Politechnika Śląska
Katedra Grafiki, Wizji Komputerowej i Systemów Cyfrowych

Zaawansowana Analiza Obrazu, Wideo i Ruchu

Laboratorium 1

Kalibracja kamer

mgr inż. Marcin Paszkuta
marcin.paszkuta@polsl.pl

Gliwice 2022

1 Wstęp

Celem kalibracji jest wyznaczenie parametrów określających zależności między układem podstawowym a układem związanym z kamerą, które występują łącznie z transformacją perspektywy oraz parametrów związanych z kamerą i układem optycznym.

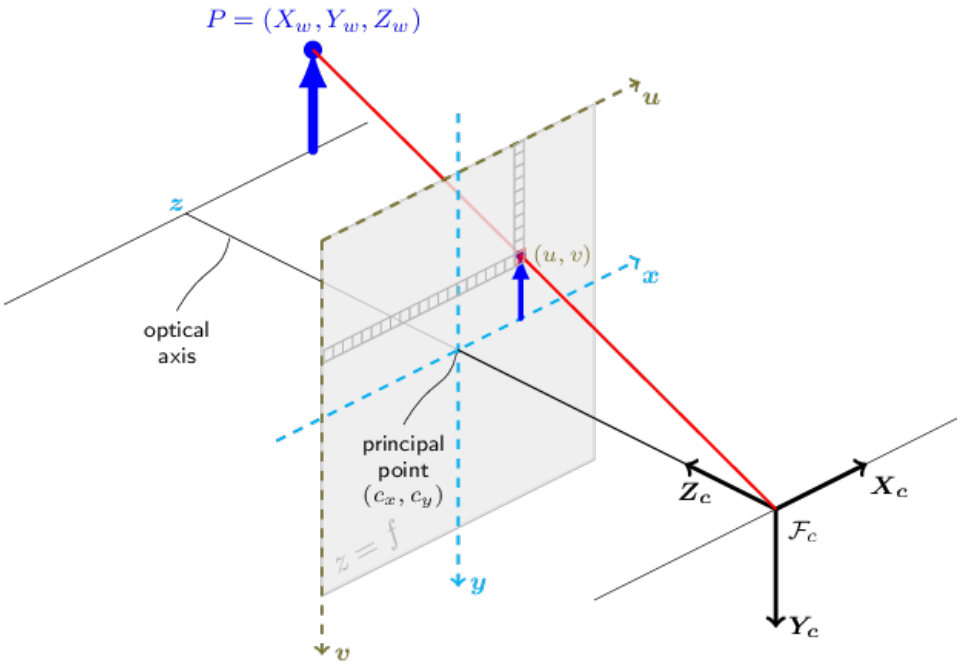
Parametry kamery Ogólnie parametry wyznaczane w procedurze kalibracji kamery dzieli się na dwie grupy:

- **parametry zewnętrzne** (ang. Extrinsic) związane z przesunięciem i rotacją układu kamery względem układu związanego z obserwowaną sceną
- **parametry wewnętrzne** (ang. Intrinsic) określające właściwości optyczne i elektryczne kamery, których liczba zależy od przyjętego modelu układu optycznego i kamery

W przypadku parametrów wewnętrznych oprócz ogniskowej obiektywu modeluje się zazwyczaj zniekształcenia radialne i styczne związane z układem optycznym. Wyznacza się również rzeczywiste współrzędne środka obrazu kamery, które nie muszą leżeć w geometrycznym środku matrycy przetwornika ze względu na skrzywienie osi optycznej obiektywu oraz zniekształcenia liniowe.

The camera intrinsic matrix A is composed of the focal lengths f_x and f_y, which are expressed in pixel units, and the principal point (c_x,c_y), that is usually close to the image center [2]:

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$



Rysunek 1: Model kamery otworkowej [2]

Real lenses usually have some distortion, mostly radial distortion, and slight tangential distortion. So, the above model is extended as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x x'' + c_x \\ f_y y'' + c_y \end{bmatrix} \tag{2}$$

where

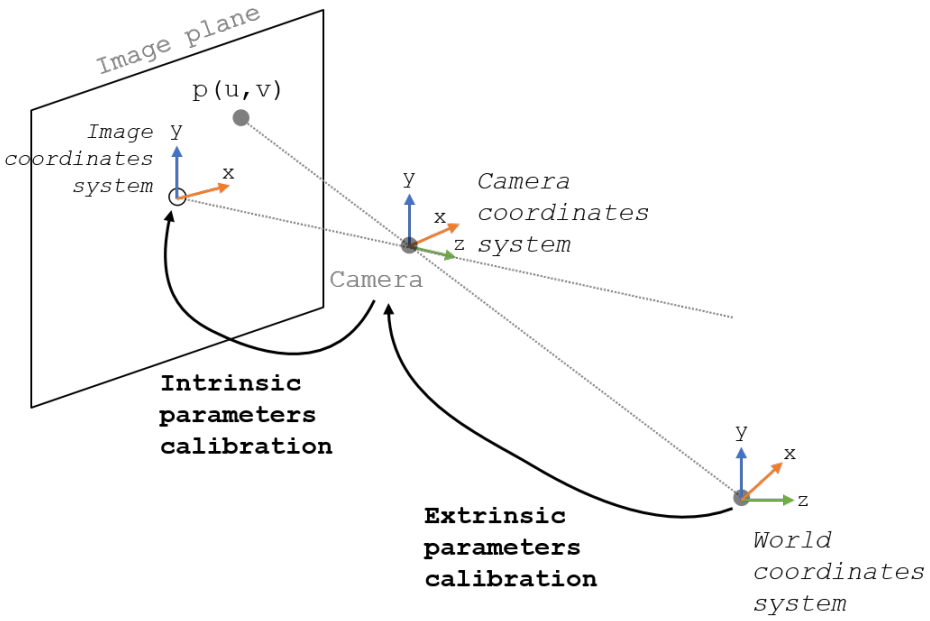
$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2(r^2+2x'^2) + s_1 r^2 + s_2 r^4 \\ y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1(r^2+2y'^2) + 2p_2 x' y' + s_3 r^2 + s_4 r^4 \end{bmatrix} \tag{3}$$

with

$$r^2 = x'^2 + y'^2 \tag{4}$$

and

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}, \tag{5}$$

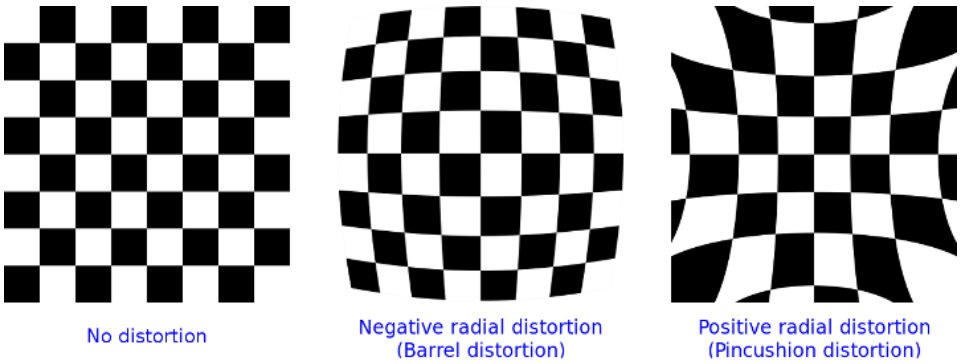


Rysunek 2: Intrinsic and extrinsic camera parameters [1]

if $Z_c \neq 0$. The distortion parameters are the radial coefficients k_1, k_2, k_3, k_4, k_5, and k_6 , p_1 and p_2 are the tangential distortion coefficients, and s_1, s_2, s_3, and s_4, are the thin prism distortion coefficients.

The next figures show two common types of radial distortion: barrel distortion ($1 + k_1r_2 + k_2r_4 + k_3r_6$ monotonically decreasing) and pincushion distortion ($1 + k_1r_2 + k_2r_4 + k_3r_6$ monotonically increasing).

More details you can find in OpenCV documentation - Camera Calibration and 3D Reconstruction [2].



Rysunek 3: Distortion examples [2]

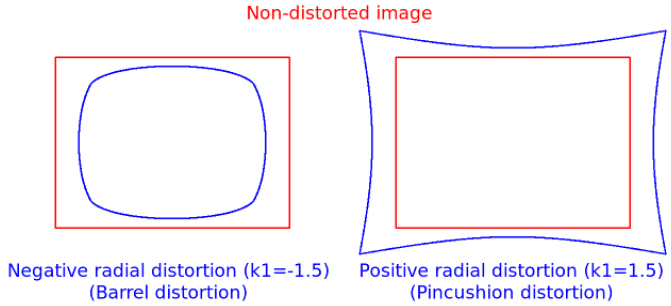
2 Zadania do wykonania

2.1 Wykrywanie wzorca kalibracyjnego na obrazie

Napisz program umożliwiający wykrycie na zdjęciu tablicy kalibracyjnej o określonym kształcie. Do wykrycia wzorca kalibracyjnego możesz skorzystać z biblioteki OpenCV. Metoda `findChessboardCorners` umożliwia wykrycie szachownicy o określonym rozmiarze. Program powinien umożliwić definiowanie wymiarów tablicy kalibracyjnej. Aplikacja powinna wczytywać do analizy wszystkie zdjęcia ze wskazanego folderu, a na wyjściu zwrócić numery zdjęć na których widoczny jest określony wzorzec kalibracyjny. Przetestuj aplikację na wskazanych przez prowadzącego obrazach. W sprawozdaniu zamieść listę z numerami zdjęć osobno dla lewej i prawej kamery.

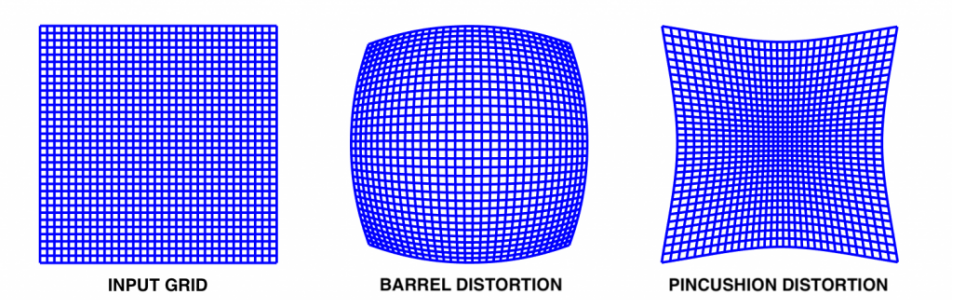
2.2 Wyznaczanie macierzy parametrów wewnętrznej

Napisz program umożliwiający wyznaczenie macierzy parametrów wewnętrznych oraz współczynników dystorsji. Skorzystaj z metody `findChessboardCorners` do wstępnego wykrycia narożników szachownicy oraz z metody `cornerSubPix`, która



Rysunek 4: Distortion examples [2]

Rysunek 5: Checkerboard dimensions



Rysunek 6: Rodzaje dystorsji

umożliwia wykrycie na obrazie narożników szachownicy z sub-pikselową dokładnością. Do wyznaczenia parametrów kalibracyjnych, możesz skorzystać z metody `calibrateCamera`. Metoda przyjmuje tablicę z wykrytymi współrzędnymi narożników na poszczególnych obrazach oraz macierz zawierającą współrzędne wierzchołków tablicy kalibracyjnej. Współrzędne w macierzy powinny uwzględniać rzeczywisty wymiar tablicy kalibracyjnej. W sprawozdaniu zamieść osobno wyniki kalibracji uzyskane na podstawie współrzędnych wykrytych przy użyciu metody `findChessboardCorners` oraz metody `cornerSubPix`.

2.3 Średni błąd reprojekcji

Bazując na wynikach kalibracji uzyskanych w poprzednim zadaniu, oblicz średni błąd reprojekcji. Skorzystaj z metody `projectPoints` oraz `norm`.

```
mean_error = 0
for i in range(len(objpoints)):
    imgpoints2, _ = cv.projectPoints(objpoints[i], rvecs[i], tvecs[i], mtx, dist)
```

```
    error = cv.norm(imgpoints[i], imgpoints2, cv.NORM_L2)/len(imgpoints2)
    mean_error += error
print("Mean reprojection error: {}", mean_error/len(objpoints))
```

2.4 Zapis parametrów kalibracyjnych

Rozszerz aplikację o możliwość zapisu parametrów kalibracyjnych do pliku w formacie JSON.

2.5 Usuwanie dystorsji na obrazie - metoda `undistort`

Rozszerz aplikację o możliwość odczytu parametrów kalibracyjnych z pliku JSON. Bazując na danych kalibracyjnych odczytanych z pliku JSON, usuń dystorsję korzystając z metody `undistort`.

```
# load image
img = cv.imread('image.png')
h,  w = img.shape[:2]
newcameramtx, roi = cv.getOptimalNewCameraMatrix(mtx, dist, (w,h), 1, (w,h))

# undistort
dst = cv.undistort(img, mtx, dist, None, newcameramtx)
# crop the image
x, y, w, h = roi
dst = dst[y:y+h, x:x+w]
cv.imwrite('calibration.png', dst)
```

2.6 Usuwanie dystorsji na obrazie - metoda `initUndistortRectifyMap` i `remap`

Bazując na danych kalibracyjnych odczytanych z pliku JSON, spróbuj usunąć dystorsję korzystając z metody `remap`. Do wyznaczenia parametrów `mapx` i `mapy` skorzystaj z metody `initUndistortRectifyMap`.

```
# load image
img = cv.imread('image.png')
h,  w = img.shape[:2]
newcameramtx, roi = cv.getOptimalNewCameraMatrix(mtx, dist, (w,h), 1, (w,h))

# undistort
mapx, mapy = cv.initUndistortRectifyMap(mtx, dist, None, newcameramtx, (w,h), 5)
dst = cv.remap(img, mapx, mapy, cv.INTER_LINEAR)
# crop the image
x, y, w, h = roi
dst = dst[y:y+h, x:x+w]
cv.imwrite('calibrated.png', dst)
```

2 Bibliografia

[1] University College London. *Intrinsic camera parameters calibration*. URL: `https://mphy0026.readthedocs.io/en/latest/calibration/camera_calibration.html`. (accessed: 24.05.2021).

[2] OpenCV. *Camera Calibration and 3D Reconstruction*. URL: `https://docs.opencv.org/master/d9/d0c/group__calib3d.html`. (accessed: 24.05.2021).