

# Relatório sobre o código de calibração de câmera utilizando a biblioteca OpenCV

Jean Carlo Sanchuki Filho

## Código

```
import cv2
import numpy as np
import glob
```

```
pattern_size = (10, 7) # number of inner corners
```

```
objp = np.zeros((np.prod(pattern_size), 3), np.float32)
objp[:, :2] = np.mgrid[0:pattern_size[0], 0:pattern_size[1]].T.reshape(-1, 2)

# Define arrays to store the object points and image points
obj_points = [] # 3D points in real world space
img_points = [] # 2D points in image plane
gray = None

# Read the chessboard images and collect the object points and image points
images = glob.glob('Documentos/Faculdade/cv-class/cv2/imgs/*.png')
for fname in images:
    img = cv2.imread(fname)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    ret, corners = cv2.findChessboardCorners(gray, pattern_size, None)

    if ret:
        obj_points.append(objp)
        img_points.append(corners)

# Calibrate the camera
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(obj_points, img_points, gray.shape[:: -1], None, None)

# Print the camera matrix and distortion coefficients
print('Camera Matrix:')
print(mtx)
print('Distortion Coefficients:')
print(dist)
```

```
Camera Matrix:
[[603.88914335   0.         328.23822976]
 [  0.         603.67911565 233.89796492]
 [  0.          0.          1.         ]]
Distortion Coefficients:
[[-2.45324350e-01  1.01086219e+00  1.41019156e-03 -7.13725801e-03
  -1.44280523e+00]]
```

## Resumo

Este código realiza a calibração da câmera detectando os cantos de um padrão de xadrez em um conjunto de imagens, computando os correspondentes pontos do objeto 3D e pontos da imagem 2D, e usando essas informações para calibrar a câmera.

## Descrição do código

Este código realiza a calibração da câmera usando a biblioteca OpenCV em Python.

Primeiro criamos uma matriz de zeros. As duas primeiras colunas de cada linha são preenchidas com as coordenadas x e y dos cantos, respectivamente.

Depois, duas listas vazias, que são definidas para armazenar os pontos do objeto 3D e seus respectivos pontos de imagem 2D, respectivamente. Para então lermos o conjunto de imagens do xadrez em um diretório.

A função "cv2.findChessboardCorners" é usada para encontrar os cantos do tabuleiro de xadrez em cada imagem. A função retorna um valor booleano indicando se os cantos foram encontrados e uma array com as localizações dos cantos. Caso encontre os cantos, os pontos do objeto e os pontos da imagem são adicionados às suas respectivas listas.

Após iterar sobre todas as imagens, a câmera é calibrada usando a função "cv2.calibrateCamera". Esta função recebe como entrada os pontos do objeto, pontos da imagem, tamanho da imagem e outros parâmetros opcionais, e retorna a matriz da câmera, coeficientes de distorção e vetores de rotação e translação.

Por fim, a matriz da câmera e os coeficientes de distorção são impressos no console.