Camera calibration

References

- http://szeliski.org/Book/
- http://www.cs.cornell.edu/courses/cs5670/2019sp/lectures/lectures.html
- http://www.cs.cmu.edu/~16385/

Contents

- **What is camera calibration?**
- Camera intrinsics
- Camera extrinsics
- Full camera matrix
- Calibration methods and distortions

What is camera calibration

- **Geometric camera calibration**, also referred to as **camera resectioning**, estimates the parameters of a lens, image sensor, position and view direction of a perspective camera.
- You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction.
- [from: https://www.mathworks.com/help/vision/ug/camera-calibration.html]

Starting from the end

- The camera matrix is a full transformation from 3D objects in the scene to a 2D image with the specific camera parameters:

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

$$P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \mapsto \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \end{bmatrix}$$

- How many DOFs do we have?

Starting from the end

- The camera matrix is a full transformation from 3D objects in the scene to a 2D image with the specific camera parameters:

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

$$P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \mapsto \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \end{bmatrix}$$

- How many DOFs do we have?
  - 11, because in homogenous coordinates the answer is always correct up to a scale.

Starting from the end

$$P_{3X4} = K_{3X3}[I|0]_{3X4}\Pi_{4X4}$$

Intrinsic camera matrix: estimates the parameters of the lens and image sensor.

Perspective projection matrix: Project from 3D to 2D as seen before:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

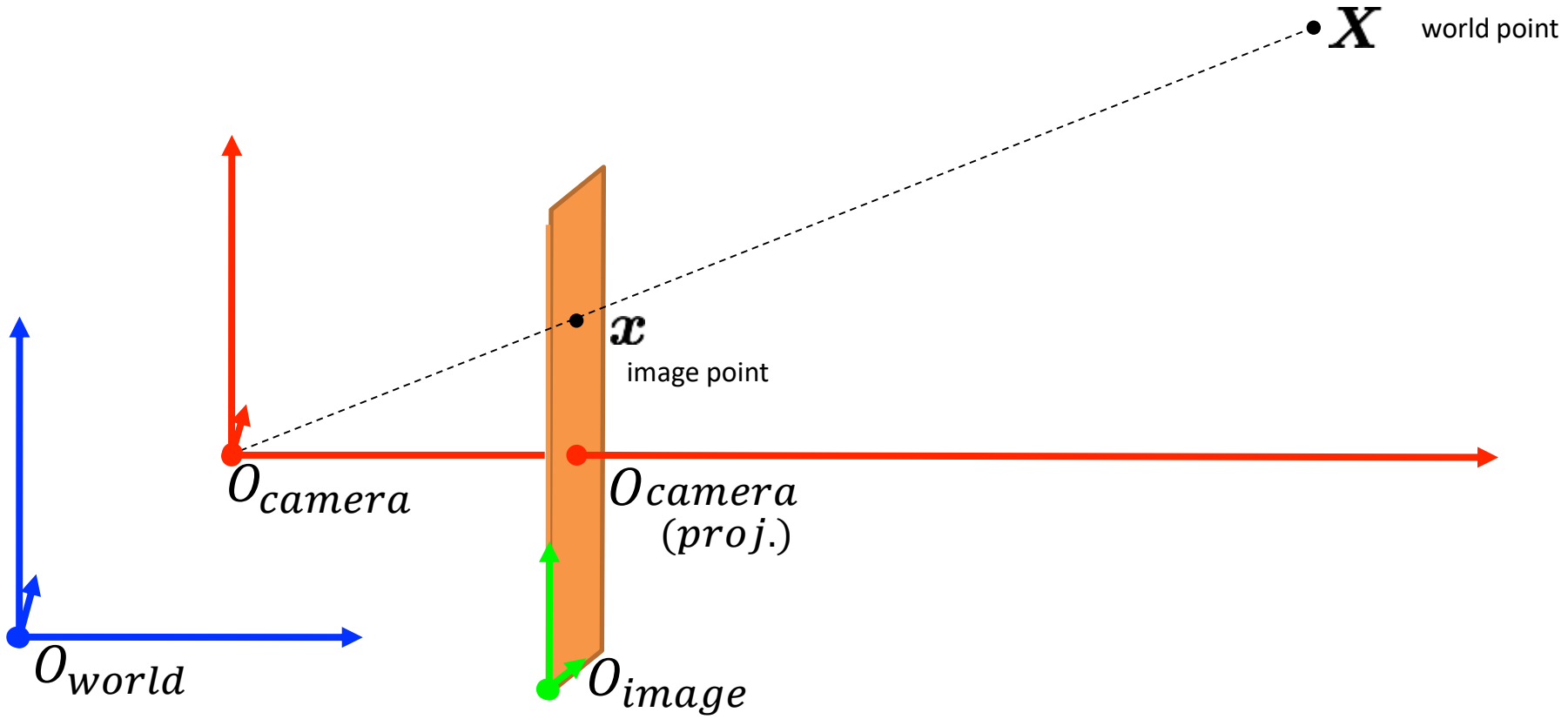
extrinsic camera matrix: estimates the parameters of position and view direction of a camera.

Contents

- What is camera calibration?
- **Camera intrinsics**
- Camera extrinsics
- Full camera matrix
- Calibration methods and distortions

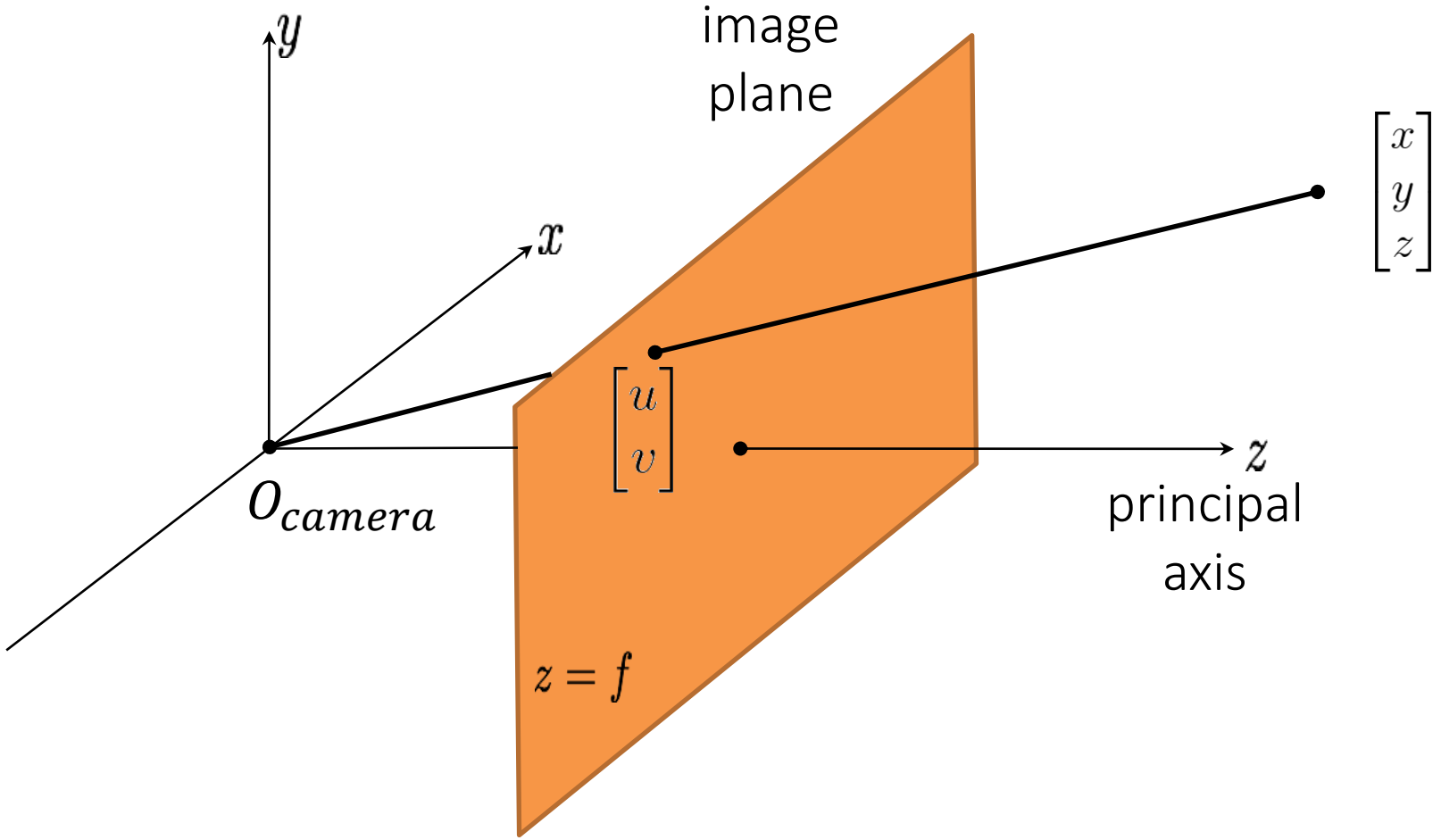
Coordinate systems

- There are 3 coordinate systems that are discussed in general:



Recap: perspective projection

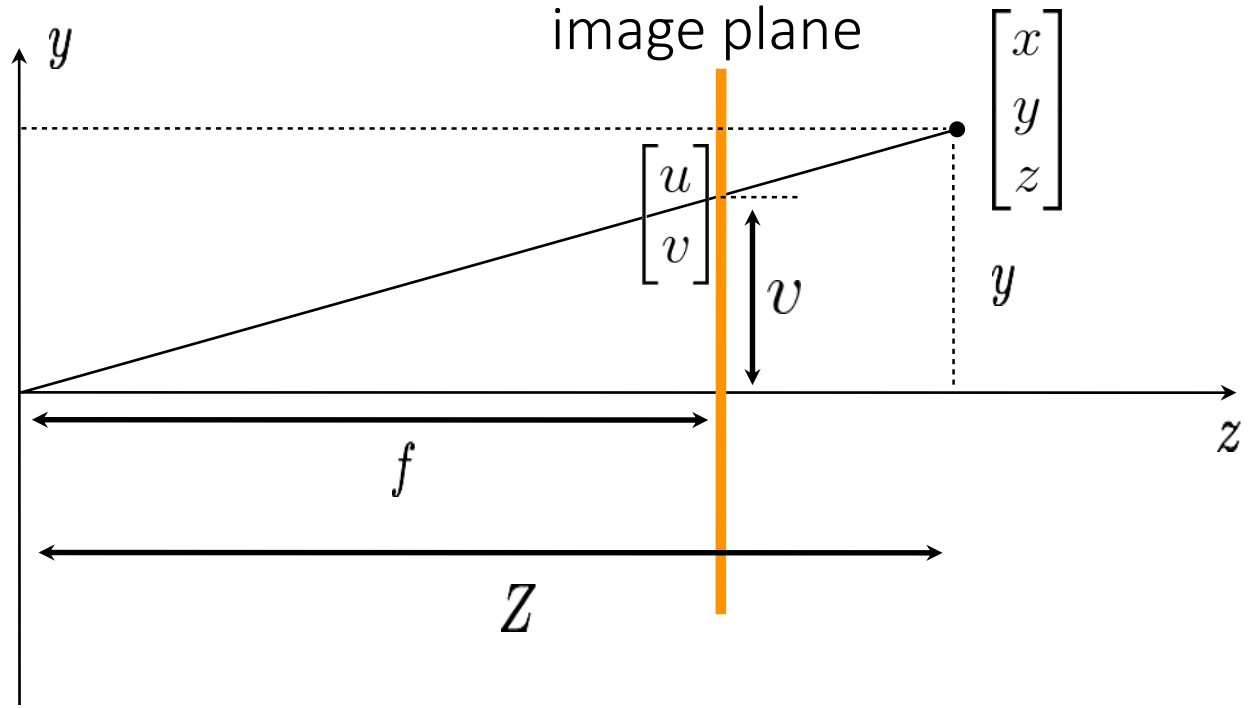
- **Perspective projection** (also known as **perspective transformation**) is a linear projection where three dimensional objects are projected on the image plane.



Recap: perspective projection

- Using triangle proportions (Thales' theorem) we can easily conclude that:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{bmatrix}$$



Recap: perspective projection

- Let's use the homogeneous coordinates:

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} \mapsto \begin{bmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{bmatrix}$$

  - Units of $[m]$

Recap: perspective projection

- Let's split into 2 matrices and use 3D->2D homogenous coordinates:

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} \mapsto \begin{bmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{bmatrix}$$

Intrinsic camera matrix

Perspective projection matrix

Intrinsic camera matrix

- The intrinsic matrix $\boldsymbol{K}$ contains 5 intrinsic parameters. These parameters encompass:
  - Scaled x & y focal length.
  - Sensor skew.
  - Principal point.
- The intrinsic camera matrix transforms a point in general image plane 2D space to the camera specific image space.

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic camera matrix

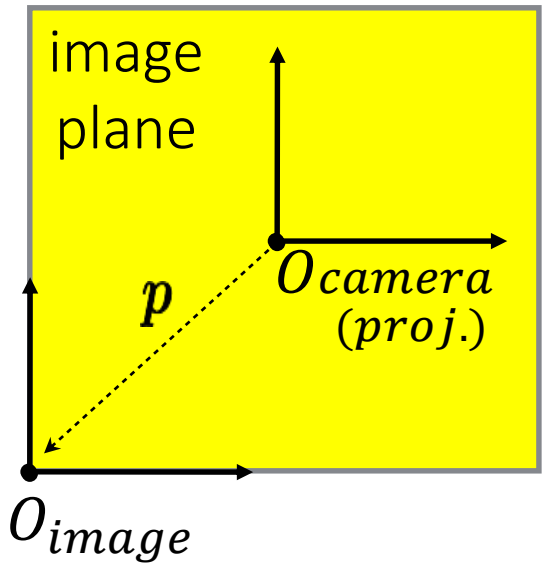
- Transforming to units of pixels in image space:
  - Pixel size in x dimension is m_x and Pixel size in y dimension is m_y.

$$f_x = \frac{f}{m_x} \;\&\; f_y = \frac{f}{m_y}$$

$$K = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic camera matrix

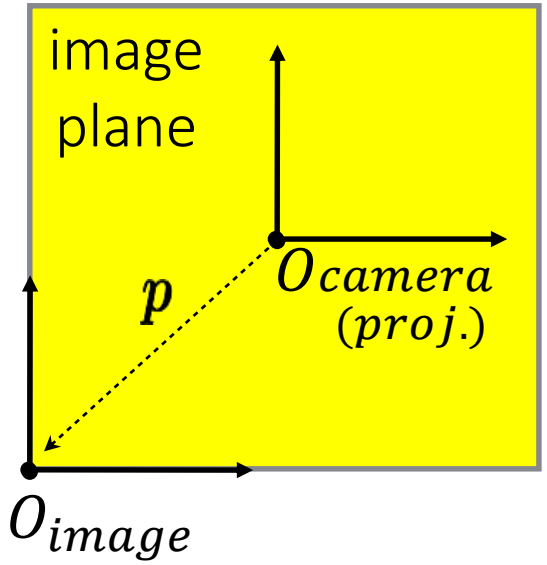
- Let's add the **principle point**: the offset vector between the projected camera coordinates to the image coordinate [units of pixels].



- How to add this to the intrinsic matrix?

Intrinsic camera matrix

- Let's add the **principle point**: the offset vector between the projected camera coordinates to the image coordinate [units of pixels].



- How to add this to the intrinsic matrix?

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic camera matrix

- In some camera sensors exist a very small skew which makes the sensor a parallelogram.

$$K = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Contents

- What is camera calibration?
- Camera intrinsics
- **Camera extrinsics**
- Full camera matrix
- Calibration methods and distortions

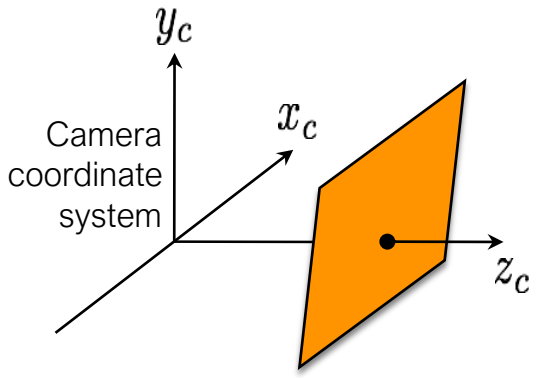
Extrinsic camera matrix

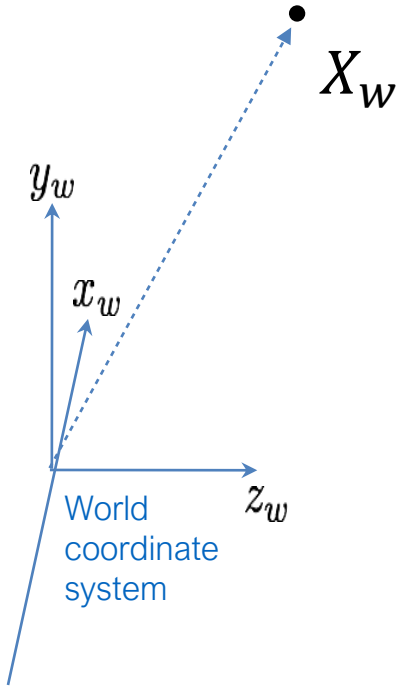
- The extrinsic camera matrix is a concatenation of a rotation and translation matrix from O_{world} to O_{camera}
- We need the world coordinate system when we are talking about multiple camera setup and the relations between one another.
- We are given a point in world coordinates and we transform it to the camera coordinate system

Extrinsic camera matrix

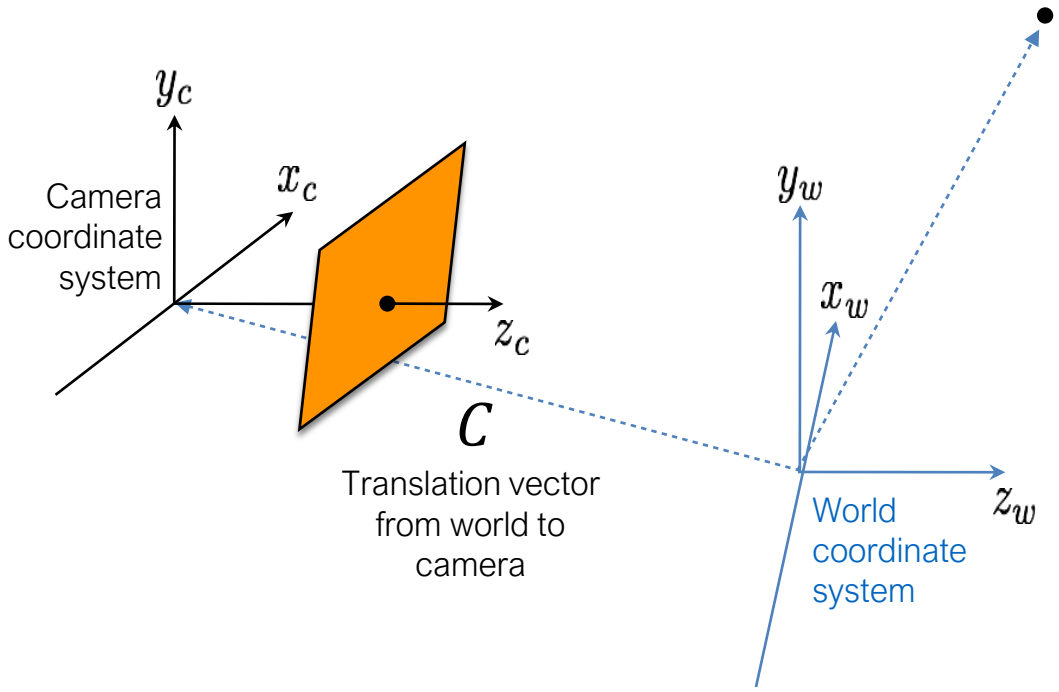
- We are given a point in world coordinates and we transform it to the camera coordinate system:

Extrinsic camera matrix

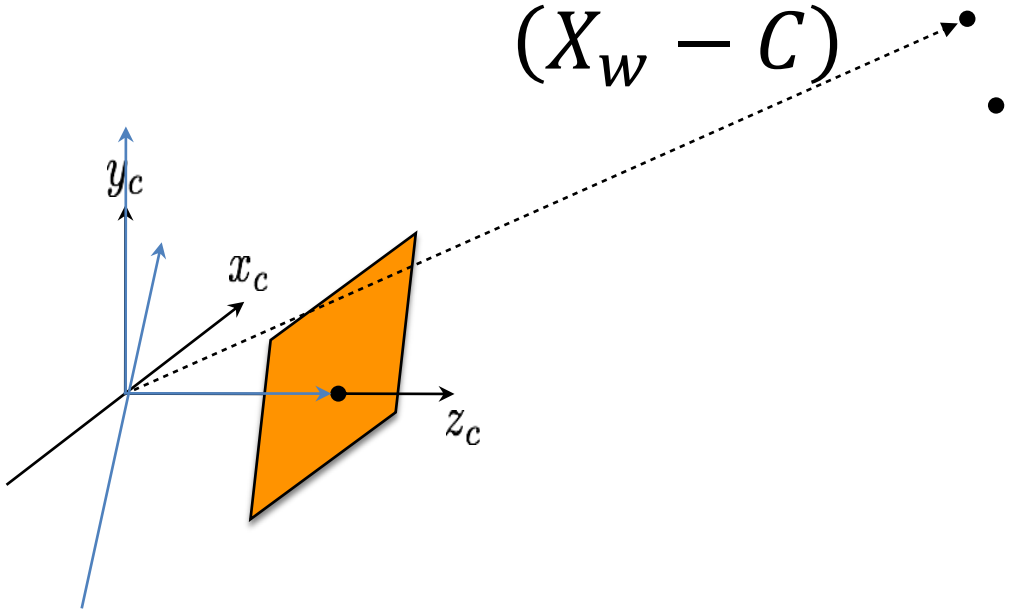




Extrinsic camera matrix



Extrinsic camera matrix

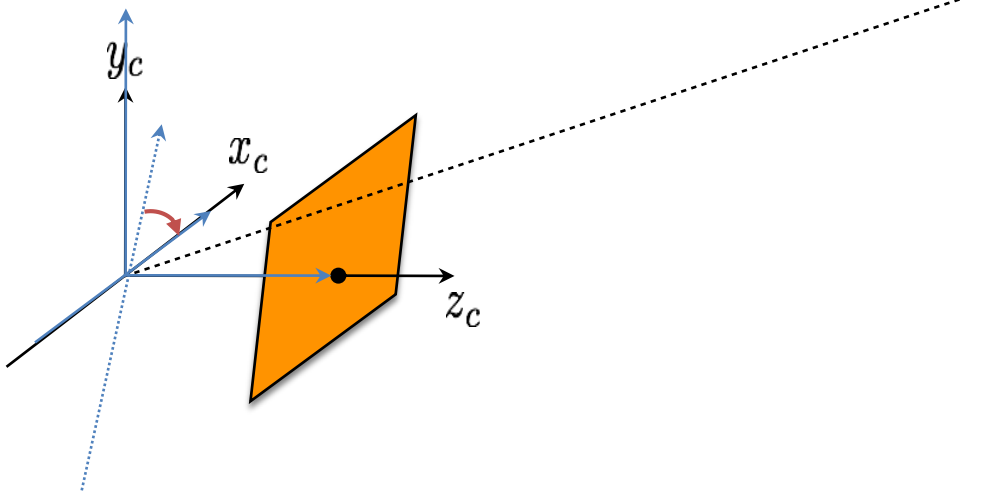


Extrinsic camera matrix

$$X_c = R(X_w - C)$$

R

Rotation matrix from world to camera



Extrinsic camera matrix

- $X_c = R(X_w - C)$
- Transform to homogenous coordinate notation:

$$X_c = \begin{bmatrix} R_{3X3} & -RC_{3X1} \\ 0_{1X3} & 1 \end{bmatrix} X_w$$

- Translation part: 3 DOFs.
- Rotation part: 3 DOFs (θ_x, θ_y, θ_z).
- In OpenCV they do a different transformation that is essentially:

$$X_c = RX_w + t$$

Contents

- What is camera calibration?
- Camera intrinsics
- Camera extrinsics
- **Full camera matrix**
- Calibration methods and distortions

Full camera matrix

$$P = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3X3} & -RC_{3X1} \\ 0_{1X3} & 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3X3} & -RC_{3X1} \\ 0_{1X3} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Full camera matrix

- Assuming we are given an imaged points and their corresponding 3D points in the real world, **camera calibration** is the process to find the camera parameters.
  - We will return to this assumption later.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Full camera matrix

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} — & \boldsymbol{p}_1^\top & — \\ — & \boldsymbol{p}_2^\top & — \\ — & \boldsymbol{p}_3^\top & — \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

Heterogeneous coordinates

$$x' = \frac{\boldsymbol{p}_1^\top \boldsymbol{X}}{\boldsymbol{p}_3^\top \boldsymbol{X}} \qquad y' = \frac{\boldsymbol{p}_2^\top \boldsymbol{X}}{\boldsymbol{p}_3^\top \boldsymbol{X}}$$

(non-linear relation between coordinates)

Full camera matrix

$$x' = \frac{\boldsymbol{p}_1^\top \boldsymbol{X}}{\boldsymbol{p}_3^\top \boldsymbol{X}} \qquad y' = \frac{\boldsymbol{p}_2^\top \boldsymbol{X}}{\boldsymbol{p}_3^\top \boldsymbol{X}}$$

Make them linear with algebraic manipulation…

$$\boldsymbol{p}_2^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} y' = 0$$

$$\boldsymbol{p}_1^\top \boldsymbol{X} - \boldsymbol{p}_3^\top \boldsymbol{X} x' = 0$$

Now we can setup a system of linear equations with multiple point correspondences

Full camera matrix

$$p_2^\top X - p_3^\top X y' = 0$$

$$p_1^\top X - p_3^\top X x' = 0$$

Vector of 1X12

In matrix form …

$$\begin{bmatrix} X^\top & 0 & -x'X^\top \\ 0 & X^\top & -y'X^\top \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = 0$$

Full camera matrix

$$p_2^\top X - p_3^\top X y' = 0$$

$$p_1^\top X - p_3^\top X x' = 0$$

In matrix form …

$$\begin{bmatrix} X^\top & 0 & -x'X^\top \\ 0 & X^\top & -y'X^\top \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = 0$$

For N points …

$$\begin{bmatrix} X_1^\top & 0 & -x'X_1^\top \\ 0 & X_1^\top & -y'X_1^\top \\ \vdots & \vdots & \vdots \\ X_N^\top & 0 & -x'X_N^\top \\ 0 & X_N^\top & -y'X_N^\top \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = 0$$

How many points do we need to solve this problem?

How do we solve this system?

Full camera matrix

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\mathbf{A}\boldsymbol{x}\|^2 \text{ subject to } \|\boldsymbol{x}\|^2 = 1$$

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{X}_1^\top & \mathbf{0} & -x'\boldsymbol{X}_1^\top \\ \mathbf{0} & \boldsymbol{X}_1^\top & -y'\boldsymbol{X}_1^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_N^\top & \mathbf{0} & -x'\boldsymbol{X}_N^\top \\ \mathbf{0} & \boldsymbol{X}_N^\top & -y'\boldsymbol{X}_N^\top \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix}$$

- We've already seen this minimization problem in the curve fitting class- linear TLS.
- 6 points will give us 12 equations- enough for 11 parameters.

Linear TLS -the minimization problem

- The minimization problem is:

$$\begin{cases} minimize & \beta^T X^T X \beta \\ s.t. & \beta^T \beta = 1 \end{cases}$$

- Recall eigendecomposition: $Av = \lambda v \mapsto v^T A v = \lambda$
  - Also recall that each eigenvector v is normalized $(||v|| = v^T v = 1)$.
- The solution to the minimization problem above is the eigenvector corresponding to smallest eigenvalue of $X^T X$.

- **Watch out:** trying to minimize the problem above without the constraint $\beta^T \beta = 1$ will result with the trivial solution of $\beta = 0$.

Full camera matrix

- The second part is to decompose the resulted P matrix into K, R and C.
- We can look at the resulted matrix as follows:

$$P = K[R_{3X3}| - RC_{3X1}] = [M| - MC]$$

- **Finding C:** take only the rightmost column of P and multiply it from the left by $-M^{-1}$.
- **Finding K & R:** RQ decomposition of M to upper triangular matrix and orthogonal matrix.
  - The exact definition is out of scope. Read more about it here: http://ksimek.github.io/2012/08/14/decompose/

Contents

- What is camera calibration?
- Camera intrinsics
- Camera extrinsics
- Full camera matrix
- **Calibration methods and distortions**

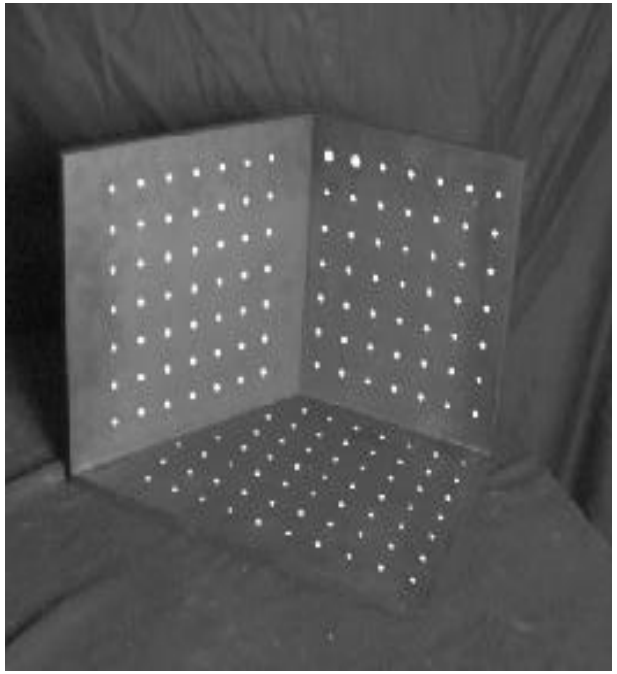
Geometric calibration

Place a known object in the scene:

- identify correspondences between image and scene
- compute mapping from scene to image

Issues:

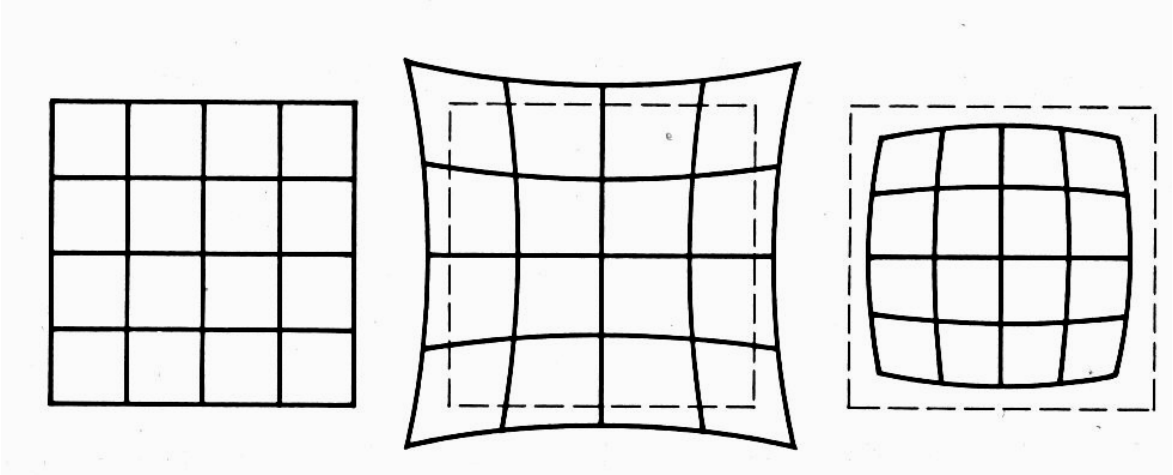
- must know geometry very accurately
- must know 3D->2D correspondence



Geometric calibration

- Advantages:
  - Very simple to formulate.
  - Analytical solution.
- Disadvantages:
  - Doesn’t model radial/ tangential distortion.
- For these reasons, *nonlinear methods* are preferred.
  - Define error function E between projected 3D points and image points.
  - E encompass intrinsics, extrinsics, radial distortion and tangential distortion.
  - Minimize E using nonlinear optimization techniques.

Radial distortion



- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

Radial distortion

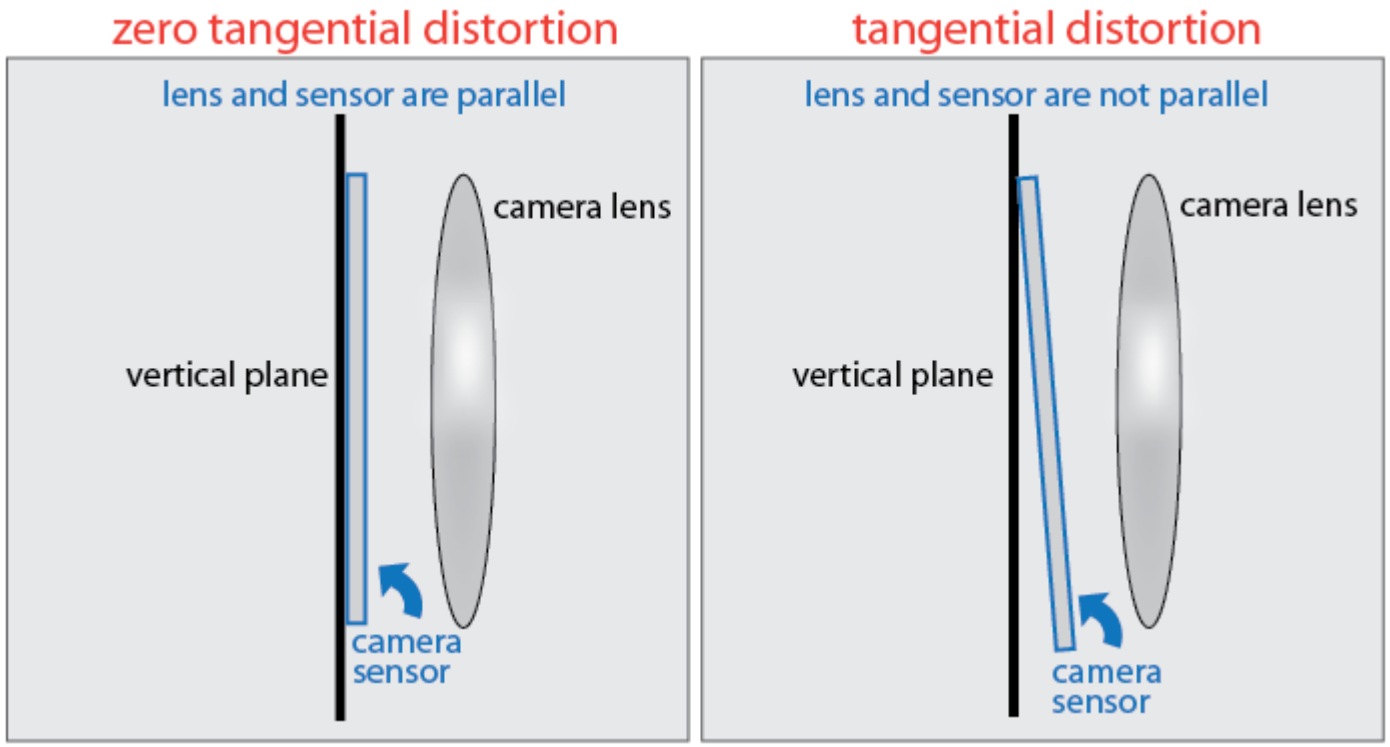
Radial distortion

before

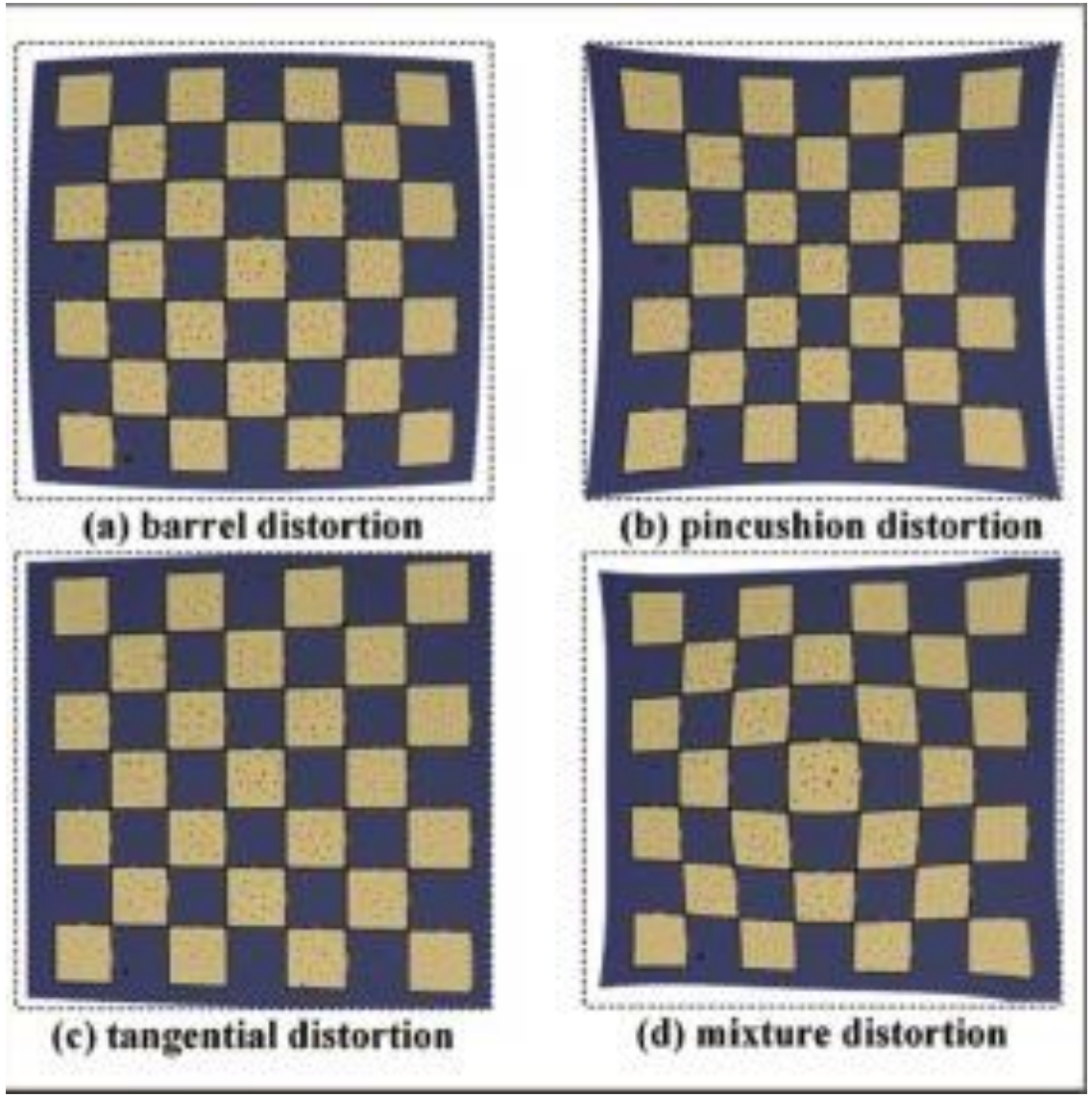
after

Tangential distortion

- Another kind of distortion caused by the camera sensor not being completely parallel to the lens and image plane.



Tangential distortion



Multi plane calibration

- Advantages:
  - Only requires a plane
  - Don’t have to know positions/orientations
- Disadvantage:
  - Need to solve non-linear optimization problem.
- https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf