

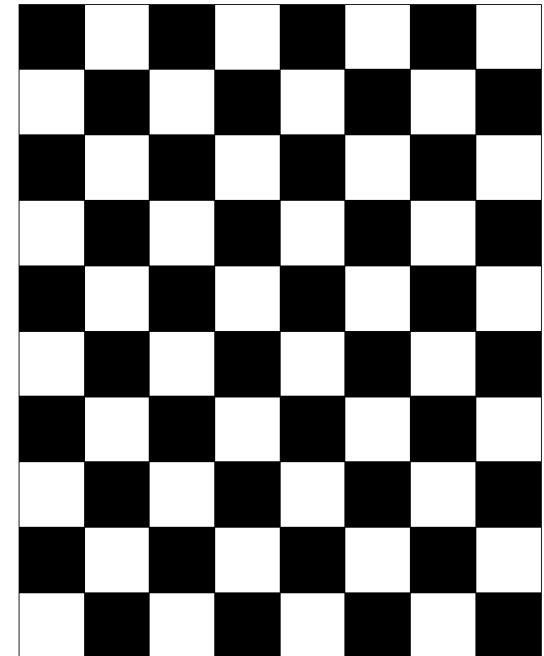
# Projektarbeit Echtzeitsysteme Kamerakalibrierung

Abschlusspräsentation der Projektarbeit | Jari Blumrich und Florian Krauß

# Kamerakalibrierung – Vorbereitung

- Kameraparameter rausfinden → Datenblatt
  - Größe der Pixel
  - Kameraparameter
- Kalibrierboard drucken
  - Druckgröße beachten!
- Passende Auflösung auswählen

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

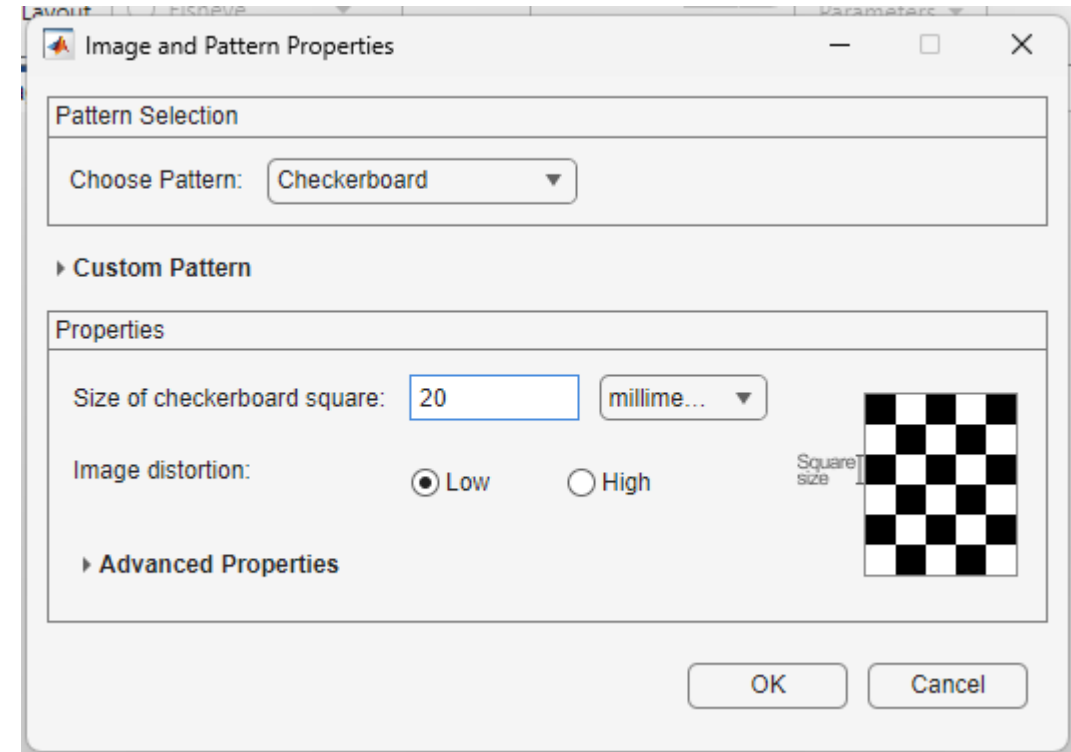


# Kamerakalibrierung – Durchführung

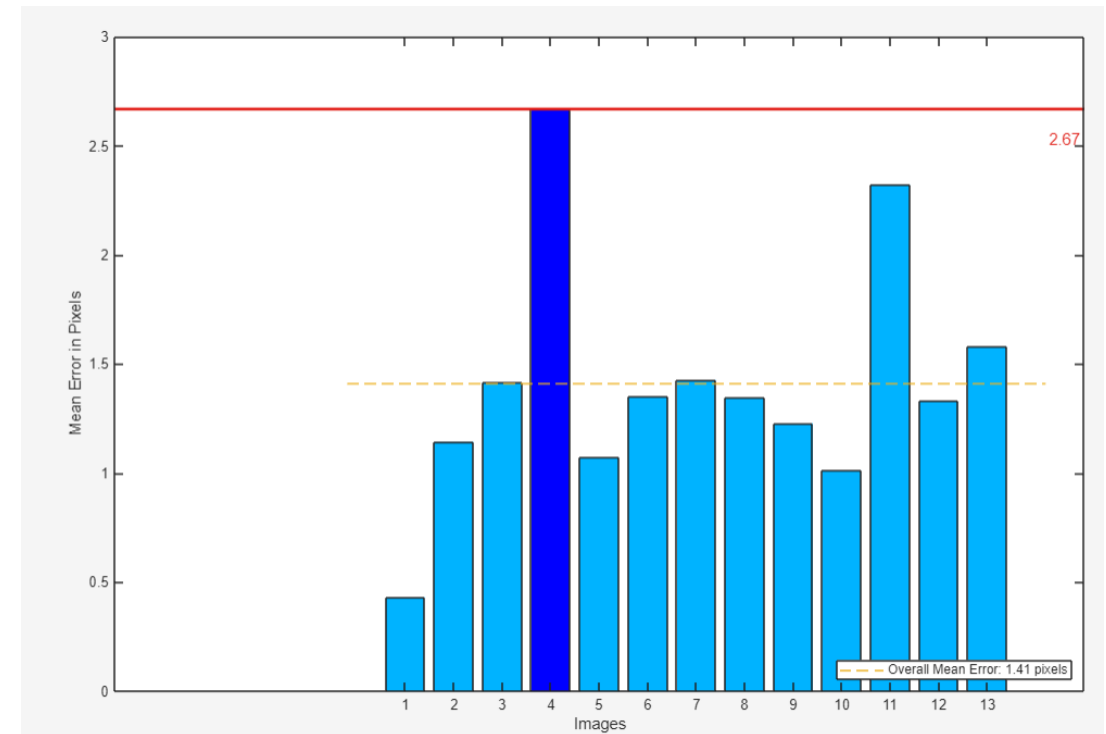
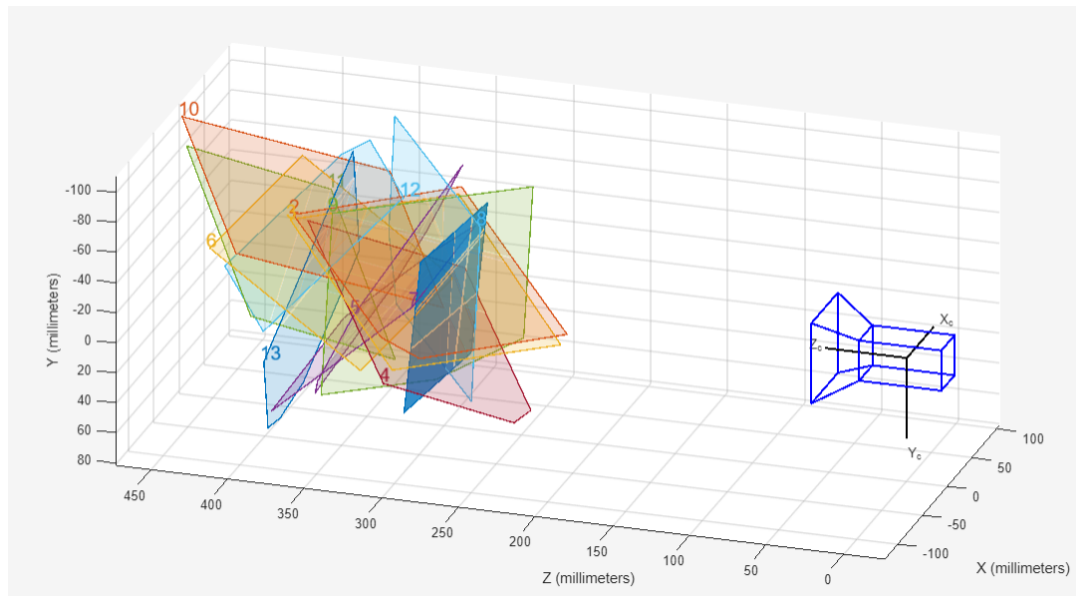
- Kamera auf die Richtige Auflösung einstellen
- Bilder von dem Kalibrierboard machen
  - Ca. 15-20 Bilder
  - Verschiedene Blickwinkel
  - Verschiedene Positionen → Auch Randbereich vom Bild abdecken

# Kamerakalibrierung – Auswertung

- Kalibrierungstool auswählen
  - Matlab – Camera Calibration Toolbox
  - OpenCV (Python / C++)
- Bilder in das Tool laden und auswerten



# Kamerakalibrierung - Ergebnisse



# Kamerakalibrierung - Ergebnisse

| cameraParams_VGA                |   |
|---------------------------------|---|
| 1x1 cameraParameters            |   |
| Property                        | Value   |
| ImageSize                       | [480,640]                                       |
| RadialDistortion                | [-0.0154,0.1551]                                |
| TangentialDistortion            | [0,0]   |
| WorldPoints                     | 63x2 double                                     |
| WorldUnits                      | 'millimeters'                                   |
| EstimateSkew                    | 0   |
| NumRadialDistortionCoefficients | 2   |
| EstimateTangentialDistortion    | 0   |
| ReprojectionErrors              | 63x2x31 double                                  |
| DetectedKeypoints               | 63x31 logical                                   |
| RotationVectors                 | 31x3 double                                     |
| K                               | [617.4263,0,324.6594;0,617.1984,245.4463;0,0,1] |
| NumPatterns                     | 31  |
| Intrinsics                      | 1x1 cameraIntrinsics                            |
| PatternExtrinsics               | 31x1 rigidtfom3d                                |
| FocalLength                     | [617.4263,617.1984]                             |
| PrincipalPoint                  | [324.6594,245.4463]                             |
| Skew                            | 0   |
| MeanReprojectionError           | 0.4512  |
| ReprojectedPoints               | 63x2x31 double                                  |

| cameraParams_UXGA               |   |
|---------------------------------|---|
| 1x1 cameraParameters            |   |
| Property                        | Value   |
| ImageSize                       | [901,1202]  |
| RadialDistortion                | [-0.0307,0.2240]                                    |
| TangentialDistortion            | [0,0]   |
| WorldPoints                     | 63x2 double   |
| WorldUnits                      | 'millimeters'                                       |
| EstimateSkew                    | 0   |
| NumRadialDistortionCoefficients | 2   |
| EstimateTangentialDistortion    | 0   |
| ReprojectionErrors              | 63x2x13 double                                      |
| DetectedKeypoints               | 63x13 logical                                       |
| RotationVectors                 | 13x3 double   |
| K                               | [1.5194e+03,0,803.6632;0,1.5198e+03,601.3306;0,0,1] |
| NumPatterns                     | 13  |
| Intrinsics                      | 1x1 cameraIntrinsics                                |
| PatternExtrinsics               | 13x1 rigidtfom3d                                    |
| FocalLength                     | [1.5194e+03,1.5198e+03]                             |
| PrincipalPoint                  | [803.6632,601.3306]                                 |
| Skew                            | 0   |
| MeanReprojectionError           | 1.4091  |
| ReprojectedPoints               | 63x2x13 double                                      |

# Kamerakalibrierung – Umsetzung in Python

```
import numpy as np

def get_camera_matrix():
    c, Lx, Ly = -1.358, 0.0022, 0.0022
    fx, fy, cx, cy = c/Lx, -c/Ly, 324.6594, 245.4463
    camera_matrix = np.array([[fx, 0, cx], [0, fy, cy], [0, 0, 1]], dtype=np.float32)
    return camera_matrix

def get_dist_coeffs():
    k1, k2, p1, p2, k3 = -0.0154, 0.1551, 0, 0, 0
    dist_coeffs = np.array([k1, k2, p1, p2, k3], dtype=np.float32)
    return dist_coeffs
```

Kalibriermatrix K

$$K = \begin{bmatrix} \frac{c}{L_x} & s & {}^s x_H \\ 0 & \frac{-c}{L_y} & {}^s y_H \\ 0 & 0 & 1 \end{bmatrix}$$

| cameraParams_VGA                |   |
|---------------------------------|---|
| 1x1 cameraParameters            |   |
| Property ▲                      | Value   |
| ImageSize                       | [480,640]                                       |
| RadialDistortion                | [-0.0154,0.1551]                                |
| TangentialDistortion            | [0,0]   |
| WorldPoints                     | 63x2 double                                     |
| WorldUnits                      | 'millimeters'                                   |
| EstimateSkew                    | 0   |
| NumRadialDistortionCoefficients | 2   |
| EstimateTangentialDistortion    | 0   |
| ReprojectionErrors              | 63x2x31 double                                  |
| DetectedKeypoints               | 63x31 logical                                   |
| RotationVectors                 | 31x3 double                                     |
| K                               | [617.4263,0,324.6594;0,617.1984,245.4463;0,0,1] |
| NumPatterns                     | 31  |
| Intrinsics                      | 1x1 cameraIntrinsics                            |
| PatternExtrinsics               | 31x1 rigidtfom3d                                |
| FocalLength                     | [617.4263,617.1984]                             |
| PrincipalPoint                  | [324.6594,245.4463]                             |
| Skew                            | 0   |
| MeanReprojectionError           | 0.4512  |
| ReprojectedPoints               | 63x2x31 double                                  |

# Anpassung der Auflösung in Python

```
ip_address = "192.168.0.156" # ESP32-CAM IP address 192.168.0.156
url = f'http://{ip_address}:81/stream'

# URL for setting resolution to VGA (640x480)
resolution_url = (f'http://{ip_address}/control?var=framesize&val=10")
response = requests.get(resolution_url)
```

