

Robotix-Academy Roadshow 2021

2D Camera Calibration

Stefan Marx

ZeMA gGmbH
Zentrum für Mechatronik und Automatisierungstechnik
Saarbrücken, 29.04.2021

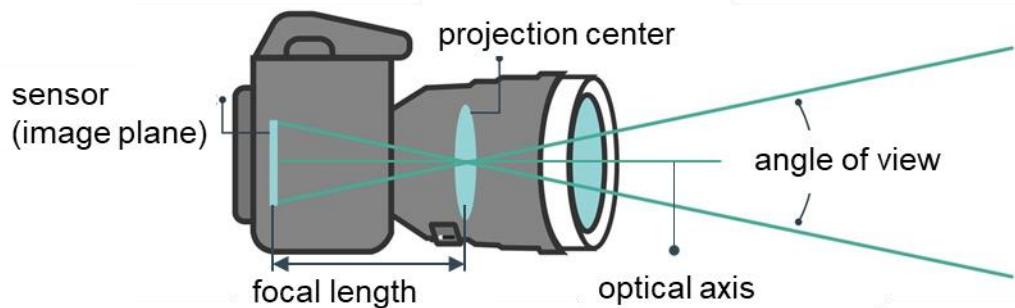
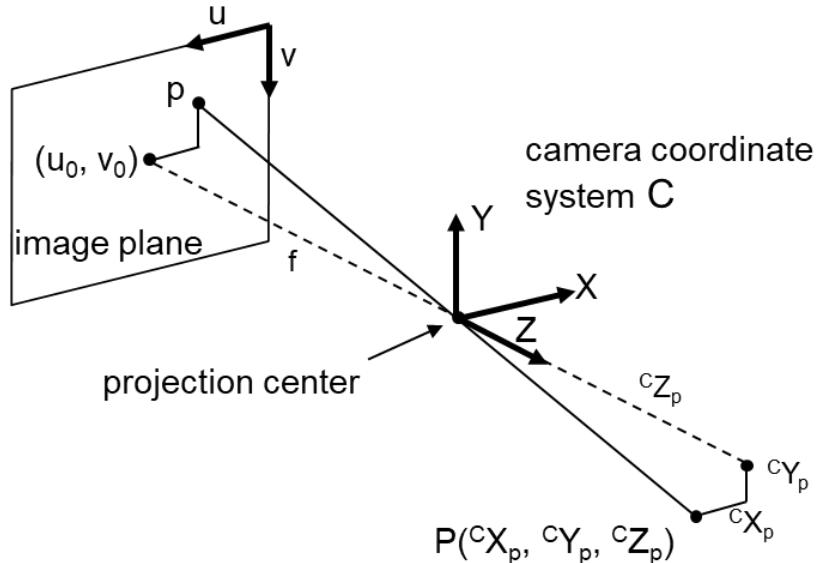
Agenda

- 1 Pinhole Camera & Intrinsic Parameters
- 2 Extrinsic Parameters
- 3 3D Position Calculation
- 4 Distortion
- 5 Robot-Camera-Calibration
- 6 Camera-Calibration with OpenCV
- 7 Demonstration

Agenda

- 1 Pinhole Camera & Intrinsic Parameters**
- 2 Extrinsic Parameters**
- 3 3D Position Calculation**
- 4 Distortion**
- 5 Robot-Camera-Calibration**
- 6 Camera-Calibration with OpenCV**
- 7 Demonstration**

Camera model: pinhole-camera

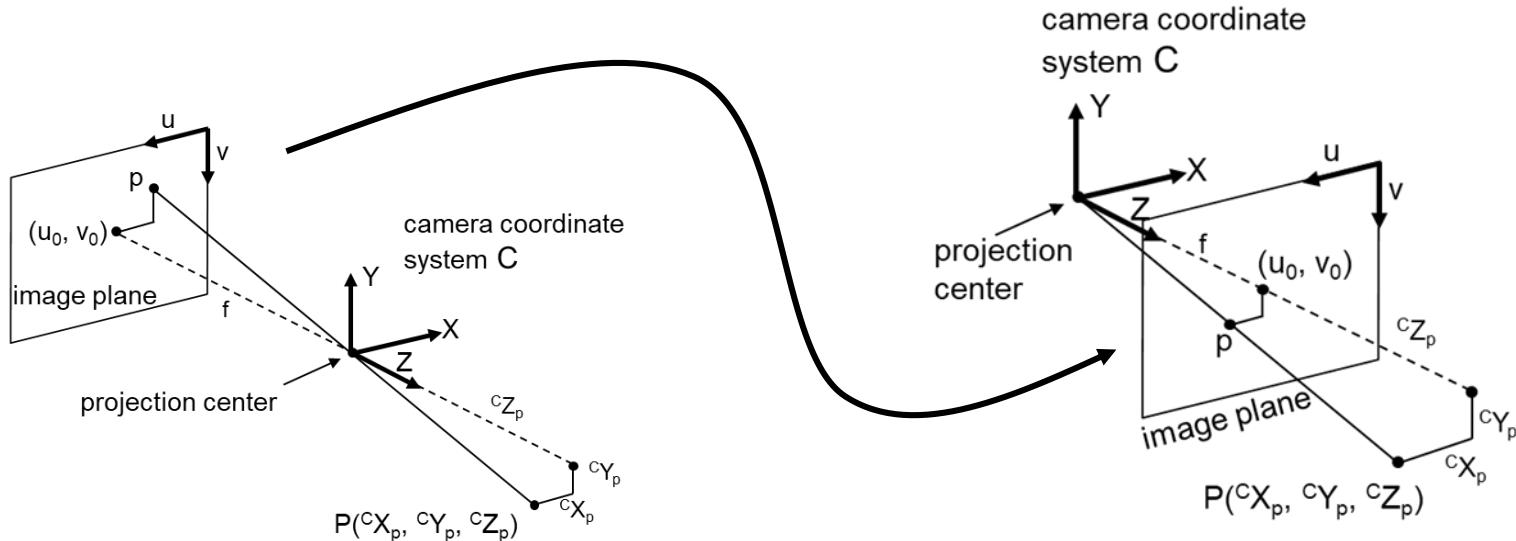


- middle:
 - camera coordinate system C (C-CS) with the origin in the projection center
 - cZ -axis = optical axis
- left:
 - image plan with axes u & v (unit = pixel)
 - in distance f to the C-CS
 - distance f = focal length
- right:
 - Some point in the free space (world coordinates)

Quelle: <https://www.researchgate.net>, <https://captureheatlas.com/what-is-focal-length/>

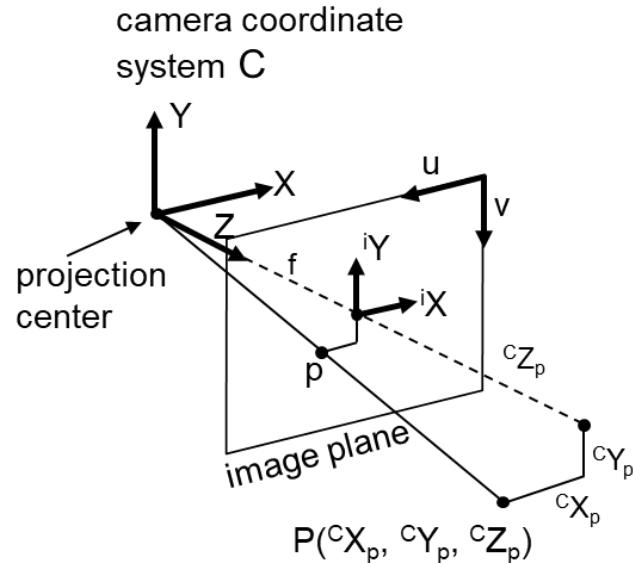
Camera model: pinhole-camera

- To simplify the model and subsequent calculations, the image plane is shifted:
 - at distance f from projection center between projection center and world coordinates
 - Mirroring of world coordinates on ${}^C X$ -axis und ${}^C Y$ -axis is prevented



Quelle: <https://www.researchgate.net>

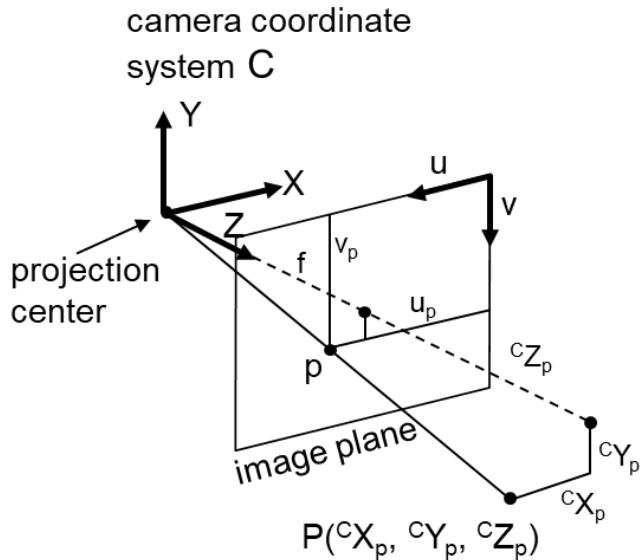
Intrinsic parameters



- for the point p in the image plane:
 - $iX_p = f * \frac{cX_p}{cZ_p}$
 - $iY_p = f * \frac{cY_p}{cZ_p}$
 - $\begin{pmatrix} iX_p \\ iY_p \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} cX_p/cZ_p \\ cY_p/cZ_p \\ 1 \end{pmatrix}$
 - iX_p und iY_p as well as cX_p , cY_p und cZ_p are in the chosen length unit (e.g. mm)
 - the conversion from mm to pixels is done in a further step

1) mm oder eine beliebige andere Längeneinheit

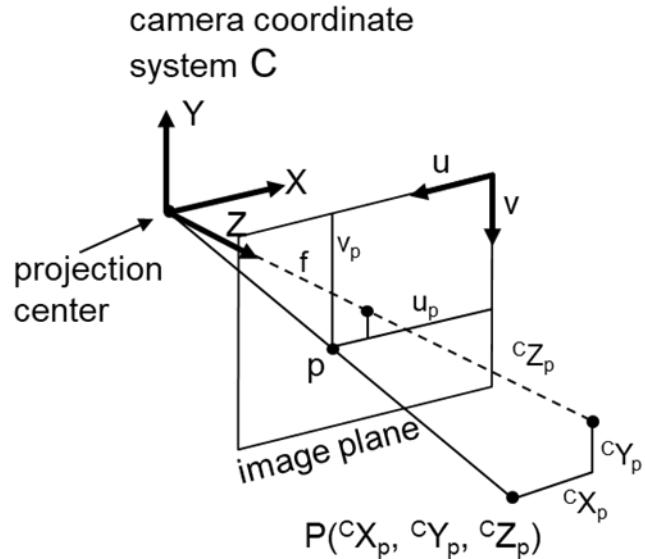
Intrinsic parameters



- conversion from mm to pixels :
 - factors s_x und s_y define the dimensions of a pixel in the image plane (unit: pixel/mm)
 - with u_0 and v_0 the coordinate system of the image plane is moved to the intersection of the image plane with the optical axis¹.
- for the point p in the image plane:
 - $u_p = f * s_x * \frac{cX_p}{cZ_p} + u_0 = f_x * \frac{cX_p}{cZ_p} + u_0$
 - $v_p = f * s_y * \frac{cY_p}{cZ_p} + v_0 = f_y * \frac{cY_p}{cZ_p} + v_0$
 - $$\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} cX_p/cZ_p \\ cY_p/cZ_p \\ 1 \end{pmatrix}$$
 - $$\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} cX_p/cZ_p \\ cY_p/cZ_p \\ 1 \end{pmatrix}$$

1) Center of the image plane ≠ Point of intersection of the image plane with the optical axis

Intrinsic parameters



- $\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^C X_p / {}^C Z_p \\ {}^C Y_p / {}^C Z_p \\ 1 \end{pmatrix}$
- $\begin{pmatrix} {}^C X_p / {}^C Z_p \\ {}^C Y_p / {}^C Z_p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix}$
- the equation only provides the ratios ${}^C X_p / {}^C Z_p$ and ${}^C Y_p / {}^C Z_p$
- to get point $P({}^C X_p, {}^C Y_p, {}^C Z_p)$ further calculations are necessary
- limitations:
 - the point P has to be in a predefined plane to get accurate results

Agenda

1 Pinhole Camera & Intrinsic Parameters

2 Extrinsic Parameters

3 3D Position Calculation

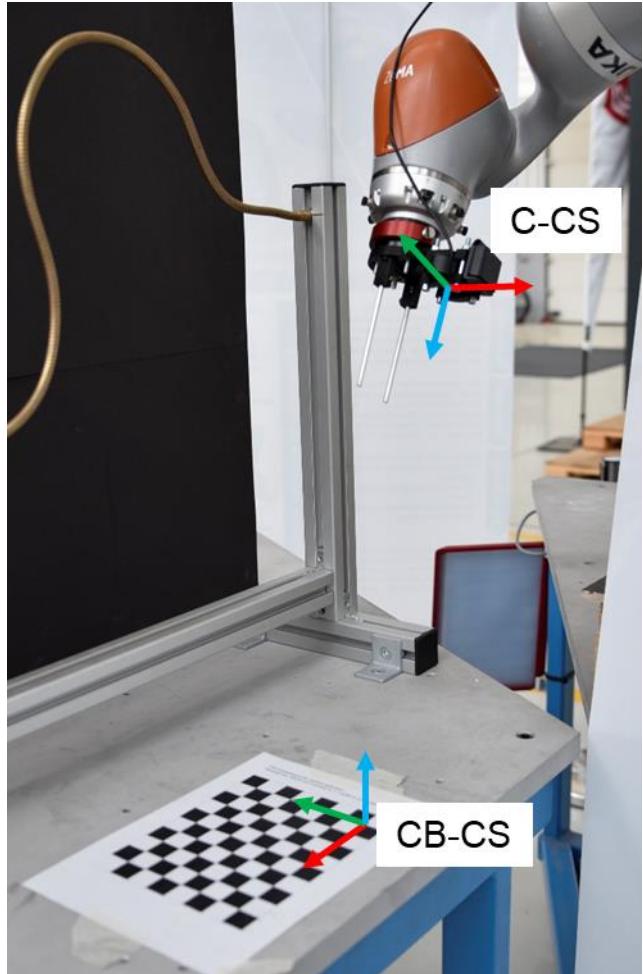
4 Distortion

5 Robot-Camera-Calibration

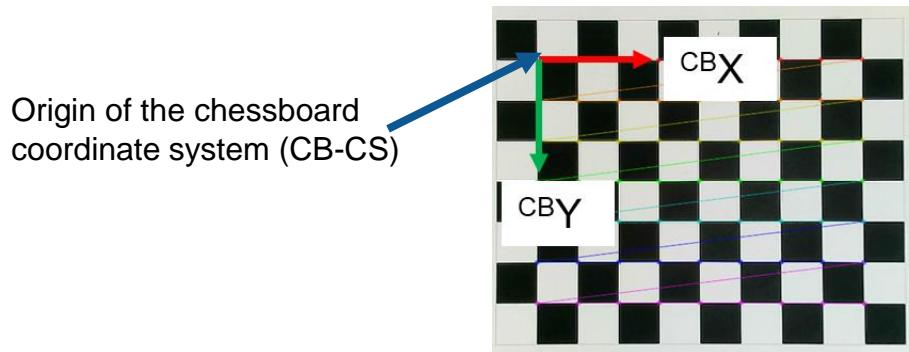
6 Camera-Calibration with OpenCV

7 Demonstration

Camera calibration with chessboard pattern



- calibration with the chessboard pattern
 - input:
 - number of rows and columns of the pattern -1
 - edge length of a square in the pattern
 - output:
 - camera matrix: 3x3, intrinsic parameters
 - translational vector from C-CS to CB-CS
 - rotational vector of the CB-CS regarding to the C-CS (Rodriguez vector)



Extrinsic parameters

- translational vector $\underline{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$
- rotational vector (Rodriguez vector) $\underline{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$
- \rightarrow transformation matrix ${}^C\mathbf{T}_{CB}$ from C-CS to CB-CS:
$${}^C\mathbf{T}_{CB} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & t_x \\ d_{21} & d_{22} & d_{23} & t_y \\ d_{31} & d_{32} & d_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
- $$\begin{pmatrix} {}^C X_p \\ {}^C Y_p \\ {}^C Z_p \\ 1 \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & t_x \\ d_{21} & d_{22} & d_{23} & t_y \\ d_{31} & d_{32} & d_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^{CB} X_p \\ {}^{CB} Y_p \\ 0 \\ 1 \end{pmatrix}$$
 $\rightarrow {}^{CB} Z_p = 0$ in the plane of the chessboard pattern

Agenda

- 1 Pinhole Camera & Intrinsic Parameters
- 2 Extrinsic Parameters
- 3 **3D Position Calculation**
- 4 Distortion
- 5 Robot-Camera-Calibration
- 6 Camera-Calibration with OpenCV
- 7 Demonstration

Calculation of point P(${}^C X_p$, ${}^C Y_p$, ${}^C Z_p$)

- Two equations:

- Intrinsic parameters:

- $$\begin{pmatrix} {}^C X_p / {}^C Z_p \\ {}^C Y_p / {}^C Z_p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} \rightarrow \begin{array}{l} {}^C X_p = a * {}^C Z_p \\ {}^C Y_p = b * {}^C Z_p \\ {}^C Z_p = {}^C Z_p \end{array}$$

- Extrinsic parameters:

- $$\begin{pmatrix} {}^C X_p \\ {}^C Y_p \\ {}^C Z_p \\ 1 \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & t_x \\ d_{21} & d_{22} & d_{23} & t_y \\ d_{31} & d_{32} & d_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^{SB} X_p \\ {}^{SB} Y_p \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{array}{l} {}^C X_p = d_{11} * {}^{SB} X_p + d_{12} * {}^{SB} Y_p + t_x \\ {}^C Y_p = d_{21} * {}^{SB} X_p + d_{22} * {}^{SB} Y_p + t_y \\ {}^C Z_p = d_{31} * {}^{SB} X_p + d_{32} * {}^{SB} Y_p + t_z \end{array}$$

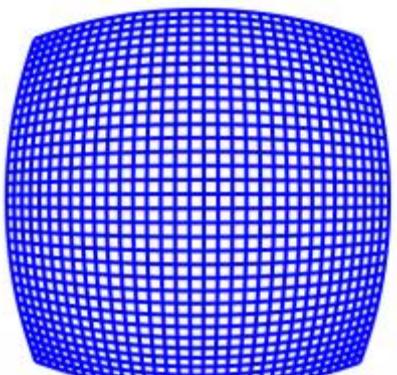
Berechnung von $P(CX_p, CY_p, CZ_p)$

- $a * CZ_p = d_{11} * SBX_p + d_{12} * SBY_p + tx$
- $b * CZ_p = d_{21} * SBX_p + d_{22} * SBY_p + ty \quad \rightarrow \quad t_x = -d_{11} * SBX_p - d_{12} * SBY_p + a * CZ_p$
- $CZ_p = d_{31} * SBX_p + d_{32} * SBY_p + tz \quad t_y = -d_{21} * SBX_p - d_{22} * SBY_p + b * CZ_p$
- $$\begin{pmatrix} tx \\ ty \\ tz \end{pmatrix} = \begin{pmatrix} -d_{11} & -d_{12} & a \\ -d_{21} & -d_{22} & b \\ -d_{31} & -d_{32} & 1 \end{pmatrix} \cdot \begin{pmatrix} SBX_p \\ SBY_p \\ CZ_p \end{pmatrix} \quad \rightarrow \quad t_z = -d_{31} * SBX_p - d_{32} * SBY_p + 1 * CZ_p$$
- $$\begin{pmatrix} SBX_p \\ SBY_p \\ CZ_p \end{pmatrix} = \begin{pmatrix} -d_{11} & -d_{12} & a \\ -d_{21} & -d_{22} & b \\ -d_{31} & -d_{32} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} tx \\ ty \\ tz \end{pmatrix}$$
- $$\begin{aligned} CX_p &= a * CZ_p \\ CY_p &= b * CZ_p \end{aligned}$$
- Conclusion : to each pixel of an image we can assign a position on a plane in space

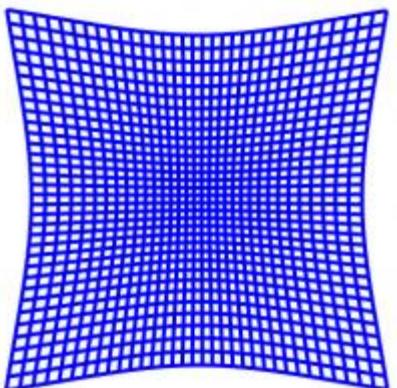
Agenda

- 1 Pinhole Camera & Intrinsic Parameters
- 2 Extrinsic Parameters
- 3 3D Position Calculation
- 4 Distortion
- 5 Robot-Camera-Calibration
- 6 Camera-Calibration with OpenCV
- 7 Demonstration

Distortion

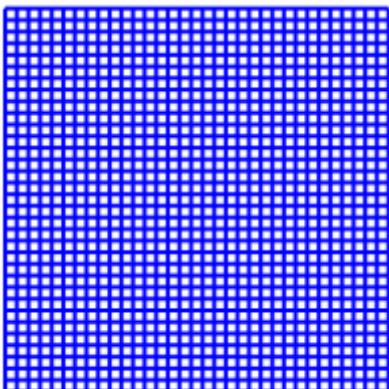


BARREL DISTORTION



PINCUSHION DISTORTION

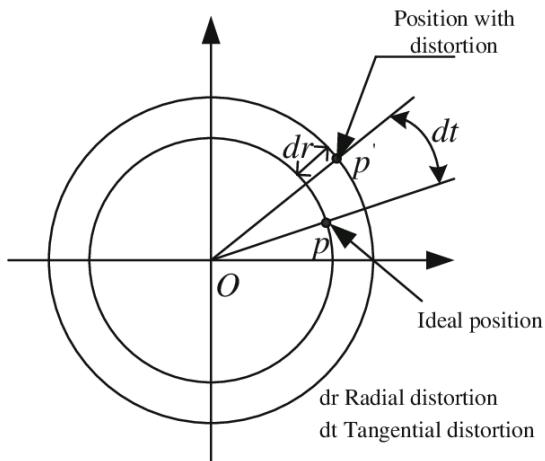
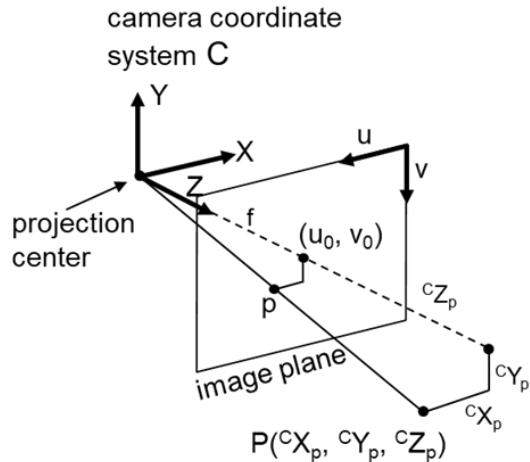
- The preceding calculations refer to an ideal camera model
- Real cameras include manufacturing tolerances and inaccuracies
 - Lens distortion (optical distortion)
- Distortion types:
 - Barrel distortion
 - Pincushion distortion



INPUT GRID

References: learnopencv.com

Distortion



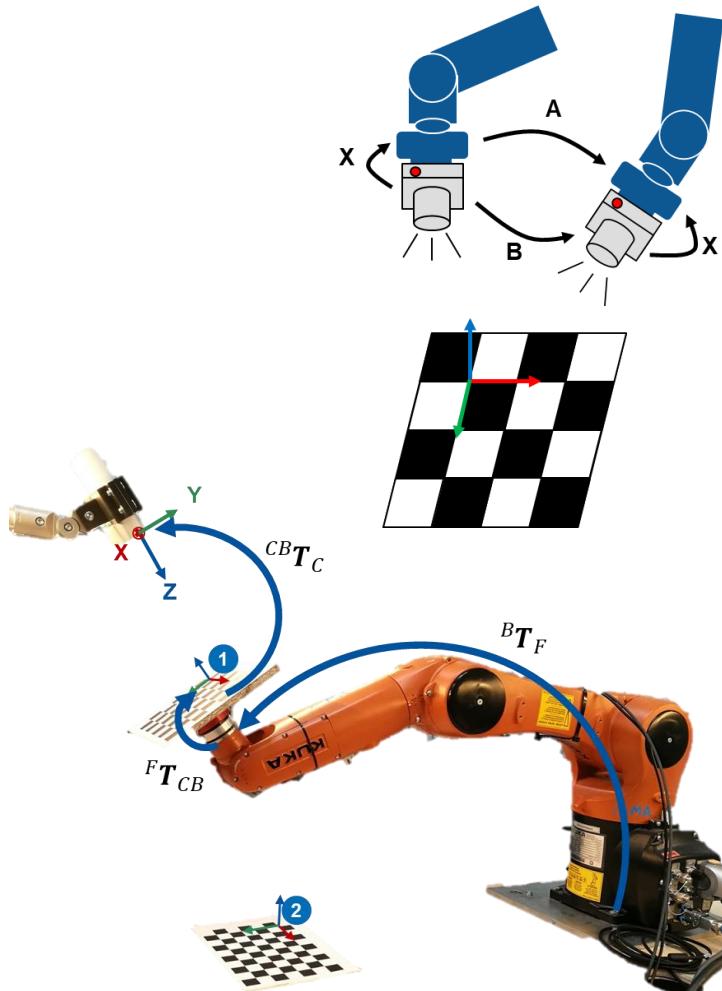
- Distortion is divided in
 - Tangential distortion
 - Radial distortion
- Distortion depends on radius of the pixel with respect to the distortion center
 - Distortion center: u_0 and v_0
 - Radius: $r = \sqrt{(u_p - u_0)^2 + (v_p - v_0)^2}$
- Tangential distortion:
 - $x_{undist} = x_d * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$
- Radial distortion:
 - $x_{undist} = x_d + 2p_1 x_d y_d + p_2(r^2 + x_d^2)$
 - $y_{undist} = y_d + p_1(r^2 + y_d^2) + 2p_2 x_d y_d$
- Distortion coefficients: k_1, k_2, k_3, p_1, p_2

References: [Laboratory calibration of star sensor with installation error using a nonlinear distortion model](#)

Agenda

- 1 Pinhole Camera & Intrinsic Parameters
- 2 Extrinsic Parameters
- 3 3D Position Calculation
- 4 Distortion
- 5 Robot-Camera-Calibration
- 6 Camera-Calibration with OpenCV
- 7 Demonstration

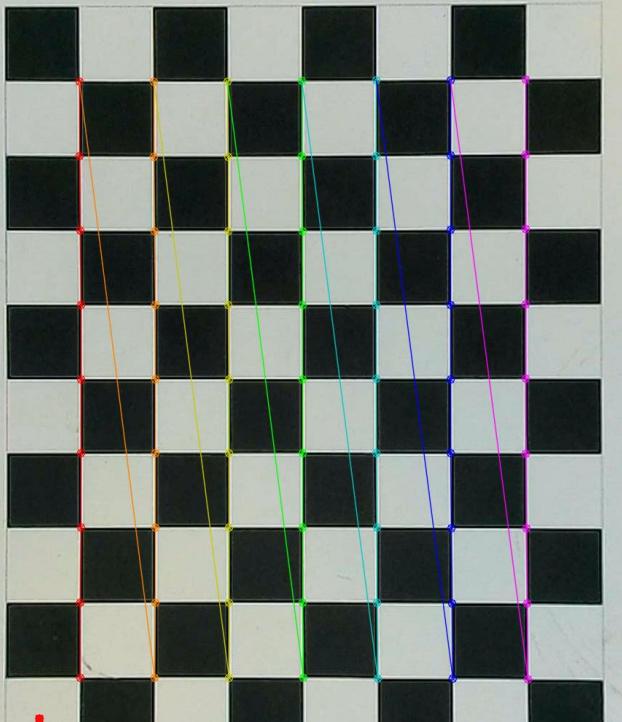
Robot-Camera-Calibration



- Calibrate chessboard to camera from four different positions and orientations:
 - store transformation matrixes ${}^{C,i}T_{CB}$ ($i = 1..4$)
 - store flange positions ${}^B T_{F,i}$ ($i = 1..4$)
- Calibration is done by solving the equation
 - $AX = XB$
 - X : transformation matrix camera to flange ${}^F T_C$
 - A : transformation matrix flange to flange ${}^{F,i+1}T_{F,i}$
 - B : transformation matrix camera to camera ${}^{C,i+1}T_{C,i}$
- for more information see paper:
 - Position and orientation calibration of a 2D line sensor using closed-form least-squares solution

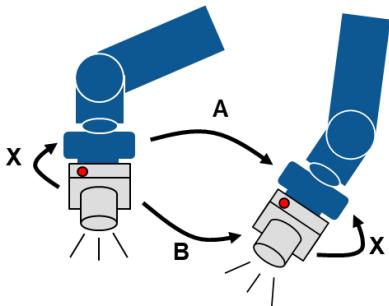
Robot-Camera-Calibration

7x9 checkerboard for camera calibration.
Squares are: 20x20 mm if printed to 1:1 scale on a A4 paper.



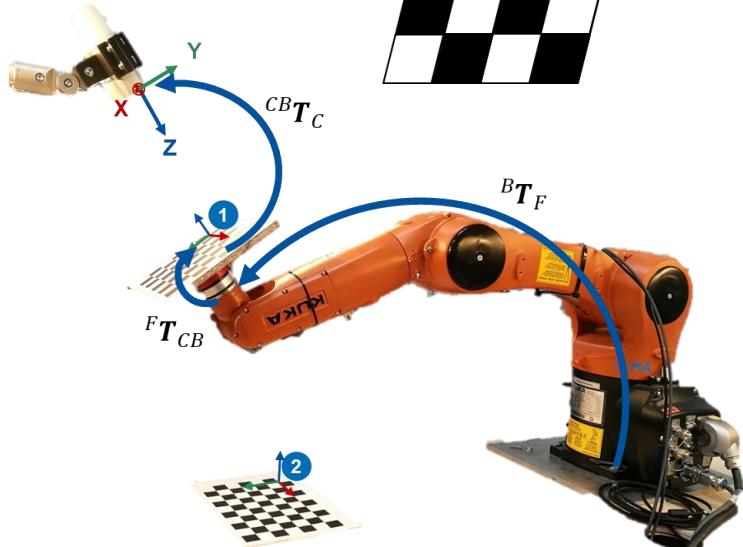
- Origin of the chessboard can switch between two positions
- Affects calculation of Matrix B
- Calculation of matrix B :
 - Camera on flange: ${}^{C,i+1}\mathbf{T}_{C,i} = {}^{C,i+1}\mathbf{T}_{CB} \cdot ({}^{C,i}\mathbf{T}_{CB})^{-1}$
 - Static camera: ${}^{CB,i+1}\mathbf{T}_{CB,i} = ({}^{C,i+1}\mathbf{T}_{CB})^{-1} \cdot {}^{C,i}\mathbf{T}_{CB}$
- Calculation of matrix A :
 - ${}^{F,i+1}\mathbf{T}_{F,i} = ({}^{B,i+1}\mathbf{T}_F)^{-1} \cdot {}^{B,i}\mathbf{T}_F$

Robot-Camera-Calibration



- Camera on flange:

- ${}^B T_C = {}^B T_F \cdot {}^F T_C$
- Has to calculated for every robot pose



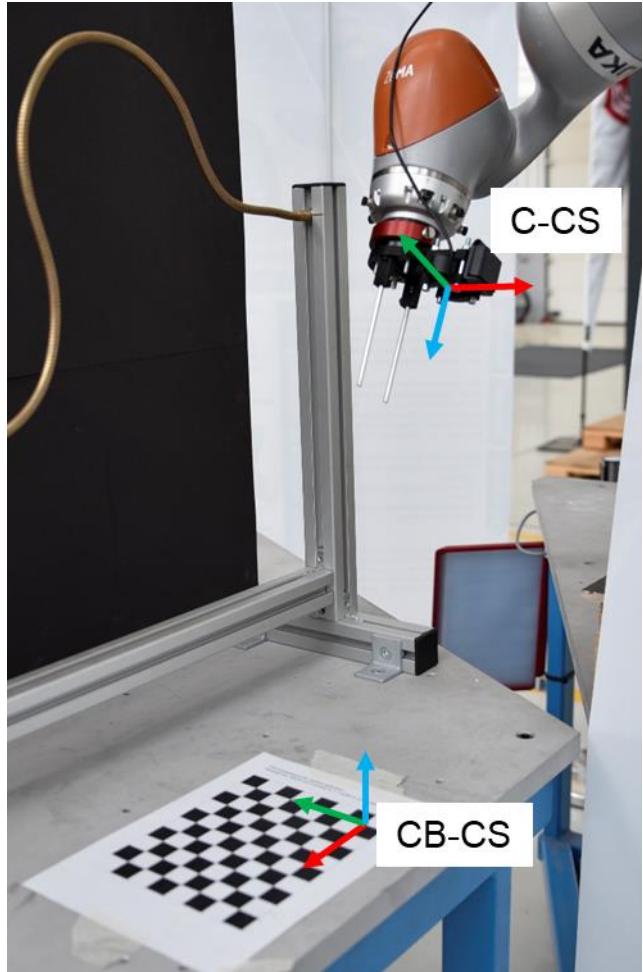
- Static Camera:

- ${}^B T_C = {}^B T_F \cdot {}^F T_{CB} \cdot {}^{CB} T_C$
- Calculated once

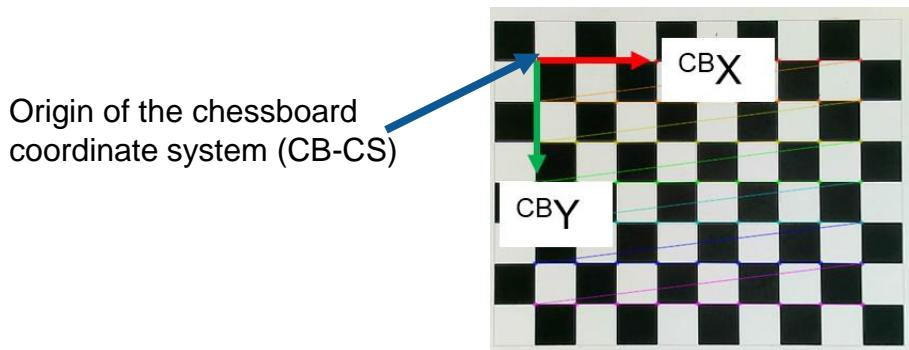
Agenda

- 1 Pinhole Camera & Intrinsic Parameters
- 2 Extrinsic Parameters
- 3 3D Position Calculation
- 4 Distortion
- 5 Robot-Camera-Calibration
- 6 Camera-Calibration with OpenCV
- 7 Demonstration

Camera calibration with OpenCV



- calibration with the chessboard pattern
 - input:
 - number of rows and columns of the pattern -1
 - edge length of a square in the pattern
 - 15 images (or more) of the chessboard pattern from different positions and orientations
 - output:
 - camera matrix: 3x3, intrinsic parameters
 - for every image: translational vector from C-CS to CB-CS
 - for every image: rotational vector of the CB-CS regarding to the C-CS (Rodriguez vector)



Camera calibration with OpenCV

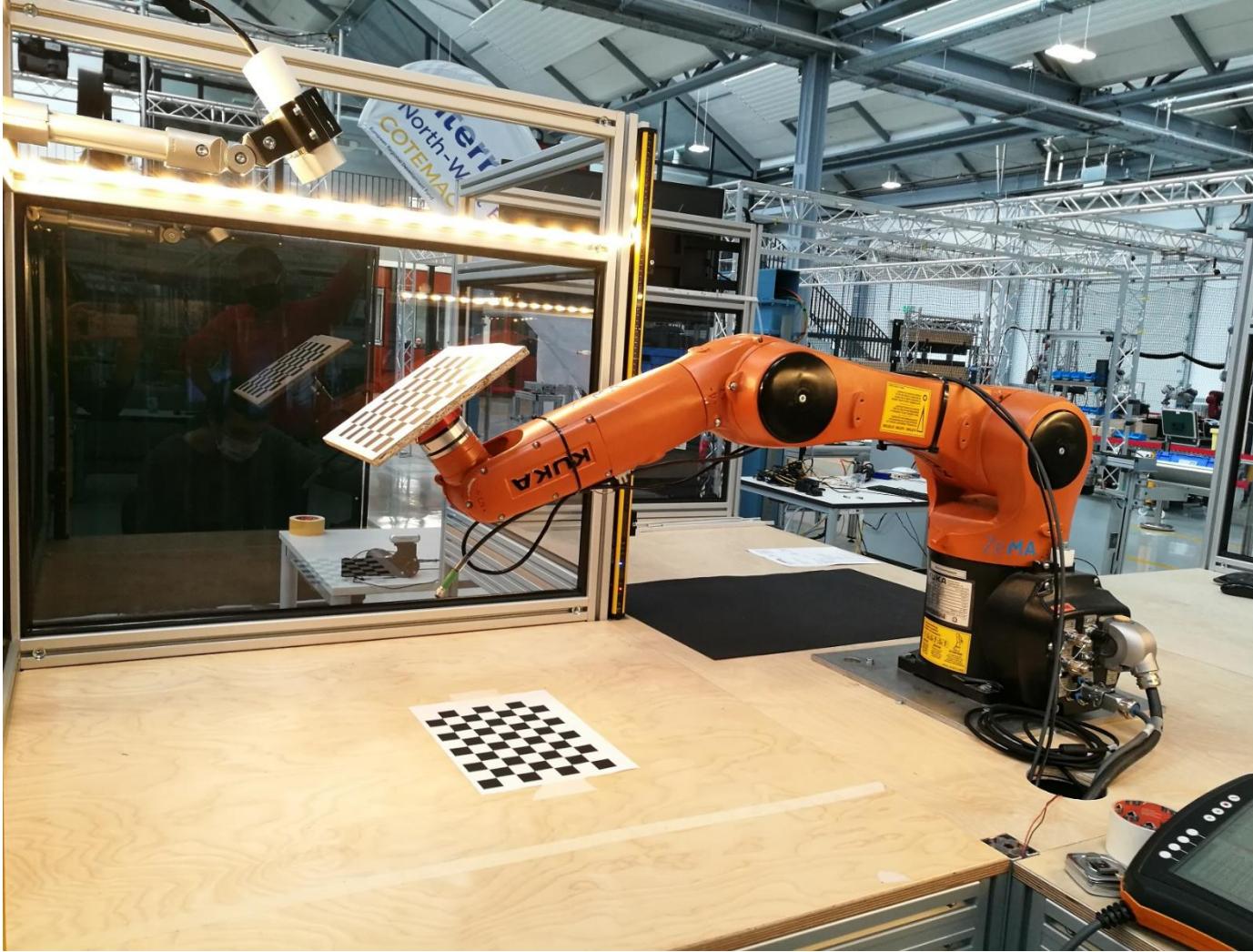
■ Important functions:

- cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) for conversion to grayscale
- cv2.findChessboardCorners(grayscale...)
- cv2.cornerSubPix() to find the accurate corner locations
- cv2.calibrateCamera() for actual camera calibration
 - Input:
 - Object points: chessboard corners with respect to chessboard-CS (e.g. (20, 0, 0); (40, 0, 0) ...)
 - Image points: chessboard corners in pixel coordinates
 - cv2.projectPoints() to project object points back to the image plane and to then calculate an calibration error
 - cv2.undistortPoints() to remove distortion for the selected pixel

Agenda

- 1 Pinhole Camera & Intrinsic Parameters
- 2 Extrinsic Parameters
- 3 3D Position Calculation
- 4 Distortion
- 5 Robot-Camera-Calibration
- 6 Camera-Calibration with OpenCV
- 7 Demonstration

Demonstration



Calibration errors



■ Intel RealSense D435:

- Resolution: 1280x720
- Basic camera Calibration: error \approx 0,3 pixel
- Robot-camera-calibration: error \approx 5 mm



■ Microsoft Azure Kinect

- Resolution: 3840x2160
- Basic camera Calibration: error \approx 0,15 pixel
- Robot-camera-calibration: error \approx 1,5 mm

■ Tool calibration:

- Error \approx 0.25mm

References: Intel, Microsoft

Thank you for your attention

Contact

Stefan Marx

s.marx@zema.de

+49 (0)681 - 85 787 - 553