



دانشکده مهندسی کامپیوتر

تمرین سری ۶ بینایی کامپیوتر

نام درس

مبانی بینایی کامپیوتر

نام دانشجو

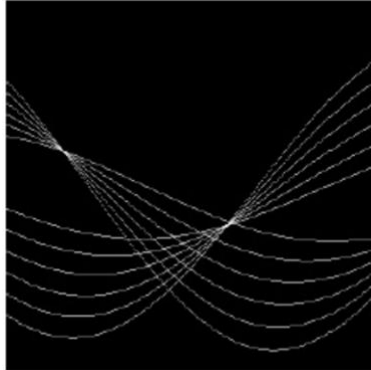
زهرا انوریان

نام استاد درس

دکتر محمدی

پاییز ۱۳۹۹

سوال ۱: تصویر زیر تبدیل هاف مربوط به چه شکلی است؟ علت پاسخ خود را توضیح دهید. (۱۵ نمره)



پاسخ: باتوجه به تعداد خطوط، متوجه می شویم که تصویر دارای ۱۲ نقطه است که با وجود دو نقطه‌ی روشن در تبدیل Hough متوجه وجود دو خط می شویم که از ۶ نقطه تشکیل شده‌اند و همچنین اگر محدوده‌ی θ را -90° تا 90° درجه در نظر بگیریم، این دو خط به طور تقریبی در زاویه‌های -30° و 30° در تصویر قرار دارند و همچنین این دو خط می‌توانند همدیگر را قطع کرده باشند (مانند X) زیرا با توجه به تبدیل Hough، تقریباً در یک نقطه خطی از هردو خط موجود در تصویر اصلی گذشته است (درواقع یک نقطه به هردو خط رای داده است) و یا می‌توانند در یک محلی، خیلی نزدیک به هم قرار داشته باشند که تشخیص آن کاملاً بستگی به میزان رزولوشن (اندازه‌ی پنجره‌های تبدیل هاف) دارد که با فرض رزولوشن کم می‌توان نتیجه گرفت این دو خط همدیگر را قطع می‌کنند.

سوال ۲: می‌خواهیم از الگوریتم RANSAC برای یافتن پارامترهای یک دایره در تصویر استفاده کنیم. در صورتی که بدانیم تنها ۴۰ درصد از لبه‌های تصویر مربوط به دایره مورد نظر است و بخواهیم با احتمال بالای ۰.۹۹ به پارامترهای صحیح دست پیدا کنیم؛ به چند تکرار نیاز است؟ (۱۵ نمره)

پاسخ: اگر w نسبت تعداد نقاط Inlier به تمام نقاط باشد و p احتمال یافتن یک مجموعه از نقاط بدون Outlier باشد، باتوجه به آنکه برای تشخیص دایره نیاز به ۳ نقطه داریم پس احتمال آنکه یک مجموعه کاملاً از نقاط Inlier تشکیل شده باشد برابر است با w^3 پس اگر k تعداد تکرار باشد، احتمال آنکه هیچ مجموعه درستی انتخاب نشده باشد برابر است با:

$$1 - p = (1 - w^3)^k$$

پس اگر از طرفین لگاریتم بگیریم می‌توان تعداد تکرار یعنی k را بدست آورد پس داریم:

$$k = \frac{\log(1 - p)}{\log(1 - w^3)}$$

$$W = 0.4 \Rightarrow w^3 = 0.064$$

$$p = 0.99 \quad k = \frac{\log(1-0.99)}{\log(1-0.064)} = 69.68 \cong 70$$

سوال ۳: می‌دانیم برای تشخیص خط از الگوریتم Hough و LSD استفاده می‌شود. این دو روش را از جنبه‌های مختلف با هم مقایسه کنید. حداقل سه مورد را بررسی کنید. (۱۵ نمره)

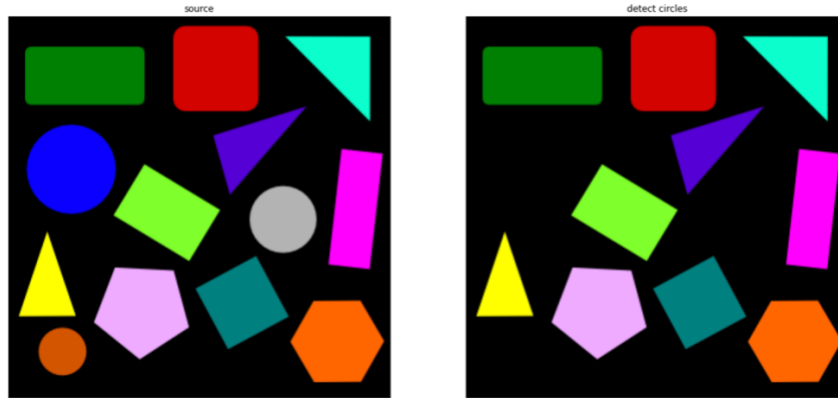
پاسخ: در الگوریتم LSD عکس ورودی از طیف خاکستری است ولی در الگوریتم Hough عکس ورودی به صورت باینری است و همچنین الگوریتم LSD بدون تنظیم هیچگونه پارامتری برای کارکردن طراحی شده است ولی الگوریتم Hough دارای چندین پارامتر (فاصله‌ی رزلوشن، زاویه‌ی رزلوشن، آستانه، حداقل طول خط و حداکثر گپ خط) است و همچنین الگوریتم LSD نتایج دقیقی می‌دهد درحالی‌که در الگوریتم Hough با هر بار اجرا کردن به یک نتیجه‌ی جدید می‌رسیم که دلیل آن، داشتن ماهیت تصادفی است. الگوریتم LSD بیشتر برای تشخیص خط بدون پارامتر مفید و کاربردی است و الگوریتم Hough برای تشخیص خط‌های خاص‌تر به دلیل داشتن پارامترهای مختلف، کاربردی‌تر است زیرا با تنظیم و ترکیب هر یک از آن پارامترها باهم (به طور مثال حداکثر گپ خط و حداقل طول خط) می‌توان خط‌های مورد نظر خود را بدست آورد.

لینک:

<https://stackoverflow.com/questions/43050075/line-segment-detector-vs-probabilistic-hough-transform>

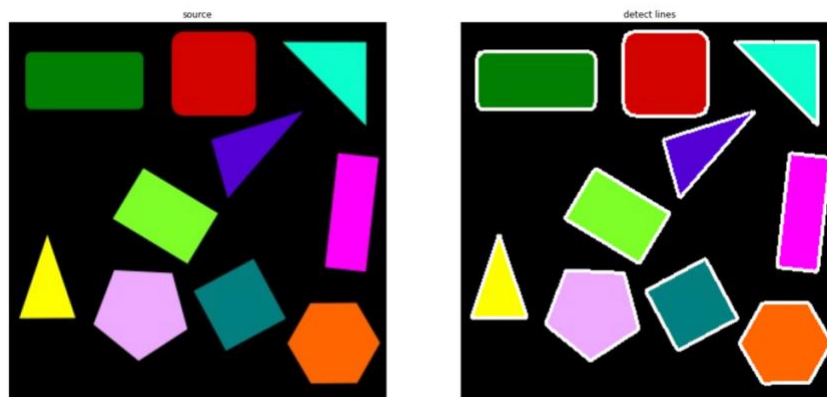
سوال ۴: الف) تصویر 1.png را در نظر بگیرید. دایره‌های موجود در این تصویر را تشخیص دهید و آن‌ها را حذف کنید. برای حل این قسمت از توابع cv2.HoughCircles و cv2.circle استفاده نمایید. (۱۵ نمره)

پاسخ: برای حل این مسئله ابتدا با استفاده از تابع cv2.HoughCircles دایره‌های موجود در تصویر را تشخیص می‌دهیم و سپس با استفاده از تابع cv2.circle دایره‌هایی روی دایره‌های موجود در تصویر به رنگ زمینه (یعنی مشکی) رسم می‌کنیم. لازم به ذکر است که برای حذف کامل دایره‌ها به شعاع آن‌ها ۲ پیکسل اضافه کردم.



ب) تصویر خروجی مرحله قبل (1b.png) را در نظر بگیرید. با استفاده از الگوریتم Hough خطوط موجود در تصویر نامبرده را به دست آورید و با خطوط سفید رنگ بر روی آن نمایش دهید. برای حل این قسمت از توابع `cv2.Canny` و `cv2.HoughLinesP` و `cv2.line` استفاده نمایید. (۱۵ نمره)

پاسخ: برای حل این مسئله ابتدا با استفاده از تابع `cv2.Canny` لبه‌های تصویر را تشخیص می‌دهیم و سپس با استفاده از تابع `cv2.HoughLinesP` لبه‌هایی که خط هستند را تشخیص می‌دهیم و سپس با استفاده از تابع `cv2.line` خط‌هایی به رنگ سفید و ضخامت ۲ بر روی خط‌های تشخیص داده شده رسم می‌کنیم.

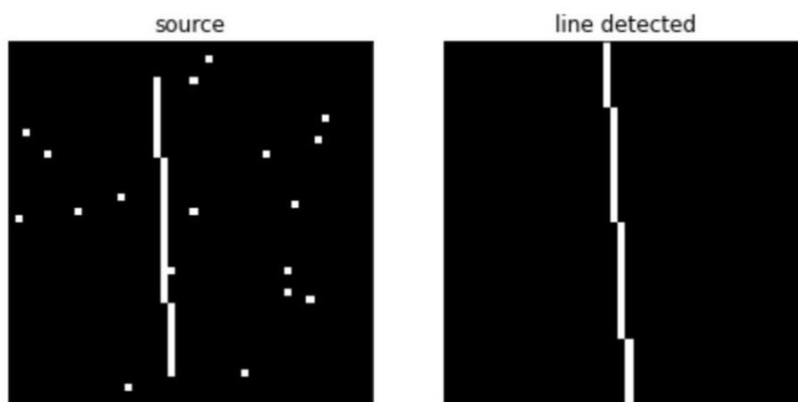


سوال ۵: در این سوال شما باید الگوریتم RANSAC را بدون استفاده از توابع آماده OpenCV پیاده‌سازی کنید و به کمک آن (θ, ρ) خط تشخیص داده شده در تصویر 2.bmp را بیابید. (۲۵ نمره)

پاسخ: برای پیاده‌سازی الگوریتم RANSAC باید ابتدا نقاط سفید رنگ تصویر را بدست آورد و سپس از میان آن‌ها دو نقطه به صورت رندوم انتخاب کنیم و فاصله‌ی خطی که از این دو نقطه می‌گذرد تا نقاط سفید دیگر بدست آوریم و آن نقطه‌هایی که فاصله‌ای کمتر از ۲۰ دارند را جزو نقاط Inlier قرار می‌دهیم و سپس با استفاده از فرمول‌های زیر مقادیر ρ, θ را بدست می‌آوریم و به کمک آن‌ها خط مناسب را رسم می‌کنیم. به دلیل آنکه این الگوریتم بر مبنای نقاط رندوم می‌باشد، با هر بار اجرا شاهد یک نتیجه‌ی متفاوتی هستیم اما بهترین نتیجه در ادامه آورده شده است.

$$\theta = \frac{1}{2} \arctan \left(2 \frac{\overline{xy} - \bar{x}\bar{y}}{\bar{x}^2 - \bar{y}^2 - \bar{x}^2 + \bar{y}^2} \right)$$

$$\rho = \bar{x} \cos \theta + \bar{y} \sin \theta$$



سوال ۶: الگوریتم تبدیل Hough را که نقاط را به فضای (ρ, θ) میبرد، بدون استفاده از توابع آماده OpenCV پیاده‌سازی کرده و به کمک آن خط موجود در تصویر ساختگی زیر (3.bmp) را تشخیص دهید. تصویر حاصل از تبدیل Hough را نیز خروجی بدهید. (۲۵ نمره)

پاسخ: ابتدا باید نقاطی که لبه هستند را شناسایی کنیم و به ازای هر نقطه‌ی لبه و برای هر زاویه‌ی 90° تا 90° باید مقدار ρ را بدست آوریم و به $H(\rho, \theta)$ یک واحد اضافه می‌کنیم، در واقع رای می‌دهیم و در آخر نقطه‌ی ماکزیمم محلی را از H بدست می‌آوریم و با آن ρ, θ خط را رسم می‌کنیم. لازم به ذکر است که این نتیجه شیفت خورده است زیرا ρ را در بازه‌ی 0 تا دو برابر قطر تصویر در نظر گرفتیم درحالیکه باید از منفی دو برابر قطر تصویر تا دو برابر قطر تصویر باشد.

