

تمرین ۷

سینا اسکندری ۹۷۵۲۱۰۵۴

۱-

ابتدا به علت هم اندازه نبودن تصویر اول با بقیه همه را `resize` می کنم سپس طبق توضیحات نوتیوک از `sticher` استفاده می کنم.

```
# read input victoria images and show them in a row together
imgs = [cv2.imread(f'images/victoria{i}.png') for i in range(1, 8)]

plt.figure(figsize=(20, 20))

for i, img in enumerate(imgs):
    imgs[i] = cv2.resize(img, (500, 500))
    plt.subplot(1, len(imgs), i + 1), plt.imshow(cv2.cvtColor(imgs[i], cv2.COLOR_BGR2RGB))
    plt.axis('off')
```

✓ 1.3s Python



```
# initialize OpenCV's image sticher object and then perform the image stitching on input images
stichy = cv2.Stitcher.create()

state, output=stichy.stitch(imgs)

if state != cv2.STITCHER_OK:
    print("stitching is not successful")
else:
    print('stiching is ok')
```

✓ 5.9s Python

stiching is ok

تصویر بدست آمده:



$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} x_1 \cos \theta - y_1 \sin \theta \\ x_1 \sin \theta + y_1 \cos \theta \end{bmatrix}$$

$$\text{Cost} = \sum (x_r^n - x_1^n \cos \theta + y_1^n \sin \theta)^2 + (y_r^n - x_1^n \sin \theta - y_1^n \cos \theta)^2$$

$$\frac{d}{d\theta} \text{Cost} = \sum 2(x_r^n - x_1^n \cos \theta + y_1^n \sin \theta)(x_1^n \sin \theta + y_1^n \cos \theta) \\ + 2(y_r^n - x_1^n \sin \theta - y_1^n \cos \theta)(y_1^n \sin \theta - x_1^n \cos \theta)$$

$$= 2 \sum (x_r^n x_1^n \sin \theta + x_1^n y_1^n \cos \theta + y_r^n y_1^n \sin \theta - y_r^n x_1^n \cos \theta)$$

$$= 2 \sin \theta \sum (x_r^n x_1^n + x_1^n y_1^n \cot \theta + y_r^n y_1^n - y_r^n x_1^n \cot \theta)$$

$$\theta \neq 0 \Rightarrow \sin \theta \neq 0$$

$$\Rightarrow \sum (x_r^n x_1^n + x_1^n y_1^n \cot \theta + y_r^n y_1^n - y_r^n x_1^n \cot \theta) = 0$$

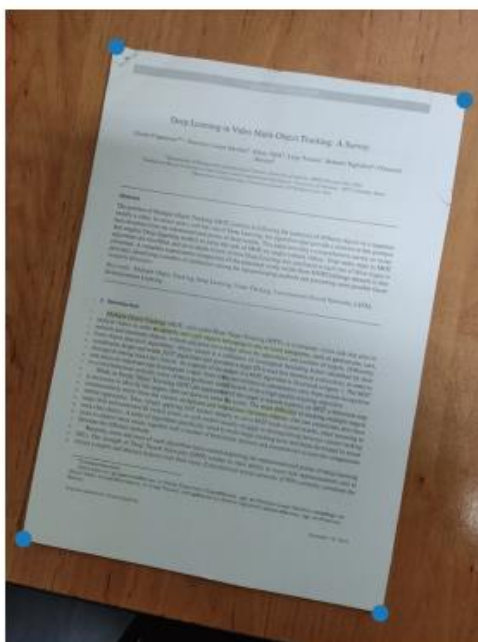
$$\cot \theta (x_r^n y_1^n - x_1^n y_r^n) + x_r^n x_1^n + y_r^n y_1^n = 0$$

$$\theta = \cot^{-1} \left(\frac{\frac{1}{N} \sum \frac{x_r^n x_1^n + y_r^n y_1^n}{x_1^n y_r^n - x_r^n y_1^n}} \right)$$

۳-

مراحل الف تا پ احتیاج به توضیح ندارند و پیاده سازی و خروجی هر کدام در نوتبوک مشخص است. در مرحله ت برای پیدا کردن رئوس طبق راهنمایی نوتبوک از `findContours` استفاده می کنم. ولی نکته قابل توجه این است که با این روش ممکن است بیشتر از ۴ نقطه پیدا شود که فاصله بعضی از آن ها خیلی کم است. در این صورت فاصله نقاط را حساب کردم و اگر فاصله ۲ نقطه کمتر از ۱۰ پیکسل باشد آن نقطه را حذف می کنم.

```
def find_vertices(im):
    # Your code goes here.
    contours, hierarchy = cv2.findContours(im, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    max_area_contour = max(contours, key=cv2.contourArea)
    arc_length = cv2.arcLength(max_area_contour, True)
    corners = cv2.approxPolyDP(max_area_contour, 0.01 * arc_length, True)
    similar_corner_indices = []
    if corners.shape[0] == 4:
        return corners.reshape((4, 2))
    for i in range(corners.shape[0]):
        for j in range(i + 1, corners.shape[0]):
            if np.linalg.norm(corners[i] - corners[j]) < 10:
                similar_corner_indices.append(j)
    corners = np.delete(corners, similar_corner_indices, axis=0)
    return corners.reshape((4, 2))
```



برای نگاشت دورنما نیز به اینصورت عمل می کنیم:

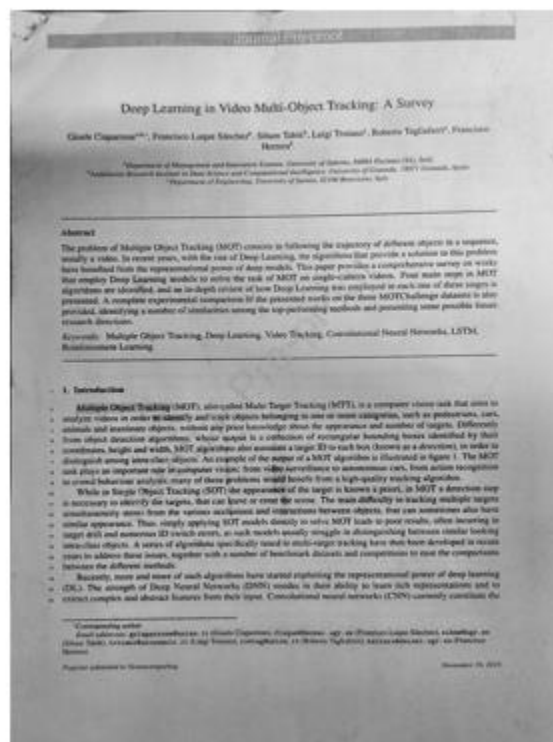
```
def crop_out(im, vertices):
    # Your code goes here.
    height, width = im.shape[0], im.shape[1]
    target = np.float32([[0, 0], [0, height], [width, height], [width, 0]])
    transform = cv2.getPerspectiveTransform(vertices.astype(np.float32), target) # get the top or bird eye view effect
    return cv2.warpPerspective(im, transform, (width, height))
```

✓ 0.4s

Python



برای بهبود تصویر نیز از روش CLAHE استفاده می کنم.
نتیجه کلی:



منابع:

<https://stackoverflow.com/questions/60941012/how-do-i-find-corners-of-a-paper-when-there-are-printed-corners-lines-on-paper-i>

<https://www.geeksforgeeks.org/opencv-panorama-stitching/>