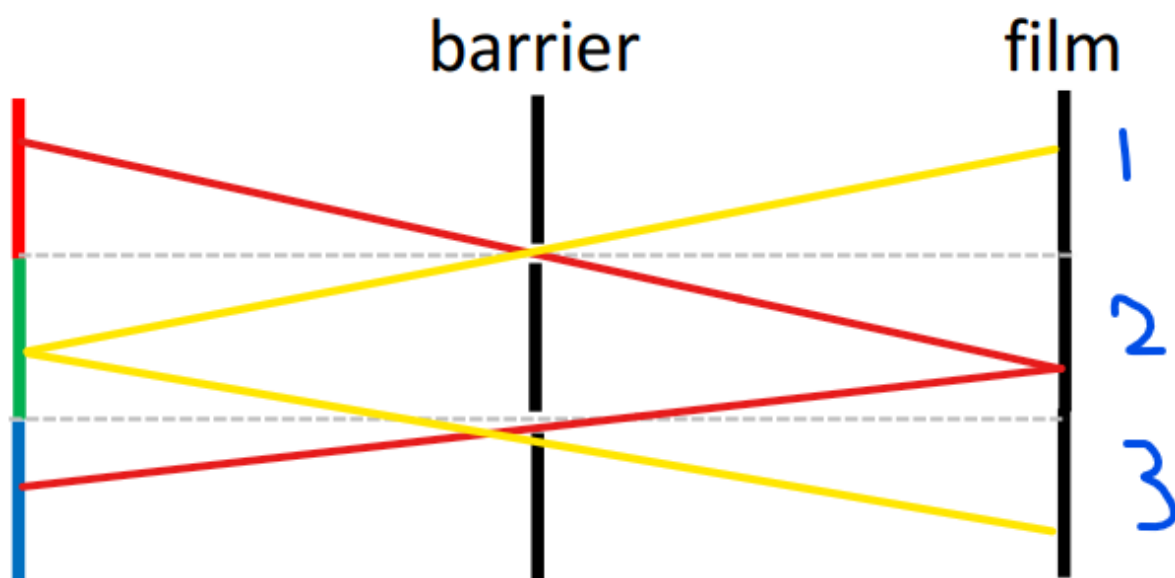


تمرین 2

سینا اسکندری 97521054

(1)



همانطور که در تصویر مشخص است در قسمت 1 و 3 فیلم فقط از قسمت سبز رنگ نور می آید پس این قسمت ها سبز می شوند. در قسمت 2 از قرمز و آبی نور می رسد پس در این قسمت ترکیب این 2 رنگ یعنی بنفش قرار می گیرد. اگر طول فیلم و شی بزرگتر بود در قسمت 1 و 3 نیز شاهد ترکیب رنگ ها می بودیم ولی در حال حاضر از رنگ های دیگر بر این قسمت نوری نمی تابد.

(2)

به علت رخ دادن پدیده depth of field می توان نتیجه گرفت که دوربین لنز دار است. چون قسمت نزدیک تر واضح است یعنی فاصله صفحه دوربین تا لنز بیشتر است و اگر آن را کم کنیم بقیه قسمت ها که تار هستند شفاف می شوند و قسمتی که الان شفاف است، تار می شود.

برای بهبود کیفیت می توان از دریچه استفاده کرد.

(3)

$$\begin{aligned}
 f &= 10 \\
 u + v &= 40 \rightarrow u = 40 \\
 v &= 10 \\
 \frac{1}{f} &= \frac{1}{u} + \frac{1}{v} \Rightarrow \frac{1}{10} = \frac{1}{40} + \frac{1}{v} \Rightarrow \frac{1}{v} = \frac{1}{10} - \frac{1}{40} = \frac{1}{15} \\
 \Rightarrow v &= 15 \rightarrow \text{ایده آل}
 \end{aligned}$$

ابتدا با استفاده از u و f فاصله ایده آل v را بدست می آوریم که برابر 15 می شود یعنی فاصله فیلم تا لنز باید 15 باشد که از فاصله فعلی لنز تا فیلم بیشتر است و تصویری تار ثبت می شود.

برای کیفیت بهتر چون فاصله فیلم تا شی یعنی $u+v$ ثابت است از روابط زیر استفاده می کنیم و u و v را محاسبه می کنیم.

$$\begin{aligned}
 \frac{1}{f} &= \frac{1}{u} + \frac{1}{v} = \frac{u+v}{uv} \\
 f &= \frac{u+v}{uv} \quad f=10 \rightarrow 10 = \frac{40}{uv} \\
 \begin{cases} uv=40 \\ u+v=40 \end{cases} &\rightarrow u + \frac{40}{u} = 40 \rightarrow u^2 - 40u + 40 = 0 \\
 \Rightarrow u &= 20 + 9\sqrt{11} \rightarrow v = 20 - 9\sqrt{11}
 \end{aligned}$$

یعنی اگر u و v را به این مقادیر تغییر بدهیم با وجود ثابت بودن فاصله بینشان تصویری واضح ثبت می شود.

(4)

.1

```

img = cv2.imread('img1.png')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

```

تصویر دارای هر دو اعوجاج شعاعی و مماسی می باشد که اعوجاج شعاعی حاصل از شکل لنز و اعوجاج مماسی حاصل از فرآیند سوار کردن دوربین است.

2.

```
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
objp = np.zeros((24*17,3), np.float32)
objp[:,2] = np.mgrid[0:24,0:17].T.reshape(-1,2)
objpoints = []
imgpoints = []

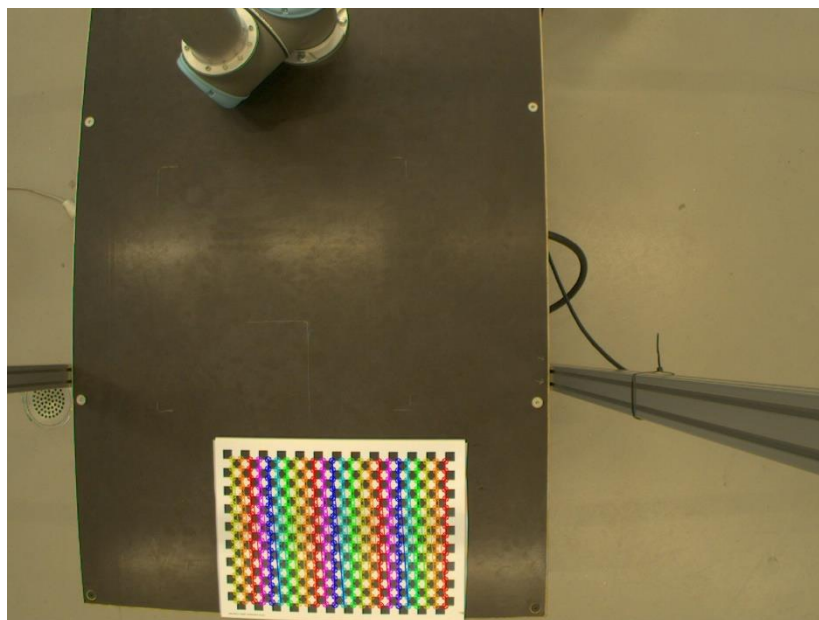
ret, corners = cv2.findChessboardCorners(gray, (17, 24), None)
```

با استفاده از دستور findChessBoardCorners صفحه شطرنجی 17x24 را پیدا می کنیم.
objpoints و imgpoints برای رفع اعوجاج هستند که جلو تر استفاده می کنیم.

3.

```
if ret == True:
    objpoints.append(objp)
    corners2 = cv2.cornerSubPix(gray,corners, (11,11), (-1,-1), criteria)
    imgpoints.append(corners2)
    # Draw and display the corners
    cv2.drawChessboardCorners(img, (17,24), corners2, ret)
    cv2.imshow(img)
```

با استفاده از دستور cornerSubPix کیفیت نقاط را افزایش می دهیم. و با drawChessboardCorners رسم می کنیم.



4 و 5.

```
[12] ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[::-1], None, None)

[13] dist

array([[ 0.06112999, -0.0110414 ,  0.05918889,  0.00035687, -0.11279651]])
```

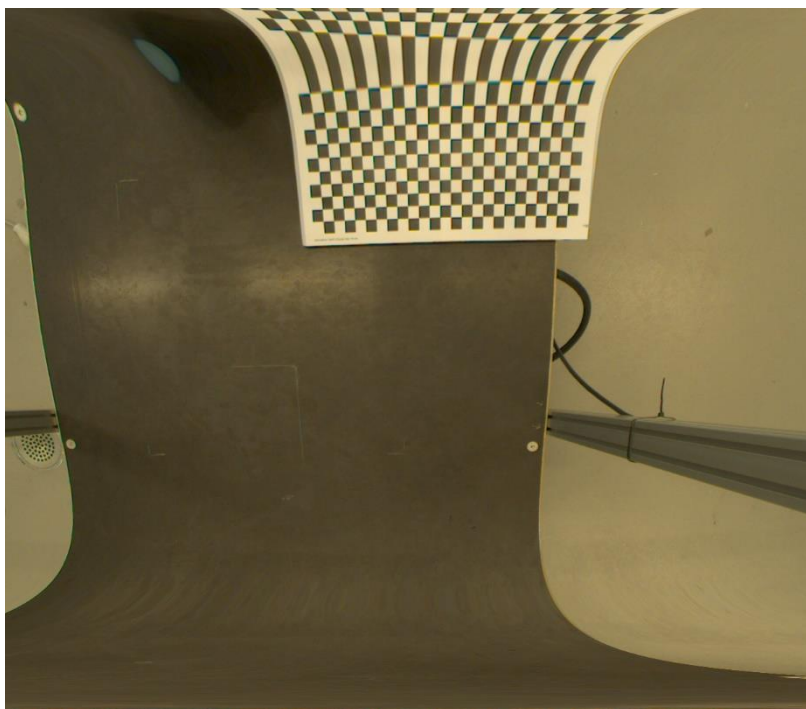
در خط اول دوربین را کالیبره می کنیم objpoints نقاط $(1, 0, 0)$ ، $(2, 0, 0)$ تا آخر هستند برای مشخص کردن نقاط صفحه شطرنجی و imgpoints نقاط صفحه شطرنجی در عکس هستند. متغیر dist نشان دهنده پارامترهای k_1, k_2, k_3 هستند.

6.

```
img = cv2.imread('img5.png')
h, w = img.shape[:2]
newcameramtx, roi = cv2.getOptimalNewCameraMatrix(mtx, dist, (w,h), 1, (w,h))

dst = cv2.undistort(img, mtx, dist, None, newcameramtx)
x, y, w, h = roi
dst = dst[y:y+h, x:x+w]
cv2.imshow(dst)
```

با استفاده از تصویر 5 و تابع getOptimalNewCameraMatrix ماتریس بهینه دوربین را حساب می کنیم (این مرحله اختیاری است). و با استفاده از undistort اعوجاج تصویر را از بین می بریم. 2 خط بعدی برای crop کردن هستند.



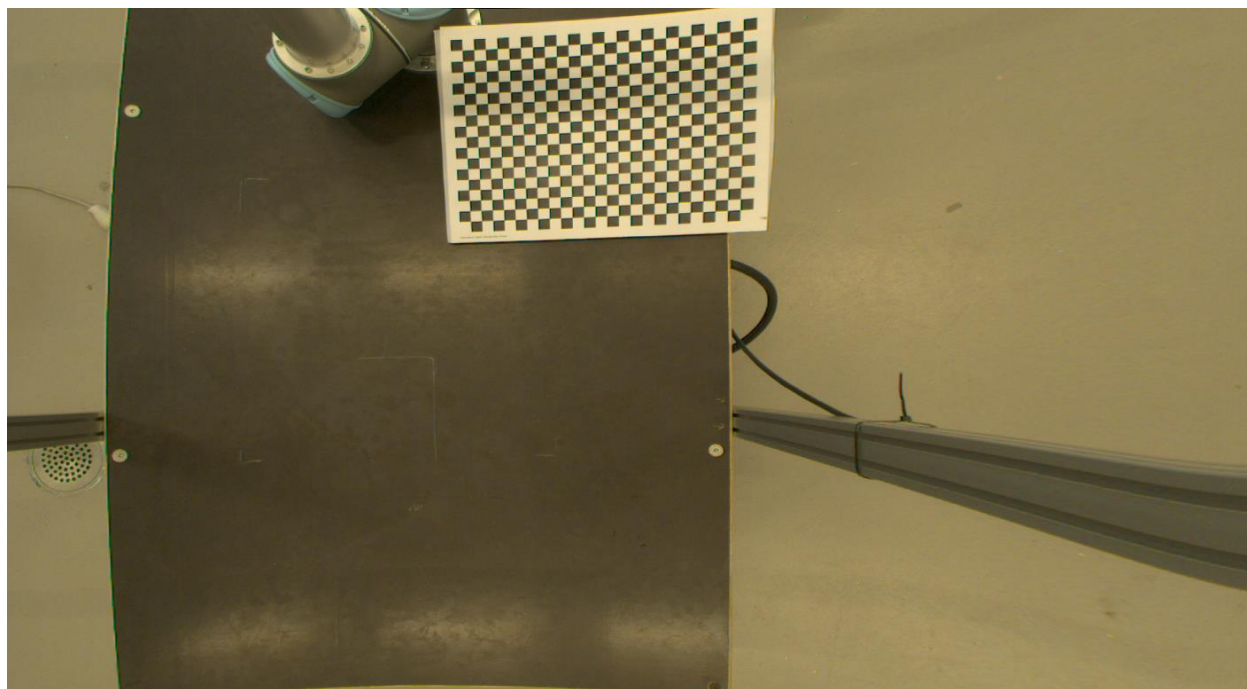
همانطور که از تصویر مشخص است رفع اعوجاج خیلی بد انجام شده و احتیاج به تصاویر بیشتر از زاویه ها بیشتر داریم.

7. مراحل قبل را برای تصاویر 1 تا 4 تکرار می کنیم یعنی به ازای تمام تصاویر objpoints و imgpoints در نظر می گیریم.

```
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
objp = np.zeros((24*17,3), np.float32)
objp[:,2] = np.mgrid[0:24,0:17].T.reshape(-1,2)
objpoints = []
imgpoints = []
imgs = [f'img{i}.png' for i in range(1, 5)]
for f in imgs:
    img = cv2.imread(f)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    ret, corners = cv2.findChessboardCorners(gray, (17, 24), None)
    if ret:
        objpoints.append(objp)
        corners2 = cv2.cornerSubPix(gray,corners, (11,11), (-1,-1), criteria)
        imgpoints.append(corners2)
        img = cv2.drawChessboardCorners(img, (17,24), corners2,ret)
        cv2_imshow(img)
```

نقاط شطرنجی هر تصویر در نوت بوک قابل مشاهده است.

نتیجه حاصل بعد از رفع اعوجاج:



همانطور که مشخص از حالت اول خیلی بهتر اعوجاج از بین رفته ولی هنوز به صورت خوب ایده آل نیست که احتمالا اگر تصاویر بیشتر از زاویه های بیشتر داشتیم، این مورد نیز رفع می شد.

منابع:

https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

•