Camera calibration

Yoni Chechik

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 extrinsics
- Perspective projection
- Camera intrinsics
- 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:

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

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

- 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:

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

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

- 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.

Coordinate systems

- There are 3 coordinate systems that are discussed in general: world, camera, and image coordinate systems.
  - (and also normalized image coo. System...)
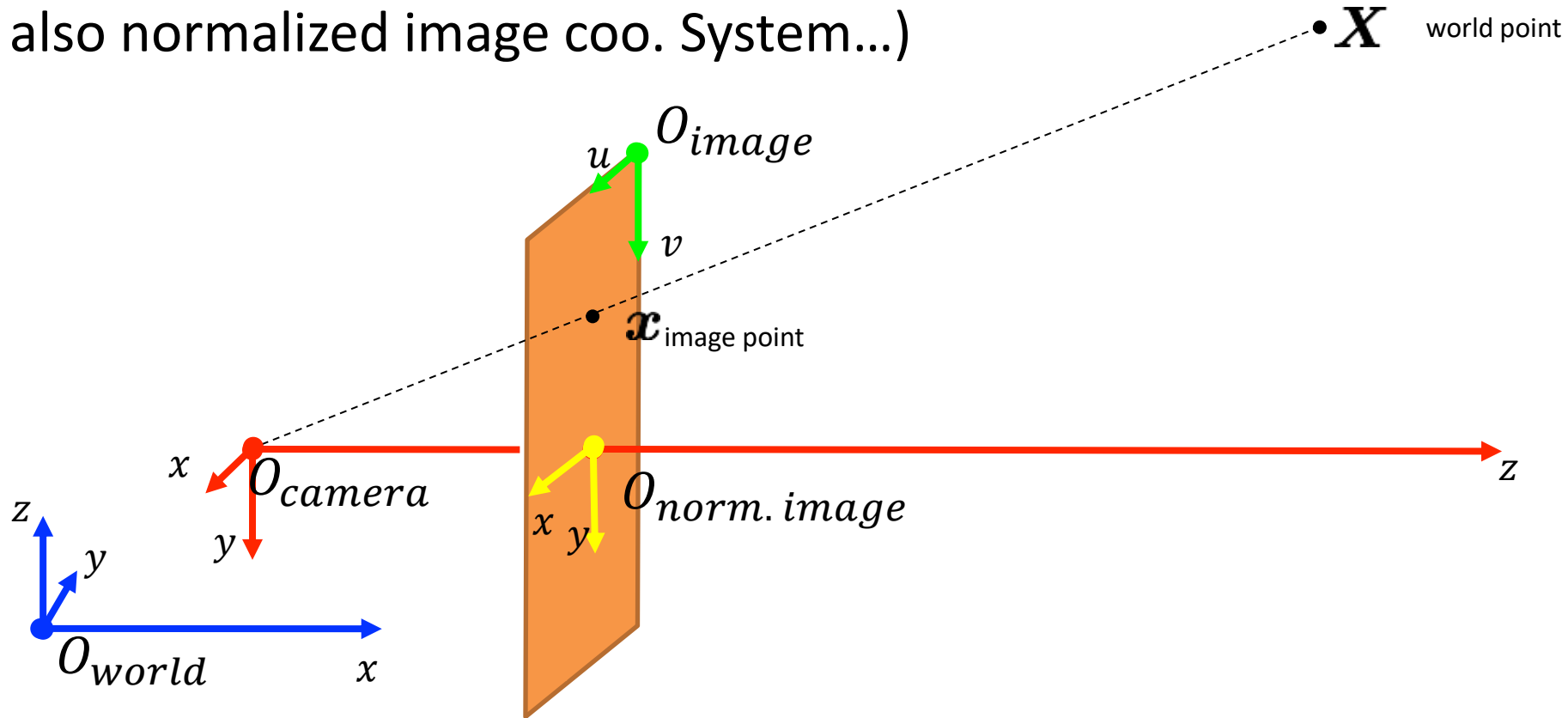
- **This is what we need to do:** $O_{world} \rightarrow O_{camera} \rightarrow O_{norm.\,image} \rightarrow O_{image}$

Contents

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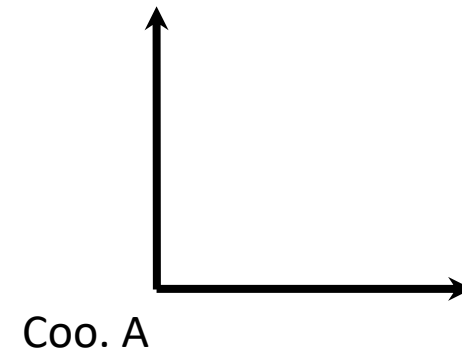- Full camera matrix
- Calibration methods and distortions

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: $O_{world} \rightarrow O_{camera}$**
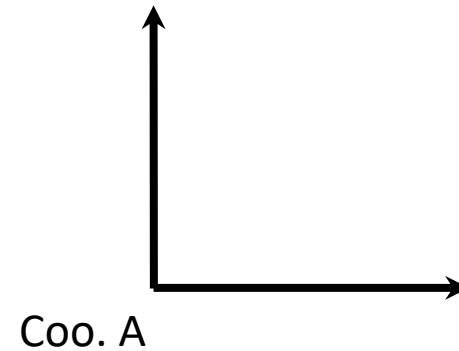
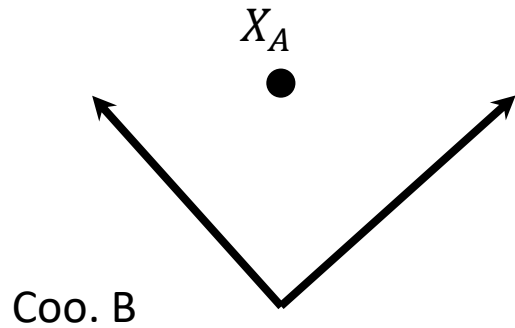# Extrinsic camera matrix

- I have a coordinate system A with a point X_A.

X_A

Coo. A

Extrinsic camera matrix

- I have a coordinate system A with a point X_A.
- <mark>How to represent X_A in coordinate B?</mark>

X_A

Coo. B

Coo. A

Extrinsic camera matrix

- I have a coordinate system A with a point X_A.
- How to represent X_A in coordinate B? **Let's change all the point's places relative to A such that we can see them as if they are in B.**
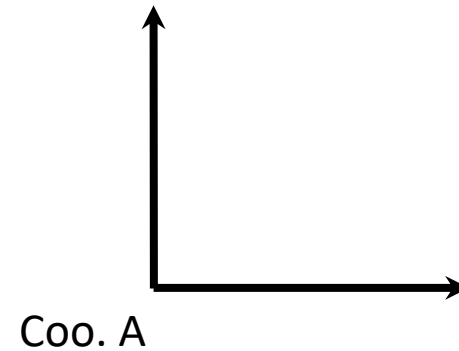
X_A

Coo. B

Coo. A

Extrinsic camera matrix

- I have a coordinate system A with a point X_A.
- How to represent X_A in coordinate B? **Let's change all the point's places relative to A such that we can see them as if they are in B.**

X_A

Coo. B

C

Translation vector

Coo. A

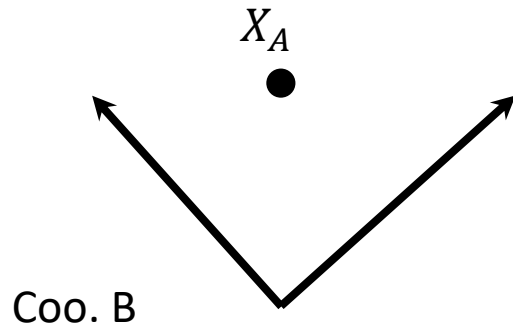# Extrinsic camera matrix

- I have a coordinate system A with a point X_A.
- How to represent X_A in coordinate B? **Let's change all the point's places relative to A such that we can see them as if they are in B.**
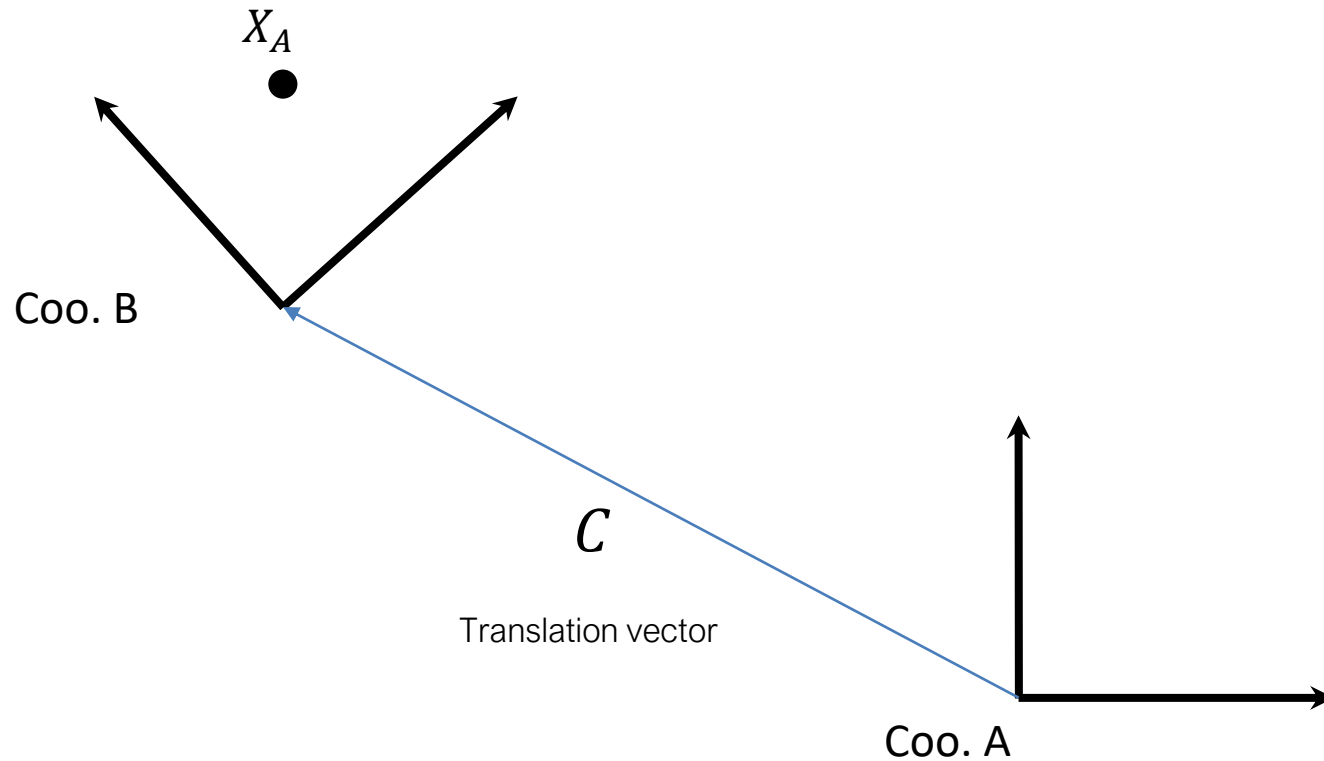
Extrinsic camera matrix

- I have a coordinate system A with a point X_A.
- How to represent X_A in coordinate B? **Let's change all the point's places relative to A such that we can see them as if they are in B.**
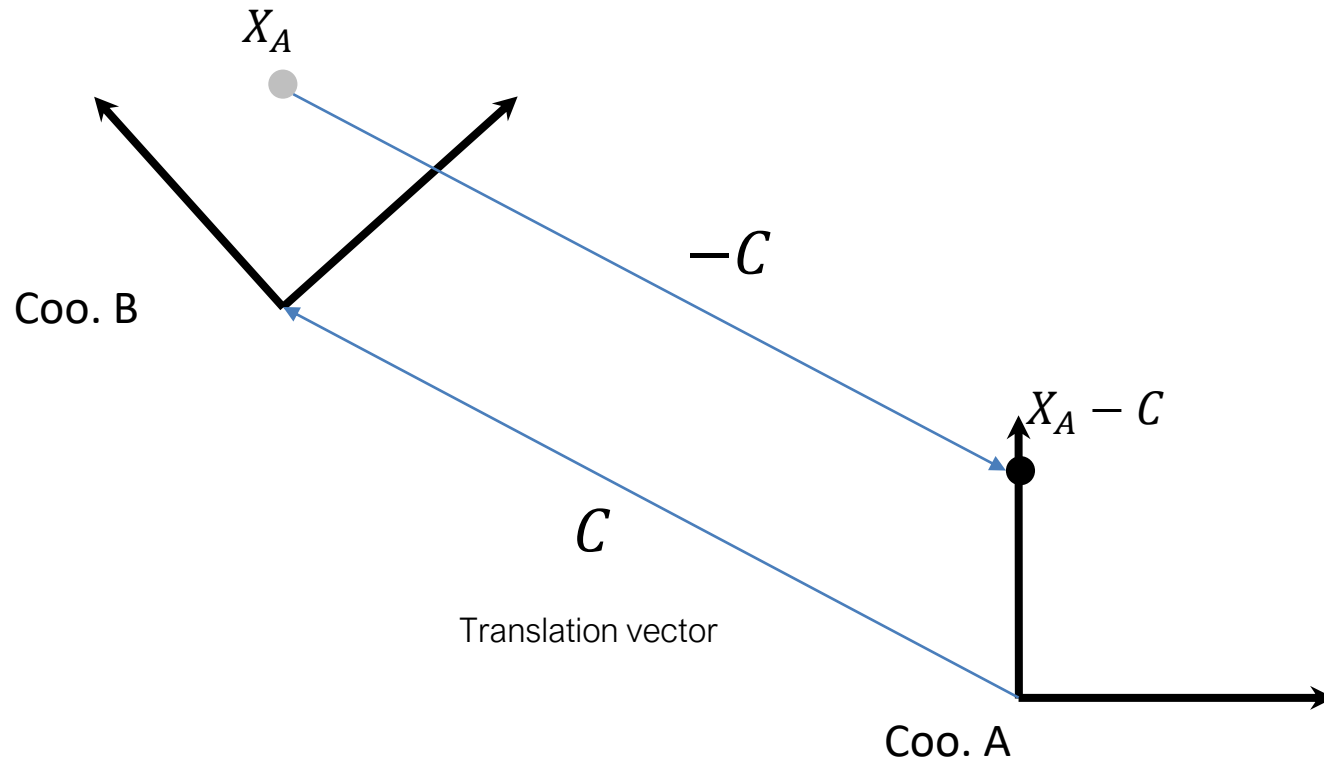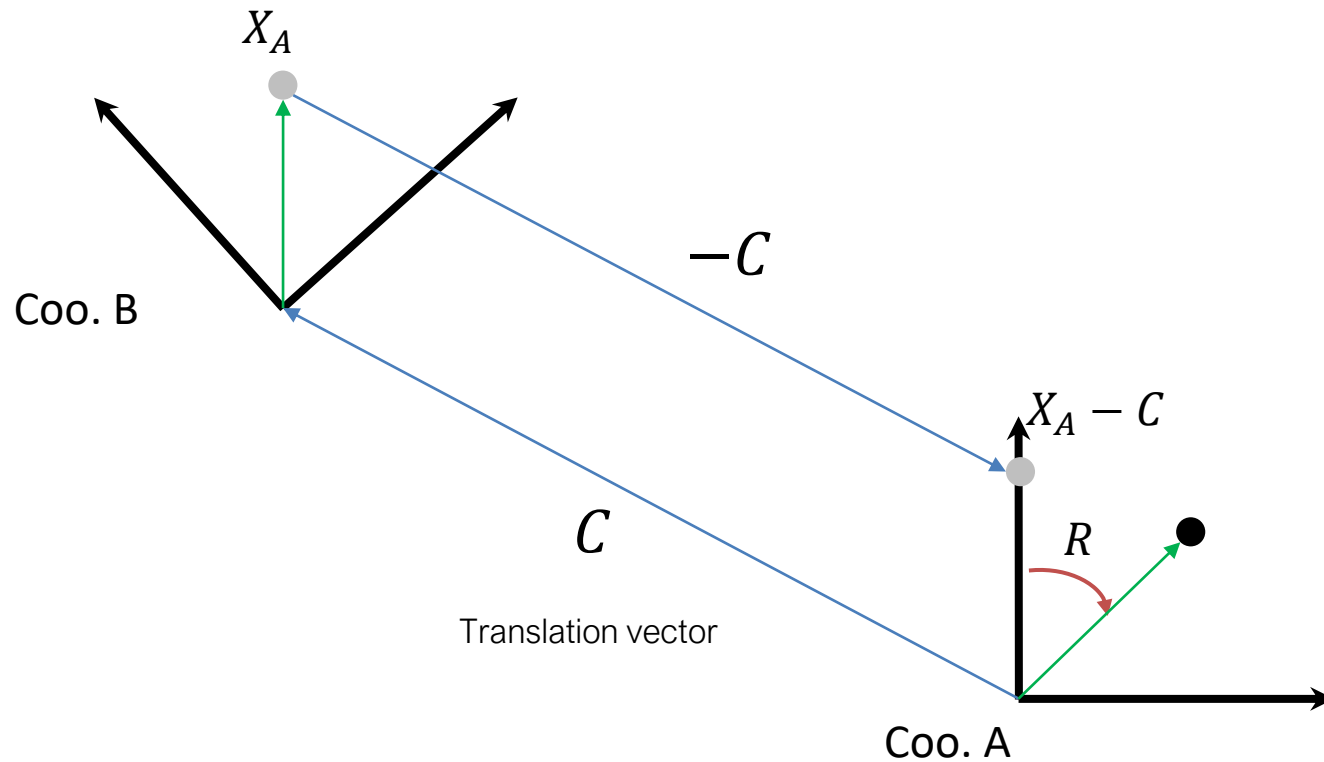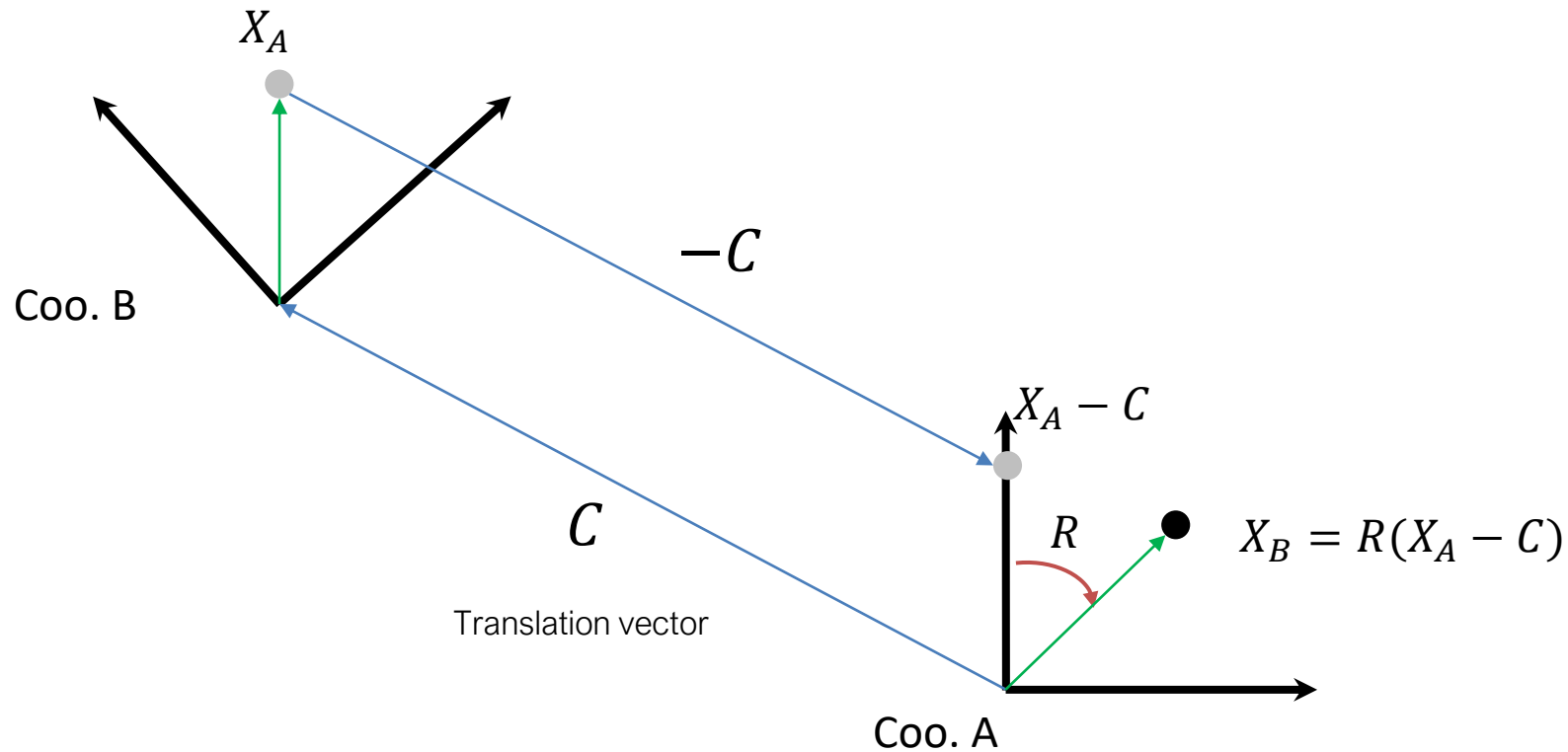
Extrinsic camera matrix

- I have a coordinate system A with a point X_A.
- How to represent X_A in coordinate B? **Let's change all the point's places relative to A such that we can see them as if they are in B.**

X_A

Coo. B

$-C$

$X_A - C$

C

R $X_B = R(X_A - C)$

Translation vector

Coo. A

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 first does a rotation and then translation:

$$X_c = RX_w + t \leftrightarrow t = -RC$$

$$X_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} X_w$$

Contents

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

Recap: perspective projection

- $O_{world} \rightarrow O_{camera}$ **Done.**
- Now: $O_{camera} \rightarrow O_{norm.\,image}$ (already saw this: 3D -> 2D perspective projection)
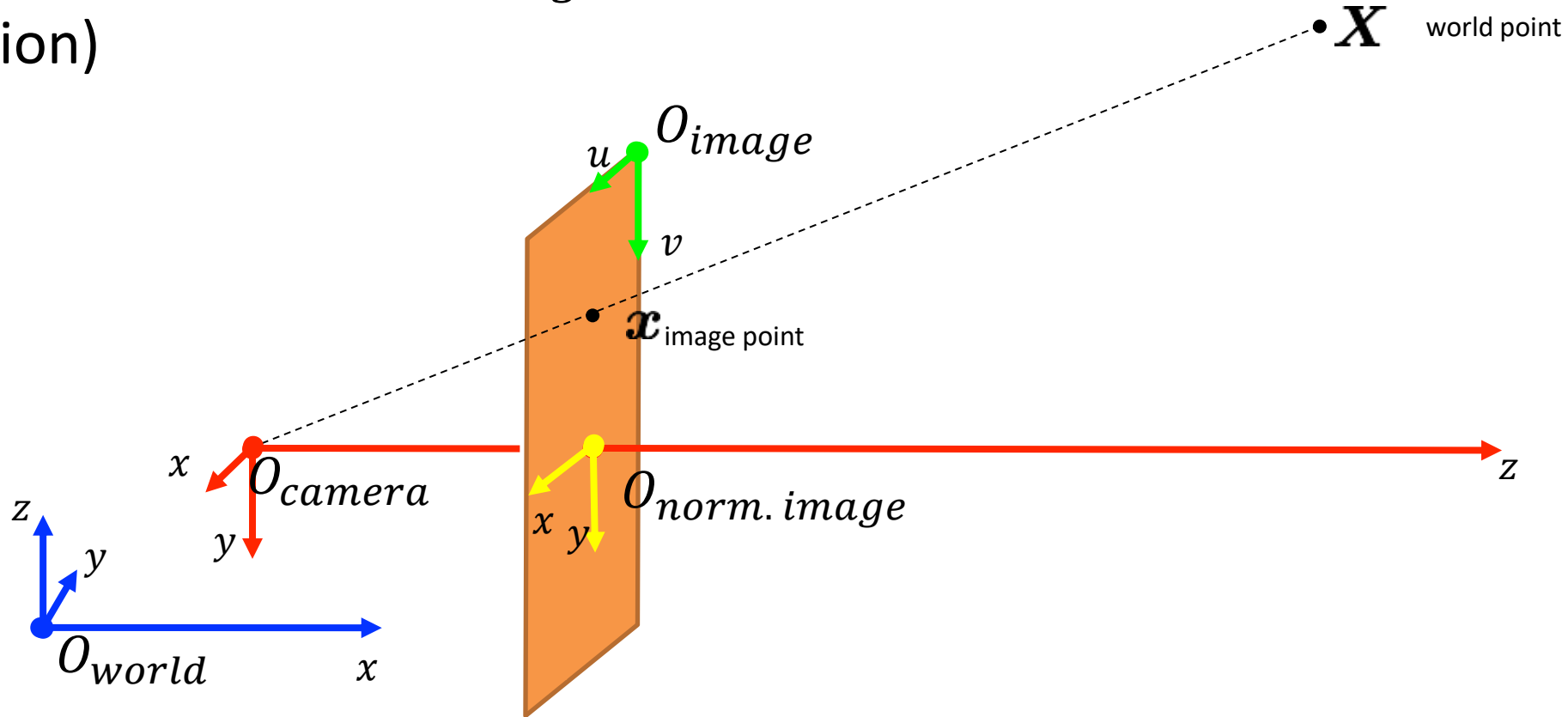
Recap: perspective projection

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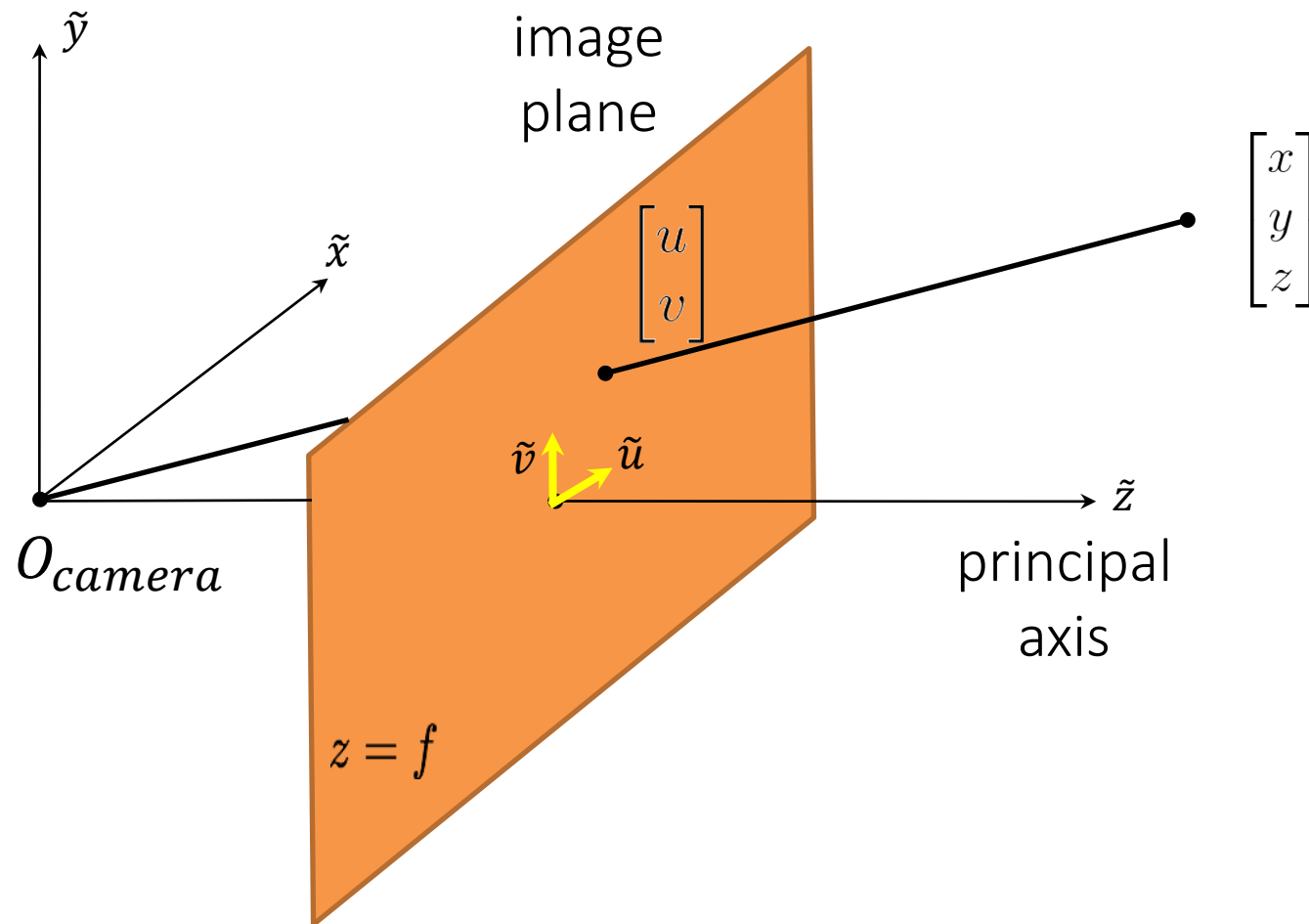
Recap: perspective projection

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

$$\begin{bmatrix} f\dfrac{x}{z} \\ f\dfrac{y}{z} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$
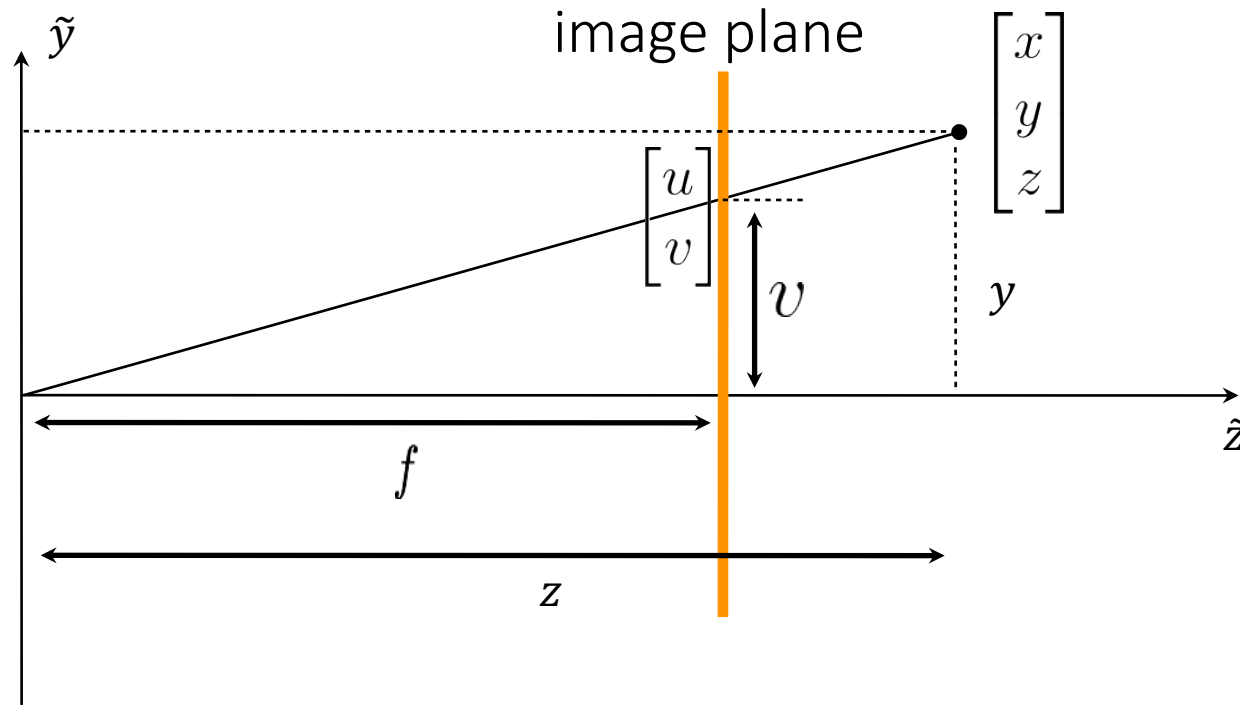
Recap: perspective projection

- Let's use the homogeneous coordinates:

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

  - Units of $[m]$

Recap: perspective projection

- Let's split into 2 matrices and use 3D->2D homogenous coordinates:
  - The perspective projection matrix transforms us from the **camera coordinate system** to the **normalized image coordinate** system.
  - The intrinsic matrix transforms us from the **normalized image space** to the **image space**.

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

Intrinsic camera matrix

Perspective projection matrix

Contents

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

Intrinsic camera matrix

- $\boldsymbol{O}_{world} \rightarrow \boldsymbol{O}_{camera} \rightarrow \boldsymbol{O}_{norm.\,image}$ **Done.**

- Now: $O_{norm.\,image} \rightarrow O_{image}$

$\bullet \boldsymbol{X}$ world point

O_{image}

u

v

$\boldsymbol{x}$ image point

x

O_{camera}

y

$O_{norm.\,image}$

x y

z

z

y

O_{world}

x
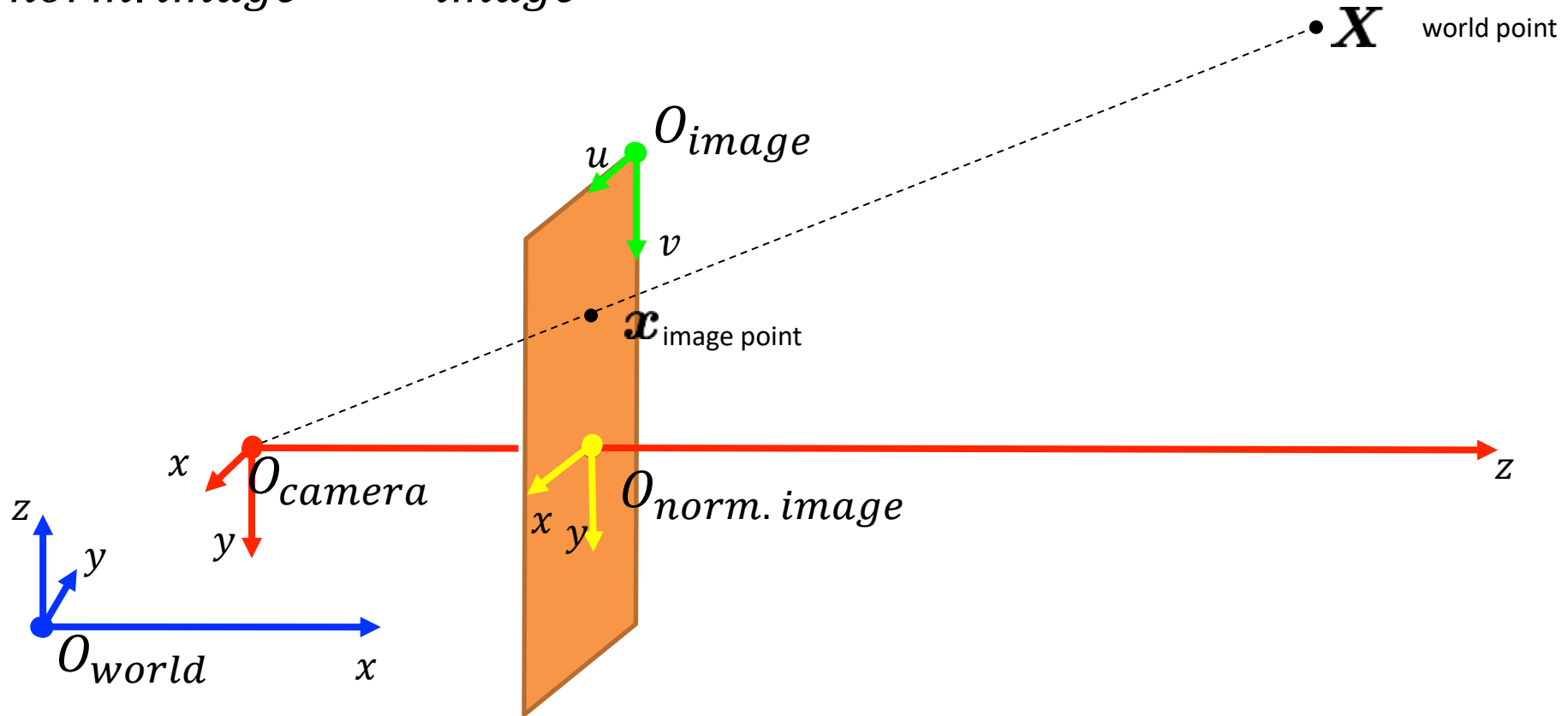
Intrinsic camera matrix

- The intrinsic matrix 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 **normalized image space** (also known as **projected camera space**) to the image space.
  - $O_{norm.\,image} \rightarrow O_{image}$

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

Intrinsic camera matrix

- Transforming to units of pixels in image space:
  - m_x is a ratio between the normalized image space to image space in pixels in x direction. (m_y is the same...)

$$f_x = m_x f \ \& \ f_y = m_y f$$

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

Intrinsic camera matrix

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

O_{image}

p

$O_{norm.\ image}$

image plane
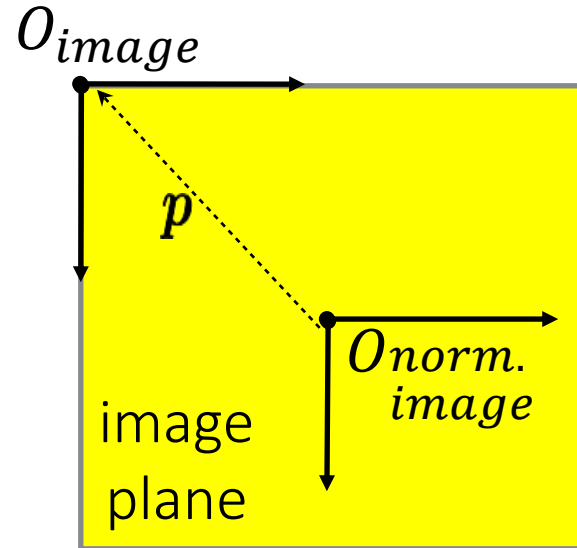
- How to add this to the intrinsic matrix?

Intrinsic camera matrix

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

O_{image}

image plane

$O_{norm.\ image}$

p
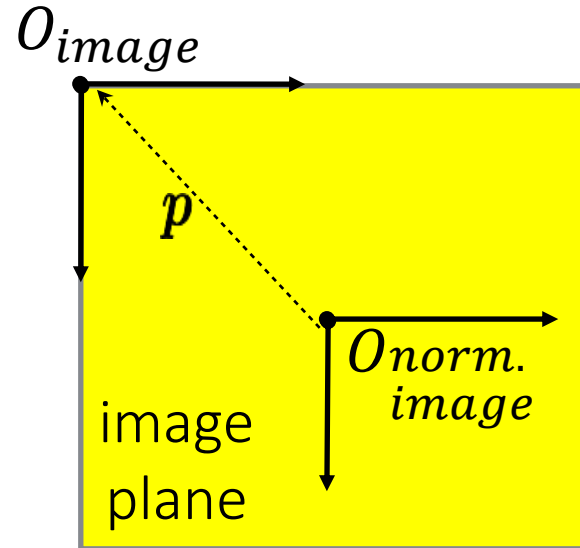
- How to add this to the intrinsic matrix?

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

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 extrinsics
- Perspective projection
- Camera intrinsics
- **Full camera matrix**
- Calibration methods and distortions

Full camera matrix

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

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

Contents

- What is camera calibration?
- Camera extrinsics
- Perspective projection
- Camera intrinsics
- 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
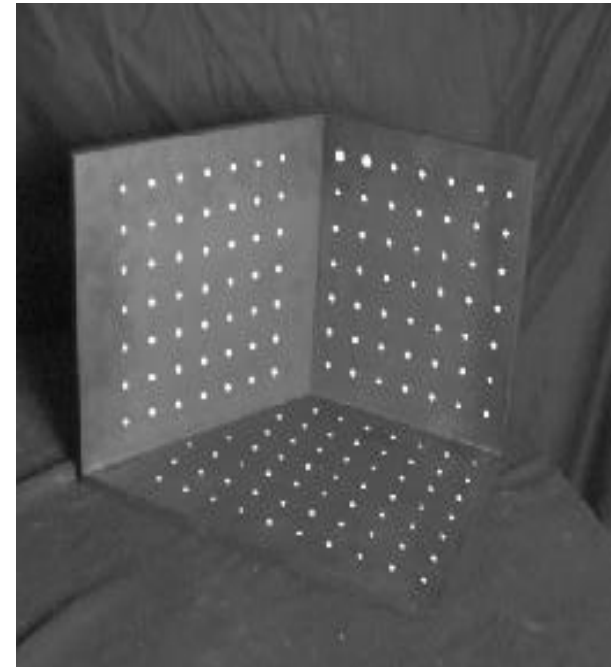
Full camera matrix

- 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.

$$Input : \{(u_i, v_i) \Leftrightarrow (x_i, y_i, z_i)\}_{i=1,...,N}$$
$$Output : P_{3x4}$$
$$s.t. :$$

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \frac{\tilde{u}}{\tilde{w}} \\ \frac{\tilde{v}}{\tilde{w}} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = P \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Full camera matrix

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\tilde{u}}{\tilde{w}} \\ \frac{\tilde{v}}{\tilde{w}} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{cases} \tilde{u} = p_{11}x + p_{12}y + p_{13}z + p_{14} \\ \tilde{v} = p_{21}x + p_{22}y + p_{23}z + p_{24} \\ \tilde{w} = p_{31}x + p_{32}y + p_{33}z + p_{34} \end{cases}$$

$$\begin{cases} u = \dfrac{\tilde{u}}{\tilde{w}} = \dfrac{p_{11}x + p_{12}y + p_{13}z + p_{14}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} \\[2em] v = \dfrac{\tilde{v}}{\tilde{w}} = \dfrac{p_{21}x + p_{22}y + p_{23}z + p_{24}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} \end{cases}$$

Full camera matrix

$$
\begin{cases}
u = \dfrac{\tilde{u}}{\tilde{w}} = \dfrac{p_{11}x + p_{12}y + p_{13}z + p_{14}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} \\[2em]
v = \dfrac{\tilde{v}}{\tilde{w}} = \dfrac{p_{21}x + p_{22}y + p_{23}z + p_{24}}{p_{31}x + p_{32}y + p_{33}z + p_{34}}
\end{cases}
$$

$$
\begin{cases}
(p_{31}x + p_{32}y + p_{33}z + p_{34})u = p_{11}x + p_{12}y + p_{13}z + p_{14} \\[0.5em]
(p_{31}x + p_{32}y + p_{33}z + p_{34})u = p_{21}x + p_{22}y + p_{23}z + p_{24}
\end{cases}
$$

$$
\begin{cases}
p_{31}xu + p_{32}yu + p_{33}zu + p_{34}u - p_{11}x - p_{12}y - p_{13}z - p_{14} = 0 \\[0.5em]
p_{31}xv + p_{32}yv + p_{33}zv + p_{34}v - p_{21}x - p_{22}y - p_{23}z - p_{24} = 0
\end{cases}
$$

$$
\begin{cases}
-xp_{11} - yp_{12} - zp_{13} - 1p_{14} - 0p_{21} - 0p_{22} - 0p_{23} - 0p_{24} + xup_{31} + yup_{32} + zup_{33} + up_{34} = 0 \\[0.5em]
0p_{11} + 0p_{12} + 0p_{13} + 0p_{14} - xp_{21} - yp_{22} - zp_{23} - 1p_{24} + xvp_{31} + yvp_{32} + zvp_{33} + vp_{34} = 0
\end{cases}
$$

Full camera matrix

$$\begin{cases} -xp_{11} - yp_{12} - zp_{13} - 1p_{14} - 0p_{21} - 0p_{22} - 0p_{23} - 0p_{24} + xup_{31} + yup_{32} + zup_{33} + up_{34} = 0 \\ 0p_{11} + 0p_{12} + 0p_{13} + 0p_{14} - xp_{21} - yp_{22} - zp_{23} - 1p_{24} + xvp_{31} + yvp_{32} + zvp_{33} + vp_{34} = 0 \end{cases}$$

$$\begin{bmatrix} -x & -y & -z & -1 & 0 & 0 & 0 & 0 & xu & yu & zu & u \\ 0 & 0 & 0 & 0 & -x & -y & -z & -1 & xv & yv & zv & v \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Full camera matrix

$$\begin{bmatrix} -x_i & -y_i & -z_i & -1 & 0 & 0 & 0 & 0 & x_i u_i & y_i u_i & z_i u_i & u_i \\ 0 & 0 & 0 & 0 & -x_i & -y_i & -z_i & -1 & x_i v_i & y_i v_i & z_i v_i & v_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Can the above equation be solved easily? What is one possible solution that we don't want and how do we avoid it?

Full camera matrix

$$\begin{bmatrix} -x_i & -y_i & -z_i & -1 & 0 & 0 & 0 & 0 & x_iu_i & y_iu_i & z_iu_i & u_i \\ 0 & 0 & 0 & 0 & -x_i & -y_i & -z_i & -1 & x_iv_i & y_iv_i & z_iv_i & v_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Can the above equation be solved easily? What is one possible solution that we don't want and how do we avoid it?

A possible (bad) answer is $\bar{p} = \bar{0}$.

Because we use homogenous vectors that are scale invariant, we can add a constraint like $\|p\| = 1$ to avoid it.

How many DOFs do we have now?

Full camera matrix

$$\begin{bmatrix} -x_i & -y_i & -z_i & -1 & 0 & 0 & 0 & 0 & x_i u_i & y_i u_i & z_i u_i & u_i \\ 0 & 0 & 0 & 0 & -x_i & -y_i & -z_i & -1 & x_i v_i & y_i v_i & z_i v_i & v_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Can the above equation be solved easily? What is one possible solution that we don't want and how do we avoid it?

A possible (bad) answer is $\bar{p} = \bar{0}$.

Because we use homogenous vectors that are scale invariant, we can add a constraint like $\|p\| = 1$ to avoid it.

How many DOFs do we have now? 11

Full camera matrix

$$
\begin{bmatrix}
-x_i & -y_i & -z_i & -1 & 0 & 0 & 0 & 0 & x_i u_i & y_i u_i & z_i u_i & u_i \\
0 & 0 & 0 & 0 & -x_i & -y_i & -z_i & -1 & x_i v_i & y_i v_i & z_i v_i & v_i
\end{bmatrix}
\begin{bmatrix}
p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34}
\end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \end{bmatrix}
$$

We have 11 DOFs.

For one corresponding $(u_i, v_i) \Leftrightarrow (x_i, y_i, z_i)$ we get 2 equations. What is the minimum amount of correspondent points that we need?

Full camera matrix

$$
\begin{bmatrix}
-x_i & -y_i & -z_i & -1 & 0 & 0 & 0 & 0 & x_i u_i & y_i u_i & z_i u_i & u_i \\
0 & 0 & 0 & 0 & -x_i & -y_i & -z_i & -1 & x_i v_i & y_i v_i & z_i v_i & v_i
\end{bmatrix}
\begin{bmatrix}
p_{11} \\
p_{12} \\
p_{13} \\
p_{14} \\
p_{21} \\
p_{22} \\
p_{23} \\
p_{24} \\
p_{31} \\
p_{32} \\
p_{33} \\
p_{34}
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0
\end{bmatrix}
$$

We have 11 DOFs.

For one corresponding $(u_i, v_i) \Leftrightarrow (x_i, y_i, z_i)$ we get 2 equations. What is the minimum amount of correspondent points that we need? 6

Full camera matrix

$$\begin{bmatrix} -x_1 & -y_1 & -z_1 & -1 & 0 & 0 & 0 & 0 & x_1 u_1 & y_1 u_1 & z_1 u_1 & u_1 \\ 0 & 0 & 0 & 0 & -x_1 & -y_1 & -z_1 & -1 & x_1 v_1 & y_1 v_1 & z_1 v_1 & v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_N & -y_N & -z_N & -1 & 0 & 0 & 0 & 0 & x_N u_N & y_N u_N & z_N u_N & u_N \\ 0 & 0 & 0 & 0 & -x_N & -y_N & -z_N & -1 & x_N v_N & y_N v_N & z_N v_N & v_N \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Let's use new notations and say $X\beta = 0$ with the constraint $\|\beta\| = 1$.

For more than 6 correspondent points we can get an error in the results, so
we want to minimize this error:

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

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:

$$Av = \lambda v \mapsto v^T Av = \lambda$$

  – Also recall that each eigenvector v is normalized $(\lVert v \rVert = v^T v = 1)$.

- 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 extrinsics
- Perspective projection
- Camera intrinsics
- Full camera matrix
- **Calibration methods and distortions**

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

No distortion   Pin cushion   Barrel
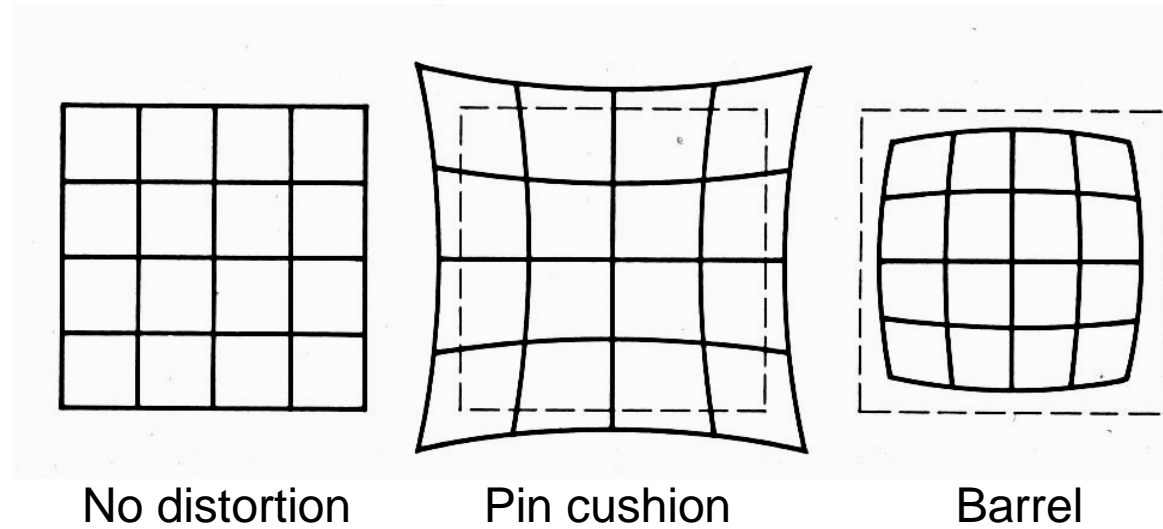
- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

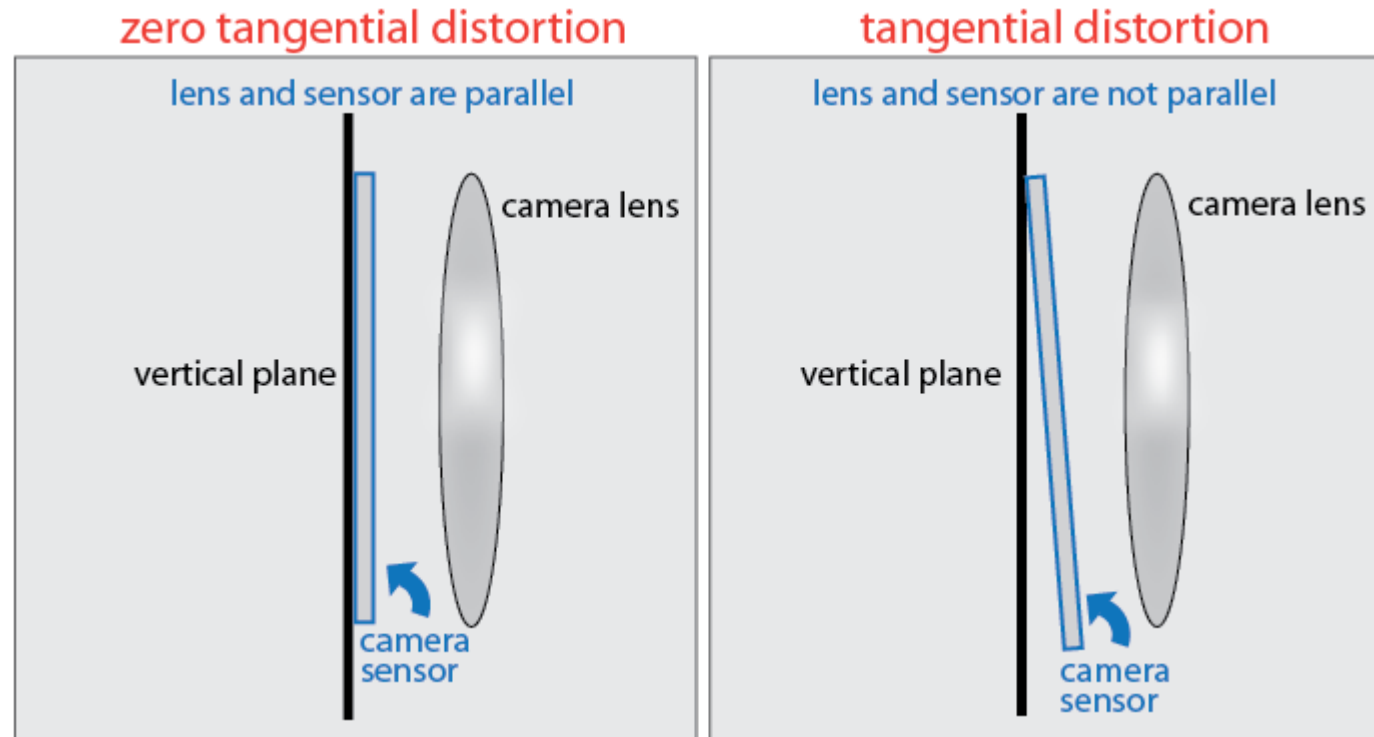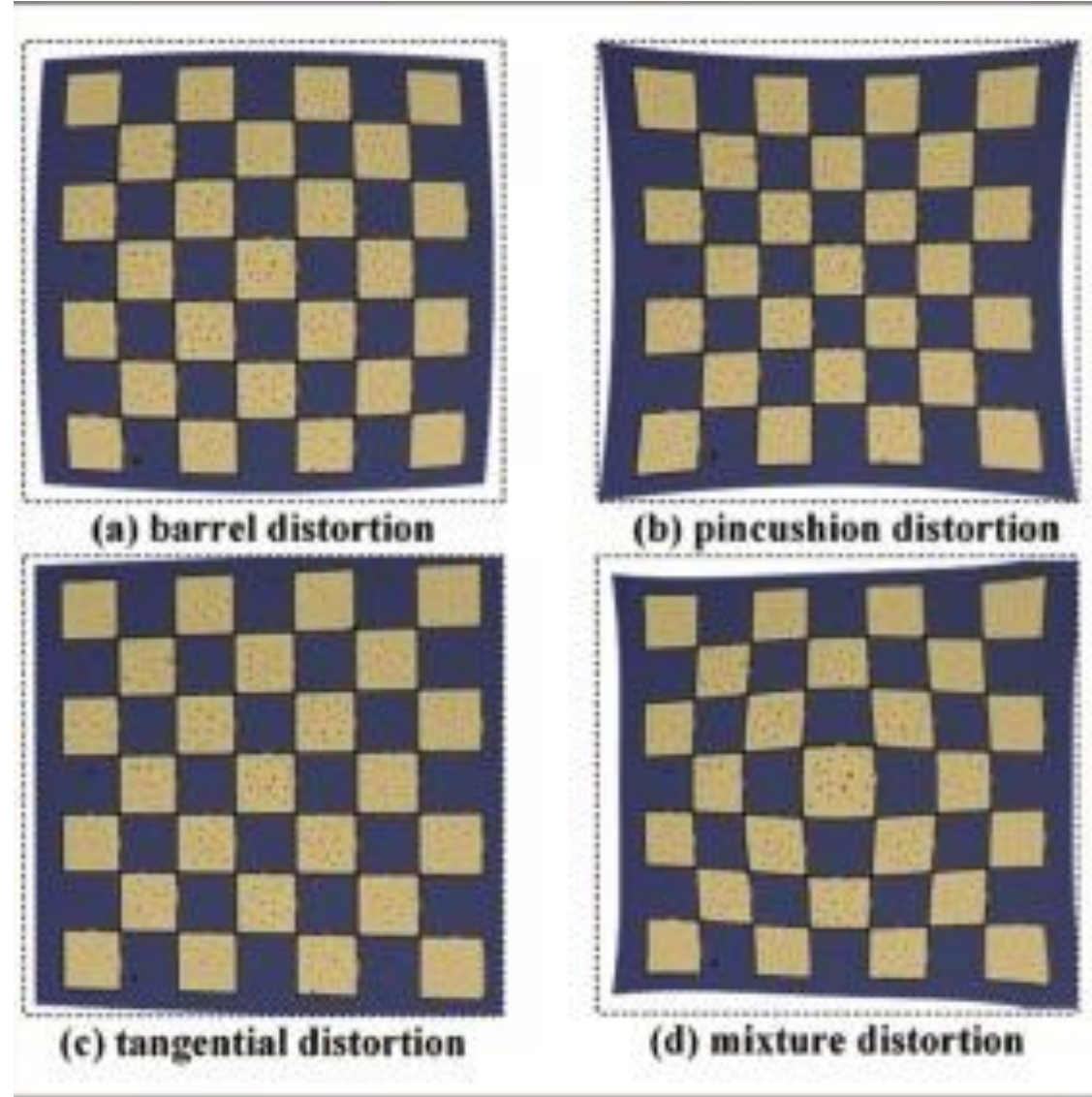# Radial distortion

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

(a) barrel distortion

(b) pincushion distortion

(c) tangential distortion

(d) mixture 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