

opencv相机标定-孙佳伟 (202122060713)

项目地址: https://github.com/sinary-sys/computer_vision/tree/master/first_work/code


opencv相机标定-孙佳伟 (202122060713)

- 1、使用glob模块将所有的图片读进来
- 2、标定棋盘格角点检测
- 3、标定、去畸变
- 4、输出结果

本次相机标定采用的设备



OpenCV使用棋盘格板进行标定（如下图），为了标定相机，我们需要输入一系列三维点和它们对应的二维图像点。在黑白相间的棋盘格上，二维图像点很容易通过角点检测找到。而对于真实世界中的三维点呢？由于我们采集中，是将相机放在一个地方，而将棋盘格定标板进行移动变换不同的位置，然后对其进行拍摄。所以我们需要知道 (X,Y,Z) 的值。但是简单来说，我们定义棋盘格所在平面为 XY 平面，即 $Z=0$ 。对于定标板来说，我们可以知道棋盘格的方块尺寸。

 WIN_20211006_13_51_31_Pro

1、使用glob模块将所有的图片读进来

```
images = glob.glob('../pictures/*.jpg')
```

2、标定棋盘格角点检测

cv2.cornerSubPix 函数实现代码：

```
corners = cv.cornerSubPix( image, corners, winSize, zeroZone, criteria )
```

参数：

image：输入图像，8位或者float型。

corners：角点初始坐标。

winSize：搜索窗口为 $2*winSize+1$ 。

zerozone：死区，不计算区域，避免自相关矩阵的奇异性。没有死区，参数为 (-1, -1)

criteria：求角点的迭代终止条件。

返回值：

corner：角点位置。

显示角点位置：`cv.drawChessboardCorners`

```
image =cv.drawChessboardCorners(image, patternSize, corners, patternWasFound
)
```

patternWasFound：标志位，检测是否所有board都被检测到，若为是，则将角点连线，否则不连线。

3、标定、去畸变

通过上面的步骤，我们得到了用于标定的三维点和与其对应的图像上的二维点对。我们使用 `cv2.calibrateCamera()` 进行标定，这个函数会返回标定结果、相机的内参数矩阵、畸变系数、旋转矩阵和平移向量。

```
retval, cameraMatrix, distCoeffs, rvecs, tvecs =
    cv.calibrateCamera(
        objectPoints, imagePoints, imageSize,
        cameraMatrix,
        distCoeffs[, rvecs[, tvecs[, flags[,
        criteria]]])
```

objectPoints：世界坐标系里的位置。

imagePoints：像素坐标。

imageSize：为图像的像素尺寸大小。

cameraMatrix：3*3矩阵，相机内参数矩阵。

disCoeffs：畸变矩阵

rvecs：旋转向量

tvecs：位移向量

flags：标定采用的算法

criteria：迭代终止条件设定。

纠正图像：

我们已经得到了相机内参和畸变系数，在将图像去畸变之前，我们还可以使用

`cv.getOptimalNewCameraMatrix()` 优化内参数和畸变系数，通过设定自由比例因子alpha。当

alpha设为0的时候，将会返回一个剪裁过的将去畸变后不想要的像素去掉的内参数和畸变系数；当alpha设为1的时候，将会返回一个包含额外黑色像素点的内参数和畸变系数，并返回一个ROI用于将其剪裁掉。

```
retval, validPixROI = cv.getOptimalNewCameraMatrix(  
    cameraMatrix, distCoeffs, imageSize,  
    alpha[, newImgSize[, centerPrincipalPoint]] )
```

`imageSize`：原始图像尺寸。

`newImageSize`：校正后图像尺寸。

`alpha`：取0或1。

`centerPrincipalPoint`：是否作用于中心，默认为opencv自己根据图像选择位置。

**** 去畸变方法有两个： ****

1、使用cv2.undistort()

这是一个最直接的办法，只用直接调用函数就可以得到去畸变的图像，使用上面的ROI可以对其进行剪裁，代码如下：

```
dst = cv.undistort( src, cameraMatrix, distCoeffs[, dst[, newCameraMatrix]] )
```

`src`：待矫正图片。

`dst`：输出图像。

2、使用 remapping：

这是一个分两步的方法，首先计算一个从畸变图像到非畸变图像的映射，然后使用这个映射关系对图像进行去畸变。

代码如下：

```
# undistort  
  
mapx,mapy = cv2.initUndistortRectifyMap(mtx,dist, None, newcameramt, (w,h), 5)  
dst = cv2.remap(img,mapx,mapy,cv2.INTER_LINEAR)  
  
# crop the image  
  
x,y,w,h = roi  
dst = dst[y:y+h, x:x+w]  
cv2.imwrite('1.jpg',dst)
```

4、输出结果

```
ret: 0.9550921084194115  
mtx:  
[[1.48952846e+03 0.00000000e+00 9.04436206e+02]  
 [0.00000000e+00 1.48527865e+03 5.38208264e+02]  
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]  
dist:  
[[-0.33080664 -0.23699863 -0.00175964 0.00461672 0.54447489]]  
rvecs:  
[array([[-0.11655587],  
        [-0.03814116],  
        [-0.00874828]]), array([[ 0.04632945],  
        [ 0.09284195],  
        [-0.05496844]]), array([[ 0.0636013 ],  
        [-0.06966733],
```

```

[-0.04221397]]), array([[ -0.18583483],
[-0.10849261],
[ 0.00823603]]), array([[ 0.0094372 ],
[ 0.05146073],
[-0.04625437]]), array([[ 0.0481083 ],
[-0.15101888],
[-0.01545515]]), array([[ 0.06008197],
[ 0.03979738],
[-0.02199297]]), array([[ -0.133961 ],
[-0.06585384],
[-0.13184593]]), array([[ -0.00191594],
[-0.00977471],
[ 0.09019472]]))
tvecs:
[array([[ -1.3932382 ],
[-1.50223651],
[ 8.17418401]])], array([[ -1.49472247],
[-1.29357848],
[ 6.94283337]])], array([[ -3.35874602],
[-1.34256709],
[ 7.18012041]])], array([[ -3.03820697],
[-0.74536643],
[ 9.73733111]])], array([[ -2.46127799],
[-1.20645216],
[ 7.26893699]])], array([[ -1.50753026],
[-1.36150628],
[ 6.89296674]])], array([[ -2.65624443],
[-1.7728709 ],
[ 7.9401901 ]])], array([[ -2.65598312],
[-1.81973848],
[ 8.45890728]])], array([[ -2.60259099],
[-1.28842331],
[ 7.87713741]]))
total error: 0.19087577281727852

```

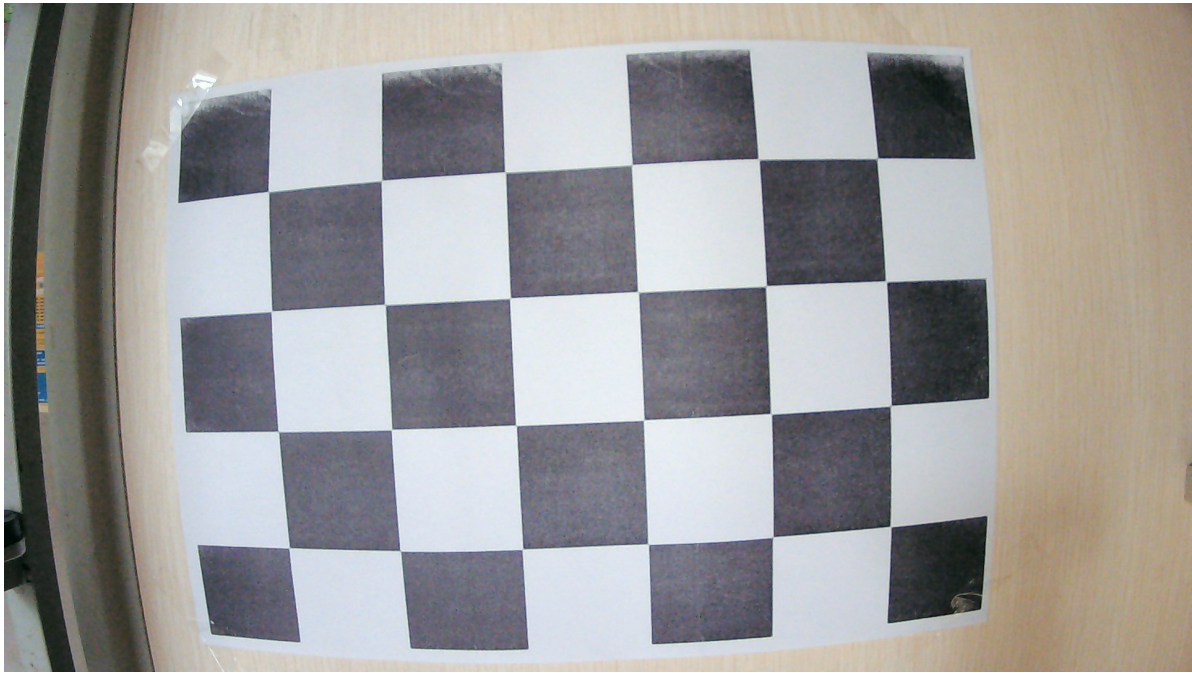
mtx: 内参数矩阵

dist: 畸变系数

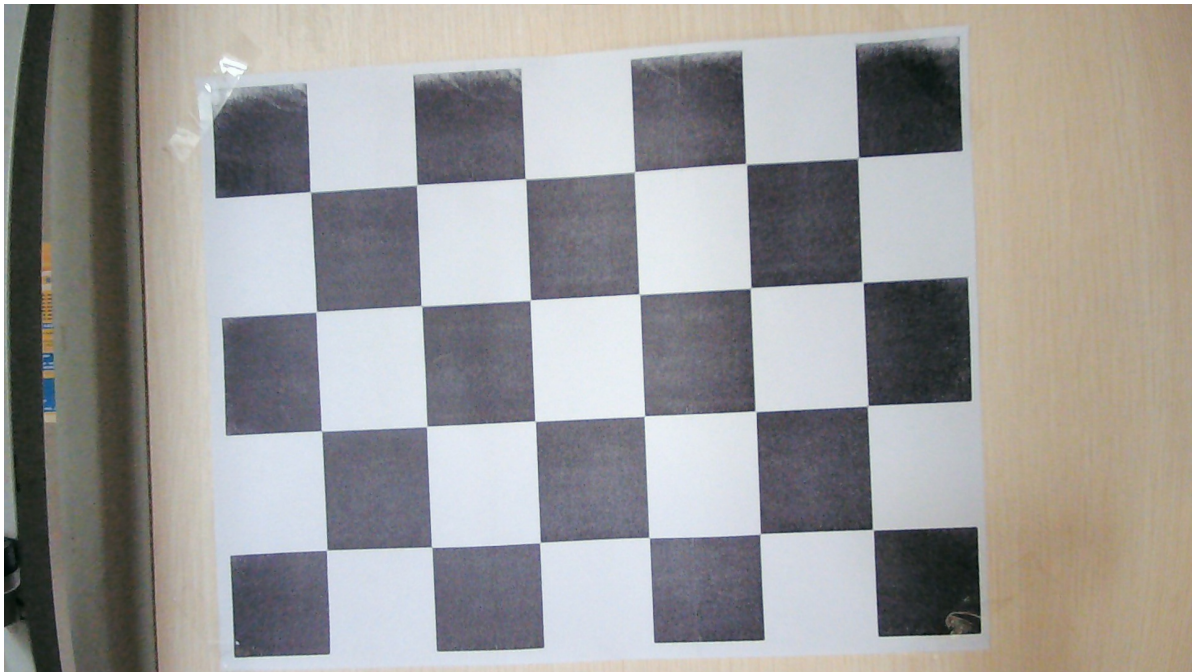
rvecs: 旋转向量 (外参数)

tvecs: 平移向量 (外参数)

校正前的图片



校正后的图片



演示视频