Computer Vision

Camera Calibration

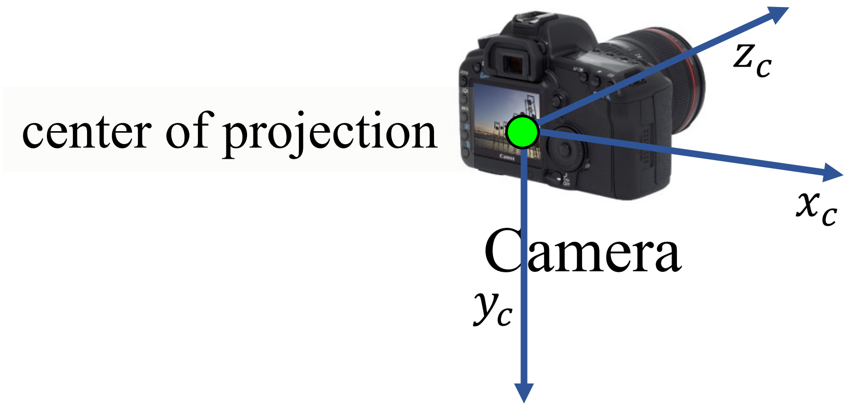
Technische Hochschule Rosenheim
Winter 2024/25
Prof. Dr. Jochen Schmidt

What is Camera Calibration?

- Determining the camera parameters
  - Imaging properties of the individual cameras
  - Relative position and orientation camera/world or cameras to each other

- What for?
  - Correct distance measurements in the image
  - Depth reconstruction (e.g. with stereo method)

3D Coordinate Systems



Two important 3D coordinate systems:

1. *world* coordinate system (X, Y, Z)
2. *camera* coordinate system (x_c, y_c, z_c)

slide credit: Linda Shapiro, ECE/CS 576, "Cameras and Stereo", University of Washington

Mapping 3D – 2D

pixel (homogeneous, 3D)

world point (homogeneous, 4D)

world coordinate system (X, Y, Z)

projection matrix 3x4

$$\boldsymbol{b} = \boldsymbol{P}\boldsymbol{p}_w = \boldsymbol{K}\boldsymbol{M}\boldsymbol{E}\boldsymbol{p}_w$$

intrinsic parameters 3x3

sensor coordinate system (u, v)

projection model 3x4

ideal image coordinates (x', y')

extrinsic parameters 4x4

camera coordinate system (x_c, y_c, z_c)

Homogeneous coordinates

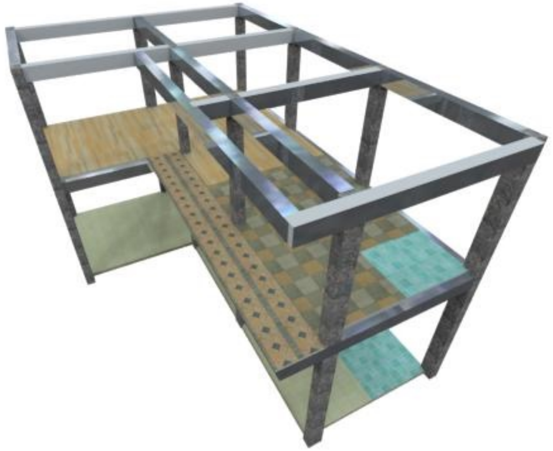
- from projective geometry
- Idea:
  - Adding an additional dimension
    - 2D becomes 3D, 3D becomes 4D

      (x, y) → (x, y, w)
      (x, y, z) → (x, y, z, w)
    - This simplifies the calculations!
  - The new component is always 1 when going to homogeneous coordinates
  - additionally: w = 0 → points at infinity
  - Notation ~: “equal up to scale"

- De-homogenization
  - divide by last coordinate

    (x, y, w) → (x/w, y/w)
    (x, y, z, w) → (x/w, y/w, z/w)

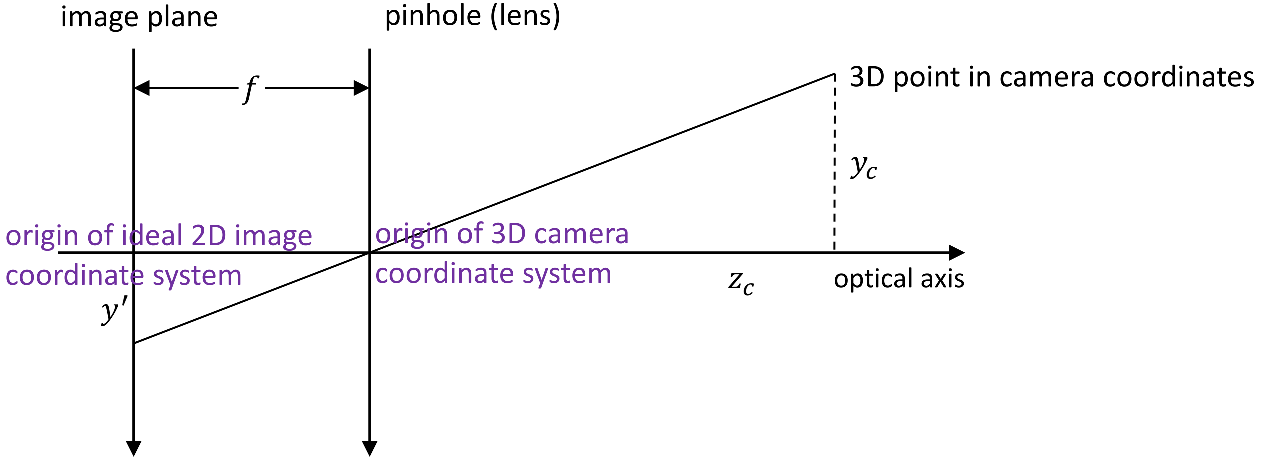
Perspective Projection

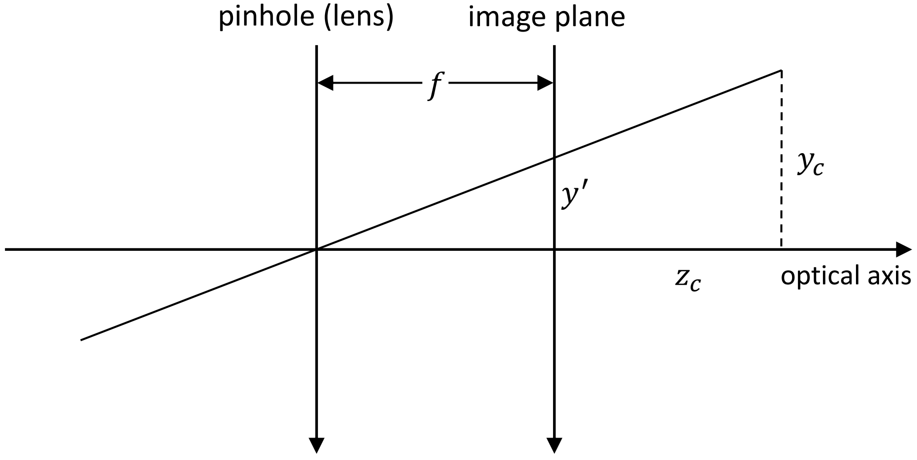



- What happens to parallel lines?
- What happens to angles?
- What happens to distances?

slide credit: Linda Shapiro, ECE/CS 576, "Cameras and Stereo", University of Washington

Perspective Projection





$$x' = -f\frac{x_c}{z_c}$$

$$y' = -f\frac{y_c}{z_c}$$

$$x' = f\frac{x_c}{z_c}$$

$$y' = f\frac{y_c}{z_c}$$

This is a non-linear mapping!

Perspective Projection

Technische Hochschule **Rosenheim**

With homogeneous coordinates, this becomes a linear mapping

- and only then can the 3D – 2D image be represented as a matrix

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} f\,\dfrac{x_c}{z_c} \\ f\,\dfrac{y_c}{z_c} \end{pmatrix} \rightarrow \begin{pmatrix} f\,x_c \\ f\,y_c \\ z_c \end{pmatrix}$$

Mapping 3D – 2D becomes:

$$\begin{pmatrix} f\,x_c \\ f\,y_c \\ z_c \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

ideal 2D homogeneous image coordinates

Intrinsic Camera Parameters

- Optical and geometric characteristics of the camera
- Do not change when camera is moved
- Parameters:
  - Focal length (*Brennweite*) f
  - Pixel size d_x, d_y
  - Principal point (*Hauptpunkt*) p_u, p_v
  - Angle between image axes α
    - is often assumed to be 90° (as a very good approximation)

  - Lens distortion
    - radial around the principal point
    - or tangential to a line through the principal point

Focal Length

- Can think of as "zoom"

24mm

50mm

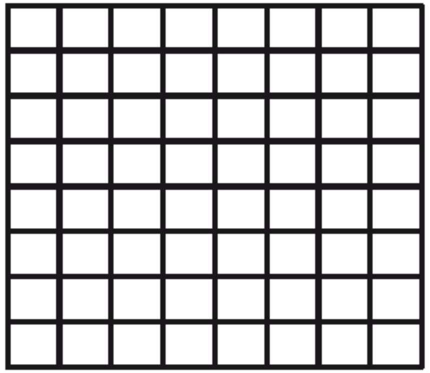
200mm

800mm

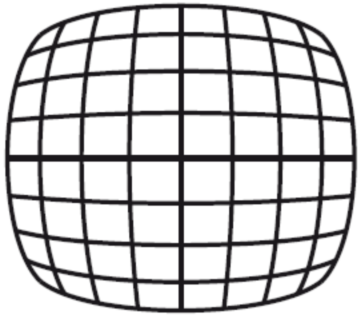
- Related to field of view

slide credit: Linda Shapiro, ECE/CS 576, "Cameras and Stereo", University of Washington

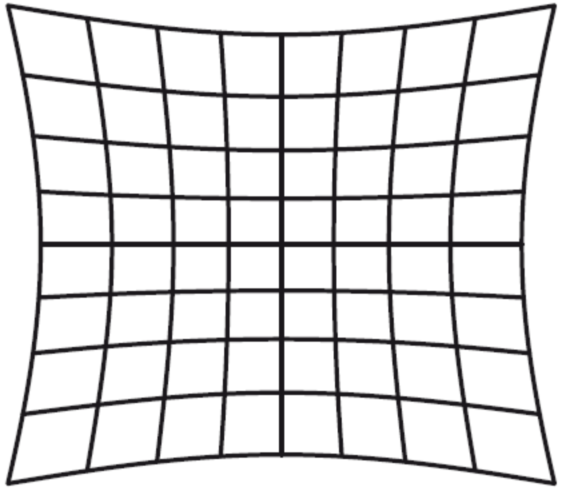
Radial Distortion



ideal



barrel-shaped



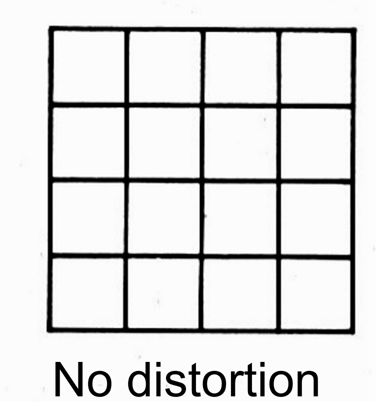
cushion-shaped

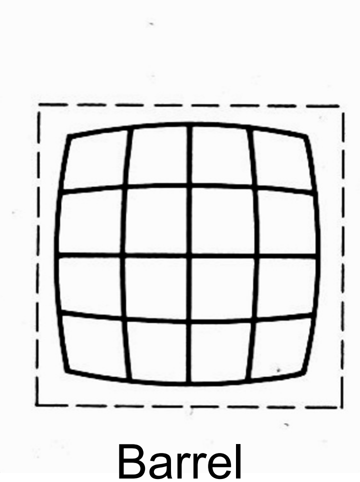
from: [Beyerer16]

Correcting Radial Distortion

from Helmut Dersch

Barrel Distortion

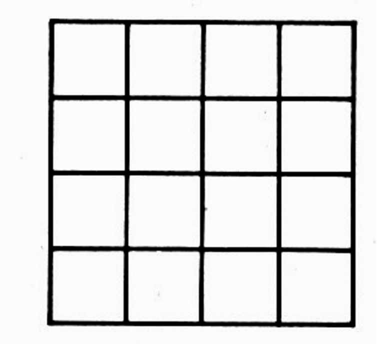




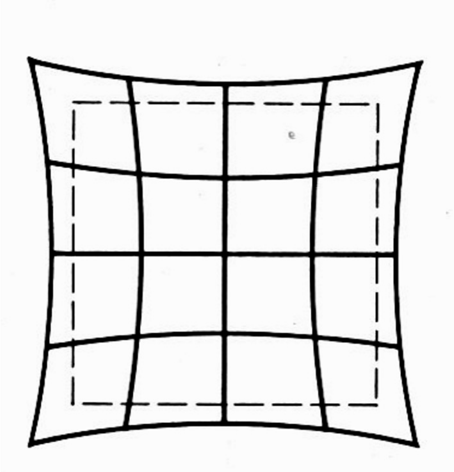
Wide Angle Lens

slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

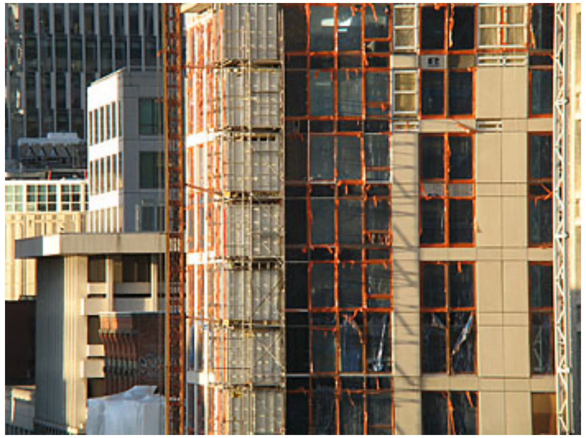
Pin Cushion Distortion



No distortion



Pin cushion



Telephoto lens

slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

Intrinsic Camera Parameters

Calibration matrix: collects intrinsic parameters

with angle:

$$\boldsymbol{K} = \begin{pmatrix} f_x & s & p_u \\ 0 & f_y & p_v \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{d_x} & \frac{-f\tan\left(\frac{\pi}{2}-\alpha\right)}{d_x} & p_u \\ 0 & \frac{f}{d_y} & p_v \\ 0 & 0 & 1 \end{pmatrix}$$

without angle:

$$\boldsymbol{K} = \begin{pmatrix} f_x & 0 & p_u \\ 0 & f_y & p_v \\ 0 & 0 & 1 \end{pmatrix}$$

effective focal length in pixels:
focal length change is indistinguishable from pixel size change
(these only occur as a product)

Modeling Radial Distortion

Technische Hochschule Rosenheim

Distortion-Free:

(u, v: sensor coordinates)

$$u = f_x \frac{x_c}{z_c} + p_u$$

$$v = f_y \frac{y_c}{z_c} + p_v$$

With Distortion:

1. Project (x_c, y_c, z_c) to normalized image coordinates

$$x_n = \frac{x_c}{z_c} \qquad y_n = \frac{y_c}{z_c}$$

2. Apply radial distortion

$$r^2 = x_n^2 + y_n^2$$

$$x_d = x_n\left(1 + \kappa_1 r^2 + \kappa_2 r^4\right)$$

$$y_d = y_n\left(1 + \kappa_1 r^2 + \kappa_2 r^4\right)$$

3. Apply focal length & translate image center

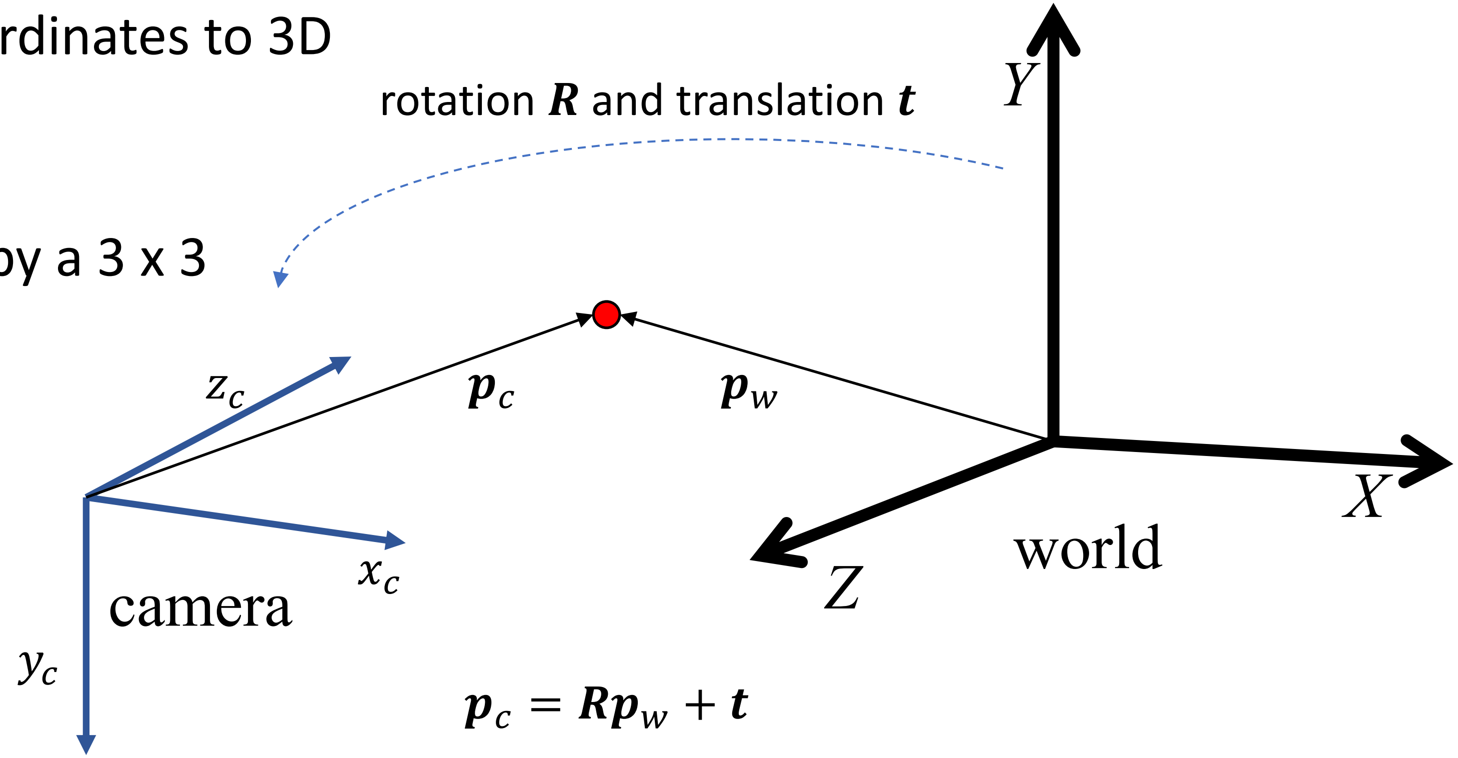
$$u = f_x x_d + p_u \qquad v = f_y y_d + p_v$$

This breaks the linear mapping from world to pixels!

Extrinsic Camera Parameters

- parameters change when camera is moved
- position and orientation of the camera (= pose)
- describes the mapping of 3D world coordinates to 3D camera coordinates

- Any rotation in 3D can be represented by a 3 x 3 matrix $\boldsymbol{R}$ with the following properties:
  - $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$
  - det($\boldsymbol{R}$) = 1
- the row and column vectors are therefore orthonormal
- the matrix has 9 elements
- but only 3 degrees of freedom



Entire Mapping 3D – 2D

pixel (homogeneous, 3D)

world point (homogeneous, 4D)

world coordinate system (X, Y, Z)

projection matrix 3x4

$$\boldsymbol{b} = \boldsymbol{P}\boldsymbol{p}_w = \boldsymbol{K}\boldsymbol{M}\boldsymbol{E}\boldsymbol{p}_w$$

intrinsic parameters 3x3

sensor coordinate system (u, v)

projection model 3x4

ideal image coordinates (x', y')

extrinsic parameters 4x4

camera coordinate system (x_c, y_c, z_c)

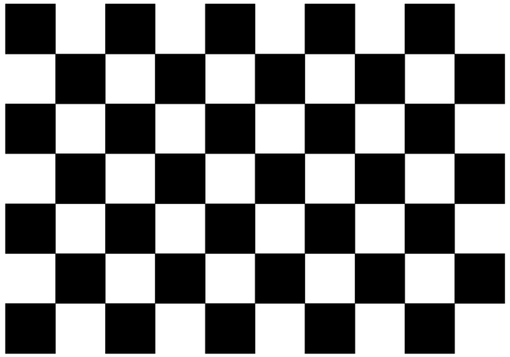
$$\boldsymbol{b} = \begin{pmatrix} u_h \\ v_h \\ s \end{pmatrix} = \boldsymbol{KME} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & p_u \\ 0 & f_y & p_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Determination of camera parameters = **camera calibration**

Camera Calibration

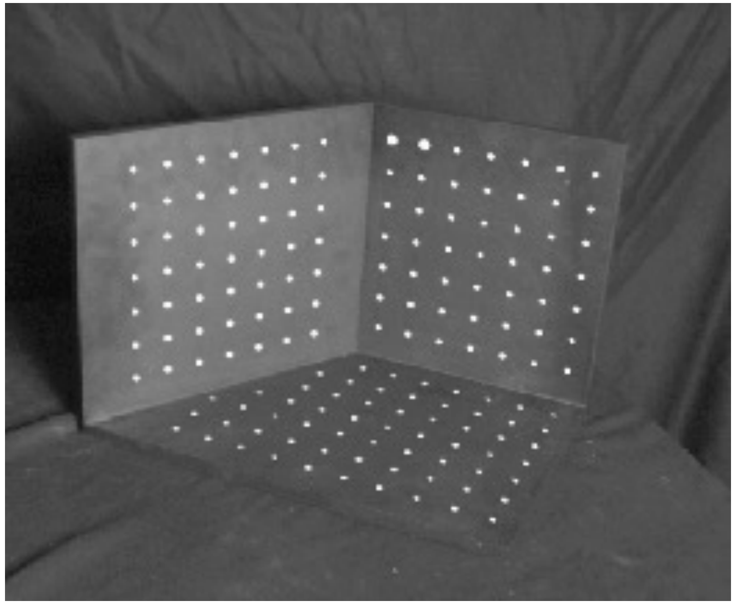
- Various methods available
  - work according to a similar basic principle
  - a simple one is presented here

- Idea
  - Take several images of a calibration pattern
    - contains N 3D points with known world coordinates
  - establish correspondences between world and pixels
  - Calculate camera parameters



Camera Calibration From Projection Matrix

Directly estimate 11 unknowns in the projection matrix using known 3D points (X, Y ,Z) and measured feature positions (u,v)



slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

Camera Calibration From Projection Matrix

- wanted: 3x4 Matrix

$$\boldsymbol{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix}$$

- $\boldsymbol{P}$ has 12 elements
- camera has only 11 parameters
  - 5 intrinsic (2 x focal lengths, principal point, angle)
  - 6 extrinsic (3 x rotation, 3 x translation)
  - Discrepancy due to homogeneous coordinates
- therefore: normalization of the matrix: $\|(p_{11} \quad p_{12} \quad p_{13} \quad \cdots \quad p_{34})^T\| = 1$

Camera Calibration From Projection Matrix

For every 2D – 3D correspondence the following equation applies

$$\begin{pmatrix} u_h \\ v_h \\ s \end{pmatrix} = \boldsymbol{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$u = \frac{u_h}{s} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \qquad v = \frac{v_h}{s} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

Multiply both equations by the denominator:

$$u(p_{31}X + p_{32}Y + p_{33}Z + p_{34}) = p_{11}X + p_{12}Y + p_{13}Z + p_{14}$$

$$v(p_{31}X + p_{32}Y + p_{33}Z + p_{34}) = p_{21}X + p_{22}Y + p_{23}Z + p_{24}$$

We obtain two equations per 2D – 3D correspondence, linear in the unknowns p_{ij}

Camera Calibration From Projection Matrix

A linear system of equations is then obtained for N points:

$$\boldsymbol{A}\begin{pmatrix} p_{11} \\ p_{12} \\ \dots \\ p_{34} \end{pmatrix} = \boldsymbol{0}$$

$\boldsymbol{A}$ has

- 12 columns and 2N rows (for N points)
- but rank 11

→ there is a non-trivial solution to the system, which can be found using numerical methods (e.g. SVD)

Camera Calibration From Projection Matrix

Now: Decompose projection matrix, assuming skew angle is 90°, i.e., $s = 0$

$$\boldsymbol{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} = \begin{pmatrix} -f_x r_{11} + p_x r_{31} & -f_x r_{12} + p_x r_{32} & -f_x r_{13} + p_x r_{33} & -f_x t_x + p_x t_z \\ -f_y r_{21} + p_y r_{31} & -f_y r_{22} + p_y r_{32} & -f_y r_{23} + p_y r_{33} & -f_y t_y + p_y t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

Let $\widehat{\boldsymbol{P}}$ = matrix estimated using calibration method.
We use homogeneous coordinates: only unique up to scale, i.e.: $\widehat{\boldsymbol{P}} = \gamma \boldsymbol{P}$

Determine **scaling factor** γ. Use the fact that rotation matrix is orthonormal: $|\gamma| = |\gamma|\sqrt{r_{31}^2 + r_{32}^2 + r_{33}^2} = \sqrt{\hat{p}_{31}^2 + \hat{p}_{32}^2 + \hat{p}_{33}^2}$

Sign of γ:

- positive if origin of world coordinate system is in front of camera: $p_{34} = t_z > 0$
- negative if origin of world coordinate system is behind the camera: $p_{34} = t_z < 0$

Divide each entry of $\widehat{\boldsymbol{P}}$ by γ to obtain a correctly scaled matrix $\boldsymbol{P}$

Camera Calibration From Projection Matrix

$$\boldsymbol{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^T & \\ \boldsymbol{p}_2^T & \boldsymbol{p}_4 \\ \boldsymbol{p}_3^T & \end{pmatrix} = \begin{pmatrix} -f_x r_{11} + p_x r_{31} & -f_x r_{12} + p_x r_{32} & -f_x r_{13} + p_x r_{33} & -f_x t_x + p_x t_z \\ -f_y r_{21} + p_y r_{31} & -f_y r_{22} + p_y r_{32} & -f_y r_{23} + p_y r_{33} & -f_y t_y + p_y t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

intrinsic parameters:

$$p_x = \boldsymbol{p}_1^T \boldsymbol{p}_3 \qquad p_y = \boldsymbol{p}_2^T \boldsymbol{p}_3$$

$$f_x = \sqrt{\boldsymbol{p}_1^T \boldsymbol{p}_1 - p_x^2} \qquad f_y = \sqrt{\boldsymbol{p}_2^T \boldsymbol{p}_2 - p_y^2}$$

extrinsic parameters:

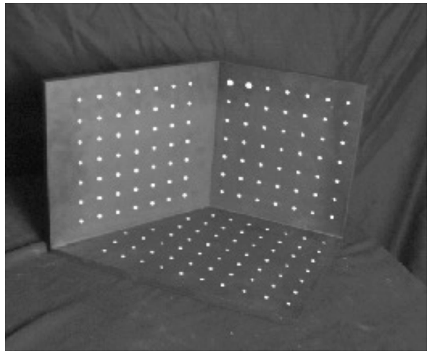
$$t_x = \frac{p_x t_z - p_{14}}{f_x} \qquad t_y = \frac{p_y t_z - p_{24}}{f_y} \qquad t_z = p_{34}$$

$$r_{1i} = \frac{p_x p_{3i} - p_{1i}}{f_x} \qquad r_{2i} = \frac{p_y p_{3i} - p_{2i}}{f_y} \qquad r_{3i} = p_{3i}$$

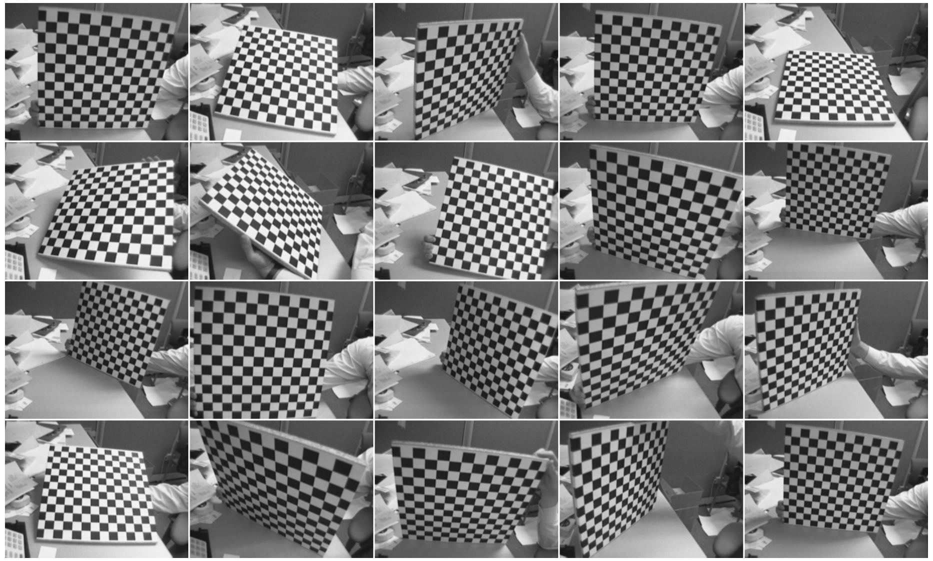
Further improvement: Nonlinear optimization of re-projection error with Gauss-Newton/Levenberg-Marquardt

Practical Issues

- we also want to consider lens distortions
- the approach requires non-coplanar points, i.e., a 3D pattern
- 3D pattern is hard to manufacture
- the 3D feature positions are difficult to measure



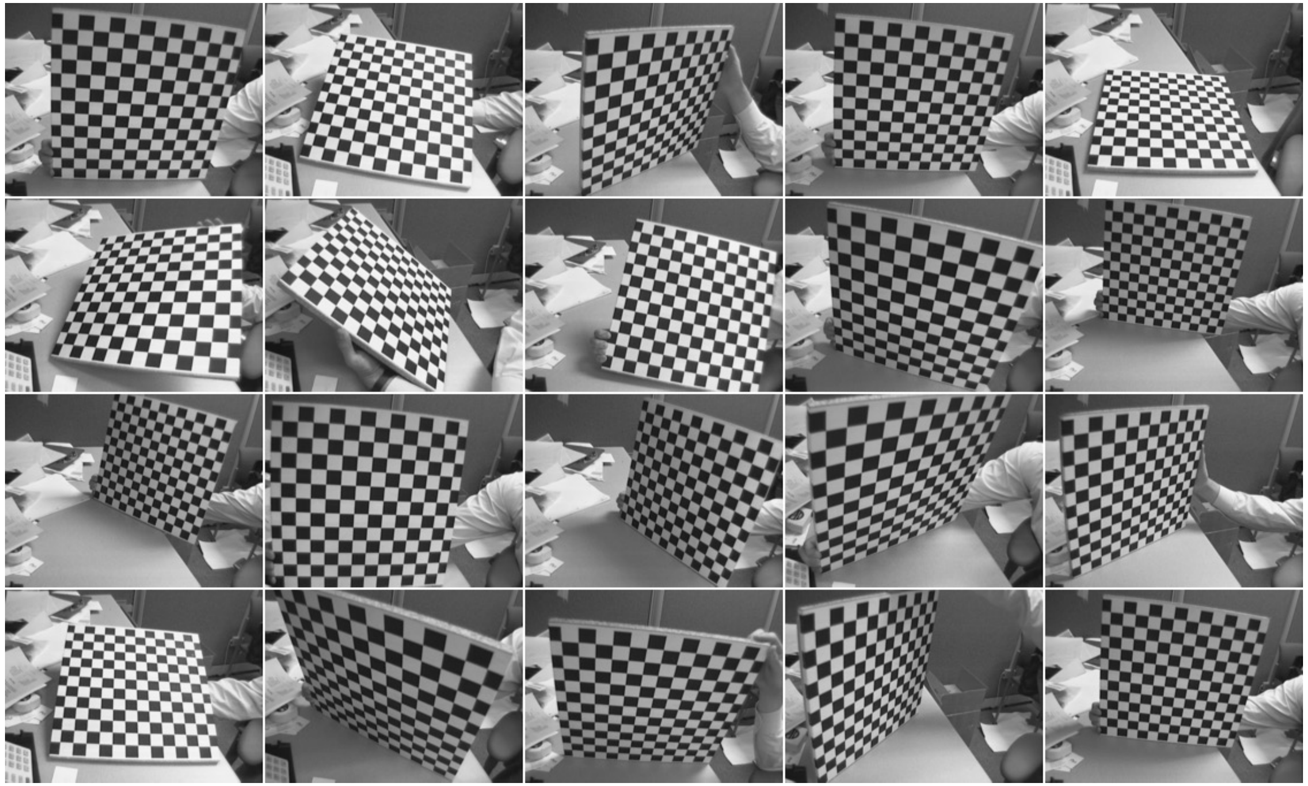
→ use planar pattern method (like the one from Zhang), take several calibration images



Code/Tools:

- OpenCV library: https://docs.opencv.org/4.10.0/dc/dbb/tutorial_py_calibration.html
- Matlab Calibration Toolbox

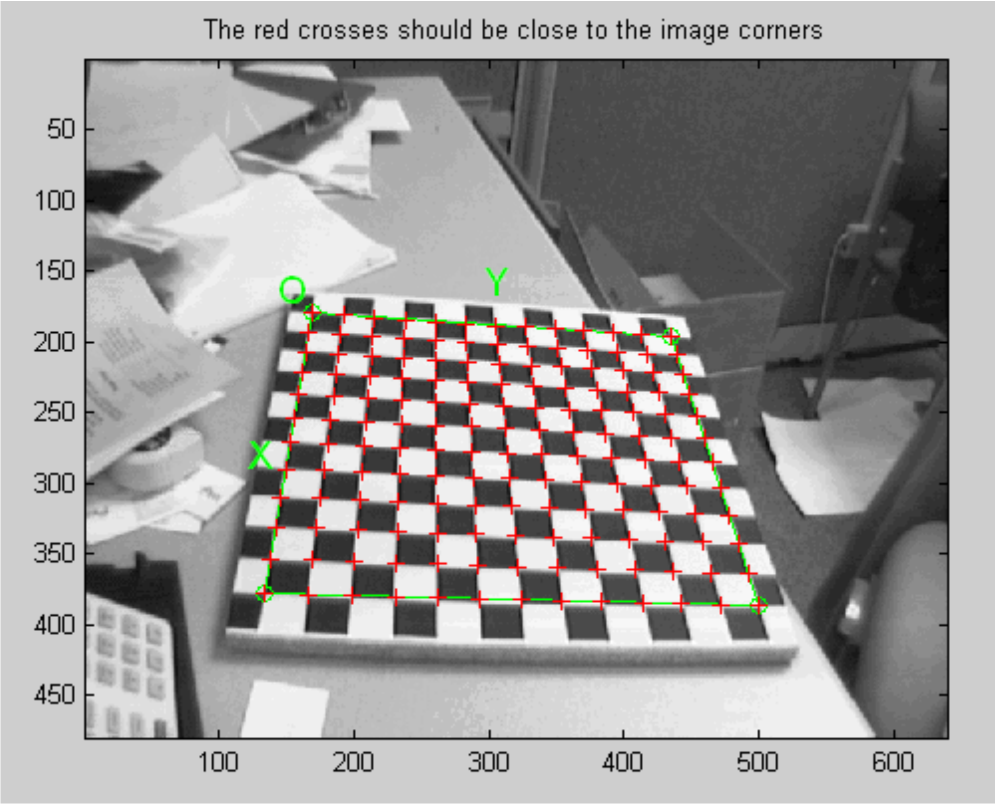
Data Acquisition

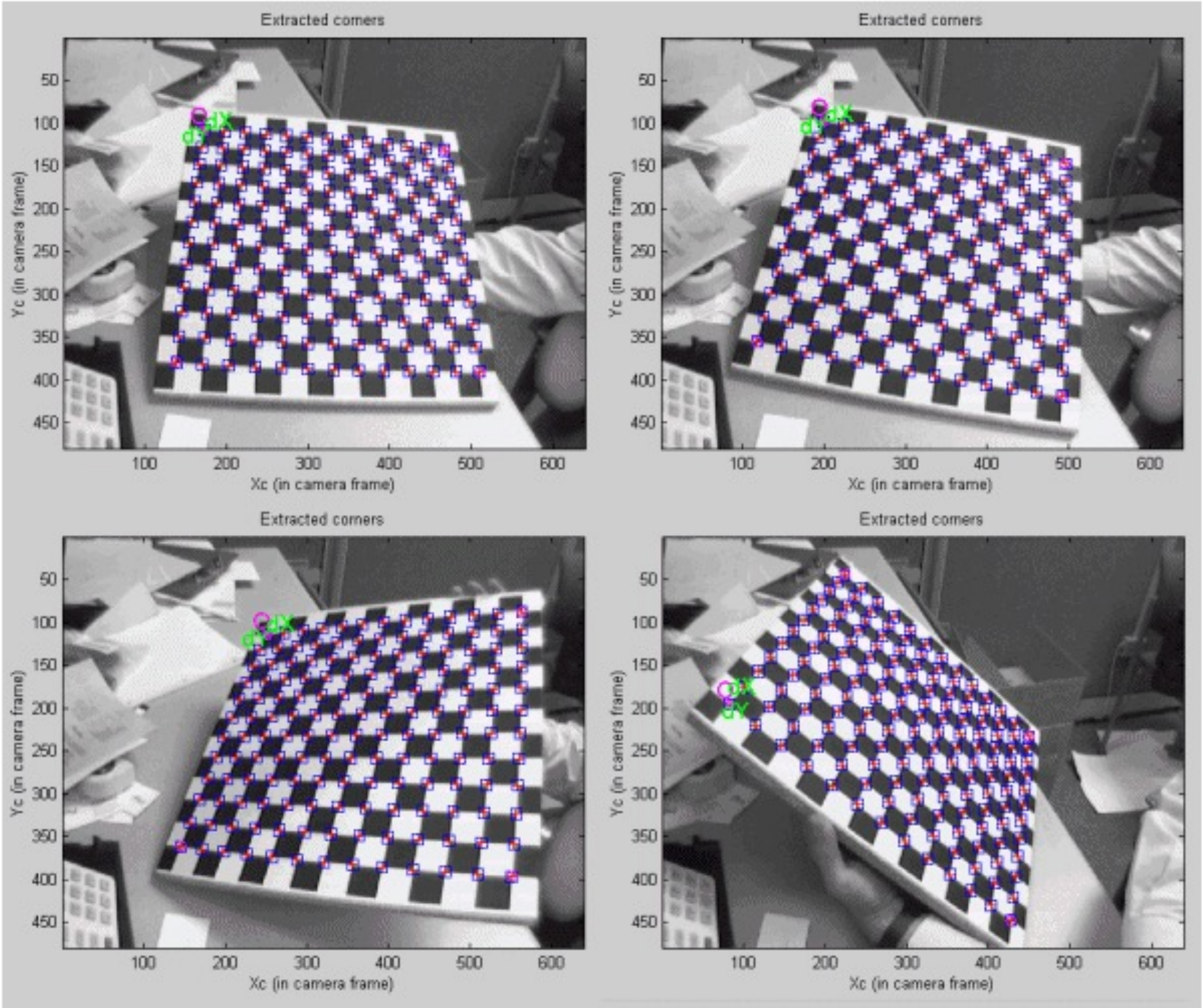


Brian moore81, Multiple chessboard views, CC BY-SA 4.0

For an exact estimation of radial distortions: recording images of the calibration pattern at the edges and corners of the visible area are particularly important

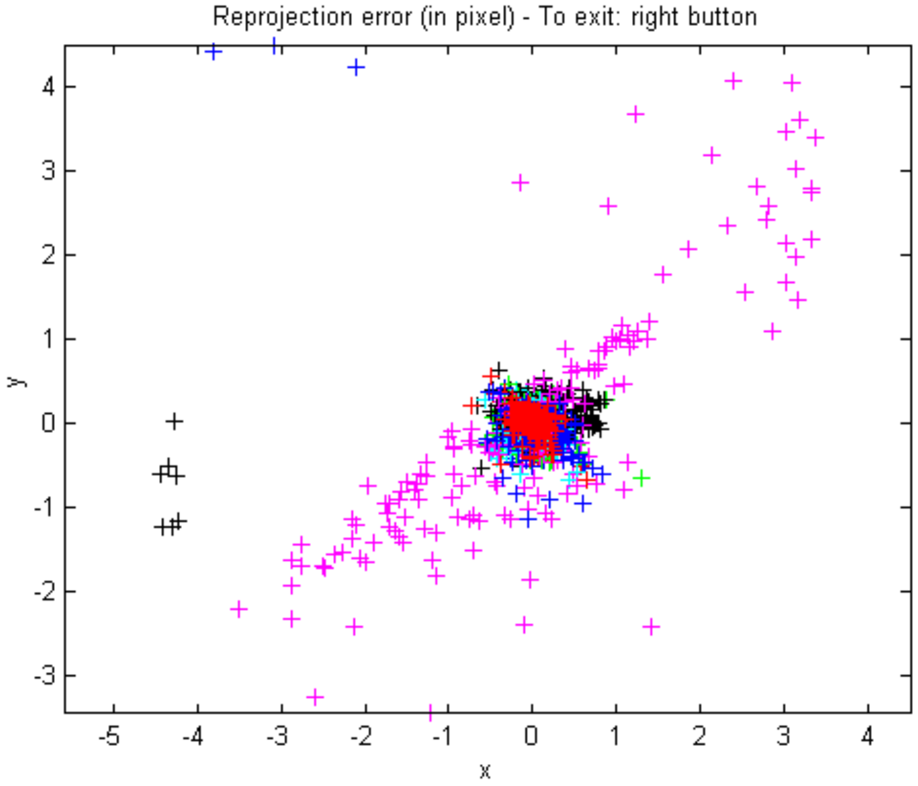
Corner Extraction





slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

Calibration

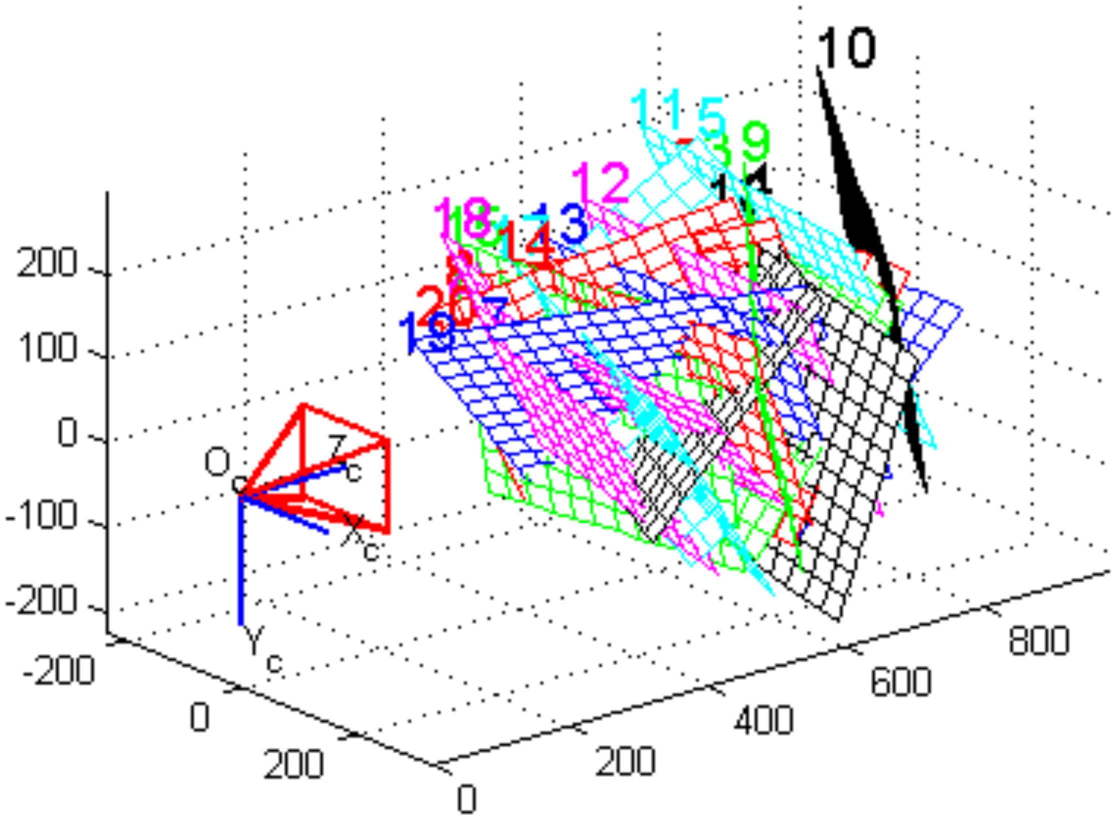


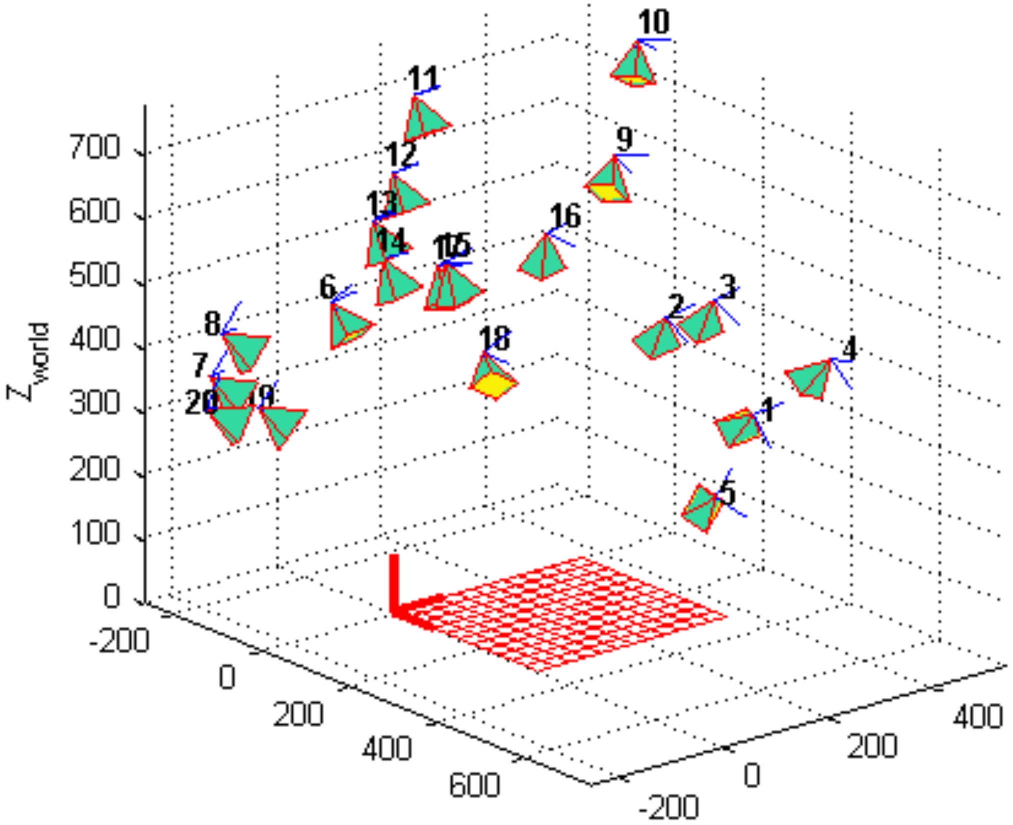

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 657.46290   657.94673 ] ± [ 0.31819   0.34046 ]
Principal point:       cc = [ 303.13665   242.56935 ] ± [ 0.64682   0.59218 ]
Skew:             alpha_c = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes =
Distortion:            kc = [ -0.25403   0.12143   -0.00021   0.00002  0.00000 ]
Pixel error:          err = [ 0.11689   0.11500 ]
```

slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

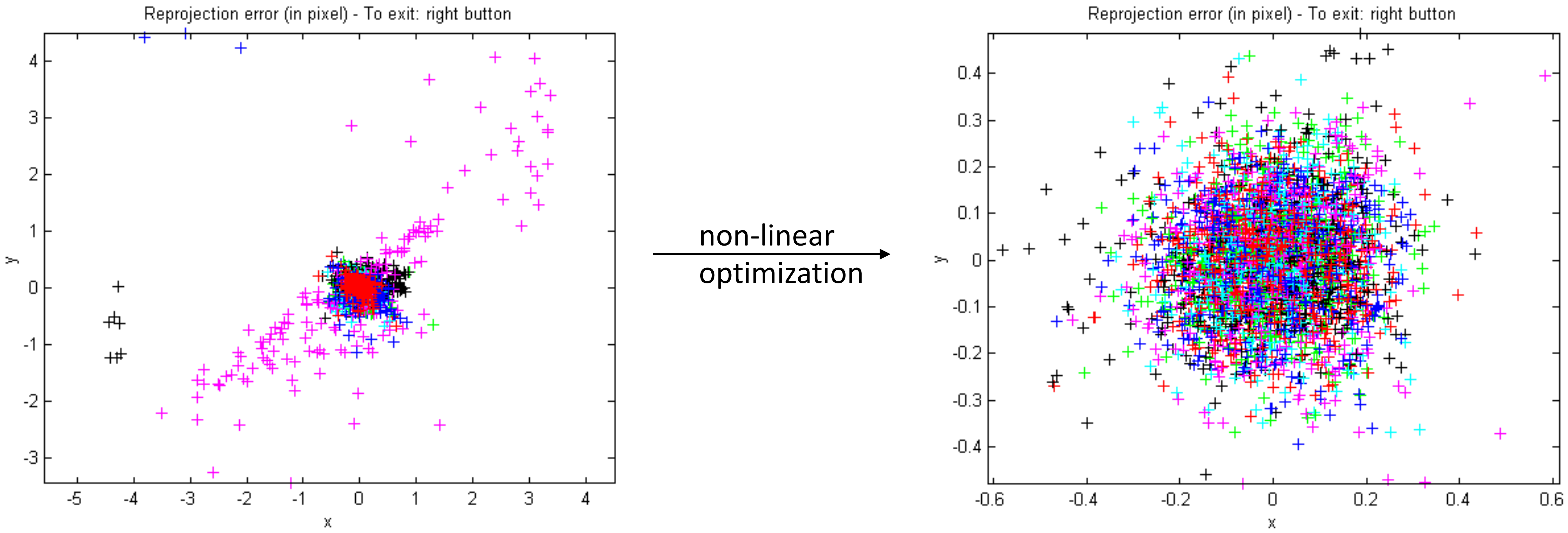
Result





slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

Refine Reprojection Error



slide credit: Li Zhang, CS 766, University of Wisconsin-Madison

Sources

- Beyerer, J., Puente Leon, F., Frese, Ch.: Automatic visual inspection, Springer Vieweg, 2012
- Li Zhang, CS 766 - Computer Vision, University of Wisconsin-Madison
- Linda Shapiro, ECE/CS 576 - Computer Vision, University of Washington
- Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22(11):1330–1334, 2000.