

به نام خدا

گزارش تمرین 6

نیما کمبرانی ۹۸۵۲۱۴۲۳

مبانی بینایی کامپیوتر

استاد: دکتر محمدی

سوال ۱)
(الف)

برای حل مسئله باید حداقل یکی از ۱۲۰ لبه ی یافته شده برای ضلع وتر را انتخاب کنیم. در نتیجه با توجه به مستقل بودن هر مرحله انتخاب ضلع، در هر مرحله احتمال انتخاب یکی از ۱۲۰ ضلع برابر است با:

$$W = 120 / N = 120 / 360 = 1/3$$

که W احتمال انتخاب یکی از وتر ها و $N = 360$ برابر با تعداد کل لبه های یافته شده در شکل است.

می‌خواهیم با احتمال $P = 90$ درصد مطمئن باشیم که در k بار انتخاب تصادفی اضلاع، حداقل یکی از اضلاع از وتر انتخاب شده باشد. برای بدست آوردن k از متمم P که احتمال انتخاب نشده هیچ یک از اضلاع وتر است، استفاده می‌کنیم. در نتیجه داریم:

$$1-P = (1-W)^k \rightarrow k = \frac{\log(1-P)}{\log(1-W)}$$

پس برای $P=0.9$ داریم:

$$k = 5.6 \approx 6$$

در نتیجه با ۶ بار انتخاب اضلاع با احتمال ۹۰ درصد یکی از اضلاع وتر را انتخاب خواهیم کرد.

(ب)

برای $P = 0.99$ با توجه به فرمول بدست آمده داریم:

$$K = 11.3 \approx 12$$

در نتیجه با ۱۲ بار انتخاب تصادفی اضلاع با احتمال ۹۹ درصد یکی از اضلاع وتر انتخاب خواهد شد.

سوال ۲)
(الف)

با استفاده از تابع پیش‌ساخته `opencv` می‌توان خط‌های یک تصویر را یافت و نمایش داد. برای اینکار پس از خواندن تصویر از حافظه، با استفاده از لبه یاب `canny`، لبه های تصویر را می‌یابیم. در مرحله بعد با استفاده از تابع `hough` و با سطح آستانه ۲۵۰ که برابر حداقل تعداد نقاط رای دهنده به یک خط است، خطوط موجود در تصویر را بدست می‌آوریم. برای رسم خطوط پیدا شده به دو نقطه نیاز است. خروجی تابع `Hough` دو مقدار شیب و فاصله از مبدا خط را به مشخص می‌کند. در نتیجه با استفاده از `sin`, `cos` زاویه شیب خط، می‌توان یک نقطه از خط را پیدا کرد. سپس با توجه به شیب خط و می‌توان دو نقطه با فاصله از ۱۰۰۰ از نقطه ابتدایی در راستای شیب خط یافت که با آن‌ها خط را بیابیم. خروجی این مرحله در شکل ۱ آمده است.

Simple hough Image



شکل ۱. تصویر خروجی تابع `cv2.hough` پس از یافتن خطوط تصویر ابتدایی

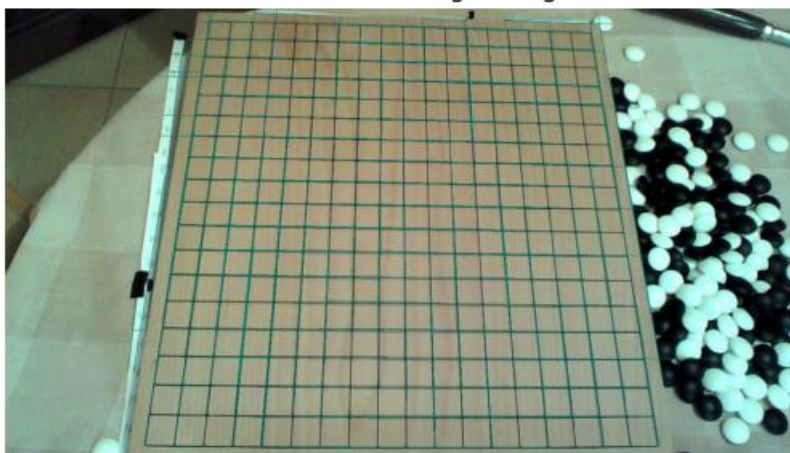
(ب)

در این بخش با استفاده از تابع `HoughLinesP` به جای یافتن خطوط به دنبال پارمخت ها می گردیم. این تابع با ۳ پارامتر ورودی عمل می کند. ۱. سطح آستانه برای تعداد نقاطی که باید به یک پارمخت برای انتخاب رای دهند. ۲. حداقل طول مجاز برای یک پارمخت. ۳. حداکثر فاصله نقاط روی یک خط.

با استفاده از این ۳ پارامتر می توان اندازه و دقت تشخیص خطوط در تصویر را بهبود داد. مقادیر استفاده شده برای این سوال بصورت زیر است که تصویر خروجی برای آن در شکل ۲ آمده است:

`threshold=150, minLineLength=10, maxLineGap=21`

Probabilistic hough Image



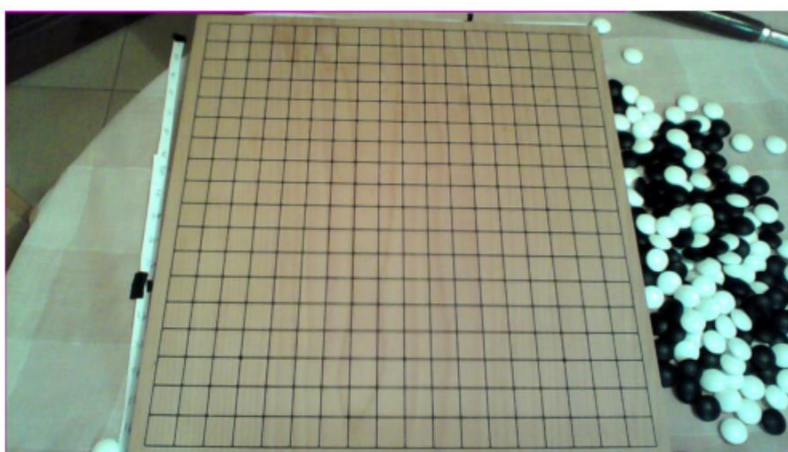
شکل ۲. تصویر خروجی با استفاده از تابع `HoughLinesP`

(ج)

برای بخش امتیازی با توجه به الگوریتم hough، از ۳ مرحله تشکیل شده است:

1. در ابتدا با استفاده از لبه یاب Canny نقاط لبه های داخل تصویر به همراه مختصات آن نقاط بدست می آوریم.
2. پس از یافتن نقاط لبه، به ترتیب برای هر یک از نقاط برای هر یک از درجات از ۰ تا ۱۸۰ مقدار فاصله از مبدا خط بدست آمده در فضای قطبی را بدست می آوریم.
3. برای هر یک از نقاط ۱۸۰ خط بدست آمده است، در نتیجه با شمارش تعداد تکرار هر یک از خطوط، خطوطی که از سطح آستانه بیشتر تکرار شده باشند، به عنوان خط در خروجی نمایش داده می شود.

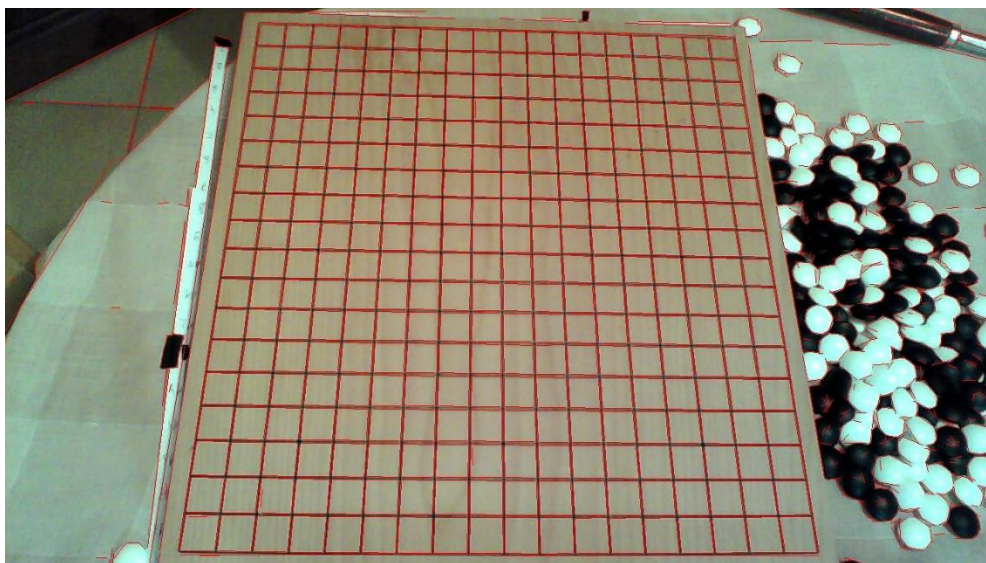
تصویر نتیجه در شکل ۳ آمده است که به درستی قادر به تشخیص خطوط نبود و فاصله از مبدا اکثر خطوط بسیار زیاد و پرت بود.



شکل ۳. تصویر خطوط یافته شده با استفاده از تابع ساخته شده

(سوال ۳)

این قطعه کد با استفاده از الگوریتم LSD که برای یافتن پارمخت با استفاده از جهت گرادیان بجای الگوریتم hough که بدنبال یافتن خط با استفاده از نقاط لبه ها است، به یافتن پارمخت های جدا از هم می پردازد. در الگوریتم hough با توجه به اینکه بدنبال یافتن خط است، اگر سطح آستانه آن بزرگ باشد، تنها خط های طولانی را شناسایی می کند یا اگر سطح آستانه آن کوچک باشد، تعداد زیادی خط های در جهت های مختلف را که ممکن است از وجود چند لبه ی غیر مرتبط بوجود آمده است شناسایی کند. اما در الگوریتم LSD با توجه به محل شروع و پایان لبه ها و جهت گرادیان آن ها قادر به شناسایی خط های کوچک و بزرگ خواهد بود. در شکل ۴ به راحتی محلی که به صورت پارمخت است، قابل تشخیص است.



شکل ۴. تصویر خروجی الگوریتم Line segment detection

سوال ۴)

برای تبدیل cmyk به rgb ابتدا مقیاس اعداد با تقسیم بر ۱۰۰ به بازه ۰ تا ۱ نگاشت می‌شود. سپس از ۳ فرمول زیر استفاده شده است که این مراحل کد پیاده سازی شده است و بدرستی خروجی را بدست می‌آورند.

$$R = 255 \times (1-C) \times (1-K)$$

$$G = 255 \times (1-M) \times (1-K)$$

$$B = 255 \times (1-Y) \times (1-K)$$

برای تبدیل RGB به CMYK مانند حالت قبل ابتدا با تقسیم اعداد بر ۲۵۵ اعداد را به بازه ۰ تا ۱ می‌بریم. سپس با استفاده از فرمول‌های زیر مقدار خروجی را بدست می‌آوریم. در نهایت برای بدست آوردن مقدار خروجی به صورت درصد نتیجه را در ۱۰۰ ضرب می‌کنیم.

$$K = 1 - \max(R', G', B')$$

$$C = (1-R'-K) / (1-K)$$

$$M = (1-G'-K) / (1-K)$$

$$Y = (1-B'-K) / (1-K)$$

سوال ۵)

برای بدست آوردن مقادیر HSI و V و L از فرمول های موجود در اسلاید ها استفاده شد، که مقادیر بدست آمده در ادامه آورده شده است. برای مقدار Y در CrCbY، از فرمول زیر استفاده شد که منبع آن در پایان آمده است.

$$Y = 16 + \frac{65.738R}{256} + \frac{129.057G}{256} + \frac{25.064B}{256}$$

نتایج بدست آمده برای هر یک از پارامتر ها به ترتیب زیر است:

H=4.8609692834254625

S=0.5301204819277108

I=138.3333333334

V=200

L=132.5

Y=107.13725490196079