{v}_i{}^2) + 2p_2\tilde{u}_i\tilde{v}_i \end{bmatrix} \tag{14}$$

$p_1$, $p_2$ are coefficients for tangential distortion, $\delta u_i^{(t)}, \delta v_i^{(t)}$ are the tangential distortion in $u$ and $v$ directions, $r_i^2 = \sqrt{\tilde{u}_i{}^2 + \tilde{v}_i{}^2}$, so total distortion will be :[5]

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u(\tilde{u}_i + \delta u_i{}^{(r)} + \delta u_i{}^{(t)}) \\ D_v(\tilde{v}_i + \delta v_i{}^{(r)} + \delta v_i{}^{(t)}) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{15}$$

In this model (15) the set of intrinsic parameters ($f$, $s_u$, $u_0$, $v_0$) is augmented with the distortion coefficients ($k_1$,..., $k_n$, $p_1$ and $p_2$). *To clear any confusion with radial coefficients $k_n$ the magnification factors are refereed here by: $D_u$, $D_v$*

## III. Experiments

***Omindirectional camera calibration:***

The following workflow is used to calibrate the camera using the Matlab toolbox app [6]:

1) Prepare images, camera, and calibration pattern.
2) Add images and select standard or fish-eye camera model.
3) Calibrate the camera.
4) Evaluate calibration accuracy.
5) Adjust parameters to improve accuracy (if necessary).
6) Export the parameters object.

The experiment is performed with the settings below.

TABLE I: Experiment settings

| Configuration | |
|---|---|
| 1st Camera lens | 27mm (wide) |
| 2nd Camera lens | 10.7mm |
| Image resolution | 3968×2976 |
| Size of pattern | 26×26 mm |
| Number of pattern squares | 7×9 |

It is necessary to rotate images from one snapshot to another in order to provide the additional constraint. The intrinsic parameters constraint (4) and (5) are derived from the properties of the rotation matrix. If the image at the second position is parallel to its first position, then it is not helpful for camera calibration. Following the procedure mentioned above we have got the following results in table(II) which represents the estimation of the parameters of the equipment used in this experiment.

TABLE II: Intrinsic Parameters

| Parameter | Value |
|---|---|
| Focal length(pixels) | 3267.1568 +/- 3.0691 |
| Principal point(pixels) | 2003.1297 +/- 1.5951 |
| Radial distortion | 0.1478 +/- 0.0031 |
| Tangential distortion | 0 |

It can be seen from the table that the distortion is only radial, whilst the tangential distortion is zero. Figure (1) illustrate the mean errors in re-projection estimation for images vs Gaussian noise. Re-projection errors provide a qualitative measure of accuracy. A re-projection error is the distance between a pattern key-point detected in a calibration image, and a corresponding world point projected into the same image. If the overall mean re-projection error is too high, usually we consider excluding the images with the highest error and re-calibrating[6].
In the next experiment, we want to examine the relation between noisy images and the error on the calculation of principal points. Therefore, Gaussian noise with 0 mean and $\sigma$ standard deviation is added to the projected image points(see Fig 3). The value of $\sigma$ is varied from 0.1 to 0.4 pixels . For each noise level, 3 different trials are performed, and the final result is by taking the average.

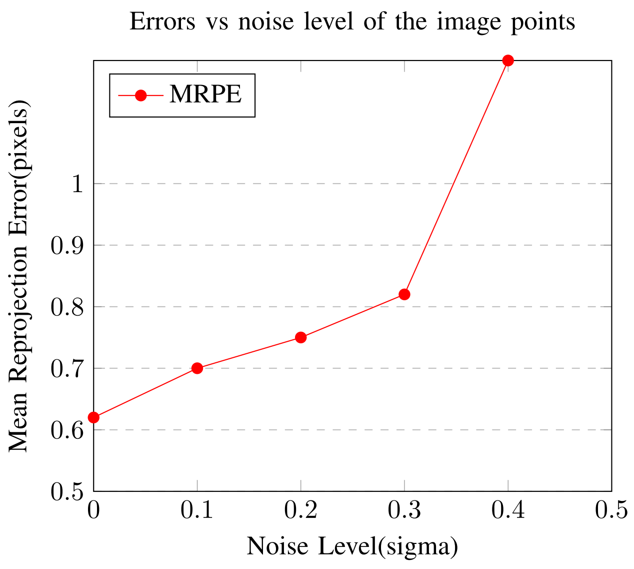

Fig. 1: Re-projection Mean Errors(pixels) vs noise.

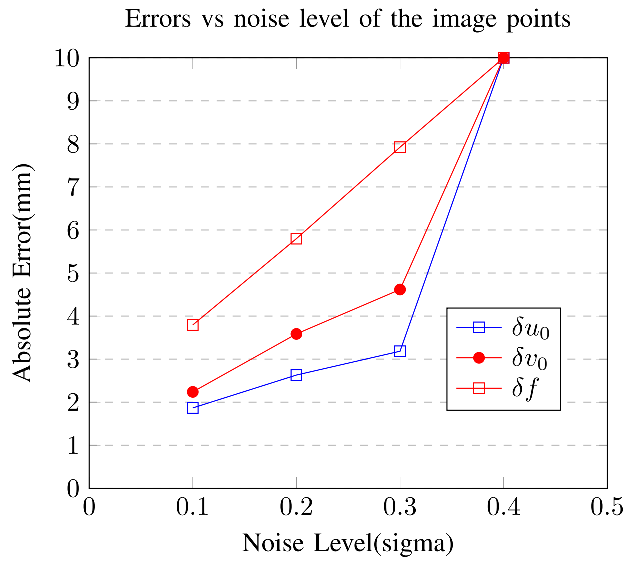

Fig. 2: Principal point, and focal length errors(mm) vs noise.

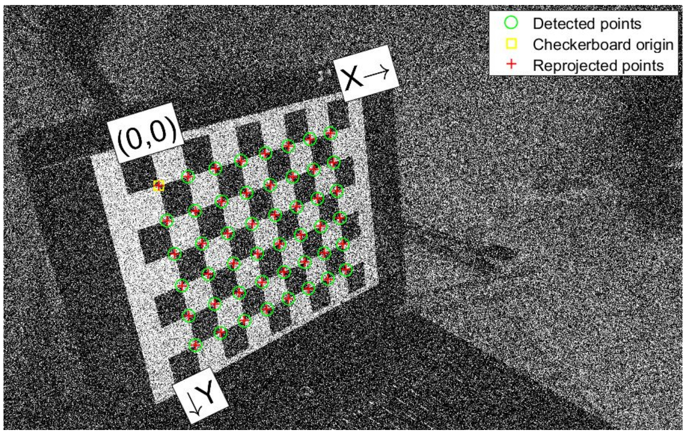

Fig. 3: Image with Gaussian noise $\sigma$=2, $\mu$= 0

The result is shown in the Figure (2).

## IV. Conclusions

Throughout this paper, we have re-implemented Zhang camera calibration technique [1]. No motion knowledge is required, however the planar pattern must be observed from at least two different orientations. A closed-form solution and a nonlinear refinement based on maximum likelihood function criterion are proposed. As we can see in the graph Fig(2), errors increase with noise level. The error in $u_0$ is smaller than the one in $v_0$ because we have more data in $v_0$ direction (9 squares) than in $u_0$ direction (7 squares) as demonstrated in the experiment settings. The algorithm is flexible, and easy to implement, but it's not robust against noise, as seen from the Figure (Fig 4) . The errors in estimation increase exponentially with noise(Fig 1),and (Fig 2). Distortion is also another important characteristic which affects calibration badly, so distortion removal is necessary (*in most of the cases*) before calibration. The size of the checkerboard has a small effect on the calibration, so it's better to be big enough for doing a precise calibration.

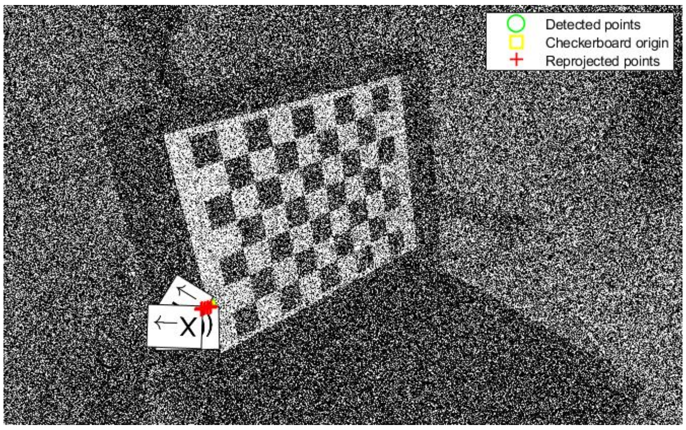

Fig. 4: Image with Gaussian noise $\sigma$=4, $\mu$= 0

## V. Source Code

Source code is publicly available at https://github.com/bhomaidan1990/Camera_Calibration.

## Appendix

### *Matrix transformation*

Here is the transformation to matrix of variable b.

$$h_i^T B h_j = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \begin{bmatrix} h_{j1} \\ h_{j2} \\ h_{j3} \end{bmatrix}$$

$$= \begin{bmatrix} h_{i1}B_{11}+h_{i2}B_{12}+h_{i3}B_{13} & h_{i1}B_{12}+h_{i2}B_{22}+h_{i3}B_{23} & h_{i1}B_{13}+h_{i2}B_{23}+h_{i3}B_{33} \end{bmatrix} \begin{bmatrix} h_{j1} \\ h_{j2} \\ h_{j3} \end{bmatrix}$$

$$= \begin{bmatrix} h_{i1}h_{j1} & h_{i1}h_{j2}+h_{i2}h_{j1} & h_{i3}h_{j1}+h_{i1}h_{j3} & h_{i2}h_{j2} & h_{i3}h_{j2}+h_{i2}h_{j3} & h_{i3}h_{j3} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \\ B_{13} \\ B_{22} \\ B_{23} \\ B_{33} \end{bmatrix}$$

$$= v_{ij}^T b$$

### A. *Intrinsic parameters extraction from matrix B*

From $B = \lambda K^{-T} K^T$, we can extract the intrinsic parameters:

$$\begin{cases} v_0 = \dfrac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \\ \lambda = B_{33} - \dfrac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \\ \alpha = \sqrt{\dfrac{\lambda}{B_{11}}} \\ \beta = \sqrt{\dfrac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}} \\ c = \dfrac{-B_{12}\alpha^2\beta}{\lambda} \\ u_0 = \dfrac{cv_0}{\beta} - \dfrac{B_{13}\alpha^2}{\lambda} \end{cases}$$

## References

[1] Zhengyou Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673 vol.1, 1999.
[2] 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.
[3] Jav_ROCK, "Step by step camera pose estimation for visual tracking and planar markers." https://dsp.stackexchange.com/questions/2736/step-by-step-camera-pose-estimation-for-visual-tracking-and-planar-markers/2737#2737, 2016. 2020-03-30.
[4] D. H. Brown, "Decentering distortion of lenses," in *Photogrammetric Engineering*, p. 444–462, 1966.
[5] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, 1997.
[6] mathworks, "Matlab computer vision toolbox camera calibration app." https://fr.mathworks.com/help/vision/ug/single-camera-calibrator-app.html, 2020. 2020-03-30.