

بسم الله الرحمن الرحيم

دانشگاه علم و صنعت ایران

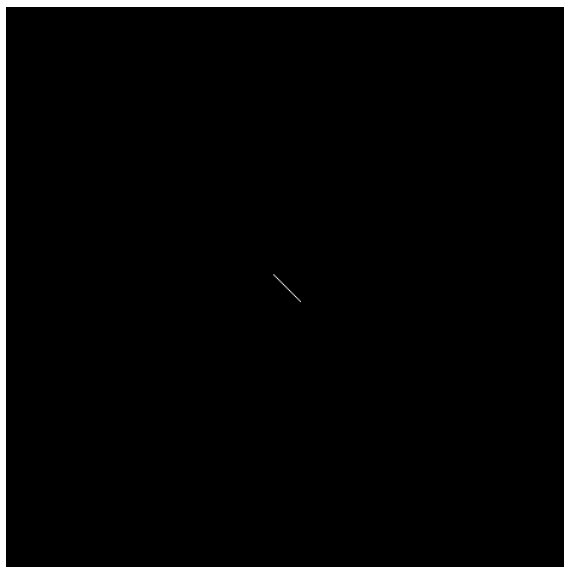
پاییز ۱۳۹۹

تمرین سری پنجم

تصویرپردازی رقمی

تحويل: شنبه ۲۲ آذر

۱. الف) تصویر Khayyam.jpg توسط شخصی ثبت شده است. به دلیل تکان خوردن دست شخص، این تصویر تار ثبت شده است. فرض کنید نویز دوربین بسیار کم و قابل صرف نظر است و مدل حرکت در فضای مکان $h(x, y)$ به شکل psf1.bmp باشد. با استفاده از فیلتر وینر تاری ناشی از حرکت را حذف کنید. (۱۵)



ب) تصویر car.jpg از یک ماشین در حال حرکت ثبت شده است. می خواهیم پلاک ماشین را بخوانیم. در اینجا مدل حرکت دوربین را نمی دانیم. همچنین دوربین مقداری نویز از نوع نامشخص دارد. با استفاده از فیلتر وینر تصویر را بازسازی کرده و پلاک را نمایش دهید. (راهنمایی: شما بایستی میزان و زاویه حرکت دوربین را حدس بزنید و تصویر PSF مربوط به آن را مشابه آنچه در بخش قبل مشاهده کردید بدست آورید. سپس برنامه شما بایستی قابلیت تغییر مقدار K و مشاهده همزمان خروجی را داشته باشد تا بتوانید مقدار K مناسب را بدست آورده و پلاک را بخوانید.) (۲۵)



۲. کد زیر تصویر وب کم را خوانده و نمایش می‌دهد.

```
import cv2
import numpy as np

# if 0 didnt work try different values. ie. 1,2,3,...
cam_id = 0

cap = cv2.VideoCapture(cam_id)

while True:
    ret, I = cap.read()
    cv2.imshow("My Camera", I)

    # Press 'e' to exit
    if cv2.waitKey(1) & 0xFF == ord('e'):
        break

cap.release()
cv2.destroyAllWindows()
```

الف) اندازه گرادیان Sobel هر فریم وبکم را بدست آورید. سپس آن را آستانه‌گذاری کنید تا پیکسل‌هایی که مقادیر بیشتر از آستانه داشته باشند ۲۵۵ و بقیه ۰ در نظر گرفته شوند. (آستانه ۸۰ می‌تواند مناسب باشد). (۱۰)

ب) با استفاده از تابع cv2.Canny لبه‌های تصویر هر فریم را بدست آورید. (آستانه بالا و پایین را می‌توانید به ترتیب ۶۰ و ۱۲۰ در نظر بگیرید). (۵)

* در قسمت الف و ب قبل از اعمال فیلترها حتماً از فیلتر پایین‌گذر برای حذف نویز استفاده کنید. cv2.GaussianBlur

پ) نتایج قسمت الف و ب را با هم مقایسه کنید. آیا می‌توانید مزیت اصلی Canny را به Sobel+Thresholding توضیح دهید؟ این مزیت به چه علت وجود دارد؟ (۱۰)

ت) دست خود را روبروی وبکم با سرعت تکان دهید. چه اتفاقی می‌افتد؟ دلیل آن را توضیح دهید. (۵)

۳. الف) یکی از مشکلات تبدیل Hough عدم استفاده از زاویه گرادیان است. برنامه‌ای بنویسید که خطوط موجود در تصویر را با استفاده از تبدیل Hough استخراج نماید اما در هنگام رای‌گیری از زاویه گرادیان نیز استفاده نماید. برنامه شما مشابه با شبه کد نوشته شده در اسلاید درس خواهد بود و تنها محدوده θ بجای ۰ تا ۱۸۰ باید محدود به $\theta - \Delta\theta$ تا $\theta + \Delta\theta$ شود (توجه کنید که زاویه‌هایی که از بازه ۰ تا ۱۸۰ درجه خارج می‌شوند باید به این بازه تبدیل شوند). کد نوشته شده خود را به ازای مقادیر مختلف $\Delta\theta$ بر روی تصویر comb.jpg اجرا کرده و نتایج را مقایسه کنید. (۳۰)

ب) با استفاده از cv2.HoughCircle و آرگومان های مناسب تعداد سکه ها در تصویر coin.jpg را بشمارید. (۱۰)



۴. الف) الگوریتم Hough و RANSAC را با یکدیگر مقایسه کنید و مزایا و معایب هر کدام نسبت به دیگری را شرح دهید. (سرعت، دقت، حساسیت به نویز و داده پرت) (۱۰)

ب) می‌خواهیم از الگوریتم RANSAC برای یافتن پارامترهای یک بیضی در تصویر استفاده کنیم. در صورتیکه بدانیم تنها ۳۰ درصد از لبه‌های تصویر مربوط به بیضی مورد نظر هستند و بخواهیم با احتمال بالای ۰,۹۹۹۹ به پارامترهای صحیح دست پیدا کنیم، به چند تکرار نیاز است؟ (۱۰)

مراجع

1. https://docs.opencv.org/3.4/d1/dfd/tutorial_motion_deblur_filter.html
2. https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html
3. https://docs.opencv.org/master/da/d22/tutorial_py_canny.html
4. https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html
5. https://docs.opencv.org/3.4/d4/d70/tutorial_hough_circle.html