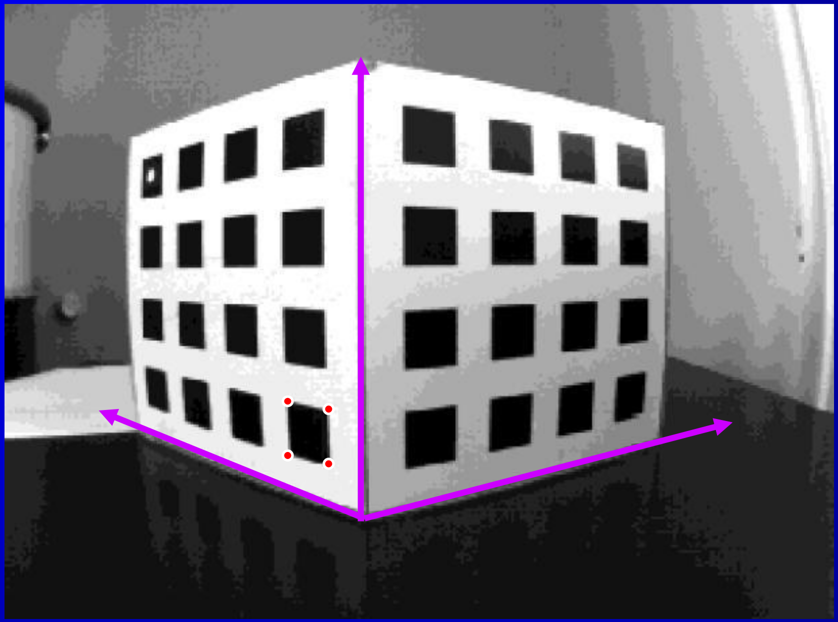
Computer Vision

---*Camera Calibration IV*

Dr. WU Xiaojun

2020.9.25

Linear camera calibration-Recap



- Two procedures of calibration process: (1) compute the perspective projection matrix $\mathcal{M}$, (2) estimate intrinsic and extrinsic parameters from $\mathcal{M}$.

Linear camera calibration-Recap

$$u = \frac{\boldsymbol{m}_1 \cdot \vec{P}}{\boldsymbol{m}_3 \cdot \vec{P}} \qquad v = \frac{\boldsymbol{m}_2 \cdot \vec{P}}{\boldsymbol{m}_3 \cdot \vec{P}}$$

3. for each feature point i we have

$$(\boldsymbol{m}_1 - u_i \boldsymbol{m}_3) \cdot \vec{P}_i = 0$$
$$(\boldsymbol{m}_2 - v_i \boldsymbol{m}_3) \cdot \vec{P}_i = 0$$

$$\begin{pmatrix} \boldsymbol{P}_1^T & 0^T & -u_1 \boldsymbol{P}_1^T \\ 0^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \cdots & \cdots & \cdots \\ \boldsymbol{P}_n^T & 0^T & -u_n \boldsymbol{P}_n^T \\ 0^T & 0^T & -v_n \boldsymbol{P}_n^T \end{pmatrix} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

2. That is $\mathcal{P}\boldsymbol{m} = 0$, when $n \geqslant 6$, homogeneous linear least-squares can be used to solve the unit vector $\boldsymbol{m}$, hence project matrix $\mathcal{M}$

Linear camera calibration-Recap

- Write $\mathcal{M}$ as $\mathcal{M} = (\mathcal{A} \;\; \boldsymbol{b})$, with $\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \boldsymbol{a}_3^T$ denoting the rows of $\mathcal{A}$, and we have

$$\rho(\mathcal{A} \;\; \boldsymbol{b}) = \mathcal{K}(\mathcal{R} \;\; \boldsymbol{t}) \Leftrightarrow$$

$$\rho \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \frac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{pmatrix}$$

where ρ is an unknown scale factor to account for the fact that the recovered matrix $\mathcal{M}$ has unit Frobenius form since $|\mathcal{M}| = |\boldsymbol{m}| = 1$.

Linear camera calibration-Recap

$$\begin{bmatrix} P_{x1} & P_{y1} & P_{z1} & 1 & 0 & 0 & 0 & 0 & -u_1P_{x1} & -u_1P_{y1} & -u_1P_{z1} \\ 0 & 0 & 0 & 0 & P_{x1} & P_{y1} & P_{z1} & 1 & -v_1P_{x1} & -v_1P_{y1} & -v_1P_{z1} \\ & & \cdots & & \cdots & & \cdots & & & & \\ P_{xn} & P_{yn} & P_{zn} & 1 & 0 & 0 & 0 & 0 & -u_nP_{xn} & -u_nP_{yn} & -u_nP_{zn} \\ 0 & 0 & 0 & 0 & P_{xn} & P_{yn} & P_{zn} & 1 & -v_nP_{xn} & -v_nP_{yn} & -v_nP_{zn} \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1m_{34} \\ v_1m_{34} \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ u_nm_{34} \\ v_nm_{34} \end{bmatrix}$$

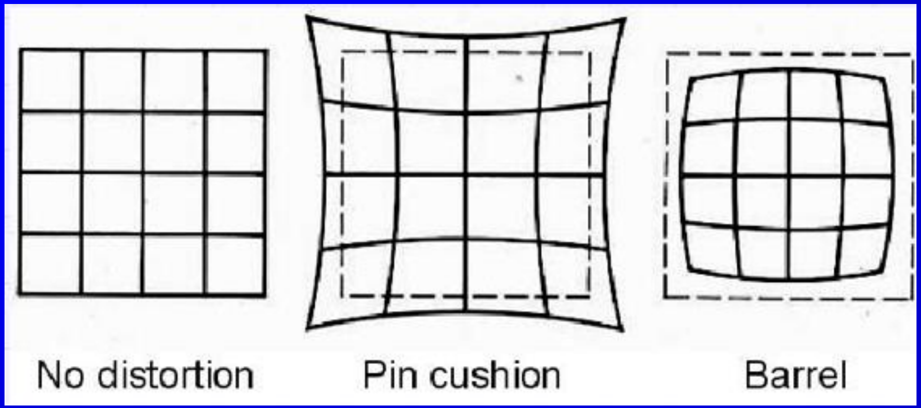
Linear camera calibration-Recap

- The matrix $\mathcal{M}$ is decomposed as

$$m_{34}\begin{bmatrix} \boldsymbol{m}_1^T & m_{14} \\ \boldsymbol{m}_2^T & m_{24} \\ \boldsymbol{m}_3^T & 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1^T & t_x \\ \boldsymbol{r}_2^T & t_y \\ \boldsymbol{r}_3^T & t_z \\ 0 & 1 \end{bmatrix}$$

where $\boldsymbol{m}_i^T(i=1,2,3)$ are columns made up with the first three elements in the ith row. $m_{i4}(i=1,2,3)$ is the elements of the 4th column in the ith row.

Linear camera calibration-Recap



$$\boldsymbol{p} = \frac{1}{z}\begin{pmatrix} 1/\lambda & 0 & 0 \\ 0 & 1/\lambda & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathcal{M}\boldsymbol{P}$$

A low degree polynomial is often used, $(\lambda = 1 + \sum_{p=1}^{q} \kappa_p d^{2p},$ with $q \leqslant 3)$ and the *distortion coefficients* $\kappa_p (p = 1, \cdots, q)$ are small.

$$d^2 = \hat{u}^2 + \hat{v}^2$$

$$d^2 = \frac{u^2}{\alpha^2} + \frac{v^2}{\beta^2} + 2\frac{uv}{\alpha\beta}\cos\theta$$

Linear camera calibration-Recap

1. *radial alignment constraint*

$$\lambda \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}} \end{pmatrix} \Rightarrow v(\boldsymbol{m}_1 \cdot \boldsymbol{P}) - u(\boldsymbol{m}_2 \cdot \boldsymbol{P}) = 0$$

2. Given n fiducial points, we have n linear equations in the eight coefficients of the vector $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, so $\mathcal{Q}\boldsymbol{n} = 0$

- $\alpha, \beta, r_1, r_2, r_3, t_x, t_y$ can be computed.

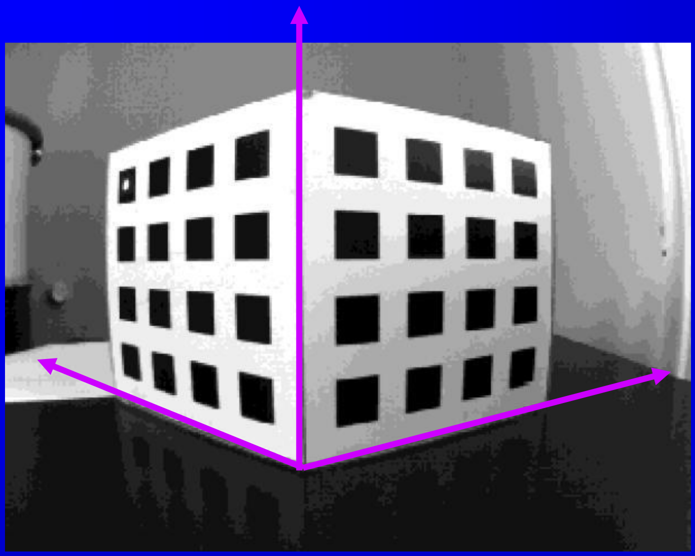
$$\begin{cases} (\boldsymbol{m}_1 - \lambda u \boldsymbol{m}_3) \cdot \boldsymbol{P} = 0 \\ (\boldsymbol{m}_2 - \lambda v \boldsymbol{m}_3) \cdot \boldsymbol{P} = 0 \end{cases}$$

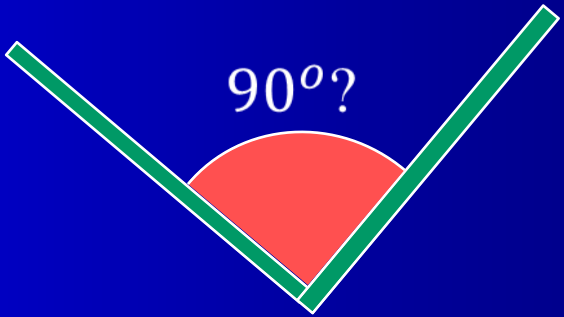
$$\lambda = 1 + \sum_{p=1}^{q} \kappa_p d^{2p}$$

$$d^2 = \frac{1}{\rho^2} \frac{|u\boldsymbol{a}_2 - v\boldsymbol{a}_1|^2}{|\boldsymbol{a}_1 \times \boldsymbol{a}_2|^2}$$

- $\rho, t_z, \kappa_p (p = 1, \cdots, q)$ can be calculated.

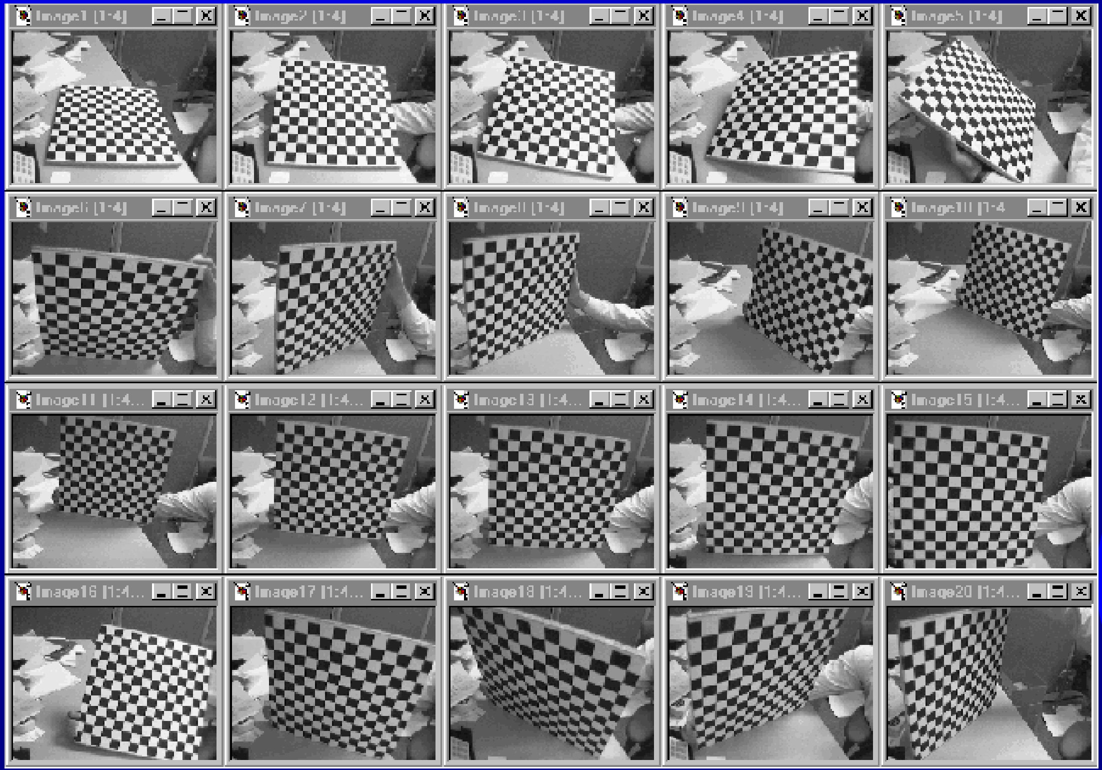
A Flexible calibration method





Zhang's Calibration Method

- Paper: Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000

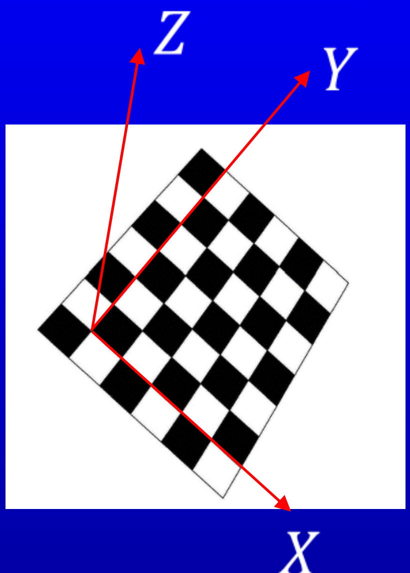


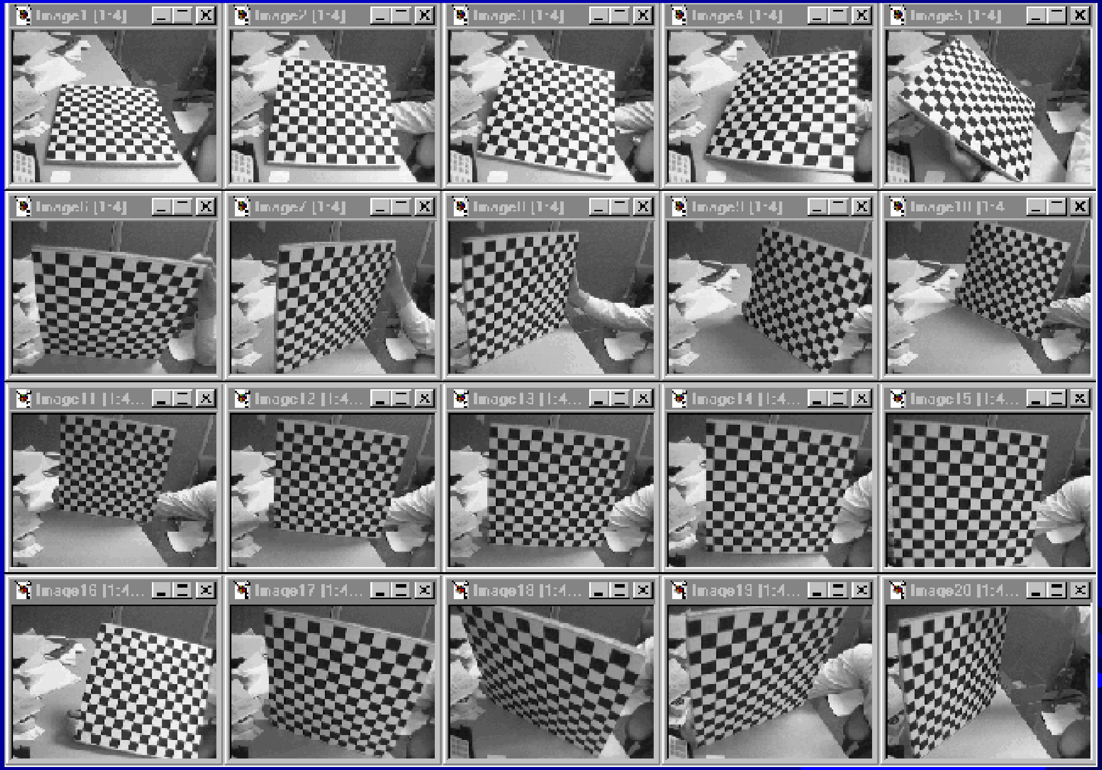
Zhang's Calibration Method

1. The flexible method:
2. Focused on a desktop vision system (DVS).
3. Considered flexibility, robustness, and low cost.
4. Only require the camera to observe a planar pattern shown at a few (min 2) different orientations.
   - Pattern can be printed and attached on planar surface.
   - Either camera or planar pattern can be moved by hand.
5. More flexible and robust than traditional techniques—Easy setup and anyone can make calibration pattern.

Zhang's Calibration Method

- Planar pattern





Zhang's Calibration Method

1. Notations: 2D point, $\boldsymbol{m} = [u, v]^T$.
2. 3D point, $\boldsymbol{M} = [X, Y, Z]^T$.
3. Augmented vector, $\widetilde{\boldsymbol{m}} = [u, v, 1]$, $\widetilde{\boldsymbol{M}} = [X, Y, Z, 1]$.
4. Relationship between 3D point $\boldsymbol{M}$ and image projection $\boldsymbol{m}$

$$s\widetilde{\boldsymbol{m}} = \boldsymbol{A}[\boldsymbol{R} \quad \boldsymbol{t}]\widetilde{\boldsymbol{M}} \quad A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

5. s is an arbitrary scale factor, $(\boldsymbol{R}, \boldsymbol{t})$ the extrinsic parameters. $\boldsymbol{A}$ is the intrinsic matrix, (u_0, v_0) the principal point, α and β teh scale factors and γ the skewness of the two image axes.

Zhang's Calibration Method

1. Homography between the model plane and its image.
2. Homography describes the relation of the two images of the same planar surface in space, i.e. $\boldsymbol{x}_1 \sim H\boldsymbol{x}_2$
3. Denote ith column of the rotation matrix R by $\boldsymbol{r}_i$.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{r}_3\ \boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2)$$

4. Relation between model point M and image m: $s\widetilde{m} = H\widetilde{M}, \;\; H = A[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}]$.
5. H is homography and defined up to a scale factor.

Zhang's Calibration Method

1. Constraints on intrinsic parameters.
2. Let H be $H = [\boldsymbol{h}_1 \ \boldsymbol{h}_2 \ \boldsymbol{h}_3]$, and $[\boldsymbol{h}_1 \ \boldsymbol{h}_2 \ \boldsymbol{h}_3] = \lambda A[\boldsymbol{r}_1 \ \boldsymbol{r}_2 \ \boldsymbol{t}]$.
3. Homography has 8 degree of freedom and 6 extrinsic parameters.
4. Two basic constraints on intrinsic parameter

$$\boldsymbol{h}_1^T A^{-T} A^{-1} \boldsymbol{h}_2 = 0 \tag{3}$$

$$\boldsymbol{h}_1^T A^{-T} A^{-1} \boldsymbol{h}_1 = \boldsymbol{h}_2^T A^{-T} A^{-1} \boldsymbol{h}_2 \tag{4}$$

Zhang's Calibration Method

- Geometric interpretation
- Model plane described in camera coordinate system

$$\begin{bmatrix} \boldsymbol{r}_3 \\ \boldsymbol{r}_3^T \boldsymbol{t} \end{bmatrix}^T [x \ y \ z \ w]^T = 0 \qquad (5)$$

where w=0 for points at infinity and w=1 otherwise.

- Model plane intersects the plane at infinity at a line

$$\begin{bmatrix} \boldsymbol{r}_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{r}_2 \\ 0 \end{bmatrix} \qquad (6)$$

- Any point on it is a linear combination of these two points, ie.

$$x_\infty = a \begin{bmatrix} \boldsymbol{r}_1 \\ 0 \end{bmatrix} + b \begin{bmatrix} \boldsymbol{r}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} a\boldsymbol{r}_1 + b\boldsymbol{r}_2 \\ 0 \end{bmatrix} \qquad (7)$$

Zhang's Calibration Method

1. The point x_∞ satisfies : $x_\infty^T x_\infty = 0$, i.e. $(ar_1 + br_2)^T (ar_1 + br_2) = 0$, or $a^2 + b^2 = 0$.
2. The solution is $b = \pm ai$, where $i^2 = -1$. That is two intersection points are $x_\infty = a[r_1 \pm ir_2 \;\; 0]^T$.
3. Their projection in the image plane is given, up to a scale, by

$$\widetilde{m}_\infty = A(r_1 \pm ir_2) = h_1 \pm ih_2. \tag{8}$$

4. Point $\widetilde{m}_\infty$ on the image of the absolute conic, described by $A^{-T}A^{-1}$. This gives

$$(h_1 \pm ih_2)^T A^{-T} A^{-1} (h_1 \pm ih_2) = 0 \tag{9}$$

5. Requiring that both real and imaginary parts be zero yields (3) and (4)

Zhang's Calibration Method

- Calibration: Analytical solution and nonlinear optimization technique based on maximum-likelihood criterion.

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

$$\begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma-u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2}+\frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma-u_0\beta)}{\alpha^2\beta^2}-\frac{v_0^2}{\beta^2} \\ \frac{v_0\gamma-u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma-u_0\beta)^2}{\alpha^2\beta^2}-\frac{v_0^2}{\beta^2} & \frac{(v_0\gamma-u_0\beta)^2}{\alpha^2\beta^2}+\frac{v_0^2}{\beta^2}+1 \end{bmatrix} \quad (10)$$

- B is defined by 6D vector b:

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

Zhang's Calibration Method

1. ith column of $H = h_i$, $h_i = [h_{i1}\ h_{i2}\ h_{i3}]^T$.
2. Following relation hold: $h_i^T B h_j = v_{ij}^T b$. v_{ij} defined as: $v_{ij} = [h_{i1}h_{j1}\ h_{i1}h_{j2} + h_{i2}h_{j1}\ h_{i2}h_{j2}\ h_{i3}h_{j1} + h_{i1}h_{j3}\ h_{i3}h_{j2} + h_{i2}h_{j3}\ h_{i3}h_{j3}]^T$
3. Two fundalmental constraints from homography become

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

4. If observed n images of model plane $Vb = 0$. V is $2n \times 6$ matrix.
5. Solution of $Vb = 0$ is the eigenvector of $V^T V$ associated the smallest eigenvalue. So, we can estimate b.

Zhang's Calibration Method

1. If $n \geq 3$, unique solution b defined up to a scale factor.
2. If $n = 2$, impose skewless constraint $\gamma = 0$.
3. If $n = 1$, can only solve two camera intrinsic parameters, α and β, assuming u_0 and v_0 are known and $\gamma = 0$.
4. Estimate B up to scale factor, $B = \lambda A^{-T}A$, the intrinsic parameters from matrix B is as follows.

$$v_0 = \left(B_{12}B_{13} - B_{11}B_{23}\right) / \left(B_{11}B_{22} - B_{12}^2\right)$$

$$\lambda = B_{33} - \left[B_{13}^2 + v_0\left(B_{12}B_{13} - B_{11}B_{23}\right)\right] / B_{11}$$

$$\alpha = \sqrt{\lambda / B_{11}}$$

$$\beta = \sqrt{\lambda B_{11} / \left(B_{11}B_{22} - B_{12}^2\right)}$$

$$\gamma = -B_{12}\alpha^2\beta / \lambda$$

$$u_0 = \gamma v_0 / \alpha - B_{13}\alpha^2 / \lambda$$

Zhang's Calibration Method

1. Once A is known, the extrinsic parameters for each image is readily computed from homography $H = [h_1 \; h_2 \; h_3]^T = \lambda A [r_1 \; r_2 \; t]^T$, we have

$$r_1 = \lambda A^{-1} h_1 \quad (12)$$

$$r_2 = \lambda A^{-1} h_2 \quad (13)$$

$$r_3 = r_1 \times r_2 \quad (14)$$

$$t = \lambda A^{-1} h_3 \quad (15)$$

2. R dose not, in general, satisfy properties of a rotation matrix because of noised in data.

3. R can be obtained through singular value decomposition (SVD), see appendix C in paper.

Zhang's Calibration Method

- Concerning with radial distortion:
- Let (u, v) be the ideal pixel image coordinates, and $(\breve{u}, \breve{v})$ the corresponding real observed image coordinates. (x, y) and $(\breve{x}, \breve{y})$ are the ideal and real normalized image coordinates.we have,

$$\breve{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (16)$$

$$\breve{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2], \quad (17)$$

where k_1 and k_2 are the coefficients of the radial distortion. From $\breve{u} = u_0 + \alpha\breve{x} + \gamma\breve{y}$ and $\breve{v} = v_0 + \beta\breve{y}$ and assuming $\gamma = 0$, then

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (18)$$

$$\breve{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]) \quad (19)$$

Zhang's Calibration Method

1. One way to estimate k_1 and k_2 after estimated the other parameters, which can give the ideal pixel coordinates (u, v). From equation (18), we have,

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \breve{u}-u \\ \breve{v}-v \end{bmatrix} \quad (20)$$

2. Given m points in n images, we get $2mn$ equations or in matrix form as $\boldsymbol{Dk} = \boldsymbol{d}$, where $\boldsymbol{k} = [k_1, k_2]^T$. The LS solution is $\boldsymbol{k} = (\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\boldsymbol{d}$
3. Once k_1 and k_2 are estimated, one can refine the other parameters by Maximum likelihood estimation until convergence.

Zhang's Calibration Method

- Complete Maximum Likelihood Estimation:
- The complete set of parameters by minimizing the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\|m_{ij} - \breve{m}(A, k_1, k_2, R_i, t_i, M_j)\|^2 \qquad (21)$$

where $\breve{m}(A, k_1, k_2, R_i, t_i, M_j)$ is the projection of point M_j in image i according to equation $s\widetilde{m} = H\widetilde{M}$, followed by distortion according to equation (18). This is a nonlinear minimization problem, which is solved with the Leverberg-Marquardt algorithm.

Zhang's Calibration Method

- Experiments

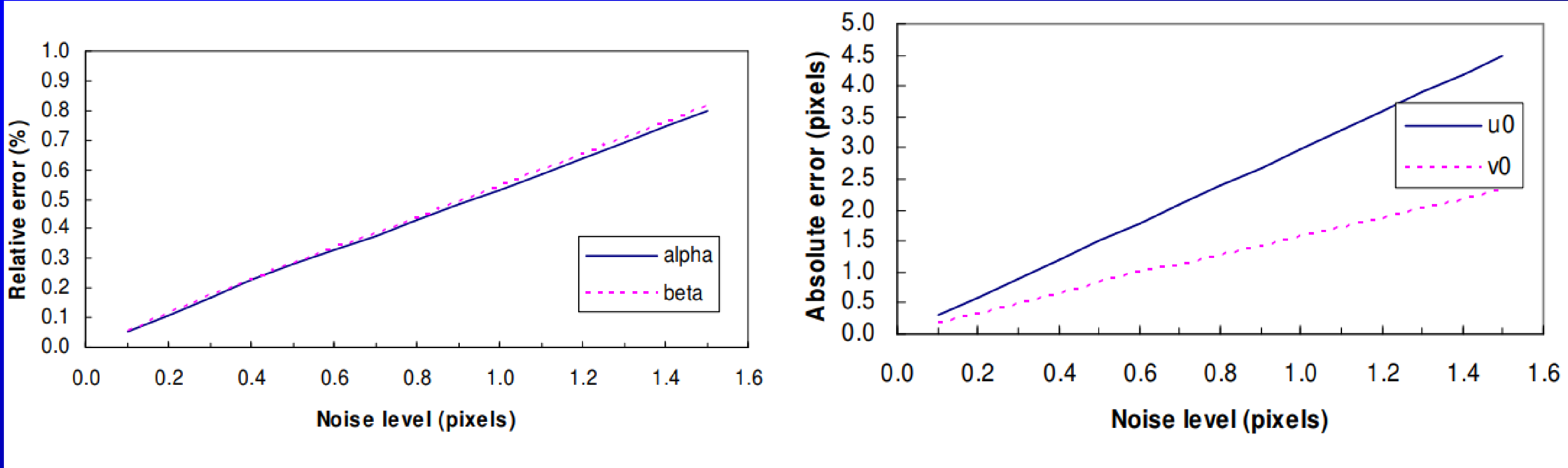


Figure 1: Errors vs. the noise level of the image points

Zhang's Calibration Method

- Experiments

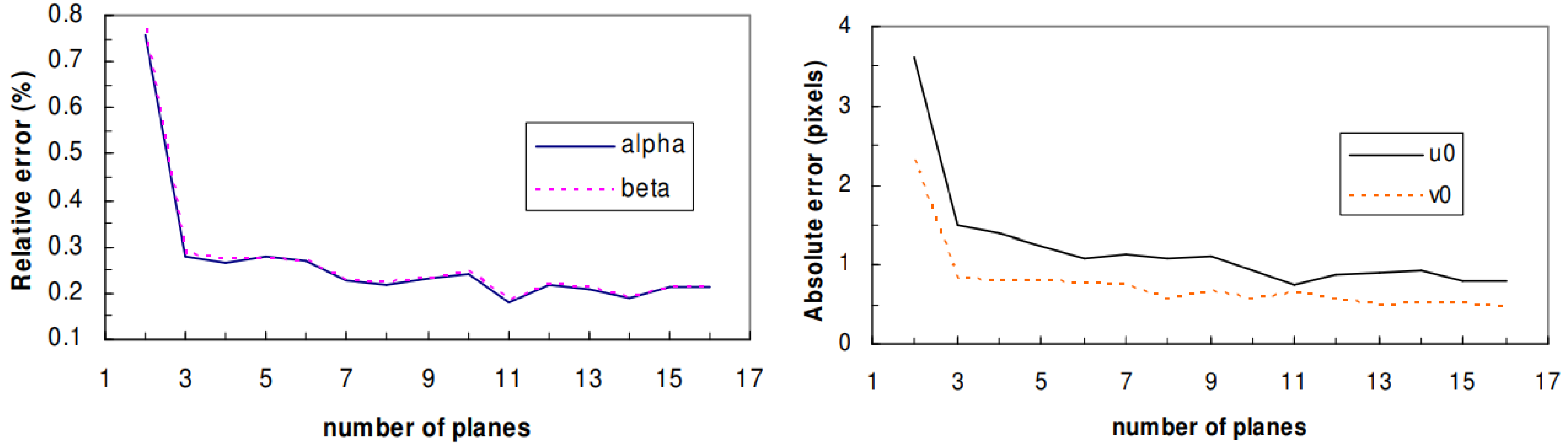
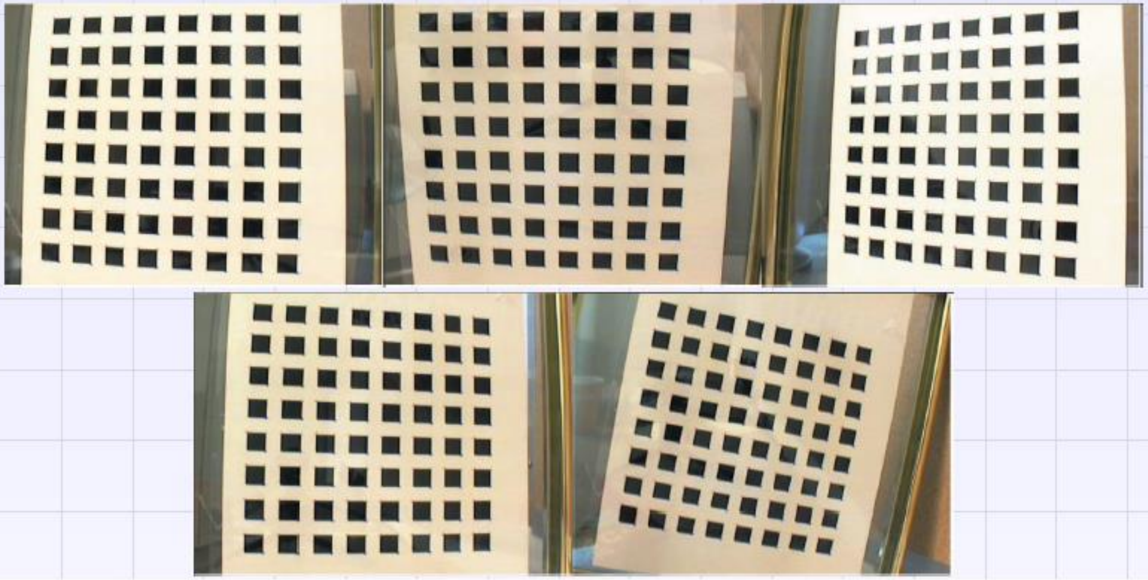


Figure 2: Errors vs. the number of images of the model plane

Zhang's Calibration Method

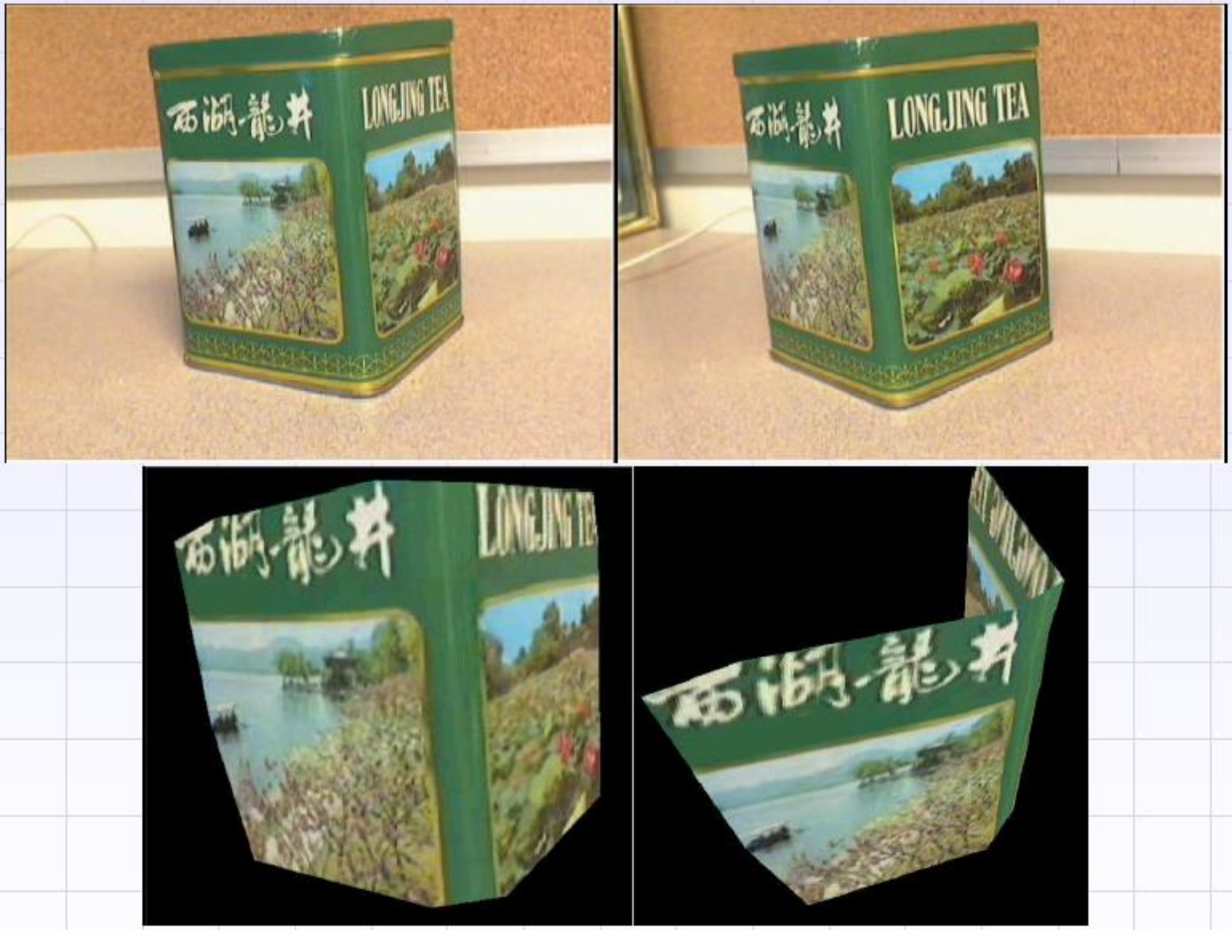
- Experimental results:



quadruple	(1234)	(1235)	(1245)	(1345)	(2345)	mean	deviation
α	831.81	832.09	837.53	829.69	833.14	832.85	2.90
β	831.82	832.10	837.53	829.91	833.11	832.90	2.84
γ	0.2867	0.1069	0.0611	0.1363	0.1096	0.1401	0.086
u_0	304.53	304.32	304.57	303.95	303.53	304.18	0.44
v_0	206.79	206.23	207.30	207.16	206.33	206.76	0.48
k_1	-0.229	-0.228	-0.230	-0.227	-0.229	-0.229	0.001
k_2	0.195	0.191	0.193	0.179	0.190	0.190	0.006
RMS	0.361	0.357	0.262	0.358	0.334	0.334	0.04

Zhang's Calibration Method

- Experimental results:

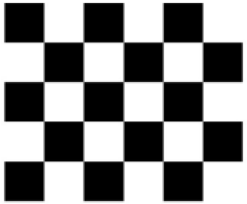


Zhang's Calibration Method

1. Calibration procedure:
2. Print a pattern and attach to a planar surface.
3. Take few images of the model plane under different orientations.
4. Detect feature points in the images.
5. Estimate five intrinsic parameters and all the extrinsic parameters using the closed-form solution.
6. Refine all parameters by obtaining maximum-likelihood estimate.

Zhang's Calibration Method

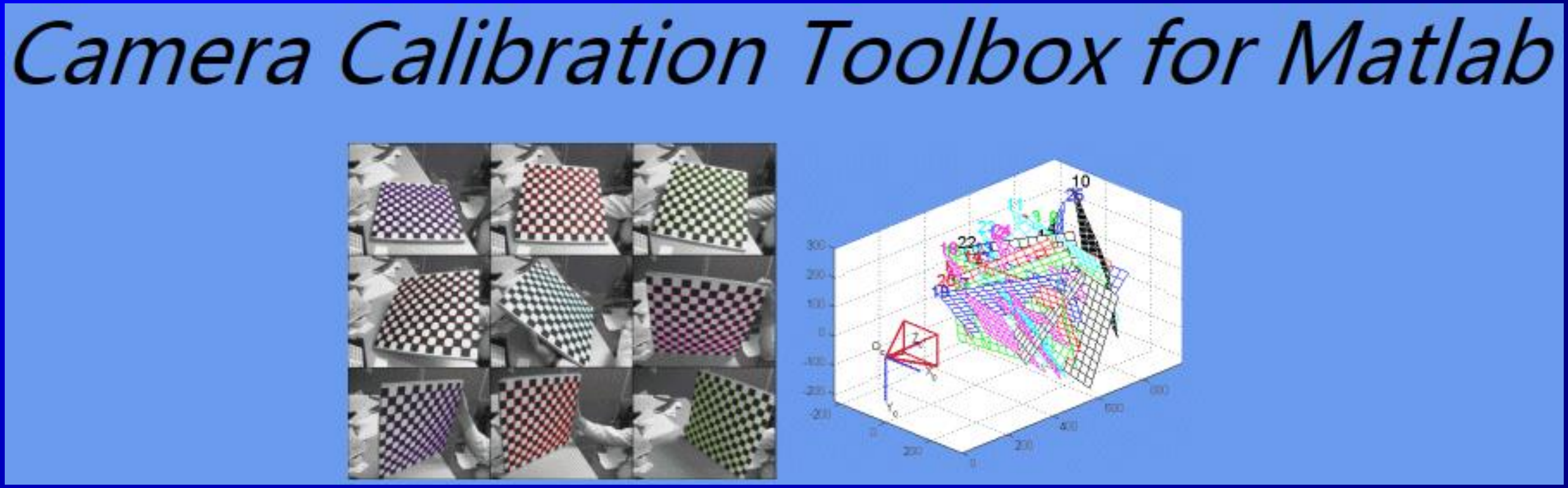
- OpenCV implementation



```
bool findChessboardCorners(image, patternSize, corners, flags);

double calibrateCamera(objectPoints, imgPoints, imgSize, camMatrix,
                       distCoeffs, rvecs, tvecs, flags, criteria);

Mat initCameraMatrix2D(objPoints, imgPoints, imgSize, aspectRatio);

void getOptimalNewCameraMatrix(camMatrix, distCoeffs, imgSize,
                               alpha, newImgSize, PixROI, cPP);

void undistort(src, dst, cameraMatrix, distCoeffs, newCameraMatrix);
```

Zhang's Calibration Method

- Camera Calibration Toolbox---matlab



Zhang's Calibration Method

- Camera Calibration Toolbox---matlab
- http://www.vision.caltech.edu/bouguetj/calib_doc/
- Go to the download page, and retrieve the latest version of the complete camera calibration toolbox for Matlab.
- Store the individual matlab files (.m files) into a unique folder TOOLBOX_calib (default folder name).
- Run Matlab and add the location of the folder TOOLBOX_calib to the main matlab path. This procedure will let you call any of the matlab toolbox functions from anywhere. Under Windows, this may be easily done by using the path editing menu. Under Unix or Linux, you may use the command path or addpath (use the help command for function description).
- Run the main matlab calibration function calib_gui (or calib).
- A mode selection window appears on the screen:

Zhang's Calibration Method

- http://www.vision.caltech.edu/bouguetj/calib_doc/
- Since both modes have the exact same user interface, in the context of this documentation, let us select the standard mode by clicking on the top button of the window. The main calibration toolbox window appears on the screen (replacing the mode selection window):

Zhang's Calibration Method

- This example lets you learn how to use all the features of the toolbox: loading calibration images, extracting image corners, running the main calibration engine, displaying the results, controlling accuracies, adding and suppressing images, undistorting images, exporting calibration data to different formats.
  - Download the calibration images all at once calib_example.zip (4461Kb zipped) or one by one, and store the 20 images into a seperate folder named calib_example.
  - Reading the images: Click on the Image names button in the Camera calibration tool window. Enter the basename of the calibration images (Image) and the image format (tif).

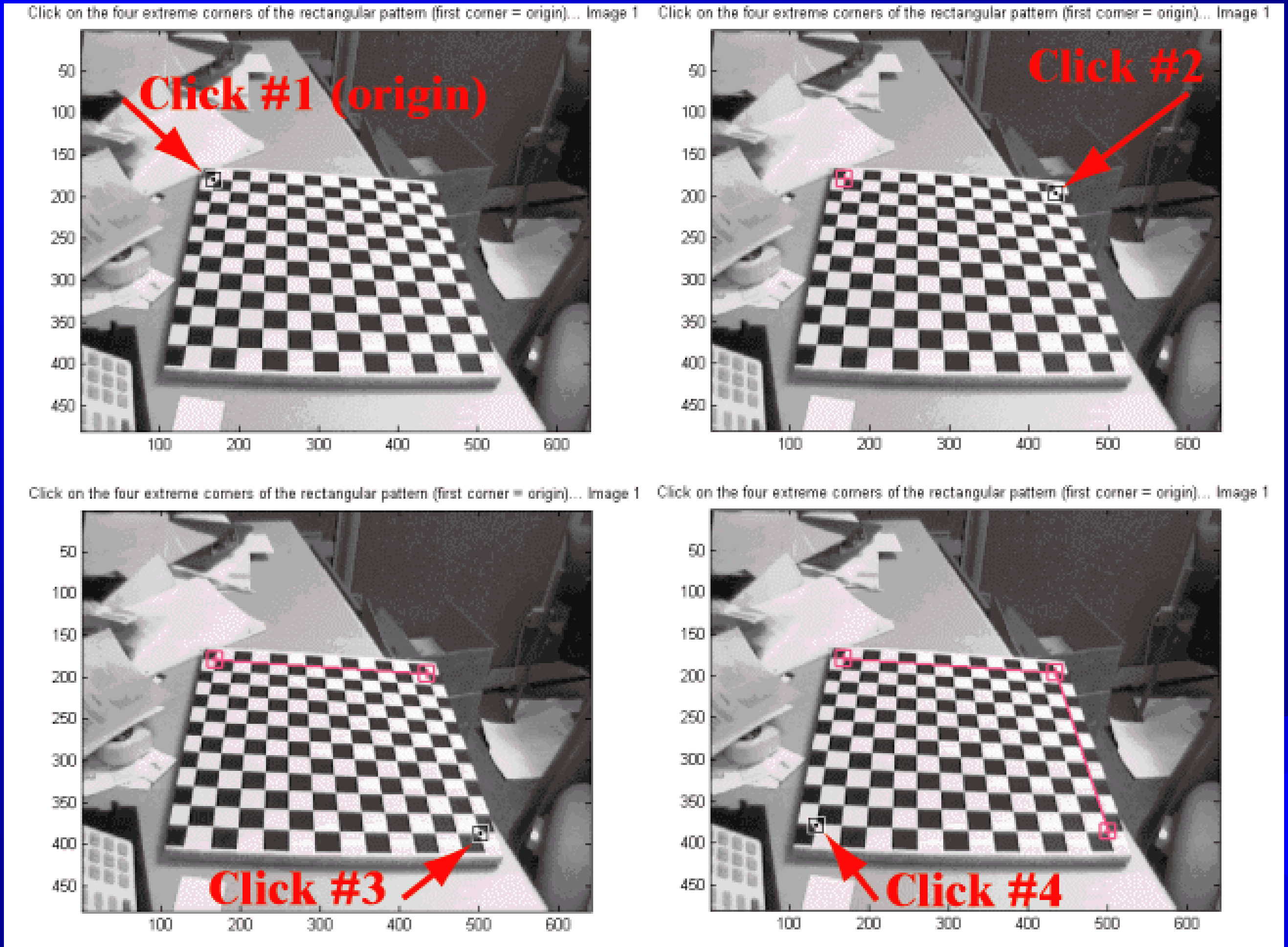
```
Basename camera calibration images (without number nor suffix): Image
Image format: ([]='r'='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpg', 'm'='ppm') t
Loading image 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...
done
```

  - Extract the grid corners:Click on the Extract grid corners button in the Camera calibration tool window.

```
Extraction of the grid corners on the images
Number(s) of image(s) to process ([] = all images) =
```

  - Press "enter" (with an empty argument) to select all the images (otherwise, you would enter a list of image indices like [2 5 8 10 12] to extract corners of a subset of images). Then, select the default window size of the corner finder: wintx=winty=5 by pressing "enter" with empty arguments to the wintx and winty question. This leads to a effective window of size 11x11 pixels.

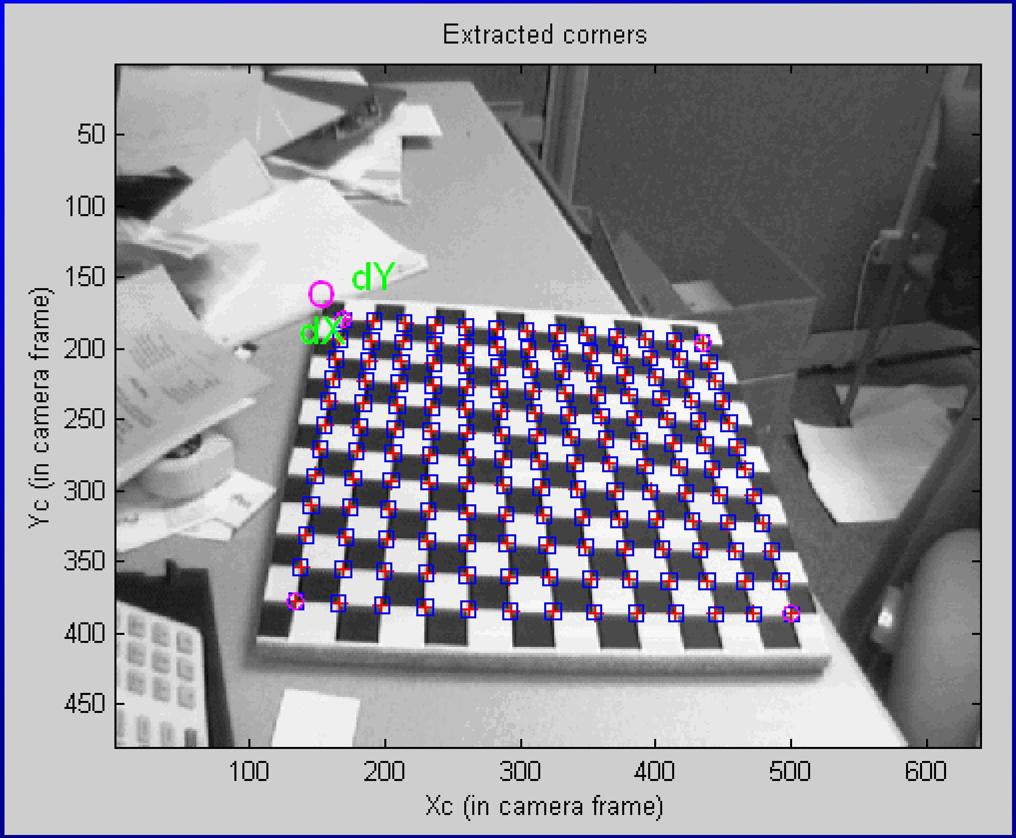
Zhang's Calibration Method



Zhang's Calibration Method

- Enter the sizes dX and dY in X and Y of each square in the grid (in this case, dX=dY=30mm=default values):

```
Size dX of each square along the X direction ([]=30mm) = 30
Size dY of each square along the Y direction ([]=30mm) = 30
```



Zhang's Calibration Method

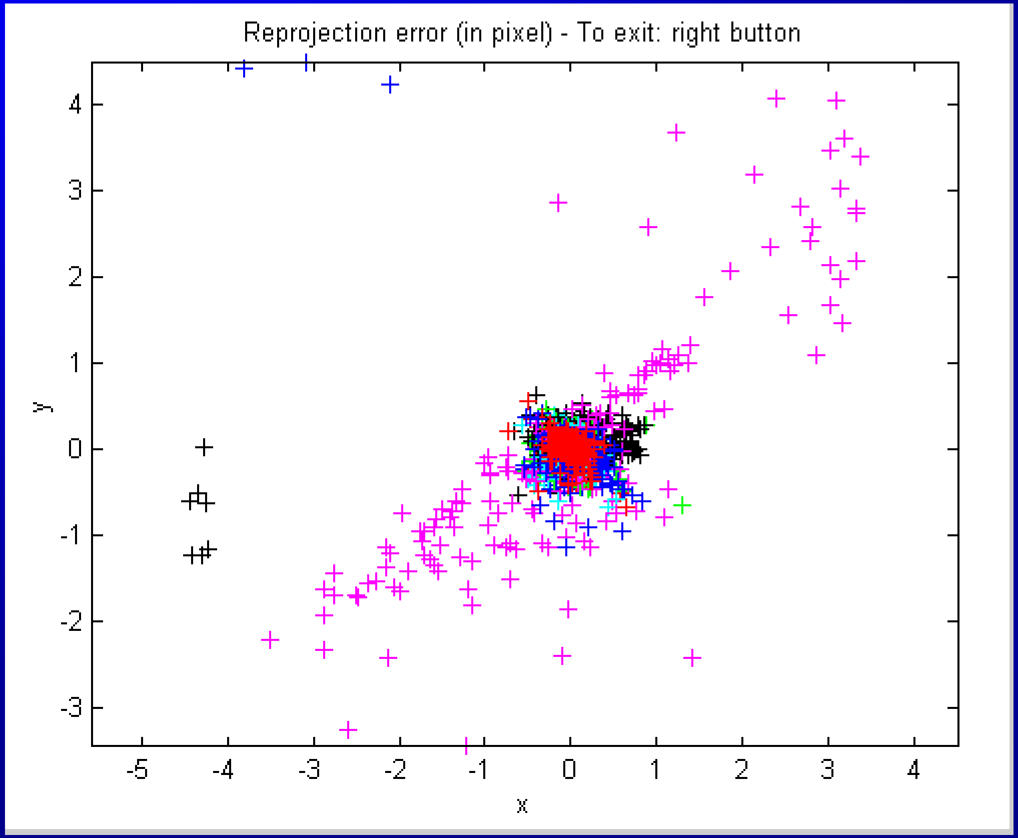
- Main Calibration step:
  - After corner extraction, click on the button Calibration of the Camera calibration tool to run the main camera calibration procedure.
  - Calibration is done in two steps: first initialization, and then nonlinear optimization.

```
Calibration parameters after initialization:

Focal Length:          fc = [ 671.13759   680.77186 ]
Principal point:       cc = [ 319.50000   239.50000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 20
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...done
Estimation of uncertainties...done
```

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 661.67001   662.82858 ] ± [ 1.17913   1.26567 ]
Principal point:       cc = [ 306.09590   240.78987 ] ± [ 2.38443   2.17481 ]
Skew:             alpha_c = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc = [ -0.26425   0.22645   0.00020   0.00023   0.00000 ] ± [ 0.00934   0.03826   0.00052   0.00053   0.00000 ]
Pixel error:          err = [ 0.45330   0.38916 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

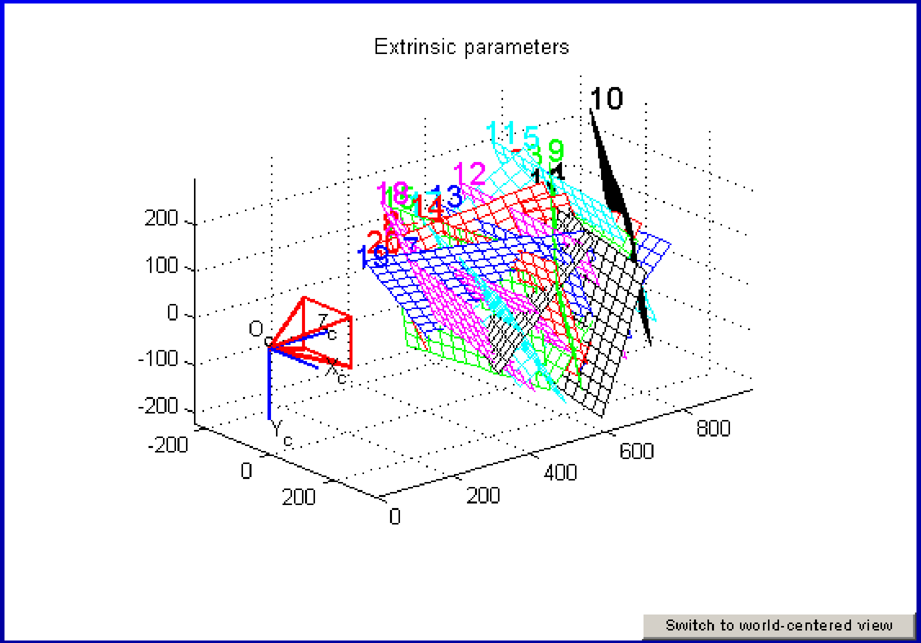
Zhang's Calibration Method

- The reprojection error is also shown in the form of color-coded crosses:



Zhang's Calibration Method

- Click on Show Extrinsic in the Camera calibration tool. The extrinsic parameters (relative positions of the grids with respect to the camera) are then shown in a form of a 3D plot:

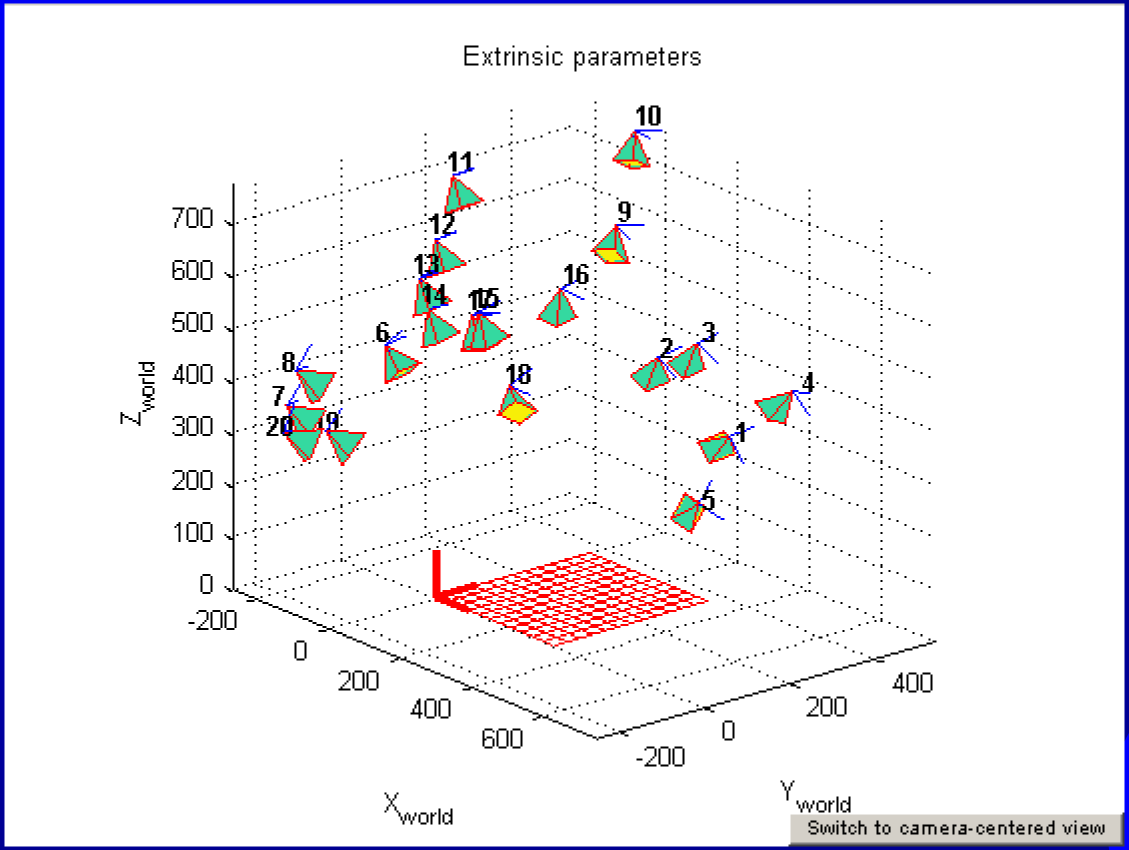


Camera-centered

- On this figure, the frame (Oc,Xc,Yc,Zc) is the camera reference frame. The red pyramid corresponds to the effective field of view of the camera defined by the image plane. To switch from a "camera-centered" view to a "world-centered" view, just click on the Switch to world-centered view button located at the bottom-left corner of the figure.

Zhang's Calibration Method

- World-centered



Homework

- Write a OpenCV based program to calibrate the camera by using the images provided by TA.
- Then, calibrate a camera by using camera calibration toolbox.
- Compare the computed parameters from two methods. Write a experiment report to present the procedures and the results. You can refer to the following document. http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html.

See You